



CREATING VALUE WITH OPEN SOURCE SOFTWARE

Fernando Almeida, DEEC, Faculty of Engineering of the University of Porto and INESC, almd@fe.up.pt

The session type selected for the paper presentation:

1) “Dialogue” Session X

CREATING VALUE WITH OPEN SOURCE SOFTWARE

Fernando Almeida, DEEC, Faculty of Engineering of the University of Porto and INESC, almd@fe.up.pt

Abstract: The open source process of creation is quickly becoming a threat and also an opportunity to businesses of all kinds. Based on a model of software development where the underlying programming code is open to inspection, modification and redistribution, some new business practices have emerged which several companies have been successfully using in the information systems market. These companies have shifted from a traditional paradigm of software development to a new paradigm where the massive collaboration is the center of the whole process. In this paper, six successful business models that have already been introduced in the market are presented and it is described how they can be embraced by willing software companies.

Keywords: business_model; technology_policy; implementation_methodology; organizational_culture; development model of software

INTRODUCTION

The concept of open source software isn't entirely consensual. To make clear the here-adopted view, we briefly present the notion of proprietary software, free software and open source software, and compare them.

During the last decades, most of the software companies have been used, all the legal means available to protect the intellectual property of their programs, with the purpose to hiding their programs' source codes from other competitor companies. They also used intellectual protection mechanisms, copyright laws and software patents, as far as they could. This kind of software is called propriety software.

Generally, free or open source software refers to a program in which the source code is available to the general public for use and modification from its original design, free of the usual royalties. However, there are some nuances in the understanding of what exactly constitutes open source software. Two of the most reputed organizations concerning this type of software, differ in terminology and scope: the Free Software Foundation (FSF), recommends the use of the expression "free software" and insists on the obligatory publication of all source code derived from original free source code; and the Open Source Initiative (OSI), which prefers the expression "open source software", and is more liberal on the demand of derivative software publication. [4]

In this paper, we will mostly use the OSI view and the expression "open source software". For its protection, every piece of Open Source software

must have a license that guarantees the freedom of use and modification. Its commercial use and how far the protection goes, justifies the existence of several types of licenses.

Another essential condition of the OSI view is the accessibility of the source code. Attending to this requirement, the term "Open Source" Software is most appropriate. One of the reasons that some people dislike the term "Free Software" is because it can easily be misunderstood: the word "free" does not have the meaning as in "free of charge", but as in free of secrets and free of use. Free and Open Source Software may or may not be free of charge, although this is a supplementary effect.

DEVELOPMENT MODEL

The basic idea behind Open Source software is very simple. When programmers can read, modify, and redistribute the software's source code, it evolves. Programmers all over the world can improve the program, adapt it to their personal needs and fix bugs. The experiences of the past years show that this can happen at a much faster speed, compared to the pace of traditional software development. People who are familiar with Open Source software have learned that this rapid evolutionary process produces better software than the conventional closed model, in which only the original programmers can see the source code and everybody else must use a so-called "black box" of bits. Because of the independent peer review and continuous evolutionary improvement and selection, Open Source software can reach levels of reliability, power and general maturity in short time frames that closed products will find hard to attain [1].

Several attempts have been made to explain this phenomenon. One of the first and most popular studies in this field was Eric Raymond's "Cathedral and the Bazaar" published in 1998. [5] Some of Raymond's ideas concerning the successful development of Open Source software will be presented here. The basis of Raymond's study is the assumption that two different categories of software development exist. One development category is called "Cathedral-like development". It is characterized by a centralized management and a strong control on design and implementation. Everything is well organized and planned in detail. This cathedral-like concept is typical of the development of proprietary software. The other development category, so-called "Bazaar-like development", is characterized by informal communication between programmers. Every programmer implements or improves parts of the code and offers it to the others. The programmers are in contact via private e-mail and other current electronic means, getting physically together very rarely. A mailing list and shared and synchronized information space (e.g. by CVS or SVN) for the core developers participating in a certain project are created. They discuss new features, bug fixes, user problems, developments in the Open Source community, etc. in an informal way. The code development takes place on the developer's local machines and new patches or improved source code is then posted to the mailing list and/or published on the appropriate webpage, or made freely available in global repositories. Everyone can then choose and pick the modules he wants, just like a bazaar.

Both cases can be considered as extremes in a continuum of software development models. While Open Source projects can be almost any model in that range, proprietary software project can hardly use models different from the cathedral-like. Whereas the openness of the Internet makes it possible for innovation to come from everywhere, big companies have difficulties gathering all necessary expertise on their teams. Furthermore, each company has only a limited number of software developers at its disposal and its goal is to make a profit with its results. Dividing the work just among its developers and in many cases not even sharing all of it among them, such a company thus cannot benefit from the advantages of the bazaar-like model. The important idea here is that Open Source software has allowed for the first time the introduction of successful alternative styles of software development that turned out to have a very positive influence on programmers' motivation and as a consequence on the quality of software products.

Another interesting point to look is about the driving forces that leads this community.

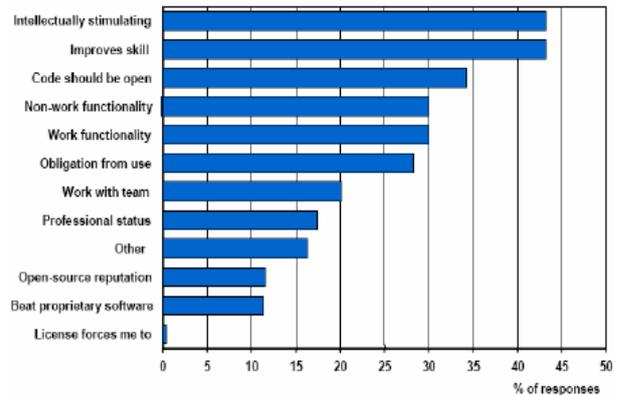


Fig. 1. Motivation to develop Open Source projects [2]

The study conducted by Boston Consulting Group (fig. 1) shows that learning is one of the driving forces that motivate developers to get involved in OSS projects because it provides the intrinsic satisfaction for OSS developers, and the role transformation in OSS communities that go along with learning offers the extrinsic motivation.

MAJOR CHARACTERISTICS

The Open Source projects have some common characteristics between them. These characteristics can also be mentioned as attributes of open source software. Some of these attributes contribute positively to the success of some open source initiatives, namely:

- Development and quality - the development of the software as well as the detection and correction of bugs takes place at a quick pace. The software is generally of very good quality because many excellent programmers are attracted by Open Source projects;
- Reliability and integrity – the code is of very high reliability and integrity because it can be reviewed and improved independently. This reduces the chances of so-called Trojan horse being hidden in the code;
- Stability – Open Source products have been proven to be much more stable than their proprietary counterparts, as was shown, for instance, by scientific analyses carried out by the Computer Science Department of the University of Wisconsin/Madison [2];
- Vendor independence and cost reduction – due to the availability of the source code and the freedom to redistribute the software, the user does not need to be bonded to a single software vendor and, as such, benefits from significant savings on licensing costs;

- Platform independence – the availability of the source code facilities portability to different hardware platforms and the potential to rapidly adapt to changing conditions;
- Support service – the support offered by mailing lists and newsgroups has been shown to be very fast and of high quality. For users who prefer the more traditional commercial support, there are many companies that offer complete solutions and supported distributions of the best-known Open Source products;
- Growing popularity – Open Source products are gaining a rising popularity among users, an increasing market for support services and customization and enjoy a growing attraction for software developers and consultants.

On the other hand, the Open Source concept also has negative aspects, which have to be considered:

- There is no guarantee that development will happen until the project reaches a kind of self-sustaining level. This problem of an initial gap might be solved or at least reduced by the strong backing of one or more big companies. On the other hand the involvement of big companies might keep away some of the idealistic Open Source programmers from joining the project because they only want to work for their ideals and ideologies and not for a big company, which is interested in making profit;
- The motivation for improving and developing an Open Source product might vanish or at least decrease if the product manages to replace proprietary software and gain a position as market leader. On the other hand, this can also have the opposite effect, i.e., the situation might persuade the proprietary software vendor to contribute the source code of some of his products to the Open Source movement;
- There may be significant problems concerning intellectual property rights and software patents because the availability of the source code simplifies the detection of patent infringements by patent holders.

Each company has to weigh up the positive and negative aspects of the use and/or the development of Open Source software in the long run. However, the rapid spread of Open Source in the past years does indicate that Open Source software will, at least in some areas, start to replace the currently common proprietary software products. More and more companies, such as *IBM*, *AT&T* and *Alcatel*, have adopted Open Source at different levels of commitments and

have integrated the Open Source model into their business concept [4].

CULTURAL IMPACTS IN ORGANIZATIONS

The adoption of open source software by an organization brings some issues that we shall give a special attention, to be prepared to deal with them. Neus and Scherf asserted the following:

“Based on our experience, we believe that very few challenges are the result of limits in technology, tools or process – rather, they are often the result of limits and boundaries in people’s minds” [3].

The open source collaboration process is based on widespread access to source code and open collaboration - a meritocratic philosophy that invites feedback from everyone, regardless of official status or formal training, and frequent releases of interim versions to encourage testing, feedback, and quick evolution of solutions.

This open source paradigm is totally opposed to the traditional collaboration models, as we can see in Fig. 2.

Traditional Approach	Open Source Approach
Brooks' Law	Linus' Law
Hierarchy	Network
Experts	Peers
Teams	Communities
Cathedral	Bazaar
Perfection	Improvement
Construction	Evolution

Fig. 2. Traditional vs Open Source paradigms [3]

Based in Fig. 2, we can affirm that the traditional development approach is based on construction and the attempt at perfection. The software is designed once in a “top-down” mode, and then programmers execute the design. This paradigm assumes that only a small and select circle of designated experts should be allowed to design, create, and improve high-quality code or information, thus forcing the vast majority of practitioners into a passive or execution-oriented role, wherein they are expected to provide little or no feedback to improve the overall design.

The open-source paradigm, on the other hand, assumes that quality is the result of massive collaboration. The key of the success of the collaborative development model is based on the lower transaction cost for information, enabling the separation of the identification and solution components of quality problems and spreading these tasks over a much larger population than could sensibly be done in traditional hierarchical approaches.

A change in perspective from the traditional collaboration approach to an open-source approach requires a change in people's minds. The tools, processes, roles, and organizational charts for supporting multisite collaboration in an open-source method exist and are freely available in the Internet. However, these methods can't be directly transposed to an organization, without taking in consideration its own culture.

The visible organization, made up of the organization's formal tools, processes and roles is the tip of the organizational iceberg (Fig. 3). The organization's cultural identity lies "under the waterline". The following definition of culture was developed in the Complex Change Facilitation group and is useful when assessing or changing an organization's culture. Culture is a complex system of shared beliefs, values, language, customs, behaviors, and artifacts that members of a group use to cope with their environment and with one another [3].

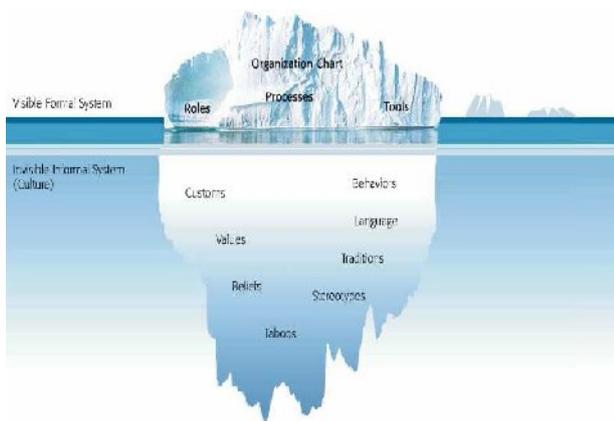


Fig. 3. Iceberg Model of an Organization [3]

OPEN SOURCE BUSINESS MODELS

During the last years several business models have been established with respect to Open Source software. In the following some of the most important models will be presented and related to companies that work according to these models. These examples show that Open Source and business are not in contrast to each other, as many people believe. It is possible to make business and earn money with Open Source. Even more, in the future it might be difficult, at least in the area of operating systems and server solutions, to have economic success with proprietary software. The fact that the European commission and many national governments, such as those of France, Germany, Italy, the USA and China, as well as some leading market research institutes have issued policy documents on the Open Source movement, is forcing software companies to think about business opportunities

offered by the Open Source model [6]. Here we describe the most important and recent Open Source business models.

Support Service Provider Model

In this model, a company (effectively) gives away the software product, but earns money by selling distribution and offering support, after-sales services, course, etc.

One of the most popular companies in this field is *Red Hat*, which was founded in 1994 and floated on the stock market in 1999 (Nasdaq: RHAT). *Red Hat* is the leading provider of Open Source Internet infrastructure solutions, ranging from small-embedded devices to high availability clusters and web servicing. *Red Hat* shares all its software freely with the Open Source community under the GNU general public license (GPL).

Among other Open Source software products *Red Hat* sells various Linux distributions. The official Red Hat Linux OS and related services are available directly from the company or through its partner, distribution and reseller programs, which include top PC and server manufacturers such as *Compaq*, *Dell*, *Gateway*, *IBM*, *Hewlett-Packard* and *Silicon Graphics*. Each Red Hat Linux distribution includes special support packages, e.g., priority online access to software updates for a certain time (30 to 180 days, depending on the distribution) and telephone and/or email-based installation support for 30 to 90 days. They also sell a couple of further and more extensive support packages for their software distributions, which address different kinds of users. Furthermore Red Hat offers training and certification programs on Linux, Open Source software in general, diverse tools and programming languages.

In summary, it can be said that *Red Hat* is an exemplary company which has proven that the business model of a support seller works and that it is possible to have commercial success and make money with Open Source in that way.

Cost Reducers Model

Another business model related to Open Source software is characterized by the aim of reducing costs. In this model Open Source software is not used in order to make profit directly, but to reduce software related costs. For most of the proprietary software products the user has to pay high per-copy fees. Since he has to use the software as a black box, he completely depends on the company which produces this product exclusively. This usually leads to the fact that the user has to buy support from that software company which is, due to the lack of competition,

not only expensive but also often very slow and of poor quality. Furthermore the user often has to buy another distribution for different hardware. Altogether the dependency from proprietary software produces huge costs and little comfort.

With Open Source software the situation is completely different. The charge for an Open Source software distribution is very low because there are competing providers. Furthermore there are no per-copy fees because it is permitted to duplicate the program as often as needed. Since the source code of the software is available, the user has several possibilities for getting support. He can either ask any person who is familiar with the software product, or he can get free support from one of the related newsgroups or he can buy the inexpensive and for the most part very quick and competent support from one of the above mentioned support service providers. Additionally, the availability of the source code also prevents the user from having to buy a new distribution if he wants to change his hardware. He can just adapt the existing software to the new conditions.

Motivated by the urgent necessity to reduce costs the European Commission and some national European governments namely the Germany, France and Italy, recently discovered Open Source software as a possible instrument for cutting down software expenses. The *Information Society Directorate General of the European Commission* recommends the promotion of Open Source software in several categories, e.g., in technical issues, organization and support, legal issues, etc. The European commission should take measures to accelerate the adoption process of Open Source software by European companies, organizations and individuals so that Europe does not miss this important development in the information society [5].

Proprietary Software Promotion Model

The Proprietary Software Promotion Model is another very interesting business model, which was recently introduced by AT&T. In this model, a company goes partially Open Source in order to promote their proprietary software products. The company releases a special distribution of a software product as Open Source. The special feature of this Open Source distribution is that some of the core modules and patches are replaced by older, worse or less efficient versions. For this purpose, the company has to use a special Open Source licence, which allows the company to use the same source code or parts of it in addition to their proprietary software.

The business strategy of this Open Source model is as follows. A company gives away those parts of their source code, which are no longer

competitive because better solutions are already available. With these subtle tactics a company can reach three aims. First of all, it hopes to benefit from the Open Source community's ideas, improvements, bug fixing, etc. Secondly, this is the best possibility of establishing a certain software product as standard software. As a consequence the company can presume that a lot of their Open Source users will not be satisfied with the weaker Open Source product for the long run and buy the higher quality proprietary software product [5].

Developer License Model

Another very interesting business model is the Developer Licence Model which, for example, is used by Object Oriented Concepts, Inc. (OOC). OOC was founded in 1996 to provide CORBA compliant middleware. This software is commercialized under a novel licensing model.

For individual, non-commercial use the source code of the software is free of charge and distributed under the ORBacus Royalty Free Public license. This license allows the user to adopt the program to his personal use and to redistribute the original as well as the modified software if, and this is the decisive restriction, all these activities are non-commercial [6].

If the software is used in connection with the user's business, he has to buy either a run-time or a developer license. As the term implies, the run-time license only allows the license to run the program whereas the developer license also includes the right to modify the source code. However, the user is not allowed to redistribute the software. If he is interested in sharing his improvements with other users, he has to submit his modifications to OOC and ask them for insertion into the original source code. This shows clearly, that this developer license is not an Open Source license in the same of the OSI certification mark, but is a clever way of combining the distributions with per-copy fees (similar to conventional software companies) while at the same time having the possibility of benefiting from users' improvements and modifications.

Effective Advertising Model

This business model aims to improve the public reputation of a company. In this model, a company, the main business of which is not selling software, releases internally used software as Open Source. If the software is not competitively relevant, the company loses nothing by publishing its source code. However, since the current level of media attention for Open Source products is very high, releasing Open Source software seems

to be an excellent opportunity for getting free publicity and for improving the company image.

An example of the technology promotion mentioned above, it was recently introduced into the telecommunications area by *Ericsson*. During the 1980s an internal project at the *Ericsson Computer Science Laboratory* aimed to find out which aspects of computer languages made it easier to program telecommunications systems. The programming language Erlang, the syntax of which is similar to Prolog, was the result of this project at the end of the 80s. Erlang was created by taking those features which made writing of telecommunication systems simpler and avoiding those which made them more complex or error prone. Erlang is particularly suitable for distributed soft real-time concurrent systems. It was designed especially for use in telecommunication applications, e.g., handling mobility in a mobile network or providing unified messaging.

Recently, *Ericsson* decided to publish the source code of Erlang. Since the core business of Ericsson relies on telecommunication products and not on selling software, they had nothing to lose. It was expected that the release of Erlang as Open Source would have a positive advertising effect. However, the advertising success has been marginal, because Erlang is not well known in the telecommunications area. Nevertheless, Ericsson have benefited from the Open Source release in other ways, because some programmers have started to fix bugs thus improving the quality of Erlang.

Start-up Company Promotion Model

Another model that is becoming very popular among start-up companies, usually associated with the so called "new economy", is to start and sponsor an Open Source project and to donate some initial code to it. Two very successful companies which have used this Start-up Company Promotion Model are *Vovida Networks*, which provide their Vovida Open Communication Applications Library (VOCAL) as Open Source, and *Lutris*, which was founded as an Open Source enterprise software and services company [7].

Usually, such companies develop complete solutions and commercial product packages (e.g., soft switches and application servers) based on the Open Source software provided by the community they sponsor. This model is a very good way of getting high quality products with reduced costs and in a short time. Additionally, the companies get free advertisement in the media and gain popularity in the rapidly expanding Open Source community, which is usually associated with the market where these companies operate.

CONCLUSION

Open source software such as the operating system Linux has created much attention as an alternative way to develop and distribute software. Open source is a movement, where communities of highly skilled developers collaborate deeply to create software, often of a quality that outperforms commercial proprietary software. The open source movement has also been said to provide important management lessons regarding the most effective ways to structure and implement innovation. The lessons of open source demonstrate the value of specialization through self-selection and how norms of meritocracy and peer recognition help ensure product quality. Attributes as quality, security, flexibility, reliability, time and cost savings are all benefits being generated by open source development model.

However, before following an open source development model a company must take in consideration not only the tools, processes and roles at the company, but also its organizational culture. The company must be ready to shift from a traditional paradigm of software development to a new paradigm where the massive collaboration is the center of all process.

There are already several business models that prove that it is possible to conduct a successful and profitable business based in Open Source software. We presented six business models and showed that they can be used by software companies.

References

- [1] Kenwood, C., A Business Case Study of Open Source Software, 2001, MITRE Corporation.
- [2] Kooths, S. and Langenfurth, M., Open-Source Software: An Economic Assessment, 2003, MICE Economic Research Studies, Vol. 4.
- [3] Neus, A. and Scherf, P., Opening Minds: Cultural change with the introduction of open-source collaboration methods, 2005, IBM Systems Journal, Vol. 44, No. 2, p. 215-226.
- [4] Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K., Perspectives on Free and Open Source Software, 2007, MIT Press, Vol. 3.
- [5] Raymond, E., The Cathedral & the Bazaar, 2001, O'Reilly Editions.
- [6] Woods, D., Guliani, G., Open Source for the Enterprise, 2005, O'Reilly Editions.
- [7] <http://www.openclovis.com> – accessed on 11/03/2008.
- [8] <http://www.eu.conecta.it> – accessed on 11/03/2008.
- [9] <http://www.omg.org> – accessed on 14/03/2008.
- [10] <http://www.vovida.org> – accessed on 17/03/2008

