

Information Technology for Education Management and Open Source Software

Improving Education Management through Open Source

Jacques Bulchand, Javier Osorio & Jorge Rodríguez

Las Palmas de Gran Canaria University, Spain

Abstract: Open Source Software has received lately a great deal of attention, specially due to its lower cost in comparison to Proprietary Software. In the education area, this is quite important due to economical restrictions. Lately, we have seen different Spanish communities embracing the OSS model following different models. This article begins by examining OSS history, as well as its main strengths and weaknesses. It follows examining the possibilities and advantages of OSS in education and presents three possible ways in which the OSS can be introduced in a territory.

Keywords: Open source, education management.

1. INTRODUCTION

There is no doubt that open source software (OSS) is a fashionable topic at present in the technological sector, together with others such as connectivity ubiquity, the improvement of man-machine interaction or future possibilities of Internet.

Before continuing it is important to establish the fact that open source software is also known as free software, but this expression creates certain confusion between two matters, free access and absence of cost. We live in a money-driven world. In such a context, the public's attention is easily drawn towards free things. Due to this there is a general perception that open source software is interesting because of its low or inexistent cost in comparison to owner software, when actually this is not so. In fact, when Richard Stallman, one of the founders of the open source software movement, first approached the subject its economic aspects were not even mentioned.

In this article first of all we will set out the fundamentals of the very concept of open source software, its main advantages and disadvantages and

its present situation in general terms. Having done this, we will look into the possibilities open source software offers in educational environments and how it can be applied in such environments.

We finish by proposing three possible models in which the decisions to implement OSS in a Community can be taken: a top-down model, a citizen pressure group model and a combined model.

2. OPEN SOURCE SOFTWARE

The first matter we must clarify is what open source software is. We will use the definition given by Richard Stallman, considered the founder of OSS. According to this author, OSS must comply with four basic freedoms (Stallman, 1998):

- Freedom 1. To be able to use the program for any purpose, that is, there can be no limitations imposed on receivers when selling or handing over a program to them as regards what they can and can't do with that program.
- Freedom 2. To be able to study how the program works and to adapt it to our own needs, that is, its code must be susceptible of analysis and allow changes, as well as the possibility of creating new programs tailored to suit our needs.
- Freedom 3. To redistribute as many copies as we wish, with or without charge.
- Freedom 4. To improve the program and to give the community access to these improvements so that the whole community benefits from them. Conceptually speaking, this freedom is very similar to the second one. The difference for Stallman derives from the fact that this freedom must also allow offering the community the improvements made.

We must keep in mind that freedoms 2 and 4 require the receiver to have access to the program's source code and, as we previously mentioned, Stallman does not mention whether OSS should be free of cost or not. Therefore, there is a possibility that OSS may be charged for. However, once purchased it can be used in whatever way we choose: introducing modifications, correcting it, distributing it and so on.

2.1 OSS history

To intend to write the history of open source software seems pretentious at the very least. Nevertheless, we will try to briefly outline the historical evolution of OSS.

In the Sixties and Seventies software was considered irrelevant in Information and Communication Technology to the point that manufacturers sold the hardware and gave away the software. But after the anti-monopoly lawsuit against IBM, the courts forced a separation of these components so

they started to be billed separately to clients. As from that moment software started having its own value.

Because of this by 1980 most software was protected by intellectual property rights. For security reasons most program owners decided to prevent general access to their source codes and forbade their programmers from talking about them to anyone. They considered this would contribute to keeping the secrets of their creations, just as with any other element subject to intellectual property rights. Obviously, bit by bit these decisions hindered cooperation between programmers.

When in 1986 Richard Stallman, an MIT researcher, was asked to sign one of these non-disclosure agreements he decided to publish the GNU manifesto. By doing so he embarked on a project to create an operating system compatible with UNIX (GNU is not UNIX) but with a difference: it would be totally free and absolutely open to alteration according to user's specific needs (Stallman, 2000).

The project developed and grew stronger. Many programmers enthusiastically took to the idea of creating free software and after several years of hard work they had developed the basics of an operating system: a compiler, a language based editor, an interpreter and tools for networking. In spite of this, the main element was still missing, the very heart of any operating system, known as the kernel. This would be the main contribution that the Linux system would make to the project.

In 1990 Linus Torvalds, a student of the University of Helsinki (Finland) decided to improve an operating system called Minix that exploited the potential of the newly arrived 80386 to the maximum. This is how he created Linux, so well known nowadays, and also how Stallman and his collaborators found the missing element for their operating system, its kernel. By merging both what we now know as GNU/Linux was born.

From that moment new open source software begins to appear until we reach the present situation in which this kind of software is available for almost any application, such as office automation (word processing, presentations, spreadsheets, etc), graphics, statistics, mathematics, design, distance learning, web services and e-mail, amongst others.

2.2 Advantages, disadvantages and risks of OSS

It seems appropriate to briefly cover the main advantages, disadvantages and risks associated to the use of open source software. Amongst the advantages we can state the following:

- Cost (Fink, 2003 and Watson *et al*, 2005). Even though we have already mentioned that freedom of use must not be confused with non-payment, the fact remains that license models contribute to significantly reducing the final costs of open source software as opposed to owner products.
- Independence from seller (Fink, 2003 and Weber, 2004). Obviously, having access to the program's source allows users, if necessary, to change sellers, develop their own versions, etc.

- Control and quality (Fink, 2003 and Watson *et al*, 2005). By having access to the program's sources, we can always know what our software can do or what it includes and examine possible faults, which enable us to carry out quality audits at any moment by means of *a posteriori* verification. As well as this, the fact that so many people take part in the development of each project ensures a higher quality of the final product, rather than when only one person or a reduced group are in charge, even if they are professionals.
- Innovation and development possibilities (Fink, 2003; Raymond, 2001 and Watson *et al*, 2005). Once again, access to source codes will allow personal alterations without depending on third parties. In turn, this encourages innovative capacities that are not linked to any particular supplier.
- Availability of qualified personnel and local promotion of the technological industry (Fink, 2003; Pavlicek, 2000 and Bulchand, 2005). Access to source codes together with the fact that products are widely known promotes quick training of technical experts which, in addition, are generally available. Also, access to source codes supports the development of local industries.

There are other less important advantages of OSS which are, nevertheless, worth mentioning. Firstly, technical support is widely available (Fink, 2003) because products are or can be well known; the option of trying the product before purchasing it; and last, but not least, OSS is associated with an ethical advantage that avoids excessive economies of scale and allows wealth to be proportional to the work put in.

There are, in spite of all these advantages, a few drawbacks to OSS, most of which are closely linked to the very causes of those same advantages. Some of these disadvantages are:

- Availability of applications for the operating system (Fink, 2003). Presently there are not so many applications available for Linux as for other systems such as Windows. However, there are also several OSS products that have been developed for proprietary systems, though in such a case the final result would be a hybrid that would include some of the advantages of OSS but lack others.
- Maturity (Fink, 2003). The fact that there is not one single firm responsible for development means that sometimes we will find on the market products which are under trial, evaluation or similar. Frequently the success or failure of a project is based on such an unsystematic factor as the number of organisations that adopted the project in its early stages. Sometimes early-adopters can find themselves in trouble if a group or community does not grow around the project. Recently we have seen an example of this concerning e-learning platforms. There were two big commercial platforms on the market, WebCT and Blackboard, which incidentally have now merged. Simultaneously, several OSS e-learning projects appeared: Moodle, Ilias, Claroline, Atutor, etc. Whereas choosing a commercial platform at least apparently

offers certain guarantees, deciding on an open one can certainly be risky if a community has not yet formed around it.

- Scalability (Fink, 2003). Sometimes OSS products are suitable in limited production environments for a small number of users, but their possibilities on a large scale are restricted. The reason is that developing goods for a large production environment is complex, as simply the trial stages require installations that are not easily available to open source programmers which many times are freelance or program as a hobby. An example of this is the Oracle database management system as opposed to MySQL.

Finally, OSS is not free of risks, amongst others the following (Golden, 2005):

- Risks associated to license models, because these licences are less known to firms than the traditional ones. As a result, their legal departments do not dominate the situation and feel less comfortable about them.
- Security and quality hazards. OSS products are created, altered and, generally speaking, handled by many people, with greater chances of, for example, having Trojans installed.

3. OSS IN EDUCATION

In general terms, most of the advantages we have mentioned are directly applicable to education. On the other hand, some of the risks and disadvantages are minimised due to the structure of the institutions involved. We will now study them briefly.

As far as advantages are concerned, most of them apply in education. However, we would like to specially draw attention to costs. We must remember that financial pressure is considerable in educational centres of most public systems, both in developed and non-developed countries. Pressure is such that finding a way of computerising schools in a legal way and for a reasonable cost is extremely important. However it would not be fair to disregard the effort made by some manufacturers of owner software, mainly Microsoft, to adjust their prices to the possibilities of educational institutions, mostly because they realise that the germ for future users and technicians is to be found here. As a result, the differences between costs are often not as relevant as one would think.

On the other hand, all advantages concerning development and innovation that are substantial to OSS are also essential in the educational environment, especially regarding technical studies, for which the chance to examine the code, to understand it, alter and improve it becomes a fundamental part of the learning process.

Finally, we must not forget the motivational plus for a student who knows that by analysing the free code of an application the services he can in future offer the market will be much closer to existing needs. His function

will go beyond simply pressing the “*Next*” button and he will have access to much more interesting options.

About the disadvantages, none of the ones mentioned above (maturity, availability or scalability) are especially important in educational institutions. Because management mostly involves a reduced scale, here we are not concerned with highly technological applications that require a superior degree of maturity of its processes nor a wide range of options or even a great capacity of scalability.

Neither does one of the risks involved, the one regarding license models, seem especially worrying. The same can not be said for security. The fact that the source code that guarantees, for example, inviolability of marks is an issue that requires a specific debate.

3.1 OSS and teaching and learning

Apart of what has been already mentioned, OSS is about an architecture of participation which can be extended beyond software (O’Reilly, 2006). This software has just been the first area to show the power of self-organizing teams in producing value. In fact, the concept behind Web 2.0 is this architecture, being some examples those of the Wikipedia, Flickr or Amazon, three cases that obtain a good deal of their value from used cooperation.

This means one of the goals teachers have is how to translate the OSS flavour into our systems of teaching and learning, since this will allow students to better collaborate and work together in the way the business environment is going to demand from them in the near future.

3.2 OSS in Spanish secondary education institutions

In the last few years Spanish secondary education institutions have gone through an intense process of adoption of OSS. The Linex Project carried out by the Junta de Extremadura is emblematic and has received ample attention from the media, including the Washington Post (November 3rd, 2002). One of the pillars of this project has been the computerisation of secondary education institutions, as described in the paper “Information Society in Extremadura” (Díaz, 2005).

A similar path has been followed by other Spanish Autonomous Communities, in Valencia under the name Lliurex or in Andalusia known as Guadalinux.

Intuitively the advantages of OSS in secondary education seem obvious, as we have already stated. But many questions arise almost immediately. Why use OSS in secondary schools? Is it just a matter of costs, an ethical issue or is there certain social pressure imposed by Linux organised groups? Are the previously mentioned benefits true in practice?

We believe that the sort of decisions that combine education management and Information and Communication Technology can become

interesting guides that will point us towards the future of ITEM in the following years.

4. IMPLEMENTATION MODELS OF OSS

Our approach to the matter suggests three possible models that could lead to implanting OSS in an Autonomous Community.

First of all, a solid political resolve. This would be a top-down model in which the ruler would be firmly convinced of the benefits of OSS, thus starting a process that would gradually involve all social agents. This is the model has been used in Extremadura, which an important financial investment in the past years in OSS technologies as well as in placing the necessary hardware in the classrooms.

The second possibility would originate in the citizenship that, organised in pressure groups and setting a neighbouring community or another country as an example, would pressurise governments to implement this kind of initiative, even though rulers were not especially convinced of its benefits. This model has been tried in various regions in Spain, none of which have been successful up to now.

Lastly, the third would be a combination model in which OSS solutions would initially be put into practise jointly, partly due to a clear interest of the political class but also because of significant pressure exercised by citizens. This is the model used in certain regions that have followed Extremadura, in which citizens have drawn the attention of politicians towards the success in Extremadura with their project.

5. CONCLUSIONS

OSS has several implications in education. For Education Management it can provide a means of obtaining better software at a cheaper price while encouraging the development of local industries and showing a better ethics model to students. In teaching it provides a model which can be shown to students on how to collaborate and develop projects between people who never get to meet on a face to face basis, but just using Information Technology tools. Considering the advantaged associated to OSS, we believe governments involved should try to promote OSS. Extremadura, a region in Spain, shows that governments can lead the road to implementing OSS.

6. FUTURE RESEARCH

Taking all we have examined into consideration, we believe a questionnaire could be developed to be sent out to those in charge of ICT in

the Spanish Autonomous Communities and in other countries as well. The questionnaire should take into account if the Community has or has not developed OSS-related projects in the classrooms and should respond to matters such as the reasons that have lead to developing OSS, financial issues, development models, etc.

7. REFERENCES

- Bulchand, J. (2005). Software libre en la Universidad. II Libro Blanco del Software Libre. Segovia, M.A. and Abella, A. Electronic book available at <http://www.libroblanco.com>. [Accessed September 30th, 2005]
- Díaz, J.A.. (2005). Information Society in Extremadura. Information Technology and Educational Management in the Knowledge Society. Tatnall, A., Osorio, J. and Visscher, A. New York, Springer / IFIP: 171:180.
- Golden, B. (2005). Succeeding with open source. Boston, Massachusetts, Addison-Wesley, Pearson Education.
- Fink, M. (2003). The Business and Economics of Linux and Open Source. Upper Saddle River, New Jersey, Prentice Hall.
- O'Reilly, T. (2006). Four Big Ideas About Open Source. Available at http://radar.oreilly.com/archives/2006/07/four_big_ideas_about_open_sour.html [Accessed August 23rd, 2006]
- Pavlicek, R. (2000). Embracing Insanity: Open Source Software Development. Indianapolis, Sams Publishing.
- Raymond, E. (2001). The Cathedral & the Bazaar. Sebastopol, California, O'Reilly & Associates.
- Stallman, R. (1998). The GNU Project. Available from: <http://www.gnu.org/gnu/thegnuproject.html>. [Accessed September 30th, 2005]
- Stallman, R. (2000) The GNU operating system and the free software movement. Open Sources: Voices from the Open Source Revolution DiBona, C., Ockman, S. and Stone, M. Sebastopol, California, O'Reilly & Associates. Available from: <http://www.oreilly.com/catalog/opensources/book/stallman.html> [Accessed September 30th, 2005]
- Watson, R.T, Wynn, D. and Boudreau, M.C. (2005) JBoss: The Evolution of Professional Open Source Software. MIS Quarterly Executive, 4(3), Sept 2005: 329-341.
- Weber, S. (2004). The success of Open Source. Cambridge, Massachusetts, Harvard University Press.