

CONFERENCE Proceedings 2005

Towards Education for All:

The Critical Role of Open and Distance Learning in National Development



NiED



**30th August - 1st September 2005
Windhoek, Namibia**

Benefits of using Open Source Software: A Review of UNAM's Learning Management System (LMS)

CM Beukes-Amiss & TK Mufeti : University of Namibia

Abstract

In acquiring new software, an organization can either, (1) buy ready-made software, (2) download open source software, or (3) develop their own software from scratch.

With a lot of debates about Open Source Software (OSS) or Proprietary Software, it becomes imperative to look at the benefits and challenges of using OSS instead of Proprietary Software. The real benefit of OSS is in terms of freedom of access to the source code to be able to customize it to suit your needs, and in some cases it is also free in terms of purchase charges, and software licensing (O.S.S. Africa, 2005).

This paper highlights direct benefits and challenges of using open source software. The benefits, among other, are reliability, stability, cost, flexibility and freedom (gbdirect, 2005) and some of the challenges are limited support from developers, and the level of programming knowledge required, which is often lacking in organizations.

It concludes focusing on the experiences of using an open source Learning Management System (LMS) for e-Learning activities, at the University of Namibia.

1. Introduction

With the popularity of Open Source Software (OSS) increasing, the challenge for companies, government institutions, parastatals, and particularly educational institutions is to make a decision as to whether to use OSS, proprietary software, or develop their own software packages. However, before embarking on a concrete decision it is important that organisations or educational institutions know what OSS is, the difference between OSS and closed software, as well as the benefits and challenges of OSS.

OSS can be defined as software that is free in terms of its source code being available, as well as free in terms of purchase charges, and software licensing (OSS Africa, 2005). Whereas a network team of engineers and software developers are responsible for the creation and maintenance of OSS as an ongoing project, and the source code is freely available to individuals to customise according to their own needs, closed software is not free, you don't have access to the source code, but closed software operates in an environment where support is available if anything fails or malfunctions (Ravnsborg, 2005).

2. OSS vs. closed software

There are a lot of heated debates all over the world about whether to use OSS, Closed Source Software also called proprietary software, or a hybrid of the two.

Considering the fact that both software types are not flawless, a discussion about what each type offers the end-user in terms of quality is needed. Ravensborg (2005) cautions that both types are far from perfect, but OSS is doing slightly better in that its code can be viewed, changed, and adapted to suit the needs of the end-user, but at the same time OSS offers limited support when anything goes wrong.

Closed software cannot be freely distributed from user to user, and users are not able to make changes to the code. OSS, however, gives the flexibility and freedom to users to make changes and freely re-distribute to others as well. There is also a shift from paying a lot of money to vendors of software to hiring programmers to develop in-house OSS for internal activities (Ravnsborg, 2005).

When considering the quality of software for end-users, Ravensborg (2005) claims that the methods used for developing OSS are superior and produce better quality software than the methods used for closed software, but there is better control over closed software.

There are also arguments against OSS in that it makes money through traditional ways and means, being that of a single license, whereas closed software demands fees for improvements.

Closed Source Software developers will often release the software before it is entirely ready to be used by end-users, and users will be expected to apply various patches themselves without having the necessary know-how. Similarly OSS is generally too complex for the end-users to fiddle with, the code being freely available and will have to rely on programmers with the necessary know-how (Ravnsborg, 2005).

Another issue surrounding the debate of OSS vs. closed software is that of security. According to OSS developers it is apparently better in preventing security issues due to the fact that when bugs are found, it can be fixed by anybody, hence patches are more readily available. Closed software developers on the other hand argue that since no one is responsible, you never know when exactly the code is fixed (Ravnsborg, 2005). Also, the same advantage of the OSS code being freely available, can lead to problems in that hackers can exploit the software (OSS tutorial, 2005).

What is also of importance is that end-users should be aware of the difference between End-User License Agreement (EULA) for closed software, and General Public License (GPL) for OSS to be able to understand that EULA is more inflexible than GPL when acquiring and sharing software. (Surran, 2003)

What could be an advantage for the one type of software is a possible disadvantage for the other type, and the overall focus of this paper brings us to the discussion of the benefits and challenges surrounding OSS.

3. Why OSS? : benefits and challenges of OSS

This discussion has set the scene for current debates about OSS versus Closed Source Software. Against this backdrop, this section will specifically focus on the benefits and challenges of OSS.

OSS offers a couple of advantages to end-users, but on condition users should be able to understand the source code to be able to customise it effectively according to internal needs. This does not imply that Closed Source Software does not have advantages, but this section is trying to argue the case for the use of OSS in educational institutions based on its advantages.

3.1 Benefits of OSS

- Independence from software vendors
- Cost savings
- Higher reliability
- Higher stability
- Better functionality
- Better performance
- Open standards (OSS Africa, 2005)

OSS is a strong competitor of Closed Source Software in that some of the listed benefits cannot be considered for Closed Source Software at all. Looking at independence from software vendors, this is impossible with Closed Source Software as specific vendors are responsible for specific closed software. Saving on costs would imply saving on paying license fees, but only pay for support and services that you would need. (RDA Systems, 2002:1)

Reliability and stability of OSS refer to the absence of bugs causing incorrect operations, which is a result of the source code being freely available and bugs can be fixed immediately. Stability is achieved through interchange formats that are often quite stable and incompatible file formats is not that much of an issue. (Gbdirect, 2005)

Flexibility is achieved through platform neutrality. Based on performance, OSS is particularly strong in the server and network environments, and hence most advantages are geared toward the use of Linux as an OSS operating system. (OSS Africa, 2005)

OSS is also making strides in terms of software auditability, in that it is referring to third party inspection, and that is highly achievable by means of publishing the source code.

Freedom is perhaps the biggest advantage of OSS, freedom to view, to make changes, to adapt to suit internal needs. Surran (2003) claims that the real advantage of using OSS, is in that students learn the fundamental principles of using a PC through the use of OSS, rather than focus on a particular application.

3.2. Challenges of OSS

No software package comes without flaws. The concern is just how soon these flaws can be ironed out. One of the biggest challenges of OSS is that someone must understand

the source code to be able to customise it to suit internal needs. Also, compatibility is a problem when using already existing and developed closed software applications. However, this is becoming less of an issue as OSS developers are working hard at reducing compatibility problems. Finally the transition from Closed Software to OSS remains a challenge for individuals especially since a lot of people are so used to and familiar with Closed Software.

In the next section of the paper, we describe:

- (a) The approach used to choose the appropriate eLearning platform at UNAM
- (b) The benefits realized with the choice
- (c) Some of the challenges experienced with the open source platform

4. The approach used to choose an eLearning platform for UNAM

In 2003, the University of Namibia (UNAM) formulated an ICT strategy to develop a road map on how to use ICT to enhance effective learning and teaching. In 2004, an eLearning committee was established to realize the appropriate use of ICT in enhancing teaching and learning and to guide the implementation of eLearning initiatives at the university. Among other things, the committee was tasked with investigating different technology enhanced learning methods and to promote these methods to the university community [ICT, 2003: pg 13]. The committee followed a systematic approach in identifying the elearning management system to be used and to determine how elearning should be implemented at the university.

Firstly, the committee established evaluation criteria to be used in determining the elearning platform. The criteria were determined by looking at the experiences of other universities using well-known elearning platforms, and by analysing the needs of the users of the system at UNAM. The committee agreed on the following points to be used in the criteria:

1. *The functionality of the system* – the system must provide all the generic features that will enable students and lecturers to teach and learn in an elearning environment. These features include course management tools (such as features to allow the lecturer to easily upload and manage content) student management tools (such as features to allow students to be registered and deleted from a course and students to be assembled in groups for collaborative work) and communication tools (which includes both synchronous and asynchronous features to facilitate student-to-student and student-to-lecturer communication, to enable students to submit assignments, and ask questions using a discussion forum or a chat room).
2. *The hardware and software requirements* - the system must be able to operate within the existing infrastructure and must not place a strain on the already limited resources of the university.
3. *The cost of ownership* – this involves how much it will cost to purchase the system, and how much it will cost to pay yearly licence fees and to maintain the system. The chosen system should minimize these costs as much as possible.
4. *Adaptability of the system* – the system must be customisable so that it reflects not only the UNAM colours but also to reflect the context in which teaching and learning will take place. It must also make it possible to freely add new features when they become necessary.

5. *Scalability* – the chosen system must be scalable in terms of additional features.
6. *Ease of use* – the system must be intuitive enough for the users and it must be able to be used by computer novices. As an example, it must have an authoring environment that will allow content to be added without the use of. The system must also provide user manuals and online help that will guide the user on how to use the system.

After establishing the criteria, the committee identified a set of elearning software that are available in the market and started evaluating it against the criteria. For open source and other shareware platforms, the committee downloaded and installed them at the university and tested them against the evaluation criteria to determine their appropriateness to the university. In addition, the committee also used the Edutool website (<http://www.edutools.info/course/>) that is “designed to assist higher education institutions in using a more rational decision process” when selecting a learning management system. The website has an evaluation of most of the widely used available platforms, which was also compared against the criterion.

Out of the numerous learning management systems that were evaluated, the committee recommended Knowledge Environment for Web-based Learning (KEWL) as the elearning platform for the University of Namibia. KEWL is an open source learning management system that was developed at the University of Western Cape in South Africa. The system met almost all the features that were specified in the criteria, and thus offers all the common features of a learning management system including user management, course management, website administration, etc. KEWL was implemented at the University of Namibia at the end of 2004, and in 2005, nine online courses were run using a blended-learning approach at the university.

4. 1 Benefits of Using KEWL as an Open Source platform

The main benefit of using an open source learning management system at UNAM was cost. The software was acquired and installed at UNAM free of charge, and requires no license fees. The software also runs on both the Windows and Linux operating systems, which are already available at the university, and requires no additional hardware. This minimized the cost of ownership, and did not place a strain on the already limited budgetary constraints of the University.

Once the system was installed at UNAM, it was customized to reflect UNAM colors and images. Figure 1 shows the login page with the UNAM banner and motto, which were easily added by modifying the source code.



Figure 1 – Customised KEWL screen

Another benefit of using KEWL as an open source platform is realized in the flexibility it provides with regard to its features. As mentioned already, KEWL offers most of the features in proprietary elearning management system. In addition, it also provides a flexible approach for adding or removing these features from individual courses. As a result, it is therefore possible to, for example, find a discussion forum and a course submission feature in one course but not the other, even though the two courses are running on the same system. The feature can easily be added again to the course once the lecturer has decided to incorporate the missing feature in the course.

In addition, when the system is not doing what it is expected to do, one can easily go to the code and examine it to review its intent and to understand what it is currently doing. If its intent is different from what it is currently doing, the code can be modified, or be sent to a community group for further review and modification. Once the code has been changed, it can be pasted back in the application without having to uninstall and re-install the system again.

Another benefit realized is in terms of additional functionality that can be added by developers. As the use of KEWL progressed at UNAM, lecturers came up with a list of additional functionality that they needed, but it was not provided by the system at that time. The added functionality was also easily incorporated into the existing code, because the code is adequately documented, making it easy to know exactly where to place the additional functionality.

Lastly, the use of an open source platform did not lock UNAM into a specific proprietary solution, but afforded UNAM the opportunity to adapt the code at anytime to reflect the changing needs of the students and the entire university community at large.

4.2 Challenges

One of the challenges experienced with the implementation of KEWL as an open source management system is limited support. KEWL is distributed under the GNU license, which means developers are not accountable for errors or omission in the code and they cannot be held responsible for the downtime of the system. When a proprietary management system is used, the owners or vendors are under obligation to help the users and fix any errors that might be found in the system. Although there is a lot of newsgroups and community groups where users can post messages and ask questions, there is no guarantee that a required reply will come on time.

Like other open source software, one must be a professional developer to be able to add additional features to the system. UNAM have a limited number of people with experience in the programming technologies used in KEWL, such as ASP, PHP, and SQL. If a feature is desired and the institution does not have the necessary programming capacity to add it, consultants may be hired to program it, but it will mean additional cost for the university.

5. Conclusion

This paper highlighted that there is a lot of debate going on at the moment on whether to use open source or proprietary software, making it difficult for organizations to make a decision on the software to use. The paper highlighted that open source is the an appropriate solution for organizations that do not want to be locked into vendor specific software, that want to minimize the cost and that want open access to the code to be able to customize and modify the software to add additional features. It also identified limited support from the developers and the level of competency required with regard to software development as the challenges with using open source software. The experiences at the University of Namibia highlighted the benefits and challenges realized in a practical context, further highlighting the issues and opportunities inherent with the use of OSS in organizations.

6. References

- GBDIRECT. 2005. Benefits of using open source software. [online] <http://open-source.gbdirect.co.uk/migration/benefit.html> [05August 2005]
- OPENSOURCETUTORIAL. 2005. *What is Open Source Software?* [online] <http://www.opensourcetutorial.com/> [05 August 2005]
- O.S.S. AFRICA. 2005. Frequently Asked Questions. [online] <http://www.ossafrica.com/index.php?pg=faq> [05 August 2005]
- RAVNSBORG, R. Open Source Software vs. Proprietary Software: The Battle for Software Quality. [online] http://grim-reaper.blogspot.com/computers_and_technology/ [05 August 2005]
- RDA SYSTEMS, Inc. 2002. RDA open source. [online] http://www.rdasys.com/prod6_OpenSource.htm [05 August 2005]

SURRAN, M. (2003). Making the switch to open source software. [online]
<http://www.thejournal.com/magazine/vault/A4499.cfm> [05 August 2005]

UNAM. 2003. ICT Strategy Steering Committee. *The University of Namibia Information and Communication Technology (ICT) Strategy*. (University of Namibia, 2003)