Applying Open Source Software in a Development Context: expectations and experiences.
A Case Study of a University in Uganda

VICTOR VAN REIJSWOUD & EMMANUEL MULO
Uganda Martyrs University, Uganda

ABSTRACT Over recent years the issue of free and open source software (FOSS) for development in less developed countries (LDCs) has received increasing attention. In the beginning the benefits of FOSS for lower developed countries was only stressed by small groups of idealists like Richard Stallman. Now, however, it is moving into the hands of large international organizations like the World Bank, European Union and United Nations. At present FOSS is on the agendas of the donor organizations and international non-governmental organizations but not on the agendas of the decision makers in LDCs. There are growing numbers of initiatives to promote FOSS for developing countries in general and Africa in particular. At the same time there are very few organizations considering and actually implementing FOSS. The article describes the experiences of adopting FOSS in a university in Uganda with the aim of evaluating the promises made by the donor organizations and the FOSS community. The article concludes with an agenda for improving the effectiveness of adopting FOSS in LDCs.

Introduction

Over the past three years the issue of free and open source software (FOSS [1]) for development in less developed countries (LDCs) has received increasing attention. Originally, the benefits of FOSS for lower developed countries were stressed only by small groups of idealists like Richard Stallman (Williams, 2002). Now, however, involvement is moving into the hands of large international organizations like the World Bank (Dravis, 2003) and the United Nations. In the e-Commerce and Development Report released at the end of 2003, it was stated that FOSS is expected to dramatically affect the evolving Information and Communication Technology (ICT) landscape for developing countries. The United Nations Conference on Trade and Development (UNCTAD) believes that FOSS is here to stay and developing countries should benefit from this trend and start to recognize the importance of FOSS for their ICT policies (United Nations Conference on Trade and Development, 2003).

Leading organizations in the software and ICT consulting industry have embraced OSS at a rapid speed. IBM is now the major champion of open source software, and in 2002 it announced the receipt of over US$1 billion in revenue from the sale of *Linux*-based software, hardware and services. Other technology leaders, including Hewlett-Packard, Motorola, Dell, Oracle, Intel and Sun Microsystems, have also made major commitments to FOSS (United Nations Conference on Trade and Development, 2003). The sole major player resisting the open source software paradigm at the moment is Microsoft.

The advantages of FOSS are diverse, but the most often quoted benefit in relation to the developing world is the reduction in purchase and license costs of the software. Software and licenses are paid in hard currency and put an extra burden on the often disastrous financial situation of developing countries. Other advantages of FOSS include reduction of vendor lock-in, adherence

to open standards, increased transparency, minimization of security risks, increasing technical self-reliance, and providing a good starting point for local capacity development (Dravis, 2003). The last advantage is probably the most important benefit of FOSS. Local capacity is needed to understand the technical foundation of the digital divide and start initiatives to bridge it.

In spite of the obvious advantages of FOSS for developing countries, until now adoption has been low (Bruggink, 2003; Reijswoud, 2003; Reijwoud & Topi, 2004). In Africa, South Africa alone has explicitly mentioned FOSS in its ICT policy. On the contrary, governments of several of the richer countries on the continent are considering large deals with proprietary software vendors.[2] At present it seems that FOSS is on the agendas of donor organizations and international non-governmental organizations (NGOs) but not on the agenda of decision makers in LDCs. Although there is a growing number of initiatives to promote FOSS for developing countries in general and Africa in particular [3], very few organizations are considering and actually implementing FOSS.

In this article we evaluate the experiences of an organization in Uganda (in East Africa) that has decided to migrate its ICT infrastructure to FOSS. The purpose of the evaluation is to make an on-the-ground assessment of the claims about the development potential of FOSS. Accordingly, we begin with an overview of FOSS and the role it can play in the development of LDCs. Against this background we describe the case study, the progress the organization has made and the problems that were encountered. Finally, we will draw some conclusions from the experiences in the case study and outline an agenda for successful roll-out of FOSS in developing countries, especially in Africa.

FOSS for Development – an overview

When we consider FOSS in Africa we have to distinguish multiple levels in order to get a good understanding of the impact of the different initiatives. The implementation and the propagation of FOSS can be performed at micro, meso and macro levels. At the micro level we consider individual users who decide for or against FOSS. At the meso level we consider organizations that take actions to integrate FOSS into their total software infrastructure. Finally, at the macro level information technology (IT) policies and actions at a national level are considered. We begin with the macro level.

FOSS from a Macro Perspective

Governments provide a huge potential for FOSS, not only as a site for implementation of the software but, more importantly, as propagators of the philosophy behind the FOSS movement.

Over the past five years, growing numbers of countries have begun considering FOSS as a serious alternative.[4] Brazil is among the countries that have actively pursued the open source model. Brazil passed the first law in the world regarding the use of FOSS in March 2000. Brazil is one of the countries where policies regarding adoption of FOSS have been successful, notably in the states of Rio Grande do Sul and Pernambuco. Also, the Brazilian Navy has been using FOSS since 2002.[5]

In Africa, the South African government is one of the players at the forefront. The South African government released a policy framework document in September 2002 by the Open Source Work Group of the Government Information Technology Officers' Council (GITOC).[6] The GITOC policy document (Government Information Technology Officers' Council, 2002) recommends that government 'explicitly' support the adoption of open source software as part of its e-government strategy following a comprehensive study of the advantages and pitfalls of FOSS for government requirements. Besides adopting FOSS software GITOC also recommends that government promote the further development of FOSS in South Africa. There is a huge potential role for South Africa's small and medium enterprise industry in the production and implementation of FOSS as well as in setting up user training infrastructures. At the same time, the FOSS approach does represent a powerful opportunity for South African companies to bridge the technological gap at an acceptable cost.

Some *success factors* need to be considered in order to tap this potential:

Applying Open Source Software

1. *Implementation should produce value.* Value is related to economic value – that is, the reduction of costs and saving of foreign currency – and social value – that is, a wider access to information and computer training.
2. *Adequate capacity to implement, use and maintain.* There is a need to have enough trained people to support and use the FOSS solution. Training of users and developers should be a high priority.
3. *Policy support for a FOSS strategy.* Support for FOSS needs to expand to all key players at governmental level, departmental level, IT professionals and computer users in general.

The South African government's Department of Communication has already begun the move to open source by adopting *Linux* as their operating system. The government plans to save 3 billion Rand a year (approximately 383 million euros), increase spending on software that stays in their country, and increase programming skills inside the country. South Africa reports that its small-scale introductions have already made a saving of 10 million Rand (approximately 1.27 million euros).

Other countries are following this trend, with similar moves being discussed by Taiwan, China, Peru, the United Kingdom, France and Germany.[7]

FOSS from a Meso Perspective

The International Institute for Communication and Development (IICD) [8], a Dutch NGO promoting the use of ICTs in developing countries, investigated the use of FOSS in organizations in three countries in Africa: Uganda, Tanzania and Burkina Faso (Bruggink, 2003). The objective of the research was to find out how, where, and why organizations from all kinds of sectors use FOSS, what problems can be observed and what opportunities for development are available. The findings of the research show that FOSS is being used in Africa although it is not yet widespread. FOSS is mostly found at the server side of Internet service providers and is sometimes used by government and educational institutions. This means that open source operating systems, mainly *Linux* and derivatives, webservers, email servers and files servers are found where everyday computer users do not see them. Large and hierarchical organizations that have migrated completely from proprietary software to FOSS (server side and user side) have not been found. Most of the organizations that are using FOSS are relatively small. When the three countries are compared, it is concluded that Tanzanian organizations show the most initiative, while in Burkina Faso organizations show no interest in moving away from proprietary software.

The research of the IICD highlighted several reasons why organizations in Africa do not take up the challenge of FOSS. In the first place, there are some false perceptions about FOSS. Many organizations indeed believe that FOSS is *Linux* only and that FOSS is not user friendly and is only suitable for the ICT specialist. Secondly, there is limited access to FOSS. Most FOSS is distributed through the Internet. Given the scarce and/or low-bandwidth Internet connections, access to FOSS is limited as a consequence. Software companies, including FOSS companies, see little market potential in Africa (outside South Africa) and so the availability of software is low. This is also reflected in the number of resellers of FOSS. Finally, there is little expertise available to provide training and support for FOSS and, eventually, consultancy in software migration processes.

Later in the article we will further elaborate on an example of the implementation of FOSS in Uganda.

FOSS from a Micro Perspective

Most of the FOSS initiatives are small-scale projects carried out by individual people or small organizations. A growing number of individuals throughout the African continent are becoming aware of the potential of FOSS from a strategic point of view. Together with relevant advantages from an economic and technical point of view, with its lower costs, greater flexibility, availability of robust and reliable technology, and lower dependence on software vendors, FOSS does in fact represent a most important opportunity for changing the position of Africa as a whole within the information society.

At user level, and for many individuals in Africa, the challenges of FOSS provide new opportunities for development, both at personal and community level. Now that most countries in Africa are connected to the Internet, individual FOSS initiatives, which rely on this, are finally thriving. An initiative with good potential that tries to bring together the scattered FOSS society is the Free and Open Source Foundation for Africa (FOSSFA). The initiative started as the offspring of an ICT policy and civil society workshop in Addis Ababa, Ethiopia, in February 2003. During the workshop the participants agreed that FOSS is paramount to Africa's progress in the ICT arena. The mission of FOSSFA is, therefore, to promote the use and implementation of FOSS in Africa. It began to work on a coordinated approach to unite interested individuals and to support open source development, distribution and integration. The Free and Open Source Foundation for Africa envisages a future in which governments and the private sector embrace open source software and enlist local experts in adapting and developing appropriate tools, applications and infrastructures for an African technology renaissance. They foresee South-to-South cooperation in which students from Ghana to Egypt and Kenya to Namibia develop programs that are then adopted by software gurus in Nigeria, South Africa and Uganda in order to narrow the digital divide.

On a similar line a number of Internet mailing lists and user groups are emerging that focus on bringing together FOSS developers and users in Africa. At the moment there are active groups working in South Africa, Ghana, Kenya, Zambia, Zanzibar, Tanzania, Burkina Faso, and Uganda. Hosted in South Africa, Internet portals are emerging that aim to provide a starting point for knowledge of FOSS in Africa.

Similarly, at commercial level, an interesting initiative has been launched by DireqLearn.[9] DireqLearn promotes FOSS and free software (FS) as an alternative for the education sector in Africa. By adopting FOSS and FS the company can offer new solutions to the educational sector at low cost.

Finally, even if only to a limited extent, some African open source software development projects have been launched. Most of the projects are situated in South Africa, for reasons connected to the presence of infrastructure. Outside South Africa, a project that is worthy of mention is the RULE project.[10] The aim of this project is the creation of a very light *Linux* distribution for people who cannot afford modern computer systems. In order to achieve the goal, the developers are modifying a standard *Red Hat* distribution, trying to allow the greatest real functionality with the smallest consumption of CPU and RAM resources. The new distribution is mainly intended for schools and other organizations in developing countries. At present, the RULE project provides a FOSS solution with GPL license that is able to transform five-year-old computer models (Pentium 75 MHz, 16 MB RAM, 810 MB hard disk) into useful machines.

The increasing interest in FOSS is also driving the emergence of FOSS-specific organizations. In several African countries including Nigeria, Ghana, Uganda and South Africa, specialized software and consulting companies have started up. Meanwhile, young people with a background in computing are embracing the FOSS approach and trying to reform the accepted practice of buying pirated proprietary software. At present, the market share of FOSS is still small and it is a struggle for these specialized companies to survive. However, when the benefits become clear and FOSS is implemented on a larger scale, the capacity to implement the systems will be ready.

Implementing FOSS – a case study

There are hardly any documented large-scale organizational implementations of FOSS in developing countries. Where FOSS is implemented, it mostly concerns small donor-funded projects or relatively simple organizations; see, for example, the projects described in Dravis (2003). The case study presented here describes a relatively large organization that has made the strategic decision to move away from proprietary software to FOSS.

The goal of the case study is to evaluate whether the high expectations of the use of FOSS for development translate well in a practical situation.

The case study is based on documentation and interviews with the main stakeholders at the university. Since both researchers are employed at the university and participated in the migration

project, their views and experiences are also included. We have tried to avoid being subjective as far as possible.

Uganda Martyrs University

Uganda Martyrs University is a privately owned university in the central province of Uganda. The university opened its doors in 1993 after the government allowed private universities to exist alongside government-owned universities. The main reason for this was to improve the quality and the capacity of higher education in Uganda.

In February 2005 the university had 2200 students enrolled in full-time and part-time programs at diploma and degree levels. The main campus of the university is located in Nkozi, 80 km outside the Ugandan capital city, Kampala. The location of the main campus can be characterized as rural. When the university opened there were no telephone connections, no steady water supply and electricity was unreliable. This has changed over the years and now the university is recognized for its good and reliable facilities. The university has a smaller campus in Kampala, where some Masters programs are offered in the evenings and several outreach offices are available for students who cannot easily travel to the main campus.

The university employs a full-time academic staff of 86 people and an administrative and support staff of 117. With this size Uganda Martyrs University qualifies as a large organization in the private sector of Uganda.

The case study mainly focuses on the Nkozi campus of the university.

FOSS at Uganda Martyrs University – the initial stages

The FOSS project at Uganda Martyrs University had an informal start in 2001 when foreign assistance was offered to set up a mail server at the main campus. Since there was only a budget available for hardware and no provision for software, it was decided to equip the server with free and open source software. The mail server was configured with *Red Hat Linux 6.0*, *Sendmail* as the mail transfer agent and *Neomail* as the web-based mail client. A web server, to host the local intranet, was configured with *SuSE Linux 7.2* and *Apache* web server software. When a new systems administrator was hired he was trained to use and maintain the implemented configurations. The new systems administrator picked up interest in free and open source software and started to extend FOSS to other parts of the system. At the beginning of 2002, the systems administrator incorporated FOSS for the proxy server (*Squid*) and the firewall (*SuSEFirewall*) for Internet access and some other minor applications.

In mid-2002, the project got a new impulse when several guest lecturers from universities and other organizations in Europe began visiting the university to teach in the then newly started Master of Science in Information Systems program. These lecturers encountered installations of pirated software on most computers of the university and started to question the ICT policy. The university did not have an ICT policy formulated but realized that there was a need to take action. This is when the FOSS project started formally. In the course of the academic year 2002-03 the ICT Department, together with the Office of the Vice Chancellor and the Department of Computer Science and Information Systems, outlined a software policy based on open source software. The software policy was based on two underlying principles:

1. the university wants to optimize access to ICT for students and staff within the limited funds available;
2. the university will not support the use of pirated software on university property.

The first principle was derived from the mission of the university that promotes access to information for the university community. The second principle was derived from the Christian values on which the university is based, which state that you shall not steal (not even software). Free and open source software was considered a good alternative to work within the two principles.

In May 2003, the Senate of the university officially agreed on a FOSS policy and preparations started for a full migration on the server side as well as the user desktop applications.

Migrating the Desktops

The major challenge for the university was the migration of desktop applications. A review of literature revealed very little reference material and few success stories. Experiences with similar migration projects in Uganda and the rest of East Africa were not available. The university received help from the FOSS Group of the University of Huddersfield, in the United Kingdom, as a response to a message sent to one of the *Linux* mailing lists. Other than that, the university ICT staff members were on their own to plan and execute the most difficult part of the FOSS migration.

At the start of the migration project, all computers in the university were using *Microsoft Windows* (*98*, *2000* and *XP*), the *Microsoft Office* suite and other common proprietary software applications. One of the first steps in the project was to identify the main applications and their usage, in order to select FOSS alternatives. It was observed that the university staff and students used very few 'exotic' applications. They mostly used the mainstream productivity applications. This made the selection of alternatives relatively straightforward. Table I shows the alternatives that were selected to replace proprietary software.

Task	Proprietary software	Open Source alternative
Operating system	Windows 9x, 2000, XP	GNU/Linux
Office productivity suite	Microsoft Office	Open Office
Mail client	Microsoft Outlook Express	Kmail, Mozilla Mail
Internet browser	Internet Explorer	Konqueror, Mozilla
Database	Microsoft Access	MySQL/phpMyAdmin
Programming	Wordpad	Kate
	Borland Builder	Eclipse
Statistical analysis	SPSS	Open Office Calc
Web design	Microsoft Front Page	Bluefish/NVu

Table I. Main proprietary software used and open source software alternatives selected.

Since the operating system would also be migrated, a decision needed to be made on the *Linux* distribution that would be used as the standard at the university. Several alternative distributions were considered and finally the *Knoppix* distribution was selected. The main reasons for this decision were that *Knoppix* is a one-disk, bootable distribution that can also be installed easily. The distribution could be handed out to the students and used as a CD-ROM-bootable program on any available computer (even one with another operating system already installed). Research on the Internet showed us that the *Knoppix* distribution would work well on older machines, of which the university had numerous (Pentium IIs). Finally, the *Knoppix* distribution came already bundled with most of the packages that would provide alternatives for the proprietary software being used at the university.

It was decided that the implementation strategy for the migration would be staged. First, all the computers that have public access (library and computer labs) would be migrated. Once this was completed, the lecturing staff would be migrated and finally, the administration (financial and student affairs administration units) of the university. This strategy was based on the reasoning that the university's main operations should not be endangered.

The first phase was scheduled to take place during the absence of the students (June-August 2003). An evaluation of the first phase of the migration would be performed before the start of the second phase. The second phase was scheduled for the long vacation (June-August 2004). A time frame for the third phase was not determined.

Problems Encountered during the Migration

The project encountered unexpected technical and organizational problems in the first phase that delayed the time frame for the further implementation. The major problems are listed below.

Although several claims were made about the installation of *Linux* on older machines (Pentium II/Dell), it was not as smooth as these descriptions seemed to suggest. Many of the machines did not have CD-ROM drives or were not able to boot from the CD for the installation. Bootable floppy disks had to be created to solve this, but for about 20% of the older computers the

installation failed. There were also problems of maintenance at a later stage for the computers without CD-ROM drives.

Limited disk space and RAM handicapped the performance of the machines. The machines installed with *Linux* did not perform much better than similar hardware configurations with *Microsoft Windows* installed on them. The users, therefore, did not consider this as an improvement and this had a very negative impact on their acceptance of the new software.

Although it was anticipated that the *GUI* [Graphical User Interface] *(KDE 3.2)* would not cause problems for the more experienced *Windows* users, the slight differences became bigger hurdles than expected. The most common problem was that the *Knoppix* distribution requires users to mount and unmount their floppy disks. *Windows* does not require this. After losing information due to (un)mounting improperly, the users started to question and resist the user friendliness of the new systems.

A special problem was caused by the frequent power cuts in Uganda and improper (hard) shutdown. When the *Knoppix* machines were not switched off properly (through misuse or power cuts), a file system failure was created with the result that the operating system was no longer mounted automatically. In order to boot the computer, a root password was needed and the file systems needed to be repaired. This repair procedure always took a long time since the repair program was checking the entire hard disk. The users were not able to do this themselves, hence help from the ICT department was required every time the problem occurred. In the newsgroups it was explained that the problem was caused by the default use of the ext2 file system. When the file systems were converted to ext3 or Reiser, which provide journaling, the problem was solved.

There were some cases where there were no available alternatives for the software being used. Computers had to be installed with a dual boot system setting *Linux* as the default option. The same open source software applications had to be installed on both operating systems which meant double work per computer for the ICT staff. Students were still working a lot in *Microsoft Windows*. In order to encourage them to choose *Linux*, Internet access (and as a result web-based email) was restricted to the *Linux* operating system.

Finally, differences between the file formats of the office applications (*Microsoft Office* and *Open Office*) caused a problem. Since the staff and the administration were not yet migrated, the files sent by the students could not be read. With files sent by the staff there were no problems. The use of the *Open Office* option of automatically saving files into *Microsoft Office* suite formats was not valued since every time an *Open Office* document was being saved in these formats, the following worrying message appeared: 'Saving in external formats may have caused information loss. Do you still want to close?' The message was confusing to the users.

Evaluation of Phase I

Although solutions were found for most of the technical problems with the installation of the new FOSS system on the public computers in the university, the evaluation showed that acceptance of the new systems was not as high as expected.

The mounting and unmounting of floppy disks was the major cause of resistance, especially since forgetting to unmount the disk caused the loss or corruption of files. This problem was overcome by adopting the *SuSE Linux 9.1* distribution since it had an auto-mount and unmount feature.

Students, especially freshers (first-year students), responded very positively to the new systems. They had a choice between *Windows* and *Linux* (dual boot system). Observations in labs showed that most of them decided to use the *Linux* side. Among students who had already had some experience with computing in *Microsoft Windows* the resistance to the new software was extremely high. Some of the postgraduate students wrote complaint letters to the university management about the use of 'inferior' software and even threatened to strike. The resistance to the use of FOSS remained until that class of students graduated. For the incoming students, a compulsory FOSS computer literacy course was introduced based on a manual (*Easy Linux Introductory Guide to Computers*) developed by the university. This greatly reduced the resistance. At present students are experiencing few problems.

On the technical side, the problem of maintaining computers without CD-ROM drives was solved through the installation of *SuSE Linux 9.1*. It provided the option to perform installation and upgrading of software through the network. All that was needed was to make sure the computers had a network interface card and a connection point. This saved the technical staff having to carry around and/or keep copies of very many installation CDs and external CD-ROM units.

Overall, we underestimated the importance of awareness creation of the underlying motives of the university to move to FOSS. The explanation of these reasons needs to be taken extremely seriously to secure commitment of the users. We also underestimated the need to have existing continuous and constantly available support to ease the users into the new system. This meant that even with the introduction of the improved system that performed the auto (un)mounting for the users, they still had a mental block and were somewhat reluctant to trust the system. The university has embarked on an active promotion of the ideas behind FOSS.

Phase II – the staff

The second phase, the large-scale migration of the staff computers, was planned for the period June-August 2004 but was delayed due to the problems in the first phase. In order to keep the migration on track it was decided to concentrate on the new computers only. All new computers that were purchased were installed with FOSS. Since almost all computers that the university was purchasing came pre-installed with the *Microsoft Windows* operating system, a dual boot system was installed with *Linux* as a default option of the two.

Some of the computers needed to continue to operate on *Microsoft Windows* because certain applications were being used that have no satisfactory FOSS alternative. Some of these applications are listed in Table II.

Task	Proprietary software	Open Source alternative
Financial application	Tally	-
Architectural design	Vector Works	-
Word processing	Corel Word Perfect	-

Table II. Applications without satisfactory and/or compatible FOSS alternatives.

The staff of the ICT department went around the university to install FOSS applications for the *Microsoft Windows* platform on the staff computers. This was needed to support the format of documents that the students were sending to the staff. Staff were also informed that no support would be given to illegal proprietary software. Unfortunately, no member of staff, other than the staff in the Department of Computer Science and Information Systems and the staff in the ICT department, allowed their 'personal computers' to be migrated to *Linux*. Only official work computers were migrated.

For the installations that were done on the university property being used by the staff, it was rare to find them using the FOSS alternatives that were provided for them. The few who tried using these alternatives had lots of complaints about the software not being able to perform the kind of tasks that they wanted.

Evaluation of Phase II

The second phase turned out to be even more difficult than the first phase. Although there were relatively few technical problems, the high level of resistance of the staff at the university virtually stalled the project.

The biggest problem in the whole project and especially in the second phase was the acceptance of the new software by the staff. The users of *Microsoft Windows* found it difficult to switch to the new system. They felt that they were migrating to an inferior system and, as a result, small differences were turned into big problems. For example, the fact that the settings for the page layout are in a different location for *Open Office* made them feel that the new package was inferior to the well-known *Microsoft Office Suite*. Arguments that the location of the page characteristics in *Open Office* displays a more logical user–interface design were not accepted. The migration team

concluded that the differences in the user interface were underestimated and too little information was provided on the reasons and consequences of the migration to get full user commitment. When introducing a new software environment – even when the differences are small – several training workshops highlighting the reasons for and consequences of the changes should be planned.

The project also underestimated the number of *Corel Word Perfect* users and the problem of migrating their documents. *Open Office* can handle *Microsoft Office* files well, but there is no facility for *Word Perfect* files. The fact that these files could not be read in *Open Office* was used as a reason not to migrate regardless of the varying numbers of documents that the users have available in *Word Perfect* formats. The ICT department is looking at ways to handle this problem. Some considerations at the time of writing include encouraging staff to use *Corel Word Perfect* as a document reader only and to adopt *Open Office* for creating and editing newer documents. The other consideration is to get document converters to create PDF versions of the older documents that the staff may need to keep as archives.

At the moment we observe a growing divide between the staff and the students in terms of the software used. The staff tend to continue to use proprietary software while students move more onto the open source software side.

Lessons Learned – critical analysis

The migration at Uganda Martyrs University allowed us to draw some important conclusions about a large-scale migration to FOSS.

Installation of FOSS on the server side proved to be a big technical challenge. There was little hands-on guidance and support available to help the system administrators in the university. Since the university was the first organization in the country to migrate there was no possibility to hire local technical experts to assist the staff on site. Hiring support on the international market was considered unfeasible due to financial limitations (the daily fee of international consultants is in most cases higher than a monthly salary of the local staff). Online support by the FOSS community proved to be too unreliable and often not applicable to the situation at the university. Therefore, the staff of the ICT department had to rely on their own research, and much of the implementation was done through trial and error. The speed of the migration was, therefore, slow and demanded a lot of patience from the users. On some occasions email and the Internet connection were down for several days.

Whereas *Microsoft* software applications provide a standard environment for the desktops, FOSS leaves more room for choice. Advantages and disadvantages of the different FOSS desktop applications are not well documented. At the university, this led to changing standards. Where *Konqueror* was the first choice for web browsing, this was later changed to *Mozilla* when it became clear that *Konqueror* had problems with viewing some commonly visited pages on the intranet and Internet that contained javascripts. Recently, we observe a change from *Bluefish* to *NVu* for building web pages. These changing standards are confusing for most users. These changes seem to be mostly appreciated by the technical staff and the students in computer science and information technology. As far as end users go, therefore, it would be helpful to pick standard well-developed packages taking into consideration the users' possible future needs. End users would want to spend most of their time being productive rather than learning the computer environment. However, there are no guarantees because new FOSS projects are starting up everywhere and a better alternative might be developed.

The introduction and the roll-out of the migration project at the university revealed that continuous information to the users is needed. Their commitment to and support of the project are essential for success. The approach at the university was a top-down approach with a presentation for management and senate, an initiation workshop, a mid-semester workshop for all staff and individual support for all users. This approach was not enough. Although the resistance to the changes seemed to diminish after the workshop and presentations, it proved to come back quickly, and more strongly than before. Small problems were magnified. The fact that the migration team was composed of technical personnel, and with strong support from the senior management of the university and the Vice Chancellor as champion, did not guarantee complete success.

The migration of the students before the migration of the staff seems to have been disadvantageous. The expectation that the staff would support new software and request installation of FOSS on their machines turned out to be a miscalculation. Instead, several staff pushed students into using proprietary software formats, for example when handing in assignments. Documents saved in *Open Office* format were not accepted. From our experiences it may be a wise option to get staff acceptance and migrate them before any attempts to migrate the students.

The migration team is contemplating an approach to migrate the rest of the university. If this migration is not completed, there might be a quick return to the original situation.

Conclusions and Research Agenda

In spite of the high expectations of the policy makers about the development potential of FOSS, the reality of implementing FOSS in a developing country is difficult. The route to the implementation of FOSS is one with a lot of hurdles. Some of these hurdles are general and some are specific to developing countries.

At a general level we observe that there is a strong resistance to changing to FOSS applications. Many users start a migration with the idea that they are confronted by an imperative of the 'technical people' to use 'inferior software'. Their judgment is solely based on the experiences that they have with the desktop applications. On the server side, where the migration is driven by the technical staff, the clear advantages are a strong motivator for the change to FOSS.

On the desktop the portability of files between FOSS and proprietary software is still a problem. Until this issue is solved, desktop migration will remain difficult. It is high time that proprietary software producers are forced to adhere to international standards or to completely open up their own standards.

The need for education material for FOSS is high. The material currently available is mostly very technical and not understandable for general users. Availability of student material for example on *Linux*, *Open Office*, *MySQL/PHPMyAdmin*, *GIMP* and *Bluefish* as replacements for the proprietary tools may greatly improve the use of FOSS tools.

In the context of the developing countries the need for appropriate support in implementing FOSS is high. Experiences at Uganda Martyrs University show that little help was received from the international mailing list community since the questions posted were considered basic and not challenging to members on the list. On the other hand, the discussions in the mailing lists were too difficult and not (yet) applicable to the situation at hand. It seemed difficult to bridge the knowledge gap, and implementers felt isolated in their problems. In order to support the migration in developing countries international organizations like the World Bank or UNCTAD need to consider setting up a support center that deals with the questions of the system administrators and users in these countries.

Another specific problem in the context of the developing countries is the feeling that the access to the 'good' tools from the West is denied. A question that was often asked was: 'Why are the people in the West not using these [FOSS] programs when you are saying they are so good?' This argument is difficult to counter until there are success stories available from Western organizations. The situation gets even worse when international organizations that promote the use of FOSS in developing countries only accept files in proprietary software formats (.doc, .xls, .ppt), have web servers that run on proprietary software and websites that can only be browsed optimally with *Microsoft Internet Explorer*.

Finally, the market in pirated software in developing countries is well organized and institutionalized. Pirated software is readily available from every street corner at very low prices, and support for installation often accompanies the sales. Many of the new computers that are bought in Uganda, for example, have full installations of pirated software. The ones that have valid licenses cost more than the individual is willing to part with. This applies to both servers as well as desktop computers. Why would anybody want to install something else?

At present the development potential of FOSS for developing countries is still a theoretical potential. At the practical level more research, more support and a changed attitude of the organizations in developed countries are needed. Research should focus on the development of

better tools to bridge the compatibility problems. More support is paramount to the success of the acceptance of FOSS in developing countries. Support should focus on practical help with the implementation of FOSS, but also for lecturers who want to use FOSS applications in their courses. More educational material, preferably published under the Open Content license, could act as a catalyst in an environment where the need for textbooks is extremely high. Finally, organizations working with developing countries should set an example by adopting FOSS as a standard in their organization. As long as organizations in the developing world need to communicate with their counterparts in the developed world by proprietary software standards and proprietary tools, the development potential of FOSS will be considered a myth and never a real possibility.

Notes

[1] The authors are well aware of the paradigmatic differences between free software and open source software. However, it is often difficult to clearly distinguish these differences. We, therefore, prefer to use the term 'free and open source software' (FOSS) to capture both paradigms.

[2] See Free and Open Source Foundation for Africa, http://www.fossfa.net

[3] See for example, http://www.fossfa.net, http://www.eacoss.org

[4] See http://www.apc.org

[5] See http://www.pernambuco.com/tecnologia/arquivo/softlivre1.html

[6] See http://www.oss.gov.za for details and discussion about FOSS in South Africa.

[7] See *Bundesrechnungshof fordert Einsatz von Open Source, 25.02.2002*, http://www.heise.de/newsticker/meldung/25140

[8] See http://www.iicd.org

[9] See http://www.direqlearn.org

[10] *Run Up to-date Linux Everywhere* – see http://www.rule-project.org/en/

References

Bruggink, M. (2003) Open Source in Africa: a global reality; take it or leave it?, *IICD Research Brief – No UICT01*. International Institute for Communication and Development. http://www.iicd.org

Dravis, P. (2003) *Open Source Software: perspectives for development*. Washington, DC: World Bank.

Government Information Technology Officers' Council (2002*) Using Open Source Software in the South African Government. A Proposed Strategy Compiled by the Government Information Technology Officers' Council*.

Reijswoud, V. van (2003) Open Source Software – the alternative for Africa. Working Paper, Uganda Martyrs University.

Reijswoud, V. van & Topi, C. (2004) Alternative Routes in the Digital World: open source software in Africa, in P. Kanyandago & L. Mugumya (Eds) *Mtafiti Mwafrika* [African researcher], pp. 76-94. Nkozi: Uganda Martyrs University Press.

United Nations Conference on Trade and Development (2003) *E-commerce and Development Report 2003*. New York and Geneva: United Nations.

Williams, S. (2002) *Free as in Freedom: Richard Stallman's crusade for free software*. Sebastopol: O'Reilly.

VICTOR VAN REIJSWOUD is Professor of Information Systems at Uganda Martyrs University, Nkozi, Uganda (www.umu.ac.ug) and Chairman of the East African Center for Open Source Software in Kampala, Uganda (www.eacoss.org). He has been the initiator and architect of the FOSS migration at Uganda Martyrs University. Van Reijswoud has extensive work experience in academic and business environments. He has operated as (senior) business and IT consultant specializing in IT-enabled process optimization and information systems design, as project manager in information systems and e-business projects and held senior management positions in the IT industry. He has published over 50 academic publications on information systems design methodologies, organization and IT alignment and process improvement through the use of e-commerce, business and IT architecture. He has lectured at universities in Europe, the USA and

Africa. Currently he operates as an independent consultant concentrating on the use of free and open source software for development and the selection of appropriate information and communication technology for the developing countries. *Correspondence*: Victor van Reijswoud, BP 1474, Bujumbura, Burundi (victor@eacoss.org).

EMMANUEL MULO is the webmaster at Uganda Martyrs University and an assistant lecturer in the Department of Computer Science and Information Systems of Uganda Martyrs University. He provided technical assistance in the migration project. He specializes in programming and FOSS tools for web development. Currently he is pursuing his MSc. Computer Science at Delft University of Technology in the Netherlands. *Correspondence*: Emmanuel Mulo, Uganda Martyrs University, PO Box 5498, Kampala, Uganda (mulo@eacoss.org).