The Technical, Non-technical Issues and the Challenges of Migration to Free and Open Source Software

Mohamed Sarrab[1], Mahmoud Elbasir[2] and Laila Elgamel[3]

[1] Communication and Information Research Center, Sultan Qaboos University
Al-Khodh Muscat 123, Sultanate of Oman

[2] Centre for Computing and Social Responsibility, De Montfort University
Leicester, UK

[3] Software Technology Research Laboratory, De Montfort University
Leicester, UK

Abstract

Interest has been raised over software that has been developed not by dedicated software companies but by groups of independent programmers with different level of experience who collaborate over the Internet and offer the fruits of their labor free. Open source software communities have successfully developed many software pieces and suggest Free Open Source Software (FOSS) as a solution for a lot of computing problems, although most computer users only use proprietary applications. Many proprietary software users claim that open source software invites problems, such as, generating software codes that could be vulnerable to hackers and prone to errors. This paper presents an overview of the real issues and challenges for the migration to free and open source software (FOSS) and suggests some important criteria for assessing migration issues and challenges. Migratory problems may be divided into technical and non-technical issues and challenges. The technical issues and challenges include performance, technical infrastructure, usability, integrity, support availability, security, information flow control, data migration, flexibility and ease of use and management and maintenance of OSS, whereas the non-technical issues and challenges include organizational culture, human factors (staff skills) and legal issues.

Keywords: *Free Open Source Software, FOSS, Performance,* ***Usability****, Integrity, Availability, Security, Flexibility and Ease of Use, human factors and legal issues*

1. Introduction

Developments in information and communication technology support the existence of Open Source Software (OSS). The term refers to software whose source codes can be modified to make them work, as their users require. Some OSS may reserve the rights of re-distribution but in other cases, it might be free. A distributor or developer might charge for services including special training, installation, programming and technical support, etc. In general, the term OSS refers to software that is freely available, widely accessible and reusable [1, 2].

The increasing popularity of OSS has dramatically changed the software industry in recent years. OSS is often seen as a possible solution to some of the challenges presently faced by many software communities, especially among developing countries. Such challenges include controlling piracy, exerting a greater level of control over acquired software and dealing with broader policy perspectives that pertain to the development of a national domestic software industry. From a policy perspective, an 'open source' can be defined as a software-licensing model where the software's source code is made available, subsequently modified, redistributed and added to, although often with certain restrictions. In addition, a range of benefits, under commercial arrangements, may be made available, such as, updates, training and ancillary software services.

OSSs are usually, although not exclusively, developed by the collaborative efforts of a group of people who contribute components to the final version of the software. Software companies may also produce programs for the open source community. Proprietary or commercial software is developed under commercial rules and policies, in other words, it is licensed for a fee to a customer in binary, object or executable code (either directly or through channels). The company that writes the program usually provides updates, training and other services required by its users so that the software works efficiently. The software's source code might be made available to a certain number of its users through a special license or an alternative agreement but often remains unavailable to the general public and may not be copied, changed or modified except in a manner unless provided for under the terms of a prior agreement.

Each software model (Open Source Software and Commercial Software) represents a viable business strategy for their companies, as well as supporting and

providing their customers with real advantages. Solutions that the software offers are being continually updated, with providers concentrating and improving on a variety of emerging issues and problems, such as, addressing reliability, security and information flow issues. In fact, OSS has surprised many in the industry by acquiring a good reputation for its reliability, efficiency and functionality [1, 2 and 3]. This paper discusses the key challenges and issues involved in the migration to Free and Open Source Software. The paper reviews the technical and non-technical issues and challenges of this migration process. The paper is based on an argumentative and philosophical approach. No formal research methodology was used during the research; however, personal knowledge and a literature review are presented.

2. Technical Issues and Challenges

2.1 Performance

One of the main technical challenges for migration to FOSS is software performance. Does the software have a good reputation for performance? In industry, several pieces of open source software have very good performance reputations, i.e. Linux, Apache Web Server, GNU Compiler Collection (GCC), Samba, etc. Comparing the performance of Microsoft Windows and GNU/Linux on equivalent hardware has a history of argumentative claims and different results based on different assumptions. FOSS always attempts to be performance competitive with closed software and in some circumstances, beats the competition. Performance benchmarks, however, are very sensitive to the environment and assumptions. In benchmarking, everything depends upon the assumptions and configuration. Consequently, high performance is a very big challenge for FOSS because it is always compared to the performance of closed software. For the migration to FOSS, FOSS should always provide a higher performance compared to commercial software [9].

2.2 Technical Infrastructure

One of the key barriers to the development of FOSS is poor infrastructure. A good Internet infrastructure needed as a tool for developing FOSS. A reliable broadband access is very important to develop, disseminate and download FOSS documents and applications. It is also very important in creating FOSS communities. FOSS communities always serve as a support centre or help-desk. In addition, the lack of up to date application or software manuals in local languages constitute one of the factors that influence the quality of the technical infrastructure. Learning and supporting FOSS may require a greater training input than equivalent proprietary software [10].

2.3 Usability

Most users use proprietary applications. It is frequently considered that FOSS's usability is one of the reasons that limit its use. Usability is typically described in terms of five characteristics: learnability, efficiency of use, memorability, error frequency and severity and subjective satisfaction [11]. The process of software development needs to be examined in order to understand FOSS's usability. It seems that many development tools and software applications, such as; editors and compilers do not represent major problems in terms of FOSS's usability. FOSS's usability needs to be advertised more widely to encourage end users to engage with the software and experts to provide professional reports about its utility. Encouraging normal users to engage with FOSS development projects will manifest the creation of a networked development community, which may facilitate usability in the same way as functionality [12].

2.4 Integrity

The information security is more than just confidentiality. The keys to information security are integrity and availability. The integrity is more than just that information held in databases is correct. For example, how can it be demonstrated that open source software not only does what it is supposed to do but also does not do what it is not supposed to do?
The integrity of open source software can be defined as its ability to withstand security attacks and is an element of software assurance. Software assurance means that the software functions as intended. Software assurance is regularly discussed in the context of guaranteeing and ensuring that the open source code is more secure via the repeated application of secure development practices. Although there has been an increase in interest in reducing open source software vulnerabilities via secure development practices, it represents only one aspect of open source software assurance. Process security is another key factor for software suppliers and their customers to consider. Process security is reference to the processes used to handle open source software components while gathering their sources, development and delivery. Meanwhile a variety of possible attack vectors exist during the open source software lifecycle. A key problem in open source software systems is that although a software application at the beginning of execution may be verified or validated as authentic, during running, the flow of the execution can be redirected to inject externally malicious codes using, e.g. buffer overflow exploit [13].

2.5 Support Availability

Usually, the development of free software commences by a small group of people attempting to find solutions to specific problems or needs to realize an idea. This group of people will at least solve some parts of the initial problem. Then the initial solution will be published on the Net. Others will hear and learn about it, find it interesting and start to implement it to solve issues that relate to their own needs and applications. In this way, the initial small group of people will grow, which will accelerate development and support free software progress in many different areas, such as, learning, funding and distribution. Transaction costs are one of the key factors for FOSS, which are beyond free software developers' budgets, because the development environments of commercial software are expensive and frequently beyond people's budgets.

Another factor is generating a reservoir of sufficiently and properly educated persons. People should be able to learn about IT and programming concepts. Specifically, people should be provided with the knowledge and abilities that empower them to extend and continue their studies on their own. In fact, people are learning software development via pirated versions. These software development environments demonstrate the effectiveness of every-day practical use by the free software community. The education system should teach users the use of these software development environments instead of using commercial ones. An open source community could provide support for FOSS. Such a community will be devoid of legal personality or formal organization but composed of an organized community of developers with legal personality and commercial companies for software support [14].

2.6 Security

The advent of the low cost communication technologies and the high speed Internet provides an incredible new opportunity of creating a true information society. This raises several doubts about the security of any information system. Developers of OSS projects are always concerned about security in the same way as the developers of proprietary software projects. The development processes, however, may be very different and one cannot judge which one is inherently more secure [15].

Measuring the security of OSS or proprietary software is very difficult but it can clearly be said that FOSS systems are vulnerable to security bugs or flaws. The contentious issue is whether OSS reduces or improves software security. In fact, in [16] its author stated that FOSS does not guarantee security. With respect to security, FOSS is often superior to proprietary systems and this is cited as one of its most important advantages. The detection of security errors, bugs and risks in FOSS is fast and so is its elimination process. This is because in FOSS the source code is made public [13]. Finally, FOSS has a distinct advantage over proprietary systems, where it supports the ease and quick identification of potential security problems and their correction [17].

2.7 Information Flow Control

Information flow control aims to prevent the leak of sensitive information to any object that is not trusted. Security in the context of information flow has the following components.

- The presence of information flow requirements, they define the direction of the passage of data in the system.
- An information flow policy to define what steps should be taken to detect and prevent the leak of the information.
- An information flow mechanism to define what tools and methods are used to ensure that the previous steps are followed.
- A standard security mechanism, such as, access control, firewall and encryption, which focus upon controlling the release of confidential information but without limitations placed on controlling the propagation of it . The principle problem of controlling the confidentiality of sensitive information starts after access to the open source software is granted.

Information flow control aims to fill the gaps left by standard security mechanisms in both proprietary and open source software by considering the passage of data when enforcing a flow security policy. Information flow occurs from source to target objects. Whenever information is read from a source it is potentially propagated to the target object.

Research into the control of information flow concentrates upon developing a usable security mechanism within an application during runtime whether the software is closed or open. Whereas, usability refers to enabling users to manage their system's security without defining elaborate security rules before starting the application. This more easily attained with open source software because security will be achieved as an interactive process where the user will be asked to provide the system's requirements. These requirements will be made available to the software. The software will enforce these requirements on the application by tracking the flow of information. As a result, in open source software, the target application can be modified whenever the users see any violation of information flow rules or requirements [18, 19].

2.8 Data Migration

Data is managed and stored by database applications. Virtually all public administrations have huge databases. Often this data is of critical importance and huge (financial) resources have been and are allocated to collect, organize and maintain the data. It is important to divide the data into categories, namely [13].

1. Data, which can be discarded
2. Data, which is useful and in open format, such as, PDF, Postscript or can be easily translated into open format. The cost should be considered.
3. Data that must be kept but is in a legacy closed format, which cannot be easily translated into an open one. This data may need copies of the legacy software [20].

2.9 Flexibility and Ease of Use

Flexibility and ease of use are two highly interrelated attributes.

- Flexibility can be defined as a measure of how well the software can be used to handle circumstances for which it was not originally designed.
- The ease of use can be subjective in nature. It depends on factors, such as, being familiar with the software, user experience or even its direct impact on solving problems. This criterion is usually associated with the user interface. In addition, how easy it is to use the software within the development environment. In the researcher's opinion, if it is easy read/write the software's code then it is easy to use it.

FOSS products have an important advantage over most proprietary programs both in ease of use and flexibility. The different types of FOSS programs can be modified to fit a user's needs to match customer circumstances but having this advantage might require one to have programming skills or pay someone to do so. Another advantage is that some FOSS programs are easier to be extended than others [20, 21] are.

2.10 Management and Maintenance of OSS

As OSS is becoming more important and widespread, its management and maintenance are becoming significant issues. Generally, software management and maintenance are resource and time consuming processes. Error and bug detection and correction are the main duties of software management and maintenance. The best way to execute these duties is during software development. Hence, finding areas of the software that require testing can help project managers and software engineers to restructure, inspect and test critical parts [13]. As a result, developers can use software resources more efficiently in order to deliver high quality software products in a timely manner [22] because applying equal verification and testing all the software system's parts has become cost-prohibitive [23].

3. Non-technical Issues and Challenges

3.1 Organizational Culture

Switching from proprietary to open source software is neither easy nor obvious. It depends on a range of factors. One of these factors is organizational structure. The infrastructure of the IT is often decentralized so that each department has its own specialist but in the case of fundamental changes, such as; migrating to FOSS, a decentralized infrastructure does not help. This because with a centralized IT structure the changes that are require for successful migration to FOSS are easier. The migration to FOSS leads to virtual and physical organizational changes. Virtual organizational change refers to how computer software systems will be managed to make migration possible. The FOSS migration policy is enforced to assess the stock of the IT hardware and software [4].

3.2 Human Factors – Staff Skills

The new open source software tools may reduce the personnel productivity. This factor raises many questions. Are the available staffs familiar with the operating environment and programming language? Do staffs have the appropriate skills to deploy and maintain the software? In view of the personal training that is required for the new tool and the inevitable hostility to change, is the organization going to support FOSS application development and the staff training that is needed? In case that the organization is not willing to support the innovation, is there any access to an external organization that can provide the necessary support and the kind of help that is needed [5].

3.3 Legal issues

Open source software is as much about the license as it is about the development of methodology. Since the beginning of 2010, there are roughly 80 licenses [6] that are officially recognized as open source ones. The suppliers of FOSS will offer different kinds of licenses to meet the requirements of different customers ranging from private users to information technology or commercial

software providers. Reading the FOSS license and understanding its various forms are important in order to have a clear policy articulated for an organization [6]. Recognized open source licenses have well defined conditions for the code contribution to any other development of the software or for code incorporation into other packages [7, 8].

3. Conclusion

The philosophy of FOSS represents an alternative approach for proprietary and traditional software engineering processes. The research into FOSS is growing and governments are trying to free their software products from the dominance of large companies and external government policies. More investigations, however, are needed to study the quality of each approach's final product and overcome the issues and challenges of FOSS. The products of FOSS have the advantage of being available to testers and users before the software is released. These testers and users are able to examine FOSS at its early stage and suggested any possible improvements. This paper has presented an overview of the real issues and challenges of migration to free and open source software (FOSS) and suggested some important criteria for assessing its issues and challenges. The issues and challenges to migrate to FOSS are divided firstly into technical issues and challenges. These include performance, technical infrastructure, usability, integrity, support availability, security, information flow control, data migration, flexibility, ease of use and the management and maintenance of OSS. The second set of issues includes non-technical ones and challenges, such as, organizational culture, human factors (staff skills) and legal issues.

References

[1] N. Hasan. Issues and Challenges in Open Source Software Environment with Special Reference to India, technology, policy and innovation, ical 2009.
[2] P. Zuliani, and G. Succi, "Migrating public administrations to open source software". Proceedings of eChallenges on Software Engineering. Vienna, Austria.2004.
[3] Seow, BSA Business Software Alliance, Open source and commercial software. An in-depth analysis of the issues. www.bsa.org/usa/policy. September 2005.
[4] M. Cassell, 5 Factors for Open Source Success, 2011, Available at *http://www.govtech.com/pcio/5-Factors-for-Open-Source-Success-021111.html*
[5] B. Chawner, B, "Free/open source software: new opportunities, new challenges", paper presented at the 12th VALA Biennial Conference – Breaking boundaries: Integration & Interoperability, 2004, Available at. www.vala.org.au/vala2004/2004pdfs/33chawn.pdf
[6] D. Thomas, Going Open Source Software in IT Opportunities and Challenges, "Journal Of Object Technology", Vol. 4, No. 2, March-April 2005. Available at http://www.jot.fm/
[7] Open Source Initiative, Licenses by Name, Available at http://opensource.org/licenses/alphabetical/
[8] M. Huda, "Migration to FOSS: Challenges", 2009, Available at http://www.nctr.sd/images/stories/docs/Migration_to_FOSS.pdf
[9] R. Metcalfe, Top tips for selecting open source software, SS watch Open Source Software Advisory Service,2012, Available at http://www.oss-watch.ac.uk/resources/tips.xml
[10] R. Boateng, Potential Benefits and Challenges of FOSS In Developing Countries, PC Tech Technology/Business/Society, 2011.
[11] J. Nielsen, Usability Engineering. "Academic Press", Boston, ISBN 0-12-518405-0 (hardcover), 0-12-518406-9 (soft cover). Japanese translation ISBN 4-8101-9009-9.1993.
[12] D.M. Nichols, and M. B. Twidale, "The usability of open source software". Vol. 8, No. 1, January 2003. Available from: http://firstmonday.org/article/view/1018/939
[13] M. Huda and M. Hisham, Migration to FOSS: Readiness and Challenges, Free Open Source Software (FOSS) Workshop, June 8th, 2009, Sudan.
[14] H. Marit, K. Kristian and P. Andreas, The Open Source Approach: Opportunities and Limitations with Respect to Security and Privacy. Computers & Security Journal, 2002, Vol. 21, No. 5, Pp. 461-471.
[15] D. Wheeler, Why open source software/free software (OSS/FS, FLOSS or FOSS)? Look at the numbers! Available from, 2005: http://www.dwheeler.com/oss_fs_why.html [Accessed: 05/06/2007].
[16] GOVERNMENT INFORMATION OFFICERS' COUNCIL – GITOC. Using open-source software in the South African government: a proposed strategy compiled by GITOC 2003. [Online]. Available from: http://www.oss.gov.za [Accessed: 16/04/2008].
[17] D. Carlo, M. Jesús, H. Edmund, K. Werner, L. Bernard, L. Ben, A. Philippe, C. Laurent and L. Michel (Working group on Libre Software). Free Software / Open Source: Information Society Opportunities for Europe. Version 1.2, April 2000. [Online]. Available from: eu.conecta.it/paper.pdf.
[18] R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley Computer Publishing. 2001.
[19] M. Sarrab, Runtime Verification of Information flow: Policy-Based Runtime Verification of Information Flow Control, LAP LAMBERT Academic Publishing. 2011.
[20] The University of Sheffield (USFD). Computer Technology Institute & Press "Diophantus" (CTI). Open Source software usage by European Public Administrations (OSEPA). Technical efficiency guidelines for selecting between and among FOSS and proprietary SW solutions. August 2011. [Online]. Available from: http://osepa.eu/pdeliverables/TAL17_3%202%202_-_%20Technical_Efficiency_Guidelines.pdf.

[21] D. Wheeler, How to Evaluate Open Source Software / Free Software (OSS/FS) Programs. [Online]. Available from: http://www.dwheeler.com/oss_fs_eval.html [Accessed: 05/08/ 2011].
[22] E. Michael, E. Chris, R. Irene and C. Brendan. Fault Detection and Prediction in an Open-Source Software Project, Proceedings of the 5th International Conference on Predictor Models in Software Engineering, Vancouver, British Columbia, Canada. Article No. 17. 2009.
[23] B. Fernando and C. Rogério. Candidate Metrics for Object-Oriented Software within a Taxonomy Framework, Originally published in Proceedings of AQUIS'93 Conference, Venice, Italy, October 1993. Selected for publication in the Journal of Systems and Software, 26(1), North-Holland, Elsevier Science, July 1994.

Dr. Mohamed Sarrab is currently working as a research associate at CIRC (Communication and Information Research Center), Sultan Qaboos University, Muscat, Sultanate of Oman. Obtained Ph.D. degree in computer science from De Montfort university, UK. M.Sc. degree in computer science and information technology from VSB technical university of Ostrava, Czech Republic and B.Sc. degree in computer science from Al Zawia University, Libya. His research interests are in areas of computer security, in particular access control and policy-based system management, Runtime Verification and Information Flow Control. He is also interesting in computer forensic, mobile applications, open source software and Java byte code instrumentation for monitoring and security reasons

Mahmoud Elbasir has M.Sc. in information System from De Montfort University 2007. He received his B.Sc. in computer science from Tripoli University Libya a. His main research interests are Information System management, E-commerce, and Software Engineering and Free open Source Software.

Eng. Laila Elgamel has M.Sc. in Software Engineering from De Montfort University 2011. She received her B.Sc. in computer science from 7th April University Libya a. Her main research interests are Mobile Applications, M-learning, Software Engineering and Software Engineering and Free open Source Software.