# SELECTING BETWEEN OPEN SOURCE AND PROPRIETARY SOFTWARE: THE PUBLIC ADMINISTRATORS' CASE[1]

Christos Bouras, Vasileios Kokkinos and Georgia Tseliou
*Computer Technology Institute & Press "Diophantus" and*
*Computer Engineering and Informatics Department, University of Patras*
*D. Maritsas Building, N. Kazantzaki str., University of Patras Campus, GR-26504 Rion, Greece*

**ABSTRACT**

The necessity for the public sector to change over to communicating digitally is imminent. From the economic perspective, the changeover poses great challenges, as huge investments will have to be made in Information Technology (IT) in the public sector. It is therefore natural, in connection with these investments, for detailed assessment to be made of what forms of technology it is anticipated to be used, and who controls the development and ownership of this technology. This work constitutes a review of literature on pre-existing comparative studies regarding the technical, social, economic and organizational factors on Free Open Source Software (FOSS) usage. Furthermore, this work includes guidelines that Public Administrations (PAs) should follow for the selection between open source and proprietary software. Main goal of this paper is to add to knowledge resources that can help public stakeholders understand the technical / social / economic / organizational environment and reach informed decisions when selecting the appropriate software. The paper can also be useful for FOSS developers, users and communities who are either directly or indirectly involved in the software market.

**KEYWORDS**

Open source software; public administration; guidelines;

## 1. INTRODUCTION

Software (SW) can be shortly defined as the executable code that controls computer behavior and operations. The term is used, however, to describe a wide range of programming languages, applications, procedures and all related documentation resources. SW also refers to a full cycle of processes from basic architecture to development, packaging and distributing. It is responsible for controlling, integrating, and managing the individual hardware components of a computer system so that other software and the users of the system see it as a functional unit without having to be concerned with the low-level details of the computational system.

European governments are increasingly considering the use of Free and Open Source Software (FOSS) as a means of reducing costs, increasing transparency and sustainability. A number of argues have taken place on the costs and benefits of open source software. Moreover much discussion and interest has been expressed from the perspective of information technologists.

Although there are different definitions of Free and Open Source Software (FOSS), there are some basic principles on which FOSS relies on. These refer to the freedom to run a software program for any purpose, to study and modify a software program by accessing its source code and to distribute copies of a software program, whether modified or not. Additional prerequisites for FOSS programs include: no discrimination against persons, groups or fields of endeavor and distributable, technology-neutral licenses that are not specific to a product or restrict other software. These freedoms and principles are defined by the Free Software Foundation (http://www.gnu.org/philosophy/free-sw.html) and the Open Source Initiative (http://www.opensource.org/osd.html).

This manuscript examines these advantages and disadvantages of FOSS solutions and analyses the main factors that affect FOSS use and adoption by Public Administrators (PAs). Moreover this work describes some basic and important guidelines that should be followed for the evaluation and adoption of any software.

[1] Work supported by the ERDF - EU National funded Interregional Cooperation Programme (INTERREG IVC) under contract number 0918R2 (project: OSEPA - Open Source software usage by European Public Administrations)

The basic steps for evaluating all programs, both FOSS and proprietary SW, are essentially the same. However, the way that these steps are performed in an evaluation process is different for FOSS programs than for proprietary ones. A key difference for evaluation is that the information available for FOSS programs is usually different than for proprietary programs.

Indeed, most FOSS programs have a great deal of publicly available information that is not available for proprietary programs: the program's source code, analysis by others of the program design, discussions between developers about its design and future directions, discussions between users and developers on how well it is working (or not), and so on. An even more fundamental difference between FOSS and proprietary programs is that FOSS programs can be changed and be redistributed by customers. This difference affects many factors, such as support options, flexibility, customizability and costs. Proprietary programs generally do not give the user the right to view, modify, and redistribute a program, and it would not make sense to ignore these vital differences. Some administrators may decide that they wish to only use FOSS programs. However, even in that case, the user still needs to be able to evaluate FOSS programs, because there is always the need to know how well a given program meets the user's needs, and there are often competing FOSS programs.

The remainder of this paper is structured as follows: Section 2 describes in detail the work related with our study. A detailed list of guidelines for selecting between FOSS and proprietary software is provided in Section 3. Finally in Section 4 our conclusions and some proposals for future work are drawn up.

## 2. COMPARATIVE STUDIES / SURVEYS ON FOSS USAGE

This section constitutes a review of literature on pre-existing comparative studies and surveys regarding the technical, organizational, economic and social factors on FOSS usage. The surveys were executed in various regions or sectors where FOSS is applied.

In FLOSS (2002), a survey that is intended to yield information about FOSS use in several countries of the European Union is presented. Due to budgetary restrictions, interviews could only be conducted for a limited number of countries (Germany, Sweden and the UK). One of the results of this survey is that FOSS usage rates not only differ by country, but also within countries. Another survey (Ölsson and Rönnbäck 2010) that was conducted in Sweden answers the question of how common the usage of FOSS is, by informing the public that 50% of the local authorities use FOSS, mainly in operating systems. Moreover, as the survey pointed out, there is a great need for support in procurement and utilization of FOSS.

The purpose of the study of Danish Board of Technology (2002) is to illustrate the socio-economic differences between the use of FOSS and proprietary software in PAs in Denmark. The conducted socio-economic analysis assesses the total loss that follows from decisions taken against the background of limited information and imperfect market competition. The survey of Rentocchini and Tartary (2007) presents some obstacles to the adoption of Information and Communications Technology (ICT) by PAs. The survey noted immediately that there are differences among municipalities with different intensity degree. Municipalities with a high intensity of FOSS adoption, rate the low flexibility of suppliers and the low interoperability of applications as the main obstacles to a correct implementation of the ICTs. For the two other groups, namely moderate intensity and no intensity, main obstacles are the low number of employees and high costs.

Considering the survey presented in (Public Sector Forums 2009), it was conducted in UK and according to it, almost two-thirds of those surveyed believe the benefits of open source generally outweigh the drawbacks. However the general consensus is that local government fails to give sufficient consideration to open source in software procurements. The research finds that open source use in local government will, overall, only keep increasing. The majority view (42%) is that local authorities will increase their use of FOSS over the next three years. Around a third of those surveyed expect current levels of adoption to remain unchanged during this period. This highlights a significant degree of uncertainty among sections of local government over plans for future adoption.

Taking in mind the analysis of Moolman (2011), it must be noted that technological factors affect FOSS in a large scale. People that support the adoption of FOSS believe that FOSS shows more stable behavior than proprietary software. Dedrick and West (2008) claim that in organizations the use of FOSS still has to be motivated on utilitarian grounds. Technological factors that show a relevance to FOSS adoption include maturity, performance, stability, usability, security such as availability and quality of support.

As stated in (Moolman 2011), previous experience with FOSS plays a significant role in the ability to choose such kind of software. It is rather usual that organizations with little or no experience in FOSS are better off choosing software. This happens due to the fact that mature FOSS solutions supported by commercial companies and universities generally present a lower risk as they have been adopted by many organizations and documentation and support is available. It is quite interesting to observe that several FOSS projects considered immature when measured with maturity models are mature enough for adoption, given that the adopting organization has some FOSS experience (Ven et al. 2008).

The same authors mention the maturity of the organization dealing with FOSS in (James and Van Belle 2008). Their measure of maturity also takes into account the intended application within the organization, availability of support and the maturity of the development community behind the software. They highlight maturity factors that are organization-centric, solution-centric or external entity-centric. They found that the maturity of the solution under review is dependent on its intended application within the organization.

Software maturity is a decision factor that depends on the environment in which the software is used (James and Van Belle 2008; Holck et al. 2005). Reliability is an important aspect of software maturity and mature software is also seen as reliable. Reliability comparisons between FOSS and proprietary software are almost futile as both software types cover a range of software from extremely stable to rather unstable.

An organizational factor that affects the adoption of FOSS is lack of awareness that can be remedied by having FOSS advocates and boundary spanners working in an organization. Definitely boundary spanners are effective in connecting organizations to new technologies and provide the skills and knowledge needed for successful adoption (Ven and Verelst 2009). FOSS champions successfully influence adoption decisions from within an organization, reducing some of the individual uncertainty and fear (Morgan and Finnegan 2007). The amount of influence FOSS champions have within an organization is determined by the institutional limitations in the organization and their position within the organization (Holck et al. 2005). There are many economic factors that can be considered in social environments and affect the adoption of FOSS. A business benefit that can be considered is cost reduction in relation to technical benefits and drawbacks of FOSS adoption (Morgan and Finnegan 2007).

The business case of FOSS adoption is driven by lower costs, but it is also dependent on the application area, company size and price elasticity in the market. Application area and adoption scale is important as it might be prohibitively expensive to make a company-wide switch from one platform to another (Holck et al. 2005). The level of strategic importance of software to the business also plays a role in adoption decisions. Software with low strategic importance and high price sensitivity tend to be better candidates for FOSS adoption (Kwan and West 2005).

Although the low price of FOSS products is the primary factor for using these products, there are also other economic perspectives, not only in using FOSS but also in developing products. Four economic incentives for the adoption of FOSS software and support its development by governments are the following:

- Control the costs of software licensing and upgrades,
- Control and increase the access to intellectual properties,
- Reduce the reliance on proprietary software,
- Promote software use in the public sectors.

It is interesting to observe that cost as a factor in FOSS adoption decisions depend on an objective measurement of cost. The authors in (Richter et al. 2009) found that for many companies, FOSS adoption is centered on value creation. The advantage however comes not only from costs which are saved but benefits from reliability, flexibility and a higher degree of innovation capability.

Developing countries, in general, adopt FOSS due to cost advantages. The effect of software license fees are more pronounced in developing countries as it makes up a larger part of total system cost when taking into account hardware and software. Lower labor costs mean that license fees constitute a bigger percentage of IT costs (Paudel et al. 2010). One interesting example occurs in the German public sector where low cost is one of the main drivers of FOSS adoption. The German foreign office started migrating to FOSS in 2002 and by 2005 it was the cheapest ministry in German government in terms of IT expenditure. In Brazil, government uses FOSS to save on license fees, keeping money that was previously paid to foreign vendors inside the country (Richter et al. 2009). Through collaboration with local industries costs can be minimized and national competitiveness in software industries can be improved (Hwang 2005).

Table 1. Factors affecting FOSS usage and adoption

| *Factor* | | *Surveys* | | | | | |
|---|---|---|---|---|---|---|---|
| | | FLOSS | Sambruk | Danish Board of Technology | EROSS | Public Sector Forum | CENATIC |
| Technological | Functionality | X | X | | | | |
| | Support | X | X | | | | |
| | Maintenance | X | X | | | | |
| | Management | X | | | | | |
| | Longevity | X | | | | | |
| | Reliability | X | | | | | |
| | Availability | X | | | | | X |
| | Security | X | | | | | |
| | Performance | X | | | | | |
| | Usability | X | | | | | |
| | Interoperability | X | | | | | X |
| | Integration | X | | | | | X |
| | Trialability | X | | | | | |
| | Data Migration | X | | | | | |
| Organizational | Ability to find the right staff and competencies | | X | | X | X | |
| | Lack of awareness | | X | | X | X | |
| | Training issues | | X | | X | X | X |
| | Resistance to change | | X | | X | X | |
| | Strong leadership | | X | | X | X | X |
| | Management support | | X | | X | X | |
| | Availability of in-house skills and knowledge | | X | | X | X | |
| | Real world experience | | X | | X | X | |
| | Interoperability of applications | | X | | X | X | |
| Cost / Economic | Labor costs | | | X | X | X | X |
| | Control the costs of software licensing and upgrades | | X | X | X | X | X |
| | Control and increase the access to intellectual properties | | | X | X | X | X |
| | Promote software use in the public sectors | | | X | X | X | |
| Social | Knowledge sharing | | | X | | | |
| | Satisfaction of achieving something valuable | | | X | | | X |
| | Professional reputation and recognition among peers | | | X | | | |
| | Learning/Improving personal skills | | | X | | | X |
| | Legal aspects | | X | X | | X | |
| | Sense of belonging to the community | | | X | | | |
| | Enjoyment of developing | | | X | | | X |
| | Sharing knowledge | | | X | | | |
| | Improving products | | | X | | | |
| | Freedom in developing SW | | | X | | | |
| | Learning and developing new skills | | | X | | | X |
| | Sense of belonging to the community | | | X | | | X |

IT professionals from four continents have collaborated with the National Open Source Observatory (CENATIC), and their opinions and suggestions have served as the basis for the conclusions set out in the dossier presented in (CENATIC 2011). In terms of the criteria for adopting FOSS, the dossier concludes that the administrations are influenced by criteria such as vendor independence and flexibility, open standards and open development process. However, the public administrations are less easily persuaded by criteria such as faster procurement, best-of-breed solutions and political decisions and initiatives.

In Table 1 we analyze the most significant factors that occurred from the examination of the comparative studies and surveys.

## 3. GUIDELINES FOR SELECTING AMONG FOSS AND PROPRIETARY SOFTWARE

In this section, we describe some basic and important guidelines that should be followed by organizations or PAs for the adoption of any software. The basic steps for evaluating all programs, both FOSS and proprietary SW, are essentially the same. However, the way that these steps are performed in an evaluation process is different for FOSS programs than for proprietary ones. A key difference for evaluation is that the information available for FOSS programs is usually different than for proprietary programs. In Figure 1 we present the steps that should be followed for selecting among FOSS and proprietary software. The paragraphs that follow present these steps in detail.

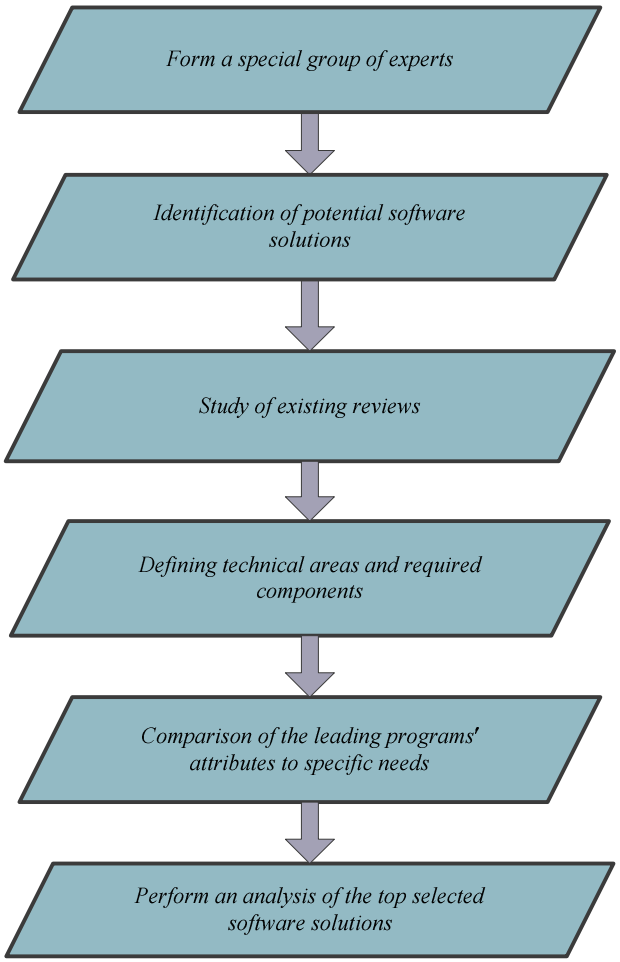


Figure 1. Guidelines for selecting among open source and proprietary software

## 3.1 Formation of a Special Group of Experts

Before the software search begins, it is necessary to form a search group of experts. This group should consist of the computer department and various department heads. This kind of approach has worked very well for many companies and PAs. By pairing the computer department that specializes in technology with the heads of departments who know the business needs, the interested company or PA develops a very strong software search team. This team should define the specific needs, the role in infrastructure and development projects, and cost saving trade-offs.

The most successful installations have been with companies that had this kind of committee, in which the computer department becomes the liaison between the users and the software implementation team translating technology to their requirements (FLOSS 2002).

## 3.2 Identification of Potential Software Solutions

A combination of techniques should be used in order to make sure that something important is not missed. An obvious way for the interested user is to make a questionnaire, if other users (organizations or PAs) also need or have used such a program. If they have experience with it, they should ask for their critique; this will be useful as input for the next step, obtaining reviews.

Moreover it is necessary to examine at lists of programs, including any list of "generally recognized as mature" or "generally recognized as safe" programs. Some products are so well-known that it would a terrible mistake to not consider them. It is advised to the interested user to ask only a few of the most relevant lists. Also general systems can be used to make requests, such as Google answers, where someone pays a fee to get an answer. The search group of experts is proper to make a more detailed search.

## 3.3 Study of Existing Reviews

After the identification of options, it is necessary to study all the existing evaluations about the alternatives. It is far more efficient to first learn about a program's strengths and weaknesses from a few reviews than to try to discern that information just from project websites. It is critical that many evaluations are biased or not particularly relevant to any circumstance. An important though indirect "review" of a product is the product's popularity, also known as market share.

Generally, a user should always try to include the most popular products in any evaluation. Products with large market share are likely to be sufficient for many needs, are often easier to support and interoperate, and so on. It is important to develop a documentation plan in support of communication and awareness of the organization's governance strategy. In addition to traditional documentation, this may include training, internal public relations campaigns, and other educational opportunities.

Developers do not want their work wasted, so they will want to work with projects perceived to be successful. Conversely, a product rapidly losing market share has a greater risk, because presumably people are leaving it for a reason.

## 3.4 Definition of Technical Areas and Required Components

It is very important, in any software selection or migration project, to have a clear view of the technical areas (server, client and network) and software components (both open source and proprietary) that are required for installation and deployment. Server-based systems, for example, require pre-existing web or application servers and more advanced installation and configuration processes. Some applications also require a parallel deployment or co-existence of both open source and proprietary components that should be carefully taken into account in order to avoid compatibility failures.

## 3.5 Comparison of the Leading Programs' Attributes to Specific Needs

There are some attributes that should be taken into consideration as far as the choice between FOSS and proprietary SW is concerned. Important attributes include functionality, cost estimation (initial license fees, license upgrade fees, installation costs, staffing costs, support/maintenance costs, indirect costs such as training, transition costs, etc.), market share, local policies and other environmental and social factors. For example, licensing costs are not the only costs of a software package or infrastructure. It is also necessary to consider personnel costs, hardware requirements, opportunity costs and training costs. Often referred to as the Total Cost of Ownership (TCO), these costs give the clearest picture of the savings from using FOSS.

The benefits, drawbacks, and risks of using a program can be determined from examining these attributes. The attributes are the same as with proprietary software, of course, but the way that a user should evaluate them with FOSS and proprietary SW is often different. In particular, because the FOSS project and code is completely exposed to the world, the user must take advantage of this information during evaluation.

## 3.6 Performance of an Analysis of the Top Selected Software Solutions

After the evaluation, the organization picks the top candidates, and performs a more analysis of them. This step is, for the most part, done the same way for both proprietary and FOSS programs. The important attributes to consider are the same as in the previous step.

More effort is spent by actually trying things out instead of quickly reading the available literature. For example, to see what functionality a program provides, a user would run it and try out the functionality that he/she is interested in using (e.g., if the user is concerned about interoperability, he/she will acquire some sample same files or systems and see how well it works). A user should always carefully identify the version number of the program, because the description of the first version may not be the same in a later one. This is particularly important for FOSS programs, because many FOSS programs undergo rapid improvement.

A more important difference is that in FOSS there are sources of information about a program that may not be available for proprietary software. In particular, a user can also have a software professional examine the program's design documentation, source code, and other related materials. The conducted analysis can be categorized in analysis for adding functionality and analysis of software security

Once a decision has been made, it is time to begin the process to install the new program.

# 4. CONCLUSIONS AND FUTURE WORK

In this work a detailed list of guidelines for selecting among open source and proprietary software is presented. Concluding this work, it can be anticipated that FOSS will be more widely used over the next few years, as a smaller proportion of those organizations or PAs that do not use FOSS expect to do so in the next two years, while a larger proportion of the organizations that already use FOSS expect to increase their use of it. The decisive reasons why FOSS is used or is not used mainly relate to economic savings, awareness of FOSS, compatibility, the development of programs and user-friendliness.

Local and regional authorities are often better positioned to directly integrate open source systems and applications in their internal processes and IT architectures by clearly defining needs and specifications through public tenders. By adapting open source solutions to regional contexts through extensive customization and localization they can also see immediate effects and improvements in administrative tasks or in services delivered to local communities.

This study serves to improve the knowledge of FOSS use in PAs. However, further room for research into FOSS, an increasingly important aspect of ICT adoption and growth in PAs is suggested. Further research on FOSS adoption in PAs could include the quantitative study of FOSS adoption and the study into the availability and perception of FOSS vendors. One more proposal for future work would be the provision of policy recommendations on issues and challenges pertaining to the use of FOSS by European PAs. The aim of this policy recommendation work would be the contribution in providing policy orientations and proposed actions that can help governments, PAs and European institutions fully harvest the benefits of FOSS.

## REFERENCES


CENATIC, National Open Source Observatory, 2011. *Criteria for adopting open source software in Public Administrations*. Available from: http://observatorio.cenatic.es/index.php?view=article&catid=5%3Aadministraciones-publicas&id=723%3Acriterios-para-adoptar-el-sfa-en-la-administracion-publica&tmpl=component&print=1&layout=default&page=&option=com_content&Itemid=21.

Danish Board of Technology, 2002. *Open-Source Software in E-Government. Analysis and Recommendations Drawn Up by a Working Group Under the Danish Board of Technology*. Available from: http://xml.coverpages.org/DanishTeknoOpenSource200310.pdf.

Dedrick, J. and West, J., 2008. Movement Ideology vs. User Pragmatism in the Organizational Adoption of Open Source Software. *In* Kenneth L. Kraemer and Margaret Elliott, eds., *Computerization Movements and Technology Diffusion: From Mainframes to Ubiquitous Computing,* Medford, NJ: Information Today, 427-452.

FLOSS Final Report, 2002. *Free/Libre Open Source Software: Survey & Study*, *Use of Open Source Software in Firms and Public Institutions, Evidence from Germany, Sweden and UK*, Berlin. Available from: http://www.infonomics.nl/FLOSS/report

Holck, J. et al, 2005. *Managerial and Technical Barriers to the Adoption of Open Source Software*, COTS-Based Software Systems, Lecture Notes in Computer Science, Berlin: Springer-Verlag.

Hwang, S., 2005. Michigan Journal of Public Affairs*, Adopting Open Source and Open Standards in the Public Sector: Five deciding factors behind the movement*, Volume 2. Available from: http://www.mjpa.umich.edu/uploads/2/9/3/2/2932559/hwang_-_mjpa_v2_final.pdf.

James, S. and Van Belle, J.P., 2008. Ensuring the Long-term Success of OSS Migration: A South African Exploratory Study, *6th Conference on Information Science Technology and Management*, New Delhi, India, 8--1 to 8--15.

Kwan, S. and West, J., 2005. A Conceptual Model for Enterprise Adoption of Open Source Software. *In* Sherrie Bolin, ed., *The Standards Edge: Open Season,* Ann Arbor, Mich.: Sheridan Books, 274-301.

Moolman, L., 2011. *A characterization of Open Source Software adoption decisions in South African organizations*, University of Stellenbosch. Available from: http://scholar.sun.ac.za/bitstream/handle/10019.1/8533/moolman_characterisation_2011.pdf?sequence=1.

Morgan, L. and Finnegan, P., 2007. How perceptions of open source software influence adoption: An exploratory study, *Proceedings of the 15th European Conference on Information Systems*, June 7th -9th, St Gallen, Switzerland.

Ölsson, C.-O. and Rönnbäck, A.O., 2010. Collaborative development of public information systems: A case study of "Sambruk" e-services development, *In*: eChallenges e-2010 Conference Proceedings Paul Cunningham and Miriam Cunningham, eds., *IIMC International Information Management Corporation,* 27-29 October, 1 – 7.

Paudel, B. et al, 2010. Open Technologies and Developing Economies. *Proceedings of CAN InfoTech (IT) Conference 2010,* January.

Public Sector Forums, 2009. *Open or Closed?, A Survey of Open Source Software in Local Government*. Available from: http://ukgovoss.org/open-or-closed/.

Rentocchini, F. and Tartary, D., 2007. *Open Source Software in the Public Sector: Results from the Emilia-Romagna Open Source Survey (EROSS)*, Available from: http://oss-institute.org/whitepapers/EROSS_Italy. pdf.

Richter, D. et al, 2009. A Comparative Analysis of Open Source Software Usage in Germany, Brazil and India. *In* Computer Sciences and Convergence Information Technology, *Proceedings of Fourth International Conference on Computer Sciences and Convergence Information Technology* (ICCIT '09), 24-26 November Seoul, 1403 – 1410.

Ven, K. et al, 2008. Should You Adopt Open Source Software?, *In Software IEEE*, Volume 25, Issue 3, May-June 2008, 54-59.

Ven, K. and Verelst, J., 2009. The Importance of External Support in the Adoption of Open Source Server Software. *In: Editor Open Source Ecosystems: Diverse Communities Interacting*, *IFIP Advances in Information and Communication Technology*, Boston: Springer, 116-128.