Telematics and Informatics 30 (2013) 100–110

Contents lists available at SciVerse ScienceDirect

Telematics and Informatics

journal homepage: www.elsevier.com/locate/tele

Methodology for Public Administrators for selecting between open source and proprietary software ☆

Christos Bouras *, Vasileios Kokkinos, Georgia Tseliou

Computer Technology Institute & Press "Diophantus", Patras, Greece
Computer Engineering and Informatics Department, University of Patras, Greece

ARTICLE INFO

Article history:
Received 10 November 2011
Received in revised form 2 March 2012
Accepted 7 March 2012
Available online 21 March 2012

Keywords:
Open source software
Proprietary software
Public Administration guidelines

ABSTRACT

The public sector needs to change over to communicating digitally. This development makes great demands both on work processes in the public sector and on the Information Technology systems, on which e-government is based. From the economic perspective, the change-over poses great challenges, as huge investments will have to be made in Information Technology in the public sector. It is therefore natural, in connection with these investments, for a 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. The question is: to what extent Free and Open Source Software can supplement or completely replace proprietary software? This work constitutes a review of literature on pre-existing comparative studies regarding the technical, social, economic and organizational factors on Free and Open Source Software usage. Furthermore, this work includes guidelines that Public Administrations should follow for the selection between open source and proprietary software. Our goal is to help public stakeholders understand the technical/social/economic/organizational environment and therefore reach informed decisions when selecting the appropriate software. The manuscript can also be useful for Free and Open Source Software developers, users and communities who are either directly or indirectly involved in the software market.

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 in order for other software and the users of the system to be able to see it as a functional unit without having to be concerned with the low-level details of the computational system.

Although there are different definitions of Free and Open Source Software (FOSS), there are some basic principles on which FOSS relies. 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

☆ 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).

* Corresponding author at: Computer Technology Institute & Press "Diophantus", Greece, N. Kazantzaki, GR-26504 Patras, Greece. Tel.: +30 2610 996951; fax: +30 2610 969016.

E-mail address: bouras@cti.gr (C. Bouras).

0736-5853/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.tele.2012.03.001

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).

FOSS has become very popular in the last few years and is advancing at a speed unknown outside the world of Information Technology (IT). Just a few years ago, FOSS was not regarded as a popular solution compared to the solutions provided by the giants of IT. Today, FOSS has become an area of business – an alternative, and therefore a competitor, to proprietary software.

This interest has also spread to the world of politics. This occurs due to different parameters: FOSS is now increasingly used for commercial purposes, it is characterized as independent from software producers, it is opposed to the creation of monopolies and it is characterized by a "free-of-charge principle". Furthermore, FOSS has had a great impact on the political agenda, nationally and internationally.

FOSS-solutions have several advantages and disadvantages compared to proprietary SW solutions. This manuscript 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. 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 parameters, 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 this 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 rest of this manuscript is structured as follows: Section 2 describes in detail the work related to our study and Section 3 describes the factors that affect FOSS-use and adoption by PAs. A detailed list of guidelines for selecting between FOSS and proprietary software is provided in Section 4. Finally in Sections 5 and 6 our conclusions and some proposals for future work are drawn up.

2. Review of literature

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

In Floss Final Report (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 and technological issues not only differ by country, but also within countries. Another survey (presented in Olsoon et al. (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. Also, this work presents examples that concern functionality, support and maintenance of FOSS-issues.

Taking in mind the analysis in Moolman (2011), it must be noted that technological factors affect FOSS on a large scale. People that support the adoption of FOSS believe that FOSS shows more stable behavior than proprietary software. The authors of Dedrick and West, 2007 claim that the use of FOSS in organizations still has to be motivated on utilitarian grounds. Technological factors that show a relevance to FOSS-adoption include maturity, performance, stability, usability, security and quality of support.

As stated in Moolman (2011), previous experience with FOSS plays a significant role in the selection of such kind of software. Usually, 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 as well as support is available. It is quite interesting to observe that several FOSS-projects considered immature when measured with maturity models are in fact 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 concluded 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