ISSN 1410 - 9018

SINERGI
KAJIAN BISNIS DAN MANAJEMEN
Vol. 10 No. 1, Januari 2008
Hal. 1 - 10

TOTAL COST OF OWNERSHIP OF OPEN SOURCE

Fitriati Akmila
School of Economics and Commerce
University of Western Australia
Perth, Australia
Email: akmilf01@uwa.edu.au.

Abstract

There has been a huge interest among academics and practitioners in open source. This is because open source offers business with its programmers the opportunity to elaborate and adapt source code. The success of open source leads to an increasing controversy of the price of its software. In other words, is the total cost of ownership of open source really lower than that of proprietary software?

This paper describes the concept of open source, compares it with proprietary software, and total cost of ownership (TCO) of open source. An example of Linux vs. Windows or Unix is taken to describe the cost of open source.

In brief, this paper sums up open source is cheaper in terms of direct cost (hardware, software and support). However, it is difficult to measure the indirect costs of open source since these costs are hidden and may vary within the business and software environment.

Keywords: *Open Source; Proprietary Software; Total Cost of Ownership*

INTRODUCTION

Open source is the process of building and improving "free" software by an internet community (Pearlson and Saunders 2006). In this case the software is developed through communities of programmers, who collaborate voluntarily. The development of open source is build like "bazaar" in which the communities have the freedom and flexibility to run, copy, distribute, and improve the software as illustrated by Eric Raymond (1999) below:

> "I also believed there was a certain critical complexity above which a more centralized, a priori approach was required. I believed that the most important software…needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time. Linus Torvald's style of development – release early and often, delegate everything you can, be open to the point of promiscuity – came as a surprise. No quiet, reverent cathedral building here – rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches… out of which a coherent and stable system could seemingly emerge only by a succession of miracles. The fact that this bazaar style seemed to work, and work well, came as a distinct shock" (Raymond 1999).

Open source software presents issues of increasing importance for business and organizations. Nowadays, open source has been adopted by many companies. In November 2003, CIO survey of 375 information executives reported that 54 percent

said that within five years open source would be their dominant platform (Koch 2003). Moreover, big vendors such as Dell, IBM and Sun Microsystems support open source development, and are developing application that would work with open source platforms (Sen 2007).

The success of open source has led to an increasing controversy of the price of its software. Although open source is often thought as a free alternative but then is it really cheaper when we add the costs of acquisition, migration, operation, and support? In other words is the total cost of ownership of open source really lower than that of proprietary software?

This paper objective is to find the answer of total cost of ownership (TCO) and open source. Total cost of ownership refers to all costs associated with the use of computer hardware and software including the administrative costs, license costs, hardware and software updates, training and development, maintenance, technical support and any other associated costs (GartnerGroup 1998).

This paper is organized as follows. Firstly, it provides a description of the concept of open source, comparison between open source and proprietary software, review of prior studies, and the concept of total cost of ownership. The discussions will be the scenario used on total cost of ownership of open source. In this case the writer will describes the comparison of cost between open source (Linux) with proprietary software (Unix and Windows) using the information provided by Kenwood (2001). Finally, the conclusion of this paper will appear as a result of this research.

THE CONCEPT OF OPEN SOURCE

Open source is software that has source code that is open, viewable, unrestricted and redistributable. It is available by downloading it from the Internet. When open source is downloaded from the Internet the users of that software are required to adhere to the license agreements of the software. Licenses for open source provide an unconditional right of any party to modify the software and allow unlimited distribution (Hubley and Muller 2002). Furthermore, Bitzer and Schroeder (2006) mentioned that the basic idea and main aim of open source is free usage and the possibility for further development by the user. “Free”, in this context, does not mean free price, but free to duplicate, modify, and distribute open source. The term “open source” covers a number of different forms of software licenses and ensures that the users have the free access of its licenses.

According to OSI (Open Source Initiatives), there are ten minimum requirements that must be met to qualify for the “OSI Certified” label. The aim for these requirements is to encourage the development of a different kind of open source licensing models. The OSI requirements can be seen in Table 1 below:

Table 1: Criteria of open source according to Open Source Initiative

1.	**Free redistribution** The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2.	**Source code** The program must include source code, and must allow distribution in source code as well as compiled form.
3.	**Derived works** The license must allow modification and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4.	**Integrity of the author's source code** The license may restrict source-code from being distributed in modified form only if the license allows the distribution of 'patch files' with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code.
5.	**No discrimination against persons or groups** The license must not discriminative against any person or group of persons
6.	**No discrimination against fields of endeavor** The license must not restrict anyone from making use of the program in a specific business, or from being used for genetic research.
7.	**Distribution of license** The right attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8.	**License must not be specific to a product** The right attached to the program must not depend on the program's being part of a particular software distribution.
9.	**License must not restrict other software** The license must not place restrictions on other software that is distributed along with the licensed software.
10.	**License must be technology-neutral** No provision of the license may be predicated on any individual technology or style of interface.

Source: www.opensource.org (as quoted by Picot, Fielder & Hummel 2004)

Based on those criteria above, Towle, McFarland & Keepler (2004) then categorized the first three criteria as the main characteristic of open source. These three criteria are mentioned below:

- No prohibitions on distribution, and no payment should be made for distribution unless for support costs.
- The source code must be made available.
- Modification of the source must be permitted for the development of the open source.

OPEN SOURCE COMPARED WITH PROPRIETARY SOFTWARE

As opposed to open source, proprietary software is software that is not free or semi free. Its use, redistribution or modification is prohibited, or requires us to ask permission or is restricted so much that we effectively cannot do it freely (Fuggetta 2003). The key difference between open

source and proprietary software is the access to the source code. Proprietary software offer closed code software with a full documentation and support service while open source offer "free" access to source code. Open Source is also less user-friendly compared to Close Source (Berger 2002). This is due to the fact that everyone can have the authority to manipulate and change the source code. A general technical knowledge might be needed to use this Software. Usually high end users have the ability to maintain this knowledge. However, proprietary software is users friendly and provides support for the development of the software. Other differences between open source and proprietary software are the costs of software in terms of licenses and the conditions or copyright. The copyright of open source software belongs to the author rather than the vendor. The Examples of open source and proprietary application are described in the Table 2.

Table 2: Open Source vs. Close Source (Howels 2007)

	Close Source	Open Source
Applications	Microsoft Office	Open Office
Operating System	Windows, Mac	Linux
DBMS	Oracle	MySQL
Server	IBM Web Sphere	Tom Cat

PREVIOUS STUDIES ON THE COST OF OPEN SOURCE

There are some studies regarding TCO of open source. These studies usually compared the TCO of open source with proprietary software. The reason for this is to get a clear picture of the true cost of open source. The studies are summarized as follows:

- Paul Kavanagh (2004) developed a cost analysis of TCO of open source. He compared open source price with the cost of closed system software. He used simple TCO to include other costs involved over a reasonable period of time when making a software decision. The cost elements that that he used were staffing, hardware and software. Staffing is the dominant cost of open source implementation even where open source already saves millions. Based on his experiment, he found that systems employing open source will usually be less expensive than alternatives. Moreover, in situations where there are many users running desktop software or server client access, licenses software cost maybe high.
- Ideas International Inc (2005) reported five findings concerning the price of Windows and Red Hat Linux: (1) Microsoft's Windows Server 2003 enterprise license and support costs are competitive with Red Hat Enterprise Linux. (2) Linux and Microsoft have different approaches for their licensing and support costs. Considering the long term period of ownership from three to six years is the best way to compare the costs ownership for both servers. (3) Microsoft separate support costs from license fee structure, while Red Hat mixed those two costs. Although the costs might be different, this can be the advantage of Microsoft. (4) Microsoft purchase cost is good for large organizations while Linux price is best for retail industries. (5) In the long term Microsoft can give substantially lower costs compared to Linux.
- The cost of acquisition of Linux is cheaper than proprietary software. However, this cost only have small portion of calculating the TCO of a PC solution. Furthermore, the support cost of Linux (Internal support cost) is also lower than proprietary software this is because there were an increase in the system stability, the reduction of the

rampaging virus and the absence of the need to reboot or reload the operating system (Pavlicek 2002).

- Bloor research as quoted by Kenwood (2001) concludes that the overall winner is Linux, although the difference with NT is often small. Linux comes out on top for file, print, Web, or mail servers, as well as for mixed workloads. In a database server environment, there is little or no difference between Linux and NT. Windows NT is better for application servers because there is so much more software available for this platform.
- The TCO of open source compared to proprietary software depends on the need of the organization environment. Organizations have to be able to identify their requirement of the types of application and estimate all the most crucial cost drivers including the hidden costs (Wheeler 2004).
- The conventional wisdom that open source is cheaper than proprietary software is not always true. For instance, a Windows server 2003 or Solaris 10 costs less than equivalent Linux editions. Furthermore it is important to look beyond short term cost advantages and consider life-cycle costs. These include evaluations, licenses, support, your own enhancements to and maintenance of the open source product, training, tests of subsequent open source iterations, and bug corrections (Ebert 2007).
- Open source although it is "free", it is not necessary cheaper than close code software. Software price is low relative to total TCO (less than 10%). Staffing costs can be up to as much as 50% to 70% of a software system (MacCormack 2003).
- There are claims that open source has a lower TCO than proprietary software and counter-claims that open source has a higher TCO. The distinction between these different TCOs is that: (1). A TCO using proprietary software sees an emphasis placed on the purchase of software licenses; while (2). A TCO using open source software sees an emphasis placed on the investments being located in people than licenses (Moyle 2004).

TOTAL COST OF OWNERSHIP AND OPEN SOURCE

Total Cost of Ownership Concept

The concept of total cost of ownership (TCO) was introduced by Bill Kirwin, the vice president and research director at Gartner Group Inc. In 1987 he applied his TCO model to desktop systems. Gartner extended this model into a wide range of computer technologies. The TCO concept was originally developed to assist private companies determine whether it was making gains or loses from deploying specific technology implementations (Moyle 2004). TCO then is often used to assess the effectiveness of an organization's IT expenditures. TCO includes all expenses related TCO includes all expenses related to owning and maintaining a personal computer or workstation within an organization. Reducing TCO can adversely affect IT service levels; in fact, IT costs are thought to be "directly proportional" to IT service levels (Lacity and Hirschheim 1998). TCO can be divided into two main sets of cost factors: acquisition costs and administration costs (David et al. 2002). The description of these cost categories can be seen in Table 3.

Table 3: Categories of TCO factors (David et al. 2002)

Acquisition Costs	Administration Costs	
	Control	**Operations**
— Hardware — Software	— Implementation and maintenance of centralization — Implementation and maintenance of standardization	— Support — evaluation — installation/upgrades — training — downtime — futz — auditing — viruses — power consumption

Table 4: Cost Element for Open Source

Direct Costs
Software and Hardware
Software
Purchase Price
Upgrades and additions
Intellectual property/licensing fee
Hardware
Purchase Price
Upgrades and additions
Support Costs
Internal
Installation and set-up
Maintenance
Troubleshooting
Support tools (e.g. books, publications
External
Installation and set-up
Maintenance
Troubleshooting
Staffing Costs
Project management
Systems engineering/development
Systems administration
Vendor management
Other administration
Purchasing
Other
Training
De-installation and Disposal
Indirect Costs
Support Costs
Peer Support
Casual Learning
Formal Training
Application development
Futz factor
Downtime

Regarding TCO of open source, Kenwood (2001) under MITRE Corporation developed a business case study of open source. Kenwood (2001) suggests that a decision on open source and proprietary source based on three factors:

(1) Direct costs and indirect costs.

Direct costs are defined as the total lifecycle costs of a system. While, indirect costs are operational costs.

(2) Benefits such as performance, scalability, reliability, and functionality

(3) Other costs (intangible costs)

These costs are difficult to calculate, however, they have direct impact on the effectiveness of implementation of open source and proprietary software.

Based on these assumptions, Kenwood then developed cost taxonomy of open source (see Table 4).

Linux vs. Windows

This section provides an example of total cost of ownership of open source and proprietary software. This example based on the cost taxonomy which basically derived from the concept of TCO from Gratner Group. Some of the data are taken from the case study of open source by Kenwood (2001) for MITRE Corp in 2001. This is because the data are still relevant. However, modifications of the costs are made by the writer to adjust with the current costs of

the software. All costs are incurred in the US Dollar.

Direct Costs

As mentioned in the preceding section, direct costs are costs the have direct impact to cost of software. These costs are lifecycle costs which include software, hardware, support costs and de-installation and disposal. The detailed descriptions and estimations of these costs defined as follows:

1. Software and Hardware
   a. Software
      Linux software cost is free by downloading from the internet. However, if we purchase it from the vendor the price will be $60 with no licensing fee (Kenwood 2001) or at least less than $100 (Wheeler 2004). The cost of Microsoft Windows NT is ranging from $600 to $800 for five users and $35 for each additional user. Additional features in Microsoft, such as telnet, news server, better DNS server, and disk quotas can run about $3,800; these features are included in Linux at no extra charge. Meanwhile, Unix software costs from $1,000 to $5,000 or $15,000 for unlimited users licenses (Kenwood 2001).
   b. Hardware
      Linux hardware costs are cheaper than Windows and Unix. Wheeler (2004) illustrates these costs by the following example: The minimum requirements for Microsoft Windows 2000 Server are a Pentium-compatible CPU (133 MHz or higher), 128 MB of RAM minimum (with 256 MB the "recommended minimum"), and a 2 GB hard drive with at least 1.0 GB free. According to Red Hat, Red Hat Linux 7.1 (a common distribution of GNU/Linux) requires at a minimum an i486 (Pentium-class recommended), 32MiB RAM (64MiB recommended), and 650MB hard disk space (1.2 GB recommended). Therefore, Linux gives lower hardware costs rather than Windows since the users do not have to upgrade their previous hardware
2. Support
   a. Internal Support
      Average annual labor costs for help desk support are divided into two: help desk manager $58,733 and help desk operator $34,713 (Wageweb 2003). With open source, it is possible for problems to be fixed internally by the user's organization. For proprietary software, problems must be fixed by its supplier so this must be creates additional cost to supplier rather only labor costs. This additional cost is charged differently in each organization based on their need.
   b. External Support
      External support for Linux can be separates into two costs: Basic support costs $179 and premium support $2,499 (RedHat 2007). On the other hand, Microsoft product services with full time equivalent costs upward of $250,000 and contracts involving a named contract with some number of incidents charges start from $50,000 annually (Kavanagh 2004).
3. Staffing
   Kenwood (2001) finds that there is no identifiable difference between labor cost for open source and proprietary software. Staffing costs then can be evaluated case by case. Staffing costs classifies as follow:
   a. Project Management

Based on the information given on wageweb.com in 2003, average labor costs for project manager in this case are defined as follow: manager data entry costs $39,935, manager data processing prices for $49,426, manager for computer operation $70,238) and IS manager for $120,974.

b. System Engineering / Development
These costs classifies into three costs: chief application programmer ($58,733), chief program analysis ($77,007), and software Engineer $86,374 (Wageweb 2003).

c. System Administration
Annual labor costs for a systems Administrator are about $75,000 (Kavanagh 2004).

d. Other administration
Annual labor costs for other administrative services are approximately $21,000 to $45,000 per person (Wageweb 2003).

e. Training
Red Hat Linux offers Intensive four days course of database training with instructions and hands-on lab activities that costs $2,298 and $2,398 for Web server administration training (RedHat 2007). Furthermore, Kavanagh (2004) estimates the training costs about $10,000 per week including class, travel, and expenses.

4. De-installation and Disposal
It is not difficult to de install and disposes the Linux software. However, integration costs might be involved to make new software compatible with the system (Kenwood 2001).

Indirect Costs

Indirect costs are costs that occurred in operational implementation of the software. It is difficult to get the real picture of these costs since these costs are not easy to quantify and usually treated as hidden costs within the organizations. Kenwood (2001) classifies these costs into seven items:

1. Support costs
Indirect support costs of open source and proprietary software are different. These costs are based on the specific use and environment of the software. Therefore, Kenwood (2001) suggests that indirect costs should be examined on an individual case basis.
2. Peer supports
These costs are labor expenses for technical support from service desk or Information System personnel. A case study showed that technical support handled informally by internal community in the workplace is cheaper than hire IT professionals and therefore save money for the IT department. Bryar as quoted by Kenwood (2001) define that hiring IT professionals can added up to 27 percent of the overall administrative costs.
3. Casual Learning
This includes labor expenses for end users training in formal training and support programs (Kenwood 2001).
4. Formal training
This indirect cost includes all of the course time spent by end-computing users on computer system and application training (Kenwood 2001).
5. Application Development
This cost involves labor expenses of end users performing development and customization of non-business or mission critical applications (Kenwood 2001).
6. Futz Factor
David, Schuff & Louis (2002) defines futz factor as costs that lies not in the system itself but in the time when employees spend using the system for non work related activities. Moreover, Gartner Group classifies this cost as indirect

cost. In addition, Kenwood (2001) describes futz factor as the labor expense when the end-user exploits corporate computing assets for their own personal use during productive work hours.

7. Downtime
   Kenwood (2001) defines downtime as losses in productivity of the desktop computer, servers, applications, or other tools. Down time usually happens as a result of failure in software and hardware installation and application (David et al. 2002). This activity is treated as the wages lost and can be calculated as planned and/or unplanned downtime hours times percent of productivity impact to users when downtime occurs times end users burdened salary. Windows users experience more downtime than Linux users.

As can be seen form the information above, direct costs for open source (Linux) are cheaper than Windows NT and Unix. Linux software and hardware costs are cheaper than Windows NT and Unix. Furthermore, support cost (internal and external) of Linux is lower than Windows NT. On the contrary, for the staffing cost such as project management, engineering development, system administration and training is similar between open source and proprietary software. Meanwhile, indirect costs for open source are difficult to calculate because these are hidden information and usually not published to the public. However, indirect costs are potentially significant and important to identify and consider.

CONCLUSIONS

Based on the discussion above some conclusions can be drawn as follows:

- Open source (Linux) offers cheaper price for direct cost (software, hardware, and support costs). However, it is difficult to know whether indirect cost of open source is cheaper than proprietary software. This cost may vary due to the need of organizations and the problem that happens during the acquisition and implementation of the software into the organizations information system technology.
- It is hard to calculate the cost of open source. Total Cost of Ownership (TCO) is just one alternative method to measure the price of open source. However, the TCO of open source based applications can be higher or lower than proprietary software. This depends on the specific requirement and environment of the software.
- It is important to carefully examine the cost and benefit of software decision and budget for them.

Further research concerning benefits, risks and migration costs of open source compared proprietary software need to be performed in order to get to know the real price of the software. Moreover, research regarding the method of calculating the cost of software such as using ABC (Activity Based Costing) and ROI (Return of Investment) should also be conducted for the purpose of the development of information system and technology adoption into the organizations.

REFERENCES

Berger, M. (2002). "Microsoft vs Open Source: Now it's Political " In IDG News Service.

Bitzer, J. and P.J.H. Schroder. (2006). *The Impact of Entry and Competition by Open Source Software on Innovation Activity In The Economics of Open Source Software Development*, eds. J. Bitzer and P.J.H. Schroeder: Elsevier B.V.

David, Julie Smith, David Schuff and Robert St. Louis. (2002). "Managing Your IT Total Cost of Ownership." *Communications of the ACM* 45(1).

Ebert, Christol. (2007). "Open Source Drives Innovation." *IEEE Software*(May/June):105 - 109

Fuggetta, Alfonso. (2003). "Open Source Software - an evaluation." *The Journal of Systems and Software* 66:77 - 90.

GartnerGroup. (1998). *The New GartnerGroup TCO Model - distributed computing chart of accounts*. GartnerGroup Inc.

Howels, A. (2007). *Open Source Barometer*. In Alfresco.

Hubley, M. and N. Muller. (2002). Linux: What major IT vendors are doing.

Ideas International Inc. (2005). "Microsoft Windows Server vs. RedHat Enterprise Linux - Cost of Acquisition and Support - A Comparison. viewed 16 November 2007 <*http://www.microsoft.com/windowsserver/compare/ReportsDetails.mpsx?recid=*>

Kavanagh, Paul. (2004). *Open Source Software*. Elsivier.

Kenwood, Carolyn A. (2001). *A Business Case Study of Open Source Software*. MITRE Corporation.

Koch, C. (2003). *Your Open Source PLan*. In CIO.

Lacity, M. and R. Hirschheim. (1998). *Four Stories of Information Technology Insourcing*. Houston: University of Houston College of Business Administration.

MacCormack, Alan. (2003). The True Costs of Software. *In Computerworld*.

Moyle, Kathryn. (2004). "What Place does OSS have in Australian and New Zealand Schools and Jurisdictions ICT Portfolio". South Australia.

Pearlson, K.E. and C.S. Saunders. (2006). *Managing and Using Information System: A Strategic Approach*. 3rd Edition: John Wiley & Son.

Pavlicek, Russel. (2002). The Open Source Considering TCO. viewed 15 November 2007 <*www.infoworld.com*>.

Raymond, E.S. (1999). *The Cathedral and the Bazaar*. Cambridge: O'Reilly & Associates.

Sen, Ravi. (2007). "A Strategic Analysis of Competition Between Open Source and Proprietary Software." *Journal of Management Information Systems* 24 (1):233 - 257.

Towle, Holly, Cestjon McFarland and Eric Keppler. (2004). Open Sources Issues in Business. *Journal of Internet Law* (December 2004):4 - 11.

Wageweb. (2003). "Information Management - National Averages". viewed 15 November 2007 <*www.wageweb.com*>

Wheeler, David A. (2004). "Why Open Source Software / Free Software (OSS/FS) Look at Numbers." viewed 16 November 2007<*http://www.dwheeler.com?contactme.html*>