On the Merits of the Open Source Model

Sherif El-Kassas
Department of Computer Science
The American University in Cairo

Abstract
This paper addresses the merits of the open source model from both an end-user's and developer/service provider perspectives. It addresses some of the main concerns and outlines some of the main work models. The paper concludes that the open source model has established itself as an important alternative to proprietary development models and should be given adequate attention when making software related decisions.

1 Introduction

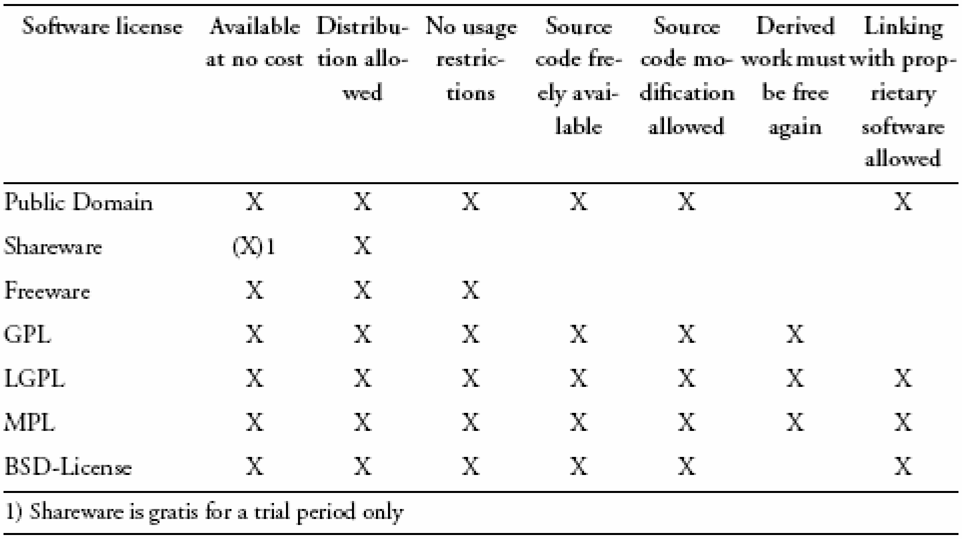
Open source software has gained a significant amount of mind share and has been the subject of much debate. Often promoted as being better than proprietary software (from an ethical and social point of view [1]), and criticized as being unrealistic or too idealistic.

According to the "Open Source Initiative" [2] software is considered "open source" if its distribution terms adhere to the following:

1. Free redistribution;
2. The distribution must include the source code, and allow distribution in source code as well as compiled form;
3. Derived Works: The license must allow modifications and derived works;
4. Integrity of The Source Code;
5. No Discrimination Against Persons or Groups;
6. No Discrimination Against Fields of Endeavor;
7. Distribution of License;
8. License Must Not Be Specific to a Product;
9. License Must Not Restrict Other Software; and
10. License Must Be Technology-Neutral.

The full and elaborated definition can be found at [2] and examples of open source licenses can be found in [3] (some are outlined in Figure 1).

This paper attempts to address the merits of the open source model from both an end-user's and developer/service firm perspectives.



Software license	Available at no cost	Distribution allowed	No usage restrictions	Source code freely available	Source code modification allowed	Derived work must be free again	Linking with proprietary software allowed
Public Domain	X	X	X	X	X		X
Shareware	(X)1	X					
Freeware	X	X	X				
GPL	X	X	X	X	X	X	
LGPL	X	X	X	X	X	X	X
MPL	X	X	X	X	X	X	X
BSD-License	X	X	X	X	X		X

1) Shareware is gratis for a trial period only

Figure 1: Different open source license models [16].

2 A users’ perspective

This Section considers an end-user’s perspective to the deployment and use of open source software; it is based on the treatment found in [4].

The main issues this section addresses are: market share, reliability, performance, security, and total cost of ownership.

2.1 Market share

Market share or whether a product “is main stream” is an important decision making criteria for users. This is because the more main stream the product is the more likely it is to find trained staff, product related resources, and reduces the risks associated with the decision to adopt the product in question.

There are many open source products that have become main stream and obtained significant market share. Perhaps among the most well known are GNU/Linux operating system and the Apache web server. Figure 2 shows the market share of various open source and proprietary web servers showing the dominance of the open source Apache web server. Figure 3 shows dominate but decaying market share of preparatory web browsers.

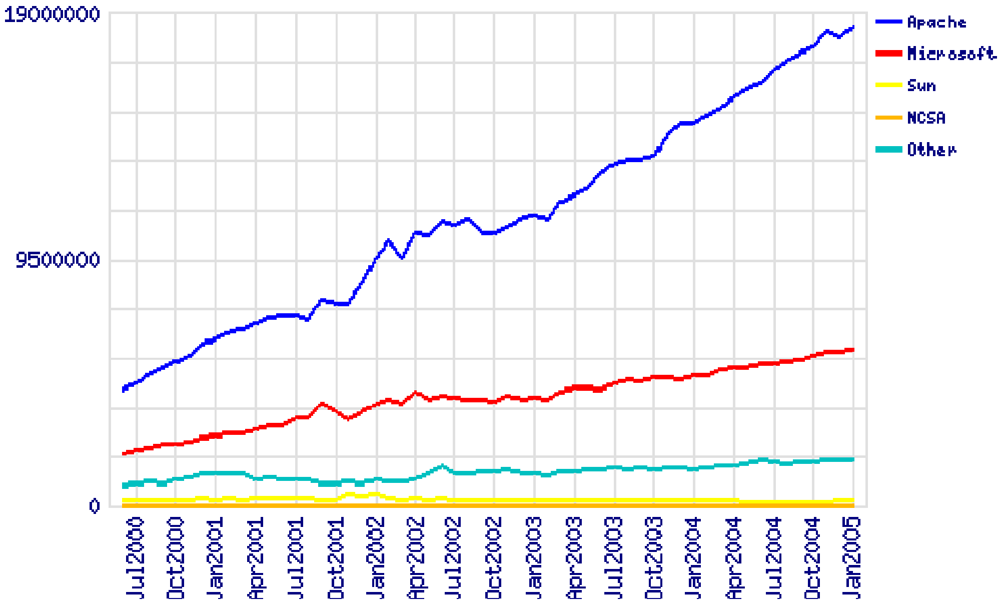


Figure 2: Totals for Active Servers across All Domains June 2000 - January 2005 [5]

One may conclude that if open source products exist and have proven their success by maintaining a healthy market share, then if would make sense to consider the adoption of such products. Of course, it is hard to determine if such products exist for all categories of interesting software. Figure 4 shows the result of a survey [10] about the use of Open Source Software.

Figure 3: Web Browser market share [4].

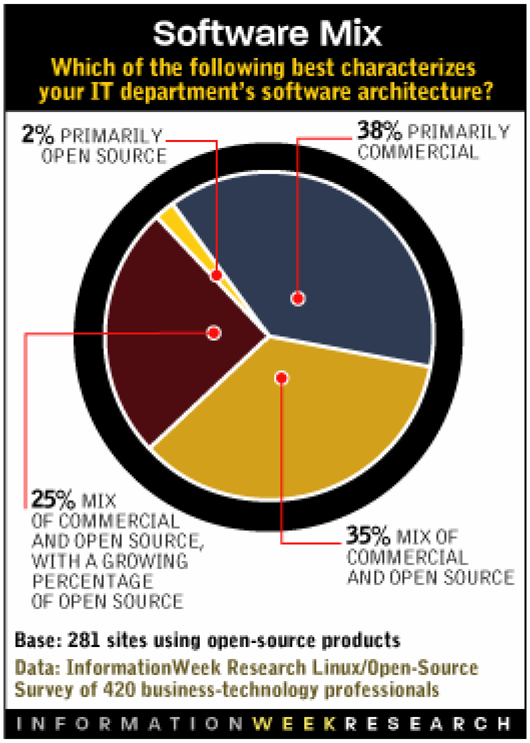


Figure 4: Open Source Software Usage (based on a survey conducted by Information Week) [12].

2.2 Reliability

Reliability is an important criterion when selecting mission critical software. There is some evidence that suggests that open source software offers better reliability. It maybe possible to attribute this to the development model of open source software that allows for rapid modifications and testing. More information about this can be found in [4,6].

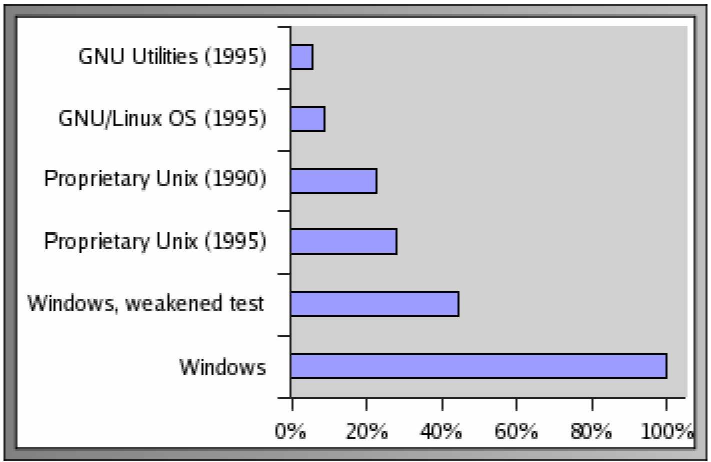


Figure 5: Failure Rates as Measured by Fuzz Tests[1] [4]

[1] Fuzz testing is a form of testing that depends on the use of random data as input to the software. See http://en.wikipedia.org/wiki/Fuzz_testing for more information.

2.3 Performance

Performance tests are often controversial due to the many factors and assumptions that affect the results. This often results in conflicting results and conclusions. However, for the purpose of this study it is safe to say that there have been successes for both open and closed source software. Examples of such studies can be found at various locations (e.g., http://www.tpc.org/ and http://www.eweek.com/article2/0,3959,293,00.asp). Hence, careful evaluation of individual products is necessary when performance considerations are important for project success.

2.4 Security

The fact that a program is open source does not, of course, make it automatically more secure. However, the same arguments made for reliability tend to apply here as well. That is, problems found in open source software tend to be fixed faster resulting in improved security. However, there are other issues that tend to favor open source systems when security is an important issue. Perhaps those are best summarized by the following statement: “Public security is always more secure than proprietary security. It's true for cryptographic algorithms, security protocols, and security source code. […] open source isn't just a business model; it's smart engineering practice” [7]. Hence, when security issues are of paramount importance is it considered prudent to use open source solutions[2].

2.5 Total Cost of Ownership

The total cost of ownership is an important measure and decision making tool for many users. It is also, however, dependant on individual and environmental needs. There have been numerous studies comparing the TCO of open vs. closed source systems with mixed and conflicting results (see [4] and [8] for example). However, it should be noted that many case studies have shown that the use of open source software has resulted in significant cost reduction (for example, Amazon.com [9] and the city of Largo [10]). Basically, every user considering open source solutions should conduct their own TCO study to apply local and environmental constrains in order to obtain accurate results. An interesting view is offered in [11] which expresses the thought that “[…] the long run total cost of operations (TCO) for a suite of proprietary software must necessarily be greater than that for an equivalent suite of free software […].”

3 A Developer and Firm perspective

One frequent concern is that without adequate protection of intellectual property there would be no real incentive for innovation. The idea being that intellectual property regulations create a form of artificial scarcity that enables creators to benefit from their intellectual product. While such arguments are accepted by many, the overwhelming presence of open source initiatives in software and elsewhere confirms that this is not the

[2] Note that for security issue the term open source is used in the broad sense of the term and, even though recommended, doesn’t necessary have to adhere to the definition provided in [1].

only motivation for creativity. Indeed, open source development is often described as an instance of the scientific method of creating knowledge.
However, the fact remains that in order to create effective knowledge based industries we need effective work and business models.

Another aspect of this phenomenon is observed in [15] where the authors' research reveals that about one third of surveyed open software developers are being paid by firms. Such firms are motivated by profit and consider open source development to be one of their tools for success. Furthermore, there have been some recent moves by firms to covert their formally proprietary software into open source software (e.g., IBM, CA, SAP, and others).

3.1 Work Models

A number of models have been applied by firms to benefit from and support the open source movement. Those are mainly identified as (this treatment is based on [17]): (1) the distributors; (2) the software producers (GPL and non-GPL models); and (3) service providers.

3.1.1 The Distributor Model

Distributors are firms that provide access to source code and open software products. Examples include, RedHat and Suse Linux. Such firms tend to make money from selling CD and customized versions of their products and by offering various update and support services to enterprise clients.

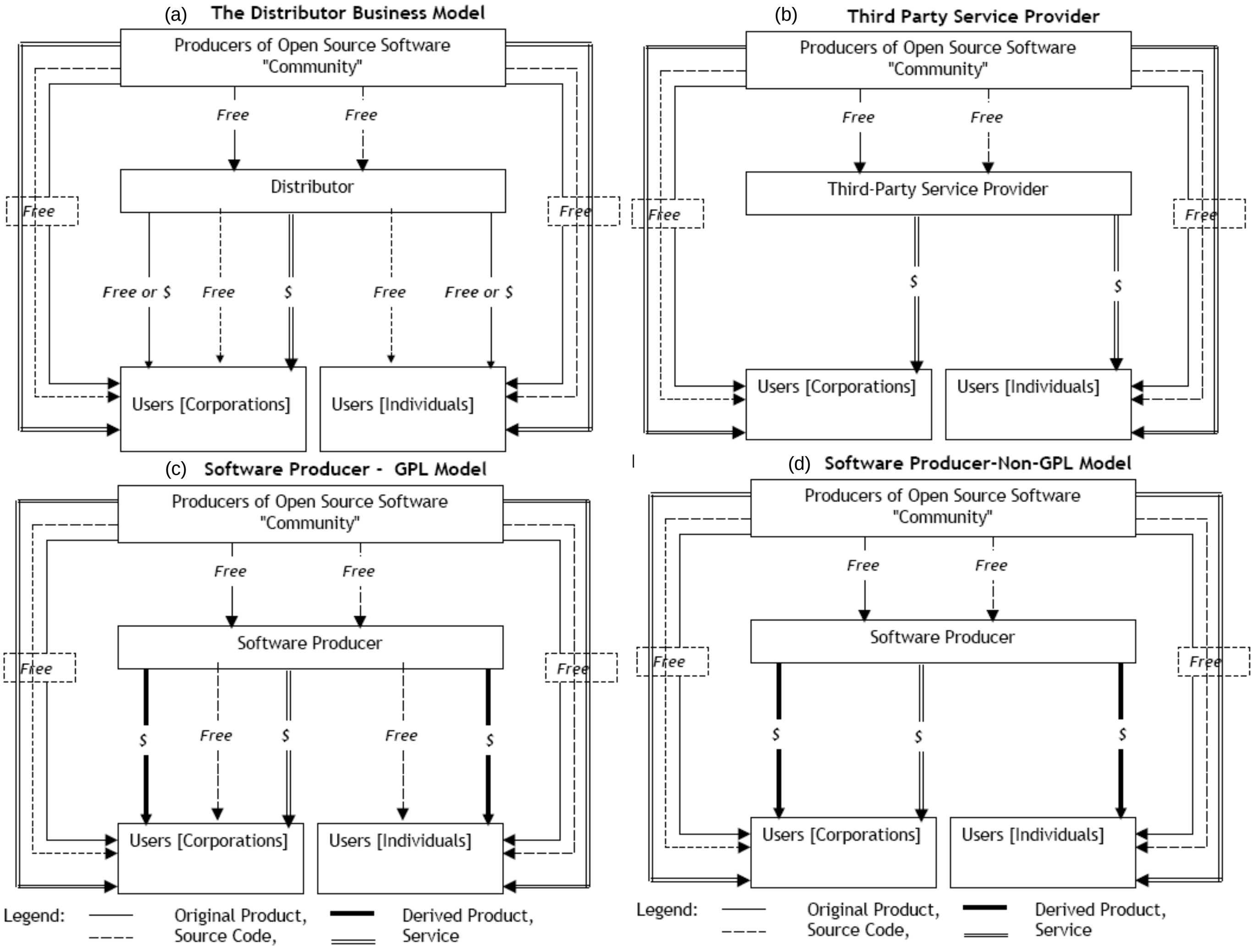


Figure 6: The Open source work models [17].

3.1.2 The Software Producer Model

Software producers that depend on non-GPL like licensing can either incorporate source code in exiting products or bundle products within packages or offerings. Both models share many similarities, but the main difference lies in that the non-GPL producer does not have to release his product for free and make the source code available. Figure 6 (c) and (d) show this basic difference.

3.1.3 Service Providers

The service provides model (depicted in Figure 6 (b)) depends solely on selling support and upgrade services. The service model is also very important as a complement for the other models discussed here. Many authors have observed (e.g. see [17]) that the sale of open source software alone may not be enough to sustain a firm; hence it is often complemented with a services model. This author believes that on the mid to long term the combination of the GPL software production with the service provider model will dominate a large part of the commodity software industry.

One should note that success in the open source services and consulting business also depends on the type of consulting being performed. It is suggested (in [16]) that open source consulting firms are expected to be more successful when product know how is important and that when more strategic consulting in required open source knowledge become less important.

3.2 Strategic motivations

Not all software firms involved in open source activities are committed to such development models. Indeed, some firm have used open source to influence their standing in other areas of the software and IT business. Some examples of such strategies (outlined in [15]) include: SAP releasing is SAP DB database product as open source enables it to reduce the over all price of its Enterprise Resource Planning application to compete more effectively with companies such as Oracle (which presumably cannot do the same since the Oracle's database product is part of its core business).

Other companies, such as IBM, may benefit from open source models to provide unified access to its diverse hardware platform which may lead to increased hardware sales.

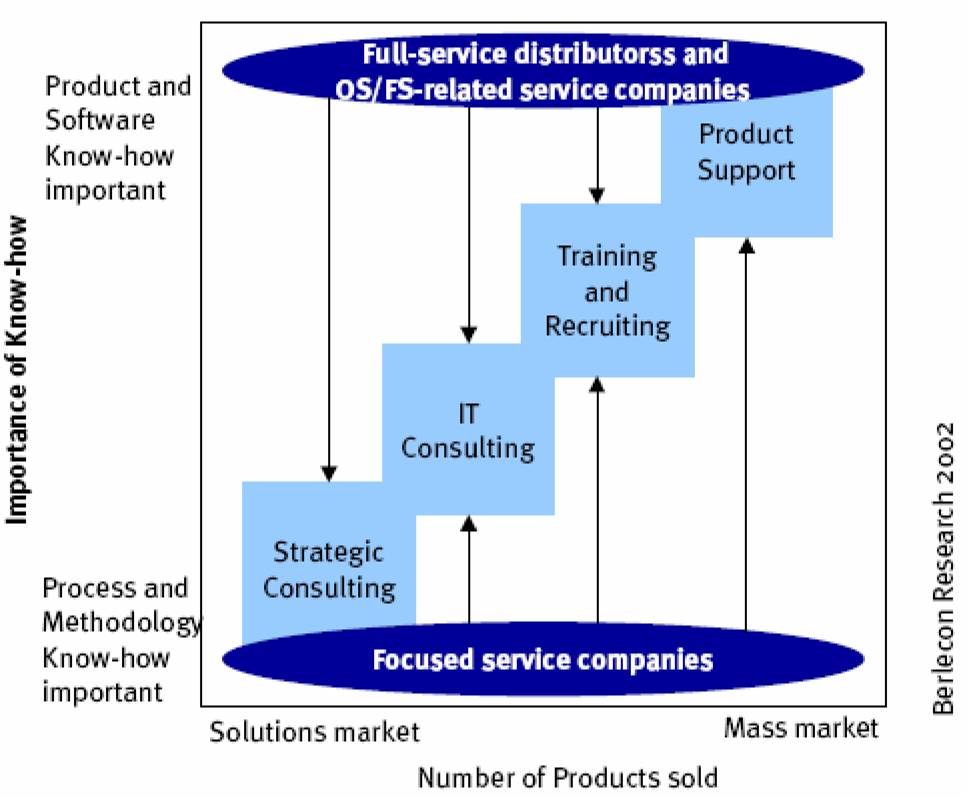


Figure 7: Areas of success in open source related services [16].

4 Conclusions

This paper addresses the merits of using and developing open source software from a number of different perspectives. It is fairly clear that the open source model has established itself as an important and successful alternative to proprietary development models.

Users and developers considering software development should consider open source platforms and models as an important alternative to proprietary software. Furthermore, it also worth noting that it may not be always possible to rely (at least entirely) on open source software. This is mainly due to reasons of availability, compatibility with existing closed systems, or due to legal requirements such as certification.

Form a developing country perspective, open source software seems to lowers the entry bar, particularly in the commodity software product area and in embedded systems; it makes sense to consider basing local development efforts on a suitable open source model.

References

1. Philosophy of the GNU Project, http://www.gnu.org/philosophy/philosophy.html

2. The Open Source Definition, http://opensource.org/docs/def_print.php

3. The Approved [open source] Licenses, http://opensource.org/licenses/

4. David A. Wheeler, “Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!”, http://www.dwheeler.com/oss_fs_why.html

5. January 2005 Web Server Survey, http://news.netcraft.com/archives/2005/01/01/january_2005_web_server_survey.html

6. Damien Challet and Yann Le Du, “Microscopic Model of Software Bug Dynamics: Closed Source versus Open Source,” http://arxiv.org/abs/cond-mat/0306511

7. Bruce Schneier, Crypto-Gram Newsletter, September 15, 1999, http://www.schneier.com/crypto-gram-9909.html#OpenSourceandSecurity

8. “7-Eleven Upgrades Its Strategic Retail System, Lowers Costs, and Improves Customer Service,” http://www.microsoft.com/costarica/hechos/casestudy.asp?CaseStudyID=14962

9. Stephen Shankland and others, “How Linux saved Amazon millions,” October 2001, http://news.com.com/2100-1001-275155.html?legacy=cnet&tag=owv

10. Lynn Haber, “City saves with Linux, thin clients,” April 2002, http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860180,00.html

11. Brendan Scott, “Why Free Software's Long Run TCO must be lower,” http://www.members.optushome.com.au/brendanscott/papers/freesoftwaretco150702.html, 15 July 2002

12. Helen D'Antoni, “Open-Source Software Use Joins the Mix,” November 1, 2004, http://www.informationweek.com/story/showArticle.jhtml?articleID=51201599&tid=5979.

13. Arjen Kamphuis, “open source as innovation model; policy implications,” http://www.creativecapital.nl/reports/pdf/Arjen_Kamphuis-session7.pdf.

14. Patrick Valduriez, “Business models for OSS,” Université Paris 6, France, http://idei.fr/doc/conf/sic/papers_2002/valduriez.pdf.

15. Dr. Thorsten Wichmann, “Firms‘ Open Source Activities: Motivations and Policy Implications,” Berlin, July 2002, http://infonomics.nl/FLOSS/report/reportPart2_firms_oss_activities_motivations_and_Policy_Implications.htm.

16. “Basics of Open Source Software Markets and Business Models,” Berlin, July 2002, http://infonomics.nl/FLOSS/report/reportPart3_basics_oss_markets_and_business_models.htm.

17. Sandeep Krishnamurthy, “An Analysis of Open Source Business Models,” February 2003, http://faculty.washington.edu/sandeep/d/bazaar.pdf.