

Running Head: OPEN SOURCE SOFTWARE: CONSIDERATIONS FOR THE
MODERN

Open source software: considerations for the modern corporation

Johnathon A. Mohr

Capella University

DRAFT ALL USUAL CAVEATS APPLY

OM8301

DR. J. Hannon

December 8th, 2006

Abstract

The range of options available for deployment to corporations has grown dramatically over the course of the past decade. This can be directly attributed to two factors, the first is the trend of firms moving from centralized mainframes to desktop units and the second is the adoption of the standardized desktop platform (Intel based PC's running Microsoft Windows). The standardized desktop platform brought computing the masses, and subsequently individuals desired alternatives to the dominant platform. As a result the open source software development process began and choice emerged. Firms are now presented with a difficult decision adopt free software or stay with standard commercial software. Using real pragmatic, empirical research this paper will demonstrate the strengths and weaknesses of both and further clarify that savings should not be the deciding factor in platform adoption.

Introduction

In the 21st century, intellectual capital is the driving force behind the American service based economy. The lifeblood of these services is the various software applications utilized by the knowledge-based worker coupled with the deployment of the modern telecommunications infrastructure. This unique coupling has allowed for the development of a true 24-7 workday and a hardwired global culture. Today we can utilize software written all around the world that works on desktop computers with more power than the super computers of yesterday, which are made of components from every place imaginable. The global supply chain, the global worker and the tools of the trade have all rapidly evolved into a series of network connected, dependent nodes allowing for a streamlined and cost efficient economy. None of this would have been possible without growing developments in the telecommunications sector and the evolution of software applications that unify the modern enterprise. Today, there is a level of interoperability that had never before existed thanks to the developments of private corporations like Microsoft which, for all intensive purposes, standardized the desktop, web browser and office productivity platforms. In the wake of global homogeneity, came increased risk, and in 2002 – 2003 (Fisher, 2003) a series of software exploits would bring networks to a halt. Two things emerged from this development. The first was a rapid evolution of the security model on Microsoft's end and the other was corporations began to look for alternatives to the Microsoft suites. What quickly returned to limelight were the UNIX's of old spearheaded by Linux, a variant that could run on regular Intel based personal computers. (Hochmuth, 2006). Arguments for the adoption of this platform ranged from the cost savings (free), to increased security, however, as with anything costs do not

necessarily decrease but transferred or shifted to another area. Further security risk is exacerbated due to the openness of the code and the dependence on the “community” to resolve emerging issues. In mission critical applications, where a new vulnerability emerges, this not only puts the system at risk, but the entire background of the business. Thus, open source, is not the panacea as claimed. In reality, it poses an increased risk to the corporation and, though free, the costs still appear elsewhere.

Open Source vs. Commercial Licenses

The open source movement is championed by two types of licenses, which for all intensive purposes; allow for free installation, distribution and modification of the product. These fall under two categories the GNU license and the GPL license.

GNU

The GNU project was launched in 1984 the goal of which was to develop a UNIX derived operating system (GNU Org, 2006). The descendants of this project, in all their incarnations, are collectively referred to as GNU/Linux systems (1). Evolutionarily speaking the operating systems are all children of the parent UNIX, System V. The predecessors, with a few exceptions of commercial distributions, are free in all forms. The GNU derived software is free to “run, copy, distribute, study, change and improve” or operates under four concepts (1).

The first, according to GNU and the GPL, is the freedom to run the program (1), for any purpose. The second is to study how the program works and, adapt it to the user’s needs, thus access to the source code is a precondition to this fact. This fact negates the concept of protected intellectual property and is the polar opposite of what we see with commercial operating systems. For example, the Microsoft Windows operating systems

are closed source and developers can only get access to the internal source code under restricted licensing and protectionist clauses. This makes sense as the goal of the firm is, of course, capitalistic in nature. Thus open source software, in the hands of a skilled user has more freedom to innovate, compared to the closed source alternative. However, the degree of freedom imparted to the end user under open source can be a double edged sword as these packages tend to cater towards the programmer or advanced user. Thus, open source, is, at present nowhere near as user friendly as its commercial competition. Next, the freedom imparted by the GNU license impacts the freedom to distribute (1) and install the software freely. This, again, is the opposite of commercial distributions which limits your licenses per seat and control this via activation procedures. Open source derivatives tend not to have any activation procedures and, again, via the licensing allow for universal deployment.

Lastly, the freedom to self innovate is another core tenant of the GNU open source movement (1). This builds off the second concept of access the source. The belief is that when the user is able to modify or improve the code, the whole community benefits. It has been argued that the open source model allows for quicker identification of bugs or flaws in the operating system and, the ultimately, a systemic effect of evolving a more secure operating system. This fact can be disputed, but open source, does allow for a greater level of customization on the user end.

GPL

The GPL or General Public License is the formal document which lays out the rights assigned, under law, to the users of GNU software. The four precepts outlined in the

prior section regarding the rights assigned to GNU provided the annotated outline of the rights imparted under the GPL. (GNU General Public License, 2006)

Commercial Products

This paper will focus primarily on Microsoft commercial products as they constitute the greatest market share of commercial distributed operating systems and software.

Currently statistics for operating system deployment are as follows according to OneStat (2006):

The 10 most popular operating systems in the world on the web are:

1.	Windows XP	86.80%
2.	Windows 2000	6.09%
3.	Windows 98	2.68%
4.	Macintosh	2.32%
5.	Windows ME	1.09%
6.	Linux	0.36%
7.	Windows NT	0.24%
8.	Macintosh Power PC	0.15%

As a result of the mass popularity and adoption of Windows Microsoft has to deploy a variety of methods that restrict the mass deployment or installation of the operating system or other Microsoft software. These methods center on the Volume License Key, Mass Activation and the Windows Genuine Advantage Program.

Volume License Key, Mass Activation Key and the Genuine Advantage Program

The volume license key and volume activations were the first generations of activation methodologies developed by the Microsoft Corporation for their various operating systems, starting with Windows 2000 (Volume Activation, 2006). Without a volume license key and end user is assigned a single activation key. This key, used during

installations, assigned the operating system a unique ID. This ID is then transmitted to Microsoft's servers, which in turn, send a unique key back to the machine in which it is activated. These keys allow for single such that if a user tries to install the OS with that same key, and once it goes to call home, they will be rejected. At this point they can contact Microsoft to purchase a new key or explain to them that the OS has been moved to a new platform. On the corporate side, organizations are issued a Volume License Key. The Volume License Key allows for mass deployment of the operating system without having to activate the product (1). However, this ultimately led to issues with piracy as keys would often be leaked. This strategy forced Microsoft to evolve their Windows Genuine Advantage Program, their Volume Activation and Multiple Activation Key's. The Volume Activation key requires the client machine to check in with a corporate server every 180 days (1). The Multiple Activation Key works like the former Volume License Key but with a limited number of activations assigned to the key (1). Lastly, the Windows Genuine Advantage program installed a piece of software on client machines that would periodically communicate information about the client to a central server (Windows Genuine Advantage, 2006). If the machine was not "genuine" or used a pirated key, users would be denied updates, directed to a site to purchase a legitimate key and would receive "nag" messages telling them their copy of the software was not valid (1). This series of procedures was designed to stem the curb of commercially pirated software, though it has already been quickly thwarted. Recently an activation server was setup in central china allowing pirated versions of vista to "authenticate" (Illegal KMS server appears on the Internet, 2006).

Strengths of Open Source Software

The core strengths of open source software revolve around the openness of the applications and the lack of license restrictions. To begin, unlike the authentication measures enacted to ensure license compliance with Microsoft Products, open source software (excluding the few commercial distributions), are free to install on an unlimited number of machines. Further, applications very similar to Microsoft's commercial suites (such as office) are available in free, open source form (Star Office – cite). If deployed successfully, the cost per client station can be decreased dramatically. Additionally, Star Office file formats are compatible with the document standards laid out by Microsoft, thus ensuring compatibility.

Secondly, the ability to modify and adapt the source is useful in customizing applications or processes to meet the needs of the firm. This also allows for custom module development and allows for the broader organizational systems to be streamlined similar to what would be seen in various ERP distributions. The cost incurred here would only be for in house development, or, in some cases the applications have already been developed under the GPL which, again, would allow the use to “snap in” modules as needed.

Cost savings and the lack of homogeneity on the desktop and servers are the driving forces behind open source software. A non-homogeneous desktop platform limits the spread of viruses/malware and the cost, being zero, for the software allows those resources that would be spent on software to be allocated elsewhere. Lastly, open source software does not have the lofty system requirements that are seen with Windows

distributions. Hence, those old Pentium 2 machines can still provide a high level of performance and, again, extend the life of the investment to the firm.

Weaknesses

Open source software is not the panacea that it would seem on the surface. To begin, the knowledge required to configure, manage and deploy this type of software is much greater than that of the competition. There is a steep learning curve and knowledge of programming is a must for the successful, secure deployment of these packages. Another core weakness is that of flaws or bugs that are inherently going to emerge in the software. In the real world there is no such thing as bug free software, this is a result of the software being coded by humans which introduces an inherent level defect into the software. On the commercial end you have dedicated support staff, a set patch cycle and clear channels of support. In a free, open source, platform, the options will center on either turning to the open source community, waiting and hoping the next distribution fixes your issue. Or there is the alternative of coding the fix by hand. This introduces an additional demand on the in house staff, requires man hours and will pull support teams away from other projects which could potentially add value to the firm.

In both cases during patching it is required that the kernel be recompiled. The kernel, in short, is the heart of the operating system. It is the piece of the operating system that provides all the services, interactions with the hardware and provides the software layer that drives the unit as a whole (Kernel Definition, 2006) Clearly, any operation that involves tampering with the kernel provides a high level of risk in terms of system stability. In recompiling the machine can be brought down critically due to an error in

compiling, additional bugs could be introduced or software/hardware can cease to function. In the business environment down time is lost revenue. When an open source distribution fails, critically, who can the business turn to? Thus, we enter a vicious circle of dependence on anonymous individuals, who volunteer their time, to fix the outstanding issue. Hence, no clear time table to resolution.

Besides the risk of server and client failure comes the inherent issues with using a non-standard platform. As compliant as the open source alternatives may be to their commercial rivals (e.g. Star Office, FireFox) issues will still emerge in terms of performance and compatibility. For instance, Star Office has dramatically longer load times as demonstrated by Zdnet's George Ou (Ou, 2005):

	Memory usage				CPU time		
	Private bytes (MB)	Private peak bytes (MB)	Working set (MB)	Working set peak (MB)	Kernel time milliseconds	User time milliseconds	Total time milliseconds
Word	4.85	5.35	10.42	10.84	62	109	171
Writer	19.20	19.20	35.64	35.64	765	1156	1921
Excel	3.71	3.71	7.61	7.61	93	47	140
Calc	18.89	18.89	35.02	35.02	484	1172	1656
PowerPoint	2.59	2.59	7.52	7.52	125	78	203
Impress	24.65	24.65	42.39	42.39	890	1422	2312
Access	2.66	2.66	7.28	7.28	171	125	296
Base	14.51	14.51	27.95	27.95	546	860	1406

For the end user perceptions are a key component of the computing experience. If the CEO is sitting there waiting what is perceived as dramatically longer load times, questions will be asked, thus it'll turn back to the software and the drive back to the de facto standard of Microsoft will begin. Additionally, though FireFox can do a lot of what Internet Explorer can do, it lacks Active X functionality, which can cause major issues on

in house web based applications. This functionality is not available in FireFox without an additional third party plug-in (Lock, 2005).

Strengths of Commercial Applications

Brand recognition and standardization are the common features of your Microsoft Windows products. The end-user, more than likely, is not going to be running a UNIX derivative on their home machine, unless it's OS X, and even then it's something the end user is familiar with. The basic familiarity with the interface provides a level of comfort and lets the user sit down and "just use" the system without additional training.

Additionally the applications which have been developed in house, will work if they were written in Active X or compiled to run as executables on the Windows. One application, for one platform, that is standardized, drives down costs and saves development time.

Especially when applications are developed with standardized application libraries that Microsoft provides to developers. Ease of program, support, standards compliance and user familiarity are at the core of what drives the commercial platform.

Weaknesses of Commercial Applications

The core weakness with commercial applications packages lies in cost. Each and every application, seat and user needs to have a licensed copy of the product. In organizations where tens of thousands of individuals work this can quickly run into the millions of dollars. Microsoft does provide volume based discounts and, unfortunately, at the time of writing this paper the exact figures were unavailable. In addition, you have increases in the cost of equipment, due to the additional "power" needed to run the latest and greatest. Additionally risk is incurred from a homogenous platform. This can manifest itself

clearly in the wake of a virus outbreak as seen in the blaster/nachi incidents of 2002-2003 (Fisher, 2003). Thus, a homogenous computing environment is inherently more at risk than a diverse computing environment. In essence, the computing environment mimics the natural environment in terms of vulnerability created by a uniform population.

Cost Metrics

All support and applications have costs incurred. To articulate the costs driven by commercial software the following metrics can help develop a baseline cost function for the organization:

Software Cost (Retail/Volume OS + Application Suites) * X Client = Collective cost across the organization

Hardware Costs (Hardware cost * Clients) = Equipment cost

Equipment Cost total + Software Cost total = organizational collective cost

Depreciation = Collective cost / 5 (assuming 5 year cycle) = true cost per year

Depreciation = Collective cost / 5 (assuming 3 year cycle) = true cost per year (realistic)

Organizational collective cost / total IT group Salary = cost per unit to support

Avg. Hourly income / down time = true cost of down time

Support availability by common platform (Microsoft Support Assistance, 2006; RedHat Global Support Services, 2006; FreeBSD Support, 2006; Apple- Support, 2006)

	Full Time Support	Regular Patching Cycle	Standardized Platform
Microsoft	x	x	X
Red Hat	x	x	
FreeBSD		x	
OS X	X	X	X*

**In some instances.*

The availability of support, regular patching and a standardized platform drives down the overall support costs across the organization.

Conclusions

Open source software and commercial applications both have their clear strengths and weaknesses depending on the nature of their deployment. In some instances going with standards based organization will drive down costs from a support standpoint. However, these savings can be quickly canceled out by the costs incurred through the additional licensing expenses. However, again, the savings gained from using open source software, to avoid the licensing aspect, can quickly be restored via the cost of retraining all the staff or the costs accrued by down time. Thus, in either extreme it is a double edged sword. It comes down to the needs of the organization. In some cases, going with a standardized commercial platform will make inherent sense, in other instances; it makes sense to deploy the open source alternative. Ultimately, in the real world, a mixture of both would yield the most desirable outcomes. It becomes clear that cost should not be the core driver behind the decision to go open source vs. commercial.

Citations

Apple - Support (2006). Retrieved December 6, 2006, from

<http://www.apple.com/support/>

Definition of Kernel (2006). Retrieved December 5, 2006, from

http://www.pcmag.com/encyclopedia_term/0,2542,t=kernel&i=45750,00.asp

Fisher, D. (2003, August 12). *Blaster Worm on the Move*. Retrieved December 5, 2006,

from <http://www.eweek.com/article2/0%2C1895%2C1654264%2C00.asp>

FreeBSD Support (2006, August 8). Retrieved December 6, 2006, from

<http://www.freebsd.org/support.html>

GNU, O. (n.d.). *The GNU Operating System - Free as in Freedom*. Retrieved December

5, 2006, from <http://www.gnu.org>

GNU General Public License. (2006, December 8). In *Wikipedia, The Free Encyclopedia*.

Retrieved 21:34, December 8, 2006, from

http://en.wikipedia.org/w/index.php?title=GNU_General_Public_License&oldid=928207

40

Hochmuth, P. (2006, June 6). *LINUXWORLD - Ultra-secure Linux evolves for the enterprise*. Retrieved December 5, 2006, from

<http://www.computerworld.com.au/index.php/id:834272493>

Illegal KMS server appears on the Internet (2006). Retrieved December 5, 2006, from

http://windowsconnected.com/blogs/joshs_blog/archive/2006/12/03/illegal-kms-server-appears-on-the-internet.aspx

Lock, A. (2005, December 15). *Mozilla ActiveX Project*. Retrieved December 5, 2006, from <http://www.iol.ie/~locka/mozilla/mozilla.htm>

Microsoft Product Assistance (2006, November 14). Retrieved December 5, 2006, from <http://support.microsoft.com/gp/assistsupport>

OneStat.com. *Microsoft's Windows dominates the OS market on the web according to OneStat.com* (2006, August 14). Retrieved December 6, 2006, from http://www.onestat.com/html/aboutus_pressbox46-operating-systems-market-share.html

Ou, G. (2005, October 25). *Performance analysis of OpenOffice and MS Office*. Retrieved December 5, 2006, from <http://www.eweek.com/article2/0%2C1895%2C1654264%2C00.asp>

Redhat Global Support Services (2006). Retrieved December 5, 2006, from <https://www.redhat.com/apps/support>

Volume Activation. (2006, November 20). *Microsoft Licensing: Volume Activation*. Retrieved December 5, 2006, from <http://www.microsoft.com/licensing/resources/vol/default.mspx>

Windows Genuine Advantage. (2006, December 6). In Wikipedia, The Free Encyclopedia. Retrieved December 6, 2006, from http://en.wikipedia.org/w/index.php?title=Windows_Genuine_Advantage&oldid=924390