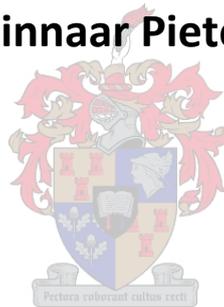


OPEN SOURCE SOFTWARE AND GOVERNMENT POLICY IN SOUTH AFRICA

Minnaar Pieters



Thesis submitted in fulfilment of the requirements for the degree of
Magister Artium (Socio-Informatics)

© Stellenbosch University 2008

Supervisor: Prof Johann Kinghorn

March 2008

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety, or in part, at any university for a degree.

Signature:

Date:

Summary

Open-source software is not something new; however, it has come into the spotlight in the last few years, mostly due to hyped initial cost savings of the Linux operating system. Consumers and businesses were made aware of shortcomings in the traditional proprietary software model and this has in turn created a surge in popularity of open-source.

The migration to open-source requires efficient research of options available and thorough analysis of the migratory process through all levels of the organization. Initial independent cost analysis has not been conclusive, with unreliable, skewed results and below average performance due to poor implementation.

The focus of this study is whether open-source software is a suitable alternative to current proprietary software packages utilized by the government sector.

The government sector offers unique challenges to which open-source software can be ideally suited. These include high initial costs of hardware and proprietary software, vendor lock-in, file format “future proofing” and the need for high security. Open-source software has addressed most of these issues admirably, but requires proper implementation.

Software packages need to be compared in order to make an educated decision. This requires in-depth analysis of features and the applicability towards government use. Once satisfactory software is found, cost factors need to be taken into account. Return on investment and total cost of ownership analysis can be calculated using existing formulas.

Despite low or no initial cost of open-source software, an analysis of initial cost does not create a reliable result. The total cost is indeed lower, but it requires a major effort to be implemented efficiently. The ideal scenario is a “hybrid” solution where a combination of open-source and proprietary software is used. This prevents a big learning curve and creates a manageable system to administer.

Any shift between software packages in an organization requires an immense effort beforehand to ensure everything goes according to plan. Migratory models can be altered depending on needs and the scope of change. If properly executed, a migration to open-source software can have immense benefits for an organization in both the short and long run.

Opsomming

Hoewel ope-inhoud sagteware nie iets nuuts is nie, het dit eers gedurende die laaste paar jaar populêr geword. Dit was hoofsaaklik die gevolg van publisiteit rondom kostebesparing van die Linux bedryfstelsel. Verbruikers en besighede is bewus gemaak van die tradisionele patentregtelike sagteware-model se tekortkominge en dit het gelei tot 'n toename in die gewildheid van ope-inhoud sagteware.

Die oorskakeling na ope-inhoud sagteware vereis doeltreffende navorsing oor die beskikbare opsies en 'n deeglike analise van die oorskakelingsproses op alle vlakke van die organisasie. Aanvanklike onafhanklike koste analise het nie onweerlegbare resultate opgelewer nie. Resultate was oor die algemeen verkeerdlik aangepas, onbetroubaar en werkverrigting was ondergemiddeld weens swak implementering

Hierdie studie probeer bepaal of ope-inhoud sagteware 'n voldoende alternatief kan wees vir die patentregtelike sagtewarepakkette soos deur die regeringsektor gebruik word.

Die regeringsektor het sy eie uitdagings waarvoor ope-inhoud sagteware ideaal kan wees. Die uitdagings sluit in die hoë aanvanklike koste van hardeware en patentregtelike sagteware asook gedwonge sagtewareinsluiting, lêerformaat "toekomsbestanding" en die behoefte aan effektiewe sekuriteit. Ope-inhoud sagteware het die meeste van hierdie aspekte effektief aangespreek, maar behoorlike implementering word vereis.

Om 'n ingeligte besluit te neem is dit nodig om sagtewarepakkette te vergelyk. Dit benodig 'n deeglike analise van eienskappe en bruikbaarheid vir die regering se doeleindes. Wanneer geskikte sagteware geselekteer word, moet kostefaktore in ag geneem word. Opbrengs op belegging en totale koste analise kan bereken word deur van formules gebruik te maak.

Ondanks ope-inhoud se lae of geen aanvanklike koste, lewer aanvanklike koste analise nie 'n betroubare resultaat nie. Hoewel die totale koste wel laer is, vereis dit 'n reuse poging om dit effektief te implementeer. Die beste oplossing is waar 'n kombinasie van ope-inhoud en patentregtelike sagteware gebruik word. Dit verhoed 'n drastiese leerproses en skep 'n meer beheerbare administratiewe sisteem. Enige verandering in sagtewarepakkette in 'n organisasie vereis 'n reuse aanvanklike poging ten einde te verseker dat alles volgens plan verloop. Oorgangsmodelle kan verander word afhangende van behoeftes en die omvang van

verandering. Indien deeglik uitgevoer, kan 'n verandering na ope-inhoud sagteware enorme voordele inhou vir 'n organisasie oor beide die kort- en die langtermyn.

Table of Contents

Introduction	1
Chapter 1	
What is Open Source and how is it different?	7
1.1 The definition of open source	9
1.2 The history of open source	11
1.3 What constitutes open source software?	13
1.4 Different variations of open source licenses	17
1.5 The Cathedral and the Bazaar	19
1.6 Shortcomings in the peer review model	20
1.7 Conclusion	21
Chapter 2	
Types of Open Source Software	22
2.1 Operating systems	22
2.1.1 Distributions of Linux	24
2.1.2 Differences between distributions	25
2.1.3 Linux desktop environments	30
2.1.4 Comparison between Linux, MS Windows and OSX	35
2.2 Open source productivity software	49
2.2.1 Comparison between MS Office and OpenOffice.org	56
2.2.2 E-Mail and messaging software	59
2.3 Web browsers	63
2.4 Antivirus	65
2.5 Choosing between packages: the open source maturity model	67
2.6 The advantages and disadvantages of using open source software	70
2.7 Conclusion	75

Chapter 3	
Total Cost of Ownership and Return on Investment	76
3.1 Forms of ownership	77
3.2 What are the costs aspects involved?	79
3.3 Total cost of ownership comparison	85
3.4 Return on Investment Analysis	90
3.5 Conclusion	95
Chapter 4	
Open Source in the Government Sector	96
4.1 The feasibility of a government open source initiative	97
4.2 International efforts to adapt open source software	104
4.3 Creating a government open source strategy	111
4.4 Typical government strategic positions and the South African revised open source policy	112
Chapter 5	
The OSS Migration Process: Challenges and Considerations	117
5.1 Phase 1: Assessment / Analysis	119
5.2 Phase 2: Management level commitment and strategy	122
5.3 Phase 3: Technical pilot testing and training	125
5.4 Phase 4: Organization wide implementation	129
5.5 Phase 5: Support phase	132
5.6 Phase 6: Re-assessment phase	134
Conclusions and Future Research	134
Bibliography	140
Appendices:	149
Appendix A: Upgrade model based on current operating system and hardware platform	149
Appendix B: The Open Source Life Cycle	150
Appendix C: Interviewees for insight into South African FOSS strategy	151

List of Tables

Table 2.1:	Comparison between Linux, Windows and Mac OSX	37
Table 2.2:	Prices of different Microsoft Office editions	50
Table 2.3:	Comparison between Microsoft Office and OpenOffice	56
Table 3.1:	Examples of companies that have adapted open source due to lower TCO	88
Table 5.1:	Open Source and Proprietary alternatives	121
Table 5.2:	Functionality layers of the typical desktop user	127

List of Figures

<i>Fig 1.1:</i>	<i>The adoption of open source over the last decade</i>	8
<i>Fig 1.2:</i>	<i>Key differences between open source and other software</i>	9
<i>Fig 1.3:</i>	<i>Open Source Community Initiation and Growth</i>	10
<i>Fig 2.1:</i>	<i>A Screenshot of the GNOME user interface</i>	32
<i>Fig 2.2:</i>	<i>A screenshot of the KDE desktop</i>	33
<i>Fig 2.3:</i>	<i>OpenOffice.org Writer and Microsoft Word Comparison</i>	52
<i>Fig 2.4:</i>	<i>OpenOffice platform and document compatibility</i>	53
<i>Fig 2.5:</i>	<i>OpenOffice.org application on Windows XP</i>	54
<i>Fig 2.6:</i>	<i>Mozilla Firefox as a percentage of total active browsers</i>	68
<i>Fig 3.1:</i>	<i>Microsoft “per-seat” CAL licence</i>	77
<i>Fig 3.2:</i>	<i>Microsoft “per-server” CAL licence</i>	78
<i>Fig 3.3:</i>	<i>Breakdown of TCO for IT infrastructure workloads over a five-year period.</i>	80
<i>Fig 4:</i>	<i>Geographical repartition of Linux developers</i>	105
<i>Fig 5.1:</i>	<i>The open source migration process</i>	119
<i>Fig 5.2:</i>	<i>Cost vs. reliability analysis of various server operating systems</i>	121
<i>Fig 5.3:</i>	<i>Main obstacles in migrating to open source in organizations</i>	123
<i>Fig 5.4:</i>	<i>Availability of open source solutions is increasing but systems integrators still have limited serious offerings and skills, creating a skills requirement “gap”</i>	128

List of Abbreviations, Acronyms and Terminology

Apache:	An open source web server application, currently the most widely deployed web server application, used to display web pages to end-users
Back Door:	An intentional method of gaining root level access to a piece of software. This is usually implemented by developers in order to gain access in case of software problems. However, these backdoors are easily exploited by malicious software.
Copyleft:	Is the licensing agreement where “copylefted” code can be used in other software for free, but the final product needs to be free as well.
FOSS:	Free / Open Source Software
Free Software:	Software which is contractually obligated not to be implemented into proprietary software.
FTP:	File Transfer Protocol
GIMP:	The GNU Image Manipulation Program or just GIMP is a free software raster image editor.
GITOL:	Government IT Officer's Council, who is in charge of implementing open source policy in SA.
GNU:	GNU is an entire operating system made up from free software. Its name, a recursive acronym for GNU's Not Unix, is specific in the sense that the design is UNIX like, but is devised from free software.
GPL:	General Public Licence
GUI:	Short for graphical user interface (often pronounced "gooey"), is a particular case of user interface for interacting with a computer which employs graphical images to represent the information and actions available to the user.

- Hacker:** A person who modifies or alters computer applications, although the term “hacker” also conjures up negative implementations. There is actually a differentiation between “hacker” and “cracker”, with a cracker being more malicious in nature.
- HTML:** Hypertext Mark-up Language, a programming language used by simple webpages which implement tags to alter text and hypertext.
- Hypertext:** Hypertext is static or dynamic text in a page which links to other documents or pages. The World Wide Web is a good example of the implementation of hypertext.
- IBM:** Acronym for the company International Business Machines
- IT:** Information Technology
- Leopard:** The codename 2007 release of Apple’s proprietary operating system. Apple gives feline names as concept names for its releases of Mac OS X. For example 10.3 is Panther, 10.4 is Tiger and the upcoming 10.5 is Leopard. Currently expected in Q4 2007.
- Lock-in:** When software needs to be replaced with a different package, this can create financial / functional problems, resulting in being forced to stick with the current software vendor due to prohibitively expensive migration costs.
- Migration:** In this research context, the process of moving from one software package to another. For example, the shift from Microsoft Windows to Linux. This term is however not specific to a timeframe, and not necessarily a more timely procedure than a “switch”.
- OEM:** Original Equipment Manufacturer. “OEM” release software is normally discounted when purchased with hardware. E.g. Microsoft Windows can be bought in OEM form with a new PC.
- Open Source Software:** Software whose source code is available under a copyright license that permits users to study, change, and improve the software, and to

redistribute it in modified form. This can be more complex, depending on the specific contract used.

- Open Standard:** Standards which are made available to the public, ensuring compatibility between vendors. This is in contrast of proprietary standards which is normally supported and maintained by a specific group or company.
- OpenDocument:** OASIS Open Document Format for Office Applications. Also known as ODF. An agreed upon open standard for documents, spreadsheets and presentations. It is freely available and can be implemented by anyone, without royalty fees.
- OS:** Short for operating system. An operating system manages hardware and acts as a layer between it and the applications.
- OSI:** Open Source Initiative, an organization dedicated to promoting open source software.
- OSS:** Open Source Software
- OS X:** Apple's proprietary operating system, also known as Mac OS X. The X is in reference to the roman numeral X which means this is the tenth major revision of the operating system.
- OSI:** Open Source Initiative
- Patch:** A small update made available normally to fix bugs in existing software packages. On rare occasion it can improve functionality.
- PIM:** Personal Information Management. Examples include Microsoft Outlook. Typically includes email, calendar and to do lists.
- Productivity Software:** Includes word processor, spreadsheet, email, presentation and (in some cases) database creation software. Normally available as n suite, examples include Microsoft Office and OpenOffice. Also known simply as "office" software.

Productization:	A term coined by Woods and Guliani referring to the user interface and general ease of use of a piece of software.
Proprietary:	A proprietor exercises private ownership, control or use over an item of property, usually to the exclusion of other parties.
RAM:	Random Access Memory, a temporary high speed storage memory used in computers.
RISC:	Reduced Instruction Set Computer. These processors do away with unnecessary commands that normal processors require. Theoretically, because of more streamlined commands, precious computing cycles can be applied to actual necessary commands.
ROI:	Return on Investment
Software Licence:	A document that stipulates what a user is allowed to do with software.
SUSE:	Novell's Linux operating system distribution
TCO:	Total Cost of Ownership, a method of calculating cost aspects, for example during a migratory process. Is especially important in planning phases.
Tiger:	Apple's current codename of its proprietary operating system, OS X in 2006. Apple gives cat names as concept names for its releases of Mac OS X. See "Leopard".
Ubuntu:	A distribution of Linux, operated by Canonical, operated by South African Mark Shuttleworth.
UNIX:	A proprietary operating system developed during the 1960's to the 1970's. Forms the basis of the Linux operating system and Apple Mac OS X.
Viral marketing:	Marketing using predefined social networks to create awareness of a product or service. This is especially popular in modern Web 2.0 websites.

- Vista: Microsoft's latest consumer Windows release, which was released early 2007. This is the follow up to Microsoft Windows XP which was released in 2001.
- VNC: Virtual Network Computing, an open standard used to control another computer remotely. The application transmits keyboard and mouse commands, while transmitting the graphical interface back to the controlling (client) user. It is available for a variety of operating systems.
- W3C: The main international standards body for the World Wide Web known as the World Wide Web Consortium or "W3C".
- x86: Processor architecture developed by Intel in the 1980's. Modern processors such as the Pentium or AMD Athlon are basically evolutionary upgrades of the same architecture. This is the standard that Windows, Linux and more recently, Mac OS X run on.
- XML: Extensible Markup Language. Extensible because it allows users to attribute their own tags. It is a fee-free open standard recommended by the W3C. Normally used as a system interface between different information systems.
- XP: Short for Microsoft Windows XP, which was released in 2001. It is the longest running release of Windows so far, with it only being replaced in 2007, 6 years after its original release.

Acknowledgements

I would like to sincerely thank the following individuals for their invaluable contribution:

Prof Johann Kinghorn for his support, guidance and for sharing his wealth of knowledge and insight with me.

My colleagues for all their practical advice, assistance and support.

My parents and brothers for their love, support and practical assistance throughout the long two years. You have provided me with every opportunity to reach my full potential.

Miss Noeline Bezuidenhout for being patient with my endless language quirks.

Maria Farelo (Director; ICT & e-Government Strategy, Policy & regulations, DPSA) for helping me get on the “inside track” with OSS developments in South Africa.

Lion Steynberg (Director: SITA Oversight Office of the Government Chief Information Officer Department: Public Service and Administration) for valuable feedback and connecting me with the right people

Introduction

Open source software is one of the most appealing phenomena in the software industry. Its notion is old, but the impact of it is only starting to be noticed more recently.

Open source software's much hyped zero licensing cost is unique, and holds many special opportunities for both developed and developing nations. On the other hand there are a number of constraints that might come to surface that require thorough research. These might include aspects such as total cost of ownership, return on investment and compatibility issues.

Open source in its initial conceptualization is a very genial proposal, yet it requires a certain skill level to fully implement in an organization. This can include difficult setup, a lengthy migration process and broad learning curve with certain applications. Spending the necessary time and resources, open source implementation can create great advantages to key segments of the economy. These sectors can include education, software development and hosting and government operations. The government sector stands to gain a lot from the peculiarities of open source software, especially in terms of operating systems or productivity software. One advantage is an already strong foothold and support of open source software in South Africa. South Africa has very specific reasons for adapting open source software (also known as OSS) and has played a key role in the development of OSS. Work by groups like The Shuttleworth Foundation and Go Open source has created acknowledgment for SA in the software community. OSS has been doing well in environments where lower cost is a key factor (like the education sector¹). Hopefully the government sector can find benefit from its application.

Cost benefits should not be the primary focus, seeing as low licensing costs do not paint the full picture. There is a tremendous effort involved in a migratory process, something that

¹ It should be mentioned that Microsoft makes its software available in special lower cost versions for schools. Despite this, open source is in many cases still cheaper to operate.

requires support from management in any organization. In South Africa, many employees are already aware of where cost savings of software can be better applied.

IT and internet penetration in South Africa are highly underdeveloped which has a severe effect on the growth of the economy, due to the lack of communication affordability. The reasons are the high cost of broadband infrastructure and a lack of proper regulation from ICASA. It is for this very reason that open source is being looked at as a true alternative. If chosen, money could instead be used for broadband connections and additional hardware. Using Linux as an operating system and an open source office suite will greatly decrease costs of entry level hardware – sometimes savings of as much as 50 percent.¹ With enough research beforehand, great costs can be saved which instead could be better put to use on developing other aspects of the IT infrastructure. Increasing infrastructure and lowering broadband costs can lead to a higher internet saturation rate, something which is still very low in the country.

It is important to remember that the IT industry shifts its focus very quickly, as is evident by the speed of software updates being released. The aim of this study is to view options available to the consumer in 2006 and 2007. There is always a great deal of speculation and bias involved with the development and reviewing of new software releases. The IT industry is broken up into groups which all fiercely back their own hardware/software solution.

First are the Windows users. They are not really “fanatical” about it, and instead choose Windows because it is the operating system which benefits from the most networkability². In the next camp we have the “Mac” users. They use Apple hardware and run Mac OS X, another proprietary operating system. Windows and Mac OS X are very different and focus on different aspects of the computing experience. Then finally we have Linux. Linux is an

¹ Probability Services. 2007. *Why use Open Source?* [online] Available:

<http://www.probabilityservices.co.uk/index.php?/content/view/9/29/> [10 October 2007]

² In this sense, networkability is not meant computer hardware networks. Windows users constitute about 90% of all computer users, and therefore there is the added benefit of more users being able to use and share a specific application (and its data and documents) without compatibility problems.

open source operating system which is continually in development by large groups of contributors. Unlike Windows and Mac OS X, Linux is marketed and supported by numerous different companies, although they all run the same kernel. To better understand this, one has to look at the way open source software is developed and marketed. All of these operating systems are being extensively updated at the end of 2006¹ and early 2007 – something which can complicate the research process and comparisons, requiring research into feature sets of developers and release candidate versions, instead of final products.

These software/hardware “camps” are primarily focussed on the operating system the computer uses. Even though the OS provides the interface and could be considered the first barrier to ease of use, other types of software should also be researched for its usefulness. This can include the alternative use of open source software for productivity software, antivirus software or even simply running an alternative browser which might give the end user a better operating experience. In fact, there exists an open source alternative for almost any application or software use.

This includes the domain of servers, an area where open source has already proven its mettle. Open source software runs the majority of the world’s web servers, and Linux has proven to be beneficial for the required uptime of some the most renowned e-commerce stores including Amazon.com and was central to the success of the Google search engine, and still is to this very day.

In this study we will focus on the types of open source software, be it operating systems or productivity software. It is important to analyse all of these solutions in order to make an educated decision regarding hybrid solutions, which can potentially save costs without expensive retraining. Despite its lower licensing cost, the real cost lies in the total cost of such a migratory effort. This can include retraining, specialist staff and general support for

¹ At the end of 2006 Microsoft released Windows Vista, the successor to Windows XP. Apple will also release Mac OS X 10.5 (Leopard) which is the successor to OS X 10.4 (Tiger) in the third term of 2007.

the new software. This requires a lot of upfront research which takes current technical systems into account (for example current operating systems and employee skill level).

The private sector has unique software requirements which in many cases are very reliant on the support and backing of large software corporations. Open source software can initially require a very high skill set from an administrator, something that not many companies can afford even with the use of outsourcing. Small companies who do not have a high skill level in employees will not necessarily benefit from a total open source solution.

Software can be analysed from a purely technical standpoint, but needs to be assessed for repercussions of choosing specific solutions or formats. There is a lot of concern over the “lock-in” that is caused by proprietary software and how it will affect sensitive information like the documents used by government administration. Formats like OpenDocument (ODF) plan to alleviate these types of problems but it still requires industry support. Many governments across the world have chosen to make these open standard formats mandatory.

The government sectors have tremendous amounts of information that needs to be well managed. For this reason, the necessary skills of creating proper systems need to be made available through proper IT administration staff. This information is of a sensitive nature and requires the “separation” of software and data. If data is captured, the person requiring the data should be free to use any software they choose to use the data in. For example, capturing a spreadsheet of figures should be readable not only in Microsoft Office, a very expensive productivity suite. Elections, polls and census reports also place tremendous strain on the IT systems and reliability and security is of the highest concern.

These systems have to function with the utmost reliability and run efficiently. With recent pressures to make the voting process more electronic, investigation needs to be made to see whether proprietary software has the track record to be able to handle such sensitive data. With corruption prevention as being one of the top reasons for making the voting process electronic, it would be a big problem if the system could be hacked and altered.

There are a few key reasons why open source software is especially beneficial to the government sector. First and foremost, cost. Implementing open source software greatly decreases initial licensing costs compared to proprietary solutions. Security is also a great concern with government software use (voting, census reports etc.) and open source operating systems have been proven to be more secure. The potential “lock-in” of using proprietary

formats can make government applications reliant on expensive software licences. Upgrading costs are also lower than using Microsoft's CAL (customer access licence) options. Of course, there are many negatives that can be pointed out, but that is not the point of this study. Instead, a thorough analysis should be made of where open source makes sense for official use and what the best method is to implement it. Factors such as return on investment and total cost of ownership can give rough figures but should only serve as guide. The actual return on the investment will be given over time in the form of better competitiveness (and bargaining power¹) at upgrade time and better performing hardware for a longer time, one of the most touted features of open source software.

The primary research question is whether open source software is a true alternative and suited for the government office use. Secondary questions are whether these open source strategies include desktop or server operating systems or both. Productivity software also has to be differentiated from operating systems. Cost is an important aspect that needs thorough analysis. This study will try to recognize the unique advantages that open source can bring to South Africa and its government sector, and the best way of attaining its potential, while also taking into account disadvantages.

Despite open source being available to any sector, this study will focus primarily on the government sector. There are some references made to the effects of software migration on the education sector, but this is mostly in reference to possible economic advantages in the future due to higher computer literacy. The study includes a variety of software types, not only operating systems. Open source software alternatives are available in almost any software type and needs to be compared with proprietary releases in order to make informed decisions. The over-enthusiastic viewpoint of an entirely open source computer system is a naïve stance which does not necessarily carry enough benefits. Cost should not be the primary factor. Instead it should be taken into account that competition in the industry is fierce and only through thorough comparison can solutions be chosen. The nature of the field is very dynamic and updates and changes occur almost daily, therefore this study will focus

¹ If proprietary software suppliers know that the client is open to the adoption of alternative open source solutions, they will in all probability try to create a better deal for the client.

on what the current situation is in 2007. Future predictions based on official product announcements are also taken into account.

Open source is filled with a variety of stereotypes and is highly susceptible to prejudicial views. Luckily this is changing in recent times as organizations are starting to analyse open source software more thoroughly.¹ It is imperative to keep in mind that this study is not meant to be promotional of open source in any way, but instead an objective analysis of options. It is because of these very stereotypes that further research is required that looks beyond sometimes blind fanaticism for the unconventional or different. At this moment, the biggest supporters of open source software are primarily people with above average computer skills and therefore people do not necessarily trust their judgement of what is the “best” software seeing as it might not be true for the everyday computer user. Software of all types need to be compared with a clear dedication to the everyday computer user, not the people who sit in the IT department. The human aspect needs to be emphasized seeing as a difficult to use application can severely hamper an entire migration process of any organization.

In a quantitative sense, only cost and features can be analysed and compared. There exists a lot of research in this area. The problems with most of those studies are that they are openly sponsored by large software companies. It does not garner a lot of trust, for example, a Microsoft sponsored study that studies cost comparisons between Windows and Linux. Instead, attention should be given to case studies that give an in depth analysis of projects which have occurred, and what kind of challenges were found. This proves very valuable for future migration attempts, as cost analysis is not the primary concern.

Aspects in this study are there only for the purpose of giving the reader examples of how existing models and frameworks can be applied to the research involved with evaluation of open source software. Of course requirements differ depending on the end-user, and therefore the application and feature set comparison should only serve as a general guide based on the average computer user’s viewpoint and requirements.

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O’Reilly. pg. 3

Chapter 1

What is Open Source and how is it Different?

In order to better understand Open source software, it is necessary to first look at the history of open software, and why it has recently gotten more interest as a supposed true alternative to proprietary software. The basic idea of open source software came into development in the seventies with the advent of the first mainframe computers that ran under proprietary software. Using the skills of many different developers all with the aim of improving an existing application, it has recently presented software of very high quality. The very notion that software can be free is something that might detract end-users making them believe that the product is of a lower quality. While this might be true of smaller projects, but many larger companies “do develop open source software on an economically feasible business model”.¹

¹ Nicholaisen, Thomas Ferris. 2006. The Use of Open Source and Open Standards in Web Content Management. *IKE '06 Proceedings*. pg. 250

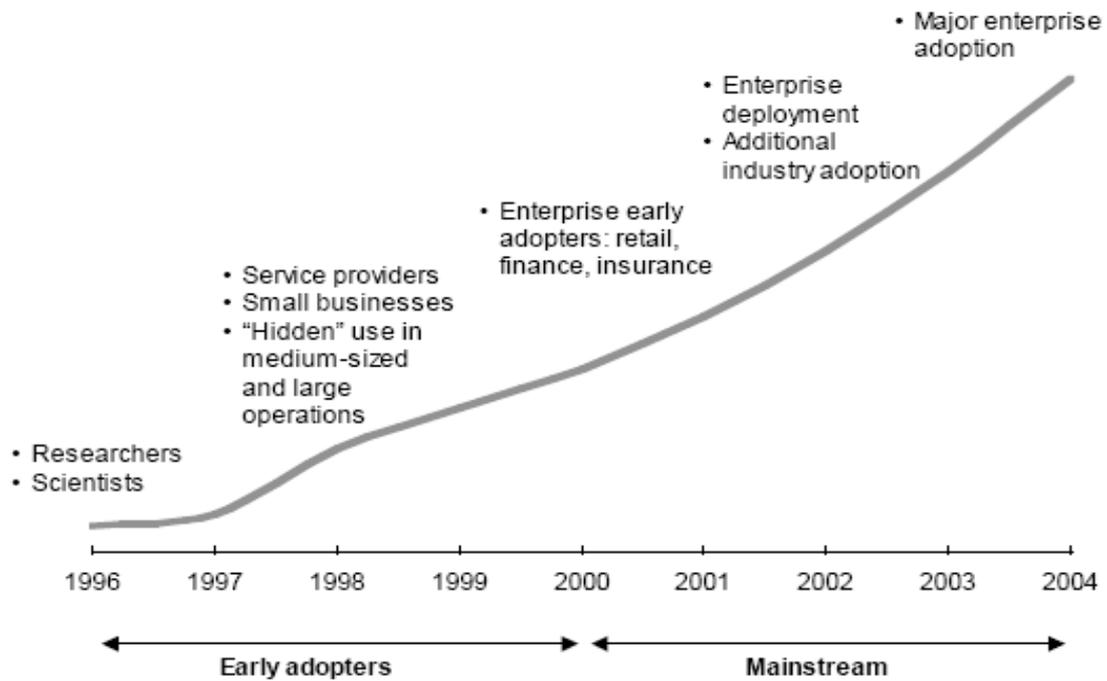


Fig 1.1: The adoption of open source over the last decade¹

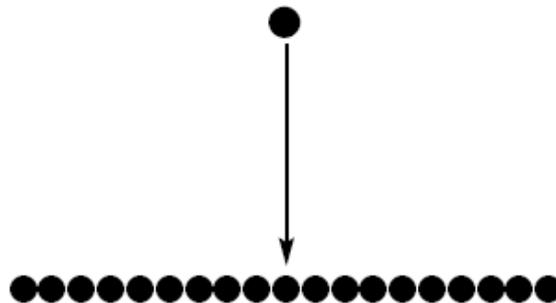
It has only recently come into the spotlight with many touting Linux as a true alternative to Windows. While OSS has always been part of the server software scene, it has most recently shifted into a “mainstream” segment of the user population, as can be seen in figure 1.1. This is due to affordability and an increase in the ease of use of OSS alternatives. Most people think “free” when they hear the words open source software. However, under closer scrutiny it is important to take other aspects into account. Yes, open source software runs in most cases under a zero cost licensing fee, but one must always consider costs of implementing such software. Open source software and Linux is not the same thing. Linux is an open source operating system that is currently receiving much attention and touted by many as a worthy alternative to Microsoft Windows.

¹ Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* pg. 36

1.1 The definition of open source

Open source software is any application which has source code that is available to anyone to use, alter and improve. The very nature of open source has created a culture where programmers look differently at the development of software. Because the code is always available, programmers are free to adapt and change software in any way they see fit without breaking licensing agreements. Because of its modifiable nature, and the fact that so many people contribute to a piece of software, it is believed that software errors and bugs are normally eliminated quickly. ¹

Commercial Software Model: Strict separation exists between producer and user.



Open Source Development Model: Author-user interaction, user-user interaction, larger community

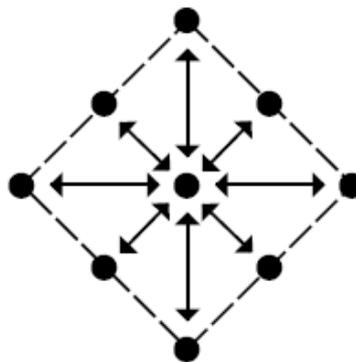


Fig 1.2: Key differences between open source and other software²

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 37

² Krishnamurthy, S. 2003. A Managerial Overview of Open Source Software. *Business Horizons*. pg. 10. This diagram shows differences in the development groups with proprietary and open source software.

The fact that such a large number of developers (as can be seen from figure 1.2) form a support base also contributes to faster development of software, bearing in mind the development time and costs of a standard piece of commercial software. Open source advocates believe this development process delivers higher quality software compared to a normally “closed” development process, where an end user merely runs compiled code, as in the case with proprietary software.¹

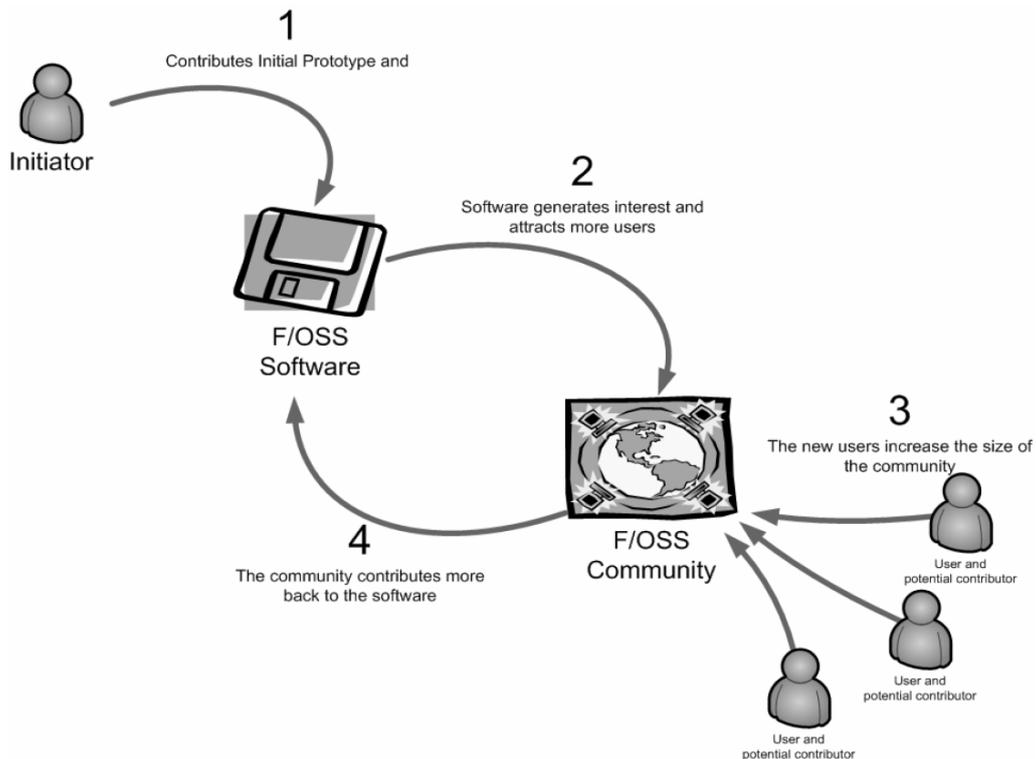


Fig 1.3: Open Source Community Initiation and Growth²

¹ For more info on this statement, see “Linus’s Law” at the end of this chapter. Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O’Reilly. pg. 37

² AlMarzouq; Zheng; Rong; Grover. Open Source: Concepts, Benefits and Challenges. 2007. *Communications of the Association for Information Systems* (Volume 16, 2005) pg. 756-784.

Figure 1.3 shows the typical development process of an OSS application. This can be applied to most OSS solutions, including Linux.

1.2 The history of open source

Open Source software originated in the “hacker” culture of the late 60’s and early 70’s¹. Four big university computer science labs in the US² all had a system of passing programming code back and forth between developers and users. When software was developed and improved over time, the codes were re-released and open for inspection to the entire community³. When someone did make improvements, it was considered compulsory to give this code back to the community.

Richard Stallman was a student at MIT in 1971 and was developing an operating system for a DEC PDP-10 machine. However, once the production of the machine was discontinued, its entire software base had to be rewritten in order to be compatible with new machines. Members of Stallman’s previous developer community who contributed to this machine code were, however, contracted by proprietary software companies. Under licence these developers could not contribute to the open source code anymore. Stallman then decided to create his own operating system which was free.

During that time, UNIX was the most popular operating system. GNU⁴ was created, which was compatible with UNIX on an application software level. The term “open source” was only coined later in the development of non-proprietary software. In 1984, the Free Software Movement was started by Richard Stallman, for which he created the GNU programme. The basis of his idea came from the belief that proprietary “closed” software is immoral, and he wanted to create a system which replaces software which has “restrictive licensing terms”. He sought to create a society where people are free to help each other in modifying computer applications, so that it was more suited towards their needs.

¹ Rasch, Christopher. 2006. *A Brief History of Free/Open Source Software Movement* [online]
Available: <http://www.openknowledge.org/writing/open-source/scb/brief-open-source-history.html>
[10 October 2007]

² Stanford, Berkeley, Carnegie Mellon, and MIT

³ At that time, the community consisted mostly of contributors from four universities in the US.

⁴ GNU stands for “GNU is not UNIX”.

In the process, Stallman created a massive amount of software that was made available to users and developers of the GNU system. In order to prevent users from taking the code and using it in proprietary software (something that happened before), Stallman created the GPL (GNU General Public Licence). The GPL basically stated that any changes and improvements made to the code had to be re-released under the original licensor's name, thereby preventing outsourcing of the code to other proprietary uses.¹

In 1991 the Linux kernel was born. Linus Torvalds, a student at the University of Helsinki, developed the Linux kernel. He sent this to various open source groups who were truly astounded by the competence in the kernel code. Instead of “cashing” in, Linus made the source code available to the open source community. In doing this, Linux became the most successful open source operating system.

Part of its success can be attributed to the quick growth of the internet in the early 90's. This increased interest in open source software, due to the quick dissemination of digital data that the internet provides. Despite free access, open source software did not prove very successful due to the inherent lack of confidence in the “free” name. In 1998, two camps were formed – open source and the free software movement.²

However, not all supporters of the Free Software Movement believe that proprietary software is wrong. They support the pragmatic idea behind free software, not the technical aspects.

¹ Free Software Foundation. 2007. *GNU General Public License*. [online] Available: <http://www.gnu.org/licenses/gpl.html> [04 June 2007]

² Day, B. & Pogue, T. 2004. Free / Libre & Open Source Software and Open Standards In South Africa: A Critical Issue for Addressing the Digital Divide, South African National Advisory Council on Innovation - *Open Software Working Group*, [online] Available: <http://www.naci.org.za/floss> [15 September 2006]

The reason for developing software for an open source project can be intriguing, and often purely academic in nature. Marc Andreessen, the person who invented the first popular graphical World Wide Web browser¹, mentions the following:

*“Open Source is nothing more than peer-reviewed science. Sometimes people contribute to these things because they make science, and discover things, and the reward is reputation. Sometimes you can build a business out of it, sometimes they just want to increase the store of knowledge in the world. And the peer review part is critical – and open source is peer review. Every bug or security hole or deviation from standards is reviewed.”*²

1.3 What constitutes open source software?

According to the Open source Initiative, for software to be labelled “Open source”, it has to comply with a few requirements. At the end of each quote is a rationale why this rule is necessary. To be considered OSS by OSI standards, the software has to comply with these requirements; the quotes are legally binding, and therefore cannot be shortened or altered.

a) Free distribution

“The licence 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 licence shall not require a royalty or other fee for such sale.”³

¹ This web browser was called “Mosaic” and its technologies still form the basis of many modern day browsers. Andreessen was later one of the co-founders of the Netscape, the first commercial browser.

² Friedman, Thomas. 2005. *The World is Flat: A Brief History of the Globalized World in the 21st Century*. Penguin Books. pg. 83

³ Open Source Initiative: Open Source Definition. *OSI*. [online] Available: <http://www.opensource.org/docs/definition.php>. [18 October 2006]

The idea behind this is to create a rationale in which developers create software that does not only make sense in the short term. Providers then cannot be tempted to make quick cash on a new release.

b) Source code

“The program must include the source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.” (OSI, 2005)¹

To make development of applications as easy as possible, the source code must be easy to obtain, regardless of who wants it. The user should also have freedom in choosing what method is used to obtain the application.

c) Derived works

“The licence must allow modifications and derived works, and must allow them to be distributed under the same terms as the licence of the original software.” (OSI, 2005)

If the source code is modified, the software should still be easy to distribute, under the exact same licensing as the original product. This encourages “rapid” evolution, thus leading to higher quality software². It can also be argued that it encourages programmers to make changes to software to further improve it.

¹ OSI is short for Open Source Initiative: Open Source Definition. [online] Available: <http://www.opensource.org/docs/definition.php>. [18 October 2006]

² “Higher quality software” in this sense can mean more reliable (bugs are fixed quicker) and also more features than comparative software with slower development cycles.

d) Integrity of the author's source code

“The licence may restrict source-code from being distributed in modified form only if the licence allows the distribution of "patch files" with the source code for modifying the program at build time. The licence must explicitly permit distribution of software built from modified source code. The licence may require derived works to carry a different name or version number from the original software.” (OSI, 2005)

End users have the right to know who is responsible for creating the software. End users might be using a specific product because of the developer's reputation, and therefore, if the software is modified, the end user needs to be made well aware that he/she is running a modified version. Open source software needs to be distributed in the “pristine” original way they were created - this protects the reputation of the developer, and also makes support for the product more readily available.

e) No discrimination against persons or groups

“The licence must not discriminate against any person or group of persons.” (OSI, 2005)

Open source software should be made available to any person interested. Often in the case of proprietary software, it is stated in the end user licence that the software may only be sold and used in specific parts in the world. If the software is of such a nature that it might be restricted in certain areas, the end user should be made aware, but the software must be made available. No end user may be locked out of the process.

f) No discrimination against fields of endeavour

“The licence must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.” (OSI, 2005)

The application should be available to anyone, regardless of field or sector. The main idea behind this is to not exclude the software from being used in certain sectors, and in doing so prevent licence “traps”.

g) Distribution of licence

“The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional licence by those parties.” (OSI, 2005)

The rights of a program should be identical to all end users. This prevents the software from being closed up by licensing methods like a non-disclosure agreement.” (OSI, 2005)

h) Licence must not be specific to a product

“The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's licence, all parties to whom the program is redistributed should have the same rights as those that are granted with the original software distribution.” (OSI, 2005)

In other words, all software that has been taken from open source distribution packages should have the same licensing agreement as the original “whole” product. This also prevents licensing traps.

i) Licence must not restrict other software

“The licence must not place restrictions on other software that is distributed along with the licensed software. For example, the licence must not insist that all other programs distributed on the same medium must be open source software.” (OSI, 2005)

The licence should only be applicable to the software it is distributed with. In other words the developers are only in control of their own software, not anything it is bundled with. In this way, distributors have full control of their software licences. The rationale behind this is that distributors of open source software have the right to make their own choices about their own software.

j) Licence must be technology-neutral

“No provision of the licence may be predicated on any individual technology or style of interface.” (OSI, 2005)

Distributors are not allowed to only release software packages in one specific form by licence. The software has to be licensed so that it takes into consideration that the software could spread using “non-Web channels”. Software is also not allowed to be distributed in only “click-wrap” form (Click wrap is a method of packaging software in one easy file to ease downloading processes from the Internet).

Open source software is made available in many formats - often over FTP servers directly for download, also on CD or DVD ROM usually for a small postage fee. However, the software licence cannot be specialized for certain distribution methods. In other words, an open source program downloaded from the Internet from the distributors' website must have the exact same licensing as a CD-ROM of the product being lent out to another person.

1.4 Different variations of open source licences

GPL:

GPL¹ is the first official open source licence, created by Richard Stallman. Richard Stallman wanted to create a licence which gave programmers the right to prevent the use of their code in proprietary software. Through the use of GPL, software (and no part thereof) may be used in the creation of proprietary, closed software. Stallman had the view that free software is the ethical responsibility of programmers. Anyone can licence their work under the GPL licence with hardly any effort. The creator can visit the Free Software Foundation website² and just copy and paste the text (available in a variety of languages).

¹ GNU General Public Licence. *Gnu.org*. [online] Available: <http://www.gnu.org/copyleft/gpl.html> [10 October 2007]

² The Free Software Foundation website can be found at: www.fsf.org

MIT:

The MIT licence is a “non-copyleft”¹ licence which allows for the re-use of code in both open source and proprietary software. Also called the X11 licence, it was developed at the Massachusetts Institute of Technology². X11 is the more accurate name, seeing as MIT holds numerous licences in different fields. The licence is not copyrighted, so the licence can be applied to other uses as well.

The Apache Licence:

Apache web server is the most popular open source server. In fact, it is considered an even better product than many commercial, proprietary alternatives³. Technically it is not compatible with the GPL licence because acknowledgement to the Apache Software Foundation is required. The Free Software Foundation sees this as incompatible.

One example of the Apache licence being used is in the IBM WebSphere product. IBM created a web server product in the middle nineties, only to discover that its market share was a sliver compared to a free, open source web server called Apache.⁴ Instead of competing with it, they drew up a legal contract that stated that any improvements made by IBM to the product had to be contributed back to the open source community. This required a legal document which is called the Apache licence.

¹ Copyleft is a licensing agreement where “copylefted” code can be used in other software for free, but the final product needs to be free as well.

² Open Source Initiative. 2006. MIT Licence. *OSI*. [online] Available: <http://www.opensource.org/licences/mit-licence.php> [10 October 2007]

³ This is reference to the fact that Microsoft bundles its own web server software with most modern Windows releases. Despite the fact that it is bundled, most web masters prefer to run Apache Web Server due to its easy setup and reliability and frequent feature updates, all the traits of mature open source software.

Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 122.

⁴ Friedman, Thomas. 2005. *The World is Flat..* pg. 90

Essentially it means that a company can build a patented commercial product, provided they cite copyright claims to Apache as well. In its modern day form, IBM WebSphere cites Apache in its “About” page.¹ Today, Apache is used by about two-thirds of all web sites. This initial “endorsement” of open source by IBM proved that trust can be put into open source software, and that successful business models can be shaped from well designed open source software.

Many software companies now use “vanilla” open source products and build their own tools on top of it. Novell is now one the biggest Linux distributors of Linux (SUSE Linux) and also follows this model. Jack Messman, the chairman of Novell states the following:² “Commercial software companies have to start operating further up the software stack to differentiate themselves. The open source community is basically focussing on infrastructure”.

1.5 The Cathedral and the Bazaar, and the peer review model

The way open source software is developed can also be different between groups. In the late 90’s Eric S Raymond wrote an essay titled “The Cathedral and the Bazaar”³. It conceptualized differences in open source software development. Stark contrasts are made between Cathedral and Bazaar models.

The Cathedral model releases the source code with every official release of the software. In between releases, the source code is not released. Only developers exclusive to the distributors have access to the source code between releases. Sometimes people refer to the Cathedral model mimicking proprietary software development. However, the Cathedral

¹ Friedman, Thomas. 2005. *The World is Flat..* pg. 90

² Searls, Doc. 2005. *Getting Flat, Part 1*. Linux Journal. [online] Available: <http://www.linuxjournal.com/node/8251/print> [10 October 2007]

³ Raymond, E.S. 1999. *The Cathedral & the Bazaar*. Thyrus Enterprises. [online] Available: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/> [10 October 2007]

model is specific in the development of open source software seeing as it does release source code, something that proprietary software developers do not do.

The Bazaar model uses the internet for the development of the source code throughout the development process, by whoever wishes to do so. This is similar to how the first release of Linux was created.

The paper highlighted the shortcomings in the Cathedral model of development. It also gave rise to “Linus’s Law”. This is in reference to Linus Torvalds, the original creator of Linux. He was also the first person to embrace the Bazaar model of development. Linus’s Law¹ states that “given enough eyeballs, all bugs are shallow”² or more elaborately “Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.”³ This is the greatest advantage to the Bazaar model.

1.6 Shortcomings in the peer review argument

In contrast to Linus’s Law, many closed source advocates use the term “security by obscurity”. The basic idea is that people who only have access to compiled form of programs will have less chance of seeing vulnerabilities in the application. However, this concept is a fallacy, as time will tell. Through trial and error or just plain luck, hackers have managed to find vulnerabilities in closed source code.⁴

If this argument is followed, published source code should increase discovery of flaws and problems. In fact, there are many occurrences where an open source application had a security flaw which was discovered many years later. This includes the SENDMAIL SMTP

¹ Another way of understanding Linus’s Law is to understand the ocular effect: "The more eyeballs that see it, the more likely it is that someone will see something that no one saw before."
(Wikipedia, 2006)

² Payne, Christian. 2002. On the security of open source software. Info Systems Journal 12, pg. 61

³ Raymond, E.S. 1999. *The Cathedral & the Bazaar*. Thyrus Enterprises. [online] Available: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/> [10 October 2007]

⁴ Hansche, S; Berti, J. 2005. *Official (ISC)2 Guide to the CISSP Exam*. pg. 229.

server, which had a lot of security problems. However, once a company was created with a dedicated team to develop the software into a commercial solution, its reputation improved dramatically.¹

1.7 Conclusion

Open source software is one of the areas of computer science with quite a rich history and has basically always been part of the computer landscape even with a recent surge in popularity. The difference now is that for the first time it can be viable to install in a main stream organization environment, not only the terrain of early adopters. This is because of open source's design philosophy should theoretically add up to better designed software. Better in this sense means software which has fewer bugs and runs more reliably due to Linus's Law. The problem with many releases of OSS is that it lacks a mature productized environment, something which will be explained in better detail in the next chapter. Open Source software could be created with a very specific type of licence, or a combination of different ones. Each of these licences gives the end-user certain rights regarding use and alteration of software. It is also important to differentiate between open source and free software.

¹ Payne, Christian. 2002. On the security of open source software. *Info Systems Journal* 12. pg. 69

Chapter 2

Types of Open Source Software

In order to make an effective transition to open source, a thorough analysis of all options available need to be done. This chapter will focus on feature comparisons in operating systems, productivity software and miscellaneous software which the average computer user might find useful. Once feature sets are analysed, packages can be chosen according to the software maturity model as well as taking specific advantages and disadvantages of open source into account. This chapter will focus on the main characteristics of different open source software types, including operating systems, productivity software amongst others.

2.1 Operating systems

An operating system is the core controller software which manages a computer's resources such as processor prioritizing and memory allocation¹. In modern iterations it is accompanied

¹ Operating Systems. [online] Available: http://en.wikipedia.org/wiki/Operating_system [15 May 2006]

by a graphical user interface which becomes the standard method of interacting with the computer and all its applications.

Linux is an open source operating system for computers and computerized devices. Development of Linux began in 1983 by the GNU project, spearheaded by Richard Stallman. GNU wanted to develop a completely free UNIX style operating system. In 1991, the Linux kernel was released. This “kernel” is the basic core of the program with compilers, device drivers and libraries which have been collected over the years by the GNU project.¹

Initially, Linux was an enthusiast project, and was only used and developed by individuals, people dubbed many as “geeks”. However, over time it has gotten more industry support, notably from major industry leaders such as IBM, Hewlett Packard and Novell. Linux has proven to be very beneficial in the company server markets (due to security benefits and reliability²), and therefore providers of this hardware have been supporting Linux. Linux is made available in “distributions”. These distributions are created by various groups, some informal, others professional and commercial in nature. These distributions are all different in the way they are installed and the applications they use.

Linux is considered the “backbone” in modern corporate server environments, because it is less expensive, more scalable and more reliable than equivalent mainframe operating systems. Also due to its much broader hardware support, the installations can be made on a variety of computer platforms.³⁴

¹ Linux Kernel. [online] Available: http://en.wikipedia.org/wiki/Linux_kernel [25 May 2006]

² Laudon, Laudon. 2006. *Management Information Systems: Managing the Digital Firm*. 9th Edition. Pearson. pg. 201.

³ A typical Linux distributions is made available for a variety of hardware platforms. This is in contrast to operating systems like Windows which is only made available to the x86 platform.

⁴ Laudon, Laudon. 2006. *Management Information Systems: Managing the Digital Firm*. 9th Edition. Pearson. pg. 201

2.1.1 Distributions of Linux

Unlike Microsoft Windows or Mac OSX, the Linux operating system exists in many different versions; all created by different developers, called distributions. Examples of different distributions include Red Hat, SUSE, Ubuntu, Mandriva (previously Mandrake), etc. The only thing common among these releases is the core Linux kernel, which is continuously being upgraded by the Linux community. No developer is in itself responsible for the development of the Linux kernel, and thus is not “owned” by any group or party. The challenge of using the Linux kernel is that it is continuously being upgraded, so not all Linux distributions can be released with the latest kernel.

Each kernel update supports more recent hardware developments and in general it improves system stability. For example a recent kernel started supporting Hyperthreading. This is Intel's technology of making a single processor core able to support two instructions at once, thus dividing the core into two “logical” instead of physical units. In multitasking environments this greatly improves performance. This makes the operating system believe the system is a multiprocessor or multicore platform. So if a significant new technology appears, new versions of the Linux kernel will target those developments fairly quickly.

What differentiates distributions? Firstly, even though they all run the same kernel, each distribution has different software packages that are built in with the kernel. This software includes the interface being used, system related utilities and productivity software that might include office applications (e.g. OpenOffice) and photo modifying software (e.g. The GIMP). Usually, during the installation of the distribution, the user can choose from an array of different applications, even if they perform the same basic function.

Therefore, despite the number of choices the end user has, in the end he/she will only need a few. Modern installation packages (e.g. the latest Mandriva) choose “default” packages depending on the user. The user simply chooses if he/she is an office user, multimedia user, gamer, developer or any combination. This eases the installation process for the end user. Previous versions of Linux installation were much more difficult to install, due to the fact that the user had to be familiar with UNIX commands.¹

¹ Andrews, J. 2005. *A+ Guide to Managing and Maintaining your PC*. pg.1154.

Vendors release “modified” distributions, which are packages that are specifically created for specific hardware platforms or usage scenarios (e.g. Mandriva might also create a distribution which runs on Apple PowerPC platform instead of Intel based x86 hardware). These modified packages might also improve on certain aspects of the original software package, such as being more accessible to disabled users or making a bootable CD (often called “Live” releases) or releases specially made for children.¹

2.1.2 Differences between Linux distributions

The differences among distributions are normally the following:

a) Language and geographic location

A lot of Linux distributions are translated for different world regions. For example, Ubuntu (which is a South African release of Linux supported by Canonical) has the option of a South African language pack, which also includes languages like Afrikaans, Xhosa and Zulu.² These language packs have been built into many distributions of Linux and they include languages from “Brazil, China, Egypt, France, Hungary, India, Israel, Italy, Japan, South Korea, New Zealand, Poland, Portugal, Russia, South Africa, Spain, Taiwan, Thailand and the United States”³

These different language packs aid greatly the ease of use of the operating system, especially in rural parts of third world countries where English might not be considered a “compulsory” language to learn. This also includes specific currencies and differences in decimalized systems of measurement, such as imperial or metric systems.

b) Hardware Support

¹ Hubley, Mary I. Librano. 2005. Cynthia. Linux Operating System Distributions: Perspective. *Gartner Research*. pg.5

² Official Ubuntu website. [online] Available: <http://www.ubuntu.com/desktopedition>[5 June 2007]

³ Hubley, Mary I. Librano. 2005. Cynthia. Linux Operating System Distributions: Perspective. *Gartner Research*. pg.5

Most Linux distributions are released in different versions depending on the hardware it should run on. Not all computer hardware is compatible - for this very reason, Linux distributions are made to run on as many platforms as possible. Most Linux distributions are released for Intel based x86 hardware and Apple PowerPC hardware. Even in the case of embedded hardware and ARM processors (found in devices such as Palm and Pocket PC handhelds) there is a version of Linux available. Recently Linux developers also successfully wrote a version of Linux for Apple iPod music players, expanding the feature set to more advanced (3 dimensional) games and video playing abilities.¹ This is because an iPod is essentially a barebones computer with processors, memory, hard drive, screen and input system.

c) Support

Commercialized versions of Linux often have full-time contracted support staff as part of support costs. Non-commercialized distributions have no formal support structure - however support is available in more “loose” structure - these support systems are not funded by the developers but are mostly run by peers using the same distribution. In most cases, users can find support in the form of official forums on the internet. In some cases, an annual payment is required to receive security bug fixes,² but this is normally only with commercial versions such as RedHat or SUSE Enterprise edition. Certain releases are also credited as “long term support” version, for which the company plans to have a longer formal support period for the software.³

d) Target users

¹ Hubley, Mary I. Librano. 2005. Cynthia. Linux Operating System Distributions: Perspective. *Gartner Research*. pg.5

² Silver, Michael A. 2005. Linux Desktop yet to Reach Potential in Emerging Markets. *Gartner Research*. pg. 2

³ For example, Canonical, the creators of Ubuntu Linux has chosen it version 6.06 as a LTS (Long Term Support) release. This can also be a good sign that the software is probably more stable than other releases, and has undergone a more rigorous development cycle.

Different distributions of Linux are made for different people, for example, Slackware is more suited towards technically skilled users, where Mandriva (or Mandrake previously) is the easiest to use for new users, with a good support base. The “expert” distributions like SUSE or Red Hat might be better suited towards enterprise level server applications, where home users might find more use from distributions like Mandriva or Ubuntu. Ubuntu also targets a version especially for younger children, called edubuntu¹.

Different users also demand different levels of support for their preferred software. For example, someone might choose a certain release of Ubuntu, because it has a longer support service period.² For someone starting out on Linux, it is always a good idea to go with the most popular distributions, seeing as the support base for these products are always bigger. The “big three” suppliers for consumers in Linux is Ubuntu, SUSE and Mandriva (however, due to the nature of the industry, this can change very quickly). Obviously, corporate and government clients will go for commercial distributions such as Red Hat or SUSE Enterprise.

e) *Live disc*

A live disc is a method of running a Linux distribution without any installation process – therefore, the operating system is temporarily stored on the hard drive for the duration of the operating session. Some distributions are available only in “live” form (like Knoppix and GoblinX), where some regular distributions of Linux are available in modified form on a bootable disc that the user can use to get a feel of the distribution. The advantage of using a product like this is that no installation is made on the computer to run the operating system. The system is then mostly full featured, but it is not as fast as an installed-to-hard disk product. Therefore, the “live” disc is mostly suited to new users and hobbyists who want to get a taste of different distributions and then the user can select the one he/she feels most comfortable with. If a user needs to decide on which distribution to use, it often makes sense to first use live discs, even though the functionality is somewhat impaired.

¹ Edubuntu runs on the same base code as Ubuntu Linux but has a modified interface which makes it easier to use for younger children. Educational applications are also included in the package.

[online] Available: <http://www.edubuntu.org/> [10 October 2007]

² Canonical calls these releases of Ubuntu the “LTS” releases, standing for “Long Term Support”.

f) Price

Once again, it is necessary to differentiate between commercial and non-commercial distribution. Non-commercialized distribution can usually be downloaded for free from the Internet and can be obtained on discs for the cost of the postage of around R100. Canonical, the fundees of the Ubuntu distribution, offers to post the discs for free to anyone in the world. Commercialized distributions might mimic the same pricing structure as proprietary software, like Microsoft Windows or Apple Mac OSX, where the user has to purchase a licence for every copy used. The cost for these commercial distributions often constitutes support for the product and updates. So therefore, it is important to remember that zero licensing cost does not mean free, although they are much cheaper to licence than proprietary software.¹

The TCO equation becomes complicated when taking aspects like training, migration and the maintenance of older PC into account. Older PC's hardware maintenance becomes more expensive with age. Hardware manufacturers focus efforts on modern hardware, which is often cheaper to manufacture. For example, a memory upgrade for a five year old PC (EDO RAM, an older, slower type of RAM) would be more than double the price of a 1 year old PC (in most cases, DDR or DDR2 type RAM, which is much faster and more efficient). Often these upgrades or part replacements can be so expensive that it indeed becomes cheaper to buy an entirely new computer system. Seeing as Linux could be run on a broad range of hardware (including older hardware platforms), the total cost of hardware could possibly be cheaper than with proprietary operating systems. This however does not take into account total cost of ownership, which is calculated over the length of use of the hardware, for example, three years.

g) Security and management tools

Depending on the distribution, all have security and system management tools. The differences occur in the way these systems are implemented. Most commercial distributions put emphasis on the management tools - they differ in the way the user interacts with it. Obviously it is easier to use a graphical user interface, but many technical users prefer to use text line interfaces to the Linux kernel (Slackware is a good example). Other distributions

¹ Andrews, J. 2005. *A+ Guide to Managing and Maintaining your PC*. pg. 1154.

choose to make management functions more of a “background” feature, something the user does not necessarily have to be made aware of. The key here is to choose a distribution of Linux that caters for your needs and skills in terms of manageability.

Commercial versions of Linux also emphasize the security features of their software - in many cases during the installation the user is asked whether he/she would prefer to have the security features. Since Linux can be installed on older or cheaper hardware, it is possible to make a dedicated hardware firewall¹ for a network Internet connection, running from computers which are normally not used anymore². Users are also given this option during a typical Linux installation of any modern version of Linux.

Each Linux distribution has specific advantages and disadvantages. Depending on the sector in which the software is being used, system administrators have to decide which distribution to use. One first has to differentiate between commercial and non-commercial releases of Linux. Commercial distributions might have a licensing cost, but they normally have much better support for the software if problems do occur and during updates. Commercial distributions are also a better choice if internal staff skill is not particularly high due to lower possible total cost of ownership.³

Often these commercial distributions might charge a support fee instead of a licensing fee. Examples of commercial distributions include Fedora, SUSE Linux Enterprise Server (recently acquired by Novell) and Red Hat. These distributions are formally supported by its developers and are often of very high quality. On the other hand, the user or administrator might choose to use non-commercial distributions. The advantage of this is that there normally is not any licensing fee (with exception of Linspire). However, non-commercial

¹ A firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be both software or, as in this case, hardware based.

Hansche, S; Berti, J. 2005. *Official (ISC)2 Guide to the CISSP Exam*. pg. 794.

² In many cases this can save costs. Old computers used as firewalls are normally better solutions than expensive network routers with built in firewall and security features.

³ In many cases the salaries of hiring new support staff might be cost more than buying a integrated support solution from the vendor.

distributions might be more focussed on more technically skilled users and the support base is normally the Linux community who is under no obligation to give support. It can happen that IT might encounter a problem which has not been addressed yet. This places great pressure on IT staff, with perhaps dedicated members whose sole responsibility is the smooth running of the system.

2.1.3 Linux desktop environments

Linux has the unique ability to be completely modular in its graphical user interface, called the “desktop environment”. A desktop environment runs on the basic desktop metaphor which forms the basis of most modern graphical user interfaces.¹ A user manipulates objects on the screen which might have some similarity to real life objects.² All in all, these elements create the graphical user interface that a user uses to interact with the computer. This can be difficult to understand at first seeing as many users are used to an operating system having only one standard set of “controls”. Microsoft Windows does not traditionally allow the user to alter its user interface, bar for the simple alteration of colours, fonts and effects. This is because Windows and Mac OS X platforms the user interface is unified by a single toolkit which have one specific set design for buttons, title bars and windows. This is to encourage a constant look and feel interface for the user. Linux desktop environments are completely customizable through the use of X Window Managers.³

In the case of Linux, there are two major desktop environments a user can choose to use: GNOME and KDE. These interfaces are quite different, each with their own feature-set and user interface. They are very customizable and can be altered for different users.

GNOME

¹ Satzinger, Jackson. 2006. Systems Analysis & Design in a Changing World. pg. 491.

² For example, in Windows the user can drag a file into a trash can called the “Recycle bin” which has a visual similarity to a actual trash can. Most modern operating systems have a similar feature.

³ Chapman, Matt. 2007. Window Managers for X. [online] Available: <http://xwinman.org/> [7 June 2007]

GNOME is a project by the Free Software Foundation to create an easy to use software environment entirely made from free software. GNOME is the official desktop environment of the GNU project as well.

According to the GNOME website,

*"The GNOME project provides two things: The GNOME desktop environment, an intuitive and attractive desktop for end-users, users, and the GNOME development platform, an extensive framework for building applications that integrate into the rest of the desktop."*¹

GNOME constitutes not only the interface, but the applications it is bundled with as well. These include²:

- AbiWord — a word processor (similar to Microsoft Word).
- Epiphany — a web browser (similar to Internet Explorer or Firefox).
- Evolution — for contacts/time management and e-mail (similar to Microsoft Outlook).
- Gaim — an instant messenger (similar to MSN Messenger, iChat or Google Talk).
- Gedit — a text editor (similar to Notepad in Microsoft Windows).
- The Gimp — an advanced image editor (similar to Adobe Photoshop).
- Gnumeric — a spreadsheet (similar to Microsoft Excel).
- GnomeMeeting — for telephony and voice over IP (similar to Skype).
- Inkscape — a vector drawing application.
- Nautilus — a file manager (mimics Windows Explorer).
- Rhythmbox — a music-management application (similar to Apple iTunes).

¹ GNOME About page. 2007. [online] Available: <http://www.gnome.org/about/> [10 October 2007]

² For a full list of applications included in GNOME, visit the Wikipedia page at: <http://en.wikipedia.org/wiki/GNOME>. A screenshot of GNOME can be seen in figure 2.1.

- Totem — a media player for video and audio files.

These above choices can of course be substituted with any choice of alternative software, for example, the user can choose to run OpenOffice.org instead of AbiWord. GNOME is available for a variety of operating systems, not only Linux. This includes Mac OS X. In other words, if a user prefers to do so, he or she could replace their operating system's interface with GNOME. It is available on a variety of hardware platforms, including portable devices.¹ This includes Pocket PC and Palm.

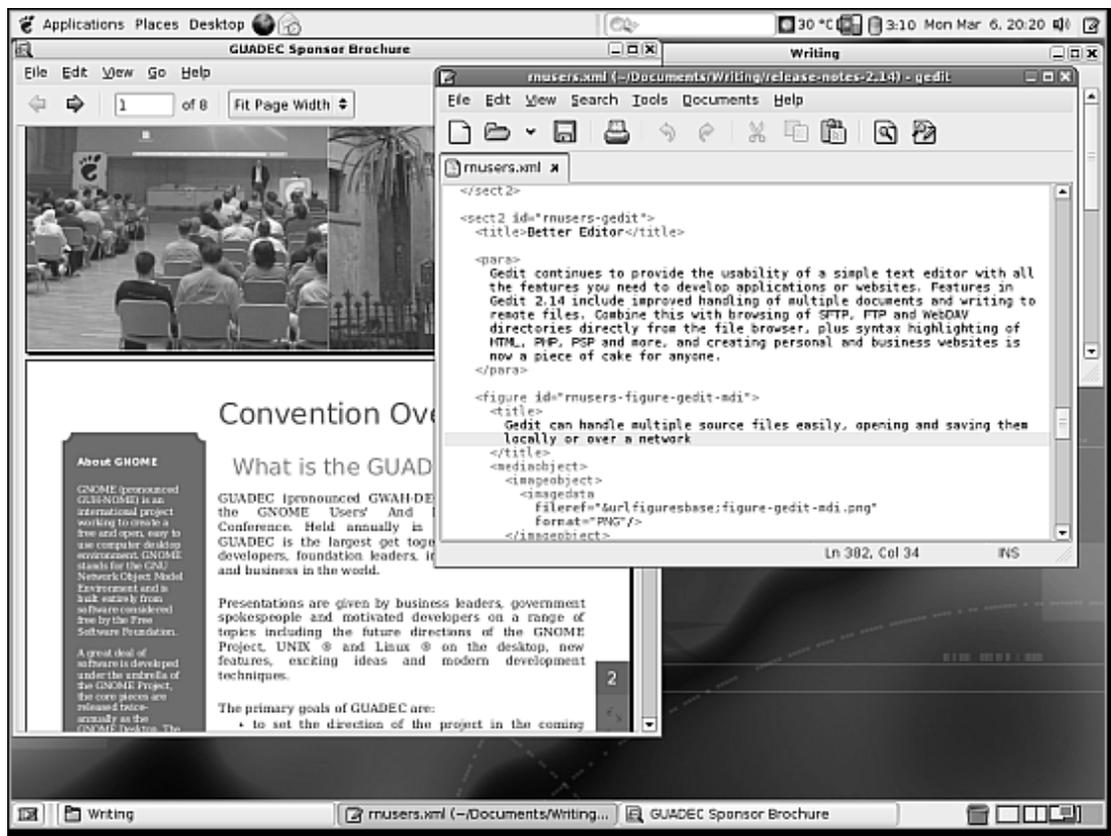


Fig 2.1: A Screenshot of the GNOME user interface

¹ Hubley, Mary I. Librano. 2005. Cynthia. Linux Operating System Distributions: Perspective. Gartner Research. pg.5

KDE

This desktop was initially designed by Matthias Ettrich in 1996.¹ The idea was to design a better interface for UNIX, due to the inconsistency in the user interface. He wanted to create a desktop that was easy to use and an accompanying application set. The project quickly became popular amongst developers. It is primarily a volunteer project, but various commercial companies' contract employees to further develop the desktop environment – these include Mandriva and SUSE (or Novell)².

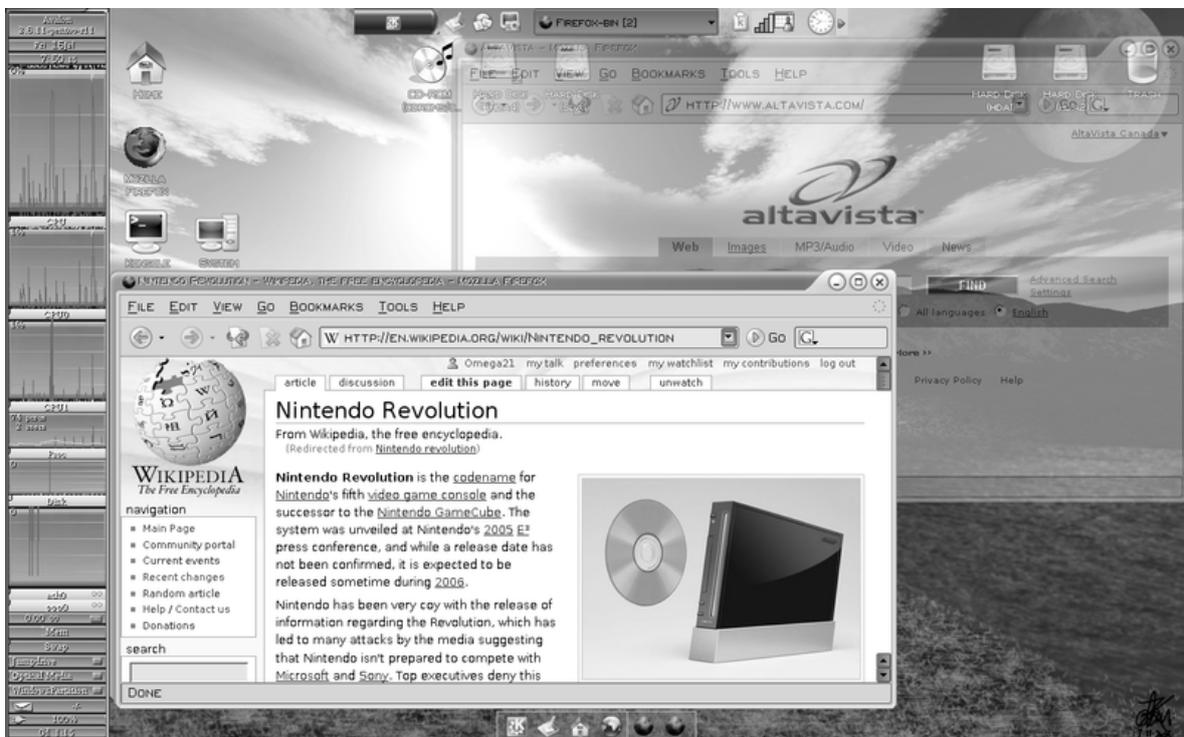


Fig 2.2: A screenshot of the KDE desktop user interface using full hardware rendering translucency, a feature which is not available in Windows XP.

¹ Background information of KDE found on Wikipedia website. Can be found at: <http://en.wikipedia.org/wiki/KDE>

² Novell recently purchased SUSE Linux in order to further penetrate the Linux server market.

KDE includes a massive library of applications¹, drivers and tools, and these applications fall under “K” categories. This library includes²,

- kdebase - The base desktop and applications. Requires kdelibs.
- kdeaccessibility - Accessibility software.
- kdeaddons - Add-on software.
- kdeadmin – System management tools.
- kdeedu - Educational software.
- kdegames - Games.
- kdegraphics – Tools for photo and graphic editing.
- kdemultimedia – Applications for the use of multimedia features.
- kdenetwork – For the administration of networks.
- kdepim – PIM and e-mail management software similar to Microsoft Outlook.
- kdesdk – Applications used in the development of KDE.
- kdetoys – “Play Things” for your desktop. Similar to widgets used in Mac OSX.
- kdeutils – System orientated utilities.
- kdewebdev – Applications used in the development of web pages.
- koffice - Office suite. This software can be substituted by any Office application, including OpenOffice.org or StarOffice.

Modern versions of KDE have very advanced interface abilities, including full interface translucency rendered on hardware level (as can be seen in figure 2.2). The advantage of this is that graphics hardware is used to create the user interface instead of using valuable CPU cycles. The idea behind this is to “spread” the processing load between the central processing

¹ Hubley, Mary I; Lubrano, Cynthia R. 2005. Linux Based Graphical User Interfaces: Perspective. *Gartner Research*. pg. 5

² For a full list of software available in the KDE package, go to: <http://en.wikipedia.org/wiki/KDE>

unit and the graphics processing unit, which dramatically improves overall system performance. Microsoft Windows does not yet support this ability; however Windows Vista takes advantage of this feature. Mac OSX has been using a similar type feature for a few years, called the Quartz engine – which is one of the many reasons why Mac OSX does have better system performance than Windows on modern computers with high powered graphics hardware. KDE is also available on a variety of operating systems including Mac OSX¹

2.1.4 Comparison between Linux, Microsoft Windows and Mac OSX²

Modern operating systems tend to share the same visual cues and in this way they become more user-friendly. Each operating system has its own history, which can add insight into why certain decisions were made during its development. This paper has already covered the history of Linux, so here we will go through a brief history of Windows and Mac OS X.

a) Apple Mac:

In 1984, Mac OS became the first OS to use a point-and-click graphical user interface using a mouse.³ It mimicked an interface which had folders and simple windows which could be moved and manipulated using a mouse. It was a stark contrast to the then popular Microsoft DOS command line operating system. Apple tried to distance itself from command line system, seeing it as archaic and technically challenging. In fact the first Mac OS was completely devoid of a command line. The operating system was designed to run only on Apple computers, a fact which many ascribe to being Apple's worst business decision. This operating system was continually developed up to 2001 when a complete revision of the operating system was released, called Mac OS X.

¹ Hubley, Mary I. Librano. 2005. Cynthia. Linux Operating System Distributions: Perspective. *Gartner Research*. pg. 5

² Information in this table is objective non-biased viewpoints of the writer.

³ Evans, Alan; Martin, Kendall. 2006. *Technology in Action: Third Edition*. pg. 196

Mac OS X was a major step forward in that it supported levels of multitasking which was unheard of in the previous versions of Mac OS. It supported protected memory and was a very stable OS. Since then a version of Mac OS X has been released on an almost yearly basis with release 10.5 being scheduled for April 2007. Apple also completed its transition to Intel hardware in 2006, which in turn enabled Mac users to boot into Microsoft Windows should they choose to do so.¹

In reading Table 2.1 comparison of the operating systems it is important to realize that Mac OS X is only available on Apple hardware. Therefore, it is important to take certain Apple Mac hardware issues into account. Apple is deliberate in this philosophy, insisting that users be made aware of the Apple “ecology” that ties its software and hardware. Apple has also used this method in its highly successful iPod and iTunes music system, where great software essentially forces people to use Apple hardware for the best integration. Despite conventional logic that “open” compatibility will benefit business of an online music store; the closed system used by Apple has been a great commercial success, with almost 69 million iPod devices selling by October 2006. Apple’s closed system forces the user to purchase Apple hardware to gain the ability to run Mac OS X, and in that way, Apple hardware is a great part of the decision of using Mac OS X.

b) Microsoft Windows:

Unlike Mac OS, Windows has always been able to run on any machine with BIOS and x86 processor architecture (or also called IBM PC compatible). Microsoft Windows was released in November 1985 as an add-on for its MS-DOS product. It was an answer to a growing demand for a graphical user interface which was popularized by Apple. During its development cycle, Microsoft placed its Windows products in two categories – business and consumer. Consumer versions (such as Windows 95/98/Me) have focussed more on “eye candy” and multimedia-centric features, whereas business versions (such as Windows NT) have focussed more on networking and security. However, consumer versions were plagued

¹ Booting into Windows requires Boot Camp (a free Apple supported product) or Parallels Workstation, a product that enables Windows to run in a virtual machine within Mac OS X.

by instability and poor security. Fortunately Microsoft merged the code base of consumer and business lines with the release of Windows XP, the first truly 32 bit Windows product. Microsoft still produces “Home” and “Professional” versions each with their own feature-sets. Microsoft now releases a number of unique versions of Windows depending on its hardware requirements and feature-sets.¹ The next version of Windows, called Vista will be released in early 2007.

Table 2.1: Comparison between Linux, Microsoft Windows and Mac OS X

	GNU/Linux 	MS Windows 	Apple Mac OS X 
Price	In most cases no cost involved. Commercial distribution support can be costly however. Costs range from free to about R1000 for	DSP licence included with hardware about R800 for Home edition. Professional DSP edition about R1200. Fully packaged versions cost about a third more. Multi	Latest Mac OS X 10.4 Tiger costs about R900. ² Five licence “family” pack costs about R1500. However, Apple hardware is required to run Mac OS X, a fact

¹ This includes XP Tablet edition for Tablet PCs, Media Center Edition which contains an enhanced interface ideal for use further away from the screen and x64, which was released to support the new 64 bit architecture. XP x64 was not a big success due to lack of drivers. Vista will hopefully improve on this.

² Pricing found from official Apple South African website. Can be found at:
[http://www.zastore.co.za/index.php?lm=IGstore&cf=ProdList&p\[prod_id\]=2561&d=1&p\[cat_id\]=ASM](http://www.zastore.co.za/index.php?lm=IGstore&cf=ProdList&p[prod_id]=2561&d=1&p[cat_id]=ASM)

	GNU/Linux 	MS Windows 	Apple Mac OS X 
	<p>packaged, well documented SUSE Linux. Server editions cost more.</p>	<p>licence packages only available to education and corporate clients. At first, Vista will cost even more than this due to inherent demand.</p>	<p>that negates the relatively low cost.¹ Even though OS X is bundled with every Mac, Apple hardware can be prohibitively expensive, especially in South Africa. In the US, Apple hardware is starting to compare favourably with PC hardware, especially since the recent switch to Intel hardware.</p>
Availability	<p>Commercial versions can only be obtained from official sources, but software can be downloaded</p>	<p>Can be easily obtained from all computer retailers. In most cases it can be cheaply bought with certain</p>	<p>Needs to be obtained from official Apple retailers. In South Africa, these are few and far between. Every</p>

¹ Apple hardware is in general more expensive than standard personal computers. Despite sharing a relatively common hardware platform, Apple computers pay a lot of attention to design of hardware, and thus charging their machines at a slight premium.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
	<p>directly from company websites and P2P networks.</p>	<p>hardware types in the form of OEM distributions¹ (e.g. RAM, processor or hardware).</p>	<p>Apple Mac computer is bundled with latest OSX and iLife bundle². This includes the Mac Mini, iMac, Mac Pro and all notebook computers.</p>
Ease of Use: First time user	<p>Intuitive interface, however, users who are used to Windows can find it difficult to use. Does share some elements, but no real resemblance to Windows due to patent issues. Linux's graphical user interfaces (GUIs) have had good</p>	<p>Quite easy to use interface. Most people learn to use computers on the Windows interface; therefore it is the most prolific of the operating systems.</p> <p>Vista promises further user interface enhancements which use 3D hardware components. This</p>	<p>Widely considered to be the most user-intuitive. Even users switching from Windows find it easy to use. Modern Intel Macs can run Windows as well.</p>

¹ OEM distributions are normally made available if a customer purchased a key hardware component of a computer as well (such as a processor or hard drive). This is full featured software with normally less attention given to packaging such as manuals or a box.

² Heid, Jim. 2006. *The Macintosh iLife '05 in the Classroom*. pg T6.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
	usability for some time, and most Windows and Mac users could adapt to them relatively easily. ¹	includes translucency, shadows and quicker rendered redraws of Windows.	
Ease of use: Power user	Linux interface can be altered completely and a variety of Desktops exist for Linux. Depending on the distribution, the software can be altered for ease of use (narrow menu hierarchy) or easy	Windows interface tends to put all functions “in your face” ² . Due to intricacies and quirks in the interface, Windows can be harder to use, especially in the case of creative applications which are available on more than one	Considered to be the easiest to use by creative professionals ³ . Most applications which are written for Windows and Mac usually are better on Mac OS. This also includes Microsoft Office. Apple has pretty stringent

¹ Silver, Michael A. 2005. Examining where desktop Linux and Open source Office makes sense. *Gartner Research*. pg. 3

² Microsoft Windows applications are made to show all available functions to the user at once. This can include menus, tick boxes and buttons. MacOSX is more intuitive by showing only the most relevant options, letting users “dig” for less frequently used applications.

³ Scott Thomas, R. *Quartz 2D Graphics for Mac OS X Developers*. pg.XV

	GNU/Linux 	MS Windows 	Apple Mac OS X 
	<p>access to all functions (wide menu hierarchy).</p>	<p>platform. Each application pretty much follows its own methods for creating the interface.</p>	<p>guidelines for the design of the interface including the placement of toolbars, the creation of icons and overall appearance which should fit in with Mac OS X in order to create a consistent user interface across all applications.</p>
Support	<p>Only real negative to Linux. Depending on distribution, support can be either formal or informal.</p> <p>Formal support can be costly and negate</p>	<p>Most popular operating system, so the support base is the biggest. Patches to system can be considered to be too late in most cases. Patches are released second Tuesday of every month.¹</p>	<p>Apple community is very pro-active and support is considered to be very good. However, Apple users only account for about 3 percent of all computer users, making it hard to find</p>

¹ If a patch is desperately needed, like in the case of a fast spreading malicious worm or virus, Microsoft can shift its update patch schedule, as was done in December 2005.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
	<p>the zero licensing cost. However, support is proven to be of high level on commercial distributions of Linux.</p>		<p>“informal” help. Apple does not formally support Microsoft Windows. Apple is also very quick to release bug fixes in the form of big “security updates” which occur often.</p>
Security Feature set	<p>Patches are released constantly, whenever needed (not a specific schedule), and updates on Linux kernel are released frequently. Very popular for server environments where security and uptime is important.</p>	<p>Worst aspect of Windows. Hackers target Windows through worms and malicious software. Microsoft is also slow in releasing patches¹, relying on a fixed patch schedule. Many times the patches do not adequately fix the problem, requiring “re-fixes”. Vista release</p>	<p>Considered to be most secure operating system. Mac OSX is very reliable and Apple is quick in releasing patches. OSX is updated with a major release every year as well. Also based on UNIX kernel similar to Linux and several open source components from Debian.</p>

¹ Microsoft releases patches on the second Tuesday of every month, called “Patch Tuesday”.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
		was built from the ground up with security as a primary concern. ¹	
Networkability	<p>Full feature set. Includes wireless networking. However, drivers can be hard to find at first, especially using new model hardware components, which also require knowledge of where to find drivers. Initial setup can also be complicated for more advanced networking setups.</p>	<p>Full feature set. Includes wireless networking. Has the best compatibility with Windows servers.</p> <p>Was initially slow with wireless network implementation, but modern releases and updates have enabled full featured use of wireless networks.</p>	<p>Full feature set. Includes wireless networking. Supports most industry standards, including many that Microsoft does not support. Very simple networking setup, especially in terms of wireless connections.</p>
Multimedia feature set	Not considered a strong feature of Linux, however, it	Windows has powerful multimedia support, although the operating	Apple hardware is bundled with multimedia suite

¹ Despite this claim, only time will tell whether Vista is indeed a more secure release of Windows. Release candidate versions have already been compromised on many occasions.

	<p style="text-align: center;">GNU/Linux</p> 	<p style="text-align: center;">MS Windows</p> 	<p style="text-align: center;">Apple Mac OS X</p> 
	<p>supports all forms of media. Open source third party multimedia software is considered to be good and Linux multimedia performance is better than Windows. Because of its free nature, many necessary codecs are not available bundled with the package.¹</p>	<p>system is blamed for its slow performance. Creative professionals tend to stay away from Windows. Microsoft also created Media Centre Edition², which created a new niche in “lounge” computing.</p>	<p>(iLife) and each release has support for the latest industry standards in video and audio. Because of this software, Macs are considered much more useful “out of the box”. Considered leaders in multimedia simplicity and is a favourite with creative professionals. Apple marketing places great emphasis on the “entertainment” side of the computing</p>

¹ Codecs are proprietary software drivers used to play back certain media files such as video and audio. For example, many videos are encoded in a “DivX” codec, and DVD discs also require certain codecs. Because the Linux software is free, it cannot pay for royalties per user of the codec. Therefore, the end user might have to go through extra trouble in finding necessary codecs in order to play back files.

² Windows XP Media Center Edition is a variation of the normal Windows. It contains the same feature set as Windows XP Home Edition but has an added “10 foot interface” which is used when sitting further away from the screen, using a remote control.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
			experience.
Software included	Software bundle included is massive in size, and user is given a large number of choices. Can be troublesome to choose the right software package though – the user is given many alternatives to a single use. For example. A typical installation of Linux will install more than one web browser.	Software bundle is not considered to be functional in all areas. Latest XP releases focus on multimedia player software. Vista will have several key updates in terms of basic software functionality. Functionality is nearing the feature set of Mac OS X and iLife in terms of photo, music and video abilities.	Software is of very high quality, and is included with the purchase of Mac hardware. This includes the multi-faceted iLife bundle which is updated yearly. Mac OS X has a lot of functionality built right into it; often the same functionality requires purchasing expensive software on other operating systems.
Reliability	Very secure and reliable. Some might even call it “crash-proof”. ¹ This has	Past consumer releases of Windows suffered from poor reliability. (Windows 95, 98 and	Mac OSX runs on a UNIX kernel which is very stable. Is a big selling point of Apple

¹ Andrews, J. 2005. *A+ Guide to Managing and Maintaining your PC*. pg. 1154.

	GNU/Linux 	MS Windows 	Apple Mac OSX 
	<p>been one of the key selling points of Linux and is the also one of the primary reasons for its popularity in the server market.</p>	<p>Millennium Edition) Windows XP fixed this. However, Windows is still not considered exemplary. Vista is very reliant on performance of hardware and quality of drivers for overall user experience.</p>	<p>software and hardware. Modern releases of Intel Mac portables have however been plagued by overheating issues, but this is primarily a hardware problem, not an OS one.</p>
Quality of 3rd party software	<p>Software is of high quality. Problem is that it can be hard to find the relevant tool for the task at hand, also, choosing and installation can require an in depth knowledge of Linux and its quirks.</p>	<p>Quality can range from poor to brilliant. Can in some way be blamed upon the Windows kernel, which is very reliant on how well software is written.</p>	<p>Good quality. In many cases, software which is available on many platforms is considered best on the Mac OSX platform. Does have less software available than Windows though.</p>
Operating	<p>Most versatile operating system.</p>	<p>Available only on x86, x86 64bit and Itanium</p>	<p>Available for PowerPC hardware (IBM and</p>

	GNU/Linux	MS Windows	Apple Mac OSX
			
System's Quality of Hardware compatibility	Most distributions are available on a variety of hardware platforms including x86, PowerPC, embedded systems and portable devices with ARM processors (like Palm and Pocket PC).	(in servers) hardware. X86 hardware is also the cheapest to manufacture ¹ and can be found in most computer hardware.	Motorola). However, in 2006, Apple moved to Intel x86 hardware. ² Emulation of PowerPC is of high quality in OSX 10.4.4 with Rosetta emulation engine. ³ Performance is further improved on Mac Intel hardware. Gets low score because OS X can only run on

¹ X86 hardware is the cheapest to manufacture not because it is of inferior quality or performance, but because of economies of scale, which drops the prices of the hardware. It is one of the primary reasons of the Apple shift to Intel hardware, despite Apple always marketing their previous generation PowerPC machines as the “fastest in the world”.

² In January 2006, Apple released their first Intel machines. First released was the iMac and MacBook Pro, both with dual core Intel processors. Initial reactions are that machines are much faster. Apple completed its transition to Intel hardware by September 2006. These Intel machines also have the added benefit of being able to boot Mac OS X, Windows (using Boot Camp) or Linux, giving the end user more choice.

³ More info about the Rosetta emulation engine in Intel Macs can be found at <http://www.apple.com/rosetta> . Rosetta is an emulation engine that makes it possible for Intel Macs to run older PowerPC based software. Initial reaction is that it runs satisfactorily and reliably.

	GNU/Linux	MS Windows	Apple Mac OSX
			
			Apple hardware.
Emulation	Variety of emulation options exist, the most famous of which is WINE ,which emulates Windows within a window or full screen. No options exist for emulating Mac OSX.	VMware can emulate any Linux distributions and even illegal versions of Mac OSX (Intel) have been ported. PearPC can be used to emulate the PowerPC architecture.	VirtualPC, a Microsoft application, can emulate Microsoft Windows XP on any G4 or G5 system. Linux can run on PowerPC platform without emulation. In April 2006, Apple released Boot Camp, an extension that enables the booting of Windows XP on any Intel based Mac, something which Mac users have never been able to do. ¹ The upcoming release of Mac OS X will have

¹ Despite a strong “anti-Microsoft” attitude amongst Mac users, this feature was in high demand. In essence, Boot Camp effectively makes Macs the only computers that can boot into Windows, Linux or Mac OS X. Apple has been quick to market this, calling their machines “the only computer you will ever need” in its “Hi, Im a Mac” campaign.

	GNU/Linux	MS Windows	Apple Mac OSX
			
			native emulation of Windows built in.

Table 2.1: Comparison between Linux, Microsoft Windows and Mac OS X

2.2 Open source productivity software

In the modern business world users communicate using e-mail attachments in the form of Microsoft Office file formats. The proprietary “doc” (Word) and “xls” (Excel) formats are about the most popular document file formats which are sent using e-mail or sharing methods. These file formats have led to Word and Excel becoming so popular that their names have become synonymous in their function – users do not call spreadsheet files a “spreadsheet”, but rather an “Excel” document. Word and Excel have essentially become almost household names. While some might argue that the software is the best solution for those uses, many alternatives exist which most people are not aware of.

The biggest problem with these file formats is the cost to purchase the software. Microsoft Office is available in Basic, Professional, Small Business and Education editions. Each one of these editions can effectively contribute to at least the third or more of a new desktop computer’s price¹. Education licences are cheaper (although only by a couple of hundred Rand), but it still contains only the most basic Office applications - Outlook (personal information and e-mail management), Word (word processor), Excel (spreadsheet) and

¹These prices can be found at Microsoft’s Office website: <http://office.microsoft.com/en-us/suites/FX101754511033.aspx>

PowerPoint (presentation software). For more advanced programs the user needs to buy more expensive licences; however, these features are rarely used by the average office user.

Microsoft Office Edition (2007 release)	Applications included in the specific edition:	Price (in US dollars)
Microsoft Office Ultimate Edition 2007	Access, Excel, Outlook 2007 with Business Contact Manager Update, PowerPoint, Publisher, Word.	\$ 679
Microsoft Office Professional Edition 2007	Access, Excel, Outlook 2007 with Business Contact Manager Update, PowerPoint, Publisher, Word, Small Business Accounting.	\$ 499
Microsoft Office Small Business Edition 2007	Excel, Outlook 2007 with Business Contact Manager Update, PowerPoint, Publisher, and Word.	\$ 449
Microsoft Office Standard Edition 2007	Excel, Outlook 2007, PowerPoint, and Word.	\$ 399
Microsoft Office Student and Teacher Edition 2007	Excel, Outlook 2007, PowerPoint, and Word.	\$ 149

Table 2.2: Prices of different Microsoft Office editions

It is sometimes hard to comprehend why Microsoft Office has to be so expensive, especially considering the lack of complexity in office applications, or the basic users' effective use of

the application's features. A word processor or spreadsheet is a piece of software that has been around since the seventies and can be considered the "power application" that rescued computers out of obscurity and into everyday use. Yet, only small changes to the Office suite have been made in each new release. For example, Office 2003 primarily focussed on the collaboration of Office documents. Users who worked together on a document could see who edited or added certain parts of the document. These features still do not add to the basic idea of a word processor or spreadsheet; it is still just an application to add words onto a page.

Not all users need sophisticated features like macros and collaboration features, but they feel they need to pay the thousand-plus rand to do some of the simplest tasks a computer can do - create documents. Creating a document is just about the simplest thing a modern computer can perform. Taking into account the processing power of a modern computer, and the highly complex calculations and processing that a modern computer game requires, the relative simplicity of an application creating a document becomes more apparent. One has to ask the question of whether the high cost is indeed justifiable for an average computer user.

Examples of open source office applications include Star Office, OpenOffice¹, and Thinkfree, etc. Each of these application suites has lower price points (or often free) than Microsoft Office. The open source office application with the best compatibility and ease of use compared to Microsoft Office is OpenOffice.org. It opens, edits and creates files which are completely compatible with Microsoft Office. It saves files in proprietary formats like "doc" or "xls" without any problems. The program is just as easy to use, with a very similar interface,² some argue that it is even better to use.³

¹ OpenOffice.org shares the same source code as StarOffice. StarOffice is a product made by Sun, which is a product users need to pay for. It has improved migration tools for enterprises moving from Microsoft Office.

² See figure 2.3.

³ Miller, Robyn. 2006. How OpenOffice 2.0 stacks up against Microsoft Office. [online] Available: http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci1174145,00.html [5 June 2007]

One great feature of these open source office applications is that they are available on most operating systems,¹ whereas Microsoft Office is only available on the two major proprietary operating systems - Microsoft Windows and Apple Mac OSX. For example, OpenOffice.org is available for Windows, Linux, Solaris, FreeBSD and Mac OSX²(using the X11 script application³). All these versions have the exact same functionality and are entirely compatible with one another. So even if someone wishes to use open source office applications, they can choose to do so without a conversion to an open source operating system like Linux, which can create initial usability problems. A switch of operating system can be considered the biggest barrier, seeing as many users require training to get used to the new interface.

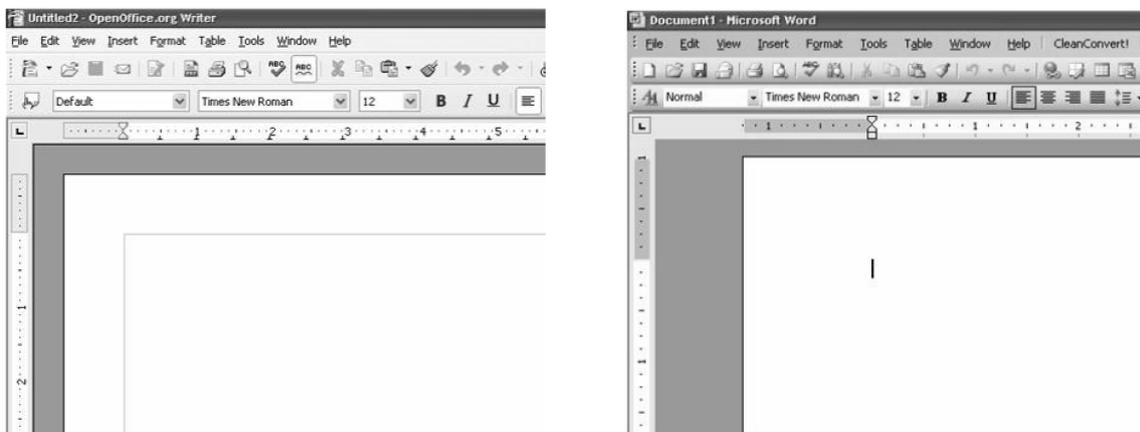


Fig 2.3: OpenOffice.org Writer and Microsoft Word Comparison

OpenOffice.org has the same basic applications as Microsoft Office, albeit with different names. These include Writer (a word processor),⁴ Calc (a spreadsheet application), Impress (a

¹ This is shown in figure diagrammatically in figure 2.4.

² All compatibility and system requirements for OpenOffice.org can be found at http://www.openoffice.org/dev_docs/source/sys_reqs_20.html

³ X11 is an add-on component for Mac OSX which is included in the Tiger edition, but previous releases of OSX needs to download the component. <http://www.apple.com/macosx/features/x11/>

⁴ A screenshot can be seen at figure 2.5.

presentation application), Base (a database application) and Math (a mathematical equation editor). Every application saves and opens files which are completely compatible with Microsoft Office. The user is given absolute freedom of which file format to use, and is made aware of specific advantages of using certain formats in terms of compatibility.

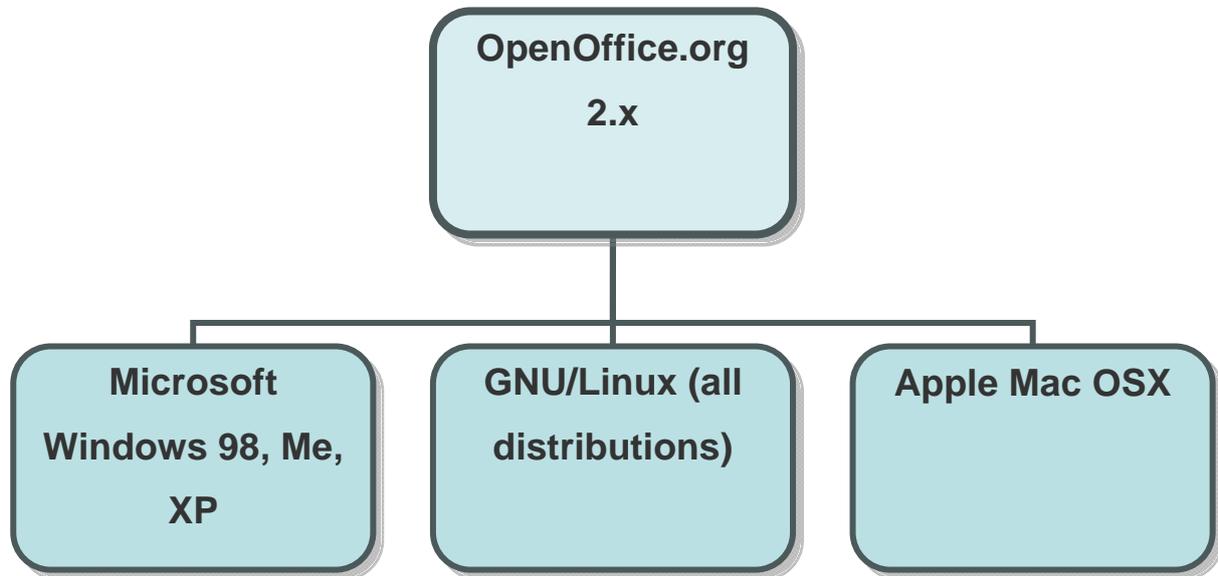


Fig 2.4: OpenOffice platform and document compatibility

The only application missing is an e-mail application similar to Microsoft Outlook. Users can then just use the open source and highly acclaimed Mozilla Thunderbird which many users protest is better than Microsoft Outlook for E-mail management. For calendar functionality, the Sunbird add-on for Thunderbird has been proved to be highly capable. This is Mozilla product (same creators of Firefox) and is described in more detail in the 3rd chapter.

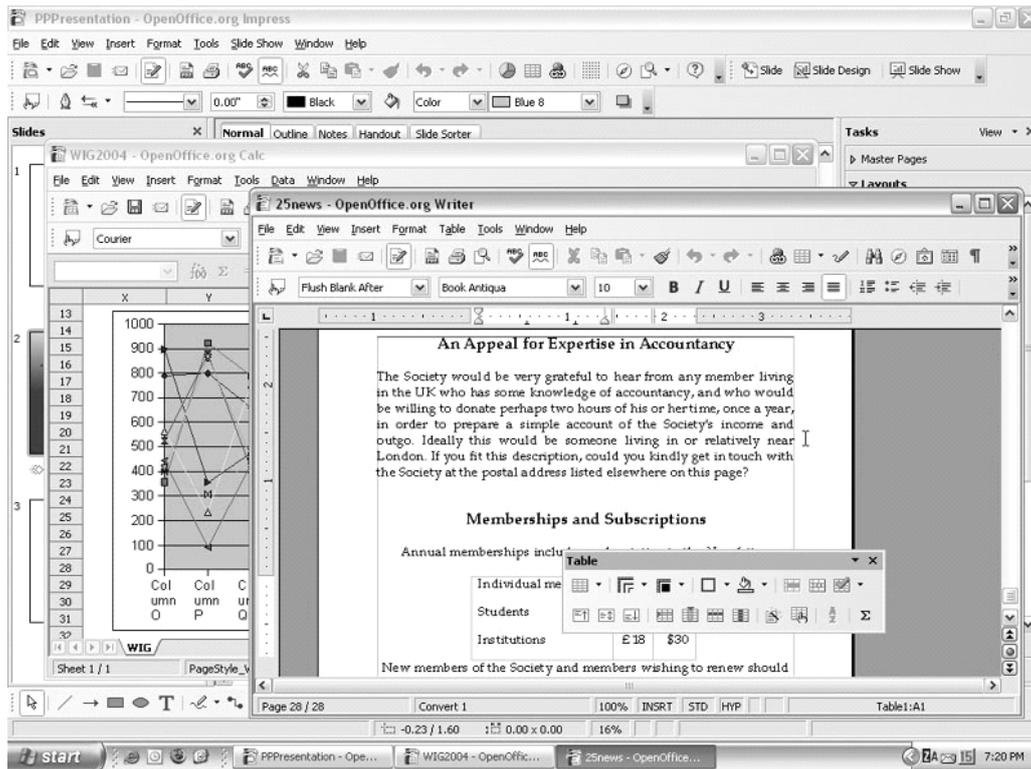


Fig 2.5: OpenOffice.org application on Windows XP

Many governments across the world are starting to implement an open source document strategy.¹ The other major advantage of using OpenOffice.org is that users can effortlessly start implementing the OpenDocument (“ODF”) format in all their office files. OpenDocument is short for “OASIS Open Document Format for Office Applications”². The major advantage of using this file format is that users can save a file, be it a word processor, spreadsheet or presentation file and can safely know their file can be viewed in most open source office applications. If users save files in this format, it prevents future readers of the file to be “locked in” by having to use proprietary software, which gives more freedom when upgrading future hardware and software.

¹ See Chapter 4 for a comparison of international open source initiatives.

² OpenDocument information and history page can be found at <http://en.wikipedia.org/wiki/OpenDocument>

This file format is especially important in academic applications, because research articles and theses need to be accessible to anyone, anywhere, and proprietary software like Microsoft Office is often financially out of reach for students. Because of massive industry support, Microsoft is under pressure to implement compatibility with a new open type file format in its Office suite. According to Wikipedia, Microsoft refuses for the moment to support the OpenDocument format, claiming that it “lacks independent implementations and important functionality”; although it has hinted that it may change its position in the future.¹ Microsoft is implementing its own open format, called Open XML, in its next release of Office, called Office 2007. Open XML will in all probability become the default file format.

Still problems do exist with using alternative productivity software. Sometimes file compatibility errors do occur, causing differences in the way a document looks or behaves. Even though compatibility is considered excellent, OpenOffice might not be powerful enough for “power users” of office applications. Even Sun (the creators of the source code behind StarOffice and OpenOffice) has conceded that StarOffice may not be sufficient for all types of users². In these cases, the extra cost of Microsoft Office might be well spent.

Even if the switch to an open source operating system can be a difficult decision, open source productivity software has hardly any disadvantages. Even if the government or education sectors decide not to use open source operating systems, open source office applications should be strongly considered, especially regarding price of proprietary solutions and the advantages of the new OpenDocument format.

¹ Even though Microsoft claims it lacks functionality, they have not been specific about the functions. In essence, OpenDocument is very similar to OpenXML (another open source document format), a file Microsoft is promoting for its next release of Office.

² Silver, Michael A; Rust, Bill. 2005. Choosing desktop Linux helps High Schools Buy More PCs. *Gartner Research*. pg. 3

2.2.1 Comparison between MS Office and OpenOffice.org

As previously explained, many computer users believe that the only way to use file formats like “doc” and “xls” is to have Microsoft Office installed. There exists many alternatives, most notably Open Office.

In this comparison the latest versions of this software are used.¹ They are:

- Microsoft Office 2007 Basic Edition (DSP licence included with new hardware)
- OpenOffice.org 2.3 (Provided as-is from OpenOffice.org website)

Table 2.3: Comparison between Microsoft Office and OpenOffice

	Microsoft Office 2007 	OpenOffice.org 2.3 
Price per licence:	About R1345 (much more if bought separately from hardware, see Office cost table)	Free ²
Platforms supported by the application:	Windows XP/2000 and Mac OSX.	Windows 98, XP, 2000; Mac OSX and Linux.
Applications included in suite:	Microsoft Outlook, Word, PowerPoint, Excel, InfoPath.	Writer (a word processor), Calc (a spreadsheet application), Impress (a presentation application),

¹ Information in this table is objective non-biased viewpoints of the writer.

² OpenOffice.org is based on the source code of StarOffice, a commercial product made by Sun Microsystems. Even though the feature set is very similar, StarOffice has built in tools for the migration from Microsoft Office.

	Microsoft Office 2007 	OpenOffice.org 2.3 
		Base (a database application) and Math (a mathematical equation editor).
Word processor functionality	Full functionality including macros. Considered the industry standard in word processing formats.	Full functionality including macros. Can open and save a lot of formats, including industry standards such as Microsoft Word.
Spreadsheet functionality	Excel included. Fully compatible with Lotus 123 files. Is considered the industry standard in spreadsheet file formats. No OpenDocument formats.	Calc included. Fully compatible with Lotus 123, Excel and OpenDocument formats.
Presentation functionality	PowerPoint is considered the industry standard.	Fully compatible with PowerPoint, but lacks in template design.
Database functionality	Microsoft Access is only included in the Professional and Small Business editions, not the base model.	Base is included.
PIM functionality	Excellent Outlook application. Industry standard in PIM and E-mail	Not included. Users should instead use Mozilla Thunderbird (E-mail) and

	Microsoft Office 2007 	OpenOffice.org 2.3 
	functionality. Is considered the industry standard in personal information managers. Handheld hardware such as phones and PDA's often interface with it.	Mozilla Calendar (PIM). ¹
File format compatibility	Compatible with all major file formats except OpenDocument ² . PDF ³ export is not supported.	Supports all major file formats including OpenDocument and PDF export.
System Requirements⁴	Microsoft Office can be a resource hungry application to run, and is only available on Windows and Mac.	OpenOffice.org has less rigorous requirements and is also available on Linux. PC:

¹ Firefox, Thunderbird and Calendar are all products of the Mozilla Organization. They are all open source, and based on open standards, ensuring interoperability. Available for download at: www.mozilla.com

² In May 2006 an unofficial extension for Microsoft Office was made available that enables users to save in OpenDocument format.

³ PDF is short for Portable Document Format. This allows formatting of a document to stay “locked” and allows it to be viewed on any compatible device.

⁴ System requirements are gathered from official websites of Microsoft Office and OpenOffice.org.

	Microsoft Office 2007 	OpenOffice.org 2.3 
	<p>PC: Windows XP/2000, Pentium 3 or better, 128MB RAM, 260MB Hard disk space, SVGA resolution.</p> <p>Mac: Mac OSX 10.2.8 or later, G3 processor, 256MB RAM, 450MB Hard disk space, XGA resolution.</p>	<p>Windows 98, Me, 2000 or XP. 128MB RAM, 200 MB disk space, SVGA resolution.</p> <p>Mac: Mac OSX 10.3 or later, G3 processor, 256MB RAM, 350MB Hard disk space, SVGA 32bit resolution.</p> <p>Linux: Pentium class processor or later, 128MB RAM, 200 MB hard disk space, SVGA resolution.</p>

Table 2.3: Comparison between Microsoft Office and OpenOffice

2.2.2 E-Mail and messaging software

Firstly, distinction should be made between E-Mail and messaging software. Many articles refer to the instant messaging brethren as “messaging software” – this is incorrect, because modern implementations of a combination e-mail and instant messengers haven’t been successful. Mail software is specialized towards the handling of e-mail messages. Messaging

software can also focus on e-mail messaging but primarily focuses on instant messaging, and recently, voice over IP features.¹

a) E-mail software for the server:

E-mail server software must be reliable and fully scalable in order to meet growing demands and new features. The e-mail server software landscape is governed by proprietary software such as Microsoft Exchange and Novell GroupWise. The advantages of these products are that they combine most features in an easy to implement system, whereas open source products tend to be more broken up into pieces. For example, SPAM filtering has to be done in a separate gateway from the mail server.

Woods and Guliani states that the following features are needed from an enterprise level e-mail server²:

- All major protocols should be easily applied if needed.³ These include POP3, IMAP, and SMTP, etc.
- Full support for enterprise wide address books, and easy “name checking” features.
- Some enterprise level e-mail servers grant users the ability to access their e-mail through a web browser. This gives a similar experience to using a service like Gmail or Hotmail⁴. Mobile devices can also access modified XHTML versions of these services should their phones not have e-mail clients.

¹ Recent messenger applications are starting to implement voice features. This includes AOL IM, Yahoo Messenger and Windows Live (previously known as MSN messenger).

² Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly.. pg 161.

³ In enterprises it might be necessary to be able to access more than one e-mail protocol. Whereas POP3 might be preferred from a desktop machine, IMAP is beneficial for use on mobile devices which tend to have high internet costs. SMTP is a secure capable sending protocol.

⁴ Gmail is a free e-mail service provided by Google. [online] Available: <http://mail.gmail.com>. Hotmail is a similar free service provided by Microsoft. <http://www.hotmail.com> [10 October 2007]

- Enterprise level e-mail servers enable users to use more advanced PIM features. Even though these features are not necessary for an e-mail server, they prove beneficial to users of Outlook 2003 (or 2007). For example, shared calendars and collaborative features can be enabled when using Exchange type servers.
- Content scanner support is very important. Users can be bombarded by SPAM and e-mail viruses daily and it is also the e-mail server's responsibility to help reduce unwanted e-mails. Content scanners are separate applications that act as a plug-in for e-mail servers.

The two best known open source e-mail servers are Qmail and Postfix. Each of these products has specific strengths and weaknesses. However, these products are not nearly as easy to implement as Exchange or GroupWise. Additionally, with the dominance of Outlook on the client side, Exchange looks all the more attractive due to its additional features. Again, it is up to the IT department to decide whether they are up to the job of administering and supporting these products.¹

b) E-mail software for the desktop client:

Client e-mail software entails the effective sending and receiving of e-mail but also organizing. The most popular e-mail software is Microsoft Outlook, bundled in the Microsoft Office suite. It cannot normally be purchased separately from Office².

Microsoft Outlook is, however, classified as personal information management software. PIM software focuses on time management software. Microsoft Outlook includes calendar, task list, journal and contacts components. Each of these integrates tightly to form a well designed piece of PIM software. Many would argue that one of Microsoft Outlook's greatest abilities is the fact that so many plug-ins are written for the software. For example, modern

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 161

² The only instance where Outlook can be purchased separately from the Office suite is where it is included with Microsoft PocketPC / Windows Mobile software. These are embedded versions of Windows which are built into smartphones and PDA's.

“intelligent” cellular phones (called smart phones) can rely heavily on Outlook for synchronization features.

There are a number of features which are expected from e-mail client software¹:

- Again, support for multiple e-mail protocols. These include IMAP, POP3, HTTP and Exchange profiles.
- An easy to use interface which does not confuse the user. These include multiple panes which allow the user to preview messages without opening them.
- Easy management of e-mails. These can include rules which place certain e-mails into specified folders.
- Searching of e-mails should work efficiently and be easy to use. This should search headers, body text and certain dates.
- Good industry standard support of all variations of address books. These address books integrate fully into the e-mail client. For example, the user should only type the first few letters of an e-mail address and then autocomplete should give suggestions.
- E-mail messages must be composed with a versatile word processor type system, with inline spell checking. Users should have complete control over the look of the e-mail message. Naturally e-mails should have full HTML support.
- Third party applications should be able to access the e-mail client securely. These can include content scanners and instant messaging applications.

Microsoft also bundles free e-mail software with Windows called Outlook Express. Outlook express primarily focuses on e-mail, with limited address book / contact storage capability. There exists a variety of open source e-mail software, many with very powerful features comparable to Microsoft Outlook. These include most notably Mozilla Thunderbird and

¹ Adapted from Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly.. pg. 167.

Novell Evolution. Novell Evolution is also a personal information manager. It is the most feature complete open source e-mail client with PIM functionality as well.

Mozilla Thunderbird is closer in functionality to Microsoft Outlook Express. It does, however, support third party add-ons which expands its functionality. ¹These include Mozilla Sunbird which adds PIM features. Mozilla software is always of high quality and it is a very viable replacement for Outlook Express.

2.3 Web browsers

This is one area where open source applications have truly shone. Microsoft has a history of unfair competition when it comes to web browsers. In the late 90's, Netscape Navigator was the most popular web browser. It was an era when web browsers had to be paid for. Microsoft decided to bundle their own Internet Explorer into Windows. Even though it was not necessarily a better product, it started gaining market share, because users did not see the reason to pay for a product whose functionality they can have for free. This led to the big Microsoft monopoly case of the late 90's and early 2000's.

Microsoft continued to improve their product in order to comply with standards set by bodies like the World Wide Web consortium². Internet Explorer became the prominent web browser and the de facto standard. IE came under criticism of its lack of security – a shortcoming that was in due part because of an ideological vision by Bill Gates³.

¹ Third party plug ins for Mozilla applications can create additional support problems. Once again, plug ins need to be assessed to ensure that problems do not later occur.

² The World Wide Web consortium (or “W3C”) is a web standards governing body that creates standards in order to ensure interoperability of the web on a variety of devices and platform.
<http://www.w3c.org>

³ To read more about the integration of the internet into Windows, read this preliminary beta review of Windows XP before it was released:
http://www.winsupersite.com/reviews/windowsxp_beta2.asp

Microsoft had a vision long ago of the computer interface being tightly integrated with internet content – this led to the integration of Internet Explorer into the kernel of Windows. Therefore, if IE crashed, so did the computer. IE became one of the primary targets of malicious software which led to many security concerns – the greatest shortcoming of IE. It can be a very wise shift to move to other browsers, if only for the security benefits. But browsers like Firefox and Opera both have support for newer standard in websites and both have phishing filters and tabbed browsing¹, a feature that many users can't live without.

Comparison of alternative browsers:

a) *Mozilla Firefox:*

The same creators of Netscape Navigator created the Mozilla foundation, who wanted to create a secure web browser which was separate from the operating system (which did not lead to complete crashes of the computer). It was initially part of a bundle called the Mozilla Application Suite which consisted of Firefox and Thunderbird (an e-mail client). Over time, Firefox became the Mozilla foundation's main focus, and decided to split the Mozilla application into separate internet browsers and e-mail clients like it is known today.

Firefox has had tremendous success², in part due to its interface, which is customizable and extensions, which add functionality to the browser interface. These extensions can be very useful depending on the application. For example, Google makes a Firefox extension that lets AdSense advertisers know how much money their website is making. Firefox is available on different operating systems including Windows, MacOSX and Linux. It is an commendable

¹ Tabbed browsing is an alternative method to navigate between multiple open web pages. Instead of each website taking up an additional window on the taskbar, Firefox and Opera make it possible to switch quickly between multiple open web pages. Microsoft only addressed this type of functionality in its Internet Explorer 7 release, a good few years after Opera and Firefox had this functionality.

² On October 19, 2005, over 100 million copies of Firefox have been downloaded. This makes it one of the most successful open source applications of all time.

open source piece of software released under the Mozilla licence. It is a prime example of a mature open source product, into which we will go into more detail later.

b) Opera Web Browser

Although not open source, Opera is a free web browser. Opera was one of the last browser for which people had to pay, but in 2005 it also became a free browser, similar to Internet Explorer and Firefox. Its features include faster downloads and rendering of web pages and a customizable interface. It also uses tabbed browsing. Its main difference is the integration of a mailing client right within the browser interface. This can be confusing as many users prefer a separation between e-mails and internet browsing. It is not a very useful feature, but some users might find it beneficial in an environment where not a lot of application switching occurs. Users can just ignore the e-mail feature if they choose to do so.

One area where it might be beneficial is on home computers which are primarily used for web browsing and e-mail. Opera is available on a variety of platforms, including PC's, Pocket PC, and Symbian and as a Java midlet called Opera Mini for cell phone use. The advantage of the Opera browser on these portable devices is that web pages accessed by the user are delivered to the device through Opera's own company servers. These servers reformat web pages so that they correctly display on smaller screens. In almost all cases the result are a great improvement over using traditional WAP browsers and Pocket Internet Explorer. Unfortunately, the Pocket PC version of Opera, called Opera Mobile, is not a free product.

On the desktop PC front, Opera is available for Windows, Mac OSX and Linux¹ (it is also available for a variety of other operating systems).

2.4 Antivirus

This is a hotly contested area, so options available to organizations are constantly changing. This is perhaps the area of software where reputation is the paramount factor in using a

¹ These different versions can be viewed at: <http://www.opera.com/download/>.

specific solution. Many free antivirus software packages exist, but the big names in antivirus are not free or open source. Seeing as security becomes more important every day, it certainly makes sense that many organizations automatically go for big names like Symantec or McAfee.

What administrators should keep in mind is that in many cases these popular antivirus applications are not necessarily the best one for the job. Just a casual read through some antivirus group tests (like on CNET.com and PC Magazine)¹ will prove that point.

What does a user expect from an antivirus scanner?

- Reliability to prevent viruses from infecting the system
- Regular definition updates whenever possible in order to prevent new infections
- Ease of use – in fact, it must be relatively out of sight except when user intervention is needed. This is one of the most important reasons why some users choose not to run antivirus software at all.
- Full integration with e-mail clients – the software must have subject header and text analysis in order to recognize patterns which could indicate viruses. Attachments should be scanned before being opened. Some software companies have started charging extra for junk mail prevention software².
- Minimal performance degradation with use. Many antivirus software applications use an excess amount of system resources to run. The software should be a transparent part of using a computer, not something the user should be constantly made aware of. Symantec security software is currently under fire for this very reason.

Even though all operating systems can be attacked by viruses theoretically, virus creators tend to focus on Windows because of its popularity. Linux and Mac OSX viruses do exist,

¹ These computer related web sites can be found at: www.pcmag.com and www.cnet.com.

² Symantec differentiates between its Antivirus and Spam/Junkmail prevention software (called Norton Internet Security). Its normal antivirus software only scans the attachments in e-mail where the spam/junkmail software can recognize patterns and have features like black and whitelisting.

but in many cases they are more a “proof of concept” creation. If real threats do appear, updates are available quickly. Essentially, antivirus software can be considered compulsory for Windows PC’s.

2.5 Choosing between packages: the open source maturity model

Woods and Guliani¹ created a maturity model which creates scores based on certain key aspects of open source software. These factors are divided into product, usage and integration criteria.

Product Criteria:

Age, platform support, momentum, popularity and design quality.

Usage criteria:

Setup cost, Usage cost, end user support.

Integration criteria:

Modularity, collaboration with other products, standards compliance, developer support.

Example of the maturity model applied to Firefox:

We can apply these criteria to Mozilla Firefox. It is considered by many as the best internet browser; its interface is an update to the users of Internet Explorer because of tabbed browsing² and it is easy to install on almost any machine or operating system. It is free, updates are frequent, and it is considered the most secure browser. It collaborates with a variety of applications and services by the use of “extensions” and it supports the latest web standards, unlike Internet Explorer.

Product Criteria:

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O’Reilly. pg. 42

² Microsoft did release tabbed browsing capabilities in Internet Explorer 7 in 2007. This is however years after the feature has been available on alternative browsers.

Firefox was released as version 1.0 in November 2004. Because of the quick development cycle of OSS, the product has gone through many product revisions and is constantly under development. It is supported on Linux, Windows and Mac OS X. In October 2007, Mozilla announced that they will also be releasing mobile versions of the browser in 2008.¹ In October 2007 it reached a market share of almost 15% of the browser market share, normally at the expense of Internet Explorer, which is installed by default on all Windows machines.²

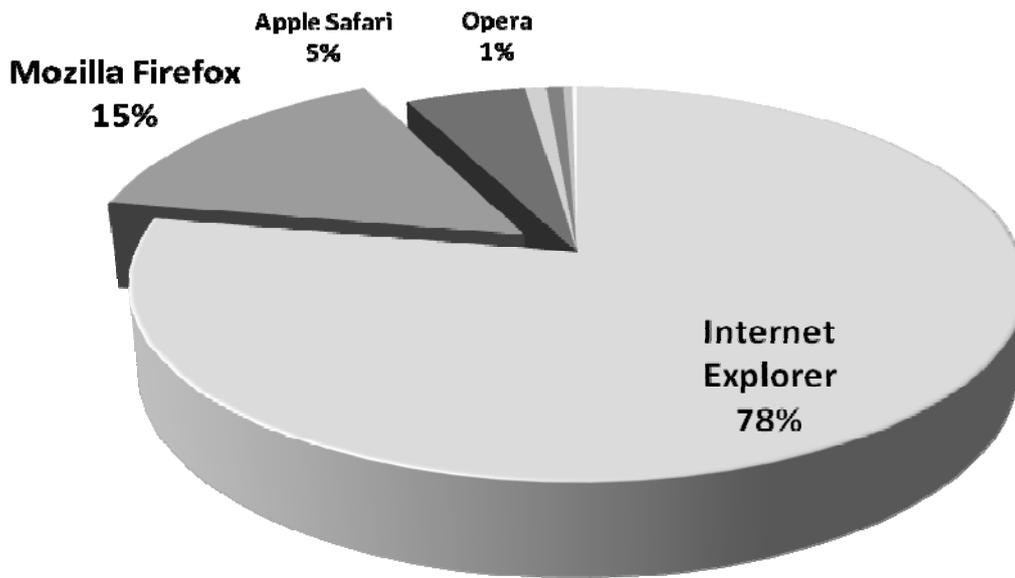


Fig 2.6: Mozilla Firefox commands a almost 15% usage share in percentage of total active browsers³

¹ This is an important move for Mozilla, seeing as people are becoming more and more reliant on their mobile phones for internet access. This version of Firefox will run on Windows Mobile, Symbian and Linux operating systems, further enhancing the product range as a multiplatform application.

Gorhing, Nancy. 2007. *Mobile Firefox likely to work on Symbian*. PC World. [online] Available: <http://www.pcworld.com/article/id,138375-c,mozilla/article.html> [14 October 2007]

² Net Applications. 2007. *Top Browser market share trend for October 2007*. [online] Available: <http://marketshare.hitslink.com/report.aspx?qprid=3> [14 October 2007]. As seen in figure 2.6.

³ Net Applications. 2007. *Top Browser market share trend for October 2007*. [online] Available: <http://marketshare.hitslink.com/report.aspx?qprid=3> [14 October 2007]

Firefox is considered very easy to use and has won a variety of usability awards from major technology publications such as PC Magazine, CNET and PC World. It is also consistently chosen as a winner of product of the year by these publications.¹

Usage criteria:

Firefox is typically very easy to install with next to nothing installation cost. It automatically imports bookmarks and settings from the default browser, requiring almost no user intervention. It is considered very stable, even though no formal support exists. Most errors and bugs are normally fixed the moment they are noticed by an very proactive community of developers.

Integration criteria:

Users can customized Firefox through the user of plugins which can alter the interface, add new tools and completely customize the browsing experience. Mozilla also has an extensive collaboration agreement with Google in order to further improve the product. For example, the browser gets an anti-phishing update file about every thirty minutes from Google in order to protect the user.²

Because of all these reasons, it is the second most popular browser behind Internet Explorer and is still experiencing immense growth. The reason why Internet Explorer is the number one browser on the market is not because it is the best product, but because it is preinstalled on any Windows machine.³

¹ Mozilla. 2007. *Firefox Awards*. [online] Available: <http://www.mozilla.com/en-US/press/awards.html>. [14 October 2007]

² Mozilla Organisation. 2007. Phishing Protection: Design Documentation. [online] Available: http://wiki.mozilla.org/index.php?title=Phishing_Protection:_Design_Documentation&oldid=46996#Overview [14 October 2007]

³ Microsoft did release certain European editions of Windows without Internet Explorer and Media Player because of anti-competitive lawsuits. However, these releases proved unpopular due to reduced functionality. This was further crippled by the fact that modern releases of Windows were heavily reliant on a Internet Explorer “backbone” as a interface to many key OS components.

2.6 The advantages and disadvantages of using open source software

This section will focus on the specific benefits that South Africa might gather from using open source software. First, one has to consider the specific advantages and disadvantages of open source from the end-user level and see which of these aspects need to be emphasized when considering it on a larger scale. Even though this section does not focus on Linux only, it is a key consideration in an open source implementation strategy, since the operating system can contribute to almost a third of the price of a baseline computer system.

This section will also focus on productivity software which is to be used as an alternative to Microsoft Office. On a baseline computer system, Microsoft Office 2007 trades for around R1200 for the basic and education-licensed version (OEM licence); this contributes another third to the price of a new computer. In the past, Linux has proven to be adopted quickly by certain users depending on field of endeavour or skill level. However, in this situation we have to presume we are adopting the most user friendly distribution for implementation into the government sector. Administrators responsible for server or enterprise installation will obviously choose the distribution with the best suited feature set, after a thorough needs analysis.

Advantages of using open source on a desktop computer:

a) Price

As stated earlier, using open source solutions drops the price of a desktop or notebook dramatically. Using a Linux operating system and OpenOffice productivity software instead of Microsoft Windows XP and Office 2007, can save more than R3000 on a new desktop computer (about \$300 for OEM versions). However, many computer manufacturers insist on installing Windows on their machines, without giving the user any choice in the matter.

Microsoft have however shifted away from this system seeing as any security hole in IE could result in the entire OS being rendered almost useless. A good example of this is the “Add/Remove Programs” function which was still HTML based before XP Service Pack 2.

However, this is slowly changing as major computer manufacturers such as Dell and IBM are starting to make Linux available as an alternative on consumer machines.¹

“Linux and open source software are unlike any competition that Microsoft has ever faced. Microsoft is now perceived as expensive, at least in preliminary discussions regarding Linux”². It can be seen as a viable alternative to proprietary solutions, provided it is judged against the same criteria, including total cost of ownership³. Therefore, although cost is a tremendous advantage, TCO can be more than just licensing fees.

b) Added security

Microsoft software is traditionally plagued with bugs and security vulnerabilities. This does not necessarily mean the software is of a lower quality. Hackers and malicious software writers just focus on Windows because it is so much more common than Linux or Mac OS X. Linux support groups also tend to be faster in bringing out security updates and patches due to a pro-active support community, resulting in a stellar reputation compared to Windows. This is further emphasized by the fact that Microsoft releases patches and “hot fixes” to Windows in schedules (normally the second Tuesday of every month). This can sometimes prove to be too infrequent for certain infections or malicious attacks.

c) Stability and Hardware Longevity

As mentioned before, Linux updates occur more frequently than Windows, but these updates are not just for bugs. If a new type of hardware needs to be supported, the Linux community is very quick to support that hardware or device. Usually Linux software is faster to run than

¹ In April 2007 Dell announced to make Ubuntu available on its desktop and notebook machines.

Report [online] Available:

<http://www.dell.com/content/topics/global.aspx/alliances/en/linux?c=us&cs=555&l=en&s=biz> [10 October 2007]

² Smith, D.M. 2003. A Look at Alternatives to Microsoft. *Gartner Research*. pg. 2

³ Di Maio, Andrea. 2005. Look Beyond TCO to Judge Open source Software in Government. *Gartner Research*. pg. 2

Windows. Modern distributions can still be run on an Intel 80486 processor with 16 megabytes of RAM. Modern releases of Microsoft Windows XP need at least 400MHz processors and 128 megabytes to run at all¹.

These requirements increase exponentially with every release of Windows. Vista will require a 1GHz processor, 512MB RAM and preferably a modern 3D-enabled graphics card to further enhance the user experience. Seeing as Linux can run on lower end (and older) machines, organizations do not have such stringent hardware refresh periods as with Windows. Windows machines require a typical hardware refresh every three to four years, whereas with a typical Linux system a hardware refresh is only needed every six to eight years.²

d) Productivity future-proofing

Modern versions of OpenOffice and StarOffice support an open source document format called OpenDocument. This file format was created because of pressure from consumers who needed a word processor file format that did not need Microsoft Office to open the file. When a user saves a word processor document in Microsoft Word's proprietary software (called a "doc" file), the file cannot be opened by any other software than Microsoft Word. So in effect, saving a document in the "doc" means that anyone, even if it is ten years in the future, will supposedly need Microsoft Office software. Industry support for the OpenDocument format means that the file can be opened and read by anyone, on any productivity software. This initiative is especially popular with government groups in the United States. Using this format will also save money on implementation costs of computers in emerging nations, and

¹ However, at this hardware level Windows XP is barely usable. It is not recommended to run Windows XP on such low level hardware. This is also true for the minimum specifications of Windows Vista.

² California Office of Government Commerce. 2007. *Open Source Software Trials in Government Final Report*. [online] Available: <http://www.arb.ca.gov/oss/articles/report-v8d.pdf> [10 October 2007]

therefore the information and documents can be read without spending additional money on productivity software. OpenOffice also has the added benefit of being able to export to PDF format which greatly increases accessibility of documents. PDF documents also have security benefits which are ideal for government application.

e) More improved and creative software

The attraction of the open source model is that if open source intellectual property can help solve a specific problem or bring a new service or product to market, there are few entry obstacles¹. This is due to the relative ease with which software writers can alter and change code, which is fully permitted with open source software. However, in some cases the altered code should be returned to the original authors to be certified². This also means that many software errors or bugs can be corrected in-house if there are specialists with the required skills.

Disadvantages of using open source on a desktop computer:

a) Costs:

Even though licensing fees are in most cases free, total cost of ownership should be considered. “TCO takes into account the costs, not only for hardware and software, but also for management, product support, migration, personnel, training and possible downtime.”³

b) Usability Issues

The single biggest boundary for the adoption of open source is training people to use the new interface. The average employee is used to operating Windows, which in most cases has very

¹ Drakos, N; Driver, M. 2000. Debunking Open source Myths: Origins and Players. *Gartner Research*. pg. 2

² This is called the “Cathedral” method of development explained in chapter 1.

³ Krammer, Mika Yamamoto. 2005. Costs and Benefits Still Favor Windows Over Linux Among Midsize Businesses. *Gartner Research*. pg. 2

few corresponding visual cues to Linux (regardless of whether it is Gnome or KDE interfaces). Also, Linux's more advanced system features are only accessible by command prompt, which is not intuitive to users who are used to the graphical user interface of Microsoft Windows or Macintosh OSX.

c) Compatibility Issues

Microsoft Windows, Macintosh OSX and Linux are all incompatible with one another on a software application level. If a piece of software is written for Windows, the same piece will not run in Linux, unless specifically coded to do so, which happens rarely. In dire situations, emulators can be used to run software that was written for different operating systems. Examples of this include WINE for Linux (a Windows emulator) or VMware for Windows (which emulates a variety of hardware platforms). However, these emulators can be very expensive to implement, and therefore is only a viable option if used in technical support scenarios.¹ Modern processor hardware is especially adept at using virtualization software.²

One aspect which is not problematic when using Linux is network compatibility. Most networks rely on a standard TCP/IP connection, with which all the main operating systems comply. Because of Linux's high stability and reliability, it is very popular in server environments.

¹ VMWare has a variety of options available to end users. Administrators will use full versions of VMWare workstation which can "create" virtual machines on a host computer. However, there are free versions available for download which do not have the option to create virtual machines. These free versions instead use pre-made virtual machines which can be created by any owner of VMWare workstation. [online] Available: <http://www.vmware.com/products/ws/> [10 October 2007]

² Modern editions of Intel and AMD dual core processors have hardware-level support for virtualization, which helps with running multiple operating systems. This is done through software like VMWare (Windows and Linux) and Parallels (Apple Mac).

2.7 Conclusion

Seeing as open source carries a few key advantages (and challenges), it requires extensive analysis and all alternatives should be considered. Despite its many peculiarities, open source can be very cost effective to implement especially if administered correctly, and if the right skills are available within the support framework of the organization.

When looking at alternatives in open source applications, it is important to realize that open source does not necessarily negate a lower overall cost. This overall cost is referred to as total cost of ownership, and is dependent on a number of different factors which change depending on the choice of software being analysed. For example, in many scenarios OSS has no licensing cost, but might have considerable differences in support and staff costs.

Once a proper feature analysis of alternatives exists, TCO and Return on Investment needs to be done in order to get proper backing from management level within a organization.

Chapter 3

Cost Considerations - *Total Cost of Ownership and Return on Investment*

The biggest reason behind Linux's success rate in corporate server market is the perceived lower TCO. Total cost of ownership can include initial purchase price, repairs, maintenance, upgrades, service and support, networking, security, user training, and software licensing, among other expenses.¹ TCO is not the primary cost concern with many companies requiring a project return on investment in order to make an informed decision. Open source software is particularly difficult to justify using ROI, and therefore other aspects of the software need to be highlighted in proposals. Even though these statistics are shown in many situations, it is always necessary to analyse each specific implementation thoroughly.

¹ Wikipedia. 2006. *TCO*. [online] Available: http://en.wikipedia.org/wiki/Total_cost_of_ownership
[2 October 2006]

Proprietary software is made more affordable through the use of “CAL” (customer access licences). One of Linux’s hyped claims to fame is that it always leads to a lower total cost of ownership in any organization. This is not always true, and it is dependent on a variety of factors. Most people think that the lower or free licensing cost leads to savings. In order to analyse this, one has to compare open source solutions with proprietary solutions.

3.1 Forms of ownership

In the proprietary market, a few different forms of ownership exist. The most readily available form of enterprise licensing is “CAL”.

CAL’s are available in a variety of forms. Instead of purchasing a licence for every operating system and application, businesses purchase CAL variations. Each of these options is available because each organization or group has different networks. The different licensing options include:

a) *Per Seat*

Once the server product is purchased, it has to be licensed to work with a number of unique machines. For example, if a company uses 10 workstation PC’s and 1 server machine, a 10 licence product needs to be purchased. Figure 3.1 diagrammatically portrays this.

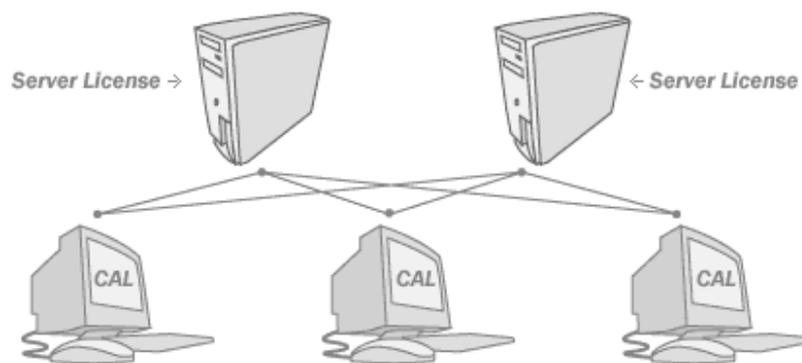


Fig 3.1: Microsoft “per-seat” CAL licence¹

¹ These diagrams can be found from Microsoft’s licensing information page.

http://www.microsoft.com/resources/sam/lic_cal.msp

b) Per Server

This licence stipulates the number of unique machines that may simultaneously connect or access the server. For example, if a 5 CAL Windows Server licence is purchased, only 5 unique workstations may simultaneously access services offered by the server. These may include document management services such as Microsoft SharePoint. If another server is connected to the network, all the unique devices require another “per server” CAL in order to also access the services on that server, as can be seen in figure 3.2.

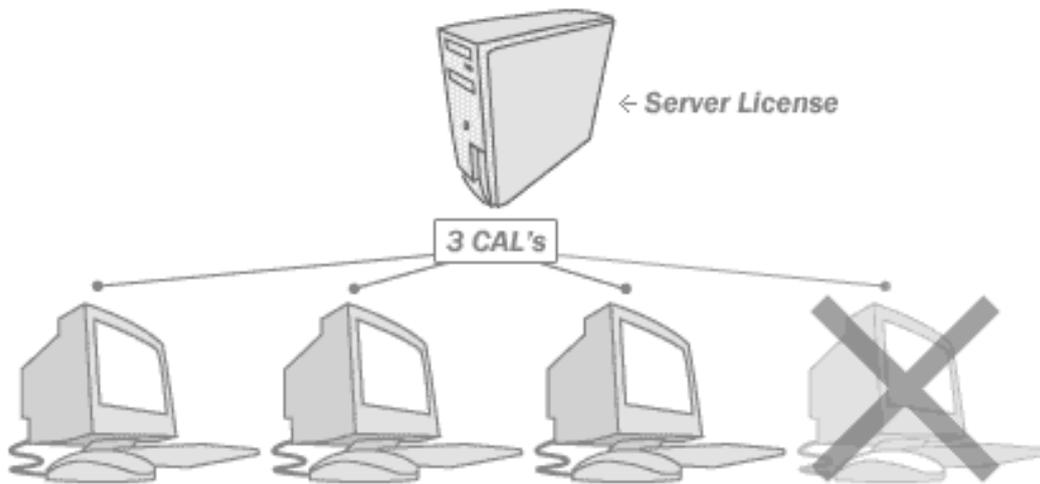


Fig 3.2: Microsoft “per-server” CAL licence

c) Per Processor

This is only available in some server products. The licence is purchased depending on the number of processors used by the server (multiple processors in a server is not uncommon). An unlimited number of users can connect to the server, regardless from where the connection comes.

d) Per Mailbox

Users of Exchange server need to purchase either a "per seat" or "per mailbox" licence, depending on the number of unique e-mail users.

3.2 What is the cost factors involved?

In order to make a clear analysis, it is necessary to identify all cost factors of an organization wide installation, at server and desktop level. A number of scenarios must also be created, seeing as different users require a different software bundle, some of which require proprietary software such as MS Windows or Mac OS X.

In most migratory cases, businesses are shifting from Microsoft Windows to a commercial version of Linux.¹ However, there are more variables to look at. Even though Windows is the dominant operating system on the desktop front, in the server market it is a different story. In the server market the dominant operating systems are UNIX, Microsoft Windows Server and Sun Solaris.

Each of these operating systems has specific operating methods and standards. Out of all of these UNIX is the closest to Linux. Perhaps the most significant cost is the retraining or recruitment of Linux specialists, however, ex-UNIX administrators can easily learn to use Linux because they share the same interface traits. The biggest costs involved with migration are hardware upgrades, software switching, migration costs, management costs and support costs.² Each of these cost factors is very volatile and flexible depending on the existing software and current hardware used. These factors are shown graphically in figure 3.3.

¹ This is because modern releases of Linux are focused on Windows users as its primary audience.

² Margulius, Dave. 2003. The Real cost of switching to Linux” *Infoworld*. [online] Available: http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html [10 October 2007]

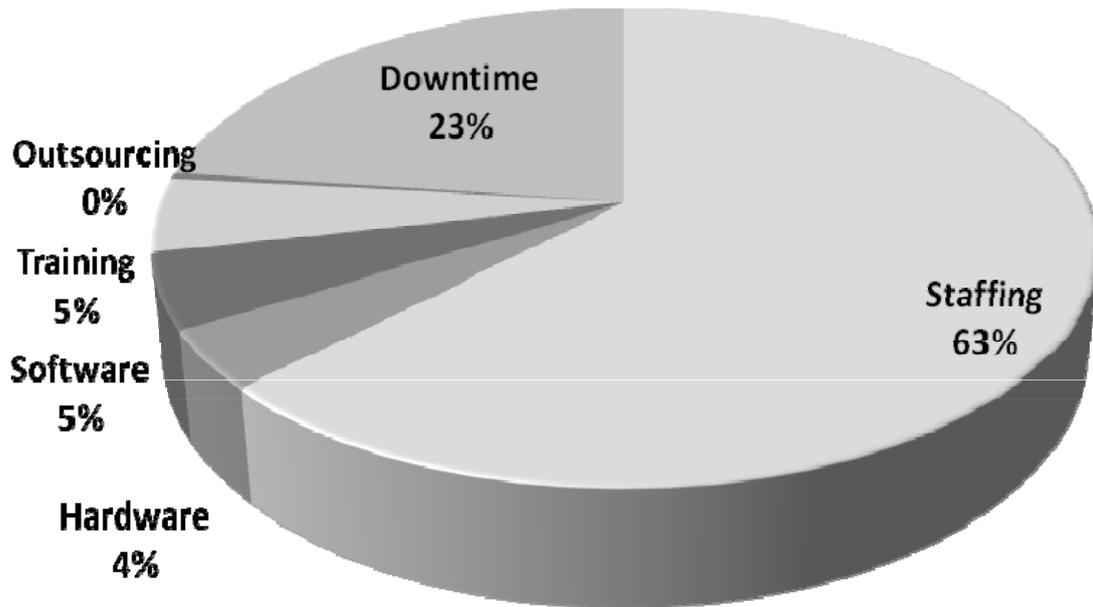


Fig 3.3: Breakdown of TCO for IT infrastructure workloads over a five-year period.¹

a) **Hardware costs**

One of the greatest advantages of Linux is that it is less strenuous on hardware requirements (depending on the distribution). In most cases Windows is re-released every 3 years where Linux is almost continually upgraded (e.g. Ubuntu²). With each Windows update, the increases in hardware requirements almost always require a hardware upgrade or replacement.³

Linux is updated much more regularly and does not necessarily require hardware upgrades necessarily. The main reasons for Linux upgrades are to support the latest hardware. For

¹ Microsoft. 2006. Get the Facts: Total Cost of Ownership. *Microsoft Corporation*. pg. 2

² Ubuntu, a Linux distribution made by Canonical is updated every six months. This results in version numbers which indicate the year and month the version was released. For example, the latest “Edgy Eft” version, called 6.10 – indicates a October 2006 release.

³ For example, Windows Vista requires extensive 3D hardware within a system. Seeing as modern notebooks cannot have their graphics cards upgraded, it will result in replacement in the case of notebooks. The graphics cards also need to be reasonably up to date.

example, if Intel brings out a new multi-core processor in 2007, Linux will be very quick to support the new architecture. Therefore, the fact that Linux runs well on older hardware does not mean that it does not take advantage of new technologies or architectures.

Almost all current computer hardware can support Linux¹, whereas Windows requires x86 based hardware, Solaris requires Sun SPARC hardware with expensive RISC based processors and Mac OS X requires Apple hardware (either PowerPC or Intel). A good example of the versatility of Linux is the ability for the kernel to run on an iPod music player, because it contains a processor, memory and hard drive.

Great savings can be made at a hardware level during migration to Linux. If the previous server was an expensive Sun SPARC based server with RISC based processors, a switch can be made to consumer-level Intel or AMD x86 processors. This dramatically decreases costs. An entry level Sun UltraSPARC based machine starts at around \$3100 where a normal industry standard machine can be used from around \$500. The Sun machine will obviously have slightly improved performance depending on the application, but at this cost savings, multiple x86 machines can be bought instead of one UltraSPARC machine. “Instead of having a super horse you have a team of horses — you don’t have to have this genetic [RISC] wonder.”²

Perhaps the greatest example of this is the online travelling agency Orbitz which migrated to Linux in 2003 from their previous UltraSPARC based system. This system is responsible for synchronizing and downloading air fares and plans. In the previous system a total of 168 Sun SPARC processors were needed to run all the business critical web services. After the migration, 100 much cheaper Intel processors running Linux were able to double the performance of the previous high-end system³.

Google also benefited greatly from lowered costs due to less expensive hardware. Part of Google’s initial success was the use of “off the rack” computer components instead of a high-

¹ This can include Intel (x86 and x86 64bit), Apple (PowerPC and Intel) and Sun SPARC hardware.

² Margulius, Dave. 2003. The Real cost of switching to Linux” *Infoworld*. [online] Available: http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html [10 October 2007]

³ Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* pg. 86

end system from IBM. They needed a lightning fast system which could process search results for a growing customer base. Instead of paying \$800,000 on a high-end IBM “super-machine”, they bought standard server components for \$250,000 which “provided comparable processing power and several times more disk storage”.¹

The secret was in using Linux instead of Windows and employing these computers to work as parallel processing units. These machines were certainly not as reliable (hardware quality wise) as an IBM dedicated server, but the components were so much cheaper that replacement could happen every two to three years. The Linux software they used could just bypass computers that have failed through a well designed redundancy system. The parallel running machines proved more than adequate for its web crawlers to function. Google still relies on this same method.²

Even on the consumer desktop front, the switch to Linux makes financial sense when purchasing new hardware. A 2003 Gartner study found that an average of \$80 can be saved on hardware acquisition and \$74 per user per year if a switch is made to StarOffice.³

Because Linux is compatible with almost any hardware architecture, the IT administrator is in much more control when it comes to upgrades. Machines are not “forced” into an upgrade every 3 or 4 years as is the case with Microsoft Windows. As stated by Chad Robinson: “With Linux, you control your own upgrade cycle”⁴

¹ Vise, David A. 2005. *The Google Story: Inside the Hottest Business, Media and Technology Success of our Time*. Delacorte Press.. pg. 79

² Vise, David A. 2005. *The Google Story: Inside the Hottest Business, Media and Technology Success of our Time*. Delacorte Press.. pg. 79

³ Fiering, L., Silver, M., Simpson, R. & Smith, D. 2003. *Linux on the Desktop: The Whole Story*. Gartner Research. pg. 3

⁴ Chad Robinson is a senior analyst at Robert Francis group, a firm which has done extensive research into Linux TCO. http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html

The main advantage of using a “consumer” class processor is that this class of processors has much better price performance ratios compared to high cost RISC¹ class processors, which are used in proprietary hardware solutions offered by companies like Sun. Depending on the application of a server or workstation, a “consumer” level processor such as an Intel or AMD model offers increased benefits, especially at upgrading time. Instead of purchasing expensive proprietary machines, organizations can instead invest in multiple workstations using standard x86 processors. If the machine is used for processor intensive tasks where an expensive high performance machine would be required, Linux can “cluster” several machines which run several machines’ processors in parallel, in order to create similar performance at a much lower cost. ²This is especially prevalent in modern server computers which would rather employ multiple consumer processors instead of one expensive RISC class processor.

It is for this very reason that Apple moved to the more industry accepted Intel standard. Apple traditionally always used IBM PowerPC based machines due to a believed performance benefit. In 2005, Apple was the only company using PowerPC processors in a consumer machine. However, as time moved on, these PowerPC processors became comparatively slower than Intel processors. This can be ascribed to the fact that Intel has a greater pressure to create high performance chips and need to be faster to implement these improvements in order to stay competitive. Now the Intel-based Apple hardware is much faster than their PowerPC equipped counterparts. Eventually the entire Apple range was switched over to Intel processors within 9 months due to the demand for faster performance hardware.

¹ RISC (Reduced Instruction Set Computer) processors are built for a more streamlined, faster method of handling application instruction sets. Examples include Sun’s UltraSPARC and the IBM PowerPC processors. These processors do away with unnecessary commands that normal processors require. Because of more streamlined commands, precious computing cycles can be applied to actual necessary commands.

² Modern distributions of Linux have made this type of setup much easier to implement. However, typical of cluster setups, it requires a lot of effort to install and special setup requirements.

Another versatile use of Linux is for security purposes. Instead of discarding an old machine because of outdated hardware, as in many cases, the computer can be used as a very secure, albeit less elegant firewall for an entire network connecting to the outside internet.

b) Software Switching

Linux's greatest cost saver is the lower licensing costs. Compared to proprietary operating systems, Linux has a very low cost per machine, in many cases absolutely free. Depending on the distribution, licensing costs are replaced by subscription models for formal support. This is what distributions like Red Hat use. Even though the software is entirely open source, support will only be given to registered users who paid a subscription fee. In many cases, this support is of a very high quality. In the case of free distributions, support takes the form of an informal support forum which to which the entire open source community can contribute.

Even though licensing costs are very low with Linux, the bigger costs are applications used by the organization. This is the area where the greatest variations in cost can occur. It has been found that smaller businesses with highly vertical applications tend to have a lower TCO when using Linux. However, larger businesses that are originally running on Windows and already have a suite of suitable applications, will not have any real cost savings running Linux, according to a recent study.¹ "And in large enterprises, a significant Linux deployment or total switch from Windows to Linux, would be three to four times more expensive and take three times as long to deploy as an upgrade from one version of Windows to newer Windows releases," was found by the study.

Initially, cost analysis has to be done for the operating system alone, and secondly applications need to be migrated, and in many cases rewritten to support Linux. In doing a cost analysis, previous research might be handy, but it is important to look at legitimacy of

¹ Foley, Mary Jo. 2005. Yankee Independently Pits Windows TCO vs. Linux TCO. *Microsoft Watch*. [online] Available: <http://www.microsoft-watch.com/article2/0,2180,1553620,00.asp> [10 October 2007]

studies. In many situations companies like Microsoft or Red Hat might sponsor a study which can possibly lead to skewed results.

Many applications can be easily replaced by an alternative open source solution; however, it can be difficult to find a solution that has the same feature set as the previously used proprietary one. If a comparable open source solution does not exist, this can further complicate matters. This is especially prevalent in highly vertical segments of the software industry. This can include specialized software such as a dentist's patient database or an engineer's CAD application.

This can also include deficiency in feature sets of alternate applications. For example, if employees are used to an extensive PIM manager like Microsoft Outlook, they cannot use OpenOffice.org as the only replacement. They need to use Mozilla Thunderbird as the e-mail client and then install a calendar plug-in as well. This small consideration can create a lot of work for IT administrators. For every application or plug in that is installed, there is a greater chance of someone having problems at a later stage within the organization. Luckily, most Linux distributions come preinstalled with a variety of "alternative" open source applications, which is well integrated within the operating system. It is the IT administrator's job to make sure these applications fulfil previous requirements.

This is perhaps one of the greatest advantages of Microsoft Office. Because the software is sold as a "suite" or bundle, updates and fixes to the software is received in the same manner. It works almost by the principle of where if something has less parts, less things can go wrong. If applications are dependent on plug-ins which are not necessarily made by the same developer, there is an inherent unstable nature to it. Naturally some applications thrive on plug-ins and make the user well aware of problems and potential updates (e.g. Mozilla Firefox). So preferably, IT managers will choose applications that have built in functionality instead of making use of third party plug-ins.

3.3 Total cost of ownership comparison

In order to make the right decision regarding migration to open source software, a TCO analysis needs to be done which compares the options available. According to Ferengul, Linux and Windows do not cost the same to integrate. "However, the potential for a lower

Linux TCO exists, depending on what choices an organization makes, how much work it wants to take on itself, and how much risk it desires."¹

The organization has to make specific choices with regard to professional services required. Choosing a non-proprietary system is inherently risky due to a lack of support, therefore support can be considered compulsory. Depending on options chosen, savings of up to 40% can be made by using Linux instead of Windows, according to Ferungal. This is of course highly dependent on a variety of factors.

Ferungal created his own formula for predicting TCO:

$$\begin{aligned} & \textit{Distribution} + \textit{Support} + \textit{Management Tools} + \textit{Software Stack} + \textit{Use Case} \\ & = \textit{User-Controllable Total Cost of Ownership} \end{aligned}$$

Here are the five factors better explained:

a) *Distribution*

Depending on the choice of distribution type and organizational staff skill, commercial “distros” will probably have much lower costs in the long run. Free distributions might have a much lower or free licensing cost, but the cost of customizing software and a lack of formal support can severely increase costs. Support then has to come from in-house experts which in many cases can cost just as much as formal support to hire, or skilled staff needs to be hir

b) *Support*

If a commercial distribution is used, is formal support being purchased as well? Even though it can be costly, it is considered compulsory in important environments where uptime is

¹Preimesberger, Chris. 2004. Four out of four experts agree: Linux lowers TCO. *IT Manager's journal*. <http://www.itmanagersjournal.com/management/04/06/04/2114222.shtml>

critical. Again, in-house experts can possibly be even more costly to pay, and thus support might be well worth the apparent steep price.

c) *Management tools*

What are the costs involved with management tools? If Red Hat or SUSE Enterprise Linux is used, a variety of options are available with regard to management and monitoring. Often, the same tools used to traditionally monitor Windows or UNIX installations work flawlessly with these distributions of Linux. However, money can be saved through the use of open source management solutions.

d) *Software stack*

Does the server run MySQL (open source) or Oracle (proprietary) as a database framework? “There are obvious cost implications to selecting open source infrastructure framework services (e.g. databases, Web servers, Web application servers)” according to Ferungal.

e) *Use case*

This has to do with Linux and how it will be used with regard to virtualization. Will the organization choose to use Linux in a virtualized¹ method, or choose to run Linux as a consolidation platform? If it is used in a virtualized manner, costs can be saved. For an exemplary plan on relative costs of migration to Linux from UNIX, Windows or Solaris, see Appendix A.

¹ Virtualization is a method of running more than one operating system concurrently on the same system. Using virtualization software, a “guest” operating system can be run at the same time as a “host” OS. This is especially helpful in support systems, seeing as many computers do not have to be used in order to test out different operating system scenarios. This also drives down hardware costs. [online] Available: <http://btquarterly.com/?mc=virtualization-big-picture&page=virt-viewresearch> [10 October 2007]

It shows that in many cases, it might prove cheaper to use virtualization instead of migrating to an entirely new operating system, especially in IT or server environments. For example, if a specific application runs only on Windows, that application can still be run on a Linux machine by using a virtual Windows machine. The virtualization software can be costly, but it is still affordable compared to purchasing a new computer. Modern computer hardware runs virtualized operating systems without any problems.

One good point made by Ferungal is that Linux is very versatile in the integration phase. Just by looking at all these variations that can occur, it is easy to imagine Linux costing the same as a proprietary package, or instead considerable savings can be made, depending on the amount of risk the organization is comfortable with.

Table 3.1: Examples of companies that have adapted open source due to lower TCO¹

Company	Description of open source application	Reasons
BMW	After being sued by SCO for the use of proprietary UNIX code, BMW shifted their R&D department to entirely open source solutions. Used in F1 for better driving and strength test simulations within a constrained budget, achieved using clustering ² .	Reliability, Security and TCO

¹ Salvaggio, S. 2006. Open Source: a r/evolution in the software industry? pg. 84

² Clustering is a method of connecting multiple low-cost computers to all contribute together towards a single application. In most cases they are connected by cheap local area network cabling. The main advantage of this is lower total cost compared to a supercomputer that will have the same processing ability. Modern distributions of Linux have made the implementation of this type of system much easier. Many different setups can be achieved. Recently, a Playstation 3 cluster has been implemented for high level math research.

Company	Description of open source application	Reasons
GlaxoSmithKline	Used for administering Glaxo's corporate portal.	Security and TCO.
Eurochamber	Used for the promotion and administration of European / Japanese trading portal. Has to support intricacies of both languages.	Security, Reliability and TCO.
Wireless Developer Network	Host of http://www.wirelessdevnet.com/ , a news and information site for internet visionaries and internet architects that specialize in all aspects of wireless communication. Is built on Web 2.0 principles requiring high amount of interoperability between services.	TCO and Reliability.
Google	Google is the world's most popular search engine. Switch to Linux improves performance using parallel clustering methods. Google aims to have 1 million servers running in parallel in the future.	TCO and parallel computing.
Orbitz	Online travel agency. Reason for switch to Linux is purely a TCO decision.	TCO
Weather.com	Weather.com is the webpage for the Weather Channel. Consistently in the top 10 for News,	Performance increase

Physorg.com.. 2007. Engineer Creates First Academic Playstation 3 Computing Cluster. *NC State University* [Online] Available: <http://www.physorg.com/pdf92674403.pdf>

Company	Description of open source application	Reasons
	Entertainment and Information websites ¹ .	
MIT	One-Laptop-per-Child is a non-profit organization who wants to develop a \$100 laptop for 3 rd world countries with the aim of improving education. Linux is a major driver in lowering the overall cost of the machines ² .	TCO and manageability

Table 3.1: Examples of companies that have adapted open source due to lower TCO³

3.4 Return on Investment Analysis

One of the most arduous tasks faced by an IT manager is the accurate representation of a new technology initiative in terms of return on investment. Seeing as open source software cost is difficult to quantify in terms of total cost of ownership, a positive return on investment becomes even harder to project.

One method managers use to justify costs is to label new implementations as a “strategic” investments⁴ - these types of investments do not require the same amount of scrutiny. Therefore, if costs are excessive it can still be excessive due to possible savings in the future.

¹ Analysis by Media Metrix. [online] Available: <http://www.comscore.com/> [10 October 2007]

² The laptop will use Linux as an operating system. The rest of the components will be very low end. More info can be found from <http://laptop.media.mit.edu/>

³ Salvaggio, S. 2006. Open Source: a r/evolution in the software industry? pg. 84

⁴ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 68

To calculate return on investment, Woods and Guliani propose following a similar formula to commercial software, albeit with a few considerations.

$$ROI = \frac{\textit{Return: (Increased Revenue + Savings)}}{\textit{Investment: (Evaluation + Licence and Maintenance + Installation and Configuration + Integration and Customization + Operations and Support)}}$$

Important aspects to take into account are (adapted from Woods; Guliani):¹

- *Time horizon*: How long can we expect to have revenue from investments? This is especially troublesome to predict with open source migration.
- *What alternatives exist?* Can these alternatives be compared to current choices? How big is the learning curve?
- *Hurdle rate*: What is the minimum return expected by the organization for any investment? Does software migration fall under this hurdle rate policy?
- *What infrastructure costs are included?* In many situations extra costs, such as electricity or broadband, is not included. In third world countries, these costs can be especially high.

By taking these costs into account, ROI studies can vary a lot. Because of these variations, many vendor based studies are altered by comparing different systems – for example, Microsoft sponsored studies comparing mainframe class machines running Linux (which is hardly ever needed, especially with server farms) with much cheaper Microsoft Windows PCs². In fact, Linux is much more likely to run on cheaper x86 hardware rather than

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 69

² Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 69

expensive mainframe hardware. This type of study makes IT administrators aware to look out for improper comparisons.

Open source software is more difficult to do ROI analysis with. Especially in test phases where software has to be evaluated, open source tends to take longer to evaluate. When using commercial software, buyers are continually made aware of possible advantages and applications of new software. Open source normally has a smaller marketing budget and the IT tester is pretty much left on their own to do the feature test research. IT has to install and test various scenarios with the software which will obviously take longer than with proprietary software.

a) Evaluation costs

At evaluation level, licensing costs are not a real issue, seeing as many vendors cater for new users by supplying evaluation or “trial” software which is handicapped by a time limit. Therefore, during the testing phase commercial software might reflect more positively towards ROI.

This can include the following factors: ¹

- The search for open source alternatives
- Test environment creation
- Writing of test applications
- Researching possible scenarios.

b) Licensing costs

Open source software is not always free, especially when looking at commercial distributions (like Redhat or SuSE). End-users or organizations are charged fees for support – this normally includes updates and dedicated installation customized to the organization. Seeing as this already negates the advantages of zero licensing costs, many IT administrators with

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 74

higher skill levels will prefer to use non-commercial distributions (like OpenSuSE or Mandriva) of Linux. Commercial software requires users to re-licence software every few years because they drop software support for older releases. Another disadvantage of using proprietary software is that user vendor lock-in occurs in terms of file formats and compatibility, which can be very costly a few years later.

c) *Installation and configuration costs*

Installation and configuration can be a time consuming period for any software implementation period. Depending on the maturity of the software, installation for many open source software packages can be a fairly delicate process that needs a certain level of comprehension from the IT users. Because many open source software packages suffer from poor productization in the interface, configuration is much harder to perform, even on highly regarded software such as Apache Webserver.

This can include the following factors:¹

- Time that engineer spends learning installation procedures (development, test and production environment)
- Time for performance testing
- Time learning to operate and monitor the software
- Training time

d) *Integration and customization*

This is one area where open source software truly shines. IT administrators have freedom in altering the applications before deployment throughout the organization. A naïve open source advocate will let you know that this is a tremendous cost savings advantage due to source code being readily available. However, this requires a high skill level in the IT staff. Even intermediate skill levels can add a lot of development time to the implementation phase. Due

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 72

to a lack of productization in the installation process, implementers do not have a variety of installation scenarios to choose from, something which most proprietary software possesses.

However, changes to proprietary software require expensive support calls or customization efforts – something that can be done with open source for free, provided the skill level in the IT department is adequate.

This can include the following factors:¹

- Time spent gathering requirements
- Times spent on bulletin boards and forums to further research and understand source code.
- Consultation fees

e) Operations and Support

Woods and Guliani argue that open source and proprietary software costs are the same once they are installed, configured and integrated. Support costs are pretty much the same; however, open source has a great advantage with manageability post installation. This is of course heavily influenced by the skill level of IT.

Proprietary vendors charge excessive fees for “development” licences, which are necessary to create scalable environments needed by larger organizations with specific needs. For example, MSDN (Microsoft Developer Network).

This can include the following factors:²

- Hardware costs
- “Call out” costs, normally charged by the hour
- Infrastructure and hire costs (Rack space, electricity, bandwidth and backups)

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 73

² Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 74

3.5 Conclusion

As can be gathered by these different factors OSS has a few characteristics that have to be carefully evaluated before implementing it in the enterprise environment. However, the principles of OSS are ideally suited for government use, and it is for this very reason why it is important to see how these governments are implementing it. Initiating a OSS strategy is normally based on a certain stance towards free software as well as preventing vendor lock in.

The many OSS advantages in security, reliability and initial low cost has made it very popular in the server market but it is now coming under the spotlight for desktop usage. When taking the typical computer user into account, OSS has plenty of opportunity to be the ideal software alternative to use. Of course, “typical computer user” is quite a broad term in today’s increasingly computer driven workplace, so we need to differentiate between those users as well in chapter 5.

South Africa is at the forefront of open source adaption in the government sector, and the reasons are pretty obvious. There is a growing trend for government sectors to switch to OSS, if not only for the perceived lower costs, but also the added convenience of not being locked into a specific file format, something which is an increasing concern in most industry sectors, not only governments. Computer users do not want to be forced to comply with a “walled garden” (or gated community) by only using one specific application or operating system because the file format requires it. Data needs to be seen as separate from the application, and a document or database should be able to be opened by a variety of applications depending on the users choice. It is for this very reason that formats such XML and ODF exist.

Seeing as the government sector is a great consumer of IT, its insistence of switching to OSS might be a signal of things to come in other sectors as well.

Chapter 4

Open Source in the Government Sector

The basis of this section will be a study of South Africa and other emerging nations who have implemented an open source strategy in the government sectors. The aspects to consider regarding the implementation of an open source strategy in the government sectors are costs, security, training and interoperability with current systems and formats.¹ In 2002 and 2003, the SA Cabinet adopted policies from the Government IT Officer's Council (GITOC) regarding Free and Open Source Software.² Up until August 2006, aspects such as technical performance, security, implementation and costs were addressed and finally set forth in a revised policy for the support of FOSS solutions. The main argument set forth was the contribution to economic development in South Africa.

¹ Department of Public Service and Administration. 2006. Policy on Free and Open Source Software Use for South African Government. Available online [May 2007]:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf

² Department of Public Service and Administration. 2006. Policy on Free and Open Source Software Use for South African Government. Available online [May 2007]:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf

4.1 The feasibility of a government open source initiative

FOSS advocates from various sectors are encouraging governments to take clear stances on its attitudes towards open source. The reasons include the following:¹

- Lower the cost Information Technology, and so increase the access of ICT for other parts of society.
- Training environment is ideal for future development of skills.
- Will help convert the nation from a predominate consumer of technology to a supplier / producer, which is especially valuable in developing nations.

Advocates of free and open source software target government sectors for very good reasons. The public sector is the largest ICT customer in Africa, with the South African Government being the largest ICT user in the continent.²

a) Costs

Since government funding is essentially from the tax payer's wallet, this is a very important aspect to consider. First, one has to consider the high cost regarding purchasing proprietary software on a computer. If proprietary software like Microsoft Windows and Microsoft Office increases the cost of a new computer to the point of being almost unaffordable to the general consumer, and the cost of thousands of computers used in the government, this can

¹ Bridges.org. 2005. *Free / open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners*. p 1. [online] Available:

http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf [September 2007]

² Bridges.org. 2005. *Free / open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners*. p 1. [online] Available:

http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf [September 2007]

potentially waste a lot of government resources on something as simple as creating word processor documents.¹

Secondly, the use of proprietary software “locks in” the government when the time comes for upgrading of hardware and software. If proprietary software is used, the government is forced to purchase the upgraded version of its current software which evades the need for competition, almost creating a monopolistic scenario. If the government uses open source software and file formats, once it is time to upgrade, providers of software can bid to supply, lowering the cost remarkably. The implementation of at least using only open source productivity software will save a lot of money for the government without the need for extensive efforts in re-training.

b) Security

The government has the crucial responsibility of keeping confidential information safe. One responsibility is preventing this confidential information from falling into the wrong people's hands. Another is preventing it from being lost. Proprietary software is traditionally plagued with security bugs the moment it is released. Because of its popularity, hackers target Windows and discover vulnerabilities in the code, creating opportunities to steal information or destroy data on computers. Linux source code is made available to anyone who wants it, and for this very reason, back doors, bugs and errors in the source code can be quickly identified and fixed by the open source community.²

Linux patches are also quicker to be dispatched than Windows patches. Microsoft only releases patches on the second Tuesday every month, where Linux patches are available to the public the instant it is created. For this reason, the government needs to seriously consider Linux regarding security of software.

¹ This does not take into account the value of content created. However, content created by either suite can be of the same high quality. Obviously the quality is dependent on the level of training received by the end user.

² Liu, Louisa; Hayward, Bob M. 2005 Open Source can be catalyst for IT in China. *Gartner Research*. pg. 5

Another point to consider is that every few years the Independent Electoral Committee has to run a voting poll for the South African government. There is pressure to implement electronic systems to decrease counting errors, fraud and excessive labour. The biggest reason why this has not been fully implemented is the aspects of security. Again, considering Windows's track record regarding hacker vulnerability, it is potentially not the best choice in these circumstances. The usage of open source applications and firewalls will prove beneficial, significantly saving costs involved in voting in terms of labour and speeding up the entire process. However, a recent report states that Microsoft Windows Vista currently has least amount of vulnerabilities compared to other operating systems.¹

Another emergence is countries which are shifting to OSS because they do not want U.S. developed binary-only software, for the reason that flaws in the code might be exploited when used in military or intelligence-orientated environments.²

c) Training

The major factor to consider regarding OSS (especially Linux) is that it requires retraining of employees in almost all cases. Even though some Linux user interfaces share characteristics and visual cues with Windows, it is not always very intuitive to someone who has only used Windows. Linux is not necessarily difficult to use, it is just not entirely similar to Windows, and most users will need training sessions to become fully accustomed to certain aspects on the screen. Training is an important aspect regarding implementation cost and total cost of ownership and can be prohibitively expensive also in terms of necessary downtime during training. In the current South African OSS migration effort this can be considered the biggest cost hurdle.

Ease of use when using open source office applications is less of an issue, and training will be minimal, if at all necessary. User interfaces are almost exactly the same as Microsoft Office

¹Jones, Jeff. 2007. Windows Vista – 6 Months Vulneribility report. *CSO security*. [online] Available: http://www.csoonline.com/pdf/6_Month_Vista_Vuln_Report.pdf [20 October 2007]

²Payne, Christian. 2002. On the security of open source software. *Information Systems Journal*. pg.

applications, and some users report that it is actually easier and more intuitive to use.¹ The only aspect that is different is the use of the OpenDocument format, should the government believe it is necessary.

d) Compatibility / Interoperability

An application that has been written for a certain operating system will in most instances not run on a different operating system. IT administrators must take into account the costs involved in finding software that Linux would use. There are virtualization options² available. Yet, the most reliable options are only affordable for use in IT departments. If the computer is primarily used for office applications, this becomes less of an issue. Open source office applications are usually almost fully compatible with Microsoft Office.

The important applications to consider are information system management software and proprietary applications which were written specifically for the government institution. Administration and enterprise level groupware are usually not available as open source, so there can be significant costs to buy new software for this function, and again, more training is required which increases costs. One aspect to consider is that front-ends for modern vertical software is becoming web-based, using web scripting tools such as PHP. This means that the actual operating of the systems can be done from web browsers such as Mozilla Firefox.³

¹ Miller, Robin. Hoe OpenOffice 2.0 stacks up against Microsoft Office. *SearchEnterpriseLinux.com* [online] Available: http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci1174145,00.html [10 September 2007]

² Emulation options exist for Linux, Windows and Mac. See Table 1. These emulation options are also called “virtualization” and can save tremendous hardware cost especially in an IT testing environment.

³ PHP is an open standard based reflective programming language. Scripts are run on a server, with the interface made available on a web browser. For more info: <http://www.php.net/tut.php>

If one considers these aspects above, it seems necessary to divide open source software between operating systems (Linux) and applications. In a government application, Linux will indeed save significant costs regarding licences and upgrading. Nevertheless, one needs to take into account total cost of ownership. If Linux is implemented, users will need to be trained to use the operating system, which increases costs significantly, and the government will lose productivity until users are trained to use Linux effectively.

If users cannot use Linux, they will be unable to use any application effectively that runs on it as well. Also, considering that thorough training is not always possible financially and time-wise, there is the additional cost of hiring or contracting a technical support group. If all these costs are added up, the government may realise that the implementation of Linux will be prohibitively costly despite the free licensing, and Windows will still prove to be cheaper in the long run. In a study by Logan, it was concluded that "Open source software can bring with it selection, support and maintenance issues that are more complex than those of commercial products"¹, something that is evidently true.

When one considers the use of open source productivity software, the picture changes dramatically. OpenOffice has almost the exact same feature set as Microsoft Office, and is for the typical user just as easy to use. The training involved in using OpenOffice will be minimal and many users can actually use the software without the need for any training. Licensing costs are still free, but the total cost of ownership is also significantly lower than Microsoft Office. Another benefit is that employees can create and receive files which are fully compatible with Microsoft Office's proprietary formats. Employees can also choose to start to use the OpenDocument format which saves money down the line when upgrading should be considered, once again preventing vendor lock in.

Therefore, in government, open source software can prove to be very beneficial regarding productivity software, but the use of Linux is not necessarily the best idea, especially for an emerging nation like South Africa, where information technology skill is relatively scarce. If South Africa does have the necessary support infrastructure, Linux can become viable. The

¹ Logan, Debra. 2004. New EU States Will Add to Demand for Open source Software. *Gartner Research*. pg. 2

current strategy is to get support from vendors, in order to stimulate growth in the support industry, and through that build the necessary skills due to enhanced demand. If this strategy is used, the government can save a lot of money at the time of purchase, but also in terms of total costs of ownership when one considers upgrading.

e) Lowering barriers for local developers

One of the key reasons for the shift towards open source solutions is to improve the South African economic development. If a migration is made to open source solutions which can be locally supported, this can positively influence the development of the ICT solution for use in the public sector. This in turn improves service in the non-government sector.

The South African government has fully dedicated itself to development and integration of open source software. In fact, through the ongoing efforts of the Department of Public Service and Administration, South Africa is one the most prominent nations to promote open source software. South Africa is also showing great potential in the development of open source software through a number of training initiatives, mostly notably the “Geek Freedom League”, a Mark Shuttleworth funded project. This will potentially increase the intellectual capital of South Africa’s IT sector in future years.

Instances where the South African government has already implemented open source strategies:¹

- DPSA (and CSIR) is developing an open source corruption management system.
- The government wide area network (WAN) uses only open source software.
- The Centre for Public Service Innovation uses open source software to give mobile access to government services in less privileged areas of the country.

The CSIR Meraka Institute is a committed “facilitate national economic and social development through human capital development and needs-based research and innovation,

¹ All SA project information was found from Go Open source, a South African open source promotion website. For more information on Open source projects in South Africa, go to http://www.go-opensource.org/oss_in_south_africa

leading to products and services based on Information and Communication Technology.” One of its key catalysts in implementing this vision is through the implementation of free and open source software. For this very reason the Meraka institute established an Open Source Centre to “amplify the beneficial impact of FLOSS across all sectors of society (public, private, civil)”.¹

Current strategic objectives of the Meraka Open Source Centre include:

- Establish the use of FLOSS to make a meaningful contribution to the region's challenges in relation to economic development, service delivery innovation and ICT education.²
- Affect a turnaround in the Southern African ICT market from an importer to an exporter, thereby contributing to better ICT value chain management within the region and improving its competitiveness.³
- Develop a leading role for Southern Africa in software development, thereby contributing to a viable indigenous software industry.⁴
- Identify and create further opportunities for such development and use of FLOSS in Southern Africa, thereby allowing fuller use of information and communication technologies. An example is in the need for human language technologies⁵
- Lower entry barriers for emerging ICT businesses and other businesses.¹

¹ Open Source Centre homepage. 2007. Meraka Insitute. [online] Available: <http://floss.meraka.org.za/> [10 August 2007]

² Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

³ Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

⁴ Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

⁵ Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

- Lower inhibitions to large-scale adoption of FLOSS as well as peace of mind for those considering various migration options.²
- Engage with other countries to extend our mission and exchange learning and experience in this essentially collaborative effort.³

4.2 International efforts to adapt open source software

Current development is focussed in European and North American continents⁴, but this does not mean that only their end-users benefit. Governments have to consider carefully whether open source is indeed the best choice for the development of IT within their specific country. Many factors play a role, including current infrastructure, costs and past IT experience. If a government does support the growth of open source, it has to choose its method of adoption carefully. Some governments choose a blend of proprietary and open source software (always choosing the best, most efficient product for the job), or an all-out open source solution. If this latter choice is decided upon, it needs to be carefully analysed. Many uninformed decisions are made only on the basis of cost and the apparent belief of “free” open source software. Again, a thorough TCO analysis is needed.

Many advocates argue that governments are key players in “levelling the playing field” with regard to the strong market position of proprietary software companies, and in doing so create

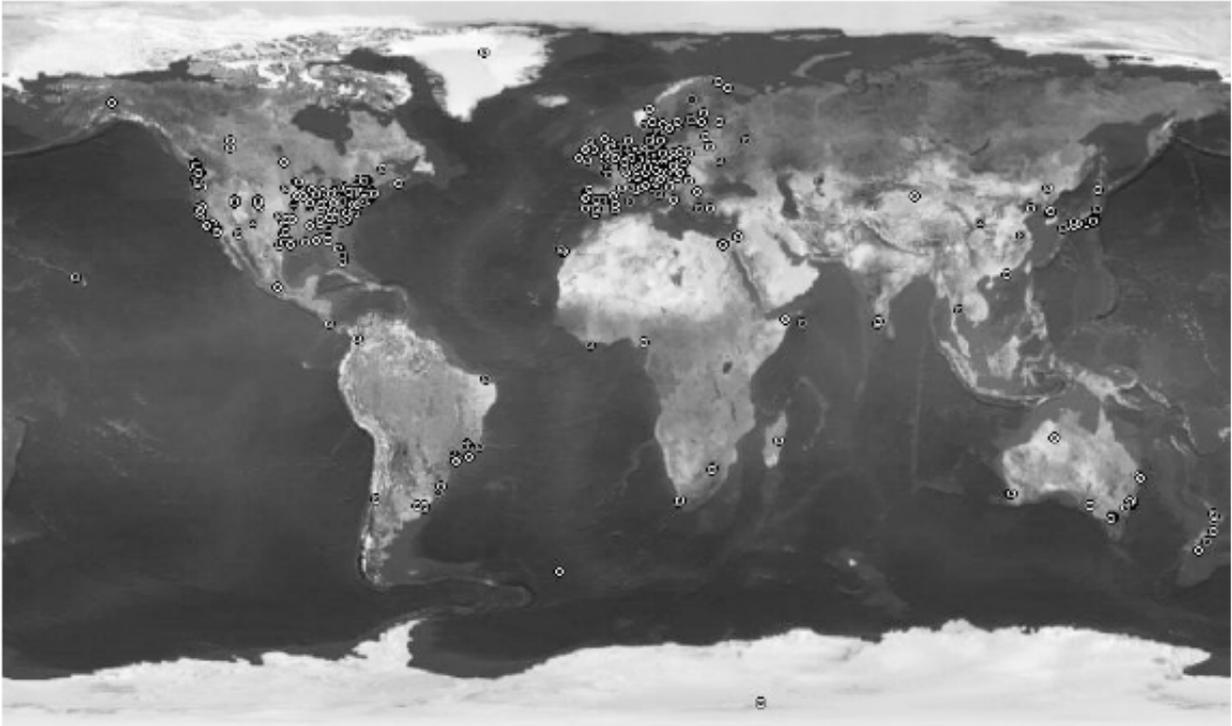
¹ Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

² Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

³ Meraka Institute. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

⁴ This can be seen in figure 4.

more healthy competition in the software industry.¹ There are, however, arguments against this, with some software companies arguing that governments are not in the position to alter with market forces, and that intervention will inevitably “hurt competition and innovation”.²



*Fig 4: Geographical repartition of Linux developers.*³

¹ Bridges.org. 2005. *Free / open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners*. pg. 1. [online] Available:

http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf [10 August 2007]

² Bridges.org. 2005. *Free / open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners*. pg. 1 & 2. [online] Available:

http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf [10 August 2007]

³ Libre SoftEng. 2006. Geographic Location of (over one million) SourceForge Developers.

Sourceforge. [online] Available: <http://libresoft.es/libresoft/68>. [25 October 2007]. This figure shows the areas with significant open source development centres across the world.

a) United Kingdom

The United Kingdom was one of the first governments to implement an interoperability policy through the use of open source software. In 2002 the government drew up a policy that it would only use products that support open standards and specifications. This was primarily to prevent vendor lock in. Upgrade cycles were also lengthened, saving additional costs.¹

b) France

France can be considered the country with the most aggressive stance for the implementation of open source software in the government sector. In October 1999 Pierre Lafitte (a senator at the time) created the Proposition de Loi 495. This bill stipulated that the French government needs to use software (at both operating system and application level) of which the source code is open to be revised. This is not only to include open source with other proprietary software, but the use of explicit open source in local government and administrative systems.²

Its reasons are mostly because of excessive proprietary software costs and heightened security awareness. In 2000 Bruno Mannoni (French ministry IT manager) switched 300 of the state's mail, file and web servers to open source software.³ These servers used a commercial Linux distribution (Red-Hat) due to formal support being needed for these

¹ eGovernment Resource Centre. 2006. *Open source software archive for United Kingdom*. [online] Available: <http://www.egov.vic.gov.au/index.php?env=-innews/detail:m1005-1-1-8-s-0:n-250-1-0-> - [22 August 2007]

² Lettice, John. 1999. French Senators propose making open source compulsory. *The Register*. [online] Available: http://www.theregister.co.uk/1999/10/24/french_senators_propose_making_open/ [10 October 2007]

³ Williams, Sam. 2002. A Timeline of Open source in Government. *Linux Dev Center*. [online] Available: http://www.linuxdevcenter.com/pub/a/linux/2002/07/15/osgov_timeline.html [10 October 2007]

mission critical applications. Microsoft's security track record is not very good and France does not want to put it to the test.

Laffitte met with Richard Stallman (Free Software Foundation and the founder of the GNU project) and decided to alter Proposition de Loi 495. Stallman made him aware of cultural and pragmatic values that were associated with free software and "software free of rights". Laffitte reworded the bill with "free software". He also proposed a new division known as the Free Software Agency.¹

Even though France should be applauded for its forward thinking mentality with regard to open access of software, it does perhaps have an over-aggressive stance. This is for the same reason that some people do not support the Free Software Foundation. Open source pundits tend to be more open to the combined use of proprietary software and open source, where France and the FSF want an open source only solution. In many situations this policy can increase TCO tremendously, because all software has to be rewritten and staff training is necessary.

Despite the challenges in their strategy, France can stand to gain a lot from its strategies in the future, especially during software upgrades, seeing as software vendors cannot implement a "lock-in" monopoly type situation. In fact, costs will be saved, as big proprietary vendors will have to bargain with lower prices due to France's pro open source stance.

The Ministry of Equipment is migrating 1500 servers to Linux during 2007.²

¹ Williams, Sam. 2002. A Timeline of Open source in Government. *Linux Dev Center* . [online] Available: http://www.linuxdevcenter.com/pub/a/linux/2002/07/15/osgov_timeline.html [10 October 2007]

² California Office of Government Commerce. 2007. Open Source Software Trials in Government Final Report. [online] Available: <http://www.arb.ca.gov/oss/articles/report-v8d.pdf> pg. 8 [14 October 2007]

c) Germany

Germany is in the unique position of being one of the countries which contributes the most to the development of open source software¹. It could be because certain countries place a higher cultural emphasis on the contributions of a community. In 2002 the German Parliament announced the use of SUSE Linux and OpenDLAP for use in its government infrastructure.

In 2002 IBM was contracted to install Linux machines across the interior ministry's infrastructure. The reason as stated by Interior Minister Otto Schilly was that they were "raising computer security by avoiding a [software] monoculture". This reason has again to do with the perception of increased security of Linux distributions compared to Microsoft's Windows Server offerings.

Munich city is migrating 14000 desktop machines to the Linux operating system, using OpenOffice and Mozilla Firefox, with the project aiming at completion in 2008/2009.²

d) China

In 2000, China's media reports that "Red Flag Linux" would become the official operating system of China. At that time it was not yet released and many competitors denounced the "official" moniker, but the Red Flag Linux name was confirmed as a Chinese language distribution of the Linux kernel. The People's Republic of China is currently in planning to

¹ It is very difficult to analyse where snippets of code come from. Paul Jones did a study from the received e-mails and found that the German ".de" suffix at the end of an e-mail address was found the most. <http://www.techweb.com/wire/story/TWB20001101S0016>

² California Office of Government Commerce. 2007. Open Source Software Trials in Government Final Report. [online] Available: <http://www.arb.ca.gov/oss/articles/report-v8d.pdf> pg. 8 [14 October 2007]

widely distribute this OSS solution on over 200 million desktop machines, with the assistance of Sun Microsystems.¹

China is against the idea of using westernized software in its government and started researching and funding an alternative to Microsoft Windows in the form of Red Flag Linux. The New York Times reporter Craig Smith stated that: “A growing number of Chinese have likened dependence on Microsoft to leaving the keys to the country’s increasingly computerized economy in the hands of a potential enemy”. This is obviously also referring to Microsoft being an American company - Windows has been known to be released with “backdoors” in case changes need to be made to the software, something which hackers tend to exploit, which can lead to tremendous software problems. China wants to make Red Flag Linux available to the public, and in doing so lower the cost of computers and overall IT infrastructure.

e) *United States of America*

The US is viewed as the father of proprietary software, because it is where all major proprietary software companies are based. However, it can also be seen as the instigator of open source and free software because the development traces back to US based Universities. Countries view open source as the primary way to compete with the US software industry which can be considered the strongest in the world.

Many states in the US are implementing open source initiatives. Most notably, the state of Massachusetts has, from 2005, started an open source initiative to convert all government documents to the open source OpenDocument format. This is considered by many as a pioneering move in the shift from proprietary formats in government offices. There are plenty of reasons, the biggest being that they want to prevent being “locked-in” into using proprietary formats in the future. Initially this meant that they had to convert to

¹ California Office of Government Commerce. 2007. Open Source Software Trials in Government Final Report. [online] Available: <http://www.arb.ca.gov/oss/articles/report-v8d.pdf> pg. 9 [14 October]

OpenOffice.org. But in May 2005, an unofficial plug-in for¹ Microsoft Office was released that enabled users to open and save OpenDocument format documents, possibly due to tremendous pressure by the Massachusetts's government offices.

f) India

The Indian government mandated the use of open standard document formats, as a way to prevent vendor lock in. India is a prime user for open source solutions, seeing as the necessary skill set is available for support, and many of its 2,5 million small and medium size businesses have not computerized. This means that these businesses will have no "migration" costs, however, the initial training might be costly.² Large Indian businesses rely on UNIX and Microsoft Windows is seen as a primarily desktop operating system. The Indian government has not, however, made its policy documents publicly available, resulting in some criticism from open source support groups. Only its e-governance sites mention the implementation of open standards.

One fact worth mentioning is that the Indian Department of IT created a strategy to implement Linux and OSS in academic institutions. This will further create both demand and supply for OSS based software and its support. This will in turn benefit the local economy.

¹ Microsoft was not the creator of this plug-in. It was written by third-party open source supporters. Many trade advocates are unhappy with Massachusetts for their pro-open source stance.
<http://www.informationweek.com/news/showArticle.jhtml?articleID=187201534&subSection=All+Stories>

² Greenemeier, Larry. 2004. Why the Open Source Model can work in India. *Information Week*. [online] Available:
<http://www.informationweek.com/story/showArticle.jhtml?articleID=47900215&tid=5979> [22 August 2007]

4.3 Creating a government open source strategy

In order to create a useful open source strategy at a government level, requires the dedication of a certain attitude from state regulators. Taking examples from European and Asian countries, states can either be users or, at a more progressive stance, regulators.

States as users

Government administration offices migrate to open source for operating systems and core (vertical) applications. One significant effort is the migration towards open source in companies where states are major shareholders. This is what is happening in India, Korea and China. This requires an immense effort from state level. One advantage of this stance is that local software industries are boosted, as well as their support based services.¹ In a sense, this fosters freedom in future software migration efforts.

States as regulators

European regulators are taking initiative by offering certain incentives to businesses that migrate to open source solutions. These may include tax cuts, subsidization of research and development. This is especially viable for newer businesses, seeing that the migration costs are decreased. These new businesses have fewer “growing pains”, as many school and university graduates are being taught skills for operating open source solutions. Just as with previous solution, local support based service industry is boosted instead of immense licensing fees being paid to large software companies. In perhaps an over-aggressive stance, China banned all Microsoft software from administration offices.

4.4 Typical government strategic positions and the South African revised open source policy

By looking at international trends with regard to open source attitudes, three typical positions can be taken:

- 1) **Mandate** the use of FOSS and open standards
- 2) **Favour** the use of FOSS, especially open standards
- 3) Remain technologically **neutral**

These stances are especially relevant when looking at international efforts. It is clear from some examples into which “camps” a certain nation might fall.

The following policy statements are taken directly from government policy documents:

1) *“The South African Government will implement FOSS unless proprietary software is demonstrated to be significantly superior. Whenever the advantages of FOSS and proprietary software are comparable, FOSS will be implemented when choosing a software solution for a new project. Whenever FOSS is not implemented, then reasons must be provided in order to justify the implementation of proprietary software.”*¹

Justification: This argument is in line with a hybrid adoption method which is described further in this thesis. While the idea behind open source software is often admirable, it is necessary to remain realistic about the absolute quality of product alternatives. If the

¹ Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available: http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

proprietary solution is significantly better, this will also improve the government's performance in terms of workflow and productivity, something which should not be ignored in order to save money. If a solution is chosen purely on the characteristic of being free or open source, it can possibly create expenses during the maintenance phase, especially if it is unproven in production environments. Unless the IT departments can assume risk, it is recommended to stick to mature open source projects.¹

2) *"The South African Government will migrate current proprietary software to FOSS whenever comparable software exists."*²

While this can be purely assessed as a financial reason, many benefits could be gained in terms of interoperability and security which is essential. As described in policy documents, it is often understood that open source software compares very favourably in terms of return on investment, total cost of ownership and technical features. Where previous iterations of the policy draft documents opted for an almost exclusive open source implementation, there exists a consensus that not all users will benefit from open source solutions, despite recent advances in user friendliness and interoperability with FOSS solutions.³

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg 68.

² Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

³ Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

3) *“All new software developed for or by the South African Government will be based on open standards, adherent to FOSS principles, and licensed using a FOSS license where possible.”*¹

This is primarily to prevent vendor lock-in. With modern software solution it is not a big change to implement open standards with data and documents. Traditional proprietary software packages are also starting to implement compatibility with open file formats.² However, it will take time and a lot of effort to convert existing data and documents into open formats. There exists a number of tools which assist with this very process of converting masses of documents, for example StarOffice.³

4) *“The South African Government will ensure all Government content and content developed using Government resources is made Open Content, unless analysis on specific content shows that proprietary licensing or confidentiality is substantially beneficial.”*⁴

Once again, this addresses the issue of vendor lock-in. By implementing policies that all documents should be created in open formats, it will create fewer problems in the long term and give more choice when software migration needs to occur.

¹ Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

² Microsoft has implemented PDF support into its latest Office 2007 suite. In the past this required a third party plug in.

³ The StarOffice webpage has more information on its enterprise migration solutions:
http://www.sun.com/software/star/staroffice/enterprise_tools.jsp#Setup

⁴ Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available:
http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

5) *“The South African Government will encourage the use of Open Content and Open Standards within South Africa.”*¹

By encouraging open source alternatives within South Africa, it promotes the development of local support of open source software. Open Source software is typically more dependent on a decent support structure, especially at a large scale implementation. Despite the savings in licensing costs, support costs are typically higher than with proprietary software. Costs typically spent on licenses of software could then be invested in local support of the ICT sector, increasing job opportunities.

The most important part of a successful implementation strategy is proper analysis and design beforehand. The business needs and requirements need to be properly documented. If this is not properly recorded, it can incur considerable costs at a later stage of migration. It is for this very reason that an overly aggressive stance for OSS can create problems. While the idea to shift to open standards can be considered exemplary, open source software can be more difficult to integrate into existing organizations, especially large consumer groups such as the SA government.

Many different implementation models exist, which are all good enough to follow when it comes to migration of software. However, OSS has unique characteristics which make certain elements more important during the design phase, for example training cost and conversion of existing data. Traditional models rely on return on investment and total cost of ownership analysis in order to make proper managerial decisions. These models need to be modified for OSS peculiarities, seeing as return on investment can be difficult to quantify.

OSS has already made significant inroads in the server market, but the migration is much more difficult on the desktop. Existing documents need to be converted in order to be

¹ Department of Public Service & Administration. 2006. Policy on Free and Open Source Software Use for South African Government. [online] Available: http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf [10 October 2007]

compatible with OSS office suites, which can take a lot of time and effort. Many organizations are not willing to accept the sunken cost into existing proprietary solutions, and the conversion can take months and require a high level of skill. Once the conversion is done, open source office suites are easy to migrate to, seeing as the interface is relatively similar to existing proprietary solutions.

Linux as an operating system is robust, secure, reliable and affordable, and because it works on a variety of hardware platforms, can be very versatile. Even though the initial licensing cost is in most scenarios free, the training is the most significant cost. In the SA government, training is considered the biggest hurdle in the implementation process, something which cannot be ignored. Even though a hybrid solution of an OSS office suite on top of proprietary operating system might save training costs, the benefits to the SA economy further down the line outweigh the initial licensing costs of proprietary software. The South African OSS support industry might not currently be at its full potential, but the additional demand from the government¹ will stimulate growth in the OSS support services. If these services become of a higher quality, other industries will hopefully follow suit and also implement OSS and stimulate growth in the IT sector of South Africa.

¹ The SA government is the biggest consumer of IT services in the country. See Appendix C for interviewees.

Chapter 5

The OSS Migration Process: Challenges and Considerations

Managing a shift to open source software is a difficult process. A company must have the right reasons for moving to it – cost cannot be the only reason. This chapter will argue that there should be more reason for the shift than cost, and that in many cases support costs negate the entire zero licensing cost of open source software. After taking the entire process into account, it can prove cheaper to adhere to whichever platform is currently used by the organization.

During a migratory process, it is necessary to differentiate between server and end-users. Both can benefit from open source, but end-users need to have adequate support systems in place beforehand. If switching to Linux, end-users will definitely need retraining which can be very expensive, especially when considering “down-time” and the initial ineffective usage of software. Migrating server operating systems might be easier however. As the typical server “operator” has a high technical skill level, initial retraining is quicker and cheaper than with end users.

While the migration process can be difficult and costly when working with end-user operating systems, the switch to open source productivity software can be reasonably easy

and cheap. If the same operating system is used, the user can normally adapt to slight changes in the user interface, without a lot of training. If the user is considered a “power user” some training might, however, be required. In order to train users adequately in the minimum amount of time, internal knowledgeable staff might be an invaluable resource. This is because current staff might have the best idea of what features are important to the organization’s end users.

A skills assessment needs to be done – sometimes skills can be found internally which can possibly decrease costs greatly. Based on the pitfalls of an open source migratory process, the following general phases of a typical software migration process will be discussed in this chapter:

- Phase 1:** *Assessment / Analysis*
 (E.g. Buy vs. build, One throat to choke)
- Phase 2:** *Foster management level commitment, formulate strategy*
 (E.g. Increase choice, reduce vendor lock in)
- Phase 3:** *Technical pilot testing and training*
 (E.g. Five Primary user segments, Migration approach)
- Phase 4:** *Organization wide implementation*
- Phase 5:** *Support phase*
- Phase 6:** *Re-assessment phase*

These phases have been developed through interviews with current government OSS implementers (see Appendix C). While certain elements might mimic existing software migration strategies, OSS has certain unique characteristics that result in different emphasis on some phases. For example, a typical OSS implementation would not be successful without management level support. If there is a lack of this support, it might result in users sabotaging the new system, as seen in phase 2.

The first action is to create a master plan. The master plan should cover all of these phases in detail and should be followed quite rigidly. Of course not all possible problems can be predicted in advance, and for this reason the master plan should be malleable in certain ways. Changes made to the plan should take place only in a meeting environment involving

strategic level management and IT. A good idea is to create a checklist for each phase that has to be completed before moving to the next phase. Depending on the size of the deployment, large organizations might need to do a methodical project management analysis in order to define time frames for phases. On smaller end-user groups and server migration, extensive time project management might not be as necessary.

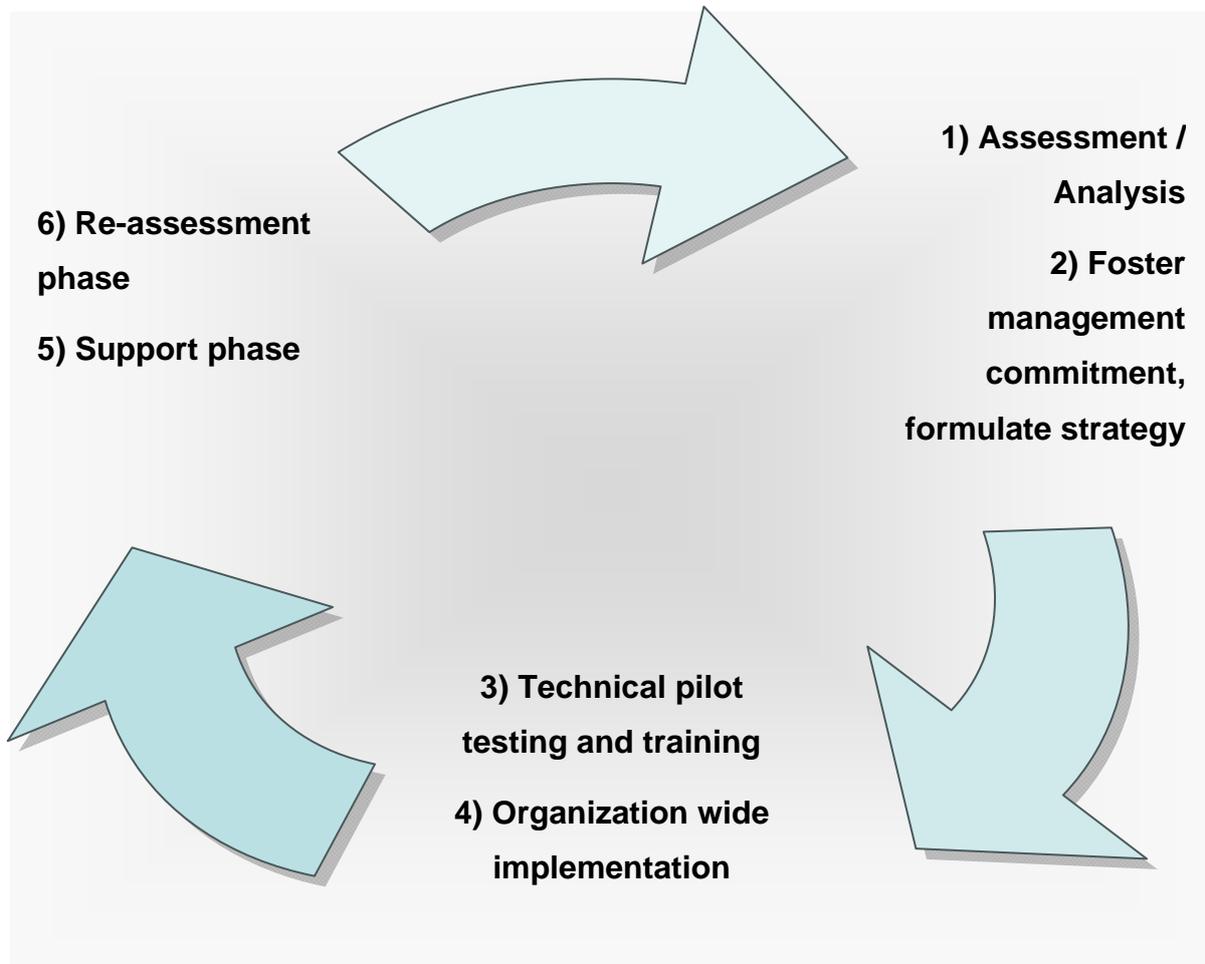


Fig 5.1: The Open Source Migration Process

5.1 Phase 1: Assessment / Analysis

At this stage it is necessary to make a full analysis if open source is indeed the best choice for the business. Depending on the previously installed software base, a full cost and feature analysis is needed between varieties of operating systems. This preliminary phase needs thorough analysis in order to make an informed decision. In terms of government migration, a multi-departmental investigation should take place.

One of the key decisions of IT managers is to decide what software to buy and which parts to build. Buying software (or support licences) gives peace of mind in the form of creating “one throat to choke”. Purchasing vendor based solutions or software creates an accountability partner which can give the necessary support in case it is needed. This is, however, not the right form of recourse for any IT manager. If the necessary skill set is to be found inside the staff, this can solve many problems. Also, the obligatory redundancy systems need to be put in place in order to prevent inconvenience. However, choosing a vendor based on accountability should not be a key aspect. The right attitude is perhaps best described by Woods and Guliani: “Why are we planning on having any throats to choke?”

When looking at buy vs. build, it is important to remember that commercial software is not exclusively “buy”, just as open source software is also not only “building”. In most cases commercial software still has a sizable “requirements gap” which needs to be filled by installing plug-ins and add-ons in order to make the software fulfil organizational needs. There exists many modern open source software applications that are so far along in terms of productization, that it requires almost no altering to be fully functional in the organization.¹ The most sensible choice is a hybrid solution where buy vs. build is handled on a per application basis, instead of a pure “buy” or “build” only strategy.

Migrating operating systems have higher migratory costs than applications. The variables include not only the operating system but also the hardware platform currently in place. In many situations it may prove to be cheaper and beneficial to switch to a different hardware platform. Even though operating systems have specific requirements regarding hardware, open source alternatives exist in almost all software sectors. Part of the analysis phase is to look at all possible alternatives.

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O’Reilly. pg. 138.

Application	Proprietary	Open source
Web browser	Internet Explorer	Mozilla Firefox
Web server	Microsoft IIS, Netscape	Apache
Application server	WebMethods	JBoss, Apache Tomcat
Office suite	MS Office, Corel WordPerfect Office	OpenOffice.org (based on Sun's proprietary StarOffice)
E-mail/collaboration	MS Exchange, Lotus Notes	Ximian Evolution
Database	Oracle9i, DB2, MS SQL	MySQL, PostgreSQL

Table 5.1: Open Source and Proprietary alternatives¹

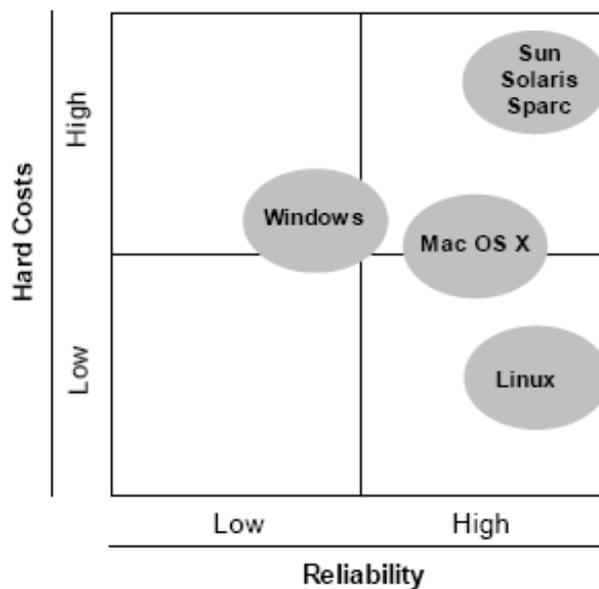


Fig 5.2: Cost vs. reliability analysis of various server operating systems²

In the server scenario, this is not automatically a Windows vs. Linux issue. As can be seen in figure 5.2, SunSPARC has the highest initial cost, but it does have its relevant business applications. It is also considered highly reliable, equal to Linux. Again, see Appendix A for

¹ Sen, Ravi. A Strategic Analysis of Competition Between Open Source and Proprietary Software. 2007. *Journal of Management Information Systems* / Summer 2007. pg. 233. [5 September 2007]

² Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* pg. 73

a modular type server migration strategy, and to see whether virtualization might prove beneficial in server areas.

Refer to Appendix A for an example. Taking this example into account, the following diagram by Salvino A Salvagio becomes useful:

In mission critical server environments reliability becomes a key issue. In tech industry server environments the feature set becomes important as well. Here a feature comparison of each operating system needs to be done:

If Linux (or any open source application) is seen as a true alternative to an existing end-user or server solution, it becomes the organization's CIO's responsibility to start formulating a plan to foster management support, as its support is critical to the success of the project as it is a giant barrier to the success of an open source migration process, or any software migratory process.

Checklist for this phase:

- ✓ Look into buy or build strategies, with regard to advantages of each.
- ✓ Do a thorough hardware analysis, and look at virtualization option available.
- ✓ Look at a variety of server operating systems, not automatically Linux. Due to open standards, most modern server operating systems will be compatible with end user server computing needs.

5.2 Phase 2: Management level commitment and strategy

This is the phase where “people skills” become more important than ever. Without the backing of management, open source migration can create issues with commitment to organizational users at a later stage. Typical of a new system implementation, a few people problems could occur. As stated by Avison and Fitzgerald, users “may feel that the new system will make their job more demanding, less secure, will change their relationships, or

lead to a loss of independence”.¹ If there is a lack of support for the new system, users will do everything to ensure that the system fails. This can fall under three types of negative reaction:

Aggression: Doing anything to ensure that the system does not succeed.

Projection: The intentional loss of data, obvious acts of sabotage. The user then blames this on the new system. In other words, they project their problems on the system.

Avoidance: Some users will simply choose not to work with the new system unless absolutely necessary. These users can be especially costly to migrate at a later stage.

Management needs to be made aware of true advantages in open source as well as risks. Although initial licensing costs are the highlight of open source software, a detailed return on investment model should be used to give a more realistic view of costs associated with a migration. See chapter 3 for a detailed analysis of Woods and Guliani's ROI model for open source software. In conversations with IT administrators Salvaggio found this to be the biggest obstacle in migrating to open source solutions:

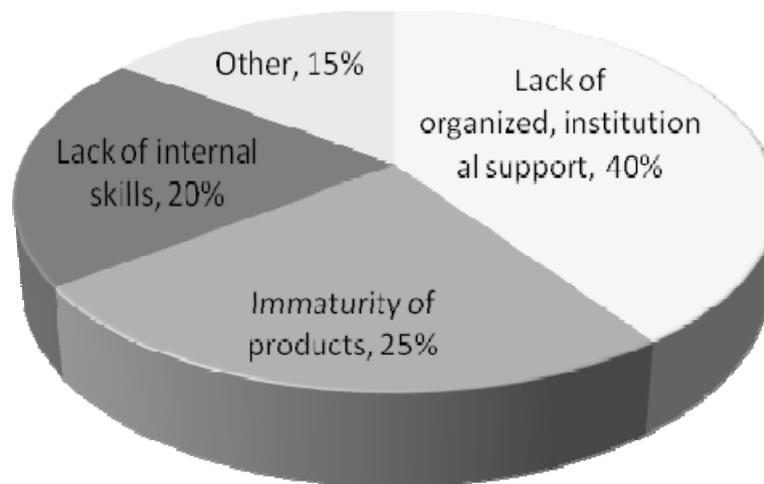


Fig 5.3: Main obstacles in migrating to open source in organizations¹

¹ Avison, Fitzgerald. 2002. *Information Systems Development: Methodologies, Techniques and Tools*. pg. 78.

In order to formulate an open source adoption strategy, key aspects need to be accepted beforehand. In proposing a strategy to management, certain aspects of OSS need to be highlighted. While it might seem biased, it is important because of the difficult nature of quantifying TCO and ROI with open source software. While many of the advantages of OSS can be technical in nature, these facts can be easily communicated.²

- *More independence from software producers*
- *More equality of opportunities*
- *More competition in the software market, with possible benefit to local economy*
- *Complete open source code (maintenance, security)*
- *Middle term monitoring of costs (reduction)*
- *Better protection from virus attacks³*

Checklist for this phase:

- ✓ Create a proposal for management which entails costs and long term benefits. This can prove difficult, because of the use of alternate cost models which are modified for OSS peculiarities. Most important cost factors to take into account is support and training costs. If these are not well analysed, it can be very costly in the future, seeing

¹ Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* p 139

² Hoffman; Harpf. LiMux – free software in Munich. 2007. *Landeshauptstadt Munchen*. [online]
Available:
http://www.muenchen.de/cms/prod2/mde/_de/rubriken/Rathaus/40_dir/limux/publ/free_softw.pdf
[12 September 2007] pg. 5

³ Hoffman; Harpf. LiMux – free software in Munich. 2007. *Landeshauptstadt Munchen*. [online]
Available:
http://www.muenchen.de/cms/prod2/mde/_de/rubriken/Rathaus/40_dir/limux/publ/free_softw.pdf
[12 September 2007] pg. 5

as the analysis should include whether current staff can accommodate new challenges in the migration process.

- ✓ If management does not give its full support, it is not recommended to commence with migration. Users will only give their cooperation if there is necessary incentive.
- ✓ ROI models can be difficult to prove beneficial, costs can be ascribed to “strategic” investments as mentioned in the TCO chapter (chapter 3). Management needs to be made aware of long-term benefits instead of focussing on initial cost savings, which could be potentially misleading due to hidden costs, as described in the TCO formula by Woods and Guliani.

5.3 Phase 3: Technical pilot testing and training

This is the sole responsibility of the IT department and requires testing of the software in a variety of scenarios. This includes testing of installation procedures, performance testing and the modifiability of the source code. If IT has highly skilled employees in this phase, great costs can be saved; otherwise, consultation services need to be used. The software needs to be monitored and managed with ease. Feasibility of a variety of training methods should be researched before moving to the next phase. On switching to Linux, it is important to note that optimizing a Linux system can take a significant amount of time and research.¹ Studies show that Linux administrators spend more time on setup and configuration and up to 30 % more time resolving driver issues than with proprietary operating systems.²

Another aspect to consider is the skill level of support services. With many organizations employing “vertical” niche type software which is written specifically for their market, a lot of software has to be rewritten or virtualized in order to maintain functionality. There is still an apparent gap between industry needs and skill level of open source programmers. If such software is used by the organization, the costs of rewriting and migrating “vertical” software

¹ Andrews, J. 2005. *A+ Guide to Managing and Maintaining your PC*. pg. 1154

² Microsoft. Get the Facts: Total Cost of Ownership. 2006. *Microsoft Corporation*. pg. 4. Available also online: <http://www.microsoft.com/windowsserver/facts/analyses/tco.msp> [02 October 2006]

need to be taken into account. If the necessary skill level is not to be found from support or consultation vendors, the migratory process should be put on hold. Even if the solutions exist, it can be difficult to implement due to immature, unproductized applications.

During technical pilot testing, IT should focus on five key user groups:¹

a) Fixed function

These are users that only use one specific application, and do not use their workstation for anything else. Typically these machines boot into this application at default. Although these users require very specific functionality, they can be difficult to satisfy, as they are used to a single method of doing something.

b) Technical workstation

These users typically do not care about the operating system or Linux distribution they are using, but insist on definitive industry standard applications. For example, graphic designers might insist on using Adobe Photoshop instead of using GIMP, an open source alternative.

c) Transactional worker

Also a user who is less bothered by the operating system compared to the overall user experience. Typically does repetitive tasks using form based software and web applications.

d) Basic office

Users of typical productivity suite applications. They are reliant on these applications to complete everyday work tasks. They are heavily reliant on the compatibility of documents and are required to be able to exchange documents with other departments without problems.

e) General purpose

These users are dependent on standard Windows applications and require industry standard file formats.

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 151

Each of these users require different pilot testing strategies. Certain user groups are heavily reliant on industry standard applications, and these users might be unable to shift to an open source alternatives. Other users are again more open to change in the sense that they aren't completely dependent on proprietary solutions. These users might be the first to undergo pilot testing in order to gain valuable feedback that might prove beneficial at later stages when migrating more technically stringent groups (such as technical workstation users). Typically initial pilot testing needs to be done in layers, not all applications at once. For example, you might want to divide the typical desktop machine into the following layers:¹

Productivity suite (e.g. OpenOffice.org)	Web Browser (e.g. Mozilla Firefox)	Messaging Client (e.g. Skype)	Desktop Database (e.g. MySQL)
Desktop Environment (e.g. KDE)			
Desktop Operating System (e.g. SUSE Linux 10.3)			

Table 5.2: Functionality layers of the typical desktop user

Although one can migrate all of these layers at once, this can lead to extensive problems in training and support, as it is perhaps too much to learn for the average user. Another method is to start switching to OSS alternatives from the top layer downwards. This will only require the user to learn new interfaces in phases, something that is much more manageable than learning all application interface intricacies at once.

Perhaps more important is choosing which OSS alternatives to use. When looking at large scale installations, it is important to look only at well established applications, as these normally have characteristics which are important to administrators. These applications are

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 152

known as “mature” applications and are dependent on a variety of factors, for example ease of use, frequency of updates and reliability. Woods and Guliani created a model to detail the typical life cycle of an open source application. If this life cycle has been repeated several times without stagnating, the software could probably be regarded as a mature example of OSS. Maturely developed software is easier to install and easier to maintain. Updates occur more frequently, and the support base for the software is of a high quality. Mature software does not necessarily mean commercial software, it can be free in most cases. Good examples of this include Apache Web Server on the server side and Mozilla Firefox for end-users.

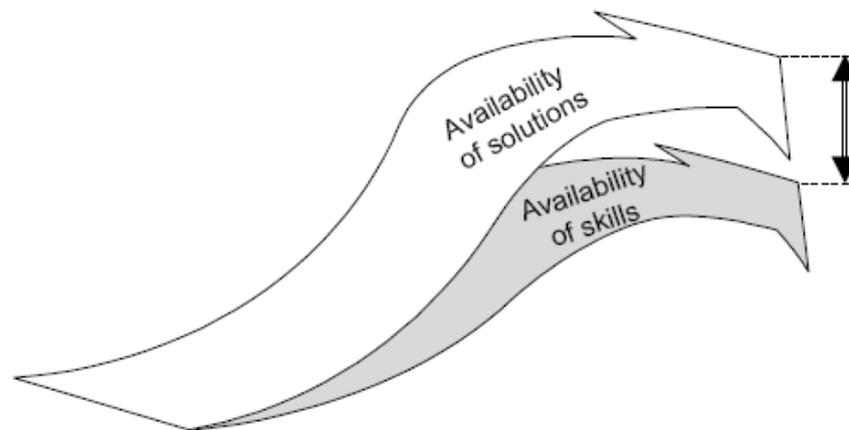


Fig 5.4: Availability of open source solutions is increasing but systems integrators still have limited serious offerings and skills, creating a skills requirement “gap”¹

Many open source solutions give the user a variety of applications to choose from in the installation phase. The installer can install a variety of applications which do similar tasks – as mentioned previously, a typical Linux installation has quite a few browsers available. This scenario is repeated with a variety of typical computer applications such as email, productivity and multimedia centric software. It is necessary to standardize on application in order to maintain consistency throughout the organization.

¹ Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* pg. 73.

Checklist for this phase:

- ✓ Requirements capture requires a large survey group. A small set of users used for testing could create problems at later stages. These late additions due to a lack of a representative user base could be very costly to add later.¹
- ✓ It is necessary to monitor typical computer usage of employees beforehand. This can be achieved using network policy orchestrator software, something which many organizations already employ. Doing this can prevent hassles once the installation has been done, and can save on support calls later down the line.
- ✓ Standardize software applications – make sure only one application is made visible to the end-user for each typical usage scenario. The software bundle chosen should comply with organizational requirements. For example, only Firefox browser, only Novell Evolution email client. This prevents end-user confusion.
- ✓ Make sure adequate skill is available with current staff, as shown in figure 5.4. Even if so, it might be beneficial to have support services available on call, in case a glitch might occur. Despite high support costs, the fees are generally well spent money during the migration process, which gives peace of mind.

5.4 Phase 4: Organization wide implementation

Implementation needs to be well orchestrated and needs to happen as fast as possible. If only key departments are updated on certain occasions, this instantly creates comparisons between software installed in different departments. Because management backing is really necessary, the new software should be installed swiftly. Even though emphasis is placed on the speed of the deployment, the quality of the software should not be poorly chosen due to a lack of planning.

Other challenges such as organizational culture start to appear at this point. While almost all migrations has its share of problems, end users can sabotage a migration effort when they do

¹ Avison, Fitzgerald. 2002. *Information Systems Development: Methodologies, Techniques and Tools*. pg. 100.

not like what they are using. If there is a lack of proper training and the software is difficult to use, widespread negativity for the software can spread very quickly throughout the organization.

It is also important to make the shift to the new system (be it operating system or productivity applications) as quickly as possible. Once the preliminary pilot testing is done, widespread installation should commence – this can be done easily with the use of remote administration tools.¹ If there are differences in application switches across different departments, resentment can arise if the application is not considered satisfactory.

Checklist for this phase:

- ✓ Differences in departmental requirements need to be fully prepared for. If possible, standardized packages should comply with as many departments' need as possible. Standardizing could cut support costs down the line.
- ✓ To enable swift implementation, there exists remote installs for a variety of operating systems. This enables the software to be installed from anywhere on the network. An “image” type installation using hard drive imaging software should work the fastest.²

5.5 Phase 5: Support phase

During the initial usage phase, IT departments will probably be inundated by support calls. If end-user Linux is used, the new software will initially require a learning curve. It is essential

¹ Remote administration can be achieved through the use of remote desktop applications, for example VNC. VNC is also an example of an open source application. See the Glossary.

² In order to prevent lengthy installs with the same procedures being repeated on each machine, hard drive imaging software can be administered from within the network. This enables deployment with minimum user intervention and can save a lot of time. An example of such software is Norton Ghost, which many IT administrators already use to quickly deploy updates and fixes on networks. This is especially useful in scenarios where the administrator wants to be in complete control of each computer's installations.

that users be made aware that they are indeed running a high performing package and those comparisons to the previous software packages should be answered with well informed answers. This prevents users forming negative opinions of the new software. This requires highly skilled support staffs that are well versed in the feature set of the new software.

Creating positive sentiment, or instead getting key employees excited about the use of software also creates a culture of “viral” training where colleagues help each other, instead of always using support systems.¹ This can save tremendous costs and should be made an important reason for marketing new software products which are being deployed. Users who are made aware of shortcomings in the proprietary software model are almost certainly more open to the learning process involved with open source software. As said before, this sentiment can be very dependent on the initial attitude portrayed by the organizational management.

Checklist for this phase:

- ✓ Find technically skilled staff that have good people skills and who can help employees understand why new software is being used, making them well aware of the advantages.
- ✓ This maintenance period could indeed uncover some requirements which could not have been discovered earlier. Some requirements are so complex and difficult to capture that it is inevitable that they will be uncovered in this stage.² It is for this very reason that the support phase exists, and is considered a mandatory period in the development of a new system. Typical requirements which could creep in at this stage include the following aspects: maintainability, scalability, safety and efficiency.

¹ Salvaggio, S. 2006. *Open Source: a r/evolution in the software industry?* pg. 71

² Avison, Fitzgerald. 2002. *Information Systems Development: Methodologies, Techniques and Tools.* pg. 105.

- ✓ Extra costs need to be assigned to support services in this phase. Again, make sure adequate staff skill is available and, if possible, have professional support services on standby.

5.6 Phase 6: Re-assessment phase

After the support phase is winding down (it will never really end), it can be beneficial to analyse what was done well and what was less successful. A project of this size can require huge efforts and it will always be beneficial to have case studies for future endeavours, be they big or small. Records should be made in order to make more efficient changes or migrations in the future. It is at this phase also that cost analysis can be done in order to see whether cost projections have been met. If it was well planned it ought to be within budget without any complications. It can be argued that shortcomings can always be identified, which is the actual valuable part of well-formulated case studies.

Checklist for this phase:

- ✓ Keep a detailed journal of the entire process, entailing all phases described here.
- ✓ This journal needs to be cross checked in a meeting environment in order to make sure that all parties involved in the process's views are taken into account. This should not only be IT and management, but a solid sample of end-users who can give valuable feedback from their perspective.

With adequate planning and management support, a migratory process can go without too many problems. While no project of this size can commence without hitches, adequate planning is the key to success. One thing that can be proven by taking all these phases into account is that third party support can be beneficial and well worth the cost. It is for this very reason that commercial distributions of Linux are used by large organizations, instead of going for great cost savings of non-supported Linux distribution. Putting it in another way, support cost will be inevitable, be it in the form of professional support services or the higher cost of keeping skilled staff on hand. Highly skilled staff will indeed be more expensive in

the long run, making commercial distributions of Linux more viable for almost any organization.

Compared to operating systems, productivity software's deployment is faster, training is less and need for consultation and support will be the most important in the beginning of migration, for example OpenOffice.org was designed as a Microsoft Office clone, and general users have little if no trouble transitioning to it¹. The barriers are smaller, because the interface between Microsoft Office and OpenOffice does not differ greatly and file formats are compatible. It can be decided whether a file format shift might be necessary – for example, it can still be fine to use the proprietary Microsoft Word “doc” format instead of the open “ODF” format. Software migration is one area where planning is the key, and poor organization can lead to costly mistakes.

¹ Leete, Gurdy. 2006. *OpenOffice.org for Dummies*.pg. 15.

Conclusions and Future Research

South Africa is in the unique position that it is one of the nations in the world which can truly benefit from an open source initiative. Even though cost can be seen as the driving factor behind the adoption of open source software, our government and education sectors can benefit greatly from the other advantages open source software offers.

Open source's basic idea is quite old and relatively simple. It is any software that gives the user freedom to view and alter its source code. It originated in the seventies to combat proprietary software's dominance in controlling computer hardware, something which is still relevant today. Care must be taken not to confuse open source with free software. Open source has a few key contractual elements that differentiates itself from free software. In fact, one of the primary reasons for using the "open source" moniker is to prevent connotations with poor quality software, something that might be hinted at with the word "free".

Open source has a very intuitive method of development, requiring the efforts of many to contribute to a single cause. Instead of distinction between developer and user, open source encourages input from anyone who wishes to contribute, releasing the source code with any software application. There exists slight differences in the way some open source software is developed, each with its own advantages and disadvantages. Over time, different companies released their products each with a slight variation in the software licence. Eric S Raymond wrote a book on these differences, highlighting two different methods – called the "Cathedral" method and "Bazaar" method. The Cathedral method is very similar to proprietary software development (with exception for source code being released), whereas

the Bazaar method is a free for all principle where anybody can contribute at any stage within the development process.

One aspect of open source software which is benefited by either model is the Linus's Law.¹ Linus's law states that "given enough eyeballs, all bugs are shallow". This is due to many people viewing the source code, resulting in bugs being spotted early in the development process. This is considered the primary advantage to open source software, and the possible reasons behind the stellar security reputation. Proprietary software vendors are quick to point out that without source code deployment, there exists some "security through obscurity".

There exists many open source alternatives for almost any software application, and given enough research, cheaper and often free solutions can be found. Open source's current hype is mostly because of Linux, an operating system. Open source advocates claim that Linux is a viable alternative to Microsoft Windows. In order to see if these claims are true, feature comparisons need to be done between Linux and the two big names in operating systems – Apple Mac OS X, and Microsoft Windows. Linux is unique in that it ships in many different forms, and is made by many different vendors, each catering for a specific market segment. These different versions, called distributions, can also be modified in user interface, giving the end-user a large amount of choice in terms of functionality and ease of use. Although this might be an advantage for some, it is also perhaps the greatest barrier to Linux. Because so many variations exist, users cannot expect consistency on different Linux machines. Given the fact that the average Linux installation shares very few visual cues with Windows, this becomes an even bigger problem for people who are used to Microsoft's operating system.

A feature comparison will prove that all operating systems have specific advantages and disadvantages. Microsoft Windows is perhaps the most popular (which aids its networkability) but it is not necessarily the best. It is unstable and has a poor security record in the past, causing its server market share to plummet. Mac OS X is easy to use with a great feature set, high security and reliability, but Apple hardware is a fundamental prerequisite. Mac OS X requires Apple hardware to run, which can be costly, especially when making

¹ Linus Torvalds was the original creator of the Linux kernel, before giving it to the open source community.. Linus's law is better described in chapter 1.

direct hardware performance comparisons. Linux has a great feature set, high performance, great security record and low price. Its biggest barrier is the inconsistency of user interfaces across different distributions. Despite the relative ease of use of different graphical interface front-ends, traditional Windows users find it hard to migrate, especially without training.

The other key area where a true open source alternative exists is in productivity software. This is normally the territory of Microsoft Office, a very expensive productivity “suite” which has become the industry standard in terms of file formats and features. Word processing, spreadsheets and presentation software is relatively simple in concept, and is one of the most basic tasks a modern computer can do. It is for this very reason that sky-high prices for Microsoft Office seems unjustified, nudging users to look for substitute applications.

The best example of such an application is OpenOffice.org, an open source product that has a great feature-set, is easy to use and unlike Microsoft Office, is available for most operating systems. Its other greatest advantage is that users are given the choice of saving applications in a new open standard document format, called ODF. The reasons for this are plenty, most of which is the prevention of being “locked-in” by a proprietary software vendor, giving organizations freedom of choice at upgrade time.

Other areas where good open source alternatives exist include anti-virus software, email clients, web server software and internet browsers. In order to compare and decide which software is a good choice, Woods and Guliani proposed a maturity model¹ which needs to be assessed in order to choose good open source software. Mature software is software which is frequently updated, decently productized, and continually in development without periods of stagnation. A prime example is Mozilla Firefox, a secure, easy to use web browser.

The main reason why businesses are interested in switching to Linux is the lure of potentially lower costs. Management tends to see only lower licensing fees, giving a misconception of cost. With the migration to open source, total cost of ownership needs to be systematically assessed. Proprietary software TCO can be quite easily calculated through incentives such as

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 31

Microsoft's CAL system, requiring organizations to pay only per seat, server or mailbox. With the switch to open source, other factors become much more important.

The organization's original operating system platform is important because virtualization options exist for certain OS's. This can save a lot on unnecessary complete migrations to other operating systems and hardware platforms. Linux tends to have lower hardware requirements and is much more scalable depending on needs. One aspect which is especially important with regard to government sectors is specialized software (also called "vertical" software). This software is specially written for certain businesses, and can often still run on old legacy system. If a switch is made to another operating system, this software often needs to be rewritten which can carry significant financial costs, unless virtualization methods are employed.

Ferungal created a software TCO formula, which is still relevant to open source software. If used with OSS, certain aspects, however, require more emphasis. When using open source, support costs are key, and the pattern of use of employees is also more important. Other factors include management tools and current software stack being used. A lot of businesses use TCO as the primary reason for shifting to Linux. Examples of businesses saving costs include BMW's F1 division, GlaxoSmithKline, Google and Orbitz travel agency. Other bonuses include much higher performance and increased reliability / uptime.

Migration plans need to be approved by management, with Return on Investment (ROI) being one of the greatest aspects to consider. Projects of this scale require management support, which can be troublesome if a full picture is not painted. Many businesses also employ a "hurdle rate", requiring projects to have a certain projected percentage return on investment in order to be permitted. Woods and Guliani gave key aspects to calculate the implementation of open source ROI. These include the following costs: evolution, licensing and maintenance, installation and configuration, integration and customization, operations and support.¹

The adoption of open source does not have to mean that all software on a computer is open source. Often, the most effective way to run a machine is a combination of well-designed proprietary software and excellent open source software. This all depends on the needs of the

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 68

user. For example, computer literate home users might prefer to have a proprietary operating system installed with which they are comfortable. Microsoft Windows might offer great software compatibility, whereas Mac OS X is considered to have very easy to use interface.¹

It is for this reason why the use of OSS is well suited to the government sector. It lowers the cost of hardware acquisition, prevents vendor “lock-in” with file formats. Security is especially important in this sector, due to ongoing efforts to make voting and census collection processes more effective through the use of electronic means, lowering errors and excessive temporary labour.

Even though the point of this thesis is to remain objective and unbiased, it is necessary to give an opinion of which solutions are the best for different users. In the case of home users, Linux is not yet intuitive enough to be a true alternative to Windows. However, in terms of productivity software, OpenOffice is a very good alternative to Microsoft Office, and possibly even a better suited option. Education users need to be split into two divisions; schools and tertiary institutions. Schools in SA do not have the money to buy big computer centres with proprietary software, which makes Linux a viable solution. Schools need to make it their responsibility to teach learners the intricacies of open source software such as Linux, therefore shaping a future where people can be comfortable with more than one operating system. Proprietary operating systems are well suited for tertiary institutions, but open source productivity suites can be very helpful, especially in terms of “open” file formats (such as ODF) which do not bind intellectual property to specific file formats.

The government needs to mimic overseas initiatives (especially those created by the European Union) by crossing over to open source office suites in order to use open document formats. Currently a lot of money is being spent to buy computers with the Microsoft Office productivity software, which is seen as a “given” for compatibility of documents. Buying a computer with Microsoft Office increases the cost by about a third. OpenOffice has the same basic feature set, and has the added compatibility with OpenDocument, so there is very little reason for the South African government to keep using proprietary office suites.

¹ This was described fully in Chapter 2.

Taking the TCO and ROI cost factors into account, a relatively simple migration process model can be created. This model can be followed to give guidance in what aspects to consider throughout the migration process. Its framework and aspects are naturally malleable depending on use case and organization size.

Suggestions for future research:

Current research in this field is of a very biased nature, due to sponsorship by parties involved with the issue at hand. A Microsoft sponsored study will naturally not prove Linux a better solution. Same goes for any study sponsored by a Linux vendor.

I would recommend more thorough analysis of TCO and ROI cost factors. While these factors are generic in its current form, they can easily be skewed in order to maintain certain results. For example, in a recent Microsoft sponsored study¹, hardware cost comparison is made between different hardware platforms, with Linux running on very expensive high-grade server equipment, whereas Windows Server is running on relatively standard commodity hardware. What makes this even worse is that it is a general consensus that Linux requires less expensive hardware in order to perform adequately. The reader is not necessarily made aware of this, which can manipulate findings.

Researchers have to look beyond corporate sponsored studies and biased fanatical viewpoints. It is the single greatest barrier to objective research in the open source field, regardless of target users.

¹ Microsoft. Get the Facts: Total Cost of Ownership. 2006. *Microsoft Corporation*. Available also online: <http://www.microsoft.com/windowsserver/facts/analyses/tco.mspx> [02 October 2006]

Bibliography

- ALMARZOUQ; ZHENG; RONG; GROVER. Open Source: Concepts, Benefits and Challenges. 2007. *Communications of the Association for Information Systems* (Volume 16, 2005) pg. 756-784
- ANDREWS, J. 2005. *A+ Guide to Managing and Maintaining your PC*. Pg 1154. Thomson Course Technology.
- AVISON, FITZGERALD. 2002. *Information Systems Development: Methodologies, Techniques and Tools. 4th Edition*. McGraw Hill. pg. 78.
- BRIDGES.ORG. 2005. *Free / open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners*. p 1. [online] Available: http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf [September 2007]
- CALIFORNIA OFFICE OF GOVERNMENT COMMERCE. 2007. *Open Source Software Trials in Government Final Report*. Available online: <http://www.arb.ca.gov/oss/articles/report-v8d.pdf>
- DAY, B. & POGUE, T. 2004. Free / Libre & Open Source Software and Open Standards In South Africa: A Critical Issue for Addressing the Digital Divide, South African National Advisory Council on Innovation - *Open Software Working Group*, [online] Available: <http://www.naci.org.za/floss> [15 September 2006]
- DEPARTMENT OF PUBLIC SERVICE AND ADMINISTRATION. 2006. Policy on Free and Open Source Software Use for South African Government. Available online [May 2007]: http://www.dpsa.gov.za/documents/ogcio/2007/foss_policy_2006.pdf
- DI MAIO, ANDREA. 2003. Conference Can't Agree on Open Source's Role in Italy. *Gartner Research*.

- DI MAIO, ANDREA. 2005. Governments Move Toward Sounder Policies for Open source Software. *Gartner Industry Research*.
- DI MAIO, ANDREA. 2005. Local Governments in France Move to Open source Applications. *Gartner Research*.
- DI MAIO, ANDREA. 2005. Look Beyond TCO to Judge Open source Software in Government. *Gartner Research*.
- DRAKOS, N; DRIVER, M. 2000. Debunking Open source Myths: Origins and Players. *Gartner Research*.
- DRAKOS, N. 2003. Public Sector Needs Balanced Open source Software Policy. *Gartner Research*.
- DRIVER, MARK. 2001. The Future of Open source Software. *Gartner Research*.
- DRIVER, MARK. 2005. Hype Cycle for Open Source Software 2005. *Gartner Research*.
- EGOVERNMENT RESOURCE CENTRE. 2006. *Open source software archive for United Kingdom*. [online] Available: <http://www.egov.vic.gov.au/index.php?env=innews/detail:m1005-1-1-8-s-0:n-250-1-0--> [22 August 2007]
- EVANS, ALAN; MARTIN, KENDALL; POATSY, MARY ANNE. 2006. *Technology in Action. Third Edition*. Pearson Prentice Hall.
- FIERING, L., SILVER, M., SIMPSON, R. & SMITH, D. 2003. Linux on the Desktop: The Whole Story. *Gartner Research*. pg. 3
- FOLEY, MARY JO. 2005. Yankee Independently Pits Windows TCO vs. Linux TCO. *Microsoft Watch*. Available online: <http://www.microsoft-watch.com/article2/0,2180,1553620,00.asp>
- FREE SOFTWARE FOUNDATION. 2007. *GNU General Public License*. [online] Available: <http://www.gnu.org/licenses/gpl.html> [04 June 2007]
- FRIEDMAN, THOMAS. 2005. *The World is Flat: A Brief History of the Globalized World in the 21st Century*. Penguin Books.
- GO OPENSOURCE.ORG. 2005. OSS in South Africa. *Go Opensource*. [Online]. Available: http://www.go-opensource.org/oss_in_south_africa/ [20 November 2005]

- GNOME. 2007. What is GNOME? *GNOME* [Online]. Available:
<http://www.gnome.org/about/> [25 October 2007]
- GNU.ORG. 2007. GNU General Public Licence. *Gnu.org*. [online] Available:
<http://www.gnu.org/copyleft/gpl.html> [10 October 2007]
- GOOGLE. 2005. Define: Open Source. *Google* [online]. Available:
<http://www.google.co.za/search?hl=en&lr=&oi=definer&q=define:open+source+software&defl=en> [15 November 2005]
- GORHING, NANCY. 2007. Mobile Firefox likely to work on Symbian. *PC World*. [Online] Available: <http://www.pcworld.com/article/id,138375-c,mozilla/article.html> [14 October 2007]
- GREENEMEIER, LARRY. 2004. Why the Open Source Model can work in India. *Information Week*. [online] Available:
<http://www.informationweek.com/story/showArticle.jhtml?articleID=47900215&tid=5979> [22 August 2007]
- HANSCH, SUSUN; BERTI, JOHN. 2005. *Official (ISC)2 Guide to the CISSP Exam*. Auerbach Publications.
- HEID, JIM. *The Macintosh iLife '05 in the Classroom*. Peachpit Press.
- HEWITT, JEFFREY. 2005. Linux Making Strong Inroads in the Server Market. *Gartner Research*.
- HOFFMAN; HARPF. LiMux – free software in Munich. 2007. *Landeshauptstadt Munchen*. [Online] Available:
http://www.muenchen.de/cms/prod2/mde/_de/rubriken/Rathaus/40_dir/limux/publ/free_softw.pdf
- HUBLEY, M.I. 2005. Linux-Based Graphical User Interfaces: Perspective. *Gartner Research*.
- HUBLEY, M.I. 2005. Linux Operating System Distributions: Perspective. *Gartner Research*.
- HUBLEY, M.I. 2004. Linux: What Major IT Vendors Are Doing. *Gartner Research*.
- HUBLEY, M.I. 2001. The Pros and Cons of Open source Software and Linux. *Gartner Research*.

- JONES, JEFF. 2007. Windows Vista – 6 Months Vulneribility report. *CSO security*. [online]
Available: http://www.csoonline.com/pdf/6_Month_Vista_Vuln_Report.pdf
[20 October 2007]
- KRAMMER, MIKA YAMAMOTO. 2005. Costs and Benefits Still Favor Windows Over Linux Among Midsize Businesses. *Gartner Research*.
- KRISHNAMURTHY, SANDEEP. 2003. A Managerial Overview of Open Source Software. *Business Horizons*.
- LANDAU, TED. 2006. *Mac OS X Help Line Tiger Edition*. Peachpit Press.
- LAUDON, LAUDON. 2006. *Management Information Systems: Managing the Digital Firm*. 9th Edition. Pearson.
- LETTICE, JOHN. 1999. French Senators propose making open source compulsory. *The Register*. Available online:
http://www.theregister.co.uk/1999/10/24/french_senators_propose_making_ope_n/
- LE TOCQ, CHRIS. 2001. Sun Starts OpenOffice Project With StarOffice Source. *Gartner Research*.
- LEETE, GURDY. 2006. *OpenOffice.org for Dummies*. Wiley Publishing.
- LIBRE SOFTENG. 2006. Geographic Location of (over one million) SourceForge Developers. *Sourceforge*. [online] Available: <http://libresoft.es/libresoft/68>. [25 October 2007].
- LINZMAYER, OWEN W. 2006. *Desktop and Portable Systems Second Edition*. Apple Training Series. Peachpit Press.
- LIU, L; HAYWARD, B.M. 2003. *Open Source Can Be a Catalyst for IT in China*. *Gartner Research*.
- LOGAN, D. 2004. *New EU States Will Add to Demand for Open source Software*. *Gartner Research*.
- MARGULIUS, DAVE. 2003. The Real cost of switching to Linux” *Infoworld*. [online]
Available:

http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html [10 October 2007]

MERAKA INSTITUTE. 2007. Vision, Mission and Strategy: *Open Source Centre*. [online] Available: <http://floss.meraka.org.za/?q=about/osc/vms>. [22 August 2007]

MICROSOFT. Get the Facts: Total Cost of Ownership. 2006. *Microsoft Corporation*. [Online] Available: <http://www.microsoft.com/windowsserver/facts/analyses/tco.mspx> [02 October 2006]

MILLER, ROBYN. 2006. How OpenOffice 2.0 stacks up against Microsoft Office. [online] Available: http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci1174145,00.html [5 June 2007]

MOHAMED, ESSACK. 2004. A Knowledge Approach to Software Testing. *University of Stellenbosch*.

MOZILLA ORGANIZATION. 2007. *Firefox Awards*. [online] Available: <http://www.mozilla.com/en-US/press/awards.html>. [14 October 2007]

MOZILLA ORGANISATION. 2007. Phishing Protection: Design Documentation. [online] Available: http://wiki.mozilla.org/index.php?title=Phishing_Protection:_Design_Documentation&oldid=46996#Overview [14 October 2007]

NET APPLICATIONS. 2007. Top Browser market share trend for October 2007. [Online] Available: <http://marketshare.hitslink.com/report.aspx?qprid=3> [14 October 2007]

NICOLAISEN, THOMAS FERRIS. 2006. The Use of Open Source and Open Standards in Web Content Management. *Proceedings of The 2006 International Conference on Information and Knowledge Engineering*. p. 250

OLIFER, NATALIA; OLIFER, VICTOR. 2005. *Computer Networks: Principles, Technologies and Protocols for Network Design*. John Wiley & Sons, Ltd.

- OPEN SOURCE INITIATIVE. 2007. Open Source Definition. *OSI*. [online] Available:
<http://www.opensource.org/docs/definition.php>. [18 October 2006]
- OPEN SOURCE INITIATIVE. 2006. MIT Licence. *OSI*. [online] Available:
<http://www.opensource.org/licences/mit-licence.php> [10 October 2007]
- PAYNE, CHRISTIAN. 2002. On the security of open source software. *Info Systems Journal*.
Vol 12.
- PEGARARO, ROB. 2005. Office Suite without the Sticker Shock. *Washingtonpost.com*
[online]. Available: <http://www.washingtonpost.com/wp-dyn/content/article/2005/11/12/AR2005111200143.html> [20 January 2006]
- PERENCE, BRYAN. 2005. The Emerging Economic Paradigm of Open Source. *Cyber Security Policy Research Institute*
- PESCATORE, J. 2000. Is Open source More Secure Than Proprietary Software? *Gartner Research*.
- PHYSORG.COM.. 2007. Engineer Creates First Academic Playstation 3 Computing Cluster. *NC State University* [Online] Available:
<http://www.physorg.com/pdf92674403.pdf>
- PREIMESBERGER, CHRIS. 2004. Four out of four experts agree: Linux lowers TCO. *IT Manager's journal*.
<http://www.itmanagersjournal.com/management/04/06/04/2114222.shtml>
- PRENTICE, BRIAN. 2005. Apple vs. Novell: An Unlikely Battle for SMB Open source Relevance. *Gartner Research*.
- PROBABILITY SERVICES. *Why use Open Source?* [online] Available:
<http://www.probabilityservices.co.uk/index.php?/content/view/9/29/>
- RAMBØLL MANAGEMENT. 2006. Estimates of the Costs of Implementing Open XML and Open Document Format in the Central Government. *Rambøll Management*.
[online] Available:
http://www.odfalliance.org/resources/Denmark_OSL_rapport.pdf

- RASCH, CHRISTOPHER. 2006. *A Brief History of Free/Open Source Software Movement*
[online] Available: <http://www.openknowledge.org/writing/open-source/scb/brief-open-source-history.html> [10 October 2007]
- RAYMOND, E.S. 1999. *The Cathedral & the Bazaar*. Thyrus Enterprises. [online]
Available: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>
[10 October 2007]
- SALVAGGIO, S. 2006. *Open Source: a r/evolution in the software industry?* [online]
Available: www.cetril.org/fileadmin/pdf/04-06-01-OpenSource.pdf
- SATZINGER, JACKSON. 2006. *Systems Analysis & Design in a Changing World*. 4th
Edition. Thomson Course Technology.
- SEARLS, DOC. 2005. *Getting Flat, Part 1*. Linux Journal. [online] Available:
<http://www.linuxjournal.com/node/8251/print> [10 October 2007]
- SEN, RAVI. 2007. A strategic analysis of competition between open source and proprietary software. 2007. *Journal of Management Information Systems* / Summer 2007. p. 233.
- SCOTT THOMAS, R. 2006. *Quartz 2D Graphics for Mac OS X developers*. Addison Wesley.
- SMITH, D.M. 2003. CIO Update: Microsoft vs. Linux and the Changing Nature of Competition. *Gartner Research*.
- SILVER, MICHAEL A. 2005. Examining Where Desktop Linux and Open source Office Products Make Sense. *Gartner Research*.
- SILVER, MICHAEL A. 2005. Linux Desktops Yet to Reach Potential in Emerging Markets. *Gartner Industry Research*.
- SILVER, MICHAEL A. 2005. Management Update: Enterprisewide Open source Office Adoption Will Be Difficult. *Gartner Industry Research*.
- SILVER, MICHAEL A. 2005. The Myths About Desktop Linux Persist. *Gartner Industry Research*.
- SILVER, MICHAEL; Rust, B. 2005. Choosing Desktop Linux helps High Schools Buy More PCs. *Gartner Research*.

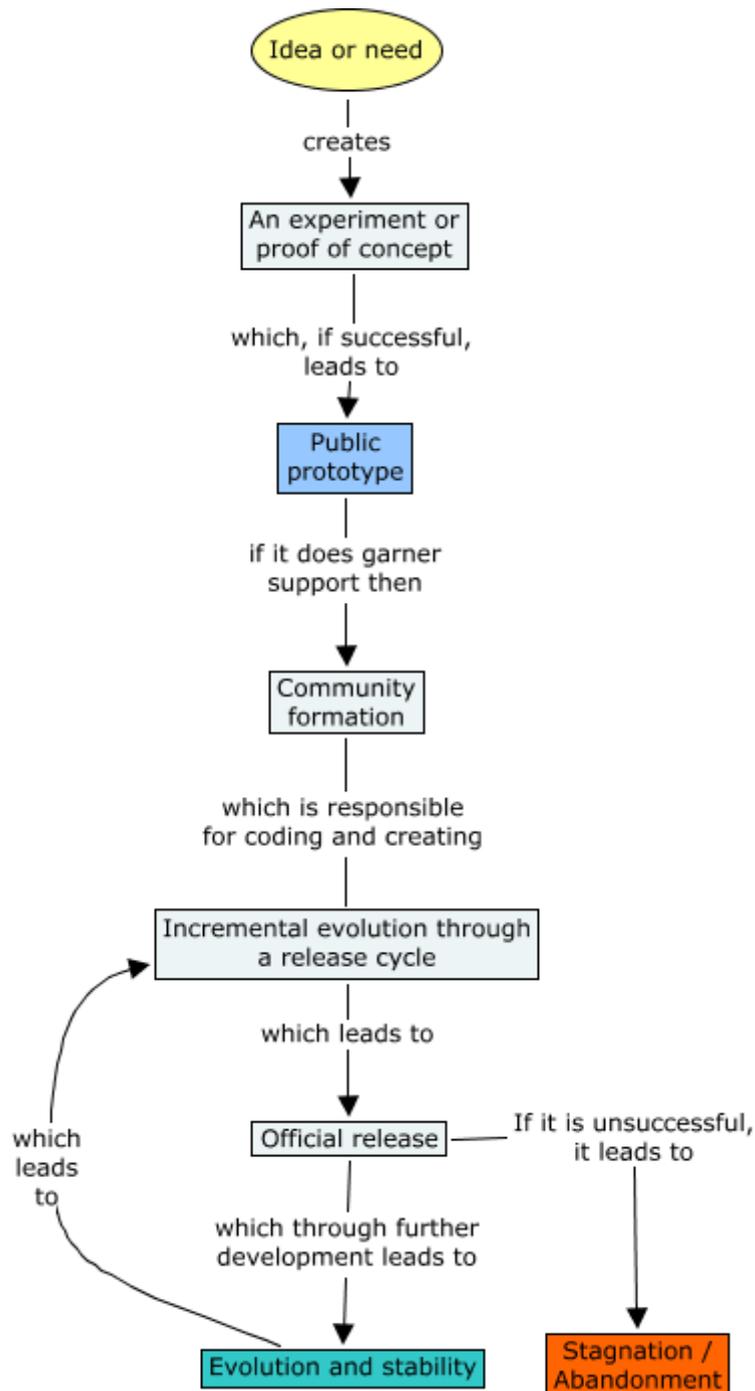
- SMITH, D.M. 2003. A Look at Alternatives to Microsoft. *Gartner Research*.
- VALDES, RAY. 2005. Portal Market Opens Up to Open Source, Slowly. *Gartner Research*.
- WISE, DAVID A. 2005. *The Google Story: Inside the Hottest Business, Media and Technology Success of our Time*. Delacorte Press.
- WEISS, GEORGE J. 2002. Assessing Our Predictions about Linux and Open source. *Gartner Research*.
- WEISS, GEORGE J. 2005. Predicts 2005: Open source Software Proliferates. *Gartner Research*.
- WEISS, GEORGE J. 2001. The Future of Linux and Open source. *Gartner Research*.
- WEISS, GEORGE J. 2005. Xen Hypervisor Shows Promise, but Challenges Loom. *Gartner Research*.
- WIKIPEDIA. 2006. Open Source. *Wikipedia.org* [online]. Available:
http://en.wikipedia.org/wiki/Open_source [25 November 2006]
- WIKIPEDIA. 2006. Open Source Software. *Wikipedia.org* [online]. Available:
http://en.wikipedia.org/wiki/Open_source_software [25 November 2006]
- WIKIPEDIA. 2006. Emulator. *Wikipedia.org* [online]. Available:
<http://en.wikipedia.org/wiki/Emulation#column-one#column-one> [25 November 2006]
- WILLIAMS, SAM. 2002. A Timeline of Open source in Government. *Linux Dev Center*.
[online] Available:
http://www.linuxdevcenter.com/pub/a/linux/2002/07/15/osgov_timeline.html
[10 October 2007]
- WOODS, DAN; GULIANI, GAUTAM. 2006. *Open Source for the Enterprise*. O'Reilly.
- WORD AUTOMATION. 2005. Comparison Guide Between OpenOffice.org and Microsoft Office. *Word Automation*. Available:
<http://www.wordautomation.com/openoffice.html> [30 October 2006]
- WU, JENNIFER; LEE, C.G. 2005. Success of Linux in South Korea Rests on School Network Plan. *Gartner Research*.

YANOSKI, RON. 2003. Client issues: Higher Education Applications and Beyond. *Gartner Research*.

YANOSKI, RON. 2003. Higher Education E-Learning meets Open Source. *Gartner Research*.

ZULIANI, PAOLO; SUCCI, GIANCARLO. 2005. An Experience of Transition to Open Source Software in Local Authorities. *Free University of Bolzano-Bozen*

Appendix B: *The Open Source Life Cycle*



The open source life cycle (adapted from Woods and Guliani)¹

¹ Woods, Dan; Guliani, Gautam. 2006. *Open Source for the Enterprise*. O'Reilly. pg. 17

Appendix C: Interviewees for insight into South African FOSS strategy:

Lion Steynberg

Director: SITA Oversight

Office of the Government Chief Information Officer

Department: Public Service and Administration

Aslam Rafee

CIO, Dept of Science & Technology

Chair of the Government Information Technology Officers Council OSS Working Group

Maria Farelo

Director; ICT & e-Government Strategy, Policy & regulations, DPSA

Bob Joliffe

IT Dept, Dept. of Science & Technology

Karl Fischer

IT Dept. of Science & Technology, leading FOSS migration at DST

Nhlanhla Mabaso

Chair of the FOSS Programme Office Steering Committee

Manager of the Open Source Centre, Meraka, CSIR.