Euro-India
ICT Co-operation
www.euroindia-ict.org

Euro-India ICT Cooperation 2nd International Event
10 – 11 December, 2009
International Management Institute, New Delhi, India

Call for Papers

Session: Innovation in Organisations

Title: Open Source Adoption in South Africa by Organisations: An Exploratory Study

Authors

Ilse Van Beulen
University of Cape Town

Jean-Paul Van Belle[1]
University of Cape Town
Jean-Paul.VanBelle@uct.ac.za

Margam Madhusudhan
University of Delhi
mmadhu@libinfosci.du.ac.in

[1] Contact Author

Abstract
The research looks at the current adoption trends in South Africa with a view to uncover how South African companies went about adopting OSS, the factors that influenced adoption and the factors that contribute to sustainability of OSS post adoption. The research is done in response to the lack of any roadmap or plan which can guide companies wanting to adopt OSS to use it profitably and gainfully, i.e. to guide successful adoption and implementation of OSS. To aid the process of discovery a number of theoretical frameworks and models are utilized with the aim of ascertaining the usefulness of these theoretical frameworks in its guidance of OSS adoption. The main aim of the paper is to see if these models be gainfully employed by other companies embarking on OSS adoption, and any additional factors which may guide successful OSS adoption. It hopes to deliver more insight into the "roadmap" which companies can use to successfully adopt OSS.

Keywords: Technology, Organisation and Environment (TOE), OSS Adoption, Open source Maturity Model, Open Source Skills and Risk Model, Cost (including Total Cost of Ownership).

Introduction

Open Source Software (OSS) is defined as those software programs whose licenses give users the freedom to run the program for any purpose, to study and to modify the program and to redistribute copies of either the original or modified program (Nepelski & Swaminathan, 2007).

Open source infiltrates markets at different levels of commercial influence and in varying timeframes and the strategies used with success by technologically aggressive early adopters are infrequently appropriate for conservative organisations (Driver, 2007).

This paper explores the current adoption trends in South Africa with a view to uncover how South African companies went about adopting OSS, the factors that influenced adoption and the factors that contribute to sustainability of OSS post adoption, and how supportive the OSS models are of successful adoption. Thus our research objectives are two-fold:

1. Investigate the factors driving initial and continuing OSS adoption by SA businesses.
2. Look at the usefulness of a number of theoretical frameworks to guide the above.

The research is important from a number of perspectives. It offers practical insights to organisations wanting to or considering adopting OSS. From an OSS industry perspective, it investigates the impacts and effects on developers and 'resellers'. And to the research community, it sheds a perspective on how various theoretical frameworks and models fared i.e. how useful they are, how comprehensive they are and how supportive they are of the findings of the research.

The OSS adoption within the South African context will be ascertained within the Technology, Organisation and Environment (TOE) framework, and, in turn, rely on a number of other OS models, as portrayed in Figure 1.

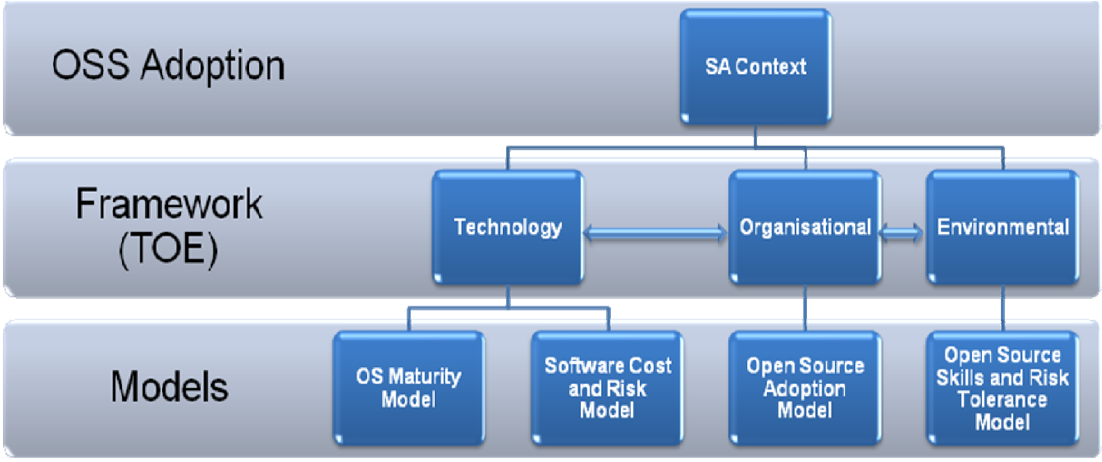
OSS Adoption
SA Context
Framework (TOE)
Technology
Organisational
Environmental
Models
OS Maturity Model
Software Cost and Risk Model
Open Source Adoption Model
Open Source Skills and Risk Tolerance Model

Fig.1- Technology, Organisation and Environment (TOE) framework for OSS adoption

TOE attempts to comprehensively cover all the aspects of successfully applying open source in companies as portrayed in figure 1. Within each concept of the TOE framework various models will be employed (and discussed) to aid the discovery process. The Open Source Maturity Model ascertains the maturity of the open source at the time companies adopt, the Software Cost and Risk Model looks at the Total Cost of Ownership of adopting Open Source, and these will be used in support of the Technology factor. The Open Source Adoption Models suggests factors within the Organisational context the company finds itself in at the time of adoption which affect adoption and the Open Source Skills and Risk Tolerance model looks at the skills and skill level required both internally and externally which influence adoption. These models will not be the exclusive factors discussed in support of TOE, there are additional factors that will be discussed under TOE.

Literature Review

The Technology, Organisational and Environmental (TOE) framework identifies three contexts in which an organisation functions that influence its ability to adopt technology and affects the process by which it accepts a new technology (Lippert et al., 2006) and implements a technological innovation (Zhu, Kraemer, Gurbaxani, & Xu, 2003). TOE represents a comprehensive method for ascertaining the features that form technology adoption in an array of information system innovations. These three groups of contextual factors influence the organisations intent to adopt an innovation, and affect its assimilation process and eventually its impacts on organisational performance (Zhu et al., 2003). TOE model defines a "context for change" consisting of three elements which interact with each other to influence technological innovation decision making. It is a useful analytical tool for distinguishing between inherent qualities of an innovation itself and the motivations, capabilities, and broader environmental context of adopting organizations (Dedrick et al., 2004). The review of related literature is presented on different aspects of subject, such as (i) Technology, (ii)Organisational, and (iii) Environmental, as follows:

Technology

The Technology context refers to the internal and external technologies available to the organisation which has a bearing on its productivity (Lippert et al., 2006) and encompasses the existing technologies in use within the organisation and the relevant technologies the organisation can draw on externally (Zhu, Kraemer, Gurbaxani & Xu, 2006). They are comprised of elements Open Source Maturity and Technology readiness. (Dedrick & West, 2003) adds five technology factors namely Compatibility, Complexity, Relative advantage, Trialability and observability citing compatibility with existing technologies, relative advantage over current technologies and complexity negatively influencing adoption as being the three most common variables linked to technology adoption.

Relative advantage is the measure of how much the new technology is relative to the existing one and is primarily measured in terms of cost and reliability (Zhu et al., 2006). The Software Cost and Risk model determines the cost and the risks of using open source (Guliani & Woods, 2005). Failure to optimally manage the potential risks and rewards of open source will put IT organisations at an increasingly serious risk in the coming years (Driver, 2007).

The open source maturity model attempts to measure these criteria, but more importantly serve as a guideline for companies wishing to adopt open source i.e. it serves to highlight possible pitfalls. It offers a rigorous set of questions that companies may use to measure the maturity of open source, with the maxim that the more mature the open source the better the chances of success. It allows companies to evaluate if they are capable of withstanding the risk of any measurement falling short.

Organisational

The Organisational context is characterised by a few descriptive measures e.g. scope, size of the organisation, the slack resources available internally. Organisational factors are comprised of different elements. Organisations have different competitive positions and roles for IT, and a high level of IT intensity is proportional to open source adoption (Kwan & west, 2004). The innovation orientation of an organisation is related to the timing of adoption and the prompts pertinent to adoption decision (Dedrick et al., 2004). The centrality of IT to the buisness strategy is core to the willingness of the organisation to adopt open source (Dedrick et al., 2004).

Choice set and Selection occurs as a response to Software Adoption policy, but more importantly occurs within the Application Context which exhibits the strategic significance of the specified system and consequently the equivalent weighted value for features, risk, cost and available products where the predilection of the buyer is restricted by a limited number of available choices (Kwan & West, 2004).

Environmental

The Environmental context refers to the arena in which the organisation operates and conducts its business (Zhu et al., 2003). (Lippert et al., 2006) contends that the organisation is influenced by the industry itself and its competitors. Environmental factors encompass factors such as rivalry, relations with buyers and suppliers (Zhu et al., 2006).

The Open Source Skills and Risk Tolerance model ascertains the propensity of the organisation to handle the risks intrinsic in open source adoption and produces a risk tolerance plan and profile. The relationship between skill and experience with value is directly proportional, where the higher the skill proficiency and the greater the experience a company has with open source, the higher the prospective value of open source (Guliani & Woods, 2005).

A higher skill set further reduces the time investment and the cost of using open source (Guliani & Woods, 2005). ICT training programs are directed on teaching students with skill in the most familiarly used proprietary software packages e.g. Microsoft and this has a repercussion in terms of the skill set available to OSS (Bruggink, 2003). This is exacerbated by the fact that few certification programs exist for computer and network support professionals wanting to specialise in open source software (Bruggink, 2003). Barriers to the successful adoption of open source are the lack of resources and/or the availability of external technological resources as well as the lack of compatibility with current technologies and skill (Holck, et al., 2004). However, developer skills may be improved by the intellectual challenge of contributing to software development when they are granted access to source code (Comino, Manenti & Parisi, 2007).

The Open Source Skills and Risk Tolerance model breaks the skill level into four tiers namely beginners, intermediate, advanced and expert and then further associates them to areas in which they are required to operate and be proficient and expands the level of proficiency required by each skill level in each area.

Research Methodology

A deductive, explanatory and qualitative research approach was taken. The qualitative approach was motivated because of the interdependence between variables and the non-measurability/intrinsic complexity of some of the variables. The aim is to provide richer and more subtle explanations than statistics can provide.

The instruments to collect data were questionnaires, documentary analysis and interviews. The sources of data used in documentary analysis included minutes of meetings, internal reports, briefings, planning documents, schedules etc. The Interviews were guided by a set of criteria so as to ensure the full scope of information is extracted.

The data analysis was based on concepts borrowed from grounded theory, the three stages and what they entail is depicted in Figure 2. Stage one, Open Coding refers to dis-aggregation of data into units. Stage two, Axial coding refers to the process of recognizing relationships between categories and stage three, Selective Coding refers to the integration of categories to align with the research purpose.

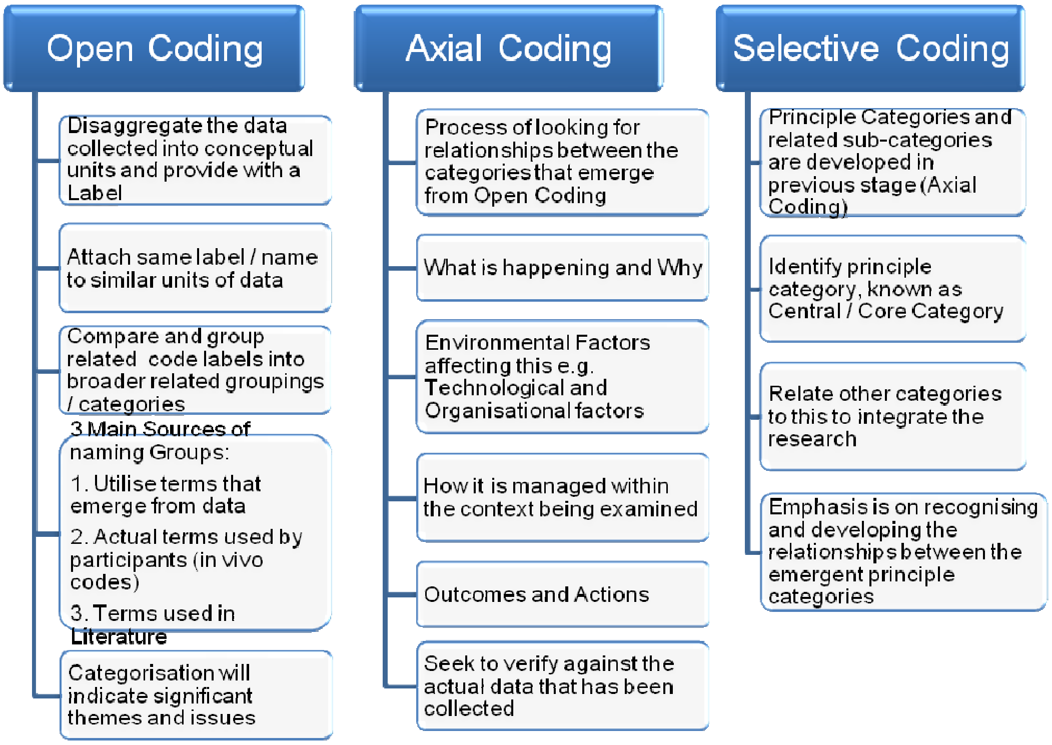
Open Coding
Disaggregate the data collected into conceptual units and provide with a Label
Attach same label / name to similar units of data
Compare and group related code labels into broader related groupings / categories
3 Main Sources of naming Groups:
1. Utilise terms that emerge from data
2. Actual terms used by participants (in vivo codes)
3. Terms used in Literature
Categorisation will indicate significant themes and issues
Axial Coding
Process of looking for relationships between the categories that emerge from Open Coding
What is happening and Why
Environmental Factors affecting this e.g. Technological and Organisational factors
How it is managed within the context being examined
Outcomes and Actions
Seek to verify against the actual data that has been collected
Selective Coding
Principle Categories and related sub-categories are developed in previous stage (Axial Coding)
Identify principle category, known as Central / Core Category
Relate other categories to this to integrate the research
Emphasis is on recognising and developing the relationships between the emergent principle categories

Fig. 2 - The Stages of Coding borrowed from Grounded Theory

Purposive or Judgmental sampling enabled the selection of cases which are particularly informative to the research. Four companies were selected based on the fact that these companies have implemented and are still using open source software (see table 1).

Table - 1: Company Sampling Segment

Company	No of OSS Packages	No of Employees	IT Size	Maturity of company	Nature of the business
IDS	40 to 50	180	6	11 years	Payroll Bureau. Services clients in South Africa.
Outprosys	20 to 30	80	3	8 years	Outsourced document processing company. Services clients locally and internationally.
Fundamo	40 to 50	75	40	8 years	Development house. Provide solutions to Banks in South Africa.
3i Solutions	30 to 40	90	3	7 years	Call centre. Services clients locally and internationally.

Within the companies, we strived for a sampling strategy which would add depth and commonality across the case studies. The people interviewed included the CEO, CIO and / or IT Manager (dependant on how these roles are defined), IT team (developers, testers and infrastructure team). The latter IT team group will also prove to be users in some cases. The sampling segment is portrayed in table 2.

Table- 2: People Sampling Segment

OSS Experience	Position	Examples of OSS packages used
1 ½ years	Developer	Python; Jasper (reporting); MySQL and PostgreSQL; Sunglass Fish ; Eclipse; ANT; Apache; Subversion ; Trac (source code); Thunderbird; Mozilla Firefox; Linux ; JIRA (open source bug tracking system) ; Hyperic (monitoring system).
7 years	Managing Director / CIO	
8 years	Solutions Architect	
4 ½ years	CIO (Exco)	
7 years	Developer	
7 years	Developer	
7 years	IT Manager	
5 years	Managing Director	
4 years	Managing Director	

It should be noted that the OSS experience does not reflect the overall OSS experience of the employee, but the number of years of OSS experience within their current roles. Many of these people have had prior OSS experience within other organisations or have had personal OSS experience.

Findings of the Study

Technology Factors

Access to Source Code

Consistently companies do not modify the source code. Across the board companies feel that they are more self reliant through access to the source code. This is evident in several areas. It is a mitigating factor in response to a lack of documentation. In addition their dependence on external parties is decreased by access to source code…by looking at the source code "we can fix an obscure bug ourselves".

With regards tailoring the code to meet an organisational need by adding features not provided for or improving on existing features there was only one reported case "Added functionality that did not exist in application.

Generally the respondents were loath to change the source code because of skill "don't want to mess with a working piece of code". There seems to be a bit of fear around the modification of source code. "No…couple of times with python source code, when it does not work the way you want it to, tweak it slightly….I am not daring enough to modify greatly". "It depends on the code. Which language it was written in and how complicated it is. If it can be changed, it will be changed".

The attitude seems to be closely correlated to skill set…the more hard core "I am an open source advocate", the more exposed (in years) and the greater the skill set, the less afraid the person is of modification. This does relate to attitude because high skill sets in other companies that are quite equipped to modify source code but are quite impartial to OSS do not modify.

Forking / Loss of Commonality

There was only one case reported. “The project had stopped because they started up an equivalent project so they had stopped contributing the project and we were using the older one and it was difficult for us to sort it out. The community was still working on it but on a later version”. The net effect was that “We had to change source code”.

In response to the additional question about incorrect choice the respondent replied “for all OSS components that we plug into our system we make sure we interface to them in a standalone patch with something else. We try not to have stuff running right through our system unless we know. We patch with a piece of glue e.g. login package log for J from Apache , we don’t call it directly, we call a Java standard framework through standard API’s, which by default already have the glue and therefore we don’t change the code, we merely have to change the glue. We architect our systems in such a way that we limit the impact of having to swop software e.g. there may be a licensing issue and you have to take it out immediately…then we are in a position to take it out quickly”.

Generally companies do not experience a loss of commonality. “In our experience e.g. Linux packages, the good ones are pulled into other distributions…it forks but it comes back together again. Or rather the ones that survive come back together again. It may have spawned 10 diff forks but the other two have been adopted , 8 fall by the wayside…it’s a stage, a phase it goes through. The net effect is that “at that stage where there are 10 different options it is very difficult to make a choice. Choosing the right option becomes difficult. You need to wait until later when it looks like 2 are leading the way… then the choice becomes easier”.

Complexity

Consistently OSS does have a degree of difficulty in terms of utilization and understanding the adopted technology. “Configuration and fit to applications or infrastructure…to slot in any piece of software that someone else has written you need to understand how to call it and how it will fit into your system – often that piece of glue that you need to write is tricky.”

“Complexity is added where three quarters of the software uses the same utilities but there are different versions…which version you use”. “Finding out how it works is difficult. I struggled to use Pentaho, I learnt how to use it and that has made the difference”. “Most Open Source tools allow you to configure things to your needs” “Complexity always influences the adoption process, be it Closed or Open software”.

Complexity for OSS is generally mitigated by trialability, the support available in the OSS community and the general attitudes of the developers…” They make it work because they want it to work”.

Cost

Organisations can download software free of charge from the internet and there are no license fees attached to the software (Russo, Zuliani & Succi, 2003). Direct cost savings can be gleamed from cheap or free software while indirect cost savings are derived from lower hardware requirements and nonexistent upgrade fees (Holck, et al., 2004). Table 3 presents the literature on the cost model as proposed by Guliani & Woods (2005) and its findings.

Table 3. Literature on the Cost Model (Guliani & Woods, 2005) and the Cost Findings

Type	Literature on Open Source	Literature on Commercial	Findings
Evaluation Cost	• More time cumbersome than commercial • Onus on company to evaluate • To really understand what an open source project can do, an IT department must install it and play with it which requires time and different resources • developers and architects take the lead • cost of creating a trial installation • cost of learning the software to play around with it and understand its functionality to determine if it can meet the company's requirements • It does remove a significant amount of risk from the process • result in a deeper understanding of both the software and the requirements	• Pushed by vendors who produce white papers, marketing materials, conferences, proof of concept • commercial software evaluation might take place with the sales staff or with IT managers who screen the products, bringing in the architects or developers later only if the product looks promising • Cost include: ○ Negotiation of a trial license that might have a time limit ○ Payment of service fees to support the trial installation ○ Training ○ Trial licenses for additional related software	• The Cost of downloading and playing in terms of resources and money is generally not measured outright. "I don't know as it has never been measured" • "they (vendors) will show me - I prefer to buy than to be sold to because anybody that is coming to sell me something, as got a vested interest in me taking the product, it's a commercial thing to them. They want me to take the product, whether the product fits into our needs, or not. But at the end of the day, they can show me 50 million demo's, I still need to evaluate the product on my own". • "I Spent a good while looking at the tool and when using my salary to measure justification it is still cheaper versus licensing and having a consultant". • "Chasing after OSS solutions are more cost effective". • "Time spent to understand / evaluate / how to use and training on both OSS and proprietary – may be quicker with proprietary". • "Evaluation costs the same- must do due diligence on both. For this OSS is quicker because you get answers faster than having to contact vendor or having techie investigate". • "OSS more expensive that proprietary…no, I don't think there is a difference. I don't think it is true". • "Switching Costs: the utilisation of OSS has been gradual and as such we saved costs because the licensing costs attached to the proprietary software that we have decommissioned have fallen considerably" • "learning costs: we hire external consultants, but we have to do that if we use proprietary because we build software therefore we put in smaller pieces of OSS" • "Hyperic monitoring system. We bought the OSS one and the cost was similar to proprietary solution. You still need support because you end up having to deal support yourself internally". • "Can I correct some of the statements. For example: Open Source apps is more time cumbersome than commercial. I find this to be a general conception with people spending 10 years working on MS Windows and then complain that OSS is too difficult after one week of using it. They are not the same and you cannot compare experience in one to experience in the other." • "Configuring the software takes longer in OSS?…ask that question to a SAP user with a consultant at R3000 an hour and 3 years later they still have the consultant on site". • "Configuring the software generally takes longer for Open source because the configuration is usually in a text file which must be edited by hand". • "TCO…I've heard that argument to death already". The defensiveness stems from what they perceive to be "untruths are being told about OSS costs" and that they are tired of these constantly being perpetuated as a truth. • "Depending on the administrators skills. Cannot throw a Windows administrator infront of a nix box and expect him to be able to do evaluation like a skilled nix administrator". • "Installation of software in our standard software repo (currently 18733 packages) takes under 20 seconds on average. Open Documentation and HowTo's in the form of Wikis, mean that anyone that solved a problem, shares it with the community. Documentation grows faster when
License and Maintenance Costs	• no maintenance fee for open source upgrades • subscription fees for Linux distributions • licensing fees for documentation	License and maintenance fees for commercial products can range from as low as a few percentage points to as much as 20 points or more of the total solution	
Installation and Configuration Costs	• Time consuming (cost inherent in the resources and opportunity cost) • Configuring the software generally takes longer for Open source		
Integration and Customisation Costs	The cost of customization to extending the software to meet requirements	The need and degree of customization is less	
Operations and Support Costs	Licensing Fees Not Applicable	Licensing fees required for: • Creating a development environment on each developer's workstation • Creating a test environment or a staging environment • Adding servers for scalability • Adding servers for disaster recovery or for a hot	

		backup site	open”.
The Cost of Narrowness	Burden is on the IT department to develop or find the skills to evaluate, install, configure, operate, and support the software	Burdens can take less time and cost more money The choice of possibilities is narrower and is limited to the common needs of the marketplace the vendor should support	• “Evaluation ourselves result in a deeper understanding of both the software and the requirements. Who do you trust most on software evaluation? You local IT ninja or Sales rep”. • “Installation is a non issue. New server installation less than 10minutes. Additional software less than 20 secs per package. Doing trail runs and installations, this is a huge benefit”. • “The cost of customization to extending the software to meet requirements is higher for OSS. The point here is, you can customise. Closed software usually does not allow customization of their software. If they do, it is limited to what they approve of, taking control away from the client”.
Switching Cost			
• Transitory transaction costs			
• Learning Costs			
• Contractual Costs			

Reliability

Across the board the most salient factor regarding OSS is it is “very reliable”. Reliability is also a major factor in turning people around from skepticism to converts “Skepticism overcome by the fact that this was a reliable piece of software”

Security

The general perception and view of OSS community are that these people are extremely hard working to the point of near extremists…generally we found that the trust level companies have is incredibly high with regards security because of experience with the software and because of the development life cycle.

Compatibility

Compatibility is the extent to which the new technology can co-exist with existing technology such as legacy systems. Compatibility is not an issue for OSS “a very famous trick of Microsoft which they've just done this year again, as they normally do, is we no longer support XP, we no longer support Windows 2000, and they force you to upgrade if you want the support. That never happens in the Open Source Community” “There are still people using Linux (Santos) 3 which we used about, I don't know, the last time I used Santos 3, was probably about five years ago, six years ago and it's never out of date. It's always out there and there's always someone in the world that's using it and you can always access it, it's free and you can download it whenever you want to. Nobody says to stop using this now." “In most cases we can swop out it (incompatible components) fairly quickly to replace them”.

Trialability

Trialability enables the organisation to learn and to understand the functionality of the software and they will not adopt in the absence of trialability. “Trialability is a major factor, always in top list when looking…”. The nice thing about OSS is that one can have a trial run of software without any restrictions. Most closed application trials cripple the software in some way.

OSS is inaccessible to companies because vendors and resellers of open source do not have a significant or established presence in Africa (SA) and because the internet is subject to unreliable connections coupled with the high cost of the internet…"Full free versions are not an issue", even when taken in the context of internet access and stability. No companies reported any accessibility concerns "The problem in accessing OSS extends only to how difficult is it to access the internet…since all companies have access to the internet this is a non entity". "If the software is included in our distribution repository, it is as trivial as a single install. Else it is a matter of downloading the package from the developers' site. All OSS is the full and free version".

Observability

Observability does not seem to impact the organisation to adopt or not to adopt because most of the OSS is not GUI related or using desktop applications. "business (users) will give me their input, but adoption is my call…" "We standardized on thunderbird….we don't use outlook – after initial moaning by the users because it does not look like outlook, they became used to it. It (the software) does what it is supposed to do. If it does more great, but only if it does less then it becomes problematic. We converted everybody whether they wanted to or not".

Open Source Maturity Model Matrix

Companies should be vigilant in identifying projects that have been dormant and that have not been updated for years (Guliani & Woods, 2005) and projects that have been registered but have been abandoned (Comino, et al., 2007). The factors indicative of this are forums portraying little or no interaction amongst developers, having recorded no bugs, patches or feature requests since their registration. (Comino, Manenti & Parisi, 2007).

The open source maturity model matrix is presented in Table 4.

Table- 4: Open Source Maturity Model Matrix. (Guliani & Woods, 2005) and the Findings

Maturity Criteria	Findings
Age: OSS efforts that are just getting underway are risky for enterprises	"Sometimes we adopt OSS in version 6 sometime we adopt when the software is in Beta release but you already know in that stage that it is going to have a big impact in the future - how? On how many people are using it and are happy with the way it works and the way it is documented….Version numbers are not reliable , not a good indication of how mature it is because some release a new version every month and some release every year"
Multiple Supported Platforms: Products that work on both Windows and Unix are more desirable	"Commercial software that does not run on an OSS platform gets a lower priority".
Momentum: This is key to helping separate vital products from ones that are withering.	"Adoption is driven by all other factors – what the support like / how well is the code written / ease of use is good / how quickly is the code fixed…adoption is the outcome of these factors".
Popularity: Popular OSS products are well tested and therefore more mature. They are also likely to be interoperable with a large number of other products.	"If the same name pops up then we investigate – recurring name makes us take a deeper look. . "The most critical factor is what is its adoption around other companies. If adoption is high then you find documentation is good, ease of use is good. Lots of stuff that have fallen by the wayside. Because I am making a choice I generally do a lot of research on internet about the different ways to do the

	thing we want to do".
Design Quality: This criterion is key to determining the effort required to extend and adapt the product for enterprise use.	"All active development projects have a source browser on the web. Any person with interest can look at newly added code. Have a look at sourceforge.net".
Setup Costs: Most products should require a setup effort of hours and days, not weeks and months	Consistently "Lack of documentation" cited across the companies. "Packages are not detailed, some fantastic enterprise level documentation, but on average below what you would expect from a licensed piece of software…and because you cannot get the support you have work it out yourself". "Evaluated Pentaho….because of the lack of documentation we looked at Jasper…it worked the way we wanted it to – worked with package and ironed out kinks myself…works best and most familiar – I evaluated the tool".
Usage Costs: This criterion is often overlooked when evaluating a product	
End-user support: User community (forums, mailing lists) and 3rd party support are vital to a products success	This factor features prominently in whether companies decide to adopt or not to adopt. "Maturity of OSS? …I search for write ups / reviews / peoples experiences etc." "Look at project history – features / releases / active user community / documentation / support requests. Momentum…user forum…quality and design? Architecture documents to understand how it is put together. But does not influence adoption. Testing practices: Yes".
Modularity	
Collaboration with other products	"Normally they have these architecture documents which help me find out how it works"
Standards Compliance	"I check on upgrades for software. E.g. Jasper server – check constantly – if there is a full release I download – check change logs to see what's added, User experience with software…bug fixes etc."
Developer support	"Critical". "Major factor". Please see point under OSS community becoming the vendor (Environment: Vendor Support).

Organisational Factors

Firm context

Generally companies can be classified as high IT intensity because they deploy systems of strategic importance and spend significant time evaluating new technologies. Consistently the companies were extremely knowledgeable about IT.

Centrality of IT

Across the board the centrality of IT to the business strategy is core to the willingness of the organisation to adopt open source in most of the cases. IT is very supportive of business strategy. Most of the company processes is automated. Without IT the company would not exists.

For SME's IT and the business have a symbiotic relationship. IT is central in putting the infrastructure in place to make the business viable, to the extent of looking at the cost of putting a solution in place which meets the business need.

Open Source Attitudes

The general attitudes have gone from purely antagonistic pooh pooh it as the fringe elements using it – not used by a serious IT environment" to favourable, from skepticism to converts.

This ties on very closely with the change management process of managing the "deskilling from proprietary", where the older people with proprietary skill are a lot more resistant and skeptical.

Attitude of employees are influenced largely by the attitude of the leaders…and vice versa. Great enthusiasm and belief does play a role in how the "non converted" perceive OSS. Passion is an element we found consistently after people become OSS converts "Now my guys are all converts…one can't preach to the converted…they make it work because they want it to work".

Standards Attitudes

Standards attitudes relates to the differing opinions about the value of tight integration provided by proprietary vendors versus the choices provided by open standards. Generally, companies are more than happy with the proliferation of choice despite the problems.

Boundary Spanners

The definition of a boundary spanner is the presence of IT staff with open source experience which affects the decision to adopt. These resources often motivate for the adoption of open source (Kwan & West, 2004). Consistently the OSS skill has been acquired through a process of evolution, and thus in this definition we broaden the term to include resources that have been exposed to OSS (they have programming skill and experience in programming languages but may not necessarily have the OSS skill proficiency). "Company adoption of OSS is largely dependent on in-house IT development and expertise (Nepelski & Swaminathan, 2007), where the expertise in our definition is extended to development expertise regardless of whether it is OSS expertise or proprietary expertise.

The factor which is not quite boundary spanners within an organization, but influence, persuasion from a third party who has been exposed to OSS. Two of the companies were directly influenced by a third party with OSS exposure who changed the skepticism when demonstrating the reliability of OSS.

Consistent with literature "As institutions begin to explore open source projects and the communities which support them, they are likely to experience push-back from those new, unfamiliar, concerned, reluctant or even opposed to—not the products' functionality, features or usability—but open source software itself" (Masson, 2007). In relation to the reasons for the skepticism "people have this trend and we are brought up this way - one get what one pay for. If something is free it's worth nothing, its rubbish and that is not the case with open source and people tend to think that because it's free, its inferior".

Slack

The degree of available slack resources and cost is important to OSS adoption. Slack is important, but not so much because the company has the additional resources and finance to consider OSS solutions, but in terms of growing the skill base. The argument against slack resources is that the same applies to proprietary software, in that you still have to make resources available to test any new software.

Environmental Factors

Vendor Support

The OSS community has become the vendor as companies rely more on the support of the OSS community. "In the absence of vendor I would not be swayed not adopt. Code is written by the OSS community, but they do maintain their products which is similar to vendor support. In this case then yes, it is very important. We want to know that the vendor (OSS community) is interested. If somebody has lost interest in this then it raises flags". "Vendor support is comforting if they give support on their own products.… Jaspersoft supports Jasper…"

Vendor support is less related to the services they offer and more with the buffer they offer in terms of fear of legal reprisals. "Vendor support is important because we use OSS a lot. From a legal point of view, if somebody contests some piece of code in an OSS package who takes the wrap for that? Currently our vendor Sun takes the wrap". "We are heavily tied to vendors because of our customers need to sue someone if it breaks". "We signed an agreement with Sun. OSS is the basis for a lot of their software, we explicitly stated in the license agreement that they cover legal ramifications arising from OSS".

OSS Support

Consistent with literature, Support is most commonly accessed via the OSS community and is always available. No case was reported with regards support which resulted in multiple conflicting answers with no one authoritative source. "Not really. There were a couple of answers with deviations… slight deviations …but the influence of moderators and administrators of the site and super users ensure that I get a standard answer…many approach the problem in same way and describe the fix in same way as well…." "I have always received decent answers…better than the vendor. Conflicting answers – no".

Support also comes in the form of bounties. "To put a bounty on something means if you have a problem like I want this piece of software developed, or if we don't have the time to figure out the solution ourselves, we simply put a system spec on internet. Accessed by enthusiasts all over the world and boom, you have a solution in your inbox…"

Competition Intensity

Competition intensity does not play a role in OSS adoption.

Firm Size

Contrary to the statement that scale effects creates barriers to OSS adoption for SMEs (Nepelski & Swaminathan, 2007) it has been consistently found that the benefits associated with OSS is realised by SME's as well. "We have gotten so much financially and in terms satisfaction using OSS…"

Contrary to literature suggesting that SME's are resource poor and therefore cannot keep up with the multitude of compatibility problems, variety of choice, reliability and future development uncertainties, SME's fair well.

Maturity and firm size seems to be closely linked with OSS adoption. The opinions of these small / medium companies is that for them to adopt OSS (and change from a proprietary shop in some instances) is more viable than for big companies "e.g. Old Mutual where is becomes a virtually unrealistic task". Generally speaking, the answers are always in comparison to a big company. SME's do not struggle to keep up with the variety and rapid development of OSS version upgrades.

Technology Skills

Generally all companies experience problems when trying to fill existing vacancies "Getting resources into a company is difficult. We are actively recruiting people but it's a problem". But there seems to be a general trend in terms of companies "living with the staff shortage". Because companies are aware of the skill shortage they tend to look after their current employees quite well and thus poaching becomes difficult because the companies try to poach from also aware of the skill shortage and thus also look well after their employees.

In terms of openness to OSS, new younger people are more eager to learn and explore and consider it quite a challenge whereas the older guys are problematic i.e. "You will rarely employ an older person skilled in proprietary software to join and OSS shop because they are less willing to make that move".

The availability of external skills (labour market or from contracting resources from IT services companies) is not a huge dependency. The reliance on skill comes from the OSS community "OSS has lots of guys from all over the world…lots more skill than any company could employ…" "We contract outsourcing companies to make up the difference".

No one company determined if they possessed the prerequisite skills for OSS implementation and deployment. The skill Gap is closed by a process, as the adoption of new OSS technologies grew the skill set matured and evolved in parallel and closed the gap. The gap between required skills and what the company has is mitigated by networking.

This is aided by the exposure of more technically inclined users to the mechanisms of the code and has resulted in a transfer of ICT skills. Consistently companies report that skills are enhanced by exposure to source code. "The developers in the OSS community follow very good development coding standards, there is uniformity in the way they name variables, the way they design things etc.…this is good for my learning".

Legitimacy

Legitimacy is bestowed on OSS by big companies, but this seems to be post adoption..."once you to start using it, you discover, 'But people like Nashua are using it and AT&T (Bell), you know, Nokia are using it, Toyota are using it.

The legitimacy is also in the realm of credibility of certain components…these big companies must have put the software through its paces, must have tested it rigorously "Otherwise one have to put it through its paces by oneself" and because of this it is ok to use it.

Political Splintering

Companies are generally aware of political splintering, but it does not affect adoption one way or the other. The core issue is the best solution which will meet their needs. “No…it raises no flags. Political splintering among the leaders of OSS causes fragmentation in its membership e.g. there currently exists serious fracture amid parties who have faith in "free" software and parties who accept "open" software as true (Glass, 2004). “Free software, propagated by Richard Stallman and the FSF, requires reciprocity i.e. those who incorporate its source code into a derivative product must license the whole product as Free Software. Open Source, propagated by Linus Torvalds and Eric Raymond, does not require reciprocity (Gopalakrishnan, 2006). This fragmentation is further evident in the distribution divides on lines of commercial as opposed to free, server versus desktop and regional requirements (Wolfe, 2008). This splintering could affect the adoption of software in so far as it could influence employees in companies that have a firm conviction in open source.

Availability

SME’s (small-to-medium-sized enterprises) globally can gain benefit from the pioneering OSS model to generate new business opportunities (Fitzgerald & Kenny, 2003) but adoption is hampered by availability. Vendors and resellers of open source do not have a significant or established presence in Africa, an while the software is available for download from the internet unreliable connections and the high cost of the internet in developing countries encumbers usage and incurs higher costs to download software from the Internet (Bruggink, 2003).

Additional Factors

Levels of OSS Engagement

Theunissen, Boake & Kourie (2004) suggest four levels of OSS engagement by adopting companies as depicted in table 5.

Table 5: Levels OSS of Engagement (based on Theunissen, Boake, & Kourie, 2004)

Level	Description	Definition
1	Simply using a product	The OSS product is downloaded in either source or binary form and companies use it to fulfill a need. The expenditure of resources is little. Registration as a user of the product is done to indicate support for the product. The primary benefit is the low investment required to acquire software solutions that address a need.
2	Modifying a product without sharing the modifications	Company takes an OSS product and customises it to suite their specific needs. The contributions (changes) are kept internally and not contributed back to the OSS community. The reasons for not contributing back may be that the modifications include royalty and/or patent regulated elements. At this level, the degree of resource investment increases in proportion to the extent of internally made modifications to the original OSS product.
3	Modifying a product and contributing the changes back into the community	The company acquires the product and makes changes or fixes problems to suit a particular need. These changes are integrated back into the original project and the changes are made available to the OSS community. The resource expenditure varies in proportion to the extent of the contribution.
4	Initiating and/or managing an OSS project	Significant resources invested into an OSS project. Participation becomes necessary when no one else is willing or able to address a need and/or when the company is the leader in the project's solution domain. The most noteworthy benefit is the ability to steer (at least to some extent) the direction of the project. E.g. Netscape's initiative in undertaking the Mozilla project and Pingtel's initiative with SIPfoundry.

Consistently it would seem that the levels of engagement for SME's run across levels One and Two. One respondent contributes back to the OSS community by "writing how to processes and installation procedures that is not well documented and by helping out on mailing lists by answering technical questions". "Developers contribute in their personal capacity and we do answer questions posted if we experienced the problem".

Generally though, companies do not contribute code back to the OSS community. This is an emotive issue as underlined by the following quote "I am ashamed to say we have not contributed source code back to the OSS community". In response to the question why you ashamed are the response was "We benefit from OSS community and that is why we are ashamed".

"Feeling bad" is a theme that runs across companies because the respondents also try hard to justify why they do not contribute back "we are a startup company and work is frantic so there is no time but we will probably do some more in future". "It is not because we don't want to, just could not be bothered due to work constraints".

We have gotten so much financially and in terms satisfaction using OSS, admittedly our contribution is small, and it is peanuts compared to what we would have spent if we were using the equivalent proprietary software". Contributions are made via other avenues such as donations. "These developers don't get paid for the stuff so when I access a website I respond to the 'please make a contribution if you can'". We sponsor camps and events, or if someone in the OSS community approaches us to attend a conference we would sponsor them".

Marketing

OSS communities are deficient in the resources required to aptly market their products via conventional media sources and consequently the awareness of OSS products diffuse via word of mouth. This means that awareness is high among developers and technical users, but obscure from mainstream acceptance. (Krishnamurthy, 2003).

"If one goes to Python - for instance, Python is the programming language used by Google and on their homepage there is a quote by Peter Norbec who is the Google Development Manager". "They also proclaim their successes. The visual effects in Star Wars, was done by this "Light and Magic" Co., who use Python. So it's those kinds of secondary marketing."

"The Open Source Movement is more than just one person. That's the beauty about it and word of mouth and that is how the Open Source is, it's like a virus - it spreads

Implications and Conclusion

Organisational Readiness

Organisational readiness (as depicted in figure 1) we find to be the biggest paradox. we assumed when we embarked on this that OSS adoption was based on a big bang implementation. A designed plan. Literature suggests that the adoption of OSS is this grand design, companies ascertain all the impacts of using OSS prior to adotpion and then conciously make the decision to adopt OSS. "One of the biggest problems with open source is the lack of any roadmap documenting how adoption can be successfully achieved…"

To the contrary, we found that across the board adoption was a gradual process. "More by accident than design. We stumbled across it…in response to a problem… Look at how productive we are using this OSS". "We adopted Open Source in baby steps".

Across the board, no one company looked at OSS and prepared themselves in advance in terms of skill set etc. As adoption was a maturation and evolving process, the skill set evolved and matured in parallel. "There is no point, there was never a point, you know, you asked me "at which point did we adopt Open Source"? There was never a point when we said, 'Right we are now adopting Open Source'. It was a gradual process where you gained trust in the product … it's like falling in love, you know, before you know it, you're in trouble".

When looking at new OSS solutions, the skill set is not prepared up front. They search for solutions in response to a need, when they find the solution, they download, install, play and test. Once they are satisfied that the product will meet their need, do they employ it in a non mission critical application and progress from there. During this "play and test" phase the skill set matures and evolves.

Adoption of OSS

Inconsistent with literature the reasons companies opted to invest in OSS was not due to pressures to cut costs or pressures to produce more without increasing costs. “It's not about the money, it's about the ethos of people in the Open Source community. You know, they are big proponents of Open Source and they actually willingly help other people”. “If we need the enterprise one, we don't stick with the standard one, we'll pay and get the enterprise one. We'll pay the license, so we are not going to take software for free when we have to pay for it we'll pay for it”.

Fit to task is the absolute criteria which sways adoption “I will use whatever is the most pragmatic to use. If I had two choices and one was open source and one was proprietary, and they were equal, I wouldn’t necessarily go for the open source one. I feel open source is better because ‘adoption drives the community’”. “I still believe the best tool for spreadsheets, is Excel, so we use Excel. I'm not saying there's anything wrong with Open Office or Sun, or whatever the others are. It's just, I don't have a problem with the Excel”.

Consistently the decision to adopt OSS is based on trust in the software, gained by a process of trying it out and OSS proving itself to be reliable, stable and with the necessary functionality that would meet the business need. Once the confidence has been gained that this piece of software is stable and reliable, only then is it used. Generally companies adopt OSS to improve the quality of services or products in order to stay in business. Fig. 3 presents the summary of the OSS Adoption.

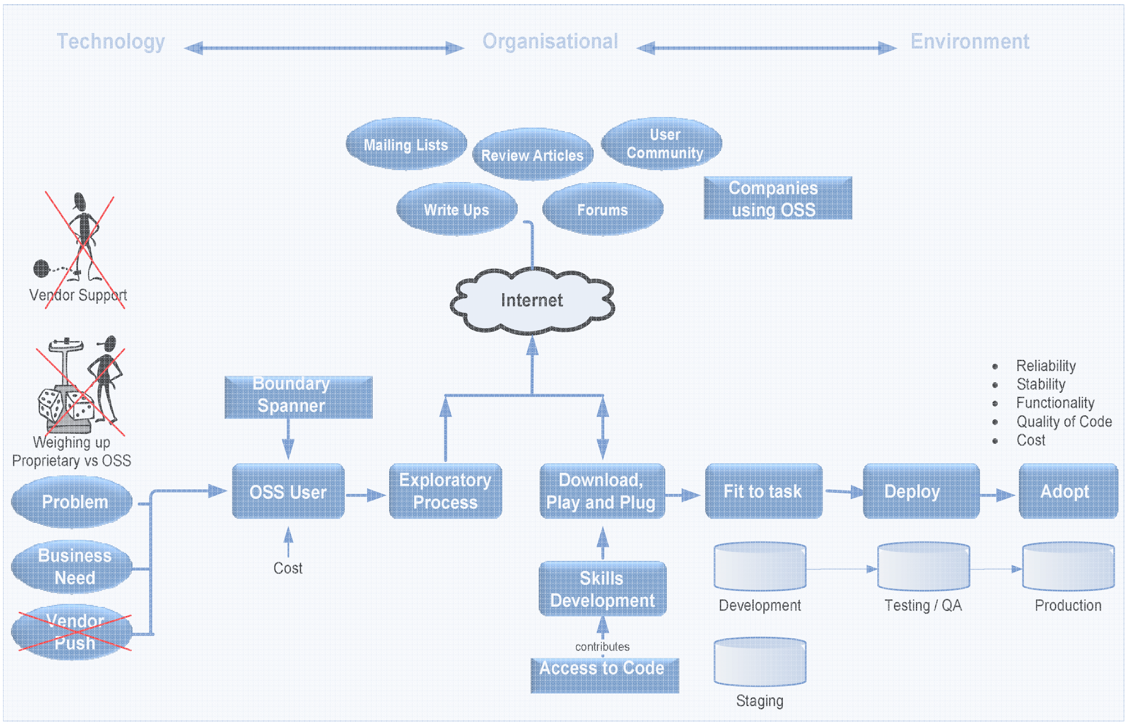

Fig.3: Summary of the OSS Adoption

The process of adoption has been cited as follows and represented in Figure 3. : “we follow a very informal process, so what would happen is, we'll download, a few guys will install it and they'll start playing with it and ask questions around what if this? What if that?... and if they start load testing it and all those kinds of things and then, at that point in time, after a number of people have looked at it, played with it, discussed the pros and cons… it's got this and that, the following shortcomings and things like that, *then* we will start using it, but then we start using it in non-mission critical applications and once again, we come back to, once it's earned our trust, then we start using it in mission critical applications". Boundary spanners expose companies and people to OSS. The actual reliability, stability, functionality and quality of the code sway people to adopt or not to adopt.

In summary we have found that the TOE framework takes cognisance of the internal and external context the company operates in it comprehensively covers the adoption process. It serves as a useful framework when looking at the full scope of OSS adoption.

Consistent with literature the growth open source software is based on a stack approach. "low in the software stack," focuses on operating systems, server software, development tools, databases, etc. and as companies become more familiar and comfortable with and dependent on OSS in these utilities, they will branch out to open for systems e.g. email, those services deemed mission critical by campus decision makers as "enterprise applications" (Masson, 2007).

Reasons for staying with OSS

The most salient reasons companies stay with open sources is based on the fact that it is "Clean, Beautiful, Reliable, stable and trustworthy". "Companies tend to OSS, the higher their requirements for open standards and interoperability" (Nepelski & Swaminathan, 2007).

Ultimately, it's about choice. People should be able to identify their needs and compare all the available options. There should not be any legislative or technical barriers that make any options unavailable (Vital Wave Consulting, 2006).

The Open Source Maturity Model which was used to ascertain the maturity of the open source at the time of adoption is useful overall even though not all the items are relevant. The Software Cost and Risk Model which looked at the Total Cost of Ownership of adopting Open Source proved to be insightful. Despite companies professing not to measure the costs upfront, many of the factors are taken into consideration. In this particular exercise the cost of Open Source would seem no more expensive that proprietary, indeed it would seem that the cost of Open Source is cheaper.

The Open Source Adoption Models suggests factors within the Organisational context the company finds itself in at the time of adoption. With the exception of competition intensity, all the other factors are valuable when looking at OSS holistically. The Open Source Skills and Risk Tolerance model which looked at the skills and skill level required both internally and externally which influence adoption is a not useful and has no impact on adoption. Skill set grows and matures as OSS adoption evolves in a company. Skill set is not gauged upfront. The levels have no bearing on OSS adoption.

Although OSS will not destroy industry giants such as IBM and Microsoft, it will place increased pressure on traditional vendors to more-aggressively innovate, improve quality and drive higher value in their own products as they endeavor to counter this growing competitive threat (Weiss & Driver, 2005).

On the economic level, OSS helps retain a large amount of money within the South African economy, which is otherwise paid out as licensing fees to foreign coffers. South Africans spend R6 billion on software licensing every year. OSS provides small, medium and micro enterprises with immediate access to high quality, customizable and powerful software. "Since good software becomes legally accessible at little or no cost, the threat of piracy is eliminated. Probably the most important benefit is that it stimulates the local IT sector" (Gopalakrishnan, 2006).

References

Bruggink, M. Open source in Africa: towards informed decision-making. *IICD : The International Institute for Communication and Development* , 7(2003): pp 1-4.

Comino, S., Manenti, F., & Parisi, M. From planning to mature: on the success of open source projects. *The international Journal of Management Science* , Research Policy 36 (2007), pp 1575–86.

Dedrick, J. and West, J. "An Exploratory Study into Open Source Platform Adoption," Proceedings of the 37th Annual Hawai'i International Conference on System Sciences, Waikoloa, Hawaii (Jan. 2004)

Dedrick, J., & West, J. Adoption of open source platforms: an exploratory study. *HBS - MIT Sloan Free/Open Source Software Conference:New Models of Software Development* (2003) pp. 1-27.

Driver, M. Key issues for open-source software in 2007. Stamford, CT: Gartner .

Fitzgerald, B., & Kenny, T. Open source software in the trenches: lessons from a large-scale OSS implementation. *Twenty-Fourth International Conference on Information Systems,* (2003) pp316-326.

Glass, R. A look at the economics of open source. *Communications of the ACM* , 47, 2 (2004), pp25-27.

Gopalakrishnan, J. Lessons in open source wisdom from South Africa. *Linux For You* (2006), pp 55-59.

Guliani, G., & Woods, D. (2005). *Open source for the enterprise.* United States of America,Sebastopol: O'Reilly Media, Inc.

Holck, J., Larsen, M. H., & Pedersen, M. K. (2004). *Identifying business barriers and enablers for the adoption of open source software.* Frederiksberg: Copenhagen Business School, Department of Informatics.

Krishnamurthy, S. A managerial overview of open source software. *Business Horizons,* (2003), pp 47-56.

Kwan, S. K., & West, J. (2004). A conceptual model for enterprise adoption of open source software. In *Enterprise Adoption: The Standards Edge: Open Season.*

Lipert S.K. & Govindarajulu. C. (2006) Technological, Organizational and Environmental Antecedents to Web Services Adoption. Communications of the IIMA, Vol 6 No 1, pp146-157.

Masson, P. (2007). *Barriers to the adoption of open source: personal and professional observations,* available at: http://blog.worldcampus.psu.edu/index.php/2007/04/17/ barriers-to-the-adoption-of-open-source-personal-and-professional-observations/ (accessed on:2 October 2009).

Nepelski, D., & Swaminathan, S. OSS adoption: who is leading and why? *DIW Berlin Weekly Report*, 1. 3,(2007), pp1-6.

Russo, B., Zuliani, P., & Succi, G. (2003). Toward an empirical assessment of the benefits of open source software. *Taking Stock of the Bazaar: The 3rd Workshop on Open Source Software Engineering, (2003),* pp. 117-120.

Theunissen, M., Boake, A., & Kourie, D. (2004). *A preliminary investigation of the impact of open source software on telecommunication software development.* Department of Computer Science School of Information Technology University of Pretoria,Pretoria, South Africa: Centre of Excelence in Teletraffic Engineering for the Information Society (CeTEIS).

Vital Wave Consulting. (2006). *The South African adoption of open source.* ?: Vital Wave Consulting, Inc.

Weiss, G., & Driver, M. (2005). *The effects of open-source software on the IT software industry.* Minnesota; pp 1-7: Gartner RAS Core Research Note G0013638; Open Source Technology Alliance.

Wolfe, A. (2007) Reasons Why Linux Won't Succeed On The Desktop. *InformationWeek* , (2008) Online at: http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=201807072 (accessed on:2 October 2009).

Zhu, K., Kraemer, K. & Xu, S. (2003) Electronic business adoption by European firms: A cross-country assessment of the facilitators and inhibitors, *European Journal of Information Systems* 12 (4), pp. 251–268.

Zhu, K., Kraemer, K.L., Gurbaxani, V., & Xu, S.X. (2006), Migration to open-standard interorganizational systems: Network effects, switching costs, and path dependency, *MIS Quarterly* 30 (Special Issue) (2006), pp. 515–539.