Working Paper

The Challenges of Open Source Software in IT Adoption: Enterprise Architecture versus Total Cost of Ownership

By

Michael Holm Larsen
Jesper Holck
Mogens Kühn Pedersen

No. 11-2004

Institut for Informatik

Handelshøjskolen
i København

Howitzvej 60
2000 Frederiksberg

Tlf.: 3815 2400
Fax: 3815 2401
http://www.inf.cbs.dk

Department of Informatics

Copenhagen
Business School

Howitzvej 60
DK-2000 Frederiksberg
Denmark

Tel.: +45 3815 2400
Fax: +45 3815 2401
http://www.inf.cbs.dk

The Challenges of Open Source Software in IT Adoption: Enterprise Architecture versus Total Cost of Ownership

Michael Holm Larsen

Jesper Holck

Mogens Kühn Pedersen

Copenhagen Business School, Department of Informatics

Howitzvej 60, 2000 Frederiksberg

E-mail: {mhl,jeh,mk}.inf@cbs.dk

***Abstract**: The paper presents an explorative study of Open Source Software (OSS) focusing on the managerial decisions for acquisition of OSS. Based on three case studies we argue that whereas small organizations often may chose adoption of OSS expecting significant cost savings, a major barrier for larger organizations' adoption of OSS lies in the organizations' consolidation of the enterprise architectures, in addition to that OSS will not be adopted before satisfactory delivery and procurement models for OSS are established.*

***Keywords**: Open Source Software, OSS, Enterprise Architecture, Total Cost of Ownership, Delivery and Procurement Model.*

1. Introduction

In developing our understanding of the Open Source "Movement" (Lakhani & von Hippel, 2003; Lerner & Tirole, 2001; O'Reilly, 1999), a multi perspective analysis needs to be undertaken in order to embrace the complexities of this phenomenon. As this research field is fairly young, mature bodies of theories such as economics (Lerner & Tirole, 2002) often set the development directions and research agendas of the research field. In our literature review we find research contributions as well as contributions from practice, e.g. Pedersen et al. (2002). Although both groups are represented, we find none addressing business challenges from a business perspective, i.e. taking a managerial point of view to OSS –that is not "only" taking a product, developer, community or industry perspective as often is seen (e.g. Nakakoji et al. 2002; Wayner, 2000; Wilson, 1999).

Our particular focus of this paper is the decision-making of managers in organizations when confronted with the challenge of open source software. The reason why the initial decision-making is of interest and highly important is that although the realized costs at this point in time is relative low, the "dispositional costs" (Olesen 1996), which are the costs derived from the decisions and dispositional mechanisms, i.e. the costs of the consequences of the explorational decisions, however, are often higher that half of the total accounted costs of a project.

The motivation of this research is based on the findings from dialogue and cooperation with businesses, public and non-profit organizations. Some of the findings seem to point to major advantages of OSS, but even though OSS is becoming more widespread, larger organizations seem hesitant to base their information systems on OSS, except for a few niche areas in their organization where OSS may proliferate as long as it remains invisible to the corporate board.

Therefore, we have developed the hypotheses that a major barrier for larger organizations' adoption of OSS lies in the organizations' strategic IT policies and enterprise architectures, and that OSS will not become adopted in these before satisfactory delivery and procurement models for OSS are established.

The research method of this paper is to position the research contribution in the OSS literature through a literature review, and then present 3 case studies of different origin and nature, and compare the findings.

In this paper we take as givens the concepts of architecture and total cost of ownership. Architecture is here considered as the overall system design where computers, networks, logical components, and the logical interrelationships of the computers, the operating system, and the related network, and select standards are specified. Thus "an architecture is a description of system

structures, of which there are several (module decomposition, process, deployment, layered, etc.). Architecture is the first artifact that can be analyzed to determine how well its quality attributes are being achieved, and it also serves as the project blueprint. An architecture serves as the vehicle for communication, is the manifestation of the earliest design decisions, and is a re-usable abstraction that can be transferred to new systems.", cf. Bass et al. (2003).

Total Cost of Ownership (TCO) is a type of calculation designed to assess direct and indirect costs and the benefits related to the purchase, ownership and use of any IT component, which includes all costs associated with deploying, operating and maintaining the system. Hence, TCO includes:

- Original cost of the computer and software
- Hardware and software upgrades
- Maintenance
- Technical support
- Training and coaching
- Downtime

The paper is organized as follows. In this section we provide a definition of OSS. Section two outlines previous research in this area of OSS related to managerial decision making. In section three we will specify potential use of open source software. Section four present the case studies in order to provide a basis for the analysis in section five, which is summarized in a table.

1.1 What is Open Source Software ?

Like many other terms, the exact meaning of "open-source software" is debatable. The definition offered by the Open Source Initiative focuses almost only on the software license, a memorandum of contract specifying the perimeters of the permission granted by the copyright owner to the user of the software [wikipedia, software license]. In contrast to proprietary software licenses, which mostly deal with restricting users' rights and vendors' liabilities, open-source software licenses, according to the Open Source Initiative, must provide users a number of rights, including:

- Anyone is free to distribute the software
- Derived works are distributed under the same license (in GNU whereas free to change in lesser GNU and in others like FreeBSD)
- The software can be used by everyone and in all fields of endeavor

- The software source code is freely available

The Open Source Initiative has approved several different licenses as conforming to these requirements, but other licenses exist for what is often referred to as open source software. Some prefer other terms like "Free Software" (proposed by the Free Software Foundation) or "Libre Software" (a term used to emphasize that the software is "unrestricted" but not necessarily gratis), but in most contexts these terms are close to synonymous.

Obviously, there may be a huge variation in software released under an open-source license. Sourceforge.net is common repository for well over 50,000 open-source projects, but many of these are very small (often one-man projects), and many are dormant or dead.

Also, the question of license is not the only characteristic of the term "open-source software" as used in daily conversation. Two other characteristics (Henkel, 2003) are:

- Software developed and maintained through the "open-source model," in which many developers contribute code to a common repository
- Software with roots in the "open-source community." It has, e.g., been debated whether proprietary software, at some point released under an open-source license (like Mozilla), is really open-source software.

2. Literature Review

Although the research area of Open source software (OSS) is relatively young, documented research contributions have started to become available in academic journals and at peer-reviewed international conferences. OSS development is investigated from different perspectives. In the following, a number of research contributions are presented relating to decision making in OSS development projects and in particular in relation to adoption of OSS initiatives.

Several contributions address the development of Linux, e.g. (Torvalds, 1999; Wayner, 2000; West & Dedrick, 2001), and identifies the development of the software project as a project with a bottom-up structure, cf. Fielding (1999) and Koch & Schneider (2002), which is based on autonomy though internal control structures are to be seen, cf. Jørgensen (2001), Holck & Jørgensen (2004), and Scacchi (2003).

An alternative perspective to decision making in OSS is provided by, Payne (2002) who explores security of open source software. Furthermore, analysis of open source code quality is presented by Stamelos et al. (2002).

Nakakoji et al. (2002) investigate evolution patterns of OSS communities, Ye et al. (2002) analyse sustainability of OSS communities, and Bergquist & Ljungberg (2001) address social relationships in open source communities and how they are organized, and the implications for decision making.

A management perspective is brought into the open source software literature by e.g. Lerner & Tirole (2002), Holtgrewe & Werle (2001) and Koch (2003). Lerner & Tirole (2002) present simple economics of OSS, Holtgrewe & Werle (2001) address the link to strategy of OSS development, and Koch (2003) advocates that OSS no more is an option. However, the question of organizations is how they should get involved in OSS activities. Therefore, he suggests that organizations should at least start experimenting with low-risk web application before transforming the entire infrastructure of the organization.

Going through the literature, we find that addressing the business challenges of Open Source from a business perspective, in particular needs further investigation.

It is apparent that the most research interest has been in answering the question of what makes OSS development possible:

- Why do individuals and organizations choose to deliver work for free?
- How do they control and coordinate their work?
- Can these organizations produce quality software?

We have only seen a very limited interest in investigating the "customer side" of OSS, i.e. why do commercial organizations choose (not) to adopt OSS. Commercial software vendors like Microsoft have argued that OSS should not be adopted, because:

- Even though acquisition is cheap or gratis, TCO (Total Cost of Ownership) may be high
- Quality of OSS may be poor

These views have been challenged by several reports, including Pedersen et al. (2002) and the adoption of OSS by major corporations (e.g. Google, Amazon) and institutions (e.g. City of Munich Council, City of Bergen Council). In this paper we will, however, investigate complementary hypotheses based on managerial decision-making in regard to OSS acquisition.

3. The Use of Open Source Software

For the present discussion, we will offer a few more characteristics of what we will refer to as "open-source software":

- Software distributed as application programs, leaving out e.g. source code libraries
- Software maintained and developed by a mature organization, including
    - Technological infrastructure: common software repository, website, mailing lists for users and developers etc.
    - Organizational infrastructure: division of labor, hierarchy, procedures, plans etc.
- Software developed and maintained through the "open-source model"

This "definition" will cover all major open-source projects, including Linux, Apache, MySQL, Mozilla, and Samba.

Even though we for an OSS product may be able to identify a "vendor organization" that develops, maintains, and distributes the product, and a number of "customer organizations" that use the product, this is clearly not a complete picture of the many possible ways in which organizations can benefit from open-source products. Based on our survey, some of the more important ways seem to be:

- Direct user, using OSS products for the organization's own administration or production processes. Example: using OpenOffice for administrative work.
- Software reseller or distributor, selling "packaged" OSS, possibly bundled with proprietary software. Example: Redhat's Linux distribution,
- Hardware vendor, bundling proprietary hardware products with OSS. Examples: IBM's computers with preinstalled Linux.
- Publisher, selling books that document and describe OSS. Example: O'Reilly (O'Reilly, 1999).
- Consultant or systems developer, providing customer solutions based on OSS. An Example is the Danish company Casalogic.

It is important to note that there are many ways in which organizations can choose to support OSS:

- Source code – improvements, corrections

- Documentation
- Error reports, assist in bug finding and removal
- Suggestions for improvements
- Supply and maintain technical infrastructure (or donate money to do this)
- Participate in management of the OSS organization

From the list of initiatives, we learn that Open Source Software may be integrated in an organization in many ways, each containing different degrees of direct investment, involvement, risk aversion, etc.

Before extending our analysis of OSS particulars in a managerial context we document the (non-)adoption decision making in three very different cases.

4. Case Studies of Open Source Software

In this section three case studies are presented; two public sector organization and one commercial organization.

4.1 Organisation A – CBA

Copenhagen Business Academy (CBA, www.kts.dk) was founded in 1843. In 2003 the Academy merged with the AMU-Centers in Copenhagen, and now provides teaching from 13 outlets in Copenhagen. In 2002 the Academy had a turnover of DKK 445M (approx. USD 73M), and the result of the fiscal year was DKK 17M. In 2003 the turnover exceeded DKK 500M with 700 employees and 17,000 students enrolled.

The Academy offers two advanced studies: Constructing Architect (3,5 year Bachelor program), and Multimedia Designer (two year Diploma course). In addition to this, the Academy offers more than 50 different types of education (bricklayer, dental technician, etc.), labour market courses, company-adapted supplementary education, and the college-level higher technical exam (HTX).

Large parts of the information systems at CBA were developed by the Danish Ministry of Education and has been mandatory for the institution: most importantly the student and teacher administration system (EASY-A), and the financial system (EASY-Ø). In addition to the ministry-developed systems, CBA has two Microsoft Exchange servers, running as post offices for students and employees, respectively, a SAS Institute executive information system, and various educational systems.

Six persons are employed in the academy's IT-group, administrating and supporting the information systems; additionally, some of CBA's teachers have support of the educational systems as part of their job. The annual cost of the IT-group is DKK 7M (approx. USD 1.1M), of which typically around DKK 0.1M is allocated to external consulting services, though in some years this amount may be 10 times larger. The prime goal of the IT-group is to provide efficient and effective administration and support of CBA's IT infrastructure. Hence, a lever for obtaining this goal is to reduce the complexity and variety of employed systems and software.

Except for educational purposes (courses in OpenOffice etc.), CBA does not at present employ OSS. According to IT director Torben Johannesson, the primary advantage of OSS should be the low acquisition cost, but because of substantial educational discounts from vendors (primarily Microsoft), this advantage is very limited and the perceived lack of support of OSS is a decisive barrier. He observed that:

> *"If you buy from Red Hat [commercial Linux distributor], there is no support, nobody has checked for security holes, no one updates the drivers, nobody does anything – you are left high and dry. This is a precarious situation."*

Another barrier comes from the importance of compatibility with the Oracle database management system (DBMS). Early in the requirements specification phase for the EASY systems it was decided to base these on the Oracle DBMS, and as the costs of now changing to another DBMS would be very high, it is of decisive importance for CBA to choose hardware and software compatible with the Oracle DBMS. Oracle should be compatible with Linux, but – according to CBA – support of the Linux platform seems to have a low priority for Oracle, causing upgrades for Linux to be almost one version number behind upgrades for prioritised platforms like Sun Solaris and Microsoft Windows. Therefore, according to the IT director, a switch to Linux would seriously increase the need for local system "patches" while waiting for new releases of the Oracle DBMS. Furthermore, CBA is worried about compatibility problems between different Oracle versions, Linux flavours and versions, and hardware.

Summing up, CBA is highly vendor dependent, and has excluded OSS due to perceived lack of support, fear of compatibility problems, and slow timing of upgrades.

4.2 Organisation B – POG

"POG" is an anonymous name for a Scandinavian organization in the Petrol, Oil and Gas (POG) industry. POG is an oil company operating an oil production of some 500,000 barrels per day and a sales gas production of up to some 1,000

million cubic feet per day. The company has a net production exceeding 300,000 barrels of oil equivalents per day from fields in Europe, Africa and the Middle East. The net revenue in 2003 was DKK 19B or more than USD 3B.

POG has 1,200 employees worldwide. The company is divided into divisons and has separate IT departments in each country. The IT department in Copenhagen has 14 employees and services approximately 300 users. Traditional "office work" like administration of personnel, economy, and contracts constitute only a minor part of the application area. IT is mainly used in relation to oil extraction and reservoir simulation, where users typically are geologists and engineers. In order to make a good working relationship with users, the department's policy is to hire personnel with university degrees, and to offer everyone a minimum of one week's education in geology and oil engineering.

For POG, OSS is primarily used for back-office applications. Samba has been used for more than a year for file services on four Solaris machines, and also DHCP and DNS services are running on Linux machines. The company is now shifting from using Microsoft's Active Directory for directory services to use of OpenLDAP. This, in combination with an upgrade to Samba version 3, will provide a common log-in for both Windows and Unix users, which has been a long-time goal for the department.

In some situations, OSS is also used for more user-oriented applications. One example is using Linux as platform for a commercial application, because performance is substantially better than with Solaris.

From the start, it has been a requirement for OSS acquisition at POG that the support should be as good as support for commercial applications from vendors like IBM, Sun, and HP. At first, a British company was paid to this for Samba, but this company could only deliver support for the Solaris platform. Because of POG's dependence on other platforms, most notably AIX and Linux, this solution proved unsatisfactory.

Following this situation, a support contract for Samba was made with the Danish company SuperUsers, including:

- Thorough tests of new Samba versions on hardware delivered by POG
- Consultancy in relation to installation and configuration, trouble-shooting, and error recovery and reporting.

An important role for SuperUsers has been to be an "interface" between POG and the Samba developer community, but on some occasions POG has been in direct contact with the community, reporting problems and errors, and receiving updated software versions. Often the turnaround time for these corrections has been much faster (2 weeks) than what POG has experienced with commercial vendors.

POG has, however, not been completely satisfied with SuperUsers' services. SuperUsers regard their Samba support as a custom, and hence expensive, service for POG, but POG would prefer SuperUsers to deliver a more generally applicable product, delivering Samba support for several companies, and in this way be able to offer a less expensive service. Also, SuperUsers only offers support services for a limited range of OSS products, which is why POG right now is exploring the possibilities of receiving OSS support from larger companies like HP and IBM.

POG prefers to avoid making changes in the source code for the OSS products. If needed, adjustments are implemented as custom-made installation or utility scripts (what Boehm et al. (2000) name glue code), the intention being to reduce the problems of implementing software updates.

An important advantage of OSS for POG has, interestingly, been the opportunity to make custom builds of the software for their specific environment (hardware, operating system, system libraries), in this way obtaining a better "fit" between software and environment than would be possible with closed source standard applications.

For POG, another important advantage of OSS is the conformance to open standards, which makes integration of a quite heterogeneous computer environment possible. According to POG, this is in contrast to commercial products, particularly the ones from Microsoft, which often use proprietary standards and in this way restricts POG's software options. On a more general level, it is important for POG to maintain a high degree of IT vendor independence. The arguments for this are of course economical, being able to have a wide range of products to choose from, when finding the most cost effective solution, but also emotional arguments like wanting to stand against Microsoft's near monopoly play a role.

Decisions regarding software acquisition are in general not based on conformance to a company-wide IT strategy or detailed analysis of TCO or ROI. Company management has refrained from implementing a formal, general IT strategy; the informal strategy is to keep the company vibrant and prepared for changes, avoiding the restrictions being the consequence of a formal IT strategy. Also, TCO or ROI analyses are believed to be too expensive and leading to uncertain results. When choosing between different products, the only financial analyses are simple and based on acquisition and yearly support contract prices. This shall be seen in the light of a relatively small number of employees, producing high returns (approx. USD 1.8M per employee, before taxes), which is why it is considered more important to "nurse" these operations than to obtain small savings by finding the exact, most cost-effective software and hardware products.

4.3 Organisation C – CFI

Center for Informatics (CFI) is the central IT department for the Danish Ministry of the Environment, including the department, three agencies (Danish Environmental Protection Agency, National Forest Agency, and National Survey and Cadastre Denmark, NSCD), two research institutes (National Environmental Research Institute, and Geological Survey of Denmark and Greenland), and three smaller institutions (Nature Protection Board of Appeal, Environmental Board of Appeal, and Environmental Assessment Institute). CFI provides basic IT services like IT infrastructure, office automation, document management, consultancy and training; administration of advanced knowledge work systems (e.g. models, simulations) is decentralized to the respective departments, but CFI may also for these systems be responsible for e.g. hardware and daily operations. The two research institutes have their own IT departments, but are provided with services like e-mail and network from CFI.

CFI was established in 2001 and replaced a number of former, individual IT departments in the various institutions under the Ministry of the Environment. From the start, the objective was to cut down expenses with 17% and staff with 25% in two years, and these objectives have been satisfied. At present, CFI employs approx. 30 employees and services 2,400 office workers and 600 woodmen etc. The yearly budget is approx. DKK 75M or USD 12M, revenues come from the various institutions serviced; each of these pay an annual fee for CFI's services.

A major goal for CFI has been to standardize the systems used across the various institutions. For servers, the operating system is primarily Windows 2003, but NSCD also has some use of Linux and Solaris. Common governmental system for accounting (Navision) and document handling (Scanjour) require use of Microsoft SQL and Oracle database management systems, respectively. E-mail services and content management are provided by Microsoft Exchange and CMS for most institutions, but NSCD uses Lotus Notes instead; it is the ambition to soon decide on a common platform for these. On the desktop, the standard configuration is "fat" clients with Windows XP, Microsoft Office 97 and Outlook 98, except for NSCD's use of Lotus Notes for e-mail. For certain resource-demanding applications, "thin" Citrix clients are used instead.

As expressed by CFI's deputy chief director, most of the institutions are "immensely tied-in with Microsoft". There is a long tradition for choosing Microsoft products, almost as a knee-jerk reaction, but CFI has deliberately wanted to challenge this. One step in this direction has been to decline Microsoft's Software Assurance programs, where a yearly fee guarantees automatic updates of Microsoft products to new versions and platforms. CFI found these programs too costly and decided to retain the old versions of Microsoft products on the desktop. This has led to the present situation, where these products need to be replaced, and where all options are open, including

choosing OpenOffice instead of Microsoft Office, and/or Lotus Notes instead of Microsoft Outlook, Exchange, and CMS. But when users learn that OpenOffice is being considered as an alternative to Microsoft Office, they are very skeptical; for most of them, Microsoft Office is the office suite and Outlook the mail program.

Recently, the Danish Ministry of Science, Technology and Innovation has initiated "Project e-government promoting e-government initiatives across the sector. An important element in this project is development of a common enterprise architecture framework for the entire sector, and as a result, CFI has begun a difficult and long-term process of defining and developing an IT enterprise architecture for all institutions under the Ministry of the Environment. Evidently, this work is closely related to the forthcoming decisions regarding desktop applications.

CFI's interest in OSS products has above all been focused on considering OpenOffice as a possible replacement for Microsoft Office. Thorough, preparatory studies have been made for this decision, including closely following the experiences from a number of pilot desktop projects across the public sector. CFI has found the cost of comprehensive TCO studies prohibitive. Instead a number of future scenarios have been analyzed, focusing on identifying the important differences between these.

Because the presently employed Microsoft applications are outdated, the analyses have shown that most switching costs will be the same, irrespective of whether the choice falls on new versions of OpenOffice or Microsoft Office. This holds for user training, and updating of interfaces and document templates. Also, experience has shown a negligible need for external support of desktop applications. Hence, the two alternatives are considered equal in this respect, too. Therefore, the remaining important issues of CFI in relation to the decision on future desktop applications are:

- Differences in acquisition costs, licensing
- Vendor independence
- Interfaces to the new mail system, existing documents, and outside partners
- Conformance with the upcoming enterprise architecture

For CFI, the availability of the source code for OpenOffice is not considered an advantage. According to the deputy chief director: "CFI should under no circumstances turn into an OSS center and "fiddle" with the code", and "OpenOffice should be considered a given product".

5. Analysis of the Case Sample

In investigating the three cases we find that:

- **TCO Evaluation.** None of the three organizations make use of formal, economic methods of analysis (like TCO or ROI) when considering software acquisition. The common belief is that these kinds of analysis are far too expensive and troublesome to perform.
- **Enterprise Architecture.** Questions regarding architecture were import for all three organizations. OSS or commercial products were not evaluated on their inherent qualities alone, but on how well they were able to "fit" into existing architecture and facilitation of business processes.
- **Support Quality.** The question of support was important for all three organizations, even though they had very different considerations to this issue:
    - For CBA, the expected lack of support was an important argument against OSS.
    - For POG, their experiences with OSS (using external consultants and various user/developer forums) showed support to be at least as good and fast as support for commercial products.
    - For CFI, experience showed that they actually did not need support for their desktop applications.
- **User Appreciation.** User attitude is a significant factor in adoption of OSS. This was perhaps most evident at CFI, where their general impression was that users wanted up-to-date Microsoft products.
- **Subjective Attitudes.** Attitudes and weakly founded suppositions play an important role in software acquisition for all three organizations. In both POG and CFI, decision makers had a positive attitude to OSS as they wanted to challenge Microsoft's monopoly. In contrast to this, decision makers at CBA were very skeptic towards OSS, even though they had no personal experiences with OSS or objective evidence to refer to. Especially experiences from CFI showed that (user) conservatism was a strong obstacle against a potential switch away from Microsoft products.

It is notable that all three case studies had the IT department as the origin of the OSS initiatives in the organization. Although POG also was influenced by users, one could have expected that vendors might have been more convincing in getting their message through.

Turnover per employee is considered as a likely explaining factor for the extent to which organizations regard the license costs, etc. as significant to their core

value-adding operations. A "high" turnover per employee is considered to be more than 2 million Danish Kroner (DKKM), a "medium" turnover per employee is between 1-2 DKKM, and a "low" turnover per employee is below 1 DKKM. In particular the POG organization had a turnover per employee of approximately 10 DKKM, which is very high.

Beneath, the table provides an overview of important characteristics of the cases.

Organization	**Organization A CBA**	**Organization B "POG"**	**Organization C CFI**
ORGANISATIONAL CHARACTERISTICS			
Sector	Public	Private	Public
Industry	Education	Petroleum	Environment
Employees	700	1.200	2.400
Employees, IT department	7	14	30
Turnover per employee	Low	High	Low
INFORMATION TECHNOLOGY IN THE ORGANISATION			
IT environment in the organization	Homogeneous	Heterogeneous	Homogeneous
Tasks of IT department			
Maintenance	Yes	Yes	Yes
Customization	Yes	Yes	Yes
Development	No	Appl. development	No
IT Strategy/Policy	None	None	None
OPEN SOURCE SOFTWARE IN THE ORGANISATION			
Origin of OSS initiatives	IT department	Users / IT department	IT department

Initial requirements for adoption of OSS	Product and support quality	At least same level of support as for commercial applications	Cost Savings
Arguments of OSS adoption			
TCO method	No	No	No
Other formal methods	No	No	Yes
Architecture	Yes	Yes	Yes
User attitudes analysis	No	No	Yes
Anti-sympathy towards Microsoft Monopoly	No	Yes	Yes
Subjective or non-substantiated Judgments	Yes	Yes	Yes
Primary OSS initiatives	Open Office	Linux servers	Linux servers
	N/A	Samba on Solaris platform	Open Office

***Table 1**. Case Study Comparison.*

6. Consolidation versus TCO

Whereas small organizations often may chose adoption of OSS due to significant cost savings e.g. in implementing Linux servers for web, file sharing, printer sharing, VPN's in addition to OSS based firewalls, ect., most decisions regarding procurement of SW components in larger companies and governmental organizations, however, are not taken on basis of the qualities or costs of a single component except those having a decisive competitive impact. In a large organization, top management will make their decisions strongly influenced by the enterprise architecture, which constitute the strategic framework for all investments in IT. And if OSS, as is most likely, is not a visible part of this framework, it will not be adopted in any significant scale – not even if certain OSS products appear highly "competitive" when compared with commercial alternatives. Exceptions may be that certain "niche" areas, more or less invisible

to company management, like software for researchers or for IT-department servers.

This situation may be different in smaller organizations without constraining, strategic IT policies, and as a consequence, these organizations may be more likely to "experiment" with new software vendors, including OSS. This argument rests upon that fact that the organization employs or are able to attract sufficiently skilled developers, supporters and system administrators mastering OSS. Otherwise, the technical skill base is a strong impediment for rejecting OSS activities in-house.

Procurement models, cf. Pedersen (1996), and their "fit" with vendors' delivery models are essential when organizations formulate IT policies. We find it unlikely that an organization will include OSS in its IT policy unless it is assured that a reliable procurement model has been established. This model must include technical elements (appropriation regarding functionality, security, interfaces etc.), legal elements (appropriation regarding license), and business elements (appropriation regarding vendor, customer support etc.). In the commercial market, satisfactory and well-proven procedures exist for these elements, but this has yet to be developed for OSS.

Based on this argumentation we conclude that larger organizations will only adopt OSS (in any significant scale) if one of two conditions is met:

- OSS is "bundled" with hardware products, delivered through commercial vendors. This is what we are now seeing with IBM's and HP's distribution of computers with the Linux operating system. In this way, the OSS is not really acquired by the organization, but rather delivered as an included subcomponent.
- A credible combination of delivery and procurement models for OSS is found. Now and in the coming years, this will be an important challenge for both users and developers of OSS to explore.

7. Conclusions and Future Research

The paper address the challenges, which management encounter when faced adoption of new information technology and in particular adoption of Open Source Software.

Although the research so far rest on a fairly limited sample, which can not prove statistical significance, the richness of the case studies may suggest tendencies of organizations in general.

The paper set up the hypotheses that a major barrier for larger organizations' adoption of OSS lies in the organizations' strategic IT policies and enterprise

architectures, and that OSS will not adopted in these before satisfactory delivery and procurement models for OSS are established.

The first hypothesis is that a major barrier for larger organizations' adoption of OSS lies in the organizations' strategic IT policies and enterprise architectures. The sample united confirms that the existing and/or future enterprise architecture is a decisive argument in the IT adoption debate, whereas the cases did not confirm that IT policies had any significance in the debate. As the organizations did not have fully developed IT policies, this does not mean that IT policies are not significant. It only suggests that in this case, it was not a priority.

The second hypothesis was that OSS will not adopted in the larger organizations before satisfactory delivery and procurement models for OSS are established. The case study sample presented strong vendor dependencies due to insufficient competences related to OSS.

Future research e.g. based on questionnaires will reveal the significance and diffusion of these managerial arguments for adopting open source software.

References

Bass, L, Clements, P. & Kazman, R. (2003). Software Architecture in Practice. 2nd edition. Addison Wesley.

Bergquist, M., & Ljungberg, J. (2001). The Power of Gifts: Organising Social Relationships in Open Source Communities. Information Systems Journal, 11(4), 305-320.

Boehm, B. W., Horowitz, E., Madachy, R., Reifer, D. J., Clark, B. K., Steece, B., Brown, A. W., Chulani, S., & Abts, C. (2000). Software Cost Estimation with COCOMO II (1 ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.

Fielding, R. T. (1999). Shared Leadership in the Apache Project. Communications of the ACM, 42(4), 42-43.

Henkel, J. (2003). Open Source Software from Commercial Firms - Tools, Complements, and Collective Invention. Unpublished manuscript.

Holck, J. & Jørgensen, N. (2004). Continuous Integration and Quality Assurance: a Case Study of two Open Source Projects. Australasian Journal of Information Systems (AJIS), Special Issue 2003/2004, pp. 40-53.

Holtgrewe, U., & Werle, R. (2001). De-Commodifying Software? Open Source Software Between Business Strategy and Social Movement. Science Studies, 14(2), 43-65.

Jørgensen, N. (2001). Putting It All in the Trunk: Incremental Software Development in the FreeBSD Open Source Project. Information Systems Journal, 11(4), 321-336.

Koch, C. (2003). Your Open Source Plan. CIO Magazine, 16(15 Mar).

Koch, S., & Schneider, G. (2002). Effort, Cooperation and Coordination in an Open Source Software Project: GNOME. Information Systems Journal, 12(1).

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: "free" user-to-user assistance. Research Policy, 32(6), 923-943.

Lerner, J., & Tirole, J. (2001). The Open Source Movement: Key Research Questions. European Economic Review, 45, 819-826.

Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. The Journal of Industrial Economics, L(2), 197-234.

Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., & Ye, Y. (2002). Evolution Patterns of Open-Source Software Systems and Communities. Paper presented at the Workshop on Principles of Software Evolution (IWPSE), Orlando, Florida.

O'Reilly, T. (1999). Hardware, Software, and Infoware. In C. DiBona & S. Ockman & M. Stone (Eds.), Open Sources: Voices from the Open Source Revolution. Sebastopol, California, USA: O'Reilly & Associates.

Olesen, J. (1992). Concurrent Development in Manufacturing – based on dispositional mechanisms, Dissertation, Intitute for Engineering Design, Technical University of Denmark.

Payne, C. (2002). On the Security of Open Source Software. Information Systems Journal, 12(1).

Pedersen, M.K. & Birk, J. & Hørlück, J. & Jørgensen, N. (2002). Open-source software - in e-government - analysis and recommendations drawn up by a working group under the Danish Board of Technology.

Pedersen, M.K. (1996). A Theory of Information(s). The Business Cycle Model. Samfundslitteratur. Copenhagen.

Scacchi, W. (2003). Issues and Experiences in Modeling Open Source Software Development Processes. 3rd ICSE Workshop on Open Source Software Engineering, May, 2003, Portland, OR.

Stamelos, I., Angelis, L., Oikonomou, A., & Bleris, G. L. (2002). Code Quality in Open-Source Software Development. Information Systems Journal, 12(1).

Torvalds, L. (1999). The Linux Edge. Communications of the ACM, 43(4), pp. 38-39.

Wayner, P. (2000). Free for All (1 ed.): HarperCollins.

West, J., & Dedrick, J. (2001). Proprietary vs. Open Standards in the Network Era: An Examination of the Linux Phenomenon. Paper presented at the Hawai'i Internatational Conference on System Sciences (HICSS-34), Maui, Hawaii.

Wilson, G. (1999). Is the Open-Source Community Setting a Bad Example. IEEE Software, 16(1), 23-25.

Ye, Y., & Kishida, K. (2002). Creating and Maintaining Sustainable Open Source Software Communitites. Paper presented at the International Symposium on Future Software Technology (ISFST'02), Wuhan, China.