JANUARY 2004 • SANJIVA WEERAWARANA • JIVAKA WEERATUNGA

Open Source in Developing Countries

Content

Published by Sida 2004

Department for Infrastructure and Economic Cooperation

Author: Sanjiva Weerawarana, Jivaka Weeratunge

Printed by Edita Sverige AB, 2003

Art. no.: SIDA3460en

ISBN 91-586-8613-4

This publication can be downloaded/ordered from www.sida.se/publications

About the Authors

Sanjiva Weerawarana received a Ph.D. in Computer Science from Purdue University in 1994. After 3 years at Purdue as visiting faculty, he joined IBM's T.J. Watson Research Center, NY, USA in 1997 where he is currently a Research Staff Member in the Component Systems Group. He was previously manager of the group from 1999 until 2002. Since joining IBM Research he has focused on component-oriented programming approaches to distributed application infrastructure and is currently focusing on Web services, where he is a lead architect of IBM's technical strategy. He was elected as a member of the IBM Academy of Technology in 2003.

He has been involved with open-source software both in IBM and in Apache for several years. He was elected a member of the Apache Software Foundation in 2003 where he is active in the Web services project as a lead architect and implementer of several of the Apache Web services projects. He is the co-founder of the Lanka Software Foundation (LSF), a non-profit foundation formed in Sri Lanka to promote open-source participation from Sri Lanka. He is currently the Chairman and Executive Director of LSF.

He is now resident in Sri Lanka and telecommutes to his work in New York. He is also a visiting lecturer in the Computer Science Department of the University of Moratuwa where he teaches classes and helps drive student R&D projects.

Jivaka Weeratunge completed his MBA from the London Business School in 1978, having served articles at Turquand Young & Co. (presently Ernst & Young) and qualified as a Chartered Accountant in 1974. He has worked as a business analyst in a multinational international headquarters in London, as a management consultant at Ernst & Young, as a chief executive of a fee related financial services company Forbes Financial Services (Pvt) Ltd., in Sri Lanka and as a joint founder and managing director of his own management consulting company, Strategy Consulting Group. He was also jointly responsible for establishing and managing a fund management company, in which International Finance Corporation, Templeton Group, National Development Bank, John Keels Group and the Central Finance Group were investors. His last assignment in

general management was to establish an investment holding company, which made strategic investments in mini-hydro power and information technology.

He is the co-founder of the Lanka Software Foundation (LSF), a non-profit foundation formed in Sri Lanka to promote open-source participation from Sri Lanka. He has also been a member of the Board of Directors of the Colombo Stock Exchange and several public quoted companies. He is engaged in a management consulting practice at present.

1 Executive Summary

The case for a developing country to adopt an open source software (OSS) driven Information Technology strategy is a compelling one. In a wider context IT is about the production, flow and control of information in a modern state and the rules which govern such activities are thus fundamental to its economic development. Software is a driver of those rules. OSS under these circumstances cannot be viewed as a mere product choice. It reflects more fundamentally an alternative strategy for building, maintaining and changing the rules that govern information flows in the economy.

To create value or wealth in a developing country economy as a matter of pragmatic strategy it must create business opportunities for private firms, reduce cost of IT investment in the private and/or public sectors of the economy and improve the efficiency and effectiveness of government. A strategy map has been developed to demonstrate that OS based strategic initiatives will create value through the key drivers of business opportunities, reduced investment cost and greater efficiency and effectiveness of government. These strategic initiatives which are primarily driven by government encompass IT policy, advocacy and education, capacity building, E-government and positioning of the country's IT industry in a globally competitive one. The actual execution of such strategies require in varying degrees, collaboration between government, universities and educational institutions and private firms. It must be emphasized however that the momentum generated by such initiatives would also result in the emergence of market driven business opportunities, which private firms can and must exploit. In developing such a strategic framework the risks and opportunities faced by developing countries were considered, in addition to which the very phenomenon of OS software, its impact on software markets, the viability and profitability of the business model that it drives and its long run sustainability were examined.

The reality of the OS phenomenon today is that the majority of the Internet infrastructure is based on OS products including send mail, BIND, Linux and Apache HTTPD, the latter running more than 69% of all active websites (approximately 15 million) with Microsoft IIS coming a distant second with less than 23% share. Similarly Linux has made a relentless assault on the server operating systems market, and the busi-

ness models driven by this outcome demonstrate their viability and profitability. It is still not clear whether OSS will thrive in traditional middleware application markets as opposed to the software stack in which it is significant, operating systems, device drivers, Internet services, basic HTTP servers and entry Web application servers. The latter categories of software serve well comprehended and standardizable needs. In the area of business models OSS is associated with technical support and integration services, add on products, priced licensed software which is inclusive of the OSS, software subscriptions for periodic, functional and fixed updates and custom software development contracts.

The OSS phenomenon could easily lead one to believe that it is substantially driven by the individual motivations of highly committed software developers on account of personal interests and the joys of programming – primarily self actualization needs. In addressing the question of OSS sustainability and evaluating the evidence of the motivational models which have been developed to explain this phenomenon it appears that economic motivations of future returns and building or creating personal brand value could have a greater weight (considered external factors) as against internal factors such as intrinsic motivations and altruism. Thus in [Hars 2001] it is concluded that "the open source movement can draw from a diverse set of motivations, a large part of which is based on external rewards. It is poised to become a strong competitor to traditional software development".

Traditional software development generally tends to suffer from both time and cost overruns and when it is eventually delivered its quality and reliability can sometimes be serious issues. OSS development on the other hand is viewed as a process by which reliable and high quality software is produced quickly and inexpensively. The process has been associated with parallel development rather than linear, involving large communities of globally distributed developers (highly talented and motivated), providing prompt feedback to user and developer contributions with truly independent peer review, increased levels of user involvement and extremely rapid release schedules. The substantial performance gains associated with OSS development are the result. To avoid chaos, as there is no formal project management, many cultural norms dictate the management of such projects. The OSS development life cycle too is very different to the traditional one and getting design issues right is even more critical to OSS than it is in conventional development. This is said to explain why many OSS products are horizontal infrastructure type ones where the requirements are generally well defined. Both motivational and organizational models of the OSS phenomenon serve to explain its historical development, its successes and its future sustainability side by side with traditional development of proprietary software products.

Many governments around the world have initiated the use of OSS as a key part of their strategic thrust in information technology, motivated by the reduction in cost of IT investments they would have to make in addition to the desire for independence, a drive for security and autonomy, and a means to address intellectual property rights enforcement. Government software procurement policies, which require the use of OSS particularly in the case of most countries, which have pursued this

course in the use of Linux, have clearly been driven by the aforementioned considerations. Increased emphasis on accountability or transparency in public sector governance has driven the need for open public data formats. Thus the usability and maintenance of software independent of suppliers with considerable market power and monopolistic conditions imposed by them, and the alternative to dealing with proprietary IP regimes in contrast with software piracy which means a reduction of IT investment cost within a framework of compliance, are reasons for the gathering momentum in the use of OSS in both developed and developing countries alike. Outside of the risks which developing countries face with alternative IT policies, the empowerment of the IT industry of a developing country through OSS development is an important opportunity which has been identified by many researchers and in particular Steven Weber in his paper "open source in developing economies" [Webe 2003]. A strong case has been made that such economies with a surplus of inexpensive technical manpower could combine the free software tools that the OS phenomenon provides, creating an interesting comparative advantage both in local and global markets alike. The sentiments expressed by President Abdul Kalam in relation to the IT industry in India, the consortium of Japan, China and South Korea for OSS development, strategies adopted by South Africa and the African region, point to a clear recognition of strategic opportunities that developing countries could exploit utilizing the OSS phenomenon.

If a comprehensive strategic approach is to be taken by developing country governments in the pursuit of value creation in the economy through IT as mentioned previously, the key drivers of reduced cost of IT investment, business opportunities and the improvement in efficiency and effectiveness of government could be driven by a strategy package of IT policy, advocacy and education, E-government, capacity building, industry positioning and building brand equity. The IT policy framework would naturally have many components including government software procurement policy, which recognizes the role of OSS in the public sector. This would also encourage sound private sector software procurement policies, particularly the non-investment cost related benefits of OSS in terms of capacity building and industry positioning and the related business opportunities, which these would give rise to. The IT policy framework would have to encompass some key aspects, the advocacy and education strategy initiative, as without such intervention the developing country market place would continue to be dominated by licensed or unlicensed proprietary software as the case maybe. Establishing OSS advocacy groups and an OSS portal are some of the action programs which may be required in addition to the initiatives in education which have been recommended such as OSS computer driving license, introduction of OSS to university curricula, establishing of training programs through professional organizations, introduction of OSS to elementary, middle and secondary schools and the like. Increasing the awareness of OSS as a platform and supporting the development of easier deployment and experimentation platforms would be necessary for an effective roll out of OSS in practice. Localization is also an area which demonstrates the advantages of OSS in building people/community oriented applications. A condition precedent to the success of an OS

strategy is the capacity of the local industry to implement OSS in addition to the traditional and other capacity building measures mentioned already in the form of educational and training programs. It is important to target capacity building through adaptation of existing and development of new OSS products. Establishing localization centers, sharing localization skills with other developing and less developed countries, establishing OSS solutions and R&D centers are some of the action programs that have been recommended in the pursuit of building capacity and skills of the local software industry.

In positioning a local IT industry which would have good IT skills and a reasonable infrastructure, global OSS development offers unprecedented opportunities. The key drivers of a strategy in this area include participation in global OS software development projects such as Apache in the more significant areas of the software stack, the availability of the right software development skills and a vehicle or mechanism to mobilize such resources and an industry which has the vision and capacity to pursue OSS based business models and exploit global opportunities arising from the skills and brand value that would be accumulated as a result of active and successful participation in this domain. It must be emphasized that the gap between strategy formulation and execution can be awesome in the absence of the right ingredients of leadership and human resources which can contend with the domain expertise and networking abilities that are necessary. The participation of individual software developers in developing countries in the absence of an organized framework would largely be a hit or miss effort, with much less potential to build brand value and the country's reputation in the global software industry.

In the formulation and execution of an E-government strategy the adoption of open standards and OSS would have considerable strategic and operational implications. In the context of limited resources the savings in investment cost through the adoption of OSS may make the difference between actual execution and failure, in addition to which the goals of security, independence and transparency could be secured. The importance of a well-conceived architecture with the development of open standards and OSS components is an action program, which has been recommended for execution in implementing E-government.

Donor agencies can hardly ignore the compelling reasons for interventions necessary to assist developing countries to realize their goals and strategies in IT, unless they are prepared to irretrievably destine these countries to the dustbin of history. As much as IT is a key driver of economic development in the modern world, OSS as a part of the country's IT strategies and policies undoubtedly, if sensibly and rigorously applied, create considerable value in the economy. The basis on which donor agencies can target their assistance and formulate and monitor their own programs has been addressed through the roll out of a strategy map with the identification of the key strategic initiatives and action programs which are necessary for the effective execution of such an initiative. The strategy package, which is appropriate to an individual developing country or least developed country, could be determined on the basis of a simple analytical framework, based on the strength of IT infrastructure and skills and the status of IT policy. Where IT policy is

not stated and IT infrastructure and skills are weak, strategic initiatives in IT policy, advocacy and capacity building must take precedence. On the other hand if IT infrastructure and skills are strong, the strategic thrust should be in positioning the local IT industry within the global IT industry. If IT policy is explicit and IT infrastructure and skills are weak, capacity building must be the emphasis in strategy with advocacy and education playing a supporting role. Where IT policy is explicit (and is being executed) and IT infrastructure and skills strong, considerable short and medium term benefits and outcomes could be expected if positioning and E-government initiatives are both rigorously and speedily unleashed. In addition to playing what is a critical catalytical role in economic development while supporting OS based IT initiatives in developing countries, the donor agencies of the developed countries should not forget the self interest they would be serving if collaborative models between developed and developing countries in the areas of capacity building, E-government, development of OSS products are pursued. The OSS phenomenon does provide some unprecedented opportunities for collaboration, which could be significant in the dynamics of the global IT industry. Another aspect, which donor agencies in particular would wish to address, is what OSS does to the competitive market place for software, production and services. The market structure and business models that have been reviewed make it manifestly clear that the OSS phenomenon has made the market place more competitive and provided greater choice to users, in an industry where proprietary software products and large industry players have carved out for themselves significant parts of territory. It is significant that OSS has not been driven by government intervention but by the power and commitment of individual initiatives, which have translated, into something of significant economic impact. Individual skills and commitment to remain at the heart of the process of software development and expert power that underlies this has had the circumstances and the means to manifest itself. The case for donor agency support given this situation and unprecedented opportunities that are there for value creation in the economy through IT is very compelling. Concepts of market distortions should not be permitted to cloud the OSS phenomenon as it creates the very market place and competitive dynamics that donor agencies desire to espouse.

2 Introduction

”Open source” is the topic of the day in the software industry. Open source is fundamentally a model of distributed, shared, open software development. The dramatic success of open source products such as Linux have fuelled the believers to predict the imminent doom of proprietary software as well as the nay-sayers to assert that open source is just another passing fad in the IT industry. The truth lies somewhere in the middle between these two extreme positions, as always.

While the open source phenomenon is indeed truly open in the sense of being open to all in the world, contributing developers are surprisingly concentrated in the developed world. Recently, however, the economic benefits of open source technologies have attracted a growing move towards adopting open source software (OSS) options in developing countries to a point that now the developing world is leading the developed world in open source adoption.

In this report we study the open source phenomenon as it applies to developing countries. Our objectives are to provide an analytical framework to understand the open source phenomenon and to provide a strategic framework by which developing countries can adopt, exploit and contribute to the open source phenomenon. Using this framework, we consider the role that funding agencies such as the Swedish International Development Agency could play in assisting developing countries in implementing the strategy.

2.1 Scope of Study

The scope of this study covers the following topics:

- Open source in general – what open source is, its history, who the players are and how it works,
- Open source in developing countries – the role of open source in developing countries, strategies, business opportunities and policies.
- Open source and donors – should donors support open source and, if so, how?

The area of open source has been widely studied and written about. This report is based upon, and builds on, the results of many of these works, which are referred to throughout. The focus of this study is to develop a

rigorous model for open source to create a strategic framework that can be executed upon successfully in developing countries.

2.2 Key Issues

In order to address the scope of the study, we identify the following key issues that will be discussed as part of this study.

- *Is open source significant and relevant in the software industry?* Clearly a precursor to having an open source strategy for developing countries is to first address this question. The sections on software market and business models address this question, answering strongly affirmatively (sections 4 & 5).
- *Is it sustainable?* If OSS is simply a passing fad as some claim then it is clearly something to avoid. In the section on open source phenomenon and dynamics we address this question and provide a rationale for why it is indeed sustainable and hardly a passing fad (section 3).
- *Can developing countries create value through OSS?* We address this question by developing a strategy map that lays out an agenda for how OSS can create economic value through OSS and how that that strategy can be implemented. This is covered in sections on OSS strategy for developing countries and strategy implementation (sections 6 & 7).
- *If yes, how can donors support it?* Since OSS can indeed be a vehicle for creating value for developing countries, we provide a framework for how donors can support OSS activities in developing countries (section 8).

The rest of this report considers each of these issues in detail and provides approaches for addressing them. The information contained here represents the authors' best judgment, at the current time, of open source in developing countries. We have made every attempt to acknowledge all other works that we have built upon and any omissions are purely unintentional.

2.3 What is Open Source

Open source is an evolution from the long standing practice of providing the source for software. This approach has been the popular distribution model for key Internet infrastructure software systems such as BIND (the widely used implementation of the Domain Name System) and Sendmail (the widely used electronic mail software) for probably 20 years. System administrators often customize/localize these systems prior to deployment and having the source readily available made it possible. However, the original software itself was typically developed by one person/group in one location.

The novel concept of open source software is the notion of community development. With the popularization of the Internet in the late 90s, it became feasible for not just one person or a team in one geographical location, but groups of interested persons in geographically dispersed locations to jointly develop software. In general, any person can participate in any project. However, the degree of control or direction setting authority a participant gets is a function of how active that person has been in that project and how long he has participated.

2.3.1 Open source developers and developer community

An important aspect of the open source culture is that participants are individuals and not organizations. Open source developers are often individuals who often have a different "day job" but yet contribute to open source projects for reasons of self-satisfaction, self-improvement and other motivating factors. In today's software development world, a majority of developers participate in very "high level" software development which is targeted at specific products or user groups. Participating in open source projects allows developers to have much deeper and fundamental impact on the computing platform of the future than by building higher level software. One of the key aspects of this study is to examine the motivation model for open source to consider the long term sustainability of the phenomenon.

The success (or failure) of an open source project is almost always characterized by the quality of the developer community around that project. The more active, diverse and lively a developer community is for a given project, then the more likely that the project would succeed. For this reason alone, one of the most successful open source software foundations of the world, the Apache Software Foundation, uses community strength as the litmus test to decide on whether to embark on a project.

Typically, open source projects have a relatively small number (often less than ten) of key thought leaders and architects that engineer the overall system and develop the majority of the code. A healthy open source project will see new developers coming along, being active for some time and then "retiring" from that project or moving on to other projects. Thus, the culture encourages community churn and in fact thrives on it; the "new blood" in the project often brings the adrenaline needed to make drastic changes periodically.

2.3.2 Impact of Open Source Software

The relative chaos of the open source culture may lead one to assume that software developed under this approach will always remain amongst the "hacker" community. However, the reality is exactly the opposite – today a majority of Internet infrastructure is based on open source products: including Sendmail, BIND, Linux and Apache HTTPD.

Apache HTTPD, the Web server developed by the Apache Software Foundation now runs more than 69% of all active Web sites (representing nearly 15 million sites). The Microsoft Web Server (IIS) comes in a distant second (at a bit below 23%) [Netc 2003] and no other vendor's Web server has any significant market share. As a further testament to the success of Apache HTTPD, several major software vendors have dropped internal Web server products and adopted the Apache HTTPD software for use in their higher level software products.

The same success story is being repeated in different parts of the Web computing infrastructure. We are already seeing the Apache Jakarta project's Tomcat servlet engine [Apac 2003] and its various components being the basis of software vendors' higher level products. It is also happening on the base programming model infrastructure of Java – many of the libraries used with XML for example are all open source software packages.

2.3.3 Legal Aspects of Open Source

One of the most common concerns about open source is related to open source licenses, patents and copyrights. A plethora of open source licenses have emerged over the last 10 or so years; all of them offering different rights with respect to licenses, patents and copyrights. This is a well-studied area and in Section Appendix A we provide an extract from [Drav 2003] which discusses how the legal landscape of OSS is evolving. The FLOSS report [FLOS 2002a, FLOS 2002b, FLOS 2002c] also provides a very good summary of the legal aspects of open source. The Open Source Initiative [OSI 2003] has an established certification program for open source licenses [OSIL 2003] which guarantees that the certified licenses grant the fundamental rights of openness associated with open source.

3 Open Source Phenomenon and Dynamics

3.1 Historic Development

This historic development of open source has been studied widely. The following is extracted from [FLOS 2002a, FLOS 2002b, FLOS 2002c].

From the early 1960s to the early 1980s, revenues in computer business were generated through selling and supporting hardware. For every hardware device, a special operating system was developed and deployed. The users of these systems were highly Specialized IT experts. They were the ones primarily responsible for the development of additional software.

Many efforts were dedicated to build an operating system that could be deployed on multiple hardware platforms. The most prominent example was Unix, which was developed at the AT&T laboratories and was published in 1969. Commercial users had to pay high license fees for using Unix, whereas academic institutions could use the software for a nominal charge. Consequently, Unix was the basis for the development of the Internet technologies. Many of these technologies were developed at universities and computer companies research laboratories, where Unix was deployed. Sharing the source code among software developers was commonplace. This tendency was reinforced by the emergence of computer networks like the Usenet that was started in 1979 to link the Unix community. A critical event in the early 1980s for cooperative software development was the turn-around in AT&T's licensing policy. Unix became restricted to those who paid for the license to use is. Following this first step into the direction of closed source, the hardware companies IBM, HP and DEC started to develop proprietary Unix operating systems. They imposed "non-disclosure agreements" on the programmers dealing with the software and recruited many developers for commercial software development who had formerly contributed to cooperative and shared software development.

At that time, the programmer Richard Stallman worked in software development at MIT. In 1984, he started a project to develop a free alternative of the Unix operating system. In addition, he established a special license, the GNU (named for Gnu's Not Unix) license, which was supposed to ensure that the software is indeed free and open for everyone. In order to support the GNU project, Stallman founded the Free Software Foundation (FSF) in 1985 [FSF 2003]. Although linked often to the Open Source movement, Stallman is a proponent of Free Software, which goes much further in its demands.

Nevertheless, the GNU General Public License (GPL) is central to the evolution of the Open Source phenomenon and has been used in many

important projects. In the GPL, the principle of "Copyleft" is realized: It means that every copy of a program governed by the GPL, even if modified, must be subject to the GPL again. The licensing principles of the GPL, especially the "viral"[1] effect, are not suited for use in commercial software development as they make license fee-based revenue models impossible.

The FSF's philosophy behind software development provided great motivation for the Free Software community. But it also resulted in antipathy from many businesses which partly remains until today. The most prominent debate over the implications of Open Source Software, especially the GPL, and its effects on innovation takes place between Microsoft and Free/Open Source Software advocates, although such discussions are commonplace in more prosaic settings as well. In the early 1990s, along with the increasing use of the Internet and the success of the World Wide Web, many new Open Source projects emerged. The most prominent example is Linux. Linux is an Unix-like operating system targeted to run on a personal computer. It was developed by the Finnish computer science student Linus Torvalds who used the GNU software tools. In 1991, he released the code of an experimental version under the GPL to a newsgroup and asked for comments and improvements.

Within the last decade, Linux developed into a powerful operating system. The project shows characteristics that are typical for successful Open Source Software development over the Internet. Eric Raymond, another central OSS developer and advocate, describes OSS development coordination as "Bazaar style," opposed to the "Cathedral" approach taken in classical software development, where development is organized in a more hierarchic, top-down and planned way. Linux has a modular structure, so individuals or groups of developers can focus on one part of the program. The principle of "Release often, release early" in combination with a constant peer-reviewing process ("Given a thousand eyes all bugs are shallow") is also opposed to commercial software development.

Linux was used increasingly in combination with the GNU tools. Because the operating system is central to IT infrastructure, it eventually became relevant for business use. In 1997, the Open Source Initiative (OSI) [OSI 2003] was founded in order to establish a more pragmatic approach to software licensing. The OSI was based on the "Debian Free Software Guidelines," which had been published in 1995. The central people for this development was Eric Raymond and Bruce Perens. Their aim was to promote OSS in commercial use because they believed that both the Free/Open Source community and the business world could benefit from wider OSS dissemination.

The OSI developed the Open Source Definition (OSD). The definition is not a license itself, but a guideline and trademark for OSS software licenses other than the GPL. Licenses according to the OSD guarantee several freedoms to software users, including commercial users. The "viral" effect of the GPL is not a requirement for OSD-approved licenses. In order to raise acceptance of OSS in the business world, the term Open Source Software instead of Free Software was established and widely accepted.

The 1990s experienced a significant rise in attention paid to Open Source projects. Many companies from the IT industry began to support the projects. IBM, for example, supports a variety of Open Source projects. In 1998, Netscape was the first prominent company to release a proprietary software product as Open Source software.

[1] "Viral" means that every derived software or software linked to any software published under the GPL has to be licensed under conditions that are compatible to the GPL.

3.2 Motivational Model

The review of the OSS phenomenon may lead one to believe that it is substantially driven by the individual motivations of highly committed software developers purely on account of their personal interest and the joys of programming. In addressing the question of whether OSS is sustainable in the long run, it is imperative that we comprehend the underlined drivers of this phenomenon. It is our contention that a behavioral model driven by "self actualization" and altruistic motives on the part of the key players in this arena the individual developers, is insufficient to deal with what is a more complex phenomenon. Our hypothesis is that in addition to the above mentioned factors, clearly of significance, economic motivations of future cash flows or returns and building or creating personal brand equity, related to the potential for future economic returns, are equally or more important in comprehending OSS and its sustainability.

A comparison with a creative piece of writing which is published in book or other form with a computer software program which would be read or used as the case may be by a substantial public, and the consequences for the authors in the respective domains, may throw more light on this issue. The joint authors of a creative work enjoy the intellectual property rights and the reputation resulting from a successful launch of such a publication. On the other hand the authors of a very successful and widely used software program would be rarely identified. The intellectual property rights clearly belong to the employers, corporate body in most instances.

In their article, "Working For Free?" motivations for participating on OSS projects Hars and Ou [Hars 2001] "identify two broad types of motivations that account for participation for OSS projects. The first category includes internal factors such as intrinsic motivations and altruism, and the second category focuses on external rewards such as expected future returns and personal needs". In fact the authors demonstrate that the external factors have a greater weight. Building human capital and self-marketing clearly fall within the domain of future economic return. The survey undertaken by the authors quoted in their paper demonstrates that different groups place different weights on the importance of the two categories; e.g. hobbyists and students are the most intrinsically motivated as against salaried and contract programmers seeking a future economic return. Very significantly the results also indicate that a high percentage of developers are actually paid for their open source development efforts. This group is the most concerned with self-marketing and the fulfillment of personal needs. Thus, it is clear that "the open source movement can draw from a diverse set of motivations, a large part of which is based on external rewards. It is poised to become a strong competitor to traditional software development".

In "Understanding Open Source Software Development" by Feller and Fitzgerald [Fell 2002] a framework has been proposed that considers a broad motivational model with three areas: Technological, economic and socio-political. Each of the areas is further divided into an individual developer perspective (micro level) and an organization/community perspective (macro level):

- Technological Motivations: Individual perspective
  - To meet a personal technological need: Almost all successful OSS projects have been initiated by someone who had a technical need that was not being addressed by available proprietary technology.
  - To exploit the efficiency of peer review etc.
  - To work with "bleeding edge" technology
- Technological Motivations: Organization/Community perspective
  - To address the software crisis, particular poor quality – Independent peer review by a large and talented pool of developers is likely to lead to high quality software
  - To share tedious development tasks (testing, documentation) with users
  - To leverage the OSS community for R & D – innovation that the OSS process affords is considered a key reason.
  - To promote innovation
  - To ensure transparency of the application – The extent to which the key IT infrastructure is dependent on OSS products with no legal owner or contracted service supported agreement could be surprising to many organizations. Raymond has suggested that buying "closed – source software for your key infrastructure is considered the height of irresponsibility".
- Economic motivations: Individual Perspective
  - To gain future career benefits – The economics of signaling incentives has been adopted to capture the economic incentives behind developer participation in OSS. This covers both ego gratification and career concerned incentives. The ego gratification incentive is based on peer recognition, discussed subsequently in socio political motivations. The career concerned incentive deals with the enhancement of future job prospects of developers working on OSS projects.
  - To improve coding skills – The review by respected peers is admittedly a key driver of the effort developers are putting into their coding. In the survey conducted by Hars and Ou [Hars 2001] 70% of respondents reported the desire to improve their programming skills as the motivation to contribute to OSS
  - "To strike it rich" through stock options etc. – There is direct evidence of OSS development work being paid for – the Hars and Ou survey found 50% of respondents were paid for their OSS work. Given that phenomenal performance of OSS related IPO's share option and share ownership schemes targeted to developers, could be a significant incentive
  - Low opportunity cost – nothing to lose – Developers may have nothing to gain in economic terms by keeping developments to themselves. Thus there would be nothing to lose by opting for OSS.
- Economic Motivations: Organization/Community Perspective
  - To exploit investor infatuation with OSS – There is much evidence that the revenue generating capacity of OSS is not as outrageous

as it may initially have seemed. This is illustrated by the staggering performance of Red Hat and V. A. Linux in their IPO launches.

- To embrace the paradigm shift from software as a commodity industry to a consumer driven service model – The paradigm shift to the software as a commodity model was initiated mainly by Microsoft who wrested control of the industry from the primarily hardware oriented model dominated by IBM et al. The shift to a customer service model appears to make intuitive sense from an economic point of view, as software has zero reproduction cost but very high service and maintenance cost. The vast proportion of the total cost of software development is incurred supposedly in the maintenance phase. The model for proprietary software based on a high purchase price with a low maintenance fee does not reflect the reality of the cost distribution. This is recognized in the OSS model where the software is distributed for a nominal fee and companies compete on service to the consumers. High reliability of OSS products would result in lower support cost than for traditional software development. Brand management becomes critical and customers learn to value a brand they can trust in terms of quality, reliability and consistency.
- To raise mind share and strengthen brand
- To exploit indirect revenues – selling related products and services, accessorizing etc.
- To make software affordable in developing countries
- To cut costs – cheaper platform than proprietary alternatives

– Socio Political Motivations: Individual Perspective

- Ego gratification and signaling incentives – Developers experience a significant rush from seeing their code in use more quickly in OSS projects. The recognition they receive is often from peers they truly respect. The more attention an OSS developer can attract the greater the enhancement of status that is achieved.
- Intrinsic motivation of coding – Underlines the intrinsic motivation factor of Hars and Ou [Hars 2001] is Mazlow's hierarchy of needs in the area of motivation that identifies self actualization and esteemed needs of individuals. Within this framework OSS is comparable to pursuing a hobby that is intrinsically satisfying.
- Sense of belonging to a community – Belonging to a global community of developers engenders a feeling of power in the leveling of the playing field. Hars and Ou indicate that more than half of the respondents sighted the sense of community belonging as the reason for their participation in OSS projects.
- Altruism – The need to help their community is a motivator and is also related to Mazlow's framework.

– Socio Political Motivations: Organization/Community Perspective

- Social movements require an enemy E.g. Microsoft – It is said that all social movements require an enemy and Microsoft appears to have assumed the mantle of anti-hero for the OSS community. Ideology has obviously played a big part in OSS. It also plays a

part at the level of desiring to rectify global iniquities and disparities by providing information technology free to the developing countries

- Overcomes "digital divide"
- Ideology – software must be free – Stallman's FSF tried to provide a complete family of free software alternative to commercial packages, and many were drawn by such an ideology, even if open source emerged as a means of diluting the blatant ideological connotations of free software. Importantly OSS is not a movement that can be pinned down easily as one that follows a particular ideology.
- Model for wider domain – future model for work – Recent accounts of OSS have stressed the potential of transferring the model beyond software to the organization of work.

3.3 Organizational Model

The organizational model of OSS has been carefully studied in [FLOS 2002a, FLOS 2002b, FLOS 2002c]. The following is an extract from those documents.

Software development tends to suffer from both time and cost overruns and when it's eventually delivered its quality and reliability becomes more questionable more often than not. Open source however has been viewed as a kind of silver bullet to solve this software crisis and is considered as a process by which reliable and high quality software is produced quickly and inexpensively. The OSS process however has no orthodoxy and the situational context has driven the methods and tools, which are adopted for this purpose. However a generic OSS development process is associated with the following characteristics:

- It is parallel rather than linear
- It involves large communities of globally distributed developers
- It utilizes truly independent peer review
- It provides prompt feedback to user and developer contribution
- It includes the participation of highly talented and motivated developers
- It includes increased levels of user involvement
- It makes use of extremely rapid release schedules

Parallel development is a key aspect of the OSS process and is a result of the highly modular nature of many such products. It means that individual or small groups of developers work on one aspect of a large system at the same time that others work on another aspect of the system. Thus it can lead to substantial performance gains with a very large number of developers collaborating efficiently. It is also important that every member of an OSS project does not need to know everything about the project. There is another aspect of parallel development – redundancy. The OSS context improves product quality by allowing multiple solutions to the same problem to compete with each other. Thus any performance loss associated with redundant activity is compensated for by the substantial gains in quality which results. The luxury of executing processes of this nature relate also to the size of the development community in an OSS project.

The global nature of OSS projects improves the overall quality of both process and product in significant ways. Functional requirements for instance would be determined from an international perspective rather than a US centric one. It cannot be forgotten that the underground, anti-establishment demeanor of the OSS tends to attract contributions from students and many others round the globe.

In traditional development environments peer review of a software product is not guaranteed to be truly independent. With OSS however a distributed development community can ensure a truly independent peer review. Further, this process is taken to a new level as the code is submitted to the OSS community for review and feedback first with suggestions and contributions from the community being reviewed again by the original author and the community. The OSS model is perceived as "shared knowledge plus peer review" and is considered as being akin to the academic model.

A feedback in OSS is rapid and even continuous since developers can work asynchronously in different time zones, and is described as a constant cycle of "see bug, fix bug, see bug fixed in new release" very different from the proprietary model. Various tools are utilized to ensure that user to user, user to developer and developer to developer communication is efficient. However, this process is successful because quality contributions from any source are recognized and users are treated as co-developers.

The high level of ability and commitment of OSS core developers is clearly critical to the success of projects. The reality is that leading pioneers in the various OSS projects have actually come from the top bracket of the programming profession. The reputation of the co-developers is also a means to attract others who are keen to work with the experts. It is important to recognize that "in a reputation driven culture, the supreme authority of a universally recognized gifted software leader, or acknowledged board of directors, is vital to prevent squabbles over the right to have one codes included. Given that conventional project management control is not possible, some unquestionable authority is ultimately necessary. The precise arrangements of vesting this authority vary in different OSS projects" [REF].

Developer-user collaboration is considered to be at the heart of OSS. Users when treated as the developer's most valuable resource in the OSS process, it appears that at least partially the hierarchical order of coders, testers and documenters seen in traditional software development could be inverted. There is evidence that in the OSS environment users "perform the majority of the testing task or request new features".

The entire OSS development process is focused on the product and its user and the size of the development community, promptness of communication and feedback and the distribution of essential development tasks support a rapid incremental release pattern.

Given the very nature of the OSS process and its reach, configuration management tools are critical to its success. Feedback and contributions can come from any where at any time and the lead developers who have to moderate or co-ordinate such contributions can very easily become a bottleneck. Concurrent Version System (CVS) is the most common tool used for configuration management. Criticism has been leveled against OSS tools about functionality and their inadequacy except to say that the desire for analysis and design for tool support does not coincide with the fervor to develop OSS products. This particular issue remains unresolved.

To avoid chaos as there is no formal project management; there are many cultural norms that govern the management of OSS projects. Some are in the form of taboos. The desire to avoid projects splitting into rival and competing

development streams termed forking is one such taboo. In an ego based economy and a culture driven by a reputation model this can be a real danger. Plagiarizing work as one's own by removing the credit to the rightful contributors is another such taboo. Rapid recognition is a key motivator of developers. A norm that appears to be important to the community is modesty and self-deprecation on the part of developers. This too is vital as contributions from others drive the entire OSS model.

The OSS development life cycle is very different to the traditional one of planning, analysis, design and implementation. Planning, analysis and design phases are largely undertaken by the initial project founder and thus may not be part of the OSS development cycle. However it cannot be over emphasized that getting design issues right is even more critical to OSS than in conventional development. This may also serve to explain why many OSS products are horizontal infrastructure type ones where the requirements are generally well defined. The OSS development life cycle is located primarily within the implementation phase and maybe disaggregated as follows:

- **Code:** The submission of code for review by very talented and respected peers tends to be a real incentive to improve the quality of such code.
- *Review:* Truly independent peer review is a central strength of this process. However it has been found that "heavy weights" with a proven reputation tend to get lot more feedback.
- **Pre-commit test:** To avoid breaking the build it is vital that committers test each contribution carefully. Testing is very much a personal process. There is no requirement that test scenarios be rigorously planned and tested in advance. However the testing process is taken very seriously as the negative implications of permitting a faulty contribution can be considerable.
- **Development release:** If the committer deems the module ready, it can be incorporated in the development release. This is the key reward for many developers when they see their code implemented quickly.
- **Parallel debugging:** Although there is no formal plan for debugging the power of debugging arises from the numbers of potential debuggers available. This also serves as an opportunity for those wishing to initiate their contribution on OSS projects
- **Production release:** Stable production and development versions are maintained separately and contributions eventually become part of the production release.

4 Open Source and Software Markets

In this section we examine the software marketplace and how it relates to OSS. There has been much written about this topic. Rather than repeat that material here, we provide a high level summary from these studies here. However, as a convenience to the reader, we have included an extract on OSS and software markets from the excellent summary in [FLOS 2002a, FLOS 2002b, FLOS 2002c] in Appendix B and strongly encourage the reader to study it.

The software products and services market within the IT market can be considered in three segments: software products, embedded software and services, and professional software-related services.

4.1 Software Products Market

Looking at the software products market, we can further segment it based on the types of customers: the packaged mass market software market and the enterprise solutions market. The latter is typified by many Microsoft products and the latter by many IBM products and solutions. Within the packaged software market, further segmentation between server and desktop markets is possible. Thus, there are three broad market segments:

- Server operating systems and applications,
- Desktop/client operating systems and applications, and
- Enterprise solutions.

Within the server operating systems and applications market, different players dominate the high-end and low-end segments typically. The customers in this market are quite different from those of the desktop market and often purchase solutions rather than products. OSS has strong influence under these conditions as it is ideally suited for packaging within solutions. Furthermore, because of Linux's rich Unix heritage, there are many server applications that be easily ported to Linux, thereby increasing its value proposition while retaining the fundamental advantage of near zero license cost. OSS has clearly made the software market significantly more competitive in this segment by enabling a larger set of ISVs (independent software vendors) and VARs (value-added retailers) to compete effectively with traditional platform owners.

The desktop operating system market segment is very strongly dominated by one player, namely Microsoft of course. This market demonstrates classical mass market characteristics where the market leader has a strong position and, if the leader continues to innovate and stay ahead, will continue to dominate. Linux and other OSS operating systems appear to be gaining marketshare in this market (especially in developing country government markets), but at current growth rates predicting any kind of balance in that market is perilous.

The desktop application market can be segmented to horizontal applications (such as office productivity applications) and vertical applications. While the OSS horizontal application software is maturing, it is still not technically comparable to proprietary versions; thus they still have little marketshare. As this segment matures in technology, it is likely that OSS products will "catch up" and offer significant competition to proprietary ones.

The enterprise solutions segment consists of applications such as ERP (enterprise resource planning) systems and CRM (customer relationship management) applications. These solutions almost always are based on large service engagements where much of the revenue is service driven rather than product driven. This is largely due to the heavy customization and maintenance required of these solutions.

4.2 Embedded Software and Services Market

The customers in this market are typically special device producers, such as cell phones and set-top boxes. This is a classical B2B market as the end user has no direct relationship with the embedded software provider. OSS has many advantages in this segment – including availability of source enabling vendors to customize the software, the low price and the availability of technical skill. However, many special devices are real time systems and the flagship OSS operating system, Linux, is not a real time operating system (RTOS). The most successful RTOS' today are yet proprietary. OSS is undoubtedly making the market more competitive but there are several barriers to be crossed to become the dominant player in this segment.

4.3 Software-Related Services Market

Software service companies are a very dynamic marketplace because of the very low entry barrier to enter the market. The service business demonstrates a high pace of innovation as the software products market demonstrates such a pace. Human capital is the critical success factor in this market as software services are a classical people-selling business with high permanent marginal costs and there is little cost-sharing across even similar projects.

OSS infrastructure is slowly but surely becoming a bigger and bigger part of many IT solutions. As the available OSS componentry matures, OSS will work to reduce the costs of solutions as expensive license costs can be avoided.

5 Business Models Based on Open Source Software

In this section we examine business models based on OSS and discuss how some or all of these may be applicable to developing countries. There has been much written about business models based on open source software. Rather than repeat that material here, we have provided a high level summary from these studies. However, as a convenience to the reader, we have included an extract on OSS business models from the excellent summary in [FLOS 2002a, FLOS 2002b, FLOS 2002c] in Appendix C and strongly encourage the reader to study it.

Almost all IT organizations today have some interaction with OSS today. However, for the purposes of this study we only consider pure-play OSS businesses – that is, businesses which would not exist without the existence of the OSS phenomenon. Business models for such organizations can be broadly classified into two categories:

(a) those that distribute and retail various open source products; and

(b) those that offer various OSS related services.

Due to the focus on system function and code quality rather than ease-of-use, classic OSS is typically not suitable for use by average developers and most certainly not by average users. Distributors address that gap by packaging quality open source software in a manner that is much more widely accessible to the user community. Within businesses in the distributors and retailers category, Linux distributors have become quite widely known and successful by addressing this problem for Linux by providing point-and-click type convenience.

However, Linux is unusual in being suitable for packaging and direct marketing to consumers. For many other OSS products, even for example the wildly successful Apache Web server, packaging and direct marketing has not been a successful business model. Instead, many such OSS products are packaged and incorporated in larger solutions by system integrators and OEM vendors. Thus distributors who specialize in packaging and retailing niche OSS products have not been successful in general.

The business of retailers of OSS complementary products has however been very successful. Successful business models here include publishers such as O'Reilley as well as organizations that operate user conferences.

The second major category of OSS business models are around OSS related services. These can be broken into two major categories: those that enable the OSS development process and/or community itself and those that provide service and support for OSS products. Several organizations have found success in the community support business, especially in enabling open source development in large corporations.

The service and support for OSS products business model has been the most successful model thus far. Many organizations have developed specialized skill by participating in various OSS projects and then in turn have found that that can be translated to a service and support business for that software.

Figure 5–1 illustrates this hierarchy of OSS businesss models.

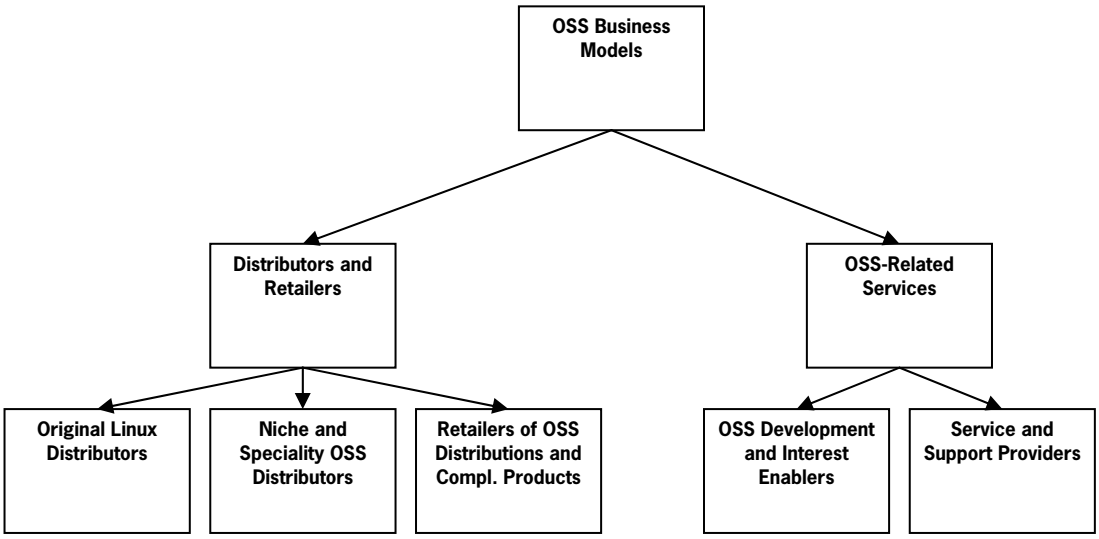

Figure 5–1: Hierarchy of OSS business models [FLOS 2002a, FLOS 2002b, FLOS 2002c].

5.1 Developing Country Business Models

OSS business models for developing countries can be structured in a similar manner. We examine the viable business models for developing countries for business that focus on the domestic market as well as those that address the export market.

5.1.1 Domestic Market Business Models

Consider the distributors and retailers segment within the domestic market. Due to the oft prevailing mindset in developing countries that "foreign is better", it is unlikely that domestic brands will be able to compete and win against suitably localized global packaging brands. This is especially true for commodity products such as operating systems, office productivity software, and back-office software. In such a scenario, the likely successful business models are those that suitably localize and add value to global brands.

Niche and speciality distributors are likely to have even a tougher

business model in developing countries. However, localized retailers of OSS distributions and complementary products such as book publishers are likely to succeed in these markets.

5.1.2 Export Market Business Models

Business models related to OSS services that cater to the export market are likely to thrive in developing countries. OSS services can be provided for software development (i.e., for manufacturing) as well as for product support and maintenance (i.e., for product services) by developing countries because of prevailing wage rates which work in their advantage.

For such models to succeed of course the IT infrastructure and skill development within the developing country must be high. In such an environment the country is well suited to exploit outsourced and off-shored manufacturing and services of OSS products. It is likely that the off-shore model, rather than the outsourcing model, will be the first approach in these businesses, as that allows businesses in developed countries to retain a degree of control and for the developing country to yet have a viable business model.

6 Open Source Strategy for Developing Countries

Our hypotheses on an OSS strategy for developing countries, illustrated in Figure 6–1, is based on a review and assessment of the phenomenon itself, its significance and sustainability as reflected in its impact on software markets and business models adopted by private firms in the industry, and opportunities and risks faced by developing countries in the adoption and use of IT. The conceptualization of OSS strategy illustrated in the aforementioned figure, attempts to present a comprehensive framework applicable to developing countries with appropriate resource endowments in the economy and IT industry. We have also referred extensively to the paper titled "Open Source in Developing Economies" by Steven Weber [Webe 2003], in formulating our own framework of open source driven IT strategy.

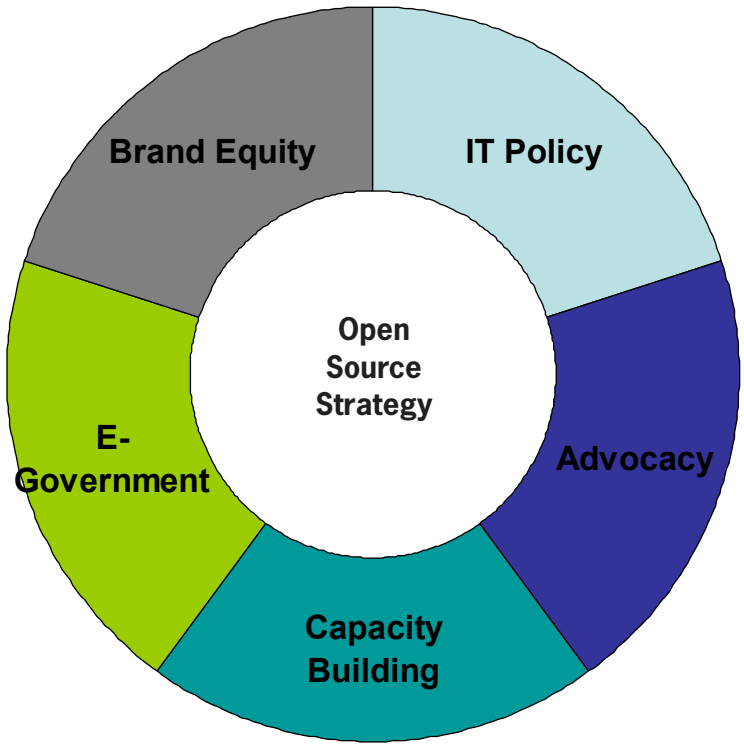

Figure 6–1: Aspects of an open source strategy for developing countries.

6.1 Opportunities and Threats for Developing Countries

In a networked world, the opportunity cost and risk for a developing country lacking sophisticated IT capabilities and means of effective interaction with the global economy could be substantial, with growth and development being seriously affected. Thus, decisions governments make relating to IT strategy and policies broadly, and in particular to procurement, the setting and adoption of standards, investment in technology, and training and skill development can have grave consequences for the future well being of their peoples. Many governments around the world have begun to consider the use of open source software as a key part of their strategic thrust in information technology, requiring that its use be considered when it provides a feasible alternative to proprietary software. Developing countries in particular, with the resource constraints they face, view OSS as a means of reducing the cost of IT investment and increasing its productivity. The imperative to adopt OSS in these countries particularly in the public sector is also motivated by a desire for independence, a drive for security and autonomy and a means to address intellectual property rights enforcement [Webe 2003]. A very compelling case for empowering the IT industry of a developing country through OS development is made by Weber in the above mentioned paper, presented in Figure 6.2.

Countries around the world have been keen to minimize their reliance on single suppliers, who may not be focused on the country's interest and to avoid opportunism by suppliers the country has locked itself through proprietary software purchases. The use of open source software "means that support and maintenance can be freely contracted out to a range of suppliers competing on quality and low cost for installation, enabling, support, and maintenance. Maintenance is furthermore replicable without incurring large costs, since modifications to the source code are also free".

> A regime built around the free diffusion of tools has an interesting characteristic: the degree to which a software tool can be utilized and expanded becomes limited only by the knowledge, learning, and innovative energy of the potential users; not by exclusionary property rights, prices, or the power of countries and corporations.
>
> Many emerging economies have a surplus of inexpensive technical manpower. Combining this with free software tools creates the possibility of an interesting kind of comparative advantage that will matter in local markets and in some cases might become important on global markets as well. This is a significant development opportunity because of the critical role that technology users play in determining new products and the overall trajectory of technology evolution. Leading-edge users and the innovative applications that they develop for their own practical purposes have been the primary drivers or at least the shapers of technological change in computing, largely because users are the creators of meaningful demand for better, faster, and cheaper machines and software. Recognize that each of these 'lead-use' applications came from the developed world, and principally from the United States. But emerging markets have their own autonomous development logics, and this will be just as true for economies transiting the information revolution as it was for the industrial revolution. The promise here is that software innovations can and should come increasingly from users in developing countries.

Emerging markets are not implicitly stuck relying on commoditized, hand-me-down innovation from the developed world. They can have their own lead users who pull technology development towards applications that fit specifically the indigenous needs and demands of emerging markets. Indeed, because information technology often has great plasticity and is more easily customized than were many industrial era technologies, the opportunity for autonomous lead users in emerging markets to deeply influence the direction of technology development is considerable.

Open source software helps to tap this potential. The empowerment that comes with free access to source code is not then simply a one-shot price advantage, it is a necessary economic prerequisite of evolving demand. In many cases the 'killer apps' for developing economies (more modestly, the applications that find widespread acceptance and drive technology and infrastructure deployment forward) will almost certainly come from within those economies. The open source process has the potential to empower developing country end users to customize applications for the very particular needs that often arise in different settings, and allows, through use, the natural evolution of information technologies and systems within unique and specific contexts.

Figure 6–2: Excerpt from [Webe 2003]

"Public data security is a foremost government concern particularly in the wake of worldwide computer virus in recent years and growing fears of cyber terrorism and cyber crimes. At a minimum, introducing diversity in to the base of functioning software codes reduces the possibility of catastrophic failures due to viruses that attack a monoculture of codes", particularly in the military a primary concern regarding software is security and autonomy. "Even outside the military, heightened concerns about the security and system reliability have prompted other government agencies to consider open source platforms" [Webe 2003]. With regard to the question of security and autonomy, it has been said "no national government, if it had alternatives, would have chosen during the 20th century to accept dependence for steel or petroleum on single or small number of suppliers based in another nation. Thus, increased emphasis on accountability or transparency in public sector governance has focused on the need for open public data formats." [Webe 2003]

Recent actions by various countries illustrate the growing concern about security and foreign control via proprietary software. Figure 6–3 provides an extract of a recent announcement by Japan, China and Korea on plans to collaborate on building open source business models, standardizing software and training software engineers.

Linux Revolution: Asian Countries Push Open Source

Analysts are usually skeptical about partnership announcements such as these, but the Japan-China-Korea initiative will be guided by coordinating bodies with considerable clout: the Japanese IT Services Industry Association, the Chinese Software Industry Association and the Federation of Korean Information Industries.

Asia is emerging as a key battleground for the open-source movement. The Japan-China-Korea (JCK) partnership, announced last month in Osaka, is the latest in a string of initiatives to promote Linux. Two weeks earlier,

Singapore hosted the second annual Asia Open Source Symposium, where 20 Asian countries discussed closer collaboration in standardization, localization and interoperability of Linux software.

Announcements of major Linux initiatives have become a daily occurrence in Asia this year. Governments in the region see open-source software as a fast track to the development of a domestic software industry, one that is not beholden to foreign licenses.

Figure 6–3: Extract from Japan-China-Korea announcement [JCK 2003]

Figure 6–4 provides an extract of comments by Indian President Abdul Kalam expressing similar sentiments about the role of open source in developing countries.

India leader advocates open source

The president of India added to a growing foreign-relations headache for Microsoft with a speech in which he advocated broader adoption of open-source software.

In a speech during dedication ceremonies Wednesday for the country's new International Institute of Information Technology in the university city of Pune, President A.P.J. Abdul Kalam recounted a conversation earlier this year with Microsoft Chairman Bill Gates.

"We were discussing the future challenges in information technology, including the issues related to software security," Kalam said, according to a transcript of the speech. "I made a point that we look for open-source codes so that we can easily introduce the users built security algorithms. Our discussions became difficult, since our views were different."

Microsoft has become an increasingly harsh critic of the open-source model, in which the underlying code for software is freely shared for users to modify and distribute, saying the approach is risky and undermines innovation. The open-source Linux operating system has become a growing threat to the dominance of Microsoft's Windows.

Kalam said open-source software offers developing nations such as India the best opportunity to modernize.

Figure 6–4: Extract from Indian President Kalam's comments [Indi 2003]

Appendix D illustrates the position taken by Peruvian Congressmen Edgar Villanueva when he introduced a bill for the use of open source software in all government systems. The key issues (summarized in Figure 6–5) relate to the use of standard and open formats and guarantee free access to public information on the one hand and on the other the guarantee of the permanence of public data, the usability and maintenance of software independent of suppliers with monopolistic positions in the market place or of monopoly conditions imposed by them. Thus, open source software is advocated on the basis of providing greater flexibility and allowing a more autonomous input into software development.

Summary of main points of E. Villanueva's letter to Microsoft Peru

Bill Number 1609 (The Use of Free Software in Public Administration), introduced by Congressman Edgar Villanueva, is intended to require the use of FOSS in all government systems, when there is a choice between FOSS and proprietary software.

Congressman Villanueva's letter to Microsoft Peru (8 April 2002) expressed the following principles:

– To guarantee free access by citizens to public information, it is indispensable that the encoding of data not be tied to a single provider.
– The use of standard and open formats guarantees free access.
– To guarantee the permanence of public data, the usability and maintenance of the software should not depend on the goodwill of suppliers or on monopoly conditions imposed by them.
– To guarantee national security, the State must be able to rely on systems without elements controlled from a distance. Systems with open-source code allow the State and citizens to inspect the code themselves and check for back doors and spyware.

In response to the concerns raised by Microsoft Peru, Congressman Villanueva argues the following:

– The bill does not meddle in private-sector transactions and protects equality under the law (i.e. nobody is denied the right to offer these goods to the State). There is no discrimination, since the bill specifies only how the goods are to be provided, not who has to provide them. Proprietary software companies are free to offer FOSS solutions to the Government in a competitive tender.
– The bill stimulates competition, since it tends to generate a supply of software with better conditions of usability, and to enhance existing work, in a process of continuous improvement.
– Proprietary software creates mainly "technical tasks of little aggregate value" in countries like Peru; free and open software creates more technically qualified employment, stimulates the market, and increases the shared fund of knowledge, opening up service alternatives to the benefit of producers, service organizations and consumers.
– As for security, bugs in free software are rarer and are fixed much more quickly than in proprietary software.
– Free software in no way implies ignorance of intellectual property laws; the great majority of free software is covered by copyright.
– The bill is not mistaken regarding the costs of free software: while the possibility for savings in payments for proprietary software licenses is mentioned, the foundations of the bill clearly refer to the fundamental guarantees to be preserved (free access, permanence and security) and the stimulus to local technological development.
– The use of free software contributes significantly to reducing life-cycle costs: support and maintenance can be freely contracted out to a range of suppliers competing on quality and cost for installation, enabling, support and maintenance; maintenance carried out is easily replicable without incurring large costs, since modifications can be included in the common fund of knowledge; and the huge costs created by non-functioning software are reduced by using more stable software, which is one of the virtues of free software.

– Migration to new systems is in fact cheaper when FOSS is used, since all data are stored in an open format.
– Interoperability is guaranteed as much by standard formats (as required by the bill) as by the possibility of creating interoperable software given the availability of the source code.

Figure 6–5: Summary of Congressman Villaneuva's letter to Microsoft (see Appendix D)

The movement towards strict compliance with standard intellectual property rules has gathered momentum even in the developing countries. Whilst, OSS undoubtedly provides an alternative to dealing with proprietary IP regimes in contrast with software piracy, and is thus a means of reducing IT investment costs within a framework of compliance. It is clearly of paramount importance that the strategic opportunities which it provides are comprehended by such countries.

6.2 Strategy Map for Developing Countries

In the final analysis, if open source is to make sense for developing countries, it must constitute a key part or prong of the country's IT strategy which creates value or wealth in the economy. At least three key areas can be identified which drive value creation in the economy through IT.

– Enhanced or new business opportunities in the IT sector for private firms
– Reduction of IT cost in the economy for both Government and private firms.
– Improvement in the effectiveness and efficiency of Government (and Governance).

The strategy map for developing countries illustrated in the Figure 6–3 demonstrates the manner in which open source strategic initiatives serve the value creating objectives, which have been enumerated above.

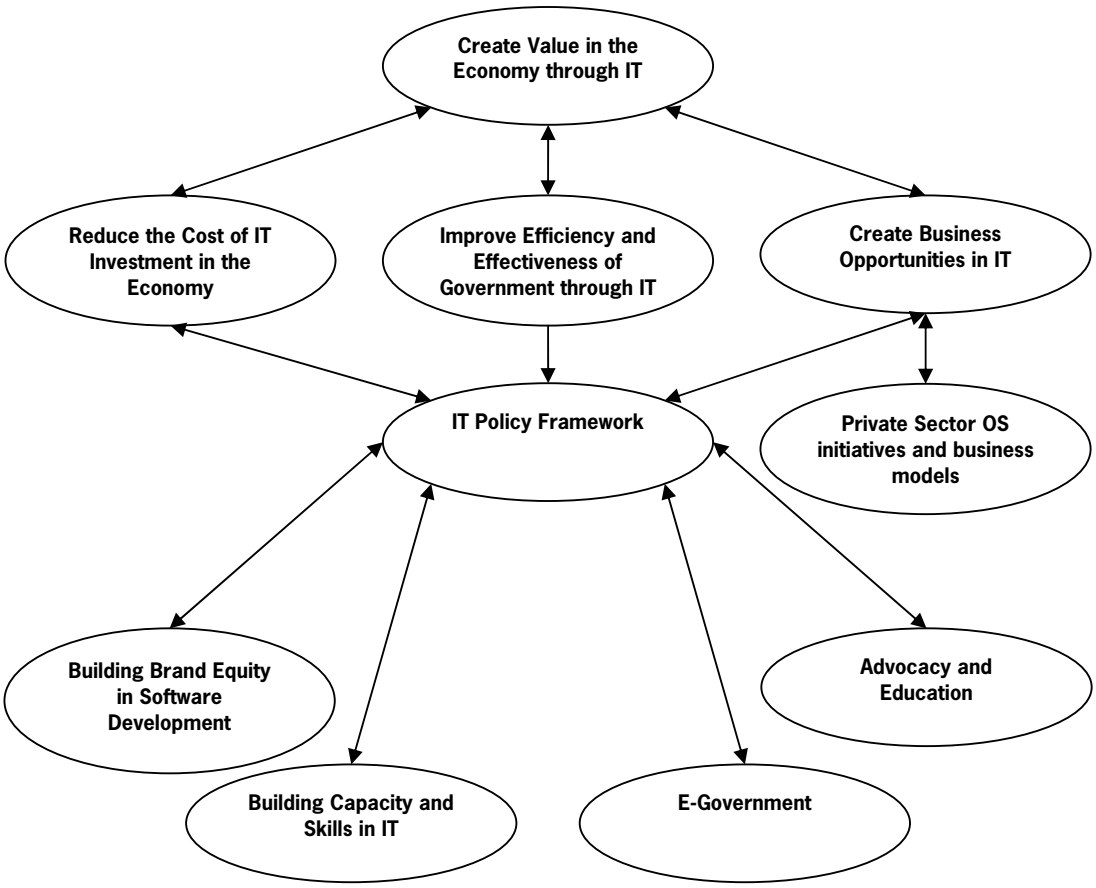

Figure 6–3: Strategy Map for Developing Countries

The creation of value through business opportunities would occur with capacity building in the private sector using OS i.e. the development of IT skills of developers and others and the adoption of open source based Business Models by private firms. OS provides a unique opportunity to build software development skills in areas of the software stack, which do not form part of the bread and butter off-shoring/outsourcing work, which developing countries are typically exposed to. Basic skill development can be more easily undertaken through the inclusion of OS in the curricula of educational institutions and the universities. More significantly the Lanka Software Foundation model (see Appendix F) is an illustration of pragmatic strategy in building the brand equity of individual developers and the countries in which they reside through participation in OS development projects (E.g. committers to Apache Foundation). Such initiatives could also trigger the actual adoption of open source business models by developing country's private firms. Even in the absence of such OS based business initiatives, the enhancement of skills and reputation of developing country software developers build brand value for the country's IT industry, thus facilitating the export of software and services.

Reducing cost of investment in an area, which should be considered a key enabler of socio economic development, would flow directly through to value creation in the economy. The extent of savings through the

adoption of open source software versus proprietary products (E.g. Microsoft) will always be argued. Figure 6–4 describes different contexts in which savings in IT cost in the public sector have been contemplated through a strategy of procuring OSS. However, it cannot be over emphasized that the enabling environment is critical to the success of such efforts. The policy framework, open source advocacy, the availability of IT skills in the use and localization of OSS, the presence of a local private sector adopting OSS based business models are some of the key enablers or drivers of an environment in which OSS becomes a real option in the IT investment equation.

Reducing cost of IT Investment

South African Government Council responsible for formulating the government's open source policy expressed foreign currency savings as an explicit rational for considering OSS deployment

The Taiwanese Government estimates that it could save nearly $300 Mn. In royalty payments through a strategic open source project that encourages research and development in office software and the opening of source code for government agencies and private enterprises as well.

A new bill introduced in Oregon in April 2003, which could be the first bill in the United States to encourage the use of OSS by a state government, says that open source options can "significantly reduce the State's cost of obtaining and maintaining software",

In Europe, where local, state, and central governments spent $ 7.8 Billion on software in 2000, numerous bills and resolutions have been introduced requiring the use or at least equal consideration of open source software solutions. A 2002 report by the European Commission encouraged EU Governments to share Open source software resources as a way to cut down on storing E-Government costs.

Figure 6–4: Reducing Cost of IT Investment

More effective and efficient government, whatever the extent of its dominance in the national economy, would without doubt create value in the economy. E-Government has demonstrated that it can make the difference in the way government provides goods and services to its citizens in an efficient and responsive way. In the execution of E-Government strategies, open standards primarily and open source software in that order would be critical to an effective implementation strategy. Thus, outside of any reduction that may arise in IT investment cost through the adoption of OSS, the "open model" becomes the very bedrock on which these initiatives are executed. On the one hand, it is necessary that a rigorous approach to E-Government software architecture is adopted in the context of competing technologies and current developments in the area; on the other, it is imperative that security and independence are considered in their proper perspective with OSS viewed not merely as a product choice but in a more fundamental and strategic context of the production, flow and control of information in the economy and the process by which the rules that governs such flows are managed. The arguments made by Congressman Villaneuva (see Appendix B) expand more on these reasons.

6.3 Strengths and Weaknesses of IT Infrastructure and skills for Developing Countries

In order for open source activities to proceed down a strategic path, certain IT infrastructural and skills conditions need to be met. These requirements include the following:

- *Intellectual property (IP) law framework and enforcement.* A common symptom in developing countries is the lack of IP laws and/or the failure to enforce IP laws. The result of course has been rampant pirating of proprietary software, thereby creating a false reality of wide availability of proprietary products at no cost. For example, it is common for a new computer to be pre-installed with pirated copies of whatever proprietary software the customer wants. In addition to being illegal, such piracy devalues the economic benefits of OSS products by falsely reducing the price of proprietary software. The economic benefit of OSS products will not be felt until IP is properly protected.
- *Low cost, widely available Internet access.* A critical factor for open source participation is the ability to become part of the Internet. Open source development occurs primarily via email communication and shared repositories published on the Internet. If high-bandwidth Internet access is not widely available in universities, companies as well as individual homes, the ability to participate in OSS projects is severely limited.
- *Educational infrastructure.* Even for base adoption of OSS products, the IT education infrastructure must be widely disseminated. A network of training/educational institutions that teach basic computing skills is essential to promulgate the dissemination of OSS products and solutions. Higher level institutions that teach software development technique and technologies are also critical. A culture of learning and further development of the workforce will help with faster and wider adoption.
- *Freedom of information.* Access to the Internet brings with it free access to information. If the political climate of the country does not permit such access, then open source cannot succeed in that country.
- *English-skilled developers.* English undoubtedly remains the *lingua franca* of computing. Once the applications are developed, they can of course be localized to support any language, but the development itself is very much English based. Even beyond traditional development, because of the geographically and temporally distributed nature of OSS development, good communication skills are a critical tool of the successful OSS developer.
- *Skilled or trainable developer pool.* Eventually success in OSS development comes from having skilled developers. If the country does not have a skilled or trainable developer pool, then it is not feasible to participate in the OSS development activities. It is of course still viable to execute other aspects of the OSS strategy.

For a country to be considered strong in IT infrastructure and skills, it is critical to have all these criteria satisfied to some extent. In particular, a strong IP law framework, good Internet access, excellent education infrastructure and English-speaking, skilled developers are absolute requirements to be able to fully execute the OSS strategies discussed in this report. If a country is weak in some or all of these, then it can execute some of the aspects of the strategy and aspire to move further down the strategy map by improving on the weak areas.

7 Strategy Implementation

The creation of actual value in the economy through an OSS driven IT strategy would demand the execution of a well conceived and integrated set of actions or strategies which would trigger business opportunities, reduce cost of IT investment and/or improve the efficiency, effectiveness and responsiveness of Government. This section deals with the details of action programs underlying the key strategies, which may be pursued in order to achieve the afore-mentioned goals or objectives. It pre-supposes a formal approach to strategy and policy making with government being pro-active in a process where collaboration between the government, the private sector, universities and other educational institutions is a pre condition to success.

Stephen Weber in his paper on "Open Source Software in developing economies" [Webe 2003] identifies three sets of policy implementation options to be considered by governments in adopting OSS.

- Formal approaches (such as legislation or government strategic plan) versus more informal, flexible approaches to letting OSS use evolve.
- Level of involvement: Sub national, national and/or regional collaboration
- Mode of development: Public sector adoption of OSS versus private sector adoption and collaboration among various users.

None of these represent mutually exclusive options but rather represent a spectrum along which governments can choose to array specific policies or a more general approach towards OSS use. We have set out below individual strategy initiatives elaborated in the previous section and illustrated in Figure 6–3, with the details of action programs relevant to each. It must be emphasized that such action programs are indicative only and are not by any means comprehensive in nature.

7.1 IT Policy

It is imperative that the IT policy framework for the country is formulated and executed in a manner, which permits a level playing field in software procurement. OSS in developing countries is unlikely to emerge as a real alternative to proprietary software if it is given step-motherly treatment in the national IT policy, as it will be eclipsed by the enormous influence of the major proprietary software players.

A government software policy which strikes the right balance and is non-discriminatory, is illustrated in Appendix G. Where there is a framework of an IT strategy and policy at national level with the presence of a government agency which could initiate the formulation of a government software policy, it would be easy to mobilize broad based participation from the IT industry and policy makers for this purpose. On the other hand, the government software policy could be formulated and implemented in the absence of a benign presence of an IT policy at national level. In whatever context, open source software should not be recognized as a mere product, which offers an economic choice. Its more fundamental and strategic character has already been addressed in the previous section.

7.1.1 Action Items

Establish government software procurement policy. In many developing countries (DCs) and least developed countries (LDCs), government is the largest business in the country. Establishing a software procurement policy which specifically recognizes the role of OSS is a critical step in ensuring that the largest consumer of software products gives due consideration to OSS options. For certain segments of the government, where security and independence is critical (e.g., in defense) OSS may indeed be the only strategically viable option. It is important for the policy to highlight the non-investment cost related benefits of OSS, including capacity building, independence and security.

*Encourage good private sector software procurement policies***.** Use government software procurement policies to demonstrate the business value of OSS to other organizations. These values include direct economic benefits, capacity building and competitive advantages. To demonstrate the value of OSS it will be necessary to develop appropriate metrics as well as to organize studies.

Keep the Internet free of tariffs, licensing and control. Critical to the success of open standards and open source activities is widespread access to the Internet. Over-regulating the Internet by introducing tariffs, licenses and other forms of control may reap some short-term benefits, but until the "Internet economy" has matured sufficiently, that will undoubtedly stifle growth and value creation.

Summary of strategic steps highlighted by South Africa's government council on open-source policy

South Africa's Government Information Technology Officer's council's FOSS strategy includes steps to consolidate and expand the capacity needed to implement and support FOSS solutions, including:

- Provision of information to key decision makers (bearing in mind the need to demonstrate convincingly the security measures and business principles of FOSS)
- Generation of expert advice on the suitability of FOSS solutions
- Trouble-shooting for newly implemented FOSS solutions
- Software development assistance
- Training for FOSS developers and users (concentrated in existing learning institutions)

– Development of a research programme to enable optimal understanding of and decision making regarding FOSS (built on the networking nature of the FOSS development model)
– Creation of FOSS support structures (some institutional development will be necessary)

Figure 7–1: Summary strategic steps of South Africa's OSS policy [Sout 2003]

Statement of the Free and Open Source Software Foundation for Africa (FOSSFA)

– *Preamble* The potential of open source will improve productivity and quality of life in developing countries. The process of transformation into information societies requires the full participation of all member states.
– *Vision* Our vision is to promote sustainable, viable and cost-effective software products for Africa through education and local capacity building.
– *Principles* Africa should investigate how to leverage the opportunities presented by the emergence of open-source software in the context of limited financial resources and expertise.
– *Specifics* Africa can bridge the "digital divide" by adopting open source, thus narrowing the effect of techno-colonialism.
– *Plan of action* It is envisaged FOSSFA, in partnership with Governments, intergovernmental organizations, civil societies and other stakeholders, will spearhead initiatives that build skills through education and empowerment of women and youth.

Lobby all stakeholders to adopt open source as the platform to engineer solutions that meet the needs of the people.

Strategies FOSSFA will:

iii. Create an awareness of free software and open source in Africa.
iii. Build capacity in free software and open source.
iii. Develop a knowledge warehouse of expertise in Africa.
iv. Develop the African Open Source Portal.

We intend to achieve these by:

iii. Lobbying key organs such as Africa Union, UNECA, UNDP, Agence la Francophonie and NEPAD among others to support open-source development in Africa.
iii. Leveraging various free and open-source capacities and resources in Africa.
iii. Lobbying donor governments and other institutions to tie ICT funding to free and open-source software.
iv. Lobbying African governments to adopt free and open-source software.
v. Promoting open-source capacity and skill development in Africa through education with emphasis on women and youth.

Figure 7–2: Statement of the Free and Open Source Software Foundation for Africa on Open Source Software [FOSS 2003]

7.2 Advocacy and Education

In order to provide substance to the policy framework, there must be active initiatives in advocacy and education relating to OSS. In a developing country market place which is dominated by proprietary software, licensed or unlicensed as the case may be, OSS cannot create the marketing dollars required to influence product choices. Countervailing forces in the form of advocacy groups, educational courses and material targeted to the user community are absolute pre-requisites in enabling the IT policy to generate real world outcomes. Linux user groups illustrate a community, which can be proactive in advocacy if the right initiatives are taken by the relevant government agencies. Training and courses in the use of OSS should be organized at universities, private IT training institutions, public sector training institutions and professional organizations. This should constitute the platform to disseminate the power of OSS through courses and material, both basic and advanced, including domain specific applications. In order to provide a single gateway to the information resource relating to this domain, a portal could also be created as a part of the advocacy and education program. A mechanism for implementation of such a portal could be government software procurement policy where coverage would be balanced in terms of both OSS and proprietary and other commercial software.

7.2.1 Action Items

*Establish OSS advocacy group***.** Given the marketing might of proprietary software vendors, it is important to create a counter-balance organization whose mission will be to advocate the use of OSS. The widely distributed network of Linux User Groups (LUGs) can be a source of resources for such a group, but of course it is important to focus on the entire spectrum of OS products and not just Linux.

Open Source Computer Driving License. The idea of a computer-driving license [CDL 2003] is now quite popular globally. Whilst it is a very important development, this is fundamentally flawed as it is taught typically using just a few proprietary products. An effective way to advance the cause of open source office-productivity software would be to augment the CDL program with OSS alternatives. In addition to demonstrating the viability of OSS in daily office activity scenarios, this would be a mechanism to provide certification for open source product skills.

Establish an OSS Portal. One of the difficulties with adopting open source software is that once one goes beyond brand-name open source products such as Linux, Apache and OpenOffice, there is a plethora of available software for any given task and little information about which of them are really suitable for production use. The objective of the OSS Portal would be to provide a one-stop location where one can find available open source solutions for specific needs, along with appropriate rating/evaluation information to indicate their quality and competitiveness with respect to available commercial products.

Establish customized training programs through professional organizations. Combining the above two activities, these specialized training programs will show how various professional communities (lawyers, doctors, ac-

countants etc.) can adopt OSS for professional practice. This addresses a significant gap today between saying, "use OSS" and being able to put it into practice for non-IT professionals.

Introduce OSS to university curricula. Require usage of OSS in university IT programs at least during one year of the curriculum. This can provide the required awareness of and introduction to OSS to the next generation of IT professionals. From an academic perspective it provides a broad outlook to students rather than providing them with a narrow view of technological options. The same awareness to the wider community would be provided by offering the OSS-enabled computer driving license to non-IT students.

Introduce OSS-enabled CDL in K-12 curricula. As more and more elementary, middle and secondary schools become computer equipped, it is important to demonstrate to the younger generation the availability of OSS solutions. Many curricula already teach the CDL or similar material – offering the OSS-enabled CDL version will bring OSS awareness to the youngest computer users and thereby to their homes.

Improve OSS platform awareness. Software development is now moving towards being platform based – that is, organizations make software decisions not on a single product basis but rather on an entire family of products. Offering educational programs to emphasize the platform viability of OSS will help demonstrate its competitiveness.

Support the development of easier deployment and experimentation platforms, such as those offered by Knoppix [Knop 2003] to encourage wider participation in OSS. Such platforms allow users to "try out" OSS operating systems and fully configured applications without having to install an entirely new operating system on their platform.

Offer appropriately localized versions of OSS to demonstrate the viability of using OSS products for building people-oriented applications, which require national language support. One of the critical advantages of OSS is that it enables localization by communities that wish to do so. Making such localized "distributions" of OSS products will further the message that OSS has fundamental advantages over the proprietary software model.

7.3 Capacity Building in the Local Software Industry

A condition precedent to the success of an OS strategy in general and a Government software procurement policy in particular is the capacity of the local industry to actively use, localize and implement open source software. To ensure that this "critical mass" actually materializes, traditional capacity building measures in the form of targeted and structured training programs at universities, private sector IT training institutions and other training organizations, particularly in the public sector would have to be formulated and implemented. The localization of OS software and National Language Support (NLS) are specific areas which would have to be addressed in these initiatives.

Capacity building can be considered in two categories: (1) capacity building in using OSS, and (2) capacity building in adapting existing, and developing new, OSS products. The first category is covered in section

7.3.1 Action Items

Establish localization center. One of the biggest advantages of OSS is that it gives the opportunity for arbitrary localization and customization. With OSS a DC or LDC can establish its own program to localize software without waiting for proprietary software vendors to do so based purely on market economics. Since most DCs and LDCs have national languages other than English, exploiting national language support capabilities of OSS to create locally deployable software is feasible. The localization center would perform localization of certain products as well as be a focal point and support source for other localization efforts within the country, wherever it occurs. In addition to the direct benefit of delivering localized software, this will also have the indirect benefit of creating a group of developers who are familiar with the source codes of key OSS products, thus empowering them to move on to deeper adaptations/improvements of those products.

Exploit localization opportunities of other countries. Not every DC and LDC will be in a position to address their own naturalization skills – the reasons range from unavailability of a sufficiently skilled IT workforce to the poor English knowledge of IT workers (as software source code is always in English-based computer languages). In such scenarios, countries which have mastered how to localize OSS can export that skill to other DCs and LDCs, thereby creating further opportunity to expand the capacity of the local IT industry as well as a unique business opportunity.

Establish OSS Solutions Center. As software decisions are being made based on platforms instead of point products, it is important to develop patterns of solutions that can be built using OSS products [EBus 2003]. The purpose of this center would be to serve as a source of knowledge about OSS platform choices, primarily for public sector usage but also for the private sector. Once this has matured, it will be possible to export this skill to other DCs and LDCs as well.

Establish R&D Center. OSS does not typically push the envelope in technology development, but rather rises to the opportunity when open-standards are in play. Thus, as the computing industry moves to new domains, technologies and devices, significant opportunities for development and market success of OSS products are created. The purpose of the R&D center would be to develop focused efforts on specific evolving areas to establish leadership in those areas. The current work in Web services [WebS 2003] is an example where such opportunities exist.

7.4 Positioning the Local Software Industry in the Global Industry

The open source phenomenon, the market position that open source software enjoys and the associated business models that have emerged offer unprecedented opportunities for a developing country software industry to build capacity and position itself to exploit global opportunities. The Lanka Software Foundation (LSF) is an illustration of strategy formulation and execution in this regard (see Appendix F).

7.4.1 Action Items

Participation in the development of open source software as part of global projects such as Apache & Linux in high impact areas of the software stack. Many open source projects are led and developed by small groups of people. If the suitable skillset is available and/or can be developed, then significant brand value for the local industry can be achieved by becoming leading contributors of key projects.

It is important to recognize that in executing a strategy of this nature, there must be available to the local industry, key IT industry specialists with globally competitive software development skills, domain expertise and networking abilities. The gap between strategy formulation and execution in a context such as this could be considerable without the right ingredients in terms of leadership and human resources.

An alternative strategy is to catalalyze the formation of open source communities in the country, where committed individual developers, without necessarily the university – industry support that the aforesaid model entails, participate in global open source development projects. The lack of a structure and a driving force except individual initiative and focus would make this a largely "hit or miss" effort with much less potential to build brand value and reputation for the country in the global software industry.

*Develop collaborative models of joint development between developed and developing countries***.** While this approach requires government level intervention, the recent Japan-China-Korea announcement [JCK 2003] illustrates the potential of this approach and benefit to all parties.

7.5 E-Government

The adoption of open source software (and open standards) for e-government, where substantial claims are made about efficiency and effectiveness of Governance and the responsiveness of Government to its citizens, would have strategic and operational implications of great significance. The goals of security, independence etc. too would be secured more effectively in managing information flows in Government with the adoption of open source software (and standards). Thus, within the broad policy framework of software procurement for the public sector, specific strategies should be in place to deal with situations where the aforesaid considerations become paramount.

Open standards have been already differentiated from OS. The adoption of open standards in E-Government can only contribute to the very underpinnings of "good governance" and sound democracy. Although the linkage to economic development may be tenuous, it is accepted that sustained and sustainable development in the modern world can be more effectively secured through such means. The "open source model" in its generic form is something which governments can adapt and adopt in their more people centered activities.

In the execution of E-Government initiatives, a well-conceived architecture would be a critical driver of success, which developing countries could only ignore at their own peril. The development of open standards (and software) and component based architectures would fall within the ambit of "open source" strategies for developing countries. Its definition is a strategic initiative of importance, to be pursued by an

agency in a government charged with the responsibility for IT policy in the country (e.g., in Sri Lanka, Information and Communication Technology Agency [ICTA 2003]). The actual execution and development can be contemplated in a commercial context. However, OS development projects to deal with specific components may be initiated, depending on the maturity of the OS development model in the country itself or even globally.

7.5.1 Action Items

*Establish open standards based component architecture for e-government***.** A critical success factor for e-government in general is to have a viable open standards based component architecture [FEA 2003]. Having such an open architecture will allow OSS and proprietary software to compete equally, rather than if it were a closed architecture where one or two vendors would lock in every aspect of the e-government architecture.

Establish policy framework to give OSS a fair chance. While overlapping with IT policy, this is a critical need for e-government activities to succeed as well.

8 Role Of Donor Agencies

We have demonstrated that developing countries (and LDCs) could create value in their economies through the adoption of OSS as a key part of their IT strategy and policies. The treatment of strategy for developing countries necessitated the definition of a strategy map, which provides the framework for strategy formulation in this domain, further elaborated in strategy implementation, where a set of inter-dependent and concerted actions were described for the pursuit of goals and objectives under each strategic initiative. Thus donor agencies could hardly ignore the compelling reasons for interventions necessary to assist developing countries to realize their goals and strategies in IT, whether they be implicit or explicit. As a key driver of economic development in the modern world, the importance of medicines and vaccines not withstanding, IT can hardly be relegated to the background, unless the world is prepared to irretrievably destine the developing countries to the dustbin of history.

On a more pragmatic note the strategy map and the strategy actions provide the means by which donor agencies could target their assistance and formulate and monitor their own programs. Their role would be primarily catalytical and the situational context of the DCs would determine how significant that role would be and thus the extent of assistance that maybe required before critical mass in this area could be achieved. There would be little purpose if such a threshold cannot be targeted or achieved as an outcome of such assistance. A simple representation of IT policy versus IT infrastructure and skills in developing countries in the form of a matrix provides a snapshot of strategy initiatives, which donor agencies could respond to.

		IT Policy	
		Not Stated	**Explicit**
IT Infrastructure and Skills	**Weak**	IT Policy, Advocacy & Education, Capacity Building	Advocacy & Education, Capacity Building
	Strong	IT Policy, Advocacy & Education, E-Government, Positioning of Industry	Positioning of Industry, E-Government, Advocacy & Education

Table 8–1: IT Policy vs. IT Infrastructure & Skills matrix.

The thrust of donor agency assistance could be addressed on the basis of the IT policy – infrastructure & skills matrix at Table 8–1, the developing country strategy map illustrated at Figure 6–3, and the elaboration of strategic initiatives and actions under strategy implementation (Section 7).

- *IT Policy Not Stated & IT Infrastructure and Skills Weak.* Strategic initiatives in IT policy, advocacy and capacity building must take precedence.
- *IT Policy Not Stated & IT Infrastructure and Skills Strong.* Strategic thrust should be in positioning of the local IT industry in the global industry and with initiatives in IT policy, advocacy and e-government.
- *IT Policy Explicit & IT Infrastructure and Skills Weak.* Capacity building must take precedence in strategy with advocacy & education playing a critical role.
- *IT Policy Explicit & IT Infrastructure and Skills Strong.* Positioning the local IT industry in the global industry and e-government strategy initiatives could be vigorously and rigorously unleashed with expectation of considerable short term benefits and outcomes; advocacy and education too could be an important mobilizer of resources and action.

Thus the abovementioned framework would assist donor agencies to respond rigorously to OSS initiatives from both government agencies and the private sector. A proposal made on this basis will be evaluated in terms of the broader IT policy context and the specific goals and objectives, which are being pursued as part of the specific OSS strategy initiative(s) being targeted. We hasten to add that the strategy map should

not be considered static by any means. The dialogue between donor agencies, government agencies, universities and the private sector must ensure that the very dynamism of strategy is reflected in the framework applied to OSS strategy for developing countries.

For donor agencies in particular, lurking somewhere in their collective conscience must be the question of what OSS does to the competitive market place for software production and services. The sections on market structure and business models indicate that the OSS phenomenon has actually made the marketplace more competitive providing greater choice to users, whether they be government, private sector or the public, in an industry where proprietary software products and large industry players with substantial market power control large parts of the territory. The very phenomenon of OSS has not been driven by government intervention. In fact it demonstrates the power and significance of individual initiatives in a situational context and domain where such initiatives could translate into significant economic impact. Both motivational and organizational model of OSS can hardly be challenged in their basic fundamentals. It is significant that even the motivational model is weighted in favor of economic drivers rather than altruistic ones or self-actualization. What is important is the emergence of economic imperatives in a context where individual skills and commitment remain at the heart of the development of computer software and the expert power that underlies this, has had the circumstances and the means to manifest itself. If on this basis OSS has a sustainable position in the industry and the market place, the case for donor agency support becomes extremely strong rather than indifferent. In our view, any concept of market distortion cannot and should not cloud this phenomenon. On the contrary, it is a demonstration of the very market place and competitive dynamics that donor agencies would desire to espouse.

8.1 Case Study: Sri Lanka

We illustrate the broad approach that may be taken by donor agencies in initiating or responding to open source based IT strategies and programs by examining the case of Sri Lanka.

Sri Lanka could be positioned in the aforesaid policy-infrastructure and skills matrix in the lower right hand quadrant "explicit IT policy and strong IT infrastructure and skills" towards the center of the matrix itself. Thus, a comprehensive strategic approach with emphasis on positioning Sri Lanka's IT industry and e-government strategies as key components with the other components kicking in for capacity building could be a valid first iteration. Starting with IT policy, Sri Lanka has developed an ICT road map and established a government agency to execute it. However a government software procurement policy, which recognizes the role of OSS is conspicious. Thus private sector software procurement policies do not encompass the OSS option to a significant extent. The deregulation of the telecommunications industry and the wide spread availability of the Internet are areas where considerable progress has actually been made. In the area of advocacy, Sri Lanka has groups which advocate the use of OSS, such as Linux. Considerable work however would be necessary to mobilize the IT community to evangelize the use

of open source software widely. An OSS portal has not been established as yet whilst in the educational domain, training and educational programs dealing with OSS whether it be university curricula, in professional organizations or in schools is still lacking. The adoption of many of the proposed action programs in the educational arena could be a very important part of the OSS driven IT strategies for Sri Lanka.

The other aspect of capacity building in the local software industry dealing with adapting existing and developing new OSS products, establishing localization centers, OSS solution centers and a R&D center could be of great significance to Sri Lanka. A strategic initiative with university – private firm collaboration to position the local software industry in the global context has been taken already with the leadership provided by industry specialists who have the necessary wherewithal in terms of domain expertise and networking abilities. The Lanka Software Foundation [LSF 2003], which is a non-profit entity, mobilizes software developers to participate in key open source projects such as Apache. Although in one year it has demonstrated the success of such a strategy and the collaborative model between the universities, private firms in the industry and individual developers, it remains unfunded by government, international funding agencies and donor agencies.

The roll out of e-government too is of great significance to Sri Lanka where substantial progress would have to be made in relation to the efficiency, effectiveness and responsiveness of government itself.

Thus, it is manifestly clear that the strategic framework which has been developed in this study could be applied to assess the risks and opportunities that a particular developing country faces in pursuing an IT strategy, with particular emphasis on OSS and making assessments about the maturity of both IT policy and infrastructure which are key drivers of strategic initiatives in this area. We emphasize that this domain should be considered by donor agencies as a priority area for action. The lack of appreciation of the issues and the vision to formulate and execute strategic initiatives would only destine developing countries and LDCs to the dustbin of history. The donor agencies of developed countries pursuing their own self interests should not overlook the strategic opportunities which present themselves in relation to the OSS phenomenon to be exploited with well conceived, collaborative models between developed and developing countries. The impact on the map of the global industry of such initiatives could be significant.

8.2 Related Work

In order to facilitate the appreciation of the OS strategy framework that we have outlined above, the issues and conclusions of the report on "Open Source Software: Perspective for Development" prepared by the Dravis Group for infoDev and the World Bank group [Drav 2003] have been summarized below.

- *Interest in Open Source software is increasing globally.* Whilst the strength of the dynamics can be difficult to gauge, the trends are positive for OSS. Continuing *education and awareness building* are necessary to make decision makers familiar with the issues and the benefits.
- *OSS is about choice*. The potential for lower cost is the initial driver, a

longer term benefit maybe increased choice to consumers of ICT. The emergence of OSS creates more options to address ICT needs. Solutions can mix both proprietary and OSS components. New development models emphasize collaboration and community. *Market competition increases.*

- *Government leaders have a key role to play.* The software market is dominated by a few proprietary providers who spend over $ 6.5 billion annually to communicate their message. There is a *need to assist OSS initiatives in accessing market opportunities.*
- *There is opportunity for local capacity development.* OSS lends itself to providing an ICT environment based on local ownership and autonomy. Software can be adapted to address localization, while introducing more flexibility and independence to the software development process. *Business opportunities exist for various complimentary solutions and services.*
- *The debate: "Open" versus "proprietary" software.* The OSS debate is multi faceted. The discourse should be centered on *the merit of OSS versus proprietary solutions.*
- *OSS is but one part of an ICT strategy.* OSS should be evaluated and deployed using the same methodologies and disciplined care as any other ICT solution. It is interesting to note that government interest and activism is global and not aligned by geographic area, economic group or political philosophy. Examples of current governmental initiatives can be found in [Drav 2003].

9 Conclusion

In this study we have examined the open source phenomenon in general, and in particular as it relates to developing countries. We discussed the history of open source, how open source works, and why developers contribute to open source. After examining open source markets and business models, we presented a strategy framework and strategy map for developing countries to go down the open source path. The strategy framework was followed by a discussion of how the strategy map can be executed, with concrete action items under each area. Finally, we presented a classification of the current status of a developing country in relation to its IT policy & infrastructure and how a donor agency can assist towards exploiting open source to create value in the economy. A case study of Sri Lanka discussed a concrete scenario to illustrate how the donor action plan, strategy implementation and strategy map all tie together.

We set out on this study with a set of key issues to be addressed. We now briefly answer these questions:

- *Is open source significant and relevant in the software industry?* As discussed throughout the report, open source is today a major player in the software industry. Products such as Linux and Apache HTTPD have fundamentally changed the dynamics of the software industry. Many other products have the potential to have similar impact. In summary it is clear that as lower levels of the software stack become commoditized, open source products are taking over those levels. Thus, the answer to the question of whether open source is significant and relevant in the software industry today is a resounding "Yes!".
- *Is it sustainable?* We addressed the question of whether open source is sustainable in multiple ways. First we discuss an economic model for open source which not only shows it's sustainable, but indicates that open source is now a fundamental part of the software industry. Secondly we discuss how open source contributions and participation are not widely based on machismo and altruism as often believed, but indeed on sound economic principles; thus ensuring long term sustainability.

– *Can developing countries create value through OSS?* We developed a strategy framework for a comprehensive approach to OSS for developing countries. The framework covers not only the commonly done OSS activities such as advocacy and education, but also other aspects such as e-government, capacity building of the local IT industry and building brand equity for global positioning. By implementing the strategy map we present, developing countries can use open source to create significant value in the economy. Thus, indeed, developing countries can create significant value and opportunity through OSS.
– *If yes, how can donors support it?* We presented a concrete approach for how donor agencies can support the development of OSS in developing countries by supporting specific action items within each strategy area. We discussed a case study (Sri Lanka) to illustrate this in more detail.

This study demonstrates the clear value of OSS for developing countries. While there have been many recent announcements of developing countries adopting OSS products, many of those have been one-sided directions which have not examined the entire gamut of impact and potential of OSS. Our results indicate that by carefully exploiting OSS, it is possible for a developing country to establish a global position in the IT driven knowledge economies of the future.

In addition to the direct benefit OSS can provide to developing countries, we believe that significant potential for global IT industry leadership exists for *developed* countries as well by exploiting strategic partnerships between developed and developing countries in OSS efforts. We expect to consider such directions in the future.

10 References

[Apac 2003] Apache Jakarta Project, http://jakarta.apache.org/

[Bokh 1999] Shahid H. Bokhari and Rafeequr Rehman, "Linux and the Developing World", IEEE Software, January/February 1999.

[Bost 2002] Boston Consulting Group / OSDN Hacker Survey, http://www.osdn.com/bcg/

[Brod 2003] Carol Brod, "Free Software in Latin America", http://www.maailma.kaapeli.fi/lamerica.html

[CDL 2003] International Computer Driving License, http://www.icdlus.com

[Coet 2002] Nico Coetzee, "Free- and Open Source Software in Africa", http://www.maailma.kaapeli.fi/africa.html

[Drav 2003] Paul Dravis, "Open Source Software: Perspectives for Development", InfoDev Report, http://www.infodev.org/symp2003/publications/OpenSourceSoftware.pdf

[EBus 2003] IBM Patterns for E-Business, http://www-106.ibm.com/developerworks/patterns/install/index-lite.html

[FEA 2003] Federal Enterprise Architecture, http://feapmo.gov/fea.asp

[Fell 2002] Joseph Feller and Brian Fitzgerald, "Understanding Open Source Software Development", Addison-Wesley, 2002.

[FLOS 2002a] Free/Libre and Open Source Software (FLOSS): Survey and Study: Use of Open Source Software in Firms and Public Institutions Evidence from Germany, Sweden and UK, http://www.infonomics.nl/FLOSS/report/

[FLOS 2002b] Free/Libre and Open Source Software (FLOSS): Survey and Study: Firms' Open Source Activities: Motivations and Policy Implications, http://www.infonomics.nl/FLOSS/report/

[FLOS 2002c] Free/Libre and Open Source Software (FLOSS): Survey and Study: Basics of Open Source Software Markets and Business Models, http://www.infonomics.nl/FLOSS/report/

[FOSS 2003] Free and Open Source Foundation for Africa, http://www.fossfa.org/

[FSF 2003] Free Software Foundation, http://www.fsf.org/, accessed December 2003.

[Ghos 2003] Rishab Aiyer Ghosh, "License Fees and GDP Per Capita: The Case for Open Source in Developing Countries," http://firstmonday.org/issues/issue8_12/ghosh/, accessed January 2003.

[Hars 2001] Alexander Hars & Shaosong Ou, "Working for Free? – Motivations of Participating in Open Source Projects", Proc. 34th Hawaii International Conference on System Sciences, 2001.

[Heck 1999] Frank Hecker, "Setting up Shop: The Business of Open-Source Software", IEEE Software, January/February 1999.

[ICTA 2003] Information and Communication Technology Agency of Sri Lanka, http://www.icta.lk/

[Indi 2003] Indian Leader Advocates Open Source, http://news.com.com/2100-1016-1011255.html

[JCK 2003] Linux Revolution: Asian Countries Push Open Source, http://www.linuxinsider.com/perl/story/32421.html

[Knop 2003] Knoppix. http://www.knoppix.net/

[Koch 2003] Christopher Koch, "Your Open Source Plan", CIO Magazine, March 15, 2003 Issue. http://www.cio.com/archive/031503/opensource.html

[Lakh 2003] Karim R. Lakhani and Robert G. Wolf, "Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects", http://opensource.mit.edu/papers/lakhaniwolf.pdf

[Lanc 2001] David Lancashire, "The Fading Altruism of Open Source Development", First Monday, Vol. 6, Number 12 (Dec. 2001), http://firstmonday.org/issues/issue6_12/lancashire/index.html

[LSF 2003] Lanka Software Foundation, http://www.opensource.lk/

[Micr 2003] "Microsoft at the Power Point", http://www.economist.com/business/displayStory.cfm?story_id=2054746

[MITO 2003] MIT Free / Open Source Research Community, http://opensource.mit.edu/

[Mock 2000] Audris Mockus, Roy T. Fielding and James Herbsleb, "A Case Study of Open Source Software Development: The Apache Server", Proc. International Conference on Software Engineering, 2000.

[Netc 2003] Netcraft Web Server Survey, http://news.netcraft.com/archives/2003/12/02/december_2003_web_server_survey.html, December, 2003.

[Noro 2003] Frederick Noronha, "Free Software in Asia", http://www.maailma.kaapeli.fi/asia.html

[OSI 2003] The Open Source Initiative, http://www.opensource.org/

[OSIL 2003] OSI Certified Open Source Software, http://www.opensource.org/licenses/index.php

[Perk 1999] Greg Perkins, "Culture Clash and the Road to World Domination", IEEE Software, January/February 1999.

[Raja 2003] Niranjan Rajani, "Free as in Education: Significance of the Free/Libre and Open Source Software for Developing Countries", http://www.maailma.kaapeli.fi/FLOSSReport1.0.html

[Raym 1999] Interview with Eric S. Raymond, "Linux and Open-Source Success", IEEE Software, January/February 1999.

[Sout 2003] South African Government Open-Source Strategy, http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf

[UNCT 2003] United Nations Conference on Trade and Development, E-Commerce and Development Report, 2003, http://www.unctad.org/ecommerce/

[Varn 1999] Philip E. Varner, "The Economics of Open Source Software", http://www.cs.virginia.edu/~pev5b/writing/econ_oss/

[Webe 2003] Steven Weber, "Open Source Software in Developing Economies", http://www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf

[WebS 2003] The Web Services Industry Portal, http://www.webservices.org/

[Weer 2003] Sanjiva Weerawarana and Jivaka Weeratunga, "The role of open-source in e-Sri Lanka", Proc. 22nd National IT Conference, Colombo, Sri Lanka, July 2003. http://www.cssl.lk/conferences/The%20Role%20of%20Open-Source%20in%20e-Sri%20Lanka%20sanjeewa%20weerawarana%20letter%20size.doc

[Wils 1999] Greg Wilson, "Is the Open-Source Community Setting a Bad Example?", IEEE Software, January/February 1999.

Appendix A. OSS Licenses

This material is extracted from [Drav 2003].

A.1 The Legal Landscape is evolving

A large part of the debate around OSS, its implications and its significance for developing countries, focuses on Intellectual Property Rights which encompass software licenses, patents and copyrights. The following excerpt from UNESCO's "World Information Report 1997/98" provides insights on the forces driving the discussion.

> "The creation and ownership of knowledge products are of increasing importance because of the centrality of information and knowledge to post-industrial economies. The concept of copyright, originally intended to protect authors and publishers of books, has broadened to include other knowledge products such as computer programs and films. Copyright has emerged as one of the most important means of regulating the international flow of ideas and knowledge-based products, and will be a central instrument for the knowledge industries of the twenty-first century. Those who control copyright have a significant advantage in the emerging, knowledge-based global economy. The fact is that copyright ownership is largely in the hands of the major industrialized nations and of the major multimedia corporations placing low per capita income countries as well as smaller economies at a significant disadvantage."

Open source software is differentiated from proprietary software because the programming code used is available for inspection, modification, reuse and distribution by others. While OSS can be free of charge, it can be purchased for a fee as well. The concept of "free", in this context, emphasizes what can be done with the source code rather than its acquisition cost.

A.1.1 Many Open Source Licenses Exist

While open source is associated with Linux and the GNU GPL licensing agreement, there are many types of open source software programs available and the Open Source Initiative has certified over 40 different open source licensing agreements. The following are among the most prevalent licenses in use.

- GNU General Public License (GPL) license was written by Richard Stallman in 1989,for the purpose of distributing programs released as part of the GNU project. It is the most widely-used of the open source software licenses and is considered to be the "purest" by requiring that all source code is free and available and that changes must be shared with the community of developers. Linux is among the software available under this license.
- BSD (Berkeley Software Distribution) License, adopted from Berkeley Unix, requires copyright notification, and permits the source to be used in any manner, as long as notification is provided. The FreeBSD operating system and Postgres SQL database uses this license, and Apple Computer's OS X is a derivative work based on BSD licensed technology.
- Mozilla Public License (MPL) defines terms and provides for code that can remain proprietary under very specific terms and dual licensing when necessary under the provisions of the GPL. The MPL is likely to be among the more flexible models for open source organizations. The Mozilla, Firebird and Thunderbird Web browsers are available under this license.

Each of these licenses state that source code is open and users are free to use, copy, duplicate, distribute, modify them. The GNU GPL includes the added provision that modified versions of the software, if distributed outside an organization, will come under the conditions of the original work's license.

A.1.2 Limited Open Source Case Law Exists

While much of the success of OSS has been associated with the development of the Linux community and the GPL, one must note that only once has a GPL licensed product has been tried in a court of law, in the MySQL vs. NuSphere lawsuit. In this action, the legal issue was not directly related to the GPL and the matter was settled out of court. With virtually no case law challenging OSS to date, more specifically the GPL, it remains to be determined where its legal strengths and weakness are and how enforceable its terms will be.

A.1.3 The Dual Licensing Option

Several commercial organizations are pursing a dual licensing model to support their business. These organizations provide their software to users, with a choice of an open source license or a propriety license. Users can decide, based on their needs, the type of relationship they will have with the software and its developers. Examples of commercial organizations pursuing a dual licensing model are Sleepycat Software Inc.(database); MySQL AB (database), TrollTech AS (development tools) and eZ systems (content management).

A.1.4 Hybrid Models are Emerging

As the open source market evolves, adaptation will continue in its legal infrastructure. Potential benefits include increased data interchange, and added value features incorporated into proprietary derivative offerings.

The core technology of Snort.org, an OSS project focused on intrusion detection security, is licensed under the GNU GPL. Its founding developers offer a commercial and proprietary offering through SourceFire, Inc.

RealNetworks, Inc., a proprietary software firm, is sponsoring an open source project, Helix.org, to deliver a media player to the Linux platform. This project intends to deliver an offering incorporating both proprietary and open source technologies.

A.1.5 Microsofts Shared Source Initiative

In 2001, responding to the open source movement, Microsoft Corp. announced its Shared Source Initiative, a program to share Windows source code with governments, companies and educational institutions. The program allows governments signing up to obtain free online access to source code and other technical information needed to perform security reviews.

In January 2003, Microsoft announced the Government Security Program (GSP), which allows governments and international organizations to assess the security and integrity of Microsoft software. Austria, China, Finland, Norway, Russia, Taiwan, Turkey, the U.K. and the North Atlantic Treaty Organization (NATO), among others, have signed up for this program. The GSP covers cur-rent versions of Windows 2000, Windows XP and Windows Server 2003 and Windows CE.

In contrast to open source licenses, Microsoft's shared source efforts do not allow users to modify the code or turn it into derivative Windows programs themselves.

A.1.6 Observations on Software Patents

Until the 1980s, it was generally believed in the US that patent law did not cover software programs. A frequently cited memo by Microsoft's Bill Gates in 1991 included the statement "If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the industry would be at a complete standstill today."

Organizations have increased the frequency and scope of software patenting. Recently, the US Patent Office has been criticized for actions such as its patent to Amazon, giving the firm exclusive rights to the "one-click" method for selling merchandise online. On Feb.7, 2003; the Patent Office's head James Rogan said, "This is an agency in crisis and it's going to get worse. It doesn't do me any good to pretend there's not a problem when there is." During Microsoft's financial analyst meeting (July 2003), Bill Gates noted, "last year we applied for 1,500 patents, and that's a number that's been going up at a pretty steep ramp". The US Patent and Trademark Office issues over 20,000 new software patents every year.

Appendix B. Open Source and Software Markets

This material is extracted from [FLOS 2002a, FLOS 2002b, FLOS 2002c].

B.1 The Software Market

B.1.1 Software market within the IT market

The IT market can be differentiated into four different market segments: Hardware products, hardware maintenance services, software products and services, Internet and processing services. We will limit the analysis on the segments and strategies where OSS has a significant influence. It will become clear that between the four market segments several dependencies exist.

B.1.2 The market segments

The software market can be divided into three segments:

- Software products (further to be separated in enterprise solutions and packaged mass-market software),
- Professional software-related services, and
- Embedded software and services.

The market segments will be analyzed in detail in section 4.2. The primary objective is to generate hypotheses about the influence of OSS in the segments.

B.2 Business dynamics and influence of Open Source software

B.2.1 The software products market

For the market analysis, we will adopt a customer-driven segmentation of software. We will segment the market according to customer purchase decisions.

We will make two basic distinctions: The first one is between enterprise solutions and packaged mass-market software. The market for enterprise solutions is substantially different from the market for packaged software because the revenue portion based on service fees is considerably higher. Therefore, the market for enterprise solutions can be mainly characterized as service market. But the service sales are primarily based on special software developed.

The second distinction is within the market for packaged software. We distinguish the server and the desktop market. Within these segments, one could additionally segment between operating systems and applications. But due to the fact that the purchase decision for the operating system is dependent on the decision for applications and vice versa, we will look at the server market in general and only when necessary separate between operating systems, applications or even appliances. Similarly, we will look at the desktop market in general.

Together with the segment enterprise solutions, we have three market segments:

- Server operating systems and applications,
- Desktop/client operating systems and applications, and
- Enterprise solutions.

B.2.1.1 Business dynamics in the software products market

In 1999, McKinsey analyzed the software market and identified general business dynamics in the product business. The following conclusions were derived: In the software product business, entry barriers are low. The market is knowledge-driven, only low capital investment is needed. The low financial entry barriers boost high innovation rates (short time between releases), which, in turn, lower technical entry barriers. Therefore, firms in this market face a constant threat of new entrants. Low marginal costs result in worldwide markets, high fixed costs for research and development result in the condition that many copies must sell. The latter statement becomes blurred when considering OSS development. While the fixed costs are high in terms of hours devoted to a specific project, there are not necessarily monetary costs involved. Thus, an OSS could – at least in principle – survive despite a combination of large efforts and insufficient usage. Whether the programmers find this appealing in the long-run and will not switch to other, more successful projects, is another question.

There is a race for leadership and a tendency towards concentration because of the following aspects: Firstly, there exists a network effect because of the interoperability of programs. The ability of programs to operate and communicate with each other is crucial. Hence, the more users a program has, the more the users can benefit from it. Secondly, there is a barrier for people to switch once they are trained to work with a program. Thirdly, popularity of a program is a major factor for the purchase decision (again a network effect).

The tendency for concentration seems to be stronger in consumer markets, where people only execute programs. In markets, where the purchase decisions are made by people with special IT knowledge, the training barrier and the popularity barrier are usually not as high. People with IT knowledge base their purchase decisions to a lesser extent on popularity. They also get used to a new program more easily. But any market position of a player in the software products market is never stable because of technology switches.

B.2.1.2 Server operating systems and applications

The market for server operating systems and their applications can be divided into different segments. We have to look at the high-end server market with different players, where the interdependence with hardware is inevitable because new hardware developments (such as the Itanium processor) consequently lead to various projects porting various operating systems to the new hardware. The other segment is the low-end server market with a variety of applications. In this market, we have to look at the operating systems on one hand and on the server applications on the other hand. The applications include:

- Intranet servers (database servers, file servers, print servers),
- Multi usage servers (web servers, mail servers, streaming media servers, chat servers), and
- Connectivity servers (firewalls, gateways/routers, dialup servers).

The customers in the server market are different from the customers in the desktop market. The decision makers have in general a basic IT knowledge and are corporate customers. Purchase decisions are made by management (small companies), IT managers (small and medium companies) or by IT departments (medium and large companies). The purchase decisions are made on criteria different from the desktop market. Therefore, the server market is driven by different critical success factors. Most corporate customers buy solutions rather than mere products. Hence, service companies, ISVs (independent software vendors) and VARs (value added resellers) are important distribution channels. Partnerships and co-operations with these groups are one critical success factor.

Important products and players in the high-end server operating systems market are: IBM (AIX, AS400, OS/390, OS/400), Fujitsu-Siemens (BS2000, BS2000/OSD), Sun (Solaris), HP (different Unix versions), Silicon Graphics (Irix), Compaq/Digital (Unix, VMS). The main products and players in the low-end server operating systems market are Microsoft (Windows plus server applications) and Unix in different variations and derivatives (one of which is Linux plus applications). Linux is supposed to offer several advantages and to constitute a serious alternative as server operating system due to the following advantages.

- High stability and high reliability,
- Low/zero license costs,
- Freely modifiable features, and
- Relatively small kernel.

Furthermore, many applications for Unix systems are relatively easily transferable to Linux. Also many software application vendors are interested in having their Unix applications to run on Linux and vice versa as support of a single product is much cheaper than supporting different versions. Thus, there are a number of factors and developments that support the hypothesis that Linux market share in the server market is going to rise further.

In the server applications market, we are not only talking about Linux. It is the web server software Apache that leads the market of web servers on the public Internet with a market share in December 2003 of about 69 percent. Microsoft had a market share of 23 percent [Netc 2003]. Besides the web server combination of Linux and Apache also the file server combination Linux and Samba has significant market share.

B.2.1.3 Desktop/client operating systems and applications

The market for desktop/client operating systems and their applications is the classical mass market. It has a high tendency towards concentration and is dominated by Microsoft. Customers base their purchase decision on popularity, on interoperability with other systems and applications, and on number and quality of applications available for the operating system. In addition, they face switching costs due to training. Therefore, the market leader has a strong position.

Service companies and VARs (value added resellers) are an important sales channel for private and SME sales because many customers buy computer equipment with the operating system and the applications pre-installed. Hardware companies and their sales channels are important because of the integration of software and hardware (pre-installation) and the bundled sales. For software producers, the partnerships and co-operations with hardware producers and with service companies and VARs are critical success factors.

The main players and products with regard to desktop/client operating systems are (market shares according to IDC, 2000a):

- Microsoft (Windows) dominates with a 88 percent market share in 1999.
- Apple (Mac OS, Mac OS X) comes second with a significantly lower market share of 5 percent in 1999.

Linux on the desktop has a market share of 4 percent in 1999. Since the time this data was generated, the market share of Microsoft has even increased to well above 90 percent.

The applications in the desktop/client market can be segmented along several criteria. The most simple differentiation is into horizontal (the general office programs) and vertical (functionally specific or industry-specific) applications. As already mentioned, the market for applications is closely linked to the market for operating systems. Therefore, the players are basically the same. The market leadership for applications is held by Microsoft with its Office Suite, which also runs on many Mac-OS desktops. There are various OSS applications, which run on Linux or other OSS operating systems, but also on Windows.

There are different possibilities of competition between the market leader Microsoft with its Windows and Office bundle and open source software. First of all, there is direct competition of bundles. For a long time Linux in combination with open source programs (e.g., OpenOffice) has been positioned as competitor. As these applications do, however, not provide the same level of user-friendliness and familiarity as the Microsoft family, strong competition is unlikely. The unequal positions

are even strengthened by the fact that Microsoft products are sold pre-installed with most Intel-based computers and many (private and SME) users buy a bundle of hardware, operating system and applications.

This has changed with MacOS X which, although not purely open source, is based (in parts) on an open source Unix. Together with the available Microsoft Office suite, this product bundle of hardware, operating system and applications provides a stronger competition to the Wintel bundle. Although not purely open source, the advantages of the open source development model might make the Apple alternative more powerful and might increase Apple's chances to compete more strongly with Microsoft. The outcome of this stronger competition has to be awaited.

A further influence of open source can be in the field of desktop applications. Several larger desktop application projects, e.g. Mozilla or OpenOffice, have reached a stage, where it is expected that they can become strong competitors to the respective Microsoft products Internet Explorer and Office within one or two years.

B.2.1.4 Enterprise solutions

The major products in the enterprise solutions segment are ERP (enterprise resource planning) systems. But several other products belong to this category as well, e. g., CRM (customer relationship management) software, SCM (supply chain management) software, KM (knowledge management) software, groupware, e-learning soft-ware, etc. Worldwide, the main players are: IBM, Oracle, Computer Associates, SAP, HP, Fujitsu-Siemens, Hitachi, Parametric Technology, Peoplesoft, and Baan.

Enterprise solutions almost always need customization. Hence, the market is characterized by revenues that are based on product licenses on one hand and service fees on the other hand. According to an internal McKinsey study, 30 percent of the installation costs for an ERP system account for the software product license, 70 per-cent account for professional services fees to implement the product. Some software companies offer the service themselves, some through partnerships. Customization and installation projects typically take several months to be completed. As this market is somehow between the products and the services market, we will include the market for enterprise solutions in the analysis of the services market.

B.2.2 The market for software-related services

B.2.2.1 Business dynamics in the market for software-related services

The business dynamics in the market for software-related services differ in several ways from those for the software products market, mainly due to its service character. The capital to be invested to start a service company is low. Instead, knowledge is the major foundation and entry barrier. Thus entry barriers are low, and incumbents constantly have to be aware of new entrants.

A high pace of innovation in the software products market leads to a similarly high pace in the services business. New technologies arise and with them new companies, which are able to offer the service. Some players in the services business are offering products as well.

Software services are a classical people-selling business with constant and significant marginal costs. In this sense, the services market is different from the product market. The cost of a second project is quite the same as the cost for the first project, even when a similar solution is implemented. Human resources are the most important asset in the services market. Contrary to the product business, the services market is not ruled by the law of increasing returns. Therefore, the market faces much higher fragmentation and one can find:

- Small companies,
- Regionally focused companies, and
- Very few truly global players in service firms (e.g., Accenture, IBM Global Services, EDS, CSC, Cap Gemini Ernst & Young, KPMG, Pricewaterhouse-Coopers, Deloitte & Touche, Cambridge Technology Partners).

B.2.2.2 Market segments for software-related services

The services market consists of two basic market segments:

- IT consulting and systems integration, and
- IT services.

The firms in the IT consulting and systems integration segment provide mainly customized services (solutions), including customized software products.

The IT services include support and maintenance, training and application management (including outsourcing). The firms in this field provide mainly these services, but they can sometimes also offer consulting and systems integration. The IT services seem to be stronger related to a product business. Many of the larger firms in this field are service units of hardware producers, sometimes even separate entities, such as Siemens Business Services or IBM Global Services. Others are primarily software producers where the focus on services becomes more important and generates a higher portion of revenues than the software sales. There is no clear line between the two market segments, and many companies are active in both segments. The two segments are represented in the business structure of some firms in the service market. They usually have a business unit for "corporate solutions" (which includes consulting, implementation, and integration). The business is project-related – normally a project is finished once a new solution is implemented and the customer is trained to work with it.

Major firms active in these services markets are:

- The global service companies that usually offer services to large (global) enter-prises.
- Worldwide: Accenture, IBM Global Services, EDS, CSC, Science Appli-cations, Cap Gemini Ernst & Young, Hewlett Packard, DEC, BSO Ori-gin, Fujitsu-Siemens (Hoch et al., 1999, 27).
- Germany: CSC Ploenzke, CAP Gemini Ernst&Young, Accenture, gedas, PricewaterhouseCoopers Unternehmensberatung, KPMG Consulting, Plaut Gruppe, Atos Origin, IBM Deutschland, Siemens

Business Services, T-Systems, Hewlett-Packard, Lufthansa Systems Group, GE CompuNet Computer (Lünendonk, 2000)

– Thousands of small service firms, either focused on specific solutions and technology or on their home region. The latters' main customers are often SMEs.

With regard to the global services companies and their large integration projects – such as enterprise solutions – OSS will not yet have a significant influence. The OSS-related discussion is mainly focused on the question whether the main firms in the market of enterprise solutions will make their solutions portable on Linux. (SAP, IBM, Oracle, Software AG, Sybase all have already done so with major ERP applications.) With regard to small service companies, however, OSS has a significant influence. Firstly, there are service companies that focus exclusively on Linux and other OSS. Secondly, there are distributors of OSS that offer services and support for their products. Thirdly, there are specialized service companies that shift to OSS or include Linux and other OSS into their spectrum.

B.2.3 The market for embedded software products and services

B.2.3.1 Business dynamics in the embedded software market

The market for embedded software products is a classical B2B market. The software companies that supply embedded software are the suppliers for the device producers.

We will limit the analysis to embedded products and, in this first step, to the embedded operating systems in particular. The business dynamics in the embedded software products market are similar to the dynamics in the non-embedded market. The major business dynamics can be summarized as follows:

– The indirect network effect is strong because of the interoperability and portability of software and because of the skills of developers. The less common any embedded system tool is, the fewer common components can be used and the fewer people are able to handle the embedded software. The network effect is especially strong for operating systems, which form a basis for many other software applications and software development tools.
– The business is knowledge-driven and the capital investment is low, which results in high innovation rates. However, the network effect establishes a high entry barrier.
– The marginal costs are low and result in worldwide markets. With non-OSS, the fixed costs for development are high and result in the condition that many copies must sell in order to have a profitable business.
“ In the long term, the market position is never stable because of technology switches.

B.2.3.2 Analysis of the embedded software market

Traditionally, the largest share of the embedded operating systems and kernel market is occupied by so-called “home-grown” operating systems.

These are operating systems, which are privately developed and maintained. They used to account for between one half and two thirds of all embedded systems' operating systems (Evans Data Corporation, 2001). According to a survey among 500 developers in 2001, home grown systems are still the most widely used systems, followed by WindRiver's operating systems and DOS. Embedded Linux, as an open operating system, is already number four. The home-grown systems face several difficulties, which will become even stronger in the near future. Basically, applications in the embedded market are getting much more complicated, and networking is becoming more important, which makes developing and maintaining these private systems much more expensive. Therefore, it is predicted that many of the home-grown operating systems will switch to Linux in the near future. (Evans Data Corporation, 2001) Half of the home-grown systems operate real-time, half of them are smaller, non real-time systems. A real-time operating systems needs to guarantee that a given operation will be performed within a given time window. Hence, the critical factor for real-time is determinacy.

Linux is supposed to have several advantages compared to the home-grown systems (Cook, 2000):

- It is standardized and open source. It has a modular structure and can be trimmed down for several purposes. Hence, the features of Linux are freely modifiable for various reasons.
- It has a low price.
- There are many skilled programmers available on the labour market.
- Compared to private operating systems, there are many drivers available. (Compared to Windows, of course, not so many.)

However, there are also several disadvantages:

- Linux is not designed to be a real-time system. Although there are various projects to develop components to make Linux real-time for various situations, the results are not comparable to operating systems that have been designed for real-time performance.
- Compared to other proprietary embedded operating systems, there is still a lack of drivers for Linux.
- Linux is relatively large by embedded standards. However, it can be trimmed down and there is a trend for hardware products becoming cheaper. And the kernels of its major competitors offering proprietary software are also quite large.
- Embedded systems are often mission-critical for business processes. Here, the perceived lack of professional 24h support is a disadvantage for Linux.

Linux constitutes an alternative for home-grown systems when their operations not necessarily have to be real-time and when their operations are not mission-critical. A number of Linux embedded products is expected, such as set-top boxes, cable TV, toys, car devices, manufacturing devices, in the long-term also house-hold devices (IDC, 2000a). Here, Linux has potential to gain market share. The major suppliers of embedded Linux are seen to be Lineo, MontaVista and Red Hat. (VDC, 2001)

Appendix C. Business Models on open source software

This material is extracted from [FLOS 2002a, FLOS 2002b, FLOS 2002c].

C.1 Overview

The focus is on business models that are purely based on OSS. This means they would not exist without the occurrence of the OSS phenomenon. Similar to the software market analysis product-related businesses are differentiated from service-related ones.

C.1.1 The market for Linux distributors

Linux distributors cover two market segments: First, the mass market with standardized packages offered to SMEs and private consumers and second, the market for individual solutions, which are offered to medium to large corporate customers. The mass market for operating systems separates into the server and the desktop market. In the server market, OSS is supposed to offer several advantages and to constitute a serious alternative as server operating system. The major competitors are Windows NT and the various other Unix systems. In the desktop market, the Linux market share is very small. Here, Linux has one major competitor: Microsoft with its various Windows versions. The question for the coming years is whether the Linux operating system will be successful on the desktop or not.

The solutions market (service-related) is completely different from the mass market (product-related). Most of the distributors consider the solution business as increasing and profitable income stream. Some of them cover the solutions market through partnerships with consulting companies. On the other hand, companies like SuSE have build up their own profit centre for corporate users.

C.1.1.1 Advantages and disadvantages of business model

In the mass market, Linux itself can be regarded as a commodity because its components can be freely downloaded or copied. The packaging, which is done by every distributor and results in the different Linux versions, constitutes the added value. However, the margins per unit sold are not very high. The product business with Linux distributions is a mass market and the active companies will have to increase their product sales (and eventually their market share) to become profitable.

Therefore, the distributors are forced to develop other means to differentiate themselves from each other and from proprietary competitors. So far, this differentiation has been mainly achieved by branding, a critical success factor in the mass market. Linux distributors such as Red Hat or SuSE put much effort into marketing (advertising, trade fairs, even certification of their Linux trainings can be regarded as a brand building effort).

The second success factor is gaining access to sales channels, which are bookstores and VARs. Mandrakesoft, for example, exclusively cooperates with Macmillan bookstores in the US. In Germany, the company is trying to intensify partnerships with small consulting companies and integrators to gain access to the SME segment.

As it is difficult to survive on the product business alone, the distributors build up a second business by shifting towards the solutions and consulting market. This might be interpreted as a move towards a more lucrative business as pure software retailing provides only low margins because the marketing costs are high and, in the desktop market, the number of potential buyers of Linux box products is still low compared to the buyers of Microsoft box products. Due to their software knowledge from packaging and optimizing the Linux parts, the distributors certainly have the OSS as well as technical competence to build up consulting and service business. But it is in question whether they already have the know-how in consulting and business processes to become serious competitors for existing service or consulting firms.

C.1.2 Niche and specialty OSS distributors

The niche or specialty distributors develop and distribute different OSS but no operating systems. Their products include applications, development and administrative tools. Normally, their software is developed to run on Linux, but some products also run on Windows or other operating systems. Examples are Zope (formerly Digital Creations), Sendmail.com, Covalent Technologies, Cygnus (acquired by Red Hat), Precision Insight (acquired by VA Linux), MySQL, ActiveState, and CollabNet. In this business model, companies live symbiotically off an OSS project. OSS is collected, maintained and/or developed. The main functions of those companies are to co-ordinate the scheduling and make a commitment to delivery and support of a dedicated product. Normally, they employ some of the core developers of the specific project and rely heavily on their relations to the developer community.

The market share for some OSS products such as Samba or the Apache web server is rather high. However, this does not say much about the success of the companies that are trying to profit from this development. The market access for the niche and specialty distributors is very different from the access for the Linux distributors. Normally, they do not directly target private consumers or SMEs with their products. The major Linux distributors can easily incorporate specialty software components into their out-of-the-box packages and, because of their stronger brands, sell it to a broad customer base. Nevertheless, some specialty distributors offer a limited number of packages that can be ordered on their websites and are distributed through the same channels as the Linux distributions (e.g. VARs and retail chains).

Hence, the major customers of specialty distributors must be either VARs (Value Added Resellers) or OEMs (Original Equipment Manufacturers), which sell optimized hardware-software bundles or develop and sell embedded products.

C.1.2.1 Advantages and disadvantages of business model

Because the companies in this category develop and distribute OSS, which can normally just be downloaded and copied, it is hard to imagine what kind of direct business model they could use. Indeed, simply selling software does not propose a unique selling proposition because access to OSS is largely unrestricted, at least in most license models. As a result, there are a variety of income streams summarized in this model. Most of the companies sell additional services for their products (consulting and support).

Some of them, for example MySQL, generate income from license fees for commercial licenses for the normally GPLed MySQL. Others, for example Precision Insight, decide not to disclose the source code for the newest version of their products, but only for previous versions. Another way, used by Sendmail.com, is to develop proprietary commercial software on top of the basic sendmail functionality. Most of these ideas are critical because the companies leave the field of pure OSS players and become a player in the traditional software business. They have to be able to live in both worlds.

C.1.3 Retailers of OSS distributions and complementary products

The retailers are major sales channels for the distributors. They either sell the distributors' software products or they provide and sell additional documentation and in-formation on OSS products or merchandise. The retailers are not solely focused on OSS. The retailers and specialized Linux shops target the mass market only. Their custom-ers are private or corporate users, developers or IT-administrators. For OSS, the market is slowly shifting to users that are not "software freaks" or developers, but instead use the software.

C.1.3.1 Advantages and disadvantages of business model

The advantage of established retailers is their access to customers via their retail stores. Another advantage is the known brand of retail chains. Web-based resellers and specialized Linux shops will find it difficult and expensive to create brand awareness, especially when they compete directly with the distributors and publishers.

Merchandising, in general, is not a business model itself but only an additional income stream whenever a strong brand has been established. Therefore, income from merchandising is primarily interesting for the Linux distributors.

There was a need for documentation as OSS became more popular and was applied outside the OSS community. For commercially developed software, the documentation is normally done by the software producer or in cooperation with a publisher. O'Reilly combined its OSS knowledge with publishing knowledge and succeeded in establishing a brand for OSS books. Because the company covers many OSS projects, it is not dependent on a single software development.

C.2 OSS-related Services

C.2.1 OSS development and community enablers

This category includes primarily two different sorts of actors. These are first of all marketplaces like SourceXchange, Cosource.com, intraDAT (vshop.org) and secondly conference and trade fair organisers like LogOn Technology Transfer or Linux. The function of exchanges or marketplaces is to match potential buyers (organizations or individuals looking for "needed improvement") and sellers (OSS developer community). The software produced would be customized or build-to-order OSS. The main argument for the potential of these exchanges is the assumption that many software developers want to decide themselves what project to work on (which is not possible as an employee). Additionally, the global reach of the Internet could leverage the developer potential all around the world and possibly even drive the prices down. The marketplaces for software development offer the matching service and improve the development process through provision of a project manager and productivity tools. Multiple buyers with the same problem aggregate their funds via the marketplace to get a software solution. As far as known, no company in this business has become profitable so far. SourceXchange closed in April 2001.

Conference organizers either are specialized in OSS and Linux or are general conference organizers that generate part of their income through their focus on OSS.

C.2.1.1 Advantages and disadvantages of business model

Pure marketplaces and exchange models have failed. Probably the value added is not sufficient to carry a business model purely on the matching function. Revenues can only be generated from the demand side as the developers are probably not willing to pay for the service.

On the demand side the "buyers" of the software might not have trust in the completion of the projects. It seems unlikely that a company would trust a developer community with no or vague responsibilities and certainly no guarantees for completion of projects. On the supply side, the main competition of these business models is the OSS community itself and all the projects that are managed by volunteers. The matching function could rather be used as an additional service in the spectrum of a service company.

So far, OSS conferences and fairs have usually had comparatively low prices and it is questionable whether the organizers can operate profitably. They cannot demand high entrance fees for conferences because most people interested are OSS community members. Nor can they demand high fees from software suppliers because many suppliers are community projects or small and regionally focused service companies.

C.2.2 OSS-related services and support

OSS-related services and support include several services such as consulting, systems integration, support, maintenance, remote administration, training, and application management. The companies in the OSS-related services market differ according to their background. First of all, there are companies that have a background in Linux or other OSS products. They are trying to establish services that build on their product

knowledge. Hence, their core competence is the technological and product knowledge. Most of them offer a full range of services. Linux distributors, niche and specialty distributors as well as independent OSS service companies belong to this category.

And secondly, there are companies that have special process knowledge in how to provide a service related to IT in general. This can be knowledge in IT consulting, systems integration, IT-training or IT-recruiting, sometimes even with a vertical functional or industry-specific specialization. They can extend their offerings to OSS-related services.

Examples for full service companies offering various services based on OSS knowledge are:

- Linux distributors (e.g., Red Hat, SuSE, Caldera, MandrakeSoft, Turbolinux)
- Niche and specialty distributors (e.g., Zope, MySQL, Sendmail.com, Cova-lent Technologies)
- Independent OSS service companies (e.g., Linuxcare) and many small service and integration companies with special technological OSS and particular Linux knowledge (e.g., Linux Information Systems, B-connected).

Examples for special services based on integration and service knowledge extended to OSS are:

- For consulting and systems integration: (Global System Integrators; Accen-ture, KPMG, PricewaterhouseCoopers etc.), various small consulting and integration companies.
- For training: various training and e-learning companies (e.g., Microconsult)
- Recruiting and Staffing services: various IT-specialized recruiting companies (StepStone-IT, Monster.de, JobUniverse.de)

C.2.2.1 Product and service offerings

Consulting companies and system integrators help their customers realize IT strategies based on business needs as described in section 2.2. The critical factor is often Linux expertise: The small integrators and service companies do usually have a background in OSS development and are trying to establish a business based on services. The large integrators can activate their Unix people to acquire Linux expertise or hire OSS developers.

Support companies offer their services in various models, as the "classical" support model of OSS (send a bug report to the project's community) is not accepted by many business customers. Commercial support gives businesses the possibility to have their OSS products supported without getting involved with the developer community culture (even if the support staff are developers, e.g. at MandrakeSoft).

OSS focused training companies have two choices offering their courses: They can offer classical seminars with physical attendance (often after a certification process by the software vendor) or e-learning solutions (e.g. Red Hat).

C.2.2.2 The market for OSS-related services and support

The customers for systems integration range from small to large corporations, which pay for a solution instead of paying for a product. Hence, the service is project-related. Support is needed in any market and on any user level. For example, OEMs (original equipment manufacturers) and ISVs (independent software vendors) can be custom-ers of Sendmail.com support when incorporating Sendmail into their product. System administrators do usually need support when a new product is implemented. Buht also private and business users need support with their product (which is normally offered in a standardized way).

Customers for OSS training are users on various levels: Red Hat, for example, is offering courses for users, systems administrators and developers in classical seminars as well as in e-learning courses. Customers typically are business-related users. Training products at Red Hat have one focus on the Red Hat Linux distribution and related software; under "E-Business" they offer a course for SAP-Red Hat integration. (In their e-learning courses, they also offer C/C++/Java programming and general Unix/ networking courses.)

C.2.2.3 Advantages and disadvantages of business model

There are two fundamentally different groups of firms active in the OSS-related services market. Firms with OSS background have substantial product and technology knowledge, which they use to build up their services business. Businesses solely based on OSS products are dependent on the acceptance and development of OSS. Companies without OSS background have substantial process know-how in the services sector. They attempt to extend their offering to OSS-related services. Whether firms from one group or from the other group will succeed, depends on the importance of product know-how vs. process know-how in the separate service fields. Figure XXX illustrates this relation.

The firms with OSS background will be mainly successful in areas where product know-how is important and process know-how can be easily acquired. This is the case with support and training offerings. The players without special OSS know-how will be mainly successful in areas where this know-how plays only a minor role or can easily be acquired.

Appendix D. Edgar Villanueva letter to Microsoft

Lima, 8th of April, 2002
To: Señor JUAN ALBERTO GONZÁLEZ
General Manager of Microsoft, Perú

Dear Sir:

First of all, I thank you for your letter of March 25, 2002 in which you state the official position of Microsoft relative to Bill Number 1609, Free Software in Public Administration, which is indubitably inspired by the desire for Peru to find a suitable place in the global technological context. In the same spirit, and convinced that we will find the best solutions through an exchange of clear and open ideas, I will take this opportunity to reply to the commentaries included in your letter.

While acknowledging that opinions such as yours constitute a significant contribution, it would have been even more worthwhile for me if, rather than formulating objections of a general nature (which we will analyse in detail later) you had gathered solid arguments for the advantages that proprietary software could bring to the Peruvian State, and to its citizens in general, since this would have allowed a more enlightening exchange in respect of each of our positions.

With the aim of creating an orderly debate, we will assume that what you call "open source software" is what the Bill defines as "free software", since there exists software for which the source code is distributed together with the program, but which does not fall within the definition established by the Bill; and that what you call "commercial software" is what the Bill defines as "proprietary" or "unfree", given that there exists free software which is sold in the market for a price like any other good or service.

It is also necessary to make it clear that the aim of the Bill we are discussing is not directly related to the amount of direct savings that can by made by using free software in state institutions. That is in any case a marginal aggregate value, but in no way is it the chief focus of the Bill. The basic principles which inspire the Bill are linked to the basic guarantees of a state of law, such as:

– Free access to public information by the citizen.
– Permanence of public data.
– Security of the State and citizens.

To guarantee the free access of citizens to public information, it is indespensable that the encoding of data is not tied to a single provider. The use of standard and open formats gives a guarantee of this free access, if necessary through the creation of compatible free software.

To guarantee the permanence of public data, it is necessary that the usability and maintenance of the software does not depend on the goodwill of the suppliers, or on the monopoly conditions imposed by them. For this reason the State needs systems the development of which can be guaranteed due to the availability of the source code.

To guarantee national security or the security of the State, it is indispensable to be able to rely on systems without elements which allow control from a distance or the undesired transmission of information to third parties. Systems with source code freely accessible to the public are required to allow their inspection by the State itself, by the citizens, and by a large number of independent experts throughout the world. Our proposal brings further security, since the knowledge of the source code will eliminate the growing number of programs with *spy code*.

In the same way, our proposal strengthens the security of the citizens, both in their role as legitimate owners of information managed by the state, and in their role as consumers. In this second case, by allowing the growth of a widespread availability of free software not containing *spy code* able to put at risk privacy and individual freedoms.

In this sense, the Bill is limited to establishing the conditions under which the state bodies will obtain software in the future, that is, in a way compatible with these basic principles.

From reading the Bill it will be clear that once passed:

– the law does not forbid the production of proprietary software – the law does not forbid the sale of proprietary software
– the law does not specifiy which concrete software to use
– the law does not dictate the supplier from whom software will be bought
– the law does not limit the terms under which a software product can be licensed.

What the Bill does express clearly, is that, for software to be acceptable for the state it is not enough that it is technically capable of fulfilling a task, but that further the contractual conditions must satisfy a series of requirements reguarding the license, without which the State cannot guarantee the citizen adequate processing of his data, watching over its integrity, confidentiality, and accessibility throughout time, as these are very critical aspects for its normal functioning.

We agree, Mr. Gonzalez, that information and communication technology have a significant impact on the quality of life of the citizens (whether it be positive or negative). We surely also agree that the basic

values I have pointed out above are fundamental in a democratic state like Peru. So we are very interested to know of any other way of guaranteeing these principles, other than through the use of free software in the terms defined by the Bill.

As for the observations you have made, we will now go on to analyse them in detail:

Firstly, you point out that: "1. The bill makes it compulsory for all public bodies to use only free software, that is to say open source software, which breaches the principles of equality before the law, that of non-discrimination and the right of free private enterprise, freedom of industry and of contract, protected by the constitution."

This understanding is in error. The Bill in no way affects the rights you list; it limits itself entirely to establishing conditions for the use of software on the part of state institutions, without in any way meddling in private sector transactions. It is a well established principle that the State does not enjoy the wide spectrum of contractual freedom of the private sector, as it is limited in its actions precisely by the requirement for transparency of public acts; and in this sense, the preservation of the greater common interest must prevail when legislating on the matter.

The Bill protects equality under the law, since no natural or legal person is excluded from the right of offering these goods to the State under the conditions defined in the Bill and without more limitations than those established by the Law of State Contracts and Purchasing (T.U.O. por Decreto Supremo No. 012–2001-PCM).

The Bill does not introduce any discrimination whatever, since it only establishes *how* the goods have to be provided (which is a state power) and not *who* has to provide them (which would effectively be discriminatory, if restrictions based on national origin, race religion, ideology, sexual preference etc. were imposed). On the contrary, the Bill is decidedly antidiscriminatory. This is so because by defining with no room for doubt the conditions for the provision of software, it prevents state bodies from using software which has a license including discriminatory conditions.

It should be obvious from the preceding two paragraphs that the Bill does not harm free private enterprise, since the latter can always choose under what conditions it will produce software; some of these will be acceptable to the State, and others will not be since they contradict the guarantee of the basic principles listed above. This free initiative is of course compatible with the freedom of industry and freedom of contract (in the limited form in which the State can exercise the latter). Any private subject can produce software under the conditions which the State requires, or can refrain from doing so. Nobody is forced to adopt a model of production, but if they wish to provide software to the State, they must provide the mechanisms which guarantee the basic principles, and which are those described in the Bill.

By way of an example: nothing in the text of the Bill would prevent your company offering the State bodies an office "suite", under the conditions defined in the Bill and setting the price that you consider satisfactory. If you did not, it would not be due to restrictions imposed by the law, but to business decisions relative to the method of commercializing your products, decisions with which the State is not involved.

To continue; you note that:" 2. The bill, by making the use of open source software compulsory, would establish discriminatory and non competitive practices in the contracting and purchasing by public bodies..."

This statement is just a reiteration of the previous one, and so the response can be found above. However, let us concern ourselves for a moment with your comment regarding "non-competitive... practices."

Of course, in defining any kind of purchase, the buyer sets conditions which relate to the proposed use of the good or service. From the start, this excludes certain manufacturers from the possibility of competing, but does not exclude them "a priori", but rather based on a series of principles determined by the autonomous will of the purchaser, and so the process takes place in conformance with the law. And in the Bill it is established that *no-one* is excluded from competing as far as he guarantees the fullfilment of the basic principles.

Furthermore, the Bill *stimulates* competition, since it tends to generate a supply of software with better conditions of usability, and to better existing work, in a model of continuous improvement.

On the other hand, the central aspect of competivity is the chance to provide better choices to the consumer. Now, it is impossible to ignore the fact that marketing does not play a neutral role when the product is offered on the market (since accepting the opposite would lead one to suppose that firms' expenses in marketing lack any sense), and that therefore a significant expense under this heading can influence the decisions of the purchaser. This influence of marketing is in large measure reduced by the bill that we are backing, since the choice within the framework proposed is based on the *technical merits* of the product and not on the effort put into commercialization by the producer; in this sense, competitvity is increased, since the smallest software producer can compete on equal terms with the most powerful corporations.

It is necessary to stress that there is no position more anti-competitive than that of the big software producers, which frequently abuse their dominant position, since in innumerable cases they propose as a solution to problems raised by users: "update your software to the new version" (at the user's expense, naturally); furthermore, it is common to find arbitrary cessation of technical help for products, which, in the provider's judgement alone, are "old"; and so, to receive any kind of technical assistance, the user finds himself forced to migrate to new versions (with non-trivial costs, especially as changes in hardware platform are often involved). And as the whole infrastructure is based on proprietary data formats, the user stays "trapped" in the need to continue using products from the same supplier, or to make the huge effort to change to another environment (probably also proprietary).

You add: "3. So, by compelling the State to favour a business model based entirely on open source, the bill would only discourage the local and international manufacturing companies, which are the ones which really undertake important expenditures, create a significant number of direct and indirect jobs, as well as contributing to the GNP, as opposed to a model of open source software which tends to have an ever weaker economic impact, since it mainly creates jobs in the service sector."

I do not agree with your statement. Partly because of what you yourself point out in paragraph 6 of your letter, regarding the relative weight of services in the context of software use. This contradiction alone would invalidate your position. The service model, adopted by a large number of companies in the software industry, is much larger in economic terms, and with a tendency to increase, than the licensing of programs.

On the other hand, the private sector of the economy has the widest possible freedom to choose the economic model which best suits its interests, even if this freedom of choice is often obscured subliminally by the disproportionate expenditure on marketing by the producers of proprietary software.

In addition, a reading of your opinion would lead to the conclusion that the State market is crucial and essential for the proprietary software industry, to such a point that the choice made by the State in this bill would completely eliminate the market for these firms. If that is true, we can deduce that the State must be subsidising the proprietary software industry. In the unlikely event that this were true, the State would have the right to apply the subsidies in the area it considered of greatest social value; it is undeniable, in this improbable hypothesis, that if the State decided to subsidize software, it would have to do so choosing the free over the proprietary, considering its social effect and the rational use of taxpayers money.

In respect of the jobs generated by proprietary software in countries like ours, these mainly concern technical tasks of little aggregate value; at the local level, the technicians who provide support for proprietary software produced by transnational companies do not have the possibility of fixing bugs, not necessarily for lack of technical capability or of talent, but because they do not have access to the source code to fix it. With free software one creates more technically qualified employment and a framework of free competence where success is only tied to the ability to offer good technical support and quality of service, one stimulates the market, and one increases the shared fund of knowledge, opening up alternatives to generate services of greater total value and a higher quality level, to the benefit of all involved: producers, service organizations, and consumers.

It is a common phenomenon in developing countries that local software industries obtain the majority of their takings in the service sector, or in the creation of "ad hoc" software. Therefore, any negative impact that the application of the Bill might have in this sector will be more than compensated by a growth in demand for services (as long as these are carried out to high quality standards). If the transnational software companies decide not to compete under these new rules of the game, it is likely that they will undergo some decrease in takings in terms of payment for licences; however, considering that these firms continue to allege that much of the software used by the State has been illegally copied, one can see that the impact will not be very serious. Certainly, in any case their fortune will be determined by market laws, changes in which cannot be avoided; many firms traditionally associated with proprietary software have already set out on the road (supported by copious expense) of providing services associated with free software, which shows that the models are not mutually exclusive.

With this bill the State is deciding that it needs to preserve certain fundamental values. And it is deciding this based on its sovereign power, without affecting any of the constitutional guarantees. If these values could be guaranteed without having to choose a particular economic model, the effects of the law would be even more beneficial. In any case, it should be clear that the State does not choose an economic model; if it happens that there only exists one economic model capable of providing software which provides the basic guarantee of these principles, this is because of historical circumstances, not because of an arbitrary choice of a given model.

Your letter continues: "4. The bill imposes the use of open source software without considering the dangers that this can bring from the point of view of security, guarantee, and possible violation of the intellectual property rights of third parties."

Alluding in an abstract way to "the dangers this can bring", without specifically mentioning a single one of these supposed dangers, shows at the least some lack of knowledge of the topic. So, allow me to enlighten you on these points.

On security: National security has already been mentioned in general terms in the initial discussion of the basic principles of the bill. In more specific terms, relative to the security of the software itself, it is well known that all software (whether proprietary or free) contains errors or "bugs" (in programmers' slang). But it is also well-known that the bugs in free software are fewer, and are fixed much more quickly, than in proprietary software. It is not in vain that numerous public bodies reponsible for the IT security of state systems in developed countries require the use of free software for the same conditions of security and efficiency.

What is impossible to prove is that proprietary software is more secure than free, without the public and open inspection of the scientific community and users in general. This demonstration is impossible because the model of proprietary software itself prevents this analysis, so that any guarantee of security is based only on promises of good intentions (biased, by any reckoning) made by the producer itself, or its contractors.

It should be remembered that in many cases, the licensing conditions include Non-Disclosure clauses which prevent the user from publicly revealing security flaws found in the licensed proprietary product.

In respect of the guarantee: As you know perfectly well, or could find out by reading the "End User License Agreement" of the products you license, in the great majority of cases the guarantees are limited to replacement of the storage medium in case of defects, but in no case is compensation given for direct or indirect damages, loss of profits, etc... If as a result of a security bug in one of your products, not fixed in time by yourselves, an attacker managed to compromise crucial State systems, what guarantees, reparations and compensation would your company make in accordance with your licencing conditions? The guarantees of proprietary software, inasmuch as programs are delivered ``AS IS'', that is, in the state in which they are, with no additional responsibility of the provider in respect of function, in no way differ from those normal with free software.

On Intellectual Property: Questions of intellectual property fall outside the scope of this bill, since they are covered by specific other laws.

The model of free software in no way implies ignorance of these laws, and in fact the great majority of free software is covered by copyright. In reality, the inclusion of this question in your observations shows your confusion in respect of the legal framework in which free software is developed. The inclusion of the intellectual property of others in works claimed as one's own is not a practice that has been noted in the free software community; whereas, unfortunately, it has been in the area of proprietry software. As an example, the condemnation by the Commercial Court of Nanterre, France, on 27th September 2001 of Microsoft Corp. to a penalty of 3 million francs in damages and interest, for violation of intellectual property (piracy, to use the unfortunate term that your firm commonly uses in its publicity).

You go on to say that: "The bill uses the concept of open source software incorrectly, since it does not necessarily imply that the software is free or of zero cost, and so arrives at mistaken conclusions regarding State savings, with no cost-benefit analysis to validate its position."

This observation is wrong; in principle, freedom and lack of cost are orthogonal concepts: there is software which is proprietary and charged for (for example, MS Office), software which is proprietary and free of charge (MS Internet Explorer), software which is free and charged for (RedHat, SuSE etc Gnu/Linux distributions), software which is free and not charged for (Apache, OpenOffice, Mozilla), and even software which can be licensed in a range of combinations (MySQL).

Certainly free software is not necessarily free of charge. And the text of the bill does not state that it has to be so, as you will have noted after reading it. The definitions included in the Bill state clearly *what* should be considered free software, at no point referring to freedom from charges. Although the possibility of savings in payments for proprietary software licenses are mentioned, the foundations of the bill clearly refer to the fundamental guarantees to be preserved and to the stimulus to local technological development. Given that a democratic State must support these principles, it has no other choice than to use software with publicly available source code, and to exchange information only in standard formats.

If the State does not use software with these characteristics, it will be weakening basic republican principles. Luckily, free software also implies lower total costs; however, even given the hypothesis (easily disproved) that it was more expensive than proprietary software, the simple existence of an effective free software tool for a particular IT function would oblige the State to use it; not by command of this Bill, but because of the basic principles we enumerated at the start, and which arise from the very essence of the lawful democratic State.

You continue: "6. It is wrong to think that Open Source Software is free of charge. Research by the Gartner Group (an important investigator of the technological market recognized at world level) has shown that the cost of purchase of software (operating system and applications) is only 8% of the total cost which firms and institutions take on for a rational and truely beneficial use of the technology. The other 92% consists of: installation costs, enabling, support, maintenance, administration, and down-time."

This argument repeats that already given in paragraph 5 and partly contradicts paragraph 3. For the sake of brevity we refer to the comments on those paragraphs. However, allow me to point out that your conclusion is logically false: even if according to Gartner Group the cost of software is on average only 8% of the total cost of use, this does not in any way deny the existence of software which is free of charge, that is, with a licensing cost of zero.

In addition, in this paragraph you correctly point out that the service components and losses due to down-time make up the largest part of the total cost of software use, which, as you will note, contradicts your statement regarding the small value of services suggested in paragraph 3. Now the use of free software contributes significantly to reduce the remaining life-cycle costs. This reduction in the costs of installation, support etc. can be noted in several areas: in the first place, the competitive service model of free software, support and maintenance for which can be freely contracted out to a range of suppliers competing on the grounds of quality and low cost. This is true for installation, enabling, and support, and in large part for maintenance. In the second place, due to the reproductive characteristics of the model, maintenance carried out for an application is easily replicable, without incurring large costs (that is, without paying more than once for the same thing) since modifications, if one wishes, can be incorporated in the common fund of knowledge. Thirdly, the huge costs caused by non-functioning software ("blue screens of death", malicious code such as virus, worms, and trojans, exceptions, general protection faults and other well-known problems) are reduced considerably by using more stable software; and it is well-known that one of the most notable virtues of free software is its stability.

You further state that: "7. One of the arguments behind the bill is the supposed freedom from costs of open-source software, compared with the costs of commercial software, without taking into account the fact that there exist types of volume licensing which can be highly advantageous for the State, as has happened in other countries."

I have already pointed out that what is in question is not the cost of the software but the principles of freedom of information, accessibility, and security. These arguments have been covered extensively in the preceding paragraphs to which I would refer you.

On the other hand, there certainly exist types of volume licensing (although unfortunately proprietary software does not satisfy the basic principles). But as you correctly pointed out in the immediately precding paragraph of your letter, they only manage to reduce the impact of a component which makes up no more than 8% of the total.

You continue: "8. In addition, the alternative adopted by the bill (i) is clearly more expensive, due to the high costs of software migration, and (ii) puts at risk compatibility and interoperability of the IT platforms within the State, and between the State and the private sector, given the hundreds of versions of open source software on the market."

Let us analyze your stament in two parts. Your first argument, that migration implies high costs, is in reality an argument in favour of the Bill. Because the more time goes by, the more difficult migration to another technology will become; and at the same time, the security risks associated with proprietary software will continue to increase. In this way,

the use of proprietary systems and formats will make the State ever more dependent on specific suppliers. Once a policy of using free software has been established (which certainly, does imply some cost) then on the contrary migration from one system to another becomes very simple, since all data is stored in open formats. On the other hand, migration to an open software context implies no more costs than migration between two different proprietary software contexts, which invalidates your argument completely.

The second argument refers to "problems in interoperability of the IT platforms within the State, and between the State and the private sector" This statement implies a certain lack of knowledge of the way in which free software is built, which does not maximize the dependence of the user on a particular platform, as normally happens in the realm of proprietary software. Even when there are multiple free software distributions, and numerous programs which can be used for the same function, interoperability is guaranteed as much by the use of standard formats, as required by the bill, as by the possibility of creating interoperable software given the availability of the source code.

You then say that: "9. The majority of open source code does not offer adequate levels of service nor the guarantee from recognized manufacturers of high productivity on the part of the users, which has led various public organizations to retract their decision to go with an open source software solution and to use commercial software in its place."

This observation is without foundation. In respect of the guarantee, your argument was rebutted in the response to paragraph 4. In respect of support services, it is possible to use free software without them (just as also happens with proprietary software), but anyone who does need them can obtain support separately, whether from local firms or from international corporations, again just as in the case of proprietary software.

On the other hand, it would contribute greatly to our analysis if you could inform us about free software projects *established* in public bodies which have already been abandoned in favour of proprietary software. We know of a good number of cases where the opposite has taken place, but not know of any where what you describe has taken place.

You continue by observing that: "10. The bill demotivates the creativity of the peruvian software industry, which invoices 40 million US$/ year, exports 4 million US$ (10th in ranking among non-traditional exports, more than handicrafts) and is a source of highly qualified employment. With a law that incentivates the use of open source, software programmers lose their intellectual property rights and their main source of payment."

It is clear enough that nobody is forced to commercialize their code as free software. The only thing to take into account is that if it is not free software, it cannot be sold to the public sector. This is not in any case the main market for the national software industry. We covered some questions referring to the influence of the Bill on the generation of employment which would be both highly technically qualified and in better conditions for competition above, so it seems unnecessary to insist on this point.

What follows in your statement is incorrect. On the one hand, no author of free software loses his intellectual property rights, unless he expressly wishes to place his work in the public domain. The free software movement has always been very respectful of intellectual property, and has generated widespread public recognition of authors. Names like those of Richard Stallman, Linus Torvalds, Guido van Rossum, Larry Wall, Miguel de Icaza, Andrew Tridgell, Theo de Raadt, Andrea Arcangeli, Bruce Perens, Darren Reed, Alan Cox, Eric Raymond, and many others, are recognized world-wide for their contributions to the development of software that is used today by millions of people throughout the world. On the other hand, to say that the rewards for authors rights make up the main source of payment of Peruvian programmers is in any case a guess, in particular since there is no proof to this effect, nor a demonstration of how the use of free software by the State would influence these payments.

You go on to say that: "11. Open source software, since it can be distributed without charge, does not allow the generation of income for its developers through exports. In this way, the multiplier effect of the sale of software to other countries is weakened, and so in turn is the growth of the industry, while Government rules ought on the contrary to stimulate local industry."

This statement shows once again complete ignorance of the mechanisms of and market for free software. It tries to claim that the market of sale of non-exclusive rights for use (sale of licences) is the only possible one for the software industry, when you yourself pointed out several paragraphs above that it is not even the most important one. The incentives that the bill offers for the growth of a supply of better qualified professionals, together with the increase in experience that working on a large scale with free software within the State will bring for Peruvian technicians, will place them in a highly competitive position to offer their services abroad.

You then state that: "12. In the Forum, the use of open source software in education was discussed, without mentioning the complete collapse of this initiative in a country like Mexico, where precisely the State employees who founded the project now state that open source software did not make it possible to offer a learning experience to pupils in the schools, did not take into account the capability at a national level to give adequate support to the platform, and that the software did not and does not allow for the levels of platform integration that now exist in schools."

In fact Mexico has gone into reverse with the Red Escolar (Schools Network) project. This is due precisely to the fact that the driving forces behind the mexican project used license costs as their main argument, instead of the other reasons specified in our project, which are far more essential. Because of this conceptual mistake, and as a result of the lack of effective support from the SEP (Secretary of State for Public Education), the assumption was made that to implant free software in schools it would be enough to drop their software budget and send them a CD ROM with Gnu/Linux instead. Of course this failed, and it couldn't have been otherwise, just as school laboratories fail when they use proprietary software and have no budget for implementation and maintenance.

That's exactly why our bill is not limited to making the use of free software mandatory, but recognizes the need to create a viable migration plan, in which the State undertakes the technical transition in an orderly way in order to then enjoy the advantages of free software.

You end with a rhetorical question: "13. If open source software satisfies all the requirements of State bodies, why do you need a law to adopt it? Shouldn't it be the market which decides freely which products give most benefits or value?"

We agree that in the private sector of the economy, it must be the market that decides which products to use, and no state interference is permissible there. However, in the case of the public sector, the reasoning is not the same: as we have already established, the state archives, handles, and transmits information which does not belong to it, but which is entrusted to it by citizens, who have no alternative under the rule of law. As a counterpart to this legal requirement, the State must take extreme measures to safeguard the integrity, confidentiality, and accessibility of this information. The use of proprietary software raises serious doubts as to whehter these requirements can be fulfilled, lacks conclusive evidence in this respect, and so is not suitable for use in the public sector.

The need for a law is based, firstly, on the realization of the fundamental principles listed above in the specific area of software; secondly, on the fact that the State is not an ideal homogoneous entity, but made up of multiple bodies with varying degrees of autonomy in decision making. Given that it is inappropriate to use proprietary software, the fact of establishing these rules in law will prevent the personal discretion of any state employee from putting at risk the information which belongs to citizens. And above all, because it constitutes an up-to-date reaffirmation in relation to the means of management and communication of information used today, it is based on the republican principle of openness to the public.

In conformance with this universally accepted principle, the citizen has the right to know all information held by the State and not covered by well-founded declarations of secrecy based on law. Now, software deals with information and is itself information. Information in a special form, capable of being interpreted by a machine in order to execute actions, but crucial information all the same because the citizen has a legitimate right to know, for example, how his vote is computed or his taxes calculated. And for that he must have free access to the source code and be able to prove to his satisfaction the programs used for electoral computations or calculation of his taxes.

I wish you the greatest respect, and would like to repeat that my office will always be open for you to expound your point of view to whatever level of detail you consider suitable.

Cordially,
DR. EDGAR DAVID VILLANUEVA NUÑEZ
Congressman of the Republica of Perú.

Appendix E. Selected Survey Responses

We include below responses from a few key individuals to a survey we sent. Note that the responses are the personal views of the indicated persons and not necessarily that of ours.

E.1 Linus Torvalds, Father of Linux

> Open-source & software market

> Does open-source software make the software market in general more competitive or does it distort the market in some way?

I'm convinced that the greatest advantage of open source is that it is basically an open market approach to software, without arbitrary barriers in place between different entities. It makes a market that traditionally has huge barriers to competition _hugely_ more competitive, and by doing so not only improves things for customers, but makes it possible for small 11companies (and countries, for that matter) to enter the market at all.

> What does the future hold for open-source software and where do you see great or greater success?

Open source tends to work best in commoditizing software, and thus clearly the area where open source will make most of a difference is in infrastructure that pretty much any computer ends up needing. In contrast, proprietary software will probably tend to be mostly in specialized market niches, where the customer base isn't wide enough to create a self-maintaining open source culture.

> What role does open-source software play in developing countries?

I'm of two minds on that one. On the one hand, open source does make it _possible_ for developing countries to be part of the IT community. On the other hand, I have a somewhat pessimistic view of technology in that sense: more developed nations have huge advantages where technology tends to just widen the gap between the "have-nots" and the "haves".

In other words: I'm convinced that without open source, developing nations will be totally shut out of IT altogether, always at the mercy of companies that have

proprietary technology and without any means to close the gap. However, I'm also convinced that open source on its own is not enough to close the gap.

> Does open-source reduce the cost of IT investment in developing countries?

Quite possibly. We already see just having the _choice_ and the competition offered by open source being used to reduce costs.

However, I think that if the developing country is serious about not justseeing IT as a cost center, but as a requirement for national development, the real advantage of open source ends up being able to build up your own knowledge base. And that is not cheap in itself - you'll likely pay as much for that as you'd pay for a proprietary software solution.

The difference being that with the proprietary solution, you'll never catch up, and you'll have to pay forever, without ever learning anything yourself.

> Open-source models

> Is the open-source development model of relevance and significance to developing countries?

See above - I think the big significance is in how it makes it possible to bootstrap your own software and IT development with open source.

> Does open-source create new opportunities for developing countries'

> IT industries? Would it encourage collaboration between developing and developed countries? If so, how?

Collaboration is fundamental in any open source project. Not only is it how open source gets done in the first place in many cases, open source is also the only viable means of collaboration "as equals" rather than being tied by licensing agreements that inevitably will favor the stronger part.

> Sustainability

> Is OSS sustainable long term? Why or why not?

Software is fundamentally an endeavor where you build on existing pieces, and that makes OSS in my opinion not only sustainable, but inevitable. A proprietary model simply cannot in the long run afford to implement internally what open source creates all over the development spectrum.

> What models do you think exist for open-source sustainability?

The only model (ever) that makes sense is one of mutual benefit: companies, individuals and countries benefit from open source simply because the ever-increasing existing base of open source allows them to ignore problems that have already been solved, and concentrate on issues that actually add value.

Whenever the open source code-base has reached such a critical point, where it is cheaper to make small modifications to existing projects than it would be to try to create (or buy) a proprietary solution, open source sustains itself.

The problem has traditionally been to _get_ to that point, not to sustain the project once it has grown big enough.

> What motivates developers to participate in open source projects long term?

There are just too many answers to this. One is purely "economic" (even if the economy in question may not be directly monetary), in that peopleparticipate in existing projects because they don't have the resources to start from scratch, and open source is the only medium where they _can_ participate - since proprietary models inherently disallow participation by most of the engineers in the world. That's what "proprietary" means, after all, and that's the fundamental difference between open source and traditional proprietary software development.

At the same time, the open source development model also means thatdevelopers get _credit_ in the open by their peers - which means that you have other than just purely economic incentives to be part of the community.

> What development and management models enable sustainability of open-source projects?

> What development and management models enable high quality open-source software?

I think these two are the same question. And quite frankly, I don't really know. I've been doing this for over a decade, and I still don't have much of a "management model". The basic rule for me has always been to accept contributions as evenhandedly as possible _without_ having any barriers in place, and the model has really been self-organizing.

> Business models

> What kind of business models are available for open source exploitation in developing countries?

There's been whole papers written about this, so I won't answer in detail. Open source tends to work best when the software is seen as just one part of a larger "solution", and by using open source the cost of developing that part is lower to any particular user by being spread out over the whole open source community.

But support and maintenance ends up being other models, as do customization and distribution etc services.

> Policy

> What kind of policy frameworks are appropriate to enable the success of open-source in developing countries?

When it comes to developing countries, and _especially_ when looking at interactions with already developed countries, I think one policy that should always be

appropriate is to make sure that whatever technology transfer takes place will always end up moving the developing country towards a greater deal of self-reliance.

This fundamentally argues against just selling (or even giving away)proprietary technology of any kind. The technology used should always be aimed to be at a level where the recipient doesn't consider it to be "magic", but can actually at least try to replicate the experience.

E.2 Brian Behlendorf, Co-Founder Apache

> Open-source & software market

> Does open-source software make the software market in general more competitive or does it distort the market in some way?

OSS represents a fundamental shift in the software marketplace, away from the factory-style model and towards a services model. Every software company will have to recognize and adapt to this change or risk becoming unsuitable to customers and unable to compete. It doesn't mean you won't have companies selling proprietary solutions; that will still happen, though proprietary solutions will almost always be built on top of open source frameworks or operating systems. This shift means that the nature of competition in the software market will look more like that in the consulting industry, rather than the kind of competition you see in the automobile industry.

> What does the future hold for open-source software and where do you see great or greater success?

I see the server-side market as being almost saturated at this point - there will continue to be gains, but Microsoft will not lose as much ground here as in the desktop space, where I see non-US governments and businesses pushing Linux to the desktop in a massive way over the next two years, with US governments and businesses following suit afterward. I think the server side still will have some interesting changes take place - I think the biggest shift will be the tide of database installations that move from Oracle and MS SQL to MySQL and Postgres, driven in large part by the lower cost and grass-roots knowlegebase building up around these products.

> What role does open-source software play in developing countries?

I am not close enough to specific projects to feel qualified to answer, but from what I read, there is a lot of activity here, particularly amongst companies being forced to adopt US-style IP laws through their membership in the WTO or other treaty regimes; if they have to make all their software use legit, they'd rather not send large payments to Redmond, Washington. This is particularly true in China.

> Does open-source reduce the cost of IT investment in developing countries?

It appears to, if such IT investment is required to be legal and not involve pirated software.

> Open-source models

> Is the open-source development model of relevance and significance to developing countries?

Aside from the sheer cost difference, there are a multitude of reasons why the public policy of developing nations should be to favor open source software. Training a nation's braintrust on the particulars of Linux, Apache, Perl, etc., means that this training base does not stand the same risk of finding their skills tossed when a foreign software vendor decides to deprecate old software. Small countries also run the risk of not having sufficient leverage against a foreign software vendor who decides to drop support for their language and charset, or support for particular features. This is driving the Isreal army, for example, towards OpenOffice. IT is a necessary part of modern society, and being independent and self-supporting in IT helps build a solid, robust state.

> Does open-source create new opportunities for developing countries'

> IT industries? Would it encourage collaboration between developing and developed countries? If so, how?

Developing nations should develop a population of competent programmers familiar with both specific open source technologies and general principles of open source development. These programmers will not only help spread IT technology through the government and business sectors of that nation's economy, they will also make the IT industry of that country attractive to potential customers, investors, and business partners outside the country.

> Sustainability

> Is OSS sustainable long term? Why or why not?

OSS and proprietary software will always exist, balanced against each other like a chemical going through a state transition (ice vs. water, for example), where the balance is affected by the specific sector, number of skilled developers, maturity of the solution space, and other factors. OSS is "sustainable" in that it will never disappear, as the economics that bring it about are not likely to disappear. However, nor is proprietary software ever likely to disappear.

> What models do you think exist for open-source sustainability?

I think the model of services companies who work in conjunction with hobbyist and academic developers on common software projects is sound; services companies have enough incentive to maintain the strength of the collective solution to avoid stifling it by investing only in proprietary differentiation. However, there are important roles to play for academia and state-sponsored development projects in these models, to act as the reference point for more "basic" research, or to help with efforts to form standards between software, or to invest in security audits.

> What motivates developers to participate in open-source projects long term?

Many feel that it's simply a better way to write software - with greater peer review comes not only a smaller chance of bugs, but also comes the joint training and spark of innovation that you rarely get when people code alone and can only share the results of their work, not the process. This leads to more productive software developers, based on my anecdotal experiences.

Others are lucky enough to do it for these reasons *and* to get paid for it as a part of their job.

> What development and management models enable sustainability of open-source projects?

I think sustainability is more a question of whether there are enough users for a given project to find a non-trivial number of expert users willing to participate in (and lead) development. It helps if there is a company for whom this project becomes important, and said company is willing to "invest" by hiring a developer to work with the rest of the community on it, but there are many open source projects without any corporate developers working on it.

As for particular development methodologies or models, the only important ones I can think of are:

a) the project should seek to associate itself with a non-profitorganization, or start one for itself, so as to provide legal protection for its developers, and

b) the project should seek to recruit new developers on a regular basis and give them a path to become a "core developer" on peer with the other core developers. This is to allow original developers the chance to move on to other projects gracefully, without stopping the project.

> What development and management models enable high quality open-source software?

Peer review is the most important benefit of an open source approach towards quality, when successful. There should be enough developers such that each developer can handle a reasonable amount of code, and that their commits can be reviewed or improved by a couple others, or more. Often there are developers whose sole activity (by choice) becomes reviewing the work of others.

Quality is also achieved by the consensus-driven approach most projects take to making product decisions. Should we support feature X? Or should we implement an API to allow someone else to do that? What would "users" want to see anyways? Are we ready to make a release? Putting the control over those decisions in the hands of developers rather than an uncaring "management" who doesn't know the technical details is an important factor. It means the developers feel more *responsible* for what they've created - rather than nameless robots creating something behind the scenes, they are now as much an identifyable part of the resulting product as a movie star is a part of the movie. This causes developers to care much more about quality, to spend the extra time to do something right rather than fast or haphazardly.

> Business models

> What kind of business models are available for open source exploitation in developing countries?

I really don't know, mostly because my knowlege about how to bootstrap industries in developing nations is limited. I would look to examples like Grameen Bank and Grameen Phone, perhaps, who (as I understand it) have been pioneers in the space of microloans and cell phone deployments in developing nations.

I think the "purest" business model in the OS space, the support-for-hire model pioneered by Cygnus, is the most likely to work at the small scale yet be able to scale up over time.

> Policy

> What kind of policy frameworks are appropriate to enable the success of open-source in developing countries?

My opinion, purely.

a) Hold off on Internet tariffs, licensing, and control until the industry is mature enough that one understands its dynamics. As I understand it, Ghana is having problems right now with cybercafes providing Voice Over IP technology, which is seen as being contrary to the state-owned telecommunications company. Since open source only works over the Internet, policies that favor the use and spread of the Internet are essential to building a healthy open source community.

b) Give strategic weight to IT vendors or internal solutions that incorporate open source technologies, but don't mandate that all IT solutions are open source.

c) Create a "CTO" for Open Source technologies in government who reports at a very high level. Have this person be responsible for staying on top of the use of OSS in the government, establishing standards, encouraging different groups to work together on common solutions, etc. Such a person may communicate very frequently with those outside the country as well.

d) Have public education facilities that teach technologies focus on open source technologies as much as possible. That is, when someone takes a class in Operating Systems, have them learn about how they work by studying Linux.

e) Provide some funding, not a huge amount but enough for a dozen techies plus a manager or two, whose job it is to ensure that the popular OSS software in the world is localized for the language of the developing nation. Most OSS these days have a good story with respects to support for alternative charsets and encodings, but simply need the manpower to provide and maintain the translation bundles. Doing this legwork helps ensure the OSS technologies remain available for the citizens of this nation.

E.3 Michael Roberts, BellaNet

> Open-source & software market

> Does open-source software make the software market in general more competitive or does it distort the market in some way?

I would say it makes the market more competitive since ultimately it offers more choices to the consumer who might want to buy a service over software or not. For example, a government who wants a piece of software in their local language can often be hard to find. However Open Source drives this type of local entrepreneurship. It also forces the proprietary market to make even better software in order to justify an organization to pay for it. There is definitely room for both models and both have a place.

In some ways its changing the market as well. Open Source once adopted helps to foster a local market economy. People pay for services around the software, the support, the customization, the local language translation rather than unilingual unchangeable software where profits are often concentrated in one location.

> What does the future hold for open-source software and where do you see great or greater success?

I think the software will eventually become popular in government esp. in developing countries. This will start to help create a local software industry. In parallel training centres will need to be encouraged, but the need must be there first.

> What role does open-source software play in developing countries?

- encourages the creation of a local economy through hands on support
- local language support – not just English, French etc.
- less dependence on technology that is expensive to maintain should copyright laws become enforced

> Does open-source reduce the cost of IT investment in developing countries?

No.

> Open-source models

> Is the open-source development model of relevance and significance to developing countries?

> Does open-source create new opportunities for developing countries'

> IT industries? Would it encourage collaboration between developing and developed countries? If so, how?

Yes new opportunities for people to engage the software industry locally through hands on training, support, translation, installation. Open Source often demands these services as much as or more than proprietary software. However the money generally stays within the local economy.

Definitely. Often Open Source projects span several continents, north-south collaboration is far easier today thanks to things like the internet.

> Sustainability

> Is OSS sustainable long term? Why or why not?

Yes, since it doesn't exclude companies who make Open Source software from making money. MySQL for example uses a dual license model.

> What models do you think exist for open-source sustainability?

> What motivates developers to participate in open-source projects long term?

Often self interest. They are working on something that is often of interest to them or their company.

Other enjoy it like a hobby and enjoy working on things collaboratively.

> What development and management models enable sustainability of open-source projects?

> What development and management models enable high quality open-source software?

> Business models

> What kind of business models are available for open source exploitation in developing countries?

> Policy

> What kind of policy frameworks are appropriate to enable the success of open-source in developing countries?

Appendix F. Paper on "Role of Open-Source in e-Sri Lanka"

This is a re-print of [Weer 2003].

The Role of Open-Source in e-Sri Lanka

Sanjiva Weerawarana and Jivaka Weeratunga
Lanka Software Foundation
http://www.opensource.lk

Extended abstract

Abstract

This paper considers the role of open-source in e-Sri Lanka. Open-source is a software development approach which enables and encourages developers from anywhere to participate. Software produced in this fashion is now dominating many areas of global computing, including operating systems and Web infrastructure.

Open-source presents an opportunity to revolutionize the IT industry in Sri Lanka and to be a force in other e-Sri Lanka initiatives. We briefly discuss some of these and then discuss the Lanka Software Foundation, a non-profit organization dedicated to furthering open-source in Sri Lanka.

Introduction

Developing software is yet a very expensive proposition. While software engineering and computer science have made great strides in its fifty-year history, it is yet a labor intensive and highly error prone task. At the same time, software plays a bigger and bigger role in today's society. As more and more of life-critical functionality becomes computer driven, it is imperative that that software be 100% reliable.

At the same time, fewer and fewer people are willing to pay top dollar for "infrastructure" type of software. Today, customers who buy a computer expect to receive, at no cost, the operating environment for it, networking (Web) infrastructure for it, as well as typically some amount of user-level application software. Given the widespread use of such software, it is critical for them to be extremely reliable. At the same time, customers are not willing to pay for such software. This is a contradiction given the reality of the high cost of software development.

Developing software and making it a commercial success in this kind of environment requires radically new development models. The open-source revolution is stepping up to address this requirement.

Open-source Background

Open-source is an evolution from the long standing practice of providing the source for software. This approach has been the popular distribution model for key Internet infrastructure software systems such as BIND (the widely used implementation of the Domain Name System) and Sendmail (the widely used electronic mail software) for probably 20 years. System administrators often customize/localize these systems prior to deployment and having the source readily available made it possible. However, the original software itself was typically developed by one person/group in one location.

The novel concept of open-source software is the notion of community development. With the popularization of the Internet in the late 90s, it became feasible for not just one person or a team in one geographical location, but groups of interested persons in geographically dispersed locations to jointly develop software. In general, any person can participate in any project. However, the degree of control or direction setting authority a participant gets is a function of how active that person has been in that project and how long they have participated.

An important aspect of the open-source culture is that participants are individuals and not organizations. Open-source developers are often individuals who have a different "day job" but yet contribute to open-source projects for reasons of self-satisfaction. In today's software development world, a majority of developers participate in very "high level" software development which is targeted at specific products or user groups. Participating in open-source projects allows developers to have much deeper and fundamental impact on the computing platform of the future than by building higher level software.

The success (or failure) of an open-source project is almost always characterized by the quality of the developer community around that project. As developer communities become more active, diverse and lively, the more likely that the project would succeed. For this reason alone, one of the most successful open-source software organizations of the world, the Apache Software Foundation, uses community strength as the litmus test on whether to embark on a project.

Typically, open-source projects have a relatively small number (often less than ten) of key thought leaders and architects that engineer the overall system and develop the majority of the code. A healthy open-source project will see new developers coming along, being active for some time and then "retiring" from that project or moving on to other projects. Thus, the culture encourages community churn and in fact thrives on it; the "new blood" in the project often brings the adrenaline needed to make drastic changes periodically.

Impact

The relative chaos of the open-source culture may lead one to assume that software developed under this approach will always remain amongst

the "hacker" community. However, the reality is exactly the opposite: Today a majority of Internet infrastructure is based on open-source products: including Sendmail, BIND, Linux and Apache HTTPD.

Apache HTTPD, the Web server developed by the Apache Software Foundation now runs more than 63% of all active Web sites (representing more than 25 million sites). The Microsoft Web Server (IIS) comes in a distant second (at a bit below 27%) [1]. No other vendor's Web server has any significant market share. As a further testament to the success of Apache HTTPD, several major software vendors has dropped internal Web server products and adopted the Apache HTTPD software for use in their higher level software products.

The same success story is being repeated in different parts of the Web computing infrastructure. Linux is fast becoming the industry standard Unix platform, rapidly replacing many proprietary Unices. Apache Jakarta project's Tomcat servlet engine [2] and its various components are starting to form the basis of software vendors' higher level products. It is also happening on the base programming model infrastructure of Java – many of the libraries used with XML for example are all open-source software packages.

These latter example projects are currently wide open for active participation and leadership. These projects represent key parts of the J2EE middleware platform, which is fast becoming the leading enterprise scale middleware platform. Thus, active participation in the open-source projects related to J2EE translates to an opportunity to impact the entire J2EE community. In addition, Web services, which is currently the "hot" concept in the IT industry, represents a major opportunity for open-source participation and leadership as those projects are still in their early stages.

It is important to note that open-source products have yet to capture significant market share in the Windows platform. This is of course greatly due to the single vendor nature of that platform. However, with the advent of the.Net platform, it is expected that open-source style implementations of various key components of the software stack will gain greater mind- and market-share.

Role Of Open-source

Open-source can play several roles in the development of e-Sri Lanka. Most importantly, it presents an opportunity for Sri Lanka's IT industry to become a driving force in the global computing industry. On a different level, the usage of open-source software in societal and educational applications can greatly reduce the cost of large-scale deployments of software. Similarly, open-source has a significant role to play in e-government issues.

In the rest of this section we examine these roles briefly.

IT Industry Development. It is clear that software developed under the open-source paradigm is slowly but surely taking over the lower to middle layers of the software stack. It is also interesting to note that most of the developers who participate in open-source projects are US-based with a few European-based. There are very, very few participants from Asian and other countries [3].

This represents a significant opportunity. Given the relatively small number of developers actively participating in each project, Sri Lanka can become *the* source of top open-source developers and leaders. What matters in this environment is not size (in terms of the number of software engineers). Instead, what matters is that there be a dedicated, highly skilled and fully committed team of developers to contribute to and lead key open-source projects.

Why should Sri Lanka become the leader in open-source software development? First, given the impact of open-source software, having leadership and direction setting role in that world will provide us with the opportunity to become key players in the foundation of the software industry. Given the size of our IT industry and the level of investment in it, it is very unlikely that Sri Lanka's IT industry can become a major player in the software industry without this type of an approach. Architectural and development leadership will translate to being able to exploit the business opportunities around open-source projects. Furthermore, it will provide a large pool of highly skilled, best-in-the-world quality software architects and engineers to the Sri Lankan software market. These people will in turn spur growth of existing and new software organizations in Sri Lanka. The result will be more participation in taking ownership and leadership of more and more projects, resulting in a very healthy cycle of top-of-the-breed software developers and software organizations with global competitiveness.

Thus, open-source can revolutionize the Sri Lankan IT industry.

Societal and Educational Applications. As computing and computers get more widely deployed, the cost of infrastructure software such as the operating system and office productivity software will become more and more of a burden. While the current solution in Sri Lanka is of pirating of software, that is not legal, moral nor viable in a large public investment scenarios.

Open-source software is slowly but surely making inroads into these domains. There are already many open-source office productivity packages that provide functionality similar to many commercial packages. Similarly, in the education space there are such packages available.

Open-source changes the game in these domains by allowing a country like Sri Lanka to not just be a consumer of products from elsewhere, but also define the products for this space. That is, if instead of simply deploying commercial products for such scenarios, we build our own or extend already available open-source products, we not only get less expensive, more customized solutions for the immediate problem, but also become a source of solutions for the entire world

e-Government. This is the most talked-about national scale computing project in nearly every country now. However, it is clear that no one has any complete architecture or solution for the e-government space yet. Open-source is undoubtedly an excellent vehicle for developing e-government infrastructure & solutions as wide deployments of such software will be needed and the cost of using commercial software will be prohibitive.

Capitalizing with LSF

The Lanka Software Foundation is a non-profit organization formed to help Sri Lanka exploit the opportunities presented by the open-source world. The mission and objectives of the foundation include the following:

- To support open source software developers and projects with infrastructure, funding, motivation, research, development, consultancy, training and other enabling facilities.
- To create an identity for the open source software developer community and provide for worldwide interaction, co-operation and co-ordination of open source software developers.
- To liaise with external agencies and act as a medium to obtain and receive resources, facilities and funds for the development and enhancement of knowledge, education and research with respect to open source software and related fields.
- To organize and promote seminars, workshops, conferences and exhibitions for the purpose of training and dissemination of knowledge.

LSF is building an open-source network consisting of developers and their employers, universities, professional organizations and university-hosted development labs to fully exploit open-source in Sri Lanka.

Conclusions

In this paper we have discussed the role of open-source in realizing the visions of e-Sri Lanka. The opportunity offered by this paradigm of software development must not be taken lightly. The Lanka Software Foundation is working with the IT industry and other relevant communities such as academia and government to deliver on the promises of open-source.

References

1) Netcraft survey, http://www.netcraft.com/
2) http://jakarta.apache.org/tomcat
3) Sam Ruby, Apache Foundation member, private communication

Appendix G. E-GovOS.org Guidelines for Viable Government Open Source Policy

The Center of Open Source & Government Endorses South African Open Source Strategy and Provides Policy Guidelines

The Center of Open Source & Government endorses the South African Proposed Strategy for Using Open Source Software in the South African Government by providing rationally defensible policy guidelines. The South African Strategy [Sout 2003] is a reasonable road map for a viable Open Source Government Policy for the following reasons:

1. Official Statement of Recognition of the Legitimacy of Open Source

Since Open Source as a policy consideration is a relatively new phenomenon, a proper Open Source Government Policy begins with the explicit recognition of Open Source as a legitimate software development methodology.

The South African Strategy explicitly states that Open Source software is a legitimate alternative to proprietary software in government systems.

2. Designation of Particular Government Agency to Lead Open Source Program

A proper Open Source Government Policy provides authority and accountability to a high level government official who is responsible for coordination, communication and execution of the Program.

The South African Strategy designates the State IT Agency (SITA) to provide leadership and support for government institutions wishing to implement Open Source Software. The strategy makes provision for briefing sessions to the public and to government agencies, publishing information through appropriate media outlets, creating and maintaining an Open Source government website, and making presentations at conferences.

3. Level Playing Field in Government Procurement

A government IT policy should foster a commitment to competition without, directly or indirectly, pre-determining winners, as a result there should be no a priori procurement preferences. However, a neutral government procurement policy first ensures that all de facto and de jure preferences, prejudices and discriminations are removed. Once the playing field is leveled, purchases should be made on technical merit, giving both proprietary and Open Source software an equal opportunity to be selected.

One prerequisite for a level playing field is the removal of existing user lock-ins resulting from de facto proprietary standards in use in the current IT environment. As such, governments should immediately mandate that only products abiding by enforceable Open Standards and Open Protocols be purchased.

Until all major IT products are produced in compliance with Open Standards and Open Protocols, it is reasonably defensible for governments to have policies that deviate to a limited degree from a procurement policy that would look exclusively at technical merit. The preferred mitigating policy is charging proprietary companies a 5-10% non-compliance fee on all purchased products that are non complying, with the proceeds used to fund Open Source education and software development in the country. A less preferred policy, though still reasonably defensible, is a pre-set set-aside of between 10–20% of the IT procurement budget that will be used to procure Open Source products, even though strict adherence to a policy of purchasing on technical merit would suggest that proprietary products be purchased. The rationale for the de jure disparate treatment between proprietary and Open Source during this transitional period is the fact that a de facto disparate treatment currently exists that benefits proprietary and needs to be counter-balanced until Open Standards are implemented.

The South African Strategy requires that Open Standards are to be a prerequisite for all software development. It also says that discrimination and prejudice will be avoided in software procurement procedures to give Open Source and proprietary equal opportunity to be selected. However, the South African Strategy does not attempt to implement an interim policy to counter-balance the current slanting of the playing field towards proprietary until the policy goal of compliance with Open Standards is functional in the market.

4. Appreciation of Social Value of Open Source Software

While government procurement policy should be neutral to ensure that governments do not introduce market distortions into the world economy, there should be an appreciation of the social benefits of fostering Open Source software development in a proper Open Source Government Policy plan.

These social benefits include wider access to government information by citizens, transparency in the functioning of the software running e-government services, ability to create an indigenous software industry, and better education and training of local IT professionals. These are substantial social benefits for a country that are unavailable at all or to

the same degree from the proprietary software industry.

It should be noted that some commentators maintain that the social benefits for a country of Open Source software are so large that they should always trump the no a priori preferences principle. The Center of Open Source & Government cautions governments from placing the economic principles and social principles in opposition. With Open Source software the economic and social principles are not in competition and both are attainable simultaneously. Placing the social benefits above the requirement of a level playing field unnecessarily complicates public policy and can easily introduce unwanted market distortions with negative unintended consequences.

However, if it were proven that a level playing field were to result in an enduring disadvantage to a country's social policies, it would be necessary to re-visit the issue of a proper balance between the economic and social policies. The Center of Open Source & Government currently holds, until shown otherwise, that the economic and social policies supplement and support each other, so artificially placing them in opposition is needless.

The South African Strategy recognizes the educational and commercial benefits of Open Source development and recommends that partnerships between academic, industry and government institutions be implemented. It also finds a clever balance between a neutral procurement policy and the social benefits of Open Source suggesting that in circumstances where the advantages and disadvantages of Open Source and proprietary software are equally strong, opting for Open Source is preferable, which is sensible.

5. Phased Implementation

Given the disruptions and uncertainties inherent in moving from a proprietary framework to an Open Source one, phasing in an Open Source policy program is prudent to allow for piloting, education & training, capacity building, experimentation and experience learning.

The South African Strategy begins with a Neutral Approach and progresses through an Enabling Phase to an Aggressive Approach.

Conclusion

For the reasons given, The Center of Open Source & Government endorses the South African Policy Strategy.

Appendix H. License Fees and GDP Per Capita

Total cost of ownership (TCO) is often used as a measure of software cost. However, this appendix shows that TCO computations defer greatly for developed and developing countries.

When labor costs are high, as it is in developed countries, the initial cost (in the form of the license fee) of software is a small fraction of the TCO. However, when labor costs are low, as it is in developing countries, the license fee can easily be the dominating factor of the TCO. OSS of course typically has little or no license cost compared to proprietary software.

"This relationship is neatly demonstrated by comparing license fees with a country's GDP per capita (i.e. the average individual income). As is quickly apparent, in developing countries, even after software price discounts, the price tag for proprietary software is enormous in purchasing power terms. The price of a typical, basic proprietary toolset required for any ICT infrastructure, Windows XP together with Office XP, is US$560 in the U.S. This is over 2.5 months of GDP/capita in South Africa and over 16 months of GDP/capita in Vietnam. This is the equivalent of charging a single-user license fee in the U.S. of US$7,541 and US$48,011 respectively, which is clearly unaffordable. Moreover, no likely discount would significantly reduce this cost, and in any case the simple fact that a single vendor controls any single proprietary software application means that there can never be a guarantee that any discount offered is intended to be sustained for the long term, rather than as a temporary measure used to tempt consumers into a lock-in situation at which point in time the discount can be reduced." [Ghos 2003]

"This simple calculation is presented below in the table for 176 countries, together with 10 geographical and political aggregates. The table also includes the piracy figures published by the Business Software Alliance (BSA). It should be noted that there is a correlation between the piracy rate and the effective software licence fee — the more expensive software is, the higher the piracy rate. This is common sense, but does not seem to be reflected in the BSA estimates of the "losses" to the software industry based on piracy, which assume that all the estimated unlicensed copies of software in a country should (or could) be replaced with paid licensed copies. Ironically, the logical conclusion of the increas-

ingly stringent international campaign for strong enforcement of copyright is the reduction of piracy rates not through the take-up of licensed, proprietary software, but through the use of open source software. Anecdotal evidence shows this is the case in Argentina, Peru and other countries especially in Latin America, where a campaign for strong copyright enforcement has coincided with poor economic conditions." [Ghos 2003]

Table H-1: Comparison of per-capita GDP and Windows XP license fees. [Ghos 2003]

Country	GDP/cap	PCs ('000s)	Piracy	WinXP Cost [3]	
				Effective $	*GDP months*
Albania	1300	24	n.a.	*15196*	*5.17*
Algeria	1773	220	n.a.	*11140*	*3.79*
Angola	701	17	n.a.	*28184*	*9.59*
Antigua and Barbuda	9961	n.a.	n.a.	*1983*	*0.67*
Argentina	7166	3415	62%	*2757*	*0.94*
Armenia	686	24	n.a.	*28806*	*9.80*
Australia	19019	10000	27%	*1039*	*0.35*
Austria	23186	2727	33%	*852*	*0.29*
Azerbaijan	688	n.a.	n.a.	*28708*	*9.77*
Bahrain	12189	92	77%	*1621*	*0.55*
Bangladesh	350	254	n.a.	*56401*	*19.19*
Barbados	10281	25	n.a.	*1921*	*0.65*
Belarus	1226	n.a.	n.a.	*16120*	*5.48*
Belgium	22323	2394	n.a.	*885*	*0.30*
Belize	3258	33	n.a.	*6064*	*2.06*
Benin	368	11	n.a.	*53613*	*18.24*
Bhutan	644	5	n.a.	*30668*	*10.43*
Bolivia	936	175	77%	*21109*	*7.18*
Bosnia and Herzegovina	1175	n.a.	n.a.	*16818*	*5.72*
Botswana	3066	66	n.a.	*6444*	*2.19*
Brazil	2915	10835	56%	*6777*	*2.31*
Bulgaria	1713	n.a.	75%	*11534*	*3.92*
Burkina Faso	215	17	n.a.	*91801*	*31.23*
Burundi	99	n.a.	n.a.	*198864*	*67.65*
Cambodia	278	18	n.a.	*71184*	*24.21*
Cameroon	559	60	n.a.	*35319*	*12.01*
Canada	22343	14294	38%	*884*	*0.30*
Cape Verde	1317	31	n.a.	*14998*	*5.10*
Central African Republic	257	7	n.a.	*76998*	*26.19*
Chad	202	12	n.a.	*97728*	*33.24*
Chile	4314	1640	51%	*4579*	*1.56*

China	911	24222	92%	*21678*	*7.37*
Colombia	1915	1810	52%	*10316*	*3.51*
Comoros	386	3	n.a.	*51208*	*17.42*
Congo, Dem. Rep.	99	n.a.	n.a.	*199394*	*67.83*
Congo, Rep.	886	12	n.a.	*22288*	*7.58*
Costa Rica	4159	659	64%	*4750*	*1.62*
Cote d'Ivoire	634	118	n.a.	*31140*	*10.59*
Croatia	4625	376	67%	*4272*	*1.45*
Cyprus	12004	188	61%	*1646*	*0.56*
Czech Republic	5554	1490	43%	*3557*	*1.21*
Denmark	30144	2896	26%	*655*	*0.22*
Djibouti	894	7	n.a.	*22107*	*7.52*
Turkey	2155	2792	58%	*9167*	*3.12*
Turkmenistan	1097	n.a.	n.a.	*18010*	*6.13*
Uganda	249	71	n.a.	*79324*	*26.98*
Ukraine	766	898	86%	*25802*	*8.78*
United Kingdom	24219	21533	25%	*816*	*0.28*
United States	35277	178326	25%	*560*	*0.19*
Uruguay	5554	370	63%	*3557*	*1.21*
Uzbekistan	450	n.a.	n.a.	*43943*	*14.95*
Vanuatu	1058	n.a.	n.a.	*18677*	*6.35*
Venezuela, RB	5073	1300	55%	*3895*	*1.32*
Vietnam	411	933	94%	*48011*	*16.33*
West Bank and Gaza	1286	n.a.	n.a.	*15366*	*5.23*
Yemen, Rep.	514	35	n.a.	*38434*	*13.07*
Yugoslavia, Fed. Rep.	1020	249	n.a.	*19373*	*6.59*
Zambia	354	72	n.a.	*55824*	*18.99*
Zimbabwe	706	155	68%	*27965*	*9.51*
Regional Aggregates 4					
European Union	20863	116997	n.a.	*947*	*0.32*
EU Accession countries	4840	8286	n.a.	*4082*	*1.39*
EU applicant countries	2023	3592	n.a.	*9766*	*3.32*
The Caribbean	4560	308	n.a.	*4332*	*1.47*
Latin America	4335	18703	n.a.	*4557*	*1.55*
Africa	652	7636	n.a.	*30297*	*10.31*
Middle East	2679	9708	n.a.	*7375*	*2.51*
Asia	2128	102229	n.a.	*9282*	*3.16*
Oceania	13946	11886	n.a.	*1417*	*0.48*

Appendix I. Important OSS Information Sources

In this section we list important open source information sources to help the interested reader explore open source further.

I.1 Books

- Joseph Feller and Brian Fitzgerald, *Understanding Open Source Software Development*, Addison-Wesley, 2002.
- Eric S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilley, 2001.
- Chris DiBona, Sam Ockman and Mark Stone (Editors), *Open Sources: Voices from the Open Source Revolution*, O'Reilley, 1999.
- Martin Fink, *The Business and Economics of Linux and Open Source*, Prentice Hall, 2002.

I.2 Web sites

- The Open Source Initiative, http://www.opensource.org/OSI is a non-profit organization formed to further fundamental principles of open source.
- The Free Software Foundation, http://www.fsf.org/FSF advocates a different model of freedom in open source than OSI.
- The Apache Software Foundation, http://www.apache.org/
  A non-profit organization which hosts numerous open source software projects, most notably the market leading Apache Web server.
- Open Office, http://www.openoffice.org
  Popular OSS alternative to Microsoft Office, available on all platforms including Windows.
- SourceForge, http://sourceforge.net/
  SourceForge hosts nearly 75,000 open source projects as of January 18, 2004 with nearly 775,000 registered users.
- MIT Open Source Research Community, http:/opensource.mit.edu/
  A research focused open source community referring to various OSS information sources.
- Slashdot, http://www.slashdot.org/
  A popular developer portal for the open source community.

- O'Reilley publishers, http://www.oreilley.org
  An publisher of top quality books on open source (and other) topics.
- Open Source Developer's Network (OSDN), http://www.osdn.com/
  Very popular technical site attracting a large readership.
- Freshmeat, http://freshmeat.org/
  Provides a very large index of OSS software.
- Linux, http://www.linux.org/
  Popular Linux portal.

I.3 Articles on economics of open source

- David Lancashire, "The Fading Altruism of Open Source Development", First Monday, Vol. 6, Number 12 (Dec. 2001), http://firstmonday.org/issues/issue6_12/lancashire/index.html
- "Microsoft at the Power Point", http://www.economist.com/business/displayStory.cfm?story_id=2054746
- Steven Weber, "Open Source Software in Developing Economies", http://www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf

Appendix J. Terms of Reference

Terms of Reference for Commissioned Report on Open Source for Developing Countries
Per Einar Tröften 2003-11-11

Introduction

Sida has a strategy since 1999 for integrating ICT in Development Cooperation. The internal advisory group assigned to lead this integration connects both to Sida Departments and to Swedish Embassies in cooperating countries. The advisory group has divided its main tasks in two branches:

1. ICT as a tool for development. To facilitate the use of ICT within development co-operation areas and programmes conducted by Sida's sector departments;
2. Develop ICT in co-operating countries. To promote the development of ICT through:
   a. Support to national ICT policies, strategies and plans
   b. Support to Human Capacity Development and Education for ICT
   c. Support to physical ICT-infrastructures

Background

Most developing countries are struggling with elaboration of strategies in a context of low governmental budgets for ICT in general. The same countries are now challenged with the requirement to bridge the digital divide by rapidly increasing the level of ICT. This calls for new solutions and fresh ideas when it comes to hardware/software, systems, policies, planning and financing. New actors, often private entrepreneurs, enter the scene and probably adapt to the economic situation prevailing in developing countries.

The ICT advisory group has commissioned a series of Country ICT Surveys to illuminate the situation in developing countries regarding ICT. Infrastructure, hardware/software, human resources, education and training have been part of these studies, but mainly on a descriptive level. There is a need for more elaborated description and analysis of the various issues pertaining to raising knowledge on ICT in these countries.

Sweden, through Sida, also participates in the UN ICT Task Force and supports the participation from developing countries in the World Summit on the Information Society (WSIS) 2003.

Purpose and Scope

Sida has decided to commission Sanjiva Weerawarana to perform a study on open source for developing countries.

With this study Sida wants to shed light on:

(a) open source in general;

(b) open source in developing countries; and

(c) should donors support open source and – if yes – how can donors support open source in developing countries.

The report should be possible to use as a tool for Sida's policy discussion concerning future support to open source initiatives.

The report should in principal cover issues such as: what is open source; list/map the most important open source "players" (i.e. organisations, individuals, systems/applications, etc); compare open source with already established commercial products (i.e. list the pros & cons); should donors support open source and – if yes – how can donors support open source in an effective and catalytic manner without disrupting the "market laws" (i.e. describe the interface were donors and open source initiatives can interact); and issue recommendations for Sida in particular if – and how – Sida should support open source initiatives from developing countries.

The report should also deal with issues such as: relevance of open source for developing countries and donors; do developing countries have the adequate qualifications making sure they can benefit and maintain solutions/systems relying upon open source; have any open source initiatives in developing countries succeeded; is a strategic choice of open source also good for a medium to long term perspective – and not just the short term benefit primarily due to reduction of cost (i.e. is it really the right strategic choice and in what cases/systems/applications is open source the right long term strategic choice), and input/feed back for developing countries how to deal with open source (yes or no for developing countries to open source, and – if yes – describe how to avoid the classic "pit falls/killers").

The consultant shall attempt to make some kind of comparison/reference of the costs of similar programs/solutions and list pros/cons of going for open source vs commercial software in developing countries. In such a comparison, however, the evaluator will also have to take into account the special limitations/complications in developing country and the special needs that are different from those in industrialised countries. Thus, the consultant will have to conclude if it is worth the time and money to go for open source in developing countries; describe which segments of open source solutions are most promising to embark upon in developing countries; is it only within the large volumes like "office" software where developing countries could benefit from open source (as

volume equals the largest cost); and can/should big consumers like the governments in developing countries influence the choice of open source (e.g. Vietnam & Munich in Germany).

The consultant has also suggested examining intellectual property issues; open source license options and their implications as well as models of "market-friendly" open-source exploitation especially for developing countries. The consultant can also touch upon the potential for some of the developing countries to develop a critical mass of expertise in certain niche areas of open source (i.e. can developing countries become strong developers of products based upon open source) as mentioned in the Lanka Software Foundation paper in August 2003. Important is that the evaluator is free to include other important aspects as well.

It is important that the consultant take into consideration that Sida primarily support the Least Developed Countries. Some of the conclusions and recommendations in the report may be relevant for all developing countries, while others may have to be for Least Developed Countries in particular.

Please notice that it should not be a report describing Sri Lanka and open source. However, reference and illustrative examples from Sri Lanka may of course be included as from any other country.

The Assignment

The commissioned report should include three specific issues:

(a) open source in general;

(b) open source in developing countries; and

(c) should donors support open source and – if yes – how can donors support open source in developing countries.

The report should represent a qualitative expert report, but also give feed back for Sidas discussions concerning open source. The report shall, however, also be possible to use as a high quality reference paper in the consideration of embarking upon open source now taking place in many developing countries (including the Least Developed Countries).

The consultant shall develop a model for description, analysis and discussions about open source in developing countries. The model should be possible to use in discussions with counterparts in developing countries and for evaluation of proposals for support to open source initiatives. The consultant shall discuss how Sida can support open source in developing countries as a complement to Sida's ongoing more traditional ICT programmes.

The study should be conducted during November 2003 to January 2004 with a draft report 1 January and a final report presented not later than 15 of January 2004. The consultant should also visit Sida to present the report and meet open source people in Sweden, preferably during late January 2004.

Sanjiva Weerawarana has experience as regards research and development of open source. However, to accomplish this mission the consultant needs to involve special expertise on certain aspects of the open source sector and its suitable interface for donor support.

Methodology

The consultant need to review available publications, reports, evaluation and other information that is relevant for the study. A number of interviews with selected key persons should be used to clarify and raise important issues. See list below.

Sida also encourage the consultant to submit an academic article, based on the report, to a reviewed academic publication. However, it is very important that the writing up of such an academic paper should take place after the report have been published and the consultants visit to Sweden. However, the consultant is free to utilise parts of the Sida funding for this purpose up to the date of expiry of the contract.

The study should be done in totally 6 weeks, allowing for one week of preparations, one week of field studies and four week of compilation.

Reporting

A draft report should be presented not later than January 1, 2004.

A final report should be ready not later than January 15, 2004.

A financial report should be submitted to Sida not later than June 30, 2004.

Necessary interviews and e-mail contacts:

- Relevant open source organisations.
- Relevant open source individuals.
- Bellanet, Michael Roberts is dealing with open source issues (http://home.bellanet.org/).
- IT university in Sweden, Bjorn Pehrsson & Alberto have dealt with open source and Love Ekenberg & Lars Glimbert have done alot of ICT work in DCs.
- Namibia Schoolnet & Schoolnet Africa, Joris & Shafika & Denis may know the right open source people in Africa.
- UCSC in Sri Lanka, professor Samaranayake & Ruvan Weerasinghe have received quite a bit of "traditional" Sida ICT support and may help you understand how Sida think/work.

Additional information:

- Vietnamese authorities have ruled that all government desktop computers must run open-source operating systems by 2005 (http://news.zdnet.co.uk/business/0,39020645,39117548,00.htm). Also (http://news.osdir.com/article315.html)
- Munich Goes with Open Source Software 2003 (http://linuxpr.com/releases/5893.html). Also (http://www.eweek.com/article2/0,3959,1110809,00.asp)
- Chinese government plans to finance Linux-based computer systems! (http://www.cnn.com/2003/TECH/biztech/11/05/china.linux.reut/)
- Applied Linux Institute, Helsingfors university, Finland, promoting FLOSS across the globe, specifically in developing countries. Seppo Koskela och Sinikka Sassi. See (http://www.linuxjournal.com/article.php?sid=7110)

Acknowledgements

We would like to thank all those individuals who provided assistance in various ways throughout this study. In particular the assistance of Linus Torvalds, Brian Behlendorf and Prof. V.K. Samaranayake is gratefully acknowledged.

Halving poverty by 2015 is one of the greatest challenges of our time, requiring cooperation and sustainability. The partner countries are responsible for their own development. Sida provides resources and develops knowledge and expertise, making the world a richer place.

SWEDISH INTERNATIONAL
DEVELOPMENT COOPERATION AGENCY

SE-105 25 Stockholm Sweden
Phone: +46 (0)8 698 50 00
Fax: +46 (0)8 698 56 15
info@sida.se, www.sida.se