Study on the:

Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU

Final report

Prepared on November 20, 2006

Lead contractor: UNU-MERIT, the Netherlands

Subcontractors:

Universidad Rey Juan Carlos, Spain

University of Limerick, Ireland

Society for Public Information Spaces, France

Business Innovation Centre of Alto Adige-Südtirol, Italy

Prepared by: Rishab Aiyer Ghosh, MERIT

Disclaimer

The opinions expressed in this Study are those of the authors and do not necessarily reflect the views of the European Commission. Contract ENTR/04/112.

Authors and contributors

Parts of this report were written by, or are based on inputs from:

Philippe Aigrain, SOPINSPACE
Roberto Andradas, URJC
Raphaël Badin, SOPINSPACE,
Renaud Bernard, SOPINSPACE
Luis Cañas Díaz, URJC
Paul David, UNU-MERIT
Santiago Dueñas, URJC
Theo Dunnewijk, UNU-MERIT
Rishab Aiyer Ghosh, UNU-MERIT
Ruediger Glott, UNU-MERIT
Jesus Gonzalez-Barahona, URJC
Kirsten Haaland, UNU-MERIT
Bronwyn Hall, UNU-MERIT
Wendy Hansen, UNU-MERIT
Juan Jose Amor, URJC
Huub Meijers, UNU-MERIT
Alvaro Navarro, URJC
Francesco Rentocchini, UNU-MERIT
Gregorio Robles, URJC
Barbara Russo, BICST
Giancarlo Succi, BICST
Adriaan van Zon, UNU-MERIT

Further inputs were drawn from intermediate results of FLOSSWorld and other projects coordinated by MERIT.

Copyright

Disclaimer

Table of contents

List of figures

List of Tables

1. Executive summary: Key findings

The information economy is a large market. Including the provision of infrastructure and services for the creation, exchange and processing of information and communication services as well as the sales of information itself, this market is now in the range of 10% of GDP in most developed countries, and accounts for more than half of their economic growth. Software is one of the key elements driving ICTs' role in the economy, and the structure, competitiveness, performance of the ICT industry has potential to be strongly affected by Free/Libre/Open Source Software (FLOSS[1]). Financed by the European Commission's Directorate General for Enterprise and Industry, a study has been carried out by a team led by UNU-MERIT, the Netherlands, to identify the role of FLOSS in the economy, its direct impact on the ICT sector, its indirect impact on ICT-related sectors and to recommend policies based on forecasted scenarios.

This three-page executive summary highlights the key findings and recommendations, according to these four categories.

FLOSS role in the economy: market share and geography

- FLOSS applications are first, second or third-rung products in terms of market share in several markets, including web servers, server operating systems, desktop operating systems, web browsers, databases, e-mail and other ICT infrastructure systems. FLOSS market share higher in Europe than in the US for operating systems and PCs, followed by Asia. These market shares have seen considerable growth in the past five years.
- FLOSS market penetration is also high – a large share of private and public organisations report some use of FLOSS in most application domains. In the public sector, Europe has particularly high penetration, perhaps soon to be overtaken by Asia and Latin America. In the private sector, FLOSS adoption is driven by medium- and large-sized firms.
- Almost two-thirds of FLOSS software is still written by individuals; firms contribute about 15% and other institutions another 20%.
- Europe is the leading region in terms of *globally collaborating* FLOSS software developers, and leads in terms of global project leaders, followed closely by North America (interestingly, more in the East Coast than the West). Asia and Latin America

[1] In this report we refer to the single phenomenon known by the various terms "libre software", "free software" and "open source software" as Free/Libre/Open Source Software (or FLOSS). We note that the EU/FP5 FLOSS developer survey of over 2800 respondents showed that a majority of developers themselves identify with the term "free software", while Libre software (logiciel libre, software libre, software libero) is the favoured term in southern Europe and Latin America.

face disadvantages at least partly due to language barriers, but may have an increasing share of developers active in local communities.

- Weighted by regional PC penetration, central Europe and Scandinavia provide disproportionately high numbers of developers; weighted by average income, India is the leading provider of FLOSS developers by far, followed by China.
- While the U.S. has the edge in terms of large FLOSS-related businesses, the greater individual contribution from Europe has led to an increasing number of globally successful European FLOSS small- and medium-sized enterprises (SMEs).

Direct economic impact of FLOSS

- The existing base of quality FLOSS applications with reasonable quality control and distribution would cost firms almost Euro 12 billion to reproduce internally. This code base has been doubling every 18-24 months over the past eight years, and this growth is projected to continue for several more years.
- This existing base of FLOSS software represents a lower bound of about 131 000 real person-years of effort that has been devoted exclusively by programmers. As this is mostly by individuals not directly paid for development, it represents a significant gap in national accounts of productivity. Annualised and adjusted for growth this represents at least Euro 800 million in voluntary contribution from programmers alone each year, of which nearly half are based in Europe.
- Firms have invested an estimated Euro 1.2 billion in developing FLOSS software that is made freely available. Such firms represent in total at least 565 000 jobs and Euro 263 billion in annual revenue. Contributing firms are from several non-IT (but often ICT intensive) sectors, and tend to have much higher revenues than non-contributing firms.
- Defined broadly, FLOSS-related services could reach a 32% share of all IT services by 2010, and the FLOSS-related share of the economy could reach 4% of European GDP by 2010. FLOSS directly supports the 29% share of software that is developed in-house in the EU (43% in the U.S.), and provides the natural model for software development for the secondary software sector.
- Proprietary packaged software firms account for well below 10% of employment of software developers in the U.S., and “IT user” firms account for over 70% of software developers employed with a similar salary (and thus skill) level. This suggests a relatively low potential for cannibalisation of proprietary software jobs by FLOSS, and suggests a relatively high potential for software developer jobs to become increasingly FLOSS-related. FLOSS and proprietary software show a ratio of 30:70 (overlapping) in recent job postings indicating significant demand for FLOSS-related skills.
- By providing a skills development environment valued by employers and retaining a greater share of value addition locally, FLOSS can encourage the creation of SMEs and jobs. Given Europe’s historically lower ability to create new software businesses compared to the US, due to restricted venture capital and risk tolerance, the high share of European FLOSS developers provides a unique opportunity to create new software businesses and reach towards the Lisbon goals of making Europe the most competitive knowledge economy by 2010.

Indirect economic impact: FLOSS, innovation and growth

- Strong network effects in ICT, the related capitalization for installed dominant players, and some new forms of IPR scope extension risk leading to innovation resources being excessively allocated to defensive innovation. There is a case for a rebalancing of innovation incentives as to create a more equitable environment for innovation that targets publicly available technology for new functionality.
- FLOSS potentially saves industry over 36% in software R&D investment that can result in increased profits or be more usefully spent in further innovation.
- ICT infrastructure has a 10% share of European GDP, providing a basis for a further 2.5% share of GDP in the form of the non-ICT information content industry. However, a large and increasing share of user-generated content is not accounted for and needs to be addressed by policy makers; FLOSS increases the value of the ICT infrastructure, supporting this wider information ecosystem.
- Increased FLOSS use may provide a way for Europe to compensate for a low GDP share of ICT investment relative to the US. A growth and innovation simulation model shows that increasing the FLOSS share of software investment from 20% to 40% would lead to a 0.1% increase in annual EU GDP growth excluding benefits within the ICT industry itself – i.e. over Euro 10 billion annually.

Trends, scenarios and policy strategies

- Equitably valuing the use of FLOSS, the “true” share of software investment rises from 1.7% to 2.3% of GDP in the US by 2010, and from 1% to 1.4% of GDP in Europe. Doubling the rate of FLOSS take-up in Europe would result in a software share of investment at 1.5% of GDP, reducing but not closing this investment gap with the US.
- The notional value of Europe’s investment in FLOSS software today is Euro 22 billion (36 billion in the US) representing 20.5% of total software investment (20% in the US).
- Europe’s strengths regarding FLOSS are its strong community of active developers, small firms and secondary software industry; weaknesses include Europe’s generally low level of ICT investment and low rate of FLOSS adoption by large industry compared to the US
- FLOSS provides opportunities in Europe for new businesses, a greater role in the wider information society and a business model that suits European SMEs; FLOSS in Europe is threatened by increasing moves in some policy circles to support regulation entrenching previous business models for creative industries at the cost of allowing for new businesses and new business models.
- Europe faces three scenarios: CLOSED, where existing business models are entrenched through legal and technical regulation, favouring a passive consumer model over new businesses supporting active participation in an information society of “prosumers”; GENERIC, where current mixed policies lead to a gradual growth of FLOSS while many of the opportunities it presents are missed; VOLUNTARY, where policies and the market develop to recognise and utilise the potential of FLOSS and similar collaborative models of creativity to harness the full power of active citizens in the information society.

- Policy strategies focus mainly on correcting current policies and practices that implicitly or explicitly favour proprietary software:
    - Avoid penalising FLOSS in innovation and R&D incentives, public R&D funding and public software procurement that is currently often anti-competitive
    - Support FLOSS in pre-competitive research and standardisation
    - Avoid lifelong vendor lock-in in educational systems by teaching students skills, not specific applications; encourage participation in FLOSS-like communities
    - Encourage partnerships between large firms, SMEs and the FLOSS community
    - Provide equitable tax treatment for FLOSS creators: FLOSS software contributions can be treated as charitable donations for tax purposes. Where this is already possible, spread awareness among firms, contributors and authorities.
    - Explore how unbundling between hardware and software can lead to a more competitive market and ease forms of innovation that are not favoured by vertical integration.

2. Context

The information economy is a large market. Including the provision of infrastructure and services for the creation, exchange and processing of information and communication services as well as the sales of information itself, this market is now in the range of 10% of GDP in most developed countries, and accounts for more than half of their economic growth. Software is one of the key elements driving ICT's role in the economy, and the structure, competitiveness, performance of the ICT industry has potential to be strongly affected by Free/Libre/Open Source Software (FLOSS[2]). There is evidence that the development of the information economy has enormously benefited from the existence of the Internet and the World Wide Web, both technologies founded on FLOSS, and it appears that the emergence of FLOSS may have significant effects – whether positive or negative – on various aspects of the ICT sector and the overall economy.

The ICT goods and services sector drives economic growth and the EU's competitiveness in this sector is therefore an important element of achieving the Lisbon goals of becoming the most competitive knowledge economy by 2010. Within this context, recognising the role of DG Enterprise in understanding and monitoring the competitiveness of EU industry and the ICT sector as a driver, it is useful to better understand the impact of FLOSS on the ICT sector and Europe's industrial competitiveness.

[2] In this report we refer to the single phenomenon known by the various terms "libre software", "free software" and "open source software" as Free/Libre/Open Source Software (or FLOSS). We note that the EU/FP5 FLOSS developer survey of over 2800 respondents showed that a majority of developers themselves identify with the term "free software", while Libre software (logiciel libre, software libre, software libero) is the favoured term in southern Europe and Latin America.

3. Study Objectives

The study aimed to fill gaps in our understanding of the impact of FLOSS on innovation and competitiveness of the EU ICT sector through the use of empirical study, forecasting techniques and a variety of data sources on FLOSS, ICT, innovation and economic impact in the EU and the rest of the world. The features of the European ICT markets, and the strategic decisions of innovating European firms, the individual innovators employed by firms, individual FLOSS developers and ICT users in the public and private sector are be identified, analysed and projected into the future under a number of scenarios. Policy implications are provided along with an analysis of the economic impact of FLOSS on ICT and European competitiveness based on a sound analysis of the impact on the development of technologies and technology market dynamics.

This study provides a single-point integrative analysis of the vast amount of data available on the technical and economic impact of the intersection of FLOSS, ICT industries and the economy at large. The project team, led by MERIT, gathered much relevant data through previous and on-going projects, and some data were sourced from other public or private sources. The study provides a structured platform for integrating this wealth of evidence. Furthermore, the study makes forecasts under multiple scenarios, and provides a range of recommended policy options linked to different forecasted scenarios allowing policy makers to take decisions on the basis of sound empirical evidence, with a degree of confidence that chosen policies will likely lead to specific results.

This project was strongly grounded in high quality academic research, while remaining closely linked to industry. The draft final report was presented at the closing workshop, in Brussels, September 28, for which attendance was open to the public. The audience included academics, representatives of different parts of the software industry, policy makers, and members of the FLOSS developer community. The full report was circulated to the workshop participants as well as to all those who registered for the workshop but could not attend. Furthermore, it was circulated to members of the European ICT Task Force, the eBSN (European eBusiness Support Network for SMEs) and the European Commission's IDABC OSS experts group. The draft report attracted a lot of interest and the comments received have been, to the extent possible, taken into account in this final version of the report.

4. Structure of the project team

This project brings together expertise in the economics of innovation, econometrics, modelling and scenario planning, intellectual property and competition policy, computer science, ICT engineering and policy studies. Inputs from the various disciplines have shaped the project, in particular by ensuring the empirical foundation for the project with baseline surveys, data extraction and analysis and a consultation process with the various constituencies affected by and affecting FLOSS, ICT innovation and economic policy. Unlike many empirical studies that tend to focus on surveying firms, this study has ensured the full participation of all the different groups involved. As required by DG Enterprise, this has included large firms as well as SMEs, and the full participation of members of the free/open source community. Past surveys of different aspects are combined with huge databases of open source software, developers and projects, which are mined to provide a wealth of useful information. Carefully interpreted, this information is integrated and forms the basis of technology forecasting, economic forecasting and scenario planning. An interdisciplinary study of this sort can only be performed by a team with proven competences in:

1. Empirical study of innovation and innovative practices
2. Economic analysis especially of innovation and technology
3. Modelling and forecasting based on large data sets
4. Expertise of international (non-EU) environments
5. Technical expertise especially in-depth knowledge of the software development process and the formation of innovation in software practices
6. Technical expertise of software-related areas such as telecommunications, and the interdependence with software and computer-related innovation
7. Policy formation, recommendations and scenario planning
8. Access to various constituencies for consultation and data acquisition
9. Dissemination to various constituencies including policy-makers
10. Management of large projects with rigorous control over research quality

Such criteria cannot be met by a single organisation. This study is prepared by a consortium of proven experts in their fields that combine complementary skills, demonstrably meeting the criteria outlined above.

The project is led by MERIT at the University of Maastricht that provided the coordination and management as well as the major part of the economic research. MERIT is the lead contractor on this study, and the other partners were subcontracted by MERIT. MERIT contributes competencies to points 1-4 and 7-10. Universidad Rey Juan Carlos, Spain, is the world leader in evidence-based research on “libre software” engineering and contributes expertise and previously collected data to points 5-6, as does the Business Innovation Centre of Alto Adige-Südtirol, Italy (BICST). The Society for Public Information Spaces has contributed to points 5-6 and 7, while the University of Limerick, Ireland supported points 8-9 and organises the workshop.

5. Methodology

The design of the research methodology provides for an approach based on sound expertise in economic analysis and reliable scenario-based forecasting, proven success in large empirical economic data collection backed by high standards of academic rigour, and supported by expertise in software engineering. Interesting studies have already been done on this topic, as software technologies continue to evolve and present new challenges to policy-makers, ICT businesses and economic scholars. However, there is a serious lack of inter-disciplinary studies that are supported by quantitative measures and empirical evidence on the impact of FLOSS on ICT markets – and in turn, on innovation and economic competitiveness. In particular, few national-level studies including significant economic analysis exist[3] and no previous European or global studies provide an integrative approach to answering the question: what is the economic impact of open source?

The current study aims at providing an integrated empirical framework for evaluating the effects of FLOSS-related changes in information technologies and in the impact on ICT industry and economic competitiveness. This is accomplished by forming a list of economic, innovation and technology indicators to assess the impact of FLOSS over time, allowing for forecasting under a set of differing scenarios, based on a unique set of pre-existing and on-going databases that form perhaps the single largest set of FLOSS-related empirical data in the world.

[3] Klodt & Mundhenke's 2005 study (University of Kiel) supported by the German Ministry of Economics is theoretical rather than empirical in nature and limited to Germany.

6. FLOSS role in the economy: market share and geography

One of the tasks of this study is to describe the current picture with regards to the market share of FLOSS in the global and European markets. This task is performed through the integration of existing data sources available to MERIT, including data extraction from software projects. The following text provides an overview of FLOSS activities in the world (or in selected regions of the world) based on currently available data. Due to the heterogeneity of the data with regard to methodologies and scope of collection, the comparability of the figures presented in the following is limited. Therefore the market shares of FLOSS products in a specific region may vary between the different studies – often conducted by third parties – to which we refer. Nevertheless, the following section provides a comprehensive compilation of data on the worldwide FLOSS market shares as is currently available.

6.1.1. Drivers of FLOSS development and the market for FLOSS

FLOSS has rapidly shifted from a model driven purely by the developer community and university support to one where a main driver is industry. Not only has open source spawned a (usually) thriving environment of small focussed businesses, typically devoted to development and support of specific products or to maintenance and integration activities, it has also reshaped the business models and affected the strategies of large firms, including such major industrial players as IBM, Oracle, Philips, Nokia and SAP. The development model and licensing terms naturally provide preference to service-oriented business models where the core profit centre is not pure software development, or at any rate not software where open source has so far produced a successful alternative product. Hence the attraction of firms with a pre-existing service-oriented business model (such as IBM) or those with a niche market in software products that require considerable customisation and support (Oracle, SAP), or primarily hardware firms (Philips, Nokia). However, a number of usually smaller firms have also successfully followed a business model based on pure software sales through a process of dual licensing (GPL + proprietary) – the best known of these being MySQL, an (originally Swedish) SME that has built probably the best known open source brand in enterprise database systems today. This is further described in section 7.5.

The market for FLOSS is accelerated by the following factors:

- Availability of high-quality software
- Low cost and low barrier to entry
- Availability of customisation and local support services
- Vendor independence and flexibility

FLOSS has an impact on the traditional market for ICT in particular by expanding the total market by meeting needs of SMEs for affordable solutions, offering less-expensive or free alternatives to traditional proprietary software, and creating new business models for established and emerging provides of service and support.

6.1.2. Summary findings

- FLOSS applications are first, second or third-rung products in terms of market share in several markets, including web servers, server operating systems, desktop operating systems, web browsers, databases, e-mail and other ICT infrastructure systems. FLOSS market share for operating systems and desktops is higher in Europe than in the US, followed by Asia. These market shares have seen considerable growth in the past five years.
- FLOSS market penetration is also high – a large share of private and public organisations report some use of FLOSS in most application domains. In the public sector, Europe has particularly high penetration, perhaps soon to be overtaken by Asia and Latin America. In the private sector, FLOSS adoption is driven by medium- and large-sized firms.
- Almost two-thirds of FLOSS software is still written by individuals; firms contribute about 15% and other institutions another 20%.
- Europe is the leading region in terms of *globally collaborating* FLOSS software developers, and leads in terms of global project leaders, followed closely by North America (interestingly, more in the East Coast than the West). Asia and Latin America face disadvantages at least partly due to language barriers, but may have an increasing share of developers active in local communities.
- Weighted by regional PC penetration, central Europe and Scandinavia provide disproportionately high numbers of developers; weighted by average income, India is the leading provider of FLOSS developers by far, followed by China.
- While the US has the edge in terms of large FLOSS-related businesses, the greater individual contribution from Europe has led to an increasing number of globally successful European FLOSS SMEs.

6.2. FLOSS market share and penetration

The following sections examine available data regarding FLOSS usage (market penetration) in firms and in the public sector across different regions. This is followed by an analysis of the market share of FLOSS applications in different sector markets, such as operating systems and web servers.

6.2.1. FLOSS usage in firms

6.2.1.1. Europe

Forrester research found that European firms have been actively adopting open source software over the last two years, so that by the end of 2005 the overall share of companies using such systems amounted to 40%. Another 8% reported plans to pilot open source software systems during 2006. Utility and telecommunications firms, media companies, and public sector bodies lead enterprise adoption by a wide margin. Forty-five percent of the firms using open source have deployed it in mission-critical environments, although the vast majority (70%) uses it for non-key applications. Web server and server operating systems are the top two areas, with two-thirds of firms using alternatives like Apache, Tomcat, or Linux. Usage for application server solutions like JBoss will also see heavy piloting activity next year. When it comes to the benefits of using open source, it's all about cost: An average of 72% of European firms claim lower “total cost of ownership” and lower acquisition costs as the key advantages over commercial software.[4]

The 2002 FLOSS user survey, conducted by Berlecon Research as part of an EU-funded study led by MERIT, discriminated by sector and by size: 13.7% smaller firms (under 500 employees[5]) using FLOSS in Germany had FLOSS applications on the desktop, against only 2% of large firms (>500 employees) using FLOSS in the UK (see Figure 1). There was a high degree of diversity between the countries surveyed as well as the size classes, making it hard to come to generalised conclusions – except that, in 2002, FLOSS was less popular on the desktop than as a server operating system. More recent data shown below appears to confirm that this situation continues today.

[4] http://www.forrester.com/Research/Document/Excerpt/0,7211,38061,00.html

[5] Unfortunately the survey did not discriminate among firms below 500 employees so we do not have figures for SMEs following the EU definition of firms with under 250 employees.

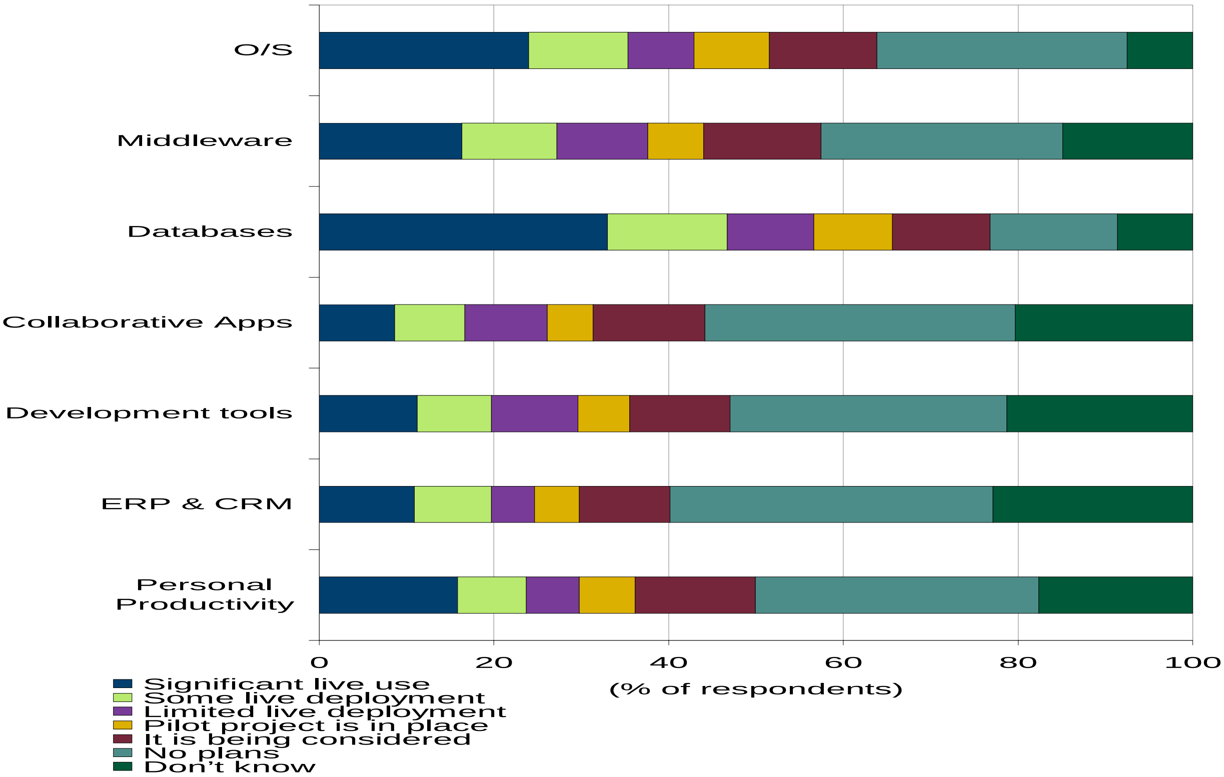
Figure 1: FLOSS usage by application in companies in the UK, Sweden, and Germany.

	UK		Sweden		Germany		Total
	small	large	small	large	small	large	
OSS as server operating system	8.1%	3.7%	9.8%	11.0%	30.7%	30.6%	15.7%
	6.4%		10.1%		30.7%		
OSS for databases	13.3%	4.6%	7.5%	8.2%	14.1%	20.8%	11.1%
	9.9%		7.6%		15.7%		
OSS on the desktop	7.6%	2.0%	3.4%	3.2%	13.7%	6.5%	6.9%
	5.4%		3.3%		12.0%		
OSS for websites	7.9%	4.3%	7.5%	8.7%	15.8%	17.3%	10.1%
	6.5%		7.8%		16.2%		

Source: Survey results (n=395).

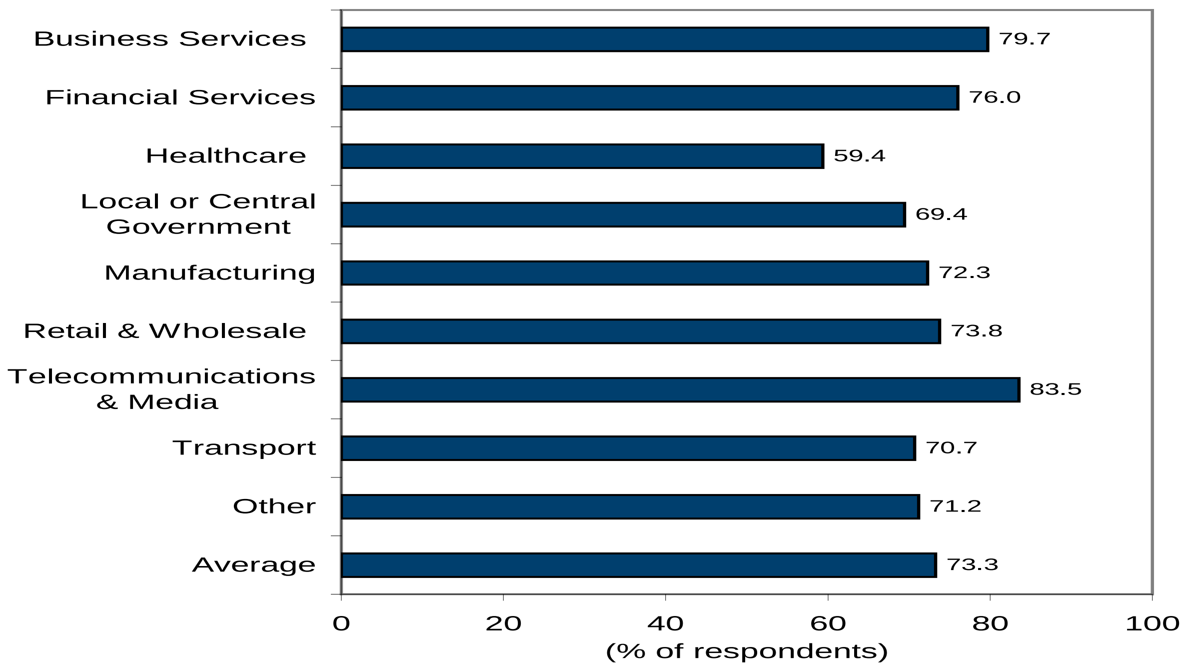
IDC's 2005 Western European Software End-User Survey of 625 firms shows a significant increase in use since the 2002 FLOSS survey, with over 40% showing "significant, some or limited" use of FLOSS in the operating systems sector and nearly 60% showing use of FLOSS databases (see Figure 2). The trend is towards greater penetration of FLOSS across several sectors of industry, as shown by the correlation between importance of software to the sector and adoption of FLOSS, (Figure 3).

Figure 2: FLOSS usage in Europe by type of application



Source: IDC's 2005 Western European Software End-User Survey (N=625))

Figure 3: FLOSS usage in Europe by industry.

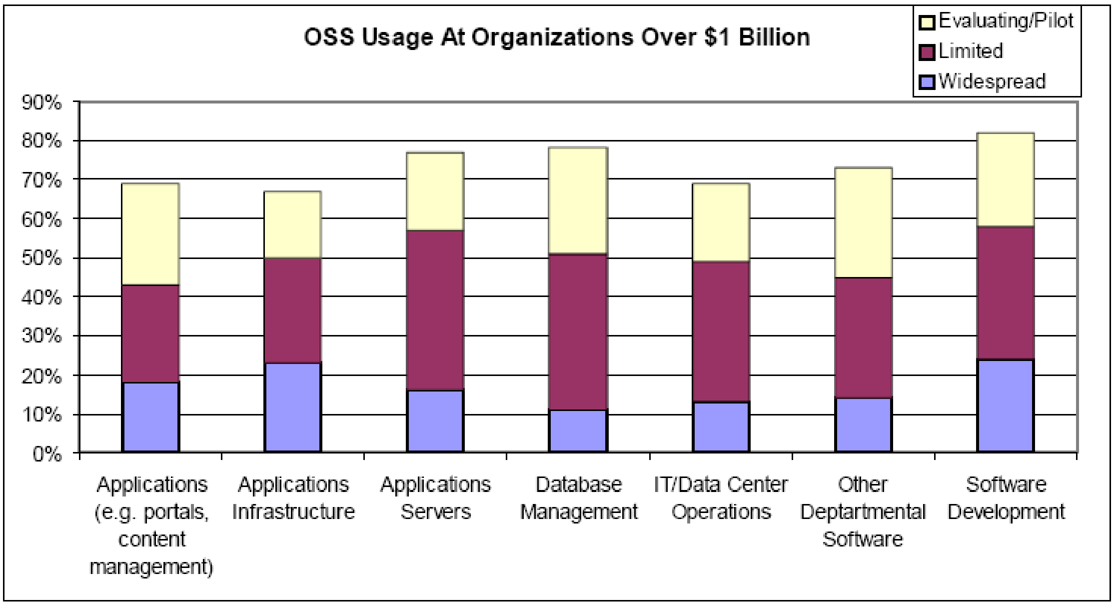


Source: IDC's 2005 Western European Software End-User Survey (N=625))

6.2.1.2. North America (USA)

Survey data suggests that FLOSS plays an important and increasing role in US companies. Walli/Gynn/von Rotz[6] report that a clear majority of U.S. companies and government institutions are turning to open source software instead of using commercial software packages. 87% of the 512 companies they surveyed were using open source software. Bigger companies are more likely to be open source users: all of the 156 companies with at least $50 million in annual revenue were using open source. Moreover, the usage of Open Source is not limited anymore to operating systems (i.e. Linux); more and more it becomes software for key departmental applications (Figure and Figure 6). Key drivers of this trend are cost savings and vendor lock-ins.

Figure 4: FLOSS usage at large organisations over $1 billion, U.S.

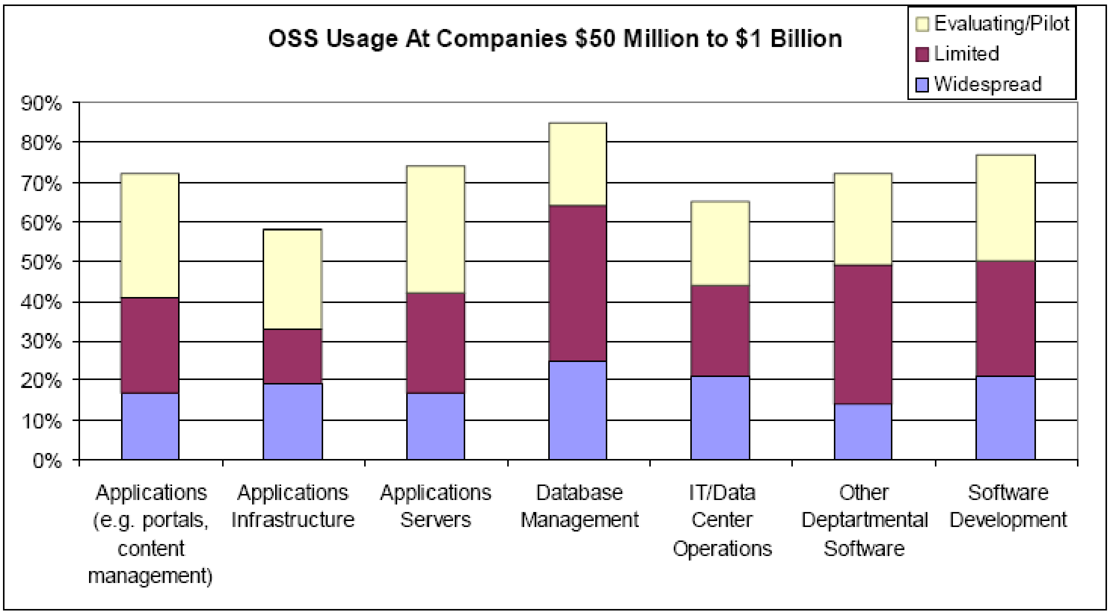


Source/Copyright © 2005 Optaros Inc (Walli/Gynn/von Rotz). (n = 612).

The authors point out "while open source software is omnipresent in U.S. companies, in certain industries, sized $50 million and up, there is significantly greater adoption (...). The telecommunications business (both service and equipment) leads all other segments that we studied in open source adoption, with a higher percentage of companies using open source databases, applications servers, portals/content management systems, software development tools, and data centre operations tools. In contrast, financial services and insurance companies had the lowest adoption rates for open source software in all but open source development tools. However, the three-year plans by financial services and insurance firms show a strong planned adoption indicating that while today's production numbers are lower, perhaps it is only a matter of time until this sector also has high production usage (**Error! Reference source not found.**).

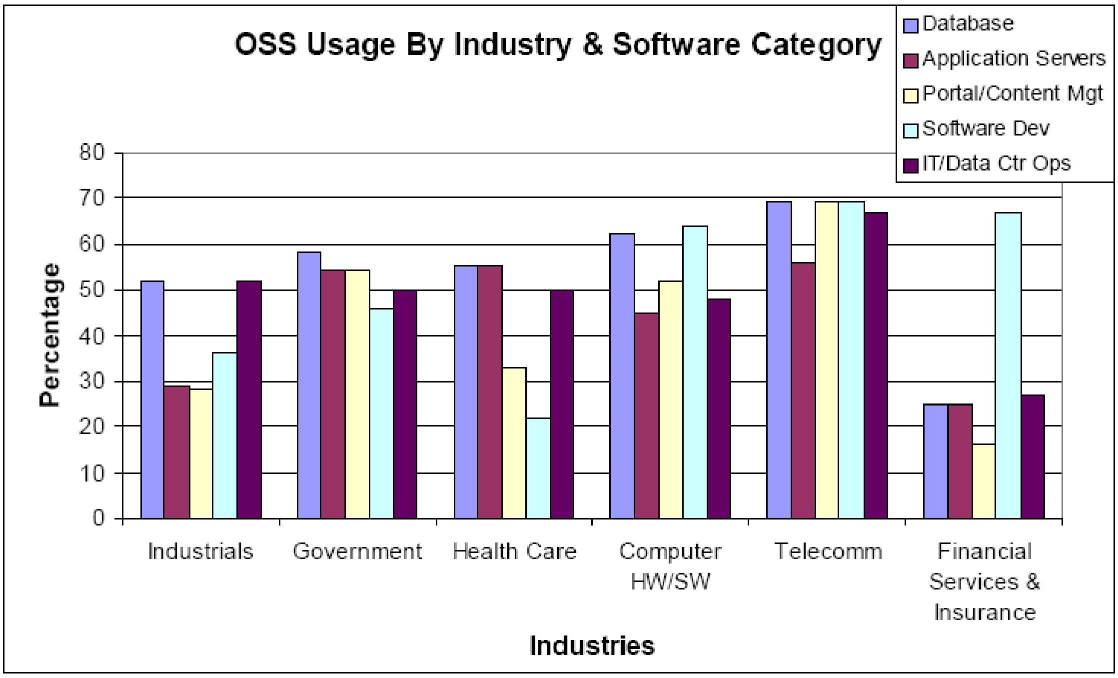
6 Stephen Walli, Dave Gynn, and Bruno Von Rotz(2005): The Growth of Open Source Software in Organizations. Optaros Publications and Thought Leadership. Boston.

Figure 5: FLOSS usage at mid-size organisations $50 million - $1 billion, U.S.



Source/Copyright © 2005 Optaros Inc (Walli/Gynn/von Rotz). (n = 612).

Figure 6: FLOSS usage at mid-size organisations, by industry and software category, U.S.

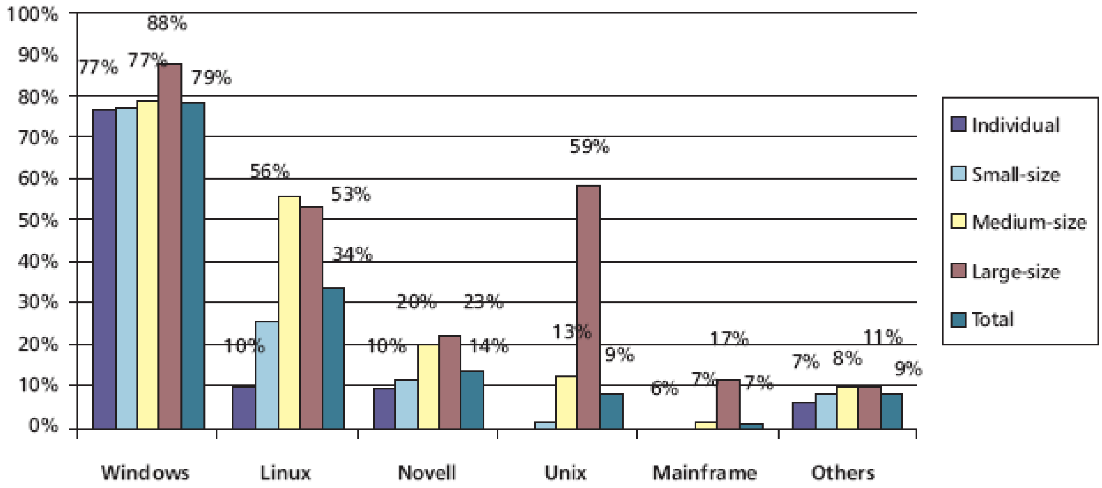


Source/Copyright © 2005 Optaros Inc (Walli/Gynn/von Rotz). (n = 612).

6.2.1.3. South America

Brazil is the country in South America with the largest number of FLOSS developers (see Figure 25 on page 43). Information on the role of FLOSS for companies appears to have been published only in this country, by Softex.[7] According to a study by I-digital[8] cited by the Softex report, “with regard to size, despite the strong presence of Unix at large companies and of Windows at companies of all sizes, Linux appears in second place over all and is used on 53% of the servers of large companies, and 56% of medium-size companies” (see Figure 7). Overall, the trend towards FLOSS in Brazil seems to be driven by medium-sized and large companies rather than by small companies[9].

Figure 7: Main operating systems used on servers in Brazil – individuals and companies.



Graph 1 - Main operating systems used on servers
Source: I-Digital: Profile of the digital company 2002/2003 (FIESP/CIESP and FEA/USP, 2004)

Copyright © 2005 Softex. (n = 1334).

The shares of FLOSS in other application areas are clearly limited: The data from the I-Digital study show that in transactional databases, “Oracle prevails for large-size companies and the Microsoft SQL Server for companies of all sizes. The use of FS/OS database (MySQL) is well behind other small-size databases such as Clipper and Access, regardless of company size. Penetration among browsers is also low. The Mozilla browser has 7% of the market and is well behind Internet Explorer which, even having lost 5% market share, continues to dominate the global market with 88.9%.” Desktop applications, such as OpenOffice.org, are also limited in use.

---

7 Softex (2005): Impact of the free software and open source on the software industry in Brazil. Campinas.

8 FIESP/CIESP & FEA/USP (2004): I-Digital: Profile of the digital company 2002/2003.

9 The fact that the figures sum up to more than 100% is explained by the fact that FLOSS and proprietary software are not mutually exclusive, as both sorts of software are frequently combined within an organisation.

As a reason for the restricted use of non-infrastructure FLOSS usage the Softex report sees a lack of user-friendly FLOSS systems, particularly a lack of user-friendly graphic interfaces (GUIs).

6.2.1.4. Asia

In Japan, Linux servers have mainly been adopted by the insurance and services sector. Evidently the adoption of Linux servers grows with increasing numbers of employees and growing annual sales (see Figure 8).

Figure 8: Adoption rates of Linux servers in the private sector in Japan

Firms adopting Linux Servers (by industry)

Sector	Adopting		Not Adopting		Total	
	n	*%*	*n*	*%*	*n*	*%*
Construction	11	39,3	17	60,7	28	100
Manufacturing	37	25,5	108	74,5	145	100
Assembly	42	39,6	64	60,4	106	100
Distribution	44	29,7	104	70,3	148	100
Service	56	53,8	48	46,2	104	100
Banking	9	24,3	28	75,7	37	100
Insurance	31	83,8	6	16,2	37	100
Total	230	38	375	62	605	100

Firms adopting Linux Servers (by size)

Employees	Adopting		Not Adopting		Total	
n	*n*	*%*	*n*	*%*	*n*	*%*
<100	19	19,8	77	80,2	96	100
100-299	53	28	136	72	189	100
300-499	54	45,8	64	54,2	118	100
500-999	42	47,2	47	52,8	89	100
≥ 1000	62	54,9	51	45,1	113	100
Total	230	38	375	62	605	100

Firms adopting Linux Servers (by size)

Annual Sales	Adopting		Not Adopting		Total	
billion of ¥	*n*	*%*	*n*	*%*	*n*	*%*
<5	23	24	73	76	96	100
5≤ bill ¥<10	22	25,3	65	74,7	87	100
10≤ bill ¥<30	81	37,9	133	62,1	214	100
30≤ bill ¥<50	17	31,5	37	68,5	54	100
50≤ bill ¥<100	27	45	33	55	60	100
≥ 100	29	50,9	28	49,1	57	100
Total	199	35	369	65	568	100

Source/Copyright © 2005-2006 Impress Corporation / Yano Research Institute, Ltd., 2005-2006

Regarding Linux adoption on the desktop the situation in Japan is clearly different, as the penetration rates are much lower than the respective rates for servers. Only assembly and service show above average adoption ratios of Linux Desktop, the lowest shares occur in insurance, banking, and manufacture. With regard to size it turns out that small (<100 employees) and large companies (between 300 and 1000 employees) show above average shares of Linux Desktop adoption. It is companies with less than 10 billion ¥ and more than 100 billion ¥ show above average shares of Linux desktop adoption (Figure 9).

Figure 9: Adoption rates of Linux desktop in the private sector in Japan.

Firms adopting Linux Desktop (by industry)

Sector	Adopting		Not adopting		No response		Total	
Sector	*n*	*%*	*n*	*%*	*n*	*%*	*n*	*%*
Construction	1	3,6					28	3,6
Manufacturing	4	2,8					145	2,8
Assembly	9	8,5					106	8,5
Distribution	4	2,7					148	2,7
Service	6	5,8					104	5,8
Banking	1	2,7					37	2,7
Insurance	1	2,7					37	2,7
Total	26	4,3	575	95,0	4	0,7	605	100

Firms adopting Linux Desktop (by size)

Employees	Adopting		Not Adopting		No response		Total	
n	*n*	*%*	*n*	*%*	*n*	*%*	*n*	*%*
<100	6	6,3					96	6,3
100-299	6	3,2					189	3,2
300-499	6	5,1					118	5,1
500-999	6	6,7					89	6,7
≥ 1000	2	1,8					113	1,8
Total	26	4,3	575	95	4	0,7	605	100

Firms adopting Linux Desktop (by size)

Annual Sales	Adopting		Not Adopting		No response		Total	
billion of ¥	*n*	*%*	*n*	*%*	*n*	*%*	*n*	*%*
<5	6	6,3					96	6,3
5≤ bill ¥<10	5	5,7					87	5,7
10≤ bill ¥<30	8	3,7					214	3,7
30≤ bill ¥<50	1	1,9					54	1,9
50≤ bill ¥<100	2	3,3					60	3,3
≥ 100	3	5,3					57	5,3
Total	25	4,4	540	95	4	0,7	568	100

Source/Copyright © 2005-2006 Impress Corporation / Yano Research Institute, Ltd., 2005-2006

There is limited information on FLOSS in the Asian business sector. In Malaysia, a survey conducted by MAMPU in 2005 showed that 74% of public sector organisations implemented FLOSS solutions.

According to IDC, in a report for OSDL[10], Linux accounted for 14% of servers and 5% of PCs in 2004 in the Asia-Pacific region, expected to grow to 25% and 9% respectively by 2008.

[10] IDC, 2004, “The Linux Marketplace - Moving From Niche to Mainstream”, prepared for OSDL.

CIO-Asia reports a market penetration 80% of Chinese organisations using FLOSS applications, and provides noteworthy cases such as ICBC: “With 5.3 trillion renminbi (US$640 billion) in total assets, ICBC is China’s biggest bank, serving 100 million individuals and 8.1 million corporate accounts through more than 20,000 branch offices across China. When the project is completed, many of ICBC’s 390,000 employees will be accessing applications hosted on Linux servers on a daily basis.”

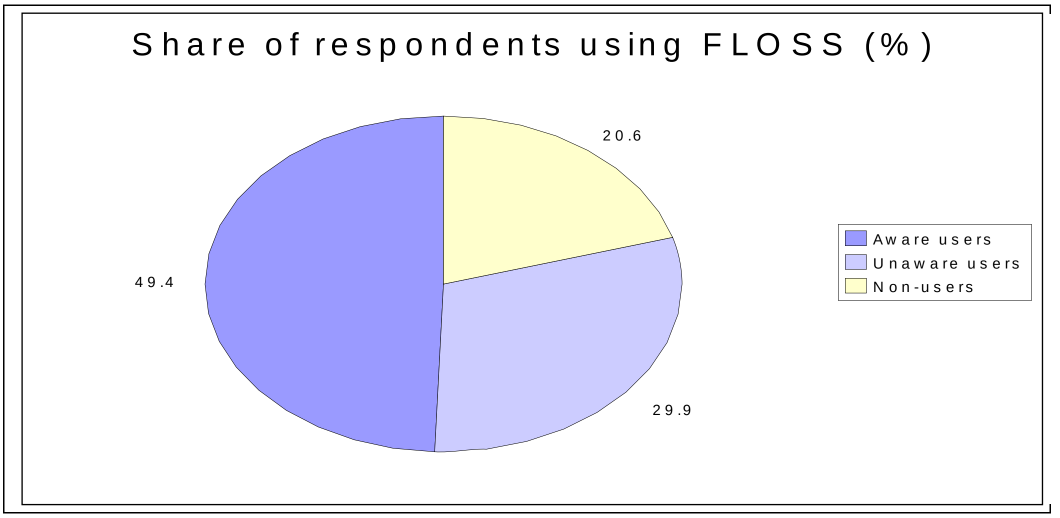
6.2.2. FLOSS usage in the public sector

The awareness for FLOSS in the public sector is continuously increasing. Therefore, policy initiatives towards Open Source play an ever-growing role. According to a CSIS publication[11], in 2005 there were 265 initiatives around the world of which, most of them are preferred initiatives (i.e. FLOSS from the inception of the initiatives). Europe launched 126 Open source policy initiatives, in Asia there were 73, there were 40 in Latin America, 17 in North America and 4 each in the Middle East and Africa. Most projects were launched by the European Union (16), France (10) and Germany (9) are also heavily involved on FLOSS projects, Italy, Spain and UK comes next to these countries with 6 projects each. Small countries like Belgium (5), Denmark (5), Finland (3), Netherlands (3), Norway (4) and Sweden (3) are also involved in FLOSS projects

6.2.2.1. Europe

However, reliable data on FLOSS usage in the public sector outside Europe is not readily available. The survey conducted by MERIT under the FLOSSPOLS project of 955 public sector organisations in 13 EU countries in the third quarter of 2004 is the single largest information source in this context, but limited to Europe. It showed that a large share of users of FLOSS is “unaware users” – they claim not to use “open source software” but say they use Linux, Apache, MySQL etc. (see Figure 10).

Figure 10: Share of EU government authorities using FLOSS.



Copyright © 2005 MERIT. Source: FLOSSPOLS Government survey (n = 955).

11 http://www.csis.org/media/csis/pubs/060101_ospolicies.pdf

The FLOSSPOLS survey of government organizations also provides very detailed data on the use of FLOSS in EU public administrations including break-up by size (see Figure 11).

Figure 11: EU public administrations using FLOSS, by size (number of IT employees).

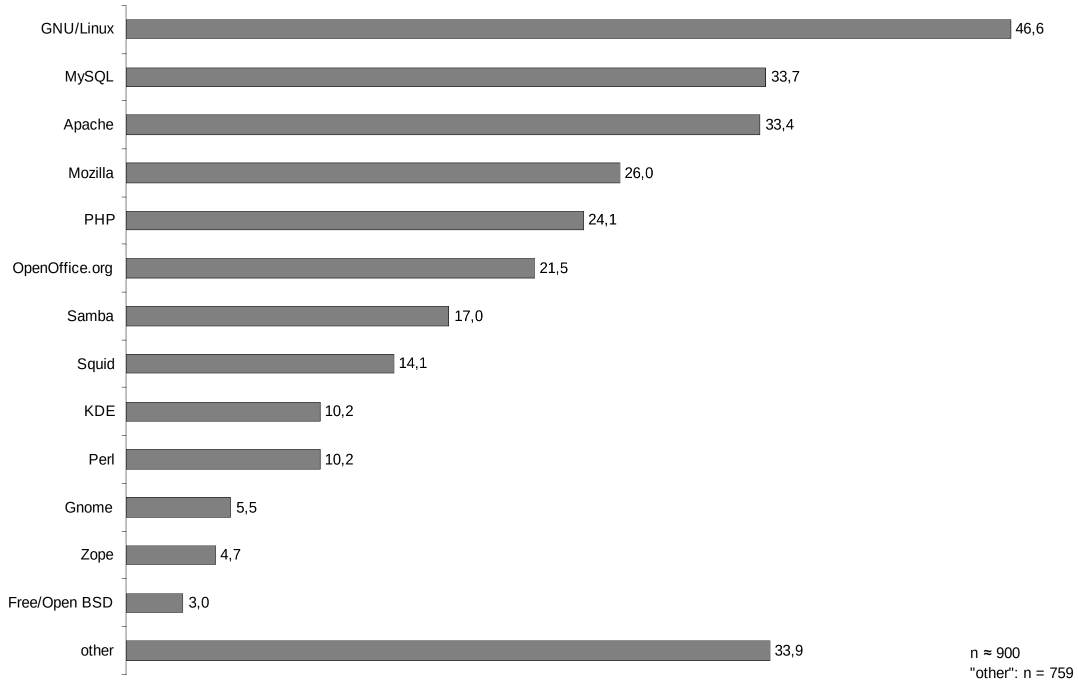
	Number of employees in IT department					
Awareness of FLOSS usage in organisation	1	2 - 3	4 - 10	11 thru 30	more than 30	Total
aware user	31.2	43.9	56.2	65.3	77.0	50.1
unaware user	50.3	34.7	22.5	16.5	6.8	30.0
non-user	18.6	21.4	21.3	18.2	16.2	19.9
Total	100.0	100.0	100.0	100.0	100.0	100.0

$p < 0.01$ Contingency Coefficient: 0.299 n = 923

Copyright © 2005 MERIT. Source: FLOSSPOLS Government survey

The FLOSSPOLS survey clearly revealed that FLOSS systems play a very important role for local governments in Europe, though with regard to operating systems there is still a prevalence of proprietary systems.

Figure 12: FLOSS systems used in European public bodies (%).



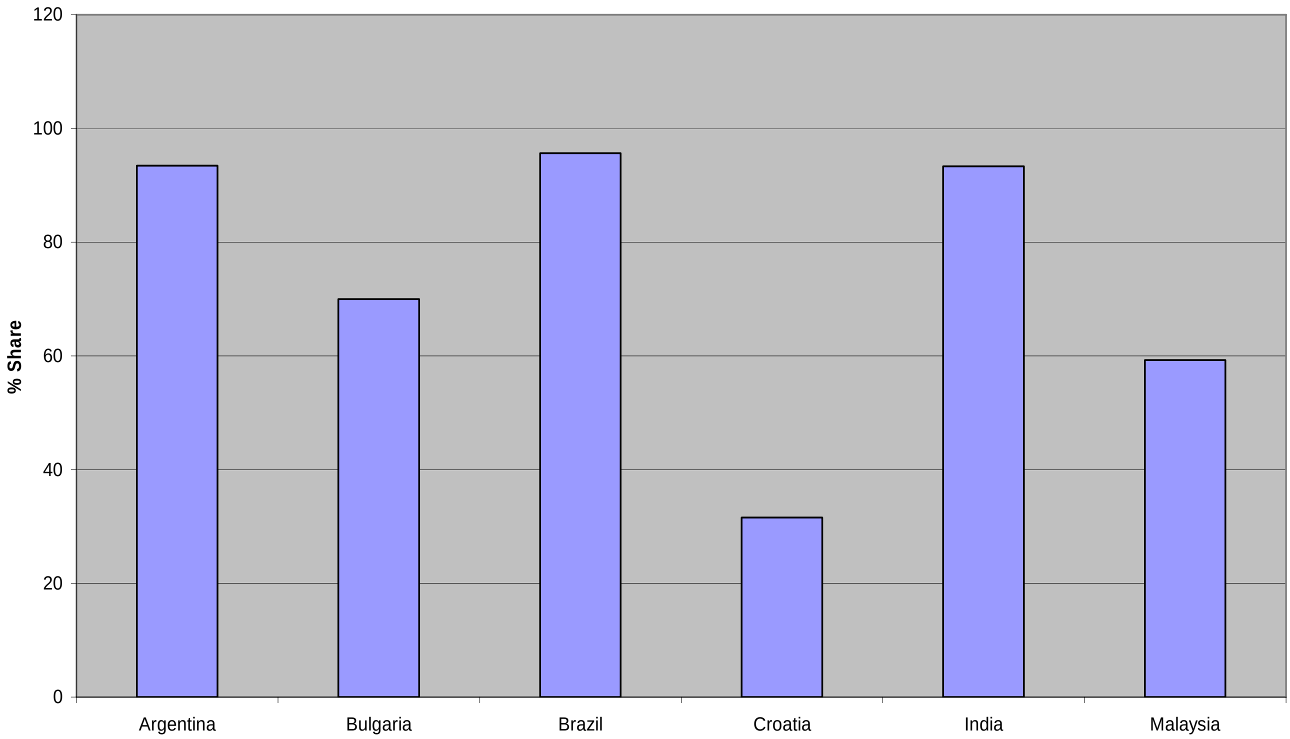
Copyright © 2005 MERIT. Source: FLOSSPOLS Government survey

6.2.2.2. World

Little official data exists for the FLOSS share of software purchased by government in other countries. However, Brazil has a policy aiming to have a FLOSS share of software in government of 80%, though this has not yet been achieved. Linux accounted for 39% of operating systems procured on Chinese government systems in 2004[12].

According to early results from the on-going surveys conducting by the FLOSSWorld project, a high share of government authorities report some use of FLOSS, especially in Latin America and Asia (Figure 13).

Figure 13: Government authorities using FLOSS.



Source: FLOSSWorld project (www.flossworld.org)

In all countries surveyed, over 60% of respondents (including non-users) plan to increase FLOSS use. It is also interesting that while as with Europe, most use of FLOSS is quite limited, a large majority of respondents in most countries report some use of FLOSS on the desktop.

Table 1 shows the shares of respondents using specific applications by country. It is clear that Mozilla/Firefox is among the most popular FLOSS applications, while a significant share of respondents especially in Latin America uses OpenOffice and the database applications MySQL or PostgreSQL.

[12] Prof. Shouqun Lu, China OSS Promotion Union, at the FLOSSWorld Beijing Workshop

Table 1: Use of specific FLOSS applications in government (% shares).

	OpenOffice	Firefox	MySQL/PostgreSQL
Argentina	58	84	78
Bulgaria	40	70	60
Brazil	91	83	83
Croatia	16	37	16
India	73	73	80
Malaysia	14	43	47

6.2.3. Market shares of FLOSS applications

6.2.3.1. Overview and approach

For this task, we draw on numerous broad-based empirical surveys of FLOSS market share, starting with the FLOSS project's extensive survey of user organisations, which provides an in-depth empirical understanding of the motivations of and benefits to user organisations and a basis for comparison with more recent data in order to determine trends. The FP5 FLOSS Final Report (part 2: "Firms' Open Source Activities: Motivations and Policy Implications") also included a survey of business strategies and policy implications relating to open source, which will further inform this section of the study.

The Tender Specifications call for a number of indicators to determine the market share of FLOSS. The simplest and most straightforward is what is gathered by a number of market research organisations (e.g. IDC, Gartner, Forrester) based on data from systems vendors. However, this kind of data is aligned with some shortcomings:

- Download figures for Linux and other FLOSS applications such as OpenOffice are available but again represent downloads, not necessarily actual use, and may over represent use; they may also under represent use since a single download can be installed on many computers.
- Furthermore, the common phenomenon of dual systems (e.g. "dual-boot" or "live" distributions, ways of using Linux in addition to Windows on a single computer) is likely to be misrepresented in market share as Windows-only usage.
- Similarly, market share data may ignore the increasingly popular use of FLOSS applications on proprietary operating systems (e.g. OpenOffice on Windows). This area of FLOSS usage is significant and growing rapidly as it is easier for users to adopt new applications than to switch operating systems.
- Due to the difficulty of determining accurate numbers from market sources, as illustrated above, a number of publicly available figures (again, from various commercial research organisations such as IDC, Gartner, Forrester etc) are based on estimating actual use through surveys of relatively small samples of firms. IDC, for instance, recently published its "European End-User Survey: 2005 Spending

Priorities, Outsourcing, Open Source, and Impact of Compliance"[13] based on a sample of 625 firms.

The approach of the study for addressing the specific attribute of market share in terms of usage of FLOSS software is based on summarising published data sources. We will compare published figures from different market research sources with a view to providing a critical analysis of the methodologies used and therefore, perhaps, a more reliable and stable indicator than any individual set of market research projections.

We also include in our analysis figures from more reliable "census" –type sources, although these tend to be limited in terms of the market that they analyse. The best known of this sort of market share indicator is the monthly Web Server Survey conducted by the research firm Netcraft. This is a "census" of "all" accessible web servers on the Internet

In the following sections we will present and discuss market shares of different FLOSS applications separately

6.2.3.2. Operating Systems

Gartner announced last year that Linux was on 5% of PCs worldwide in 2004, based on sales figures from vendors for pre-installed Linux. Gartner noted that not all PCs shipped with Linux continue to run on Linux and estimated that only 2% of PCs continue to run Linux. However, such data sources may also *under-represent* Linux use, since it is probably more likely for people to buy a PC with pre-installed Windows and replace that with Linux than the other way around, given that PCs with Linux pre-installed are hard to find.

According to IDC ("European End-User Survey: 2005 Spending Priorities, Outsourcing, Open Source, and Impact of Compliance")[14], 25% of firms (n = 625) use FLOSS operating systems.

According to the FLOSS user survey in 2002, only 15.7% of European firms (but 30.7% of large organisations in Germany) used FLOSS operating systems

Meirelles (2004)[15] estimates the share of Linux on servers in Brazil about 12%.

According to the IDC report for OSDL[16], the overall Linux marketplace revenues for server and PC hardware and packaged software on Linux is expected to reach $35.7 billion by 2008, representing an annual growth of 26%. The Linux share of total server shipments and redeployments worldwide is projected to grow from 20% in 2004 to 27% in 2008. The share in the Americas grows from 24% to 32%; Europe[17] has a lower share of Linux use on the server, according to IDC, growing from 16% in 2004 to 25% in 2008. However, Europe's

---

13 http://www.idc.com/getdoc.jsp?containerId=LC01M

14 http://www.idc.com/getdoc.jsp?containerId=LC01M

15 Meirelles, F. S. (2004): Tecnologia de Informação (Information Technology) São Paulo: EAESP/FGV. Available at http://www.fgvsp.br/cia/pesquisa/

Pesq04GV.pdf. Accessed on 12/21/2004.

16 IDC, 2004, "The Linux Marketplace - Moving From Niche to Mainstream", prepared for OSDL.

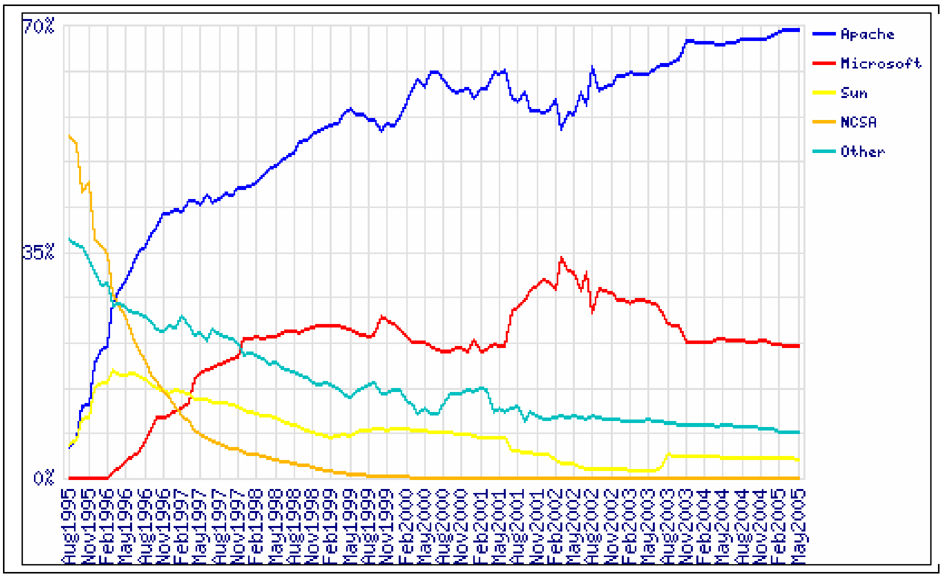
17 Actually, EMEA: Europe, Middle East and Africa

use of Linux is much higher on the desktop than the Americas: 5% vs. 2% in 2004 growing to 9% in 2008 in Europe vs. 4% in the Americas.

6.2.3.3. Web servers

The monthly Web Server Survey conducted by the research firm Netcraft provides a "census" of "all" accessible web servers on the Internet, over 63 million in May 2005 – of which 70.29% run the FLOSS web server application, Apache (see Figure 14).

Figure 14: Share of web servers (Apache = FLOSS).



Copyright © 2005 NETCRAFT LTD. Source: May 2005 Web survey of 63,532,742 hosts. Online at: http://news.netcraft.com/archives/2005/05/01/may_2005_web_server_survey.html

According to a census/poll by nexen.net the market share of Apache in Europe (May 2006) is 84%, while in the USA Apache holds a market share of 66%.

6.2.3.4. Mail servers

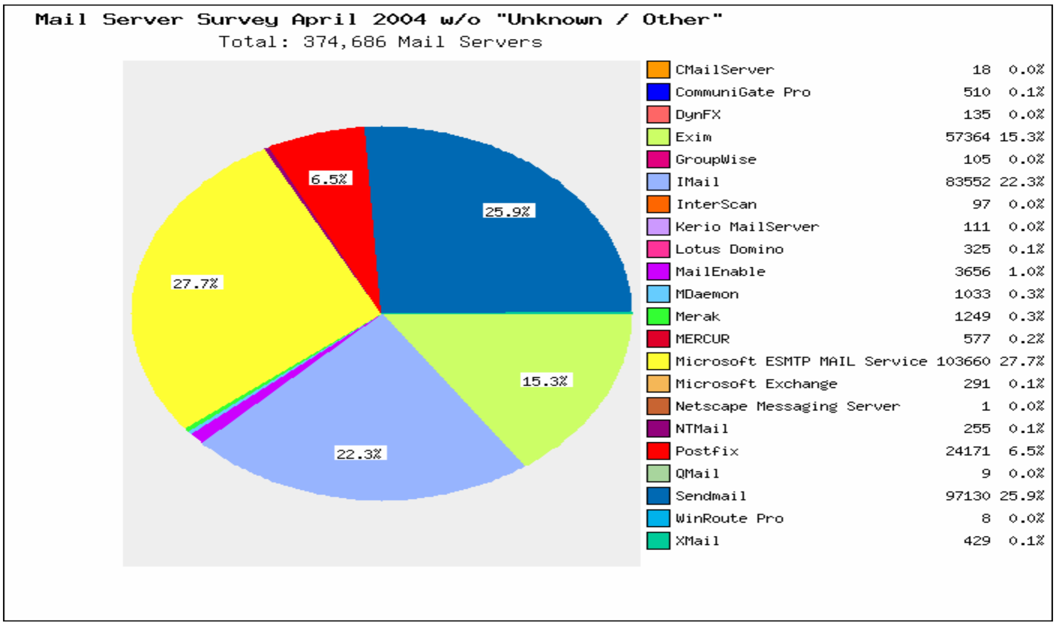
Mail servers provide the backbone of the Internet's most popular application, e-mail. The FalkoTimme mail server survey[18] provides "census-like" data (based on IP addresses scanning) in Europe and the US.

For each country there is a list of active mail servers and the frequency of their occurrence. The most popular mail servers are listed in Figure 15, other mail servers with a minor role and mail servers that do not reveal the software they are running are excluded The FLOSS share of mail servers is 47.8% (Sendmail, Exim, Postfix, Qmail), with FLOSS products occupying the 2nd, 4th and 5th ranks in market share. This may under represent FLOSS market share, as backbone mail servers have historically run on FLOSS servers

[18] Covering Austria, Belgium, Bulgaria, Czech Republic, France, Germany, Greece, Hungary, Ireland, Italy, Netherlands, Norway, Portugal, Romania, Russia, Spain, Sweden, Switzerland, Turkey, UK and US. Available at http://www.falkotimme.com/projects/survey_smtp_032004.php

(typically Sendmail, which was among the first servers for the SMTP protocol used for Internet e-mail).

Figure 15: Mail server "census"



6.2.3.5. Scripting languages (PHP)

Dynamic scripting language is a market pioneered and dominated by FLOSS applications, such as Perl and PHP. According to a poll by nexen.net the market share of PHP in Europe (May 2006) is 26%; while in the USA PHP holds a market share of 34%.

6.2.3.6. Application servers

According to BZ Research[19], the FLOSS JBoss Application server has a market share of 37% head-to-head with the 37.2% share of the 2005 market leader, IBM's WebSphere. (In 2004, JBoss was the market leader, just ahead of WebSphere.)

6.2.3.7. Databases

According to IDC 33% of firms in Western Europe (n = 625) use FLOSS databases. According to the FLOSS user survey in 2002, only 11% (but 14% of SMEs in the UK) used FLOSS databases (compared with IDC's report of 25% and 33% respectively for 2005). Thus there has been considerable growth in this sector.

This growth is tracked by other sources as well. For instance, Gartner says the open source database segment grew 47% over 2005, compared to an 8% increase of the overall database market. Linux showed the strongest momentum, with an increase of 84%. However, the top position of Oracle, with a market share of 49%, remained untouched.[20]

---

[19] BZ Research, 2005. *Fifth Annual Java Use & Awareness Study*. December. www.bzresearch.com

[20] http://www.crn-india.com/breakingnews/stories/64477.html

6.2.3.8. Desk-top office software (OpenOffice.org)

According to OpenOffice.org[21], by mid 2006 there were 62.500.000 downloads of OpenOffice.org 2.0 from the main site.

According to Jupiter Research, in 2003 OpenOffice had a 6% market share among SMEs[22]. According to Yankee Group, the 2005 market share of OpenOffice.org among SMEs was much higher, at 19%[23]. According to Forrester Research in 2004, OpenOffice has an 8.5% market share among major North American companies.

In Europe, TechConsult estimates a market share of 8% in German businesses for OpenOffice[24].

6.2.3.9. Web browsers

A compilation of several surveys by David Wheeler (2005) shows the FLOSS web browser Mozilla/Firefox growing from 5% to almost 10% of the market since the launch of Firefox in November 2004, at the expense of Microsoft's Internet Explorer. Indeed, among home users (who are free to choose their web browsers while business users must usually wait for a company-wide change) a study by XitiMonitor shows that 21.6% were using Firefox in March 2005, in Germany, and 12.2% in France; ArsTechnica, which tracks what browsers website developers themselves use, shows a majority using FireFox by March 2005 (see Figure 16).

---

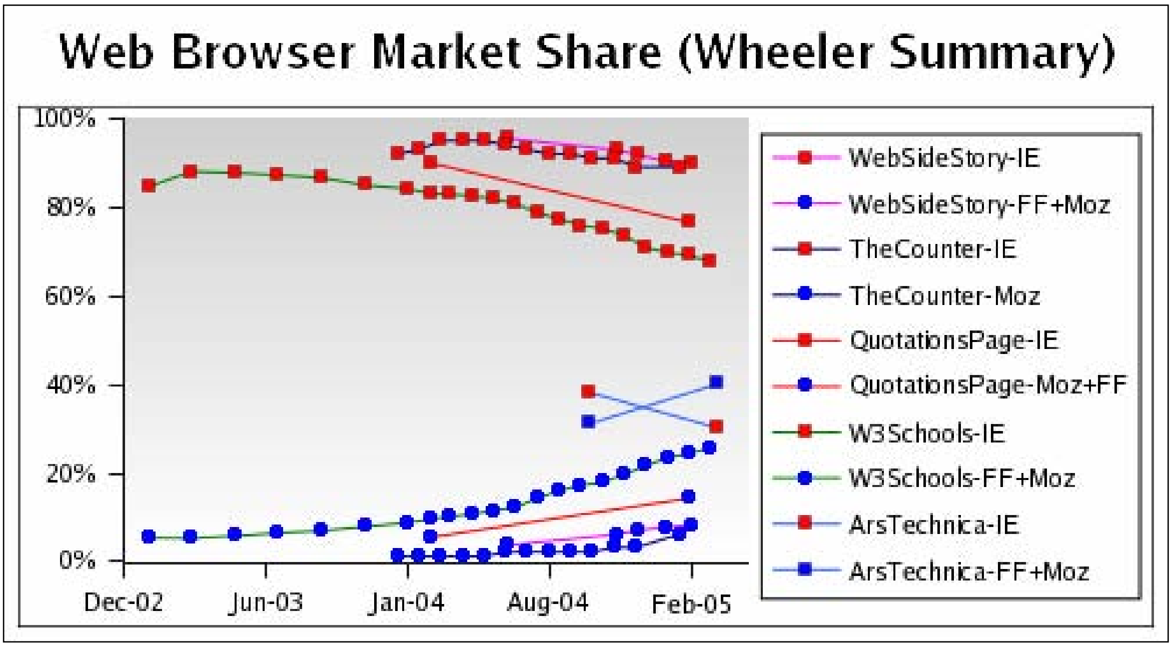
[21] http://stats.openoffice.org/spreadsheet/index.html

[22] http://www.infoworld.com/article/03/07/16/HNjupiter_1.html

[23] Http://www.toptechnews.com/news/OpenOffice-org-2-0-Release-Delayed/story.xhtml?story_id=03100339SMZN

[24] http://www.perspektive-mittelstand.de/pages/business%20forum/presse-service-meldung-detail.php?prmID=522

Figure 16: Proprietary and FLOSS web browser market share

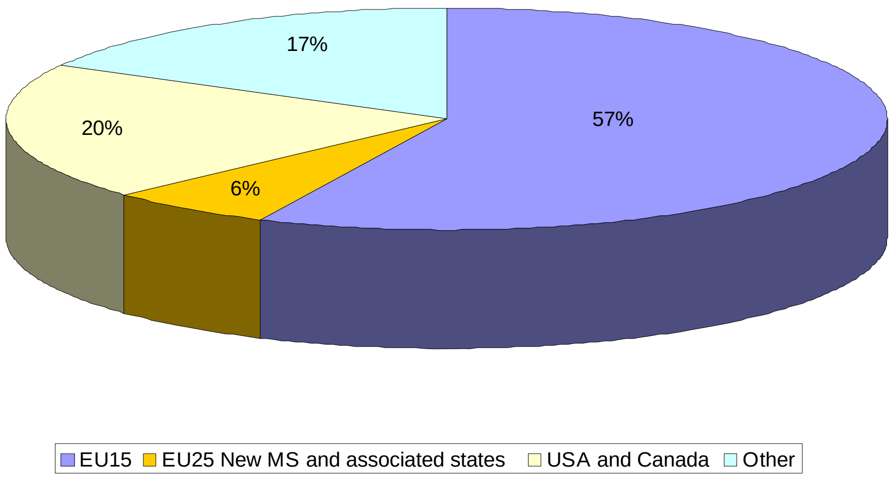


Source: David Wheeler, http://www.dwheeler.com/oss_fs_why.html.

6.3. Demographics of FLOSS developers

The combined FLOSS (MERIT/FP5) and FLOSS-US (Stanford) developer surveys form the largest survey-based dataset (4282 cases) providing the geographical distribution of developers (see Figure 17). According to these surveys, more than three fifths of the worldwide FLOSS developer community live in the EU, one fifth in North America, and another one fifth or so live in other countries.

Figure 17: Geographical distribution of FLOSS developers



Copyright © 2004 MERIT. Source: MERIT FLOSS Survey, Stanford University FLOSS-US survey (n=4282)

It should be noted that surveys are not an entirely accurate method of determining geographic distribution of FLOSS development, as they are usually random samples subject to geographical biases. The figures above in particular may under represent participation from Asia, where language barriers limit interaction with the global developer community.

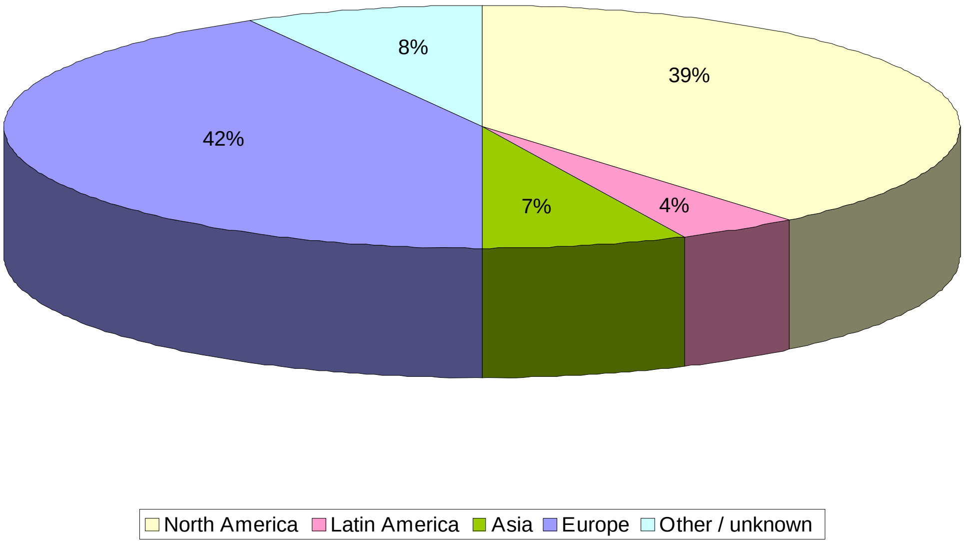
More reliable figures can be determined with “census-type” approaches[25] such as analysis of the Linux Software Map (Jones et al. 1999) or the geographical distribution of users of Sourceforge.Net, the world’s largest portal/environment for FLOSS development. While Sourceforge.Net is seen to be largely US-based, in fact the demographics (see Figure 18), which are quite reliable as they are based on the IP address[26] of developers, once again show Europe’s lead, in 2002.

---

[25] “Census-type” approaches provide information on *all* cases rather than a sample, e.g. *all* Linux developers or *all* Debian project leaders.

[26] physical Internet numbers, easily mapped to geographical locations

Figure 18: Geographical distribution of developers on Sourceforge.



Copyright © 2005 MERIT. Source: sourceforge.net. Data for end-2002

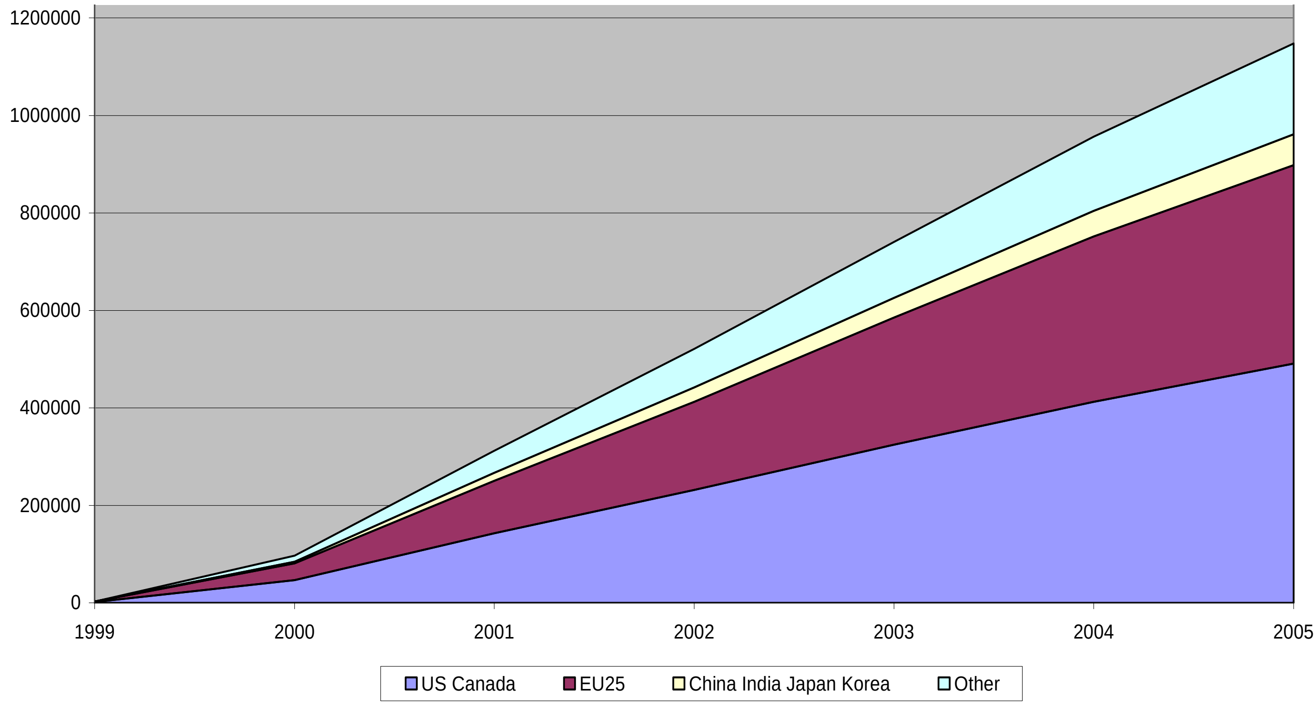
Data from Sourceforge and other sources such as the mailing lists of global projects may under represent some regions, in particular East Asia where China and Japan have a sizeable developer community, but one which does not directly interact extensively with the global community due to cultural and linguistic differences. Indeed, the participation of developers from China and Japan in global projects is often channelled through a few key "connectors", individuals who for reasons of language and culture are more comfortable contributing to global projects, and contribute on behalf of others. Nevertheless, from an economic perspective it is useful to examine the distribution of participation in global projects and portals such as Sourceforge, as they are good indicators of the population of *globally active* developers. While they under represent China and Japan in absolute numbers, they probably provide an accurate representation of the global influence on FLOSS development from China and Japan, which is arguably disproportionately low given their lower degree of global participation. This may affect the software development that is taking place in these countries – for instance, it may be that communities who are less globally active focus their development more on local needs, such as language localization. This is a topic of research in the on-going FLOSSWorld project.

The best "census-type" data on geographical distribution is derived from an analysis of developers' contribution to source code by the direct analysis of developers' identities in software project data. Such data has been extracted from URJC's vast database of FLOSS software project source code and version control information. Combined with data from Sourceforge on year of first participation or registration produces interesting results[27]. As shown in Figure 19, the US and Canada together have a slightly higher share than Europe

[27] Note that identifying countries for developers is non-trivial. URJC has developed a method that combines e-mail address analysis, time zone data and additional country data when (rarely) available.

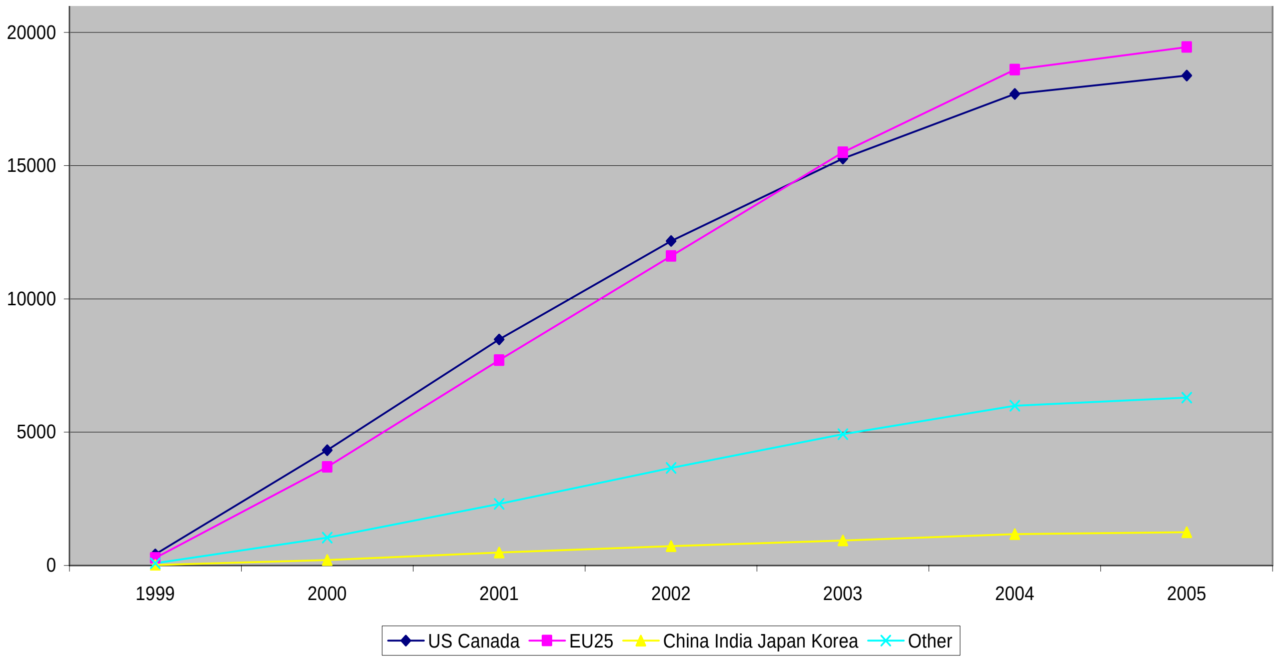
(European Union's 25 member states) in terms of the number of participants who registered on Sourceforge. As mentioned before, this portal may be biased towards US participants, but is the largest single point of access for globally active developers.

Figure 19: Globally active FLOSS participants by region



Cumulative registrations. Copyright © 2006 MERIT. Source: URJC; registration data from Sourceforge/University of Notre Dame

Figure 20: Globally active core developers by region

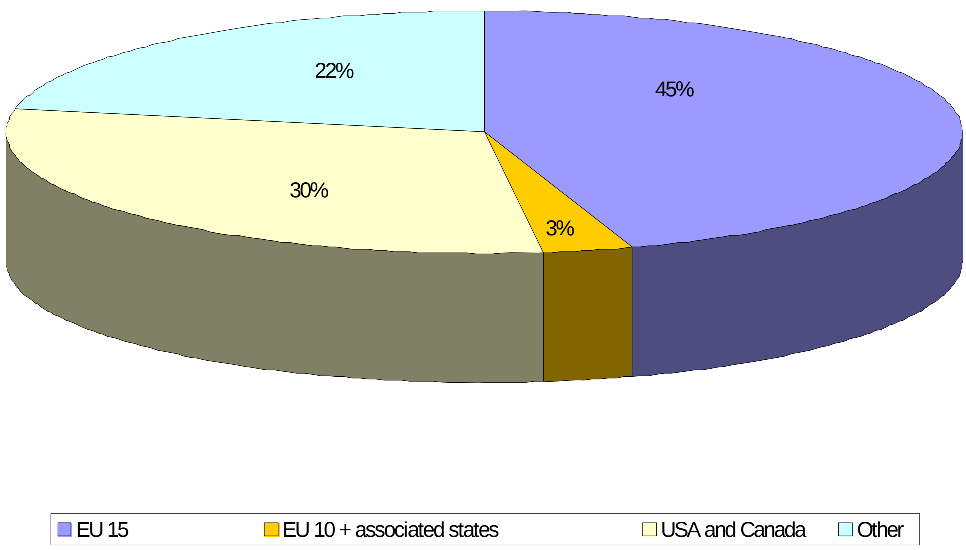


Committers only. Copyright © 2006 MERIT. Source: URJC / FLOSSWorld, registration data from Sourceforge/University of Notre Dame

However, even within registered Sourceforge users, only a small proportion actually make *commits* to FLOSS project's source code. I.e. these are people who make changes to the code in the publicly available central repositories[28], and thus also play a greater leadership role in the projects' development. Out of some 1.1 million registered participants on Sourceforge, under 50 000 commit code to the software's central repositories. As Figure 20 shows, despite the Sourceforge portal's possible US bias, European developers have in recent years overtaken North American developers in terms of such active participation.

Figure 21, which shows the location of project maintainers in the large FLOSS software collection Debian, provides a picture of coordination and leadership roles in projects. This too indicates that a plurality of project leaders among developers reside in Europe.

Figure 21: Geographic distribution of leadership in development

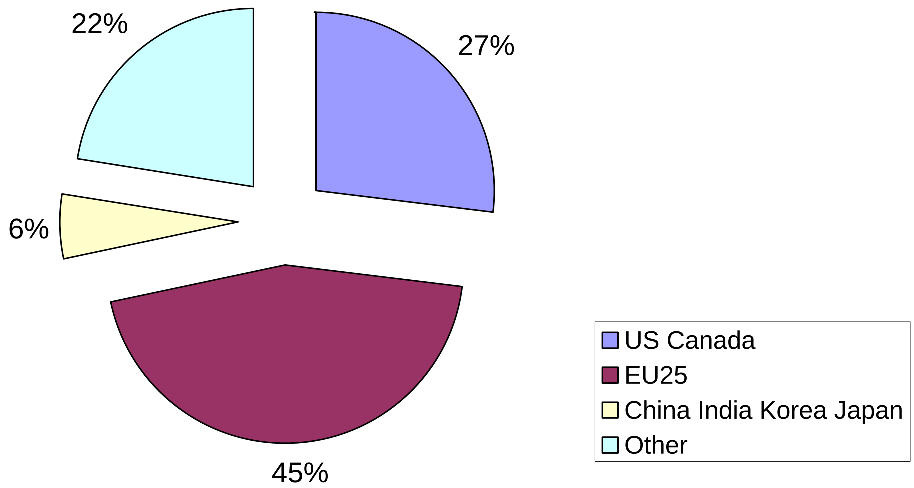


Copyright © 2006 MERIT. Source: db.debian.org

Figure 22 shows the location of active participants in developer mailing lists for selected globally developed FLOSS software projects: FreeBSD, GNOME as well as 97 mailing lists from the large FLOSS software collection Debian. This covers a time period from 1995 to 2005 (for Debian projects; for FreeBSD and GNOME the period covered is since 2003 and 2001, respectively). It provides a picture of coordination and development roles in projects, and may be considerably less biased than Sourceforge towards US registrations. Indeed, if one looks at this as representing only the specific projects examined, it is not a biased sample at all but a complete sample of active contributors, contributing to project development discussion. It is notable that this too indicates that a plurality of participants from Europe. The high share of developers from "other" regions includes Latin America, Eastern Europe and other parts of Asia as well as other parts of the world, indicating at least in part the extensive global diversity in development discussion (if not actual coding) and localisation. The inclusion of GNOME – a popular graphical user interface – in our selection of projects probably increases the apparent global diversity related to localisation.

[28] They may also make changes on behalf of other authors, by contributing the code of other authors into the central repository.

Figure 22: Geographic distribution of developer mailing list participants



 Source: URJC, authorship analysis of FreeBSD, GNOME, Debian developer mailing postings, 1995-2005

Looking beyond Western Europe and North America, it is clear that FLOSS development is closely correlated to general levels of ICT participation, population and economic wealth. The following pages show world maps plotting globally active core developers (*committers)* on Sourceforge, by country.

As Figure 23 shows, the number of FLOSS developers is far from evenly distributed in terms of population. Indeed, weighted by population there are more FLOSS developers in the US and Canada than in most European countries, with the exception of Scandinavia; Australia has a particularly high share of developers.

This is easily explained given Europe's relatively low indicators for ICT use (compared to North America) – a good proxy is Internet penetration, which is lower in Europe. As seen in Figure 24, the US has fewer FLOSS developers per million Internet users than most European countries.

Finally, the low FLOSS developer numbers from Asia and Latin America are perhaps most directly influenced by wealth. As Figure 25 shows, China, India, Russia, Brazil and even South Africa are among the higher contributors when numbers are adjusted for wealth; in fact, India with 606 committers per $1000 GDP/capita has by far the highest wealth-adjusted contribution to global FLOSS development, almost half as much again as the next contributor, China.

Figure 23: World map of committers by population

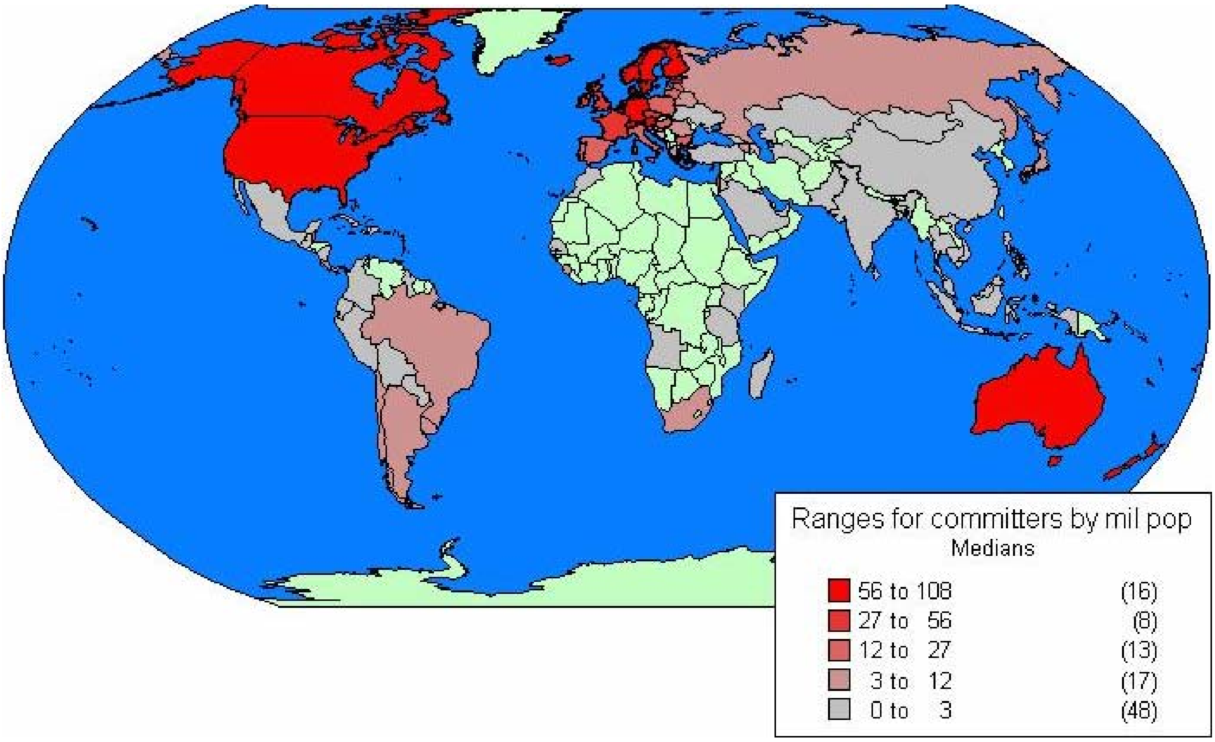


Figure 24: World map of committers by Internet penetration

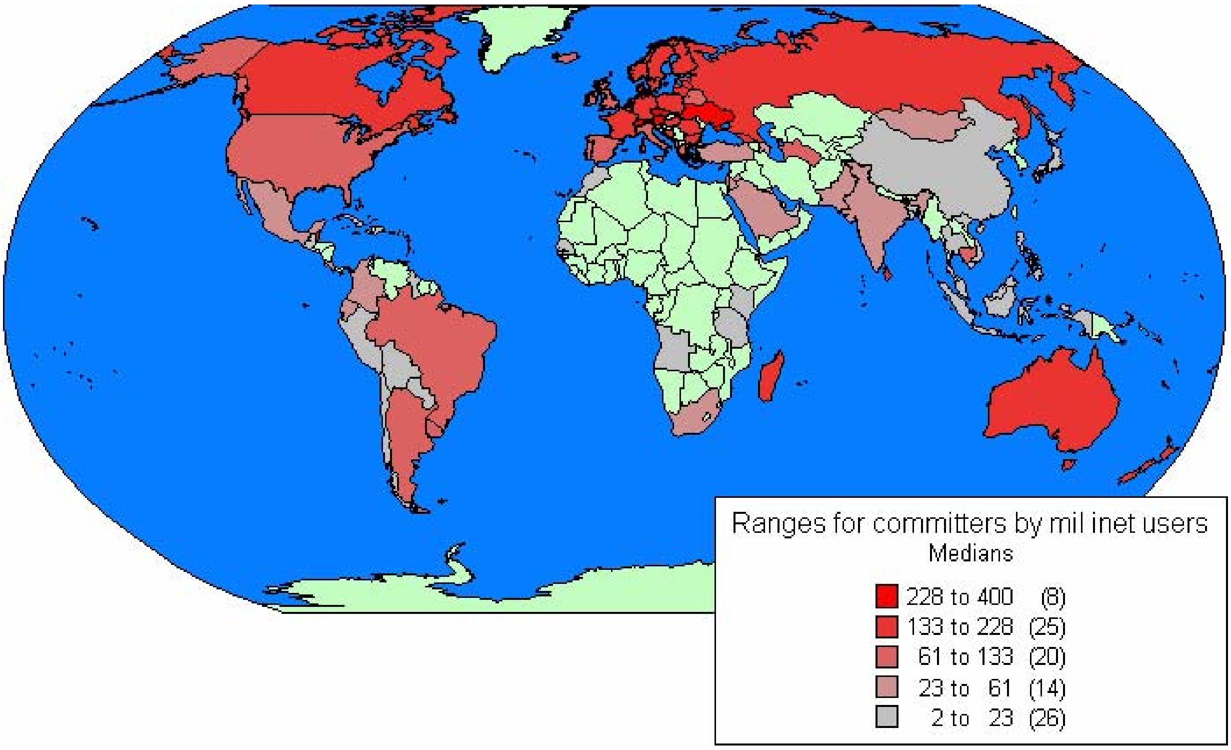
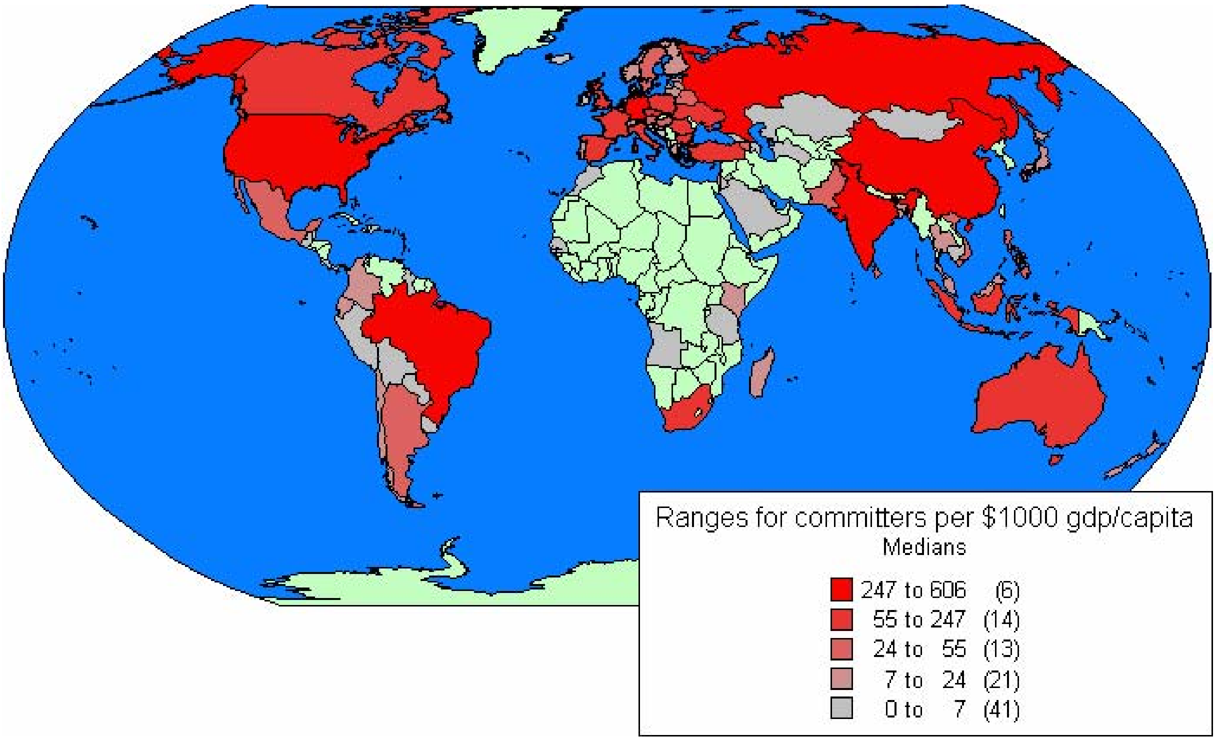


Figure 25: World map of committers by average wealth



Copyright © 2006 MERIT. Source: URJC; World Bank World Development Indicators 2001

7. Direct economic impact of FLOSS

Measuring and forecasting the impact of Free/Libre/Open Source Software (FLOSS) on the EU ICT market calls for an understanding of the variety of factors that determine the demand for ICT goods and services. Focussing only on direct effects of a growing development and usage of FLOSS on the software sector (or even on the ICT sector competitiveness and its market in general) is likely to miss the most important effects. To illustrate this point, let us consider some major transformations that the ICT market has experienced in the past, and in particular the development of Web-based information exchange. Most Web-based information exchange does not directly generate economic value. The share of access to contents or services that is provided against payment or subscription is still negligible despite some recent increase. The Web-advertising market has undergone a significant growth, but transfer of advertising from other media can account for a large part of growth. Despite that, even after the dot-com bubble collapse, the Web – which, it should be reiterated, is built essentially on FLOSS technologies – has generated more growth, more jobs and more market for ICT players of all kinds than any other recent rapid technology transformation. Between 1993 and 1999, about 750 000 new jobs were created in the US in the “information sector” that includes Internet publishing, software publishing; ISPs, and hosting[29]. Though EU statistics are less adequate in terms of sector and occupation categories, and that calls for a more in-depth analysis beyond the scope of this study, it is likely that the growth and employment impact of the Web in Europe was similar[30]. Only 18 months after the general introduction of Web access in 1993, Web traffic had become dominant on Internet backbones[31]. Napster and later P2P technologies for file sharing have undergone a similarly explosive growth, accounting for half of all Internet traffic in a few years[32], with a huge impact of the telecommunications market, though the general economic and employment impact is not yet comparable to the creation of the Web (see Figure 26). Even less disruptive technology or usage change such as blogs have a huge impact on the ICT economy, well beyond their direct economic importance.

---

[29]Source : US Department of Labor Statistics, Establishment date, historical employment, ftp://ftp.bls.gov/pub/suppl/empsit.ceseeb1.txt.

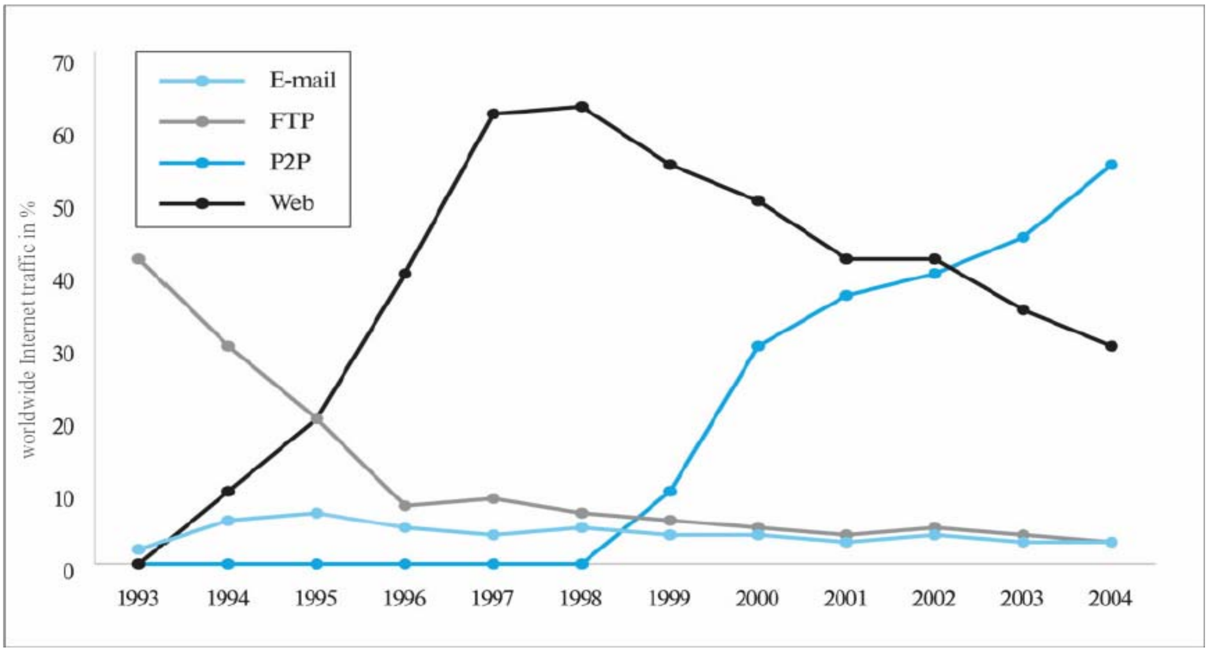
[30]As a conservative estimate for the US, one can substract the number of these jobs (about 200 000) that were destroyed during the dotcom bubble crash, still leaving a net increase of 550 000 jobs. In Europe, the Web impact on job creation has been partially hidden in total job creation figures due to some job destruction occuring in this period in non-ICT related industry and process activities.

[31]Internet growth: Is there a "Moore's Law" for data traffic?, K. G. Coffman and A. M. Odlyzko. *Handbook of Massive Data Sets*, J. Abello, P. M. Pardalos, and M. G. C. Resende, eds., Kluwer, 2002, pp. 47-93.

[32]Coffmannn & Odlyzko, op. cit.

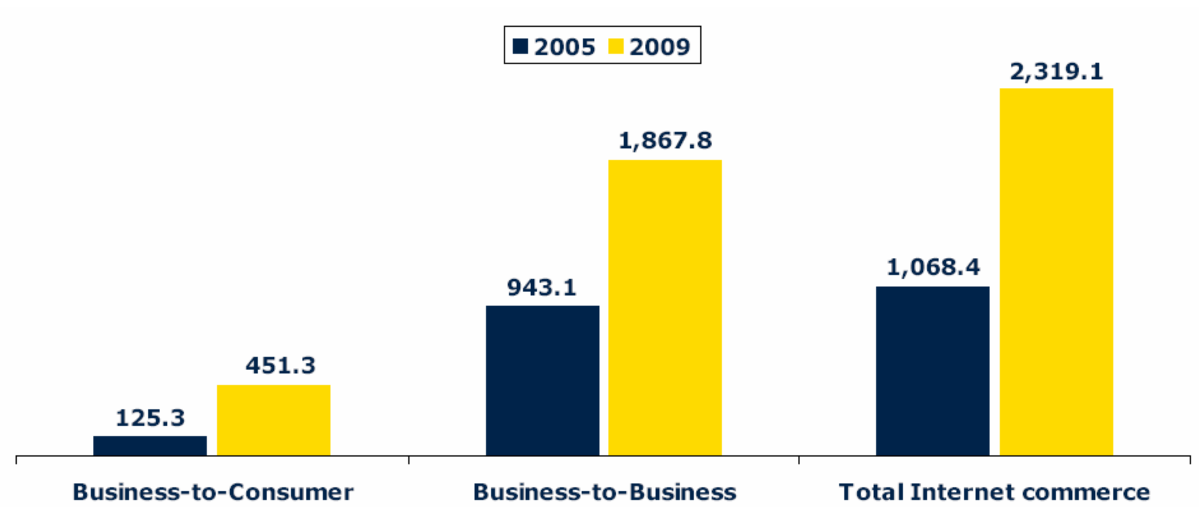
It is particularly important to consider for the importance of these indirect effects through new market / new activities creation for measuring the economic impact of FLOSS, as FLOSS has proved to be a major enabler for development of new usage, even when no immediate business model seems available (as illustrated by the examples of the Web, P2P, or RSS syndicated blogs). While the growth in Internet commerce services (Figure 27) is not directly FLOSS-related, it would not be possible without FLOSS-based infrastructure and thus can also be credited as part of the impact of FLOSS on ICT. The potential impact of FLOSS on the secondary (non-IT) industry and health, administrative or educational services is also particularly relevant, as a growth in these application areas can have a powerful feedback effect on the overall demand for software and ICT services (systems integration, solution providers, telecom).

Figure 26: Worldwide Internet traffic by type



Source: CacheLogic P2P in 2005 cited by EITO 2006, "P2P networks and market"

Figure 27: Internet commerce in Western Europe



Market in Billion Euros, for EU15 plus Norway and Switzerland. Source: EITO 2006, "ICT markets"

7.1.1. Summary findings

- Measuring the economic impacts of FLOSS needs to consider a variety of factors that determine the supply of and demand for ICT goods and services.
- FLOSS has direct economic impacts: it is credited with firms investing an estimated Euro 1.2 billion for development, adding Euro 263 billion in annual revenue and driving employment numbers. Just as with ICTs overall, it is difficult to assess the indirect impact of FLOSS. There is evidence suggesting FLOSS's influence on skills development and encouraging of growth of SMEs and employment.
- FLOSS on its own does not explain the performance of FLOSS contributing firms. However, in some of the sectors such as software producers, firms that contribute to FLOSS show better performance for some revenue indicators (e.g. revenue per employee) than firms that only produce other kinds of software.
- It would be a mistake to consider FLOSS contributors a fringe or minor player in today's economy. A conservative estimate is that the number of employees among firms contributing code to FLOSS projects is at least 570 thousand.
- Employment in ICT makes use of a wide range of skill levels and skill specializations and ICT in turn drives the demand for a wider range of skill sets in the labour force. It is important to consider FLOSS impact on direct and indirect employment.
- FLOSS, like other ICTs, drives demand for particular skill sets such as those related to software development. However FLOSS asks for skills from the formal infrastructure (e.g. computer software engineer) and also for skills related to FLOSS community actions including project management, copyright law and entrepreneurship. FLOSS may drive demand for a range of skill levels and specializations (e.g. formal education, information education, variety of specializations beyond core ICT training).
- FLOSS will drive demand for skills across all sectors of the economy, just as with ICTs in general.
- FLOSS may enhance employability given the training and life-long learning environment of the OSS community.
- FLOSS may help employers find the 'right people' for immediate job demands as they can turn to the OSS community and not have to rely solely upon formal and traditional infrastructure output (e.g. university graduates).
- FLOSS may save the employer's 'training' budget as FLOSS communities provide informal apprenticeships and training.
- FLOSS may contribute to a more dynamic ICT sector in Europe and provide additional and more attractive opportunities for young researchers and developers to remain in Europe.
- Mobility surveys tell of the influence on researchers to stay where they have developed a network and belong to a community, over the course of their studies. FLOSS is a strong community and it may be an instrument of retention of highly skilled workers for regional and national policy. It is too early to say this could slow the drain of ICT professionals to countries like the US but indications certainly suggest it is worth investigating further.

- Data to prove the impact of FLOSS on ICT use and innovation in SMEs is insufficient but one might argue that FLOSS uptake by SMEs are in themselves measures of innovative activity.
- There is clear indication that the availability of FLOSS drives ICT take-up among SMES, and the ICT take-up includes FLOSS and non-FLOSS software.

7.2. Primary production of FLOSS code

Software productivity and the estimation of primary production of the software itself is essentially the intersection of economics and software engineering, and in the case of FLOSS software it has two aspects:

1. what were the actual costs of developing a given set of software, in terms of time and effort and the equivalent in monetary terms?
2. what is the substitution cost of the same software – i.e. how much would it have cost to build the same software entirely within a single firm in a proprietary software development model[33].

The second aspect is useful to know about on its own and it provides data on the R&D substitution effects, as described later in this section. The substitution cost is calculated by using models widely used in the software industry for cost estimation of a given project on the basis of a range of factors such as the estimated size of the resulting software (lines of software source code), complexity of the software, time-critical nature, etc. The output of this standard cost estimation model (called COCOMO[34]) is the minimum number of programmers required, and the effort in person-months. Based on the person-months, it is possible using salary data for the ICT industry to estimate the total cost required to develop the given software product. Salary data is available from e.g. OECD data available at MERIT as well as publicly available data through Payscale.com that provides detailed data on salaries in different regions worldwide, based on skill level and job description. Salaries are used to compute the value of primary production on the basis of the geographical and skills distribution of developers, mapping person-months to monetary values by multiplying them with appropriate salary levels[35].

[33] Note that by the substitution cost of the software, we mean the cost if a firm were to develop the entire software. This is one way of placing a Euro value on the production represented by the existing base of FLOSS code available. We do not discuss in this section how much of the Debian software studied in this section would actually have to be developed by a firm for a given need as that would depend on specific cases. A further discussion of actual substitution of internal R&D is provided in section 8.3.1, “Nokia and Maemo: a FLOSS R&D substitution case study”.

[34] Boehm, Barry W., *Software Engineering Economics*, Prentice Hall, 1981. Until recently, this was the most widely used software project cost estimation model throughout the world, till its replacement by various versions of COCOMO-II, a model that requires parameters that cannot necessarily be estimated for a large base of existing software, and thus harder to apply to FLOSS.

[35] The cost in Table 2 and elsewhere in this report is very conservatively estimated as the unweighted average salary from Payscale.com for software engineers / developers / programmers with only 1 year of experience, for EU15 countries. The average is drawn down hugely by the unweighted inclusion Portugal and Greece. This average salary is multiplied by a factor of 2.4 estimated as overhead (for a discussion of the methodology, see Wheeler, David, 2001. “More Than a Gigabuck: Estimating GNU/Linux's Size”. Available online at http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html).

Table 2: Substitution / production cost for Debian collection of FLOSS applications

Code base collection	Debian 3.1 FLOSS distribution (2005)
Source lines of code	221,351,503
If Debian was written in a software company…	
Estimated effort	163 522 person years
Development cost estimate (till 2005)	Euro 11.9 billion
Development cost estimate (till 2010)	Euro 100 billion

Copyright © 2006 MERIT. Source: URJC estimates (cumulative effort estimation), payscale.com (salary data).

URJC has previously published rough estimates of the substitution costs for Debian, a large distribution of FLOSS software including the GNU/Linux operating system and a majority of all stable FLOSS applications and tools amounting to over 200 million lines of source code. A more detailed analysis is provided in Table 2, based on a more sophisticated version of COCOMO that takes into account a number of factors such as the code complexity, developer skill level required, reliability and type of software. These detailed parameters have been manually determined for the largest 100 FLOSS projects included within the Debian collection of FLOSS applications, and they account for over 35% of all the source code in the total collection. Thus, the table shows a very good estimate of the substitution cost of the entire publicly distributed FLOSS code base in production use today.

It should be emphasised that the quantity of publicly distributed FLOSS code (as measured by source lines of code, or SLOC) is doubling every 18 to 24 months. This has been a finding of our study on the Debian collection over an 8-year period, and is consistent with the evolution of other FLOSS software collections (such as for Java-based FLOSS code). Of course, this means that during the last 18-24 months as much FLOSS software code was created as in the entire previous history of FLOSS development. This refers only to “net” code, as much old code is rewritten or substituted with new code (a rough estimate based on a sample of FLOSS projects studied is that 50% of the code is replaced at least every 5 years). Thus, we can infer the size of the available codebase in 2010 (as the trend of the past 8 years is certainly not reducing, and unlikely to reduce in the next 4 years). From this, a lower bound for its substitution cost value of over 100 billion euro, as shown in Table 2[36].

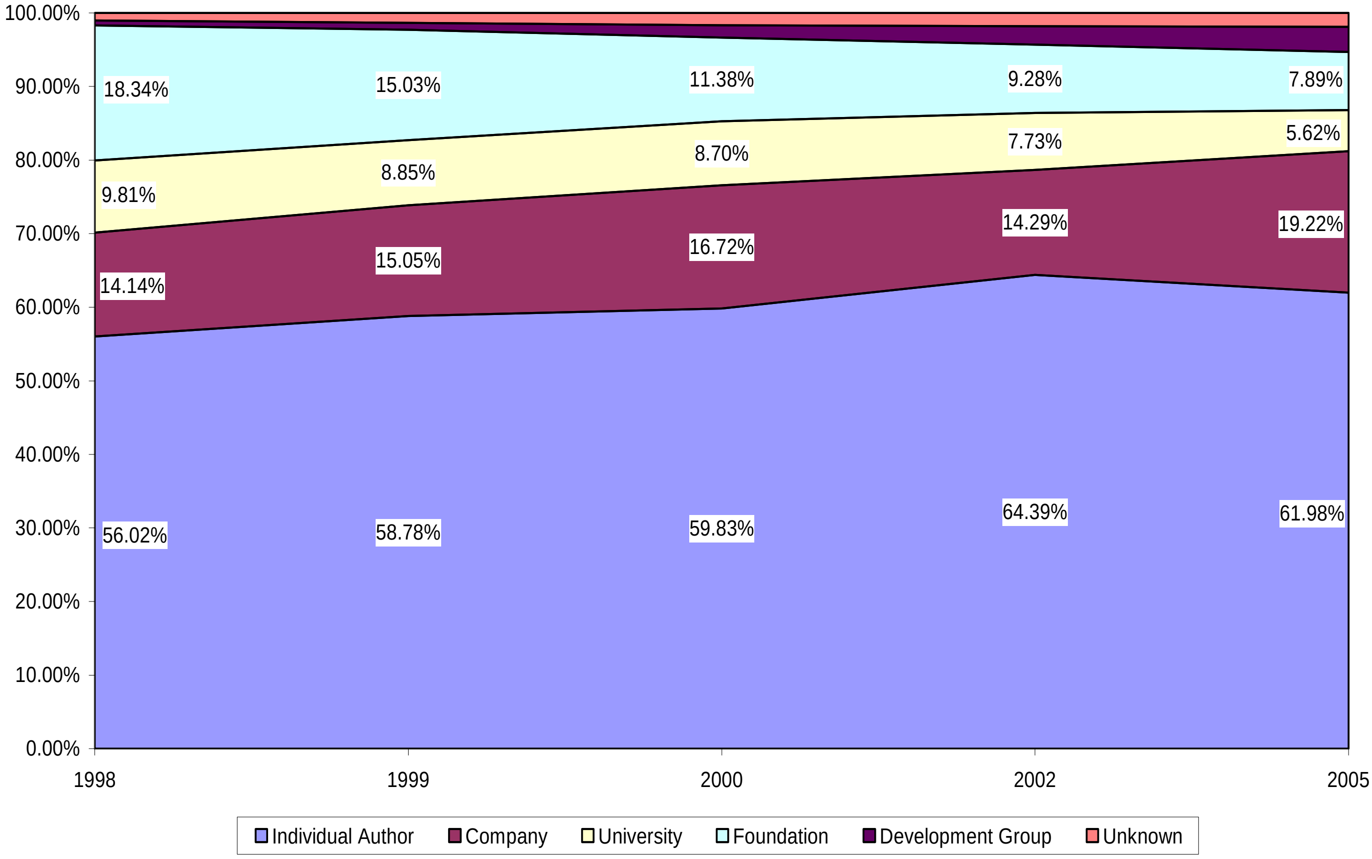
A sample is produced in Table 3 below for selected projects. Chronological data is available for the past several years, and Figure 28 shows code output stratified by type of contributor over a 7-year time period. This is based on copyright and credit claims, so it may under-represent contributions of individuals who (as many do) forget to claim credit or copyright, or assign it to a foundation or company. On the other hand, contributions from very small firms are often done in the name of the individual contributors or, for some projects (such as Zope) in the name of an umbrella “foundation”. Finally, employees of some companies may contribute without crediting their employers; it is not clear whether this is really a company contribution, though, unless the company makes a conscious choice not to claim credit or copyright of code contributed by employees.

[36] Note that the cost estimation function increases exponentially, but in the table we increase our projected cost estimation for 2010 linearly, thus it is certainly a conservative lower bound.

Table 3: Production cost estimate for 5 largest FLOSS software products

Software package	Lines of code	Months	Person-months	Cost (million Euro)
openoffice.org	5181285	130	79237	482
kernel-source-2.6.8	4033843	160	145036	882
mozilla (firefox)	2437724	87	25339	154
gcc-3.4	2422056	113	54048	329
xfree86	2316842	90	27860	169

Figure 28: Distribution of code output by individuals, firms, universities



Copyright © 2006 MERIT. Source: URJC. Shows code contribution share for the Debian collection.

Estimates of distribution of code contribution for Debian version 3.1 up to 2005 have been made for selected universities (including some European Universities) and are shown in Table 4. Preliminary estimates of the *value* of code contributed to FLOSS projects by selected firms, as well as the estimate for the value of code contributed by *all* firms, is shown in Table 5. It should be noted that firms themselves have few ways of knowing the value of their own contribution, except partially when they follow a policy of deliberately (and measurably) contributing to specific software projects. IBM, for instance, now estimates spending in excess of $100 million annually on Linux development, although this includes maintenance and forms of participation other than just writing code (this figure was reported by Doug Heintzmann Director of Technical Strategy, IBM Software Group at the “Open Standards and Libre Software in Government” conference organized by MERIT together with the Dutch EU Presidency in November 2004 and updated by IBM for this report).

Table 4: Contribution of FLOSS code by selected universities

Rank	Person-months	University / research institution
1	4955	Regents Of The University Of California
2	4774	Massachusetts Institute Of Technology
3	1687	Carnegie Mellon University
4	1340	University Of Chicago
5	1009	INRIA
6	982	University Of Amsterdam
7	870	Stichting Mathematisch Centrum Amsterdam
8	551	Ohio State University
9	518	University Of Utah
10	505	University Of Notre Dame

Copyright © 2006 MERIT. Shows estimated cumulative person-month contribution in Debian, by 2005

Table 5: Cost estimate for FLOSS code contributed by firms

Total contribution from firms			
Number of firms		986	
Source lines of code		31.2 million	
Estimated effort		16 444 person years	
Estimated cost		1.2 billion Euro	
Top contributors			
Rank	**Name**	**Person-months**	**Cost (mil euro)**
1	sun microsystems inc.	51372	312
2	ibm corp.	14865	90
3	red hat corp.	9748	59
4	silicon graphics corp.	7736	47
5	sap ag	7493	46
6	mysql ab	5747	35
7	netscape communications corp.	5249	32
8	ximian inc.	4985	30
9	realnetworks inc	4412	27
10	At&t	4286	26

Copyright © 2006 MERIT. Shows estimated cumulative substitution cost contribution in Debian, by 2006

We note that we have applied the above methodologies for substitution cost estimation not only to FLOSS projects for which proprietary software with similar functionality exist – e.g. OpenOffice – but also to cases of FLOSS software that do not have any direct equivalent in the proprietary world. Examples of this category of unique, innovative FLOSS software include mldonkey, the only application to provide connectivity to a large number of peer-to-peer networks for collaborative, distributed communication; and Plone, an extremely powerful interactive Content Management System. (Interestingly, both projects are primarily European in development and leadership, though used worldwide.)

The substitution cost also provides some indicator of the actual cost. But this is not an accurate representation of the actual cost in terms of time FLOSS developers have spent, as the COCOMO cost model is based on the costs of development in the *proprietary* software model, (in particular, a certain standard development environment), which is not necessarily the same time and effort, or cost, for FLOSS development. Determining the actual effort which has made open source software possible is not simple. Most cost and effort estimation models are designed for in-house development and classical development models. In FLOSS, there are many actors contributing directly to development, in addition to core developers.

However, estimating the effort by core developers is a first step towards estimating the effort society is putting into FLOSS programs.

We have made estimates of the real effort represented by the existing FLOSS code base using the following methodology: surveys to FLOSS developers (including answers about the effort they have devoted to specific projects during a certain period), and database of software version control data ("CVS") provides the number of lines, time of change and identity of developer for every line ever added or deleted for all software projects for which it is available. In short, the data from CVS is used to estimate the amount of code produced (or changed) during a certain period by specific individuals. This information is matched against individuals' time effort declarations in the surveys. The output of this study is a function mapping effort (in person-months) to the amount of produced (or changed) software code.

We then apply this mapping function in reverse to a certain percentage of developed FLOSS software (by analyzing the corresponding CVS data and estimating the code output). This way, the total effort put into its development in terms of time can be derived. In addition, the team size can be estimated, as well as the opportunity cost in monetary terms by computing the equivalent salary cost of the estimated time input of the developers. This is then be extrapolated for the total FLOSS code base published, controlling for differences between the sample studied in detail and the entire universe of the code base. The result is a first estimate of the value in effort of the entire primary production of FLOSS software, as well as the equivalent in monetary (opportunity cost) terms for the value of the primary production of FLOSS software (see Table 6).

Note that these figures are considerably lower than the substitution cost estimate provided in Table 2. While there may be many reasons for this, including the fact that the estimation methodology can be developed further, one important point is that the substitution cost estimate is based on a model that includes the full cost to employer of developer time. Software development requires much more than just programming, and all possible activities – i.e. full employee time – is included in the COCOMO estimate of substitution cost.

In general, FLOSS code is continuously renewed, and it has been found that in many large projects at least 50% of the code base is changed within a 5-year period[37]. Furthermore, active developers suggest that 58% of their development time is spent on coding, but this is an under estimation as developers tend to exclude essential non-coding tasks in "development time". (For reference, estimates suggest that programmers employed in the industry spend less than 50% of their time coding). Indeed, surveys of developers' range of activities show that programmers spend perhaps 33% of their development time with pure development tasks such as writing code and fixing bugs. As the code base is increasing exponentially, we estimate that true figure for effort from developers is therefore at least 131 832 person years from *programmers* alone, for the past 5 year period, or over 26 thousand full time equivalent in voluntary programmer time every year.

While estimating real effort, however, it is not possible to determine or even delimit the time spent on development, but not on actual programming. While survey respondents have been asked to discriminate between their total time spent on development and their time spent coding, and this has been included in the effort estimation, developers are likely to underestimate the time spent in activities they believe is not related to their code output.

[37] Robles, Gregorio, Gonzalez Barahona, Jesus M. and Herraiz, Israel. 2005. "An Empirical Approach to Software Archaeology", *International Conference on Software Maintenance 2005, ICSM 2005 Poster Proceedings*, ISBN: 963-460-981-3, Budapest, Hungary, September.

Indeed, if these estimates are to be used for accounting purposes, we can be certain that coding time represents opportunity cost (indeed, at a salary for a particularly productive programmer who doesn't spend time doing much else), other time input may be more reasonably treated as provided voluntarily at little or no cost.

Finally, it should be noted that there may well be a greater productivity for FLOSS developers; it is a matter of much discussion within the software engineering community. FLOSS developers appear to work independently, optimising their level of contribution with relatively low coordination costs. As the costs of developing software increase exponentially in the COCOMO model of proprietary software development mainly due to coordination costs – the larger the code base, the larger the team size and the greater the complexity – if these are avoided or substantially reduced with the FLOSS development model, it goes some way to explaining the apparently greater productivity of FLOSS developers.

Empirical findings have led to debate in the software engineering community about the nature of FLOSS development. FLOSS projects appear to violate many "laws" that state that projects must have sub-linear growth, assuming (like COCOMO) that effort grows exponentially with size, and thus – since effort cannot realistically grow exponentially, size cannot grow linearly. Evidence from the software engineering literature shows that many FLOSS projects *do* have linear or super-linear growth[38], suggesting a structural shift in the level of complexity. Indeed, our "real value" estimate shows effort increasing sub-linearly with code size, unlike COCOMO that shows effort growing exponentially.

The hypothesis here is that although the coordination and management skills required for large groups of people to develop FLOSS is significant, this is precisely because central coordination effort (and coordination costs) appear to be missing or significantly reduced. This is because the huge numbers of developers are not coordinated as a single team. There is considerable evidence in the recent software engineering literature, based among other things on social network analysis, that suggest FLOSS developers work independently in a highly modularised self-organising structure[39]. This substitutes complexity of coordination (which requires coordination costs) with a complexity of organisational structure (which may require a wider distribution of sophisticated coordination abilities and coordination systems, but less

[38] See e.g. Godfrey, Michael W. and Tu, Quiang. 2000. "Evolution in Open Source Software: A Case Study", *Proceedings of the International Conference on Software Maintenance*, San Jose, California, 131-142; Succi, Giancarlo, Paulson, J. W. and Eberlein, A. 2001. "Preliminary Results from an Empirical Study on the Growth of Open Source and Commercial Software Products", *EDSER-3 Workshop (co-located with International Conference on Software Engineering ICSE 2001)*, May, Toronto, Canada; Robles, Gregorio, Amor, Juan José, González-Barahona, Jesús M. and Herraiz, Israel. 2005. "Evolution and Growth in Large Libre Software Projects", *Proceedings of the International Workshop on Principles in Software Evolution*, Lisbon, Portugal, September, 165—174; Koch, Stefan. 2005. "Evolution of {O}pen {S}ource {S}oftware Systems – A Large-Scale Investigation", *Proceedings of the 1st International Conference on Open Source Systems*, Genova, Italy, July.

[39] See e.g. Crowston, Kevin and Howison, James. 2005. "The social structure of free and open source software development", *First Monday*, volume 10, number 2 (February), http://www.firstmonday.dk/issues/issue10_2/crowston; Wendel de Joode, Ruben van and Kemp, Jeroen. 2001. "The Strategy Finding Task Within Collaborative Networks, Based on an Exemplary Case of the Linux Community", http://opensource.mit.edu/papers/dejoode.pdf; Trung T. Dinh-Trong and Bieman, James M. 2005. "The FreeBSD Project: A Replication Case Study of Open Source Development", *IEEE Transactions on Software Engineering*, volume 31, number 6, pages 481-494, June; Mockus, Audris, Fielding, Roy T. and Herbsleb, James D. 2002. "Two case studies of Open Source Software development: Apache and Mozilla", *ACM Transactions on Software Engineering and Methodology*, volume 11, number 3, pages 309-346,

high-level coordination effort and thus lower costs). The structure ensures that nobody has (or needs to have) the "big picture" in order for a large system to be collaboratively built. This self-organising coordination structure appears ingrained in the evolution of FLOSS software projects, which as they grow larger tend to split into smaller sub-projects that operate fairly independently of one another, as has been studied for Apache, KDE and GNOME[40], among others.

Thus, coordination and management skills need to be very widely distributed in order for individuals to effectively contribute – and such skills are widely learnt in the FLOSS developer community, as discussed in section 7.4.1. Perhaps this wider distribution of skills means that the opportunity cost of FLOSS developers should be valued even higher than we do.

Table 6: Estimated real effort and opportunity cost for the FLOSS codebase

Code base collection	Debian 3.1 FLOSS distribution (2005)
Source lines of code	221,351,503
Number of developers	17000 (estimate)
Estimated effort (*coding time only!*)	43 944 person years
Development cost estimate (*coding only!*)	Euro 2.67 billion
Development cost estimate, 5-year period	131 832 person years
Full-time employee equivalent (2006)	26 000

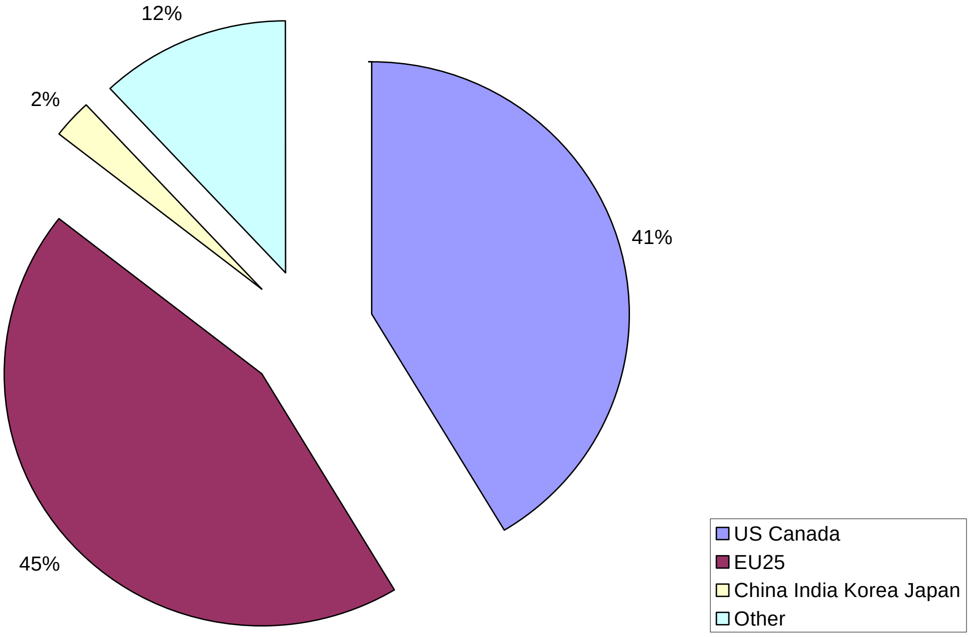
Copyright © 2006 MERIT.

As the cost estimation functions are very closely related to the extent of code contribution, the geographic distribution of investment in FLOSS development closely follows the distributions shown in section 6.3, "Demographics of FLOSS developers". Moreover, investment by region cannot be accurately estimated as estimation methods work for cohesive groups of developers, and most FLOSS projects have developers spread across several countries. However, Figure 29 shows the distribution of commits to FLOSS projects, grouped by region[41].

Figure 29: Share of code contribution by region

[40] Germán, Daniel M. 2004. "The GNOME Project: a Case Study of Open Source, Global Software Development", *Journal of Software Process: Improvement and Practice*, volume 8, number 4, pages 201--215

[41] Commits are a proxy for code contribution but every commit is not comparable



Shows number of commits. Copyright © 2006 MERIT. Source: URJC.

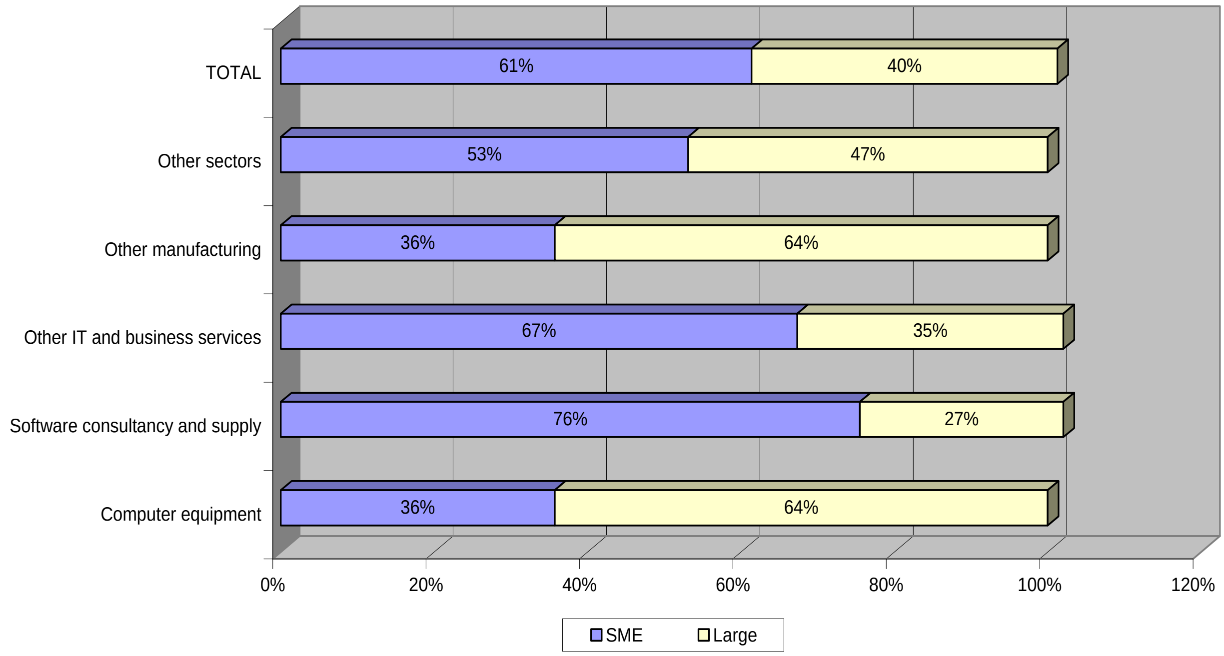
7.3. Firms developing FLOSS

These data have been estimated by the collation of information about contribution of individual firms developing FLOSS systems as identified from analysis of the source code and version control information. For 2005, as described in Table 5 on page 51, 986 firms were identified in Debian, a large base of publicly available source code screened for quality.

These data were combined with readily available information on firm size from, e.g. the Amadeus database of indicators for over 8 million firms.

What follows is a description of the results as implemented when matching the data on firms contributing FLOSS code to Debian (including, e.g. the time of contribution and the lines of code) with data from Buro van Dijk's Amadeus database (such as employees, sector and revenue). The company names identified in Debian were manually searched in the Amadeus database. Sometimes one company name from Debian returned multiple entries in Amadeus. For example, some of these were different companies having similar names while located in different countries, and others were subsidiaries of the same mother company. Amadeus is a European database, and some firms represented in Debian are located otherwise in the world. When the mother company was not located in Europe, the biggest subsidiary was selected. The size was determined in terms of operating revenue / turnover. If this information was missing, the number of employees was used as indicator. Further, consolidated data was preferred to unconsolidated data when both were available. In cases of multiple similar names, the web pages of the companies were consulted to determine which was the correct company.

Figure 30: Firms contributing FLOSS code: size shares by sector



Copyright © MERIT. Debian data from URJC; firm sector and size data from Amadeus and MERIT.
SME is the EU definition: <250 employees. n=158.

It should be noted that about 25% of the total firms have a European presence. In fact this number may be larger, as a high share of the firms not identified in European databases are probably very small (micro) firms. The results below are based on a relatively small sample (about 13%) of the total firms, for which financial and employee data was available in comparable years. This is nevertheless likely to be representative of the total firm contribution, with two biases: very small firms and non-European SMEs are under-represented; and some firms contributing to very specialised FLOSS applications that are not generally publicly available may be excluded in this sample.

Figure 30 above shows the share of small and medium enterprises (SMEs, using the EU definition of <250 employees) and larger firms among code contributors, by sector. It is clear that in general, SMEs dominate FLOSS code contribution for most sectors in terms of number of contributors (especially considering that our sample under represents SMEs among contributors of code), though among manufacturers, including computing equipment firms, larger companies dominate.

Table 7 shows the actual code output in our sample, by size and sector. Unsurprisingly, large firms – especially large computing equipment firms – provide the largest share of code. This is heavily biased by the code credited to Sun Microsystems (classified as a large computing equipment firm) and IBM (classified as software consultancy and supply). Sun alone, in particular, is credited with 30% of the total code contribution in our sample, which highlights one of the flaws inherent in the technique used for identifying company code contribution, which is based on copyright credits[42]. In the case of Sun, most of its contribution is accounted for by OpenOffice, for which Sun holds the copyright. The entire codebase of OpenOffice is not, in fact, Sun's sole creation, but contributors – individuals and other firms, small and big – sign an agreement assigning Sun joint copyright of their contributions, in order to simplify licensing and liability management[43]. Our algorithms only identify Sun's copyright message; a similar effect may bias the crediting of lines of code towards larger firms[44].

Table 7: Firms' code output (source lines of code) by size and sector

Size class: number of employees:	Small <51	Medium 51-250	Large >250	TOTAL
Computing equipment	0.11%	0.20%	34.35%	34.65%
Software consultancy and supply	2.77%	10.28%	25.60%	38.65%
Services - excl software consultancy and supply	1.11%	0.02%	3.35%	4.48%
Manufacturing - excl computer equipment	0.07%	0.03%	1.87%	1.96%
Other	0.73%	0.65%	18.88%	20.26%
TOTAL	4.78%	11.17%	84.05%	100.00%

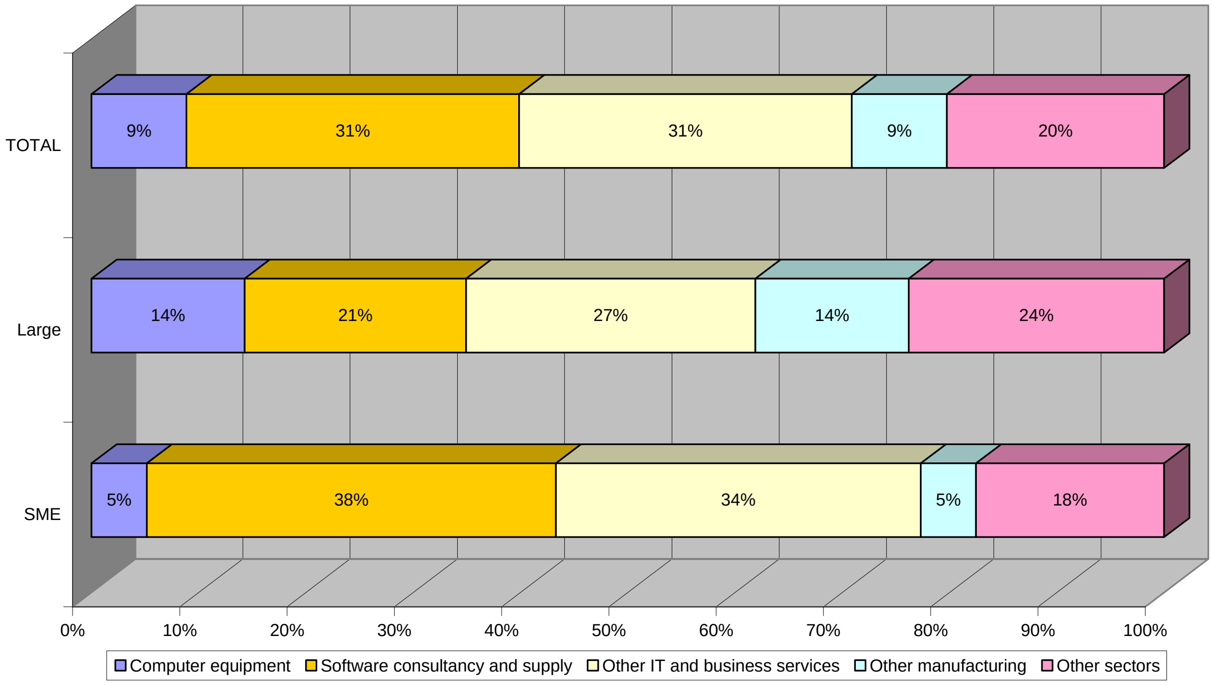
Copyright © MERIT.

[42] Other methods, such as analysing version control information, are possible to use, but not at the scale at which this analysis has been conducted; version control information is not available for all projects – but the source code is, and contains copyright statements and other authorship credit claims.

[43] Of course, as the software is open source, contributors are not obliged to sign the copyright assignment agreement, but the version included within the Debian collection, and measured here, is one where all copyright holders have assigned joint copyright to Sun.

[44] In some cases, this bias may work towards smaller firms, such as MySQL, which also asks for copyright transfer from contributors, and is classified as a medium-sized firm.

Figure 31: Firms contributing FLOSS code: sector shares by size



Copyright © MERIT. Debian data from URJC; firm sector and size data from Amadeus and MERIT.

As Figure 31 shows, though, despite the contribution of large computer hardware manufacturers (shown as “Computer equipment”) it is in fact the services sector that provides the bulk of FLOSS contributing firms (though not the bulk of actual contributed code), 62% evenly divided between software (product and service suppliers) and other business services (which includes business consulting, natural sciences and engineering research, and data processing, among other sectors). “Other manufacturing” provides a small share equal to computer equipment at 9%, and includes electronic equipment and telecom manufacturers as well as other manufacturing, such as paints. The sizeable “Other sectors” category, present both among large and small firms contributing code, indicates the surprisingly high share of firms among FLOSS code contributors from sectors seemingly unrelated to IT production, such as wholesalers and retailers of books. In fact, 27% of firms from “low IT intensity” sectors such as retail and tourism, when responding to the 2002 FLOSS User survey said that they “somewhat” or “totally agree” that their employees are free to work on FLOSS projects *during their time at work*. The rationale for this, as for the contributing of “Other manufacturing” firms, appears to be linked to their role as ICT *users* for whom software is normally a *cost centre* and not a source of profits nor a discriminating factor (see discussion of the reasons for IT user firms to contribute to FLOSS in section 7.7.4, “Maintenance cost sharing”). By using, participating in and contributing code to FLOSS projects they are able to share their development costs with other firms in their (and other) sectors. For firms with business strategies that are more IT dependent, the importance of services to their business (rather than software sold as a product) is likely to drive FLOSS contribution, hence the strong presence of software and other service providers.

7.3.1. Comparing FLOSS contributors to industry averages

Table 8 shows a comparison of the distribution of firms by size class for each sector as described above, discriminating between firms contributing FLOSS code to Debian and the industry average distribution for that sector. For each sector, the higher share is shaded grey. It is clear that the contributors to FLOSS code in our sample is skewed towards medium or large (rather than small) firms for most categories, except for manufacturing where the skew is particularly strong towards only large firms.

Table 8: FLOSS contributors compared to industry averages: firm size distributions

		% firms in sector for each size class		
	Size class: (number of employees):	Small <51	Medium 51-250	Large >250
Computer equipment	FLOSS firms	21%	14%	64%
	Industry	88%	9%	3%
Software consultancy and supply	FLOSS firms	52%	22%	26%
	Industry	96%	3%	1%
Services - excl software consultancy and supply	FLOSS firms	52%	14%	34%
	Industry	94%	4%	1%
Manufacturing - excl computer equipment	FLOSS firms	29%	7%	64%
	Industry	86%	11%	4%
Other	FLOSS firms	38%	16%	47%
	Industry	72%	17%	10%

Copyright © MERIT. Debian data from URJC; firm sector and size data from Amadeus and MERIT.

Table 9: FLOSS contributors compared to industry: average annual revenue, mil Euro

Size class: number of employees:		Small <51	Medium 51-250	Large >250	ALL
Computing equipment	FLOSS	3	25	4308	2877
	Industry	2	10	648	214
Software consultancy and supply	FLOSS	1	17	160	59
	Industry	1	14	163	59
Services - excl software consultancy and supply	FLOSS	5	19	5108	3539
	Industry	2	13	216	77
Manufacturing - excl computer equipment	FLOSS	7	25	6484	3769
	Industry	5	19	964	329
Other	FLOSS	3	17	3064	1306
	Industry	2	18	437	152
ALL	FLOSS	3	25	4308	2877
	Industry	2	26	682	237

Copyright © MERIT. Industry revenue for 2004. FLOSS firm European revenue for latest available years.

Table 9 shows average revenue for FLOSS contributing firms compared to industry averages for the size and sector (i.e. the 158 firms in the sample are compared to 171523 firms meeting the same size and sector classes). Note that the average revenue for all size classes combined (the right-most column) is much lower for the industry averages, since "micro" enterprises with 10 or fewer employees account for between 58% and 82% of all firms, except for the "Other" sector where this figure is 40%.

It is noteworthy that firms contributing to FLOSS almost always have higher average revenues than their peers. Table 10 illustrates this – other than medium computing equipment firms all firms contributing FLOSS have revenues several times above the industry average for their size and sector class. Of course, there is no implied causality here. Indeed, given the very small dependence on software as a revenue factor for most firms in most sectors of the sectors (manufacturing, non-software services and "others"), and the fact that other forms of software must also be produced by several of these firms (by definition, for the software supply sector), their FLOSS contributions are likely due to their increased IT awareness. For the IT sectors, though, FLOSS contributions are more likely to be part of a business strategy, and may affect revenues. This may be particularly true for small software service firms, among whom FLOSS contributors see between a 129% increase over the industry average for firms of similar sectors and size classes. However, the ratio of medium sized firms (51-250) shows a decline relative to the industry average, explained by a smaller firm size than average (see Table 11). This may be partly due to the fact that our data reflects European revenues, which can affect how non-European FLOSS-contributing firms are presented. Small non-European firms are likely not to appear at all; large firms are likely to appear with revenues and employees in proportion to their global revenues; medium firms with a European presence would appear in our dataset, but may have a disproportionately low revenue and employee numbers in Europe compared to the industry average – which, after all, mainly counts medium sized *European* firms with a presence in Europe.

The revenue ratio of service firms outside the software sector, especially medium and small service firms, suggests that such firms may be adopting FLOSS-related strategies, contributing FLOSS software to boost other revenues. As they are not in the software sector, they may face a lower chance of cannibalising their own commercial software revenue streams; when software is more of a cost centre, FLOSS strategies seem more attractive (see section 7.7, “Secondary production and services”).

Table 10: Revenue ratio: FLOSS firms over industry average

Size class: number of employees:	Small <51	Medium 51-250	Large >250	ALL
Computing equipment	65%	-7%	532%	1115%
Software consultancy and supply	129%	-40%	306%	262%
Services - excl software consultancy and supply	197%	45%	202%	177%
Manufacturing - excl computer equipment	209%	45%	2264%	4501%
Other	39%	33%	573%	1045%
ALL	57%	-6%	601%	758%

Copyright © MERIT. Difference in mean revenues of FLOSS firm to industry as a % of mean industry revenue.

As Table 11 shows, FLOSS contributor firms are also somewhat larger in terms of employees than the industry average for their sector and size class.

Table 11: FLOSS contributors compared to industry: average employees

Size class: number of employees:		Small <51	Medium 51-250	Large >250	ALL
Computing equipment	FLOSS	15	71	1595	946
	Industry	7	101	1431	513
Software consultancy and supply	FLOSS	15	53	759	290
	Industry	5	96	1047	383
Services - excl software consultancy and supply	FLOSS	17	96	1681	556
	Industry	4	97	988	363
Manufacturing - excl computer equipment	FLOSS	23	103	10273	7125
	Industry	9	102	1230	447
Other	FLOSS	15	125	18988	10777
	Industry	16	83	7232	3167

Copyright © MERIT. Industry employee numbers for 2004. FLOSS firm employees for latest available years.

It is worth examining the ratio of the difference between our FLOSS-contributing sample of firms and the industry average for revenue per employee, by size and sector. The fact that this is generally positive – on average, for all size and sector classes, FLOSS-contributing firms have a revenue per employee 221% above the industry average – indicates that revenue increases are not only a result of firm size. Indeed, this figure shows that revenue per employee increases even for medium-sized FLOSS contributing firms (indicating that their reduced revenue per firm in Table 10 is a result of smaller firm size – as shown in Table 11). For small software firms, though, the revenue per employee slightly declines, indicating that their increased revenue is mainly a result of increased employment.

Table 12: Revenue per employee ratio: FLOSS firms over industry average

Size class: number of employees:	Small <51	Medium 51-250	Large >250	ALL
Computing equipment	0%	25%	163%	182%
Software consultancy and supply	-9%	29%	1105%	427%
Services - excl software consultancy and supply	40%	39%	526%	211%
Manufacturing - excl computer equipment	18%	44%	143%	136%
Other	82%	-20%	202%	204%
ALL	20%	206%	377%	221%

Copyright © MERIT. Difference in mean rev/empl for FLOSS contributor to industry as a % of industry mean.

We should note that all firms in sectors such as software consultancy and supply are software producers by definition. All firms not contributing to FLOSS, and most if not all firms contributing to FLOSS, must also be producing non-FLOSS software. Note that they may not be producing proprietary packaged software generating licensing revenue, as the proprietary share of total software is small in terms of revenue and employment[45]. Data on the share of firms in other sectors who produce some software (the "secondary software" industry) is provided in section 7.7.2, including the share of such firms that incorporate FLOSS into their software products or services.

So, FLOSS alone is not an explanation for the performance of FLOSS contributing firms – as we noted earlier, we highlight the correlation, rather than identifying any causal relationship. But what is important to recognise is that – most clearly in some sectors such as software producers – firms that contribute to FLOSS perform better, on some measures of revenue or revenue per employee, than firms that produce only other kinds of software.

It is also important to recognise that contributors to FLOSS are not marginal players in the economy. Our sample of 158 FLOSS contributing firms for which employment and financial data was found indicates that firms that contribute code to FLOSS projects have in total 530 thousand employees with a total annual revenue of Euro 231.4 billion. A very conservative estimate for the total number of 986 firms identified as contributing code to publicly available global FLOSS projects, assuming that all the unmatched firms were SMEs and extrapolating from the data for our sample would suggest that firms contributing code to FLOSS projects have in total at least 570 thousand employees and annual revenue of Euro 263 billion.

[45] The share of proprietary packaged software in the total software market 19% in Europe and 16% in the US in terms of sales, as estimated by the FISTERA network as well as some national statistics bodies, see Table 24, "The software economy: sales, services and in-house" on page 124, and associated text. Similarly, its share of employment is also small. See 7.4.5 for data on US employment of software developers by sector.

7.4. Skills development and employment generation

The potential for FLOSS as a means of broader ICT skills development is important, as the skills learnt from FLOSS participation are broad-based, valued by employers in the ICT sector even when they don't use FLOSS, and most important – the skills training is not explicitly paid for. ICT has been recognised as a major source of economic and social change (OECD, 2003). Skills demanded by traditional occupations are changing (e.g. a long-haul truck driver who now uses global positioning systems; computer software integrated with beef farm operations) and new industries are emerging (e.g. web-based industries). Skills across sectors too are changing (computer programming occupations are not limited to traditional ICT sectors but are important elsewhere, e.g. in the financial sector). The extent to which Europe can participate in and benefit from the creation, production and diffusion of ICT depends a great deal upon its pool of knowledge workers. Given the pressures of globalisation, Europe needs to not only produce a supply of skilled workers with various levels of qualification to meet its own industry needs but must also pay attention to the potential losses through the outflows of highly skilled workers abroad as well as the potential gains to be made by attracting skilled workers from abroad. The Lisbon strategy recognises the need for Europe to be more attractive to foreign-born skilled workers (Lisbon Strategy, November 2004).

Jobs in ICT make use of a wide range of diplomas, degrees and other forms of training. Core IT occupations include computer scientists, computer engineers, computer programmers and systems analysts. Skills levels range from informal training to postsecondary certification to doctorate level researchers. Doctorate graduates are only a small segment of the ICT work force but they are the most advanced in terms of training and experience for research and are a catalyst for start-ups — they are highly sought after by Europe's competitors, in particular the United States who is having good success keeping Europe's fresh doctorate graduates and bringing European workers over to meet short term industry demands. Thus jobs growth in Europe are not only directly related to new businesses using FLOSS models, but also through increasing ICT skills in Europe due to FLOSS participation, and providing increased job opportunities through new FLOSS-related business models in order to retain skills within Europe (see Section 7.4.3, "Retaining ICT skills in Europe").

7.4.1. Developing local skills

The Free/Libre/Open Source Software (FLOSS) study in 2002[46], a comprehensive study of developers and users, showed that the most important reason for developers to participate in open source communities was to learn new skills — "for free". These skills are valuable, help developers get jobs and can help create and sustain small businesses. The skills referred to here are not those required to use free software, but those learnt from participation in free software communities. Such skills include programming (of course), but also skills rarely taught in formal computer science courses, such as copyright law and licenses (a major topic of discussion in many free software projects). Teamwork and team management are also learnt – after all, the team management is required to coordinate the smooth collaboration of 1500-plus people who rarely see each other is more intensive and far subtler than what is required to coordinate smaller teams employed in a single software company.

[46] Ghosh et al, 2002. FLOSS Final report, part IV. http://flossproject.org/report/

(Coordination and management skills required for large groups of people to develop FLOSS are significant, precisely because the central coordination effort and associated coordination costs appear to be missing or reduced in relation to standard software development models, as described in the text with Table 6. Thus, coordination and management skills need to be very widely distributed – and are apparently widely learnt within the community – in order for individuals to effectively contribute.)

Some findings from the FLOSS survey are appropriate here: 78% of developers join the free software community "to learn and develop new skills"; 67% continue their participation to "share knowledge and skills". These learnt skills have economic value to developers – 30% participate in the free software community to "improve … job opportunities"; 30% derive income directly from this participation and a further 18% derive indirect income – such as getting a job unrelated to free software thanks to their previous or current participation in free software developer communities. Being a Linux kernel developer proves a certain level of skills in many ways far better than having a computer science degree from MIT; employers benefit from such informally learnt skills. 36% of organisations polled in the FLOSS User Survey "totally" or "somewhat" agree that employees can work on free software projects on employer time. These are not necessarily IT companies – 16% of low IT-intensity companies (e.g. retail, automobiles, tourism, and construction) "totally" agreed with this point.

7.4.2. Informal apprenticeships – employers benefit, but don't pay the cost

FLOSS communities are like informal apprenticeships – but the apprentice/students and master/teachers contribute their own time "for free", without any monetary compensation for the training process. There is certainly a social cost, but it is borne voluntarily by the participants themselves and not paid for directly by those who benefit (such as current or future employers, or society at large). Everyone can benefit equally from this training – any employer can hire someone informally "trained" through participation in the free software developer community. However, not everyone invests equally in it. As many "teachers" may have been formally trained at university or at work, which is explicitly paid for, explicit costs are being borne for some proportion of community participants who have been formally trained.

In the larger perspective, this training system where all parts of society benefit from the products of the system, but only some explicitly pay for it, represents a subsidy – or technology transfer – from those who pay for formal training to those who do not (or cannot). Within countries, this represents a technology transfer from big companies who often formally pay for training to small and medium-sized enterprises (SMEs), who can less afford formal training expenses. Globally, this represents a technology transfer from the usually richer economies who can afford formal training, to the usually poorer ones who cannot.

There are also sector benefits, especially within poorer economies. Poor countries may have formal computer training during computer science degree courses, but perhaps not in other subjects, such as biology. Anecdotal evidence (in the case of biology, from India) suggests that the use of free software platforms during formal training in non-computer subjects may encourage informal learning of computer skills by students, thereby increasing their understanding of their own course subject (by better being able to conduct biology experiments through more sophisticated computer analysis). FLOSS usage can thus provide students of other subjects to informally learn computer skills, programming skills and enhance their competence in their formal training.

The term “students” should not be misinterpreted in this context, as “students” in FLOSS communities are often much younger than students at university, as indicated in Table 13, which shows the average age of the FLOSS community members at the time of joining the community (as surveyed in the FLOSS developer survey 2002).

Table 13: Average entry-age of FLOSS community members (% for each age cohort)

	Period / Year joining the FLOSS community										
Age when joining the FLOSS community	**1950 - 1985**	**1986 - 1990**	**1991 - 1995**	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**	**Total**
10 - 15 years	16,1	12,2	10,2	5,7	4,6	5,6	5,2	6,6	0,8		**6,6**
16 - 18 years	27,4	17,6	15,7	24,2	22,0	20,1	13,6	16,5	10,1	8,1	**17,1**
19 - 21 years	19,4	25,2	24,9	22,2	32,6	26,2	27,3	19,1	28,0	16,2	**25,1**
Total 10-21 years	**62,9**	**55,0**	**50,8**	**52,1**	**59,2**	**51,9**	**46,1**	**42,2**	**38,9**	**24,3**	**48,8**
22 - 25 years	11,3	24,4	25,7	26,3	22,5	26,9	25,8	30,5	28,8	32,4	**26,3**
26 - 30 years	21,0	12,2	12,4	13,4	14,2	12,3	17,6	18,2	17,9	27,0	**15,2**
Older than 30 years	4,8	8,4	11,0	8,2	4,1	9,0	10,6	9,1	14,4	16,2	**9,7**
Total	100	100	100	100	100	100	100	100	100	100	**100**

N = 2402 $p < 0.01$; Contingency coefficient: .225 Source: FLOSS Survey (Ghosh et al. 2002)

The fact that the shares of very young starters in the community decreases over time should not be overestimated, since this is to a considerable degree due to the fact that earlier years of joining are grouped together (in the columns) whereas figures are shown for single years of joining from 1996 onwards. In fact, 41% of those active in the FLOSS community in 2002 were between 14 and 23 years old.

Table 14: Project experience by age cohorts

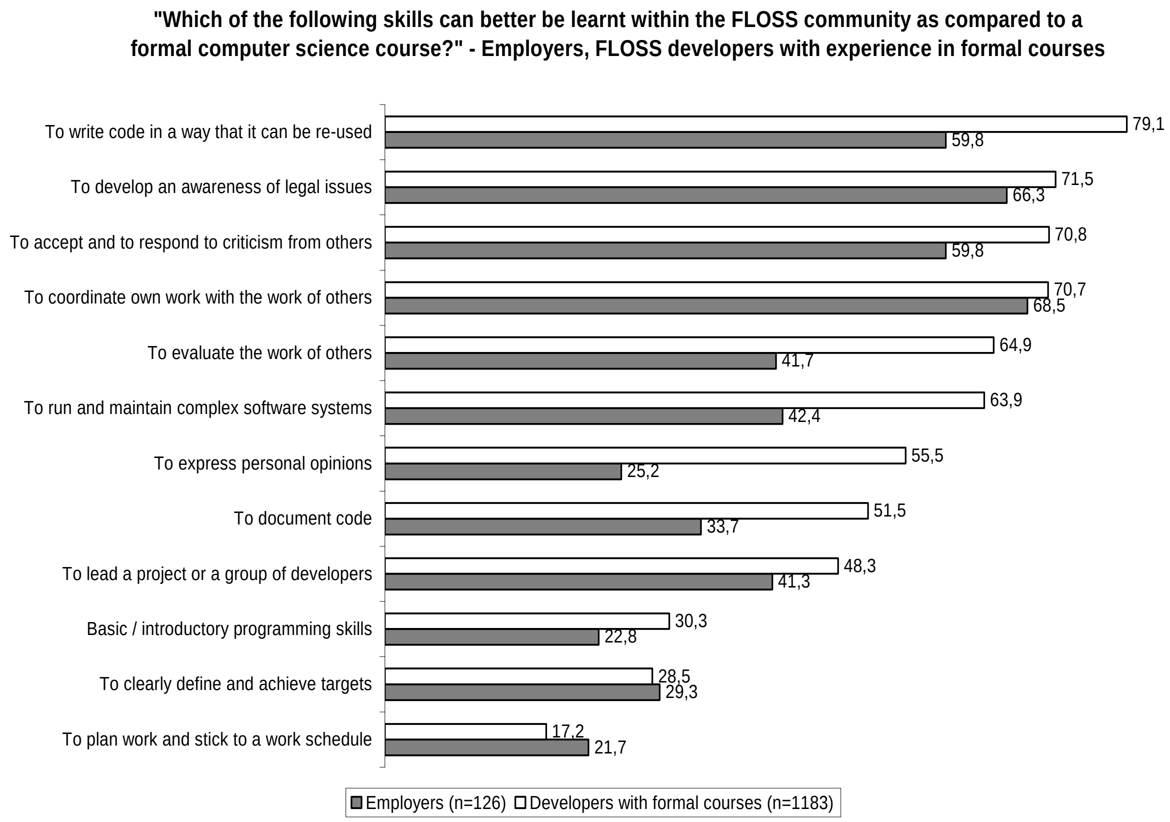
		Average (mean) number of …		
		present projects	all projects since joining community	led projects
young (15-25 years)	novices (0-3 years in community)	1,98	9,21	0,84
	semi-experienced (4-5 years in community)	2,36	6,63	1,63
	experienced (6-7 years in community)	3,00	10,13	1,26
	experts (more than 7 years in community)	3,73	17,64	2,12
middle-aged (26-32 years)	novices (0-3 years in community)	3,08	14,53	0,61
	semi-experienced (4-5 years in community)	2,87	5,16	1,23
	experienced (6-7 years in community)	3,25	7,83	1,40
	experts (more than 7 years in community)	3,50	9,69	1,82
"old" (33-66 years)	novices (0-3 years in community)	2,00	3,26	0,65
	semi-experienced (4-5 years in community)	2,37	5,96	0,57
	experienced (6-7 years in community)	2,28	6,18	1,12
	experts (more than 7 years in community)	3,41	10,09	1,67
Total		**3,02**	**9,06**	**1,44**

n = 1453 Source: FLOSSPOLS developer survey, Ghosh & Glott 2005

The FLOSSPOLS survey in 2005 revealed that young community members have the same opportunities to become experts and to take on responsibility for projects as older community members (see Table 14). As the table shows, “young experts”, who were 18 years or younger when they joined the community, show a comparable degree of current project involvement (i.e. number of actual projects when the FLOSSPOLS survey was carried out) as middle-aged and "old" expert. They show a significantly higher degree in overall project experience (i.e. on average they participated in more projects than the two older cohorts), and they also show a significantly higher degree of experience in project leadership than the two older cohorts. The performance of the young cohort is partly explained by the dynamics of the FLOSS phenomenon within the last ten years, which probably led to the provision of comparably more opportunities to test software, participate in workshops and projects etc. than the older cohorts found when they joined the community. Nevertheless, what becomes evident from these two tables is that the FLOSS community provides an extremely efficient learning environment, as well as teamwork and leadership experience, for young people even at an age when formal courses are not yet an opportunity.

More than programming skills, however, participation in FLOSS communities teaches other skills that are often not learnt in formal computer science classes: management and teamwork, understanding of legal issues, and general linguistic skills. Interestingly, in parallel surveys carried out by MERIT[47] of a large sample of community participants and HR managers at employer firms, respondents indicated that several skills were better learnt through FLOSS community interaction than through formal courses (see Figure 32)

Figure 32: Skills learnt from FLOSS compared to formal courses



47 FLOSSPOLS Skills Survey – Ghosh and Glott, 2005, see flosspols.org. . The number of respondents shown here differs from the cited report as for the report a sub-sample of developers was used, consisting of respondents who also participated in the FLOSS developer survey in 2002. In this report we use the full FLOSSPOLS developer sample. However, as indicated in Figure 32, developers without experience of formal courses are *excluded*.

 Excluding developers with no experience of formal courses.

The survey showed that developers believe that the skills they learn through FLOSS communities can compensate for a lack of a formal degree (see Figure 33). Employers believe a number of skills are learnt *better* in the "learn-by-doing" methods of participation in FLOSS communities than in formal courses (see Figure 32) – unsurprisingly developers are even more positive than employers about this. Furthermore, 55% of employers say that employees with proven FLOSS development experience would be paid as much as employees with a formal degree and 13% even say they would pay employees with FLOSS development experience more than those with a formal degree.

We acknowledge that for most skills (8 out of 12) a majority of firms felt that formal courses are better than FLOSS. But we emphasise that in itself, the fact that for many skills the majority of employers believe formal education is superior to learning from the FLOSS community is hardly the point – one would expect this, indeed one would expect *all* employers to think so. The efficacy of the formal education system is not really being questioned; it is FLOSS that has been (so far) unproven as an efficient method of skill straining. What we provide as significant evidence for the value of FLOSS as a skills training method is the *significant minority* of employers that believe FLOSS teaches skills better.

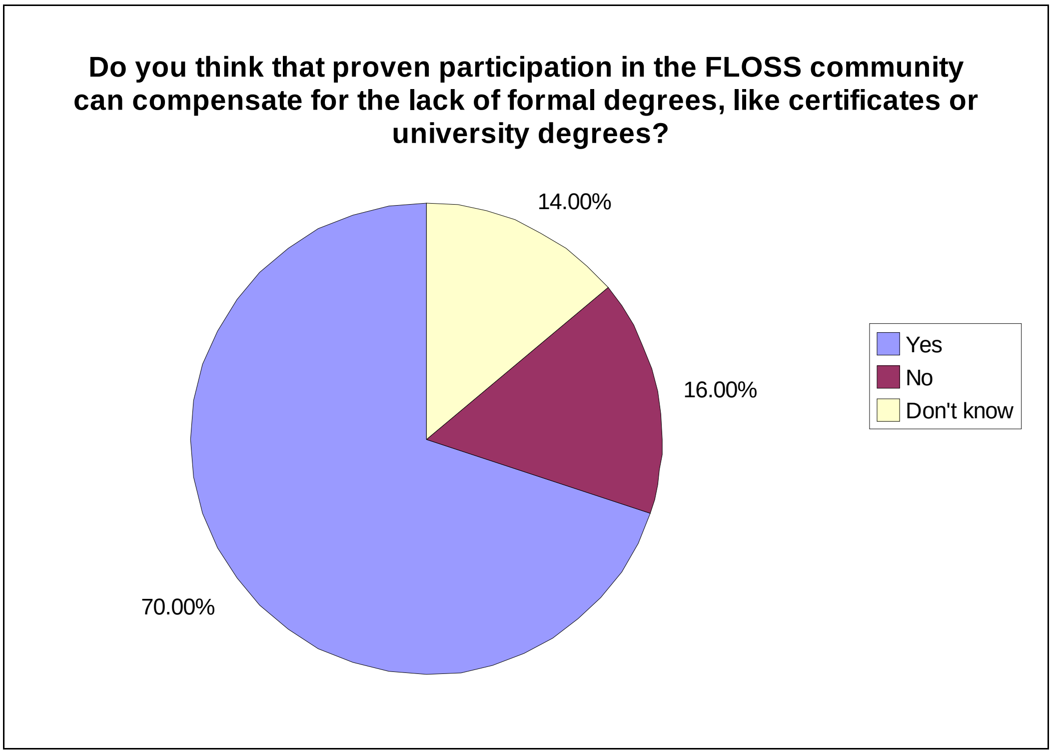
Moreover, the absolute share of skills firms report as being better learnt in formal courses is meaningless in itself. While the majority of employers believe that the numbers of skills that are better learnt in FLOSS communities are fewer, the *level* of these skills is relatively high. These are complex technical, management and legal skills (writing reusable code, understanding of licensing, coordination of work with other people). In contrast, the levels of skills the majority of employers believe are better learnt in formal courses are relatively basic technical and management skills (basic programming, planning and working according to a schedule, defining and achieving targets).

These skills are learnt through participation in FLOSS communities and investigation, tinkering with and sharing of software that is possible due to FLOSS licences. But it should be emphasised that the skills, once learnt, are not specific to FLOSS and are valuable technical, management and general skills useful within the general ICT sector occupations including in the development of commercial or proprietary software. Indeed, as Figure 52, "FLOSS developers also develop proprietary software" 113 shows, about half of FLOSS developers also develop proprietary software. (Though this share may be declining as more job opportunities have become available in FLOSS development since 2002, when this survey finding was made).

Furthermore, there was no significant difference between the response of employers who are *not* FLOSS users (and therefore are certain to need other forms of software including proprietary software) and employers who are FLOSS users (and therefore may need some FLOSS-specific experience in their workforce) when it came to the value of FLOSS experience versus a formal degree when it came hiring. A majority (54%) of both groups of firms said they would pay people with FLOSS experience but no formal degree the same as those with a formal degree but no experience[48].

[48] It could be argued that if firms value experience more than a formal course, this should also apply to people with experience developing proprietary software with no formal degree; perhaps, but this begs the question of how people get experience developing proprietary software in the first place. That typically requires being hired by a firm, first (or starting your own) where the experience comes at the cost of the firm; with

Figure 33: Skills learnt through FLOSS compensate for lack of degree



Copyright © 2005 MERIT. Source: FLOSSPOLS survey of developers (preliminary results)

Finally, 85% of the employers who use some FLOSS and are thus (more) likely to be aware of how its developer community works (n=115) think that FLOSS experience adds value to formal computer science experience, while only 6% think that no such value is added (9% don't know).

This skills development effect is clearly closely tied to employment generation, and this effect is heightened by any public support of the open source software sector. For example, the take-up by the Extremadura Region in Spain of FLOSS through its support for the LinEx project (a localised, Spanish-language version of the GNU/Linux operating environment) has led to an economic regeneration in a relatively poor region of the EU. This has not just allowed the implementation of activities for a lower price, but activities especially in education and training that which would simply not be possible with proprietary software; it has also led to the growth of a number of small businesses to provide commercial support, since with open source there is no need to approach vendors for support – approaching local entrepreneurs is possibly and an obvious choice (the economic impact of these policies in Extremadura was subject to a study being conducted by MERIT for the regional government, the results of which are briefly summarised in section 7.5.4).

This evidence presented above on skills leads directly to possible policy measures, especially in relation to the use of FLOSS in educational environments. For instance, designing educational environments in an open way, so that students find enough leeway for participating in such self-organised learning processes, would be a valuable measure in order to increase the positive net effect of these skills effects. This would require to ease restrictions of tight curriculae and to find solutions for testing and evaluating learning

FLOSS, as has been clearly demonstrated with the cohort analysis, people gain experience of a level and at a scale – and at an age – that is hardly possible through other means.

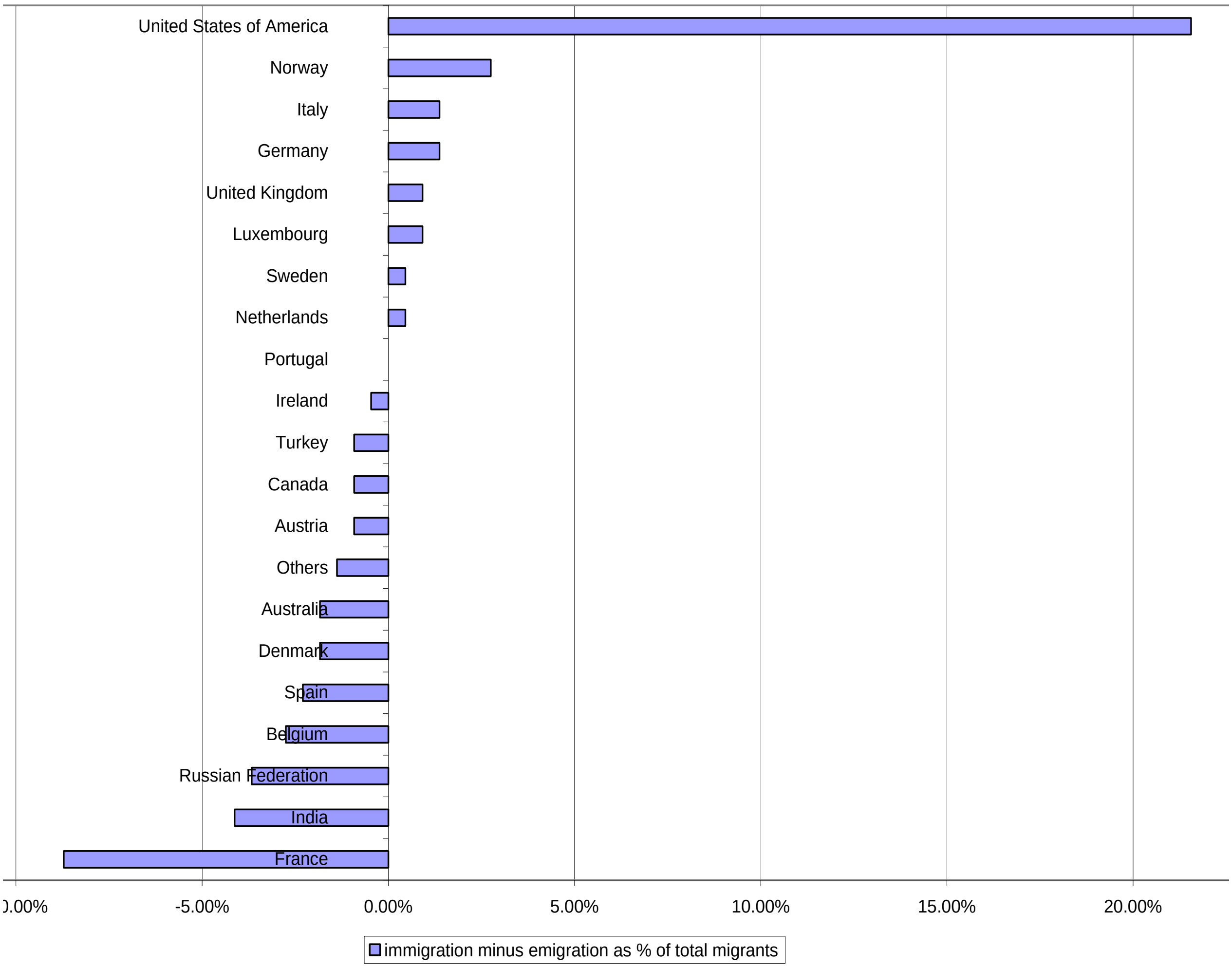
progress of students that has not been organised by formal educational institution, but in the context of formal education (in the sense of education that leads to a formal degree). Another useful measure would be to increase the acceptance of these informally attained skills in those sectors of the economy that are not familiar with informal learning environments. Measures for the control of learning progress appear to be a crucial factor in this respect, too. Finally, the simple inclusion of FLOSS-based tools and development environments is likely to lead to participation within the FLOSS developer community, especially in an educational environment, thus adding value to the skills learnt by students – value that employers recognise.

On a related note, the more widely awareness of FLOSS development methods is among employers, the more employers are likely to appreciate that proven FLOSS experience is a proof of ability, whether or not FLOSS software itself matters (or is used by) these employers.

7.4.3. Retaining ICT skills in Europe

Skills development is an extremely important aspect of increasing and retaining the skilled workforce and skilled employment in Europe. Several studies have shown that the EU is a net exporter of highly skilled personnel, especially to countries like the US. Figures from the FLOSS survey show that even in FLOSS development, while Europe does account for the largest share of developers, slightly fewer than 10% of all developers do not live in the country of their nationality. Of these, the largest gap between emigration and immigration percentages is for the US – i.e. only 5% of mobile developers leave the US, while 26% of mobile developers move to the US. The main countries accounting for international mobility among FLOSS developers are shown in Figure 34.

Figure 34: Emigration versus immigration of mobile FLOSS developers, selected countries.



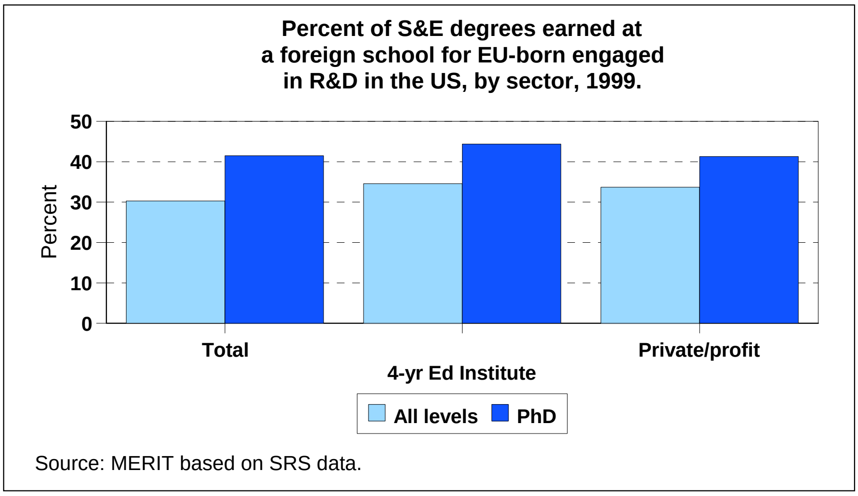
Copyright © 2005 MERIT. Source: FLOSS survey of developers.

In terms of ICT skills and employment in general, previous studies and data available at MERIT[49] show that the US relies upon foreign degrees at all levels. In 1999, more than 17,000 of the EU-15 persons engaged in R&D in the US had earned his/her degree outside of the U.S. While we do not know which country supplied the researcher, we know the people were born in the EU-15 and brought his/her degree to the US.[50]

[49] Hansen, Wendy, 2004. *The Brain-Drain - Emigration Flows for Qualified Scientists.* MERIT. Prepared for European Commission DG Research. Available at http://www.merit.unimaas.nl/braindrain/

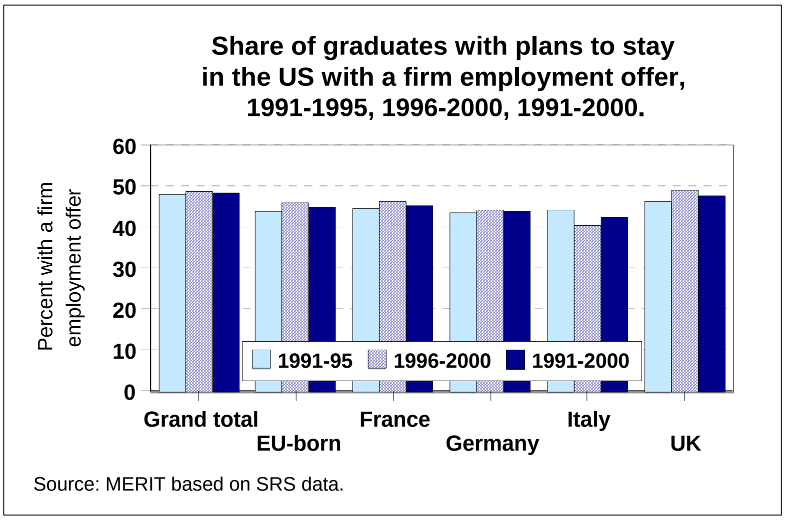
[50] (This figure is also probably even under-representative given the database is only refreshed every ten years and so the 1999 figure does not include Europeans who entered the Science and Engineering (S&E) work force over the previous decade).

Figure 35: US relies on EU-born skilled persons for R&D



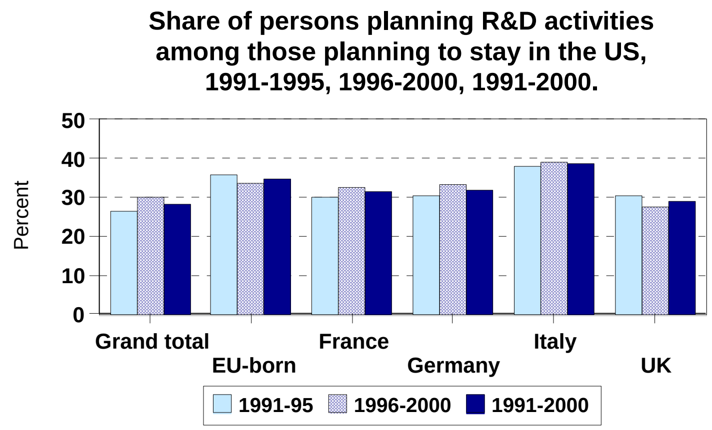
Apart from the direct drain of very highly skilled persons to the US from the EU in science and engineering, there is another important dimension to the loss of scientific and technical workers to the US, which is the loss of fresh graduates. Figure 36 shows the share of EU-born doctorate graduates who, when surveyed upon graduation, indicated they had plans to stay in the US on the basis of a firm US employment offer. Throughout the decade 1991 to 2001 almost half of all fresh EU-born graduates in the US indicated they had plans to stay in the US, with more than 45% of those from the EU-15 saying they would not to return to Europe – as high as 47% of those born in the UK. This is a powerful demonstration of how the U.S. continues to rely upon foreign talent, and how the EU continues to lose talented persons.

Figure 36: Share of EU-born US graduates with firm plans to stay in the US



The graduates were asked about type of work activities they were planning on pursuing and the chart below shows the share that had intentions of going directly to R&D activities.

Figure 37: Share of EU-born US graduates planning to stay in the US for R&D

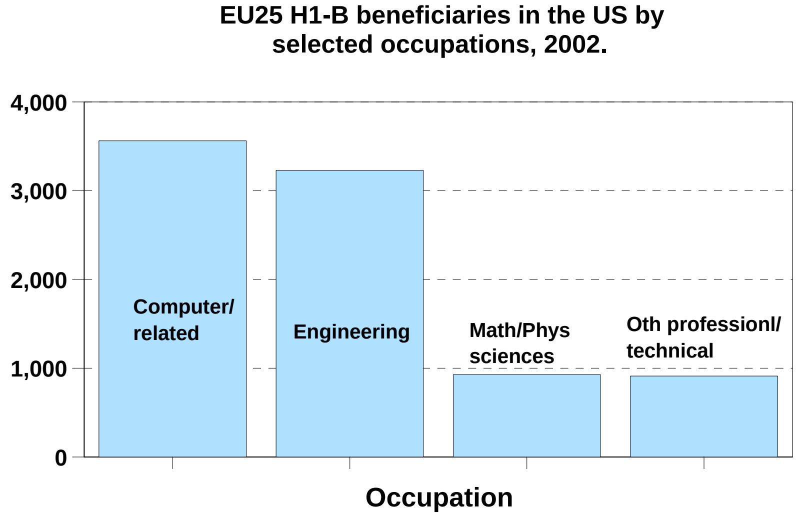


Source: MERIT based on SRS data.

Although the graduates are asked of their 'intentions' and perhaps not all ended up in the jobs as indicated, the data provides a powerful indicator for the challenges Europe faces in terms of not only attracting additional talent from abroad but even at drawing their natives back to Europe once they graduate in the U.S.

This shows us how the U.S. is creating new skilled workers for scientific and technical fields in engineering and computer sciences. But the U.S. has also succeeded in bringing foreign trained workers in to meet the short-term demands of industry, in particular in 'hot sectors' where the production of skilled workers can lag behind the demand. The obvious case was the ability of the U.S. to fast-track IT professionals and it did so with great success. Other countries did try to follow suit (e.g. Germany's Green Card), but the capabilities of the U.S. to capitalise on worldwide supplies of ICT professionals as well as other scientific and technical skills seem unmatched. Figure 38 shows that in 2002 alone, the U.S. brought in more than 3,500 skilled workers from the EU-25 to fill its needs in computer and related occupations.

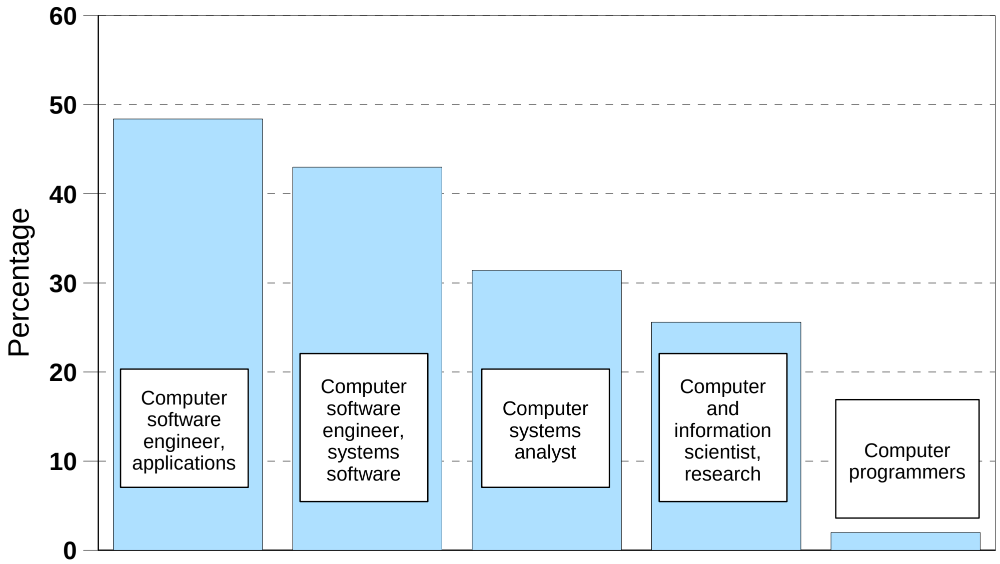
Figure 38: US attracts 3500 computer related employees from EU in 2002



Source: MERIT, based on USCIC data.

There seems little doubt the U.S. will continue to exert pressure on global supplies of skilled workers for occupations in computer and mathematics, the group of occupations that includes computer scientists, computer software engineers (applications), computer software engineers (systems) computer programmers and computer systems analysts. According to the US Bureau of Labor Statistics' projections for employment growth by occupations (Figure 39), occupations in computer software engineers (applications) and computer software engineers (systems software) rank among the top ten.[51]

Figure 39: Change in total software developer employment, 2004-2014, United States



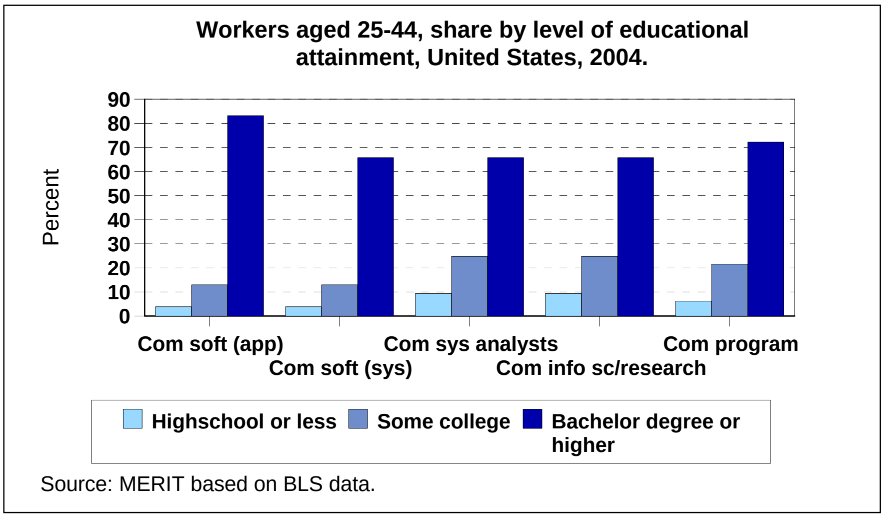
Source: MERIT based on U.S. Dept of Labor.

Occupations in computer software engineers (applications) are projected to see an employment change from 460,000 to 682,000 or an increase of 48.4% and occupations in computer software engineers (systems software) have employment projections showing growth from 340,000 to 486,000, a 43.3% rise in employment.

All of the occupations but for computer and information scientist (research) show a Bachelor or Associate's degree as the training category. That said, not all of the workers in these occupations in 2004 had a postsecondary qualification at the university level. Figure 40 shows the educational attainment of workers in the occupations related to software development.

[51] Source: *Employment Outlook: 2004-2014, Occupational employment projections to 2014*, Monthly Labor Review, November 2005, U.S. Department of Labor

Figure 40: Change in software developer employment by qualification, 2004-2014, US



There are also indicators available on the source of training for persons employed in these occupations and this has particular relevance to the open source community.

There is also information on self-employment of skilled workers in these ICT occupations. The highest share was reported by persons employed in computer and information scientists (research) with 5.1% reporting self employment in 2004, followed by computer systems analysts with 5.0% reporting self employment. The lowest share that reported self-employed was by computer software engineers (applications) with 2.4%.

Among the 460,000 or so computer software engineers (applications) in the U.S. in 2004, about a third were employed in computer system design/related services. The other main employers were software publishers, management of companies and enterprises, management and technical consulting services, and insurance carriers. For the 340,000 or so computer software engineers (systems software), it was also the case that most were working in computer systems design/related services or software publishers.

Salary levels were not commensurate with employment intensity. Table x gives a matrix of employment of the computer software engineers. Annual mean wages are given based on the industry ranking by highest level of employment and for industry ranking by highest level of pay.[52]

[52] Source of data: tabulations downloaded from U.S. Department of Labor, www.bls.gov, September 2006.

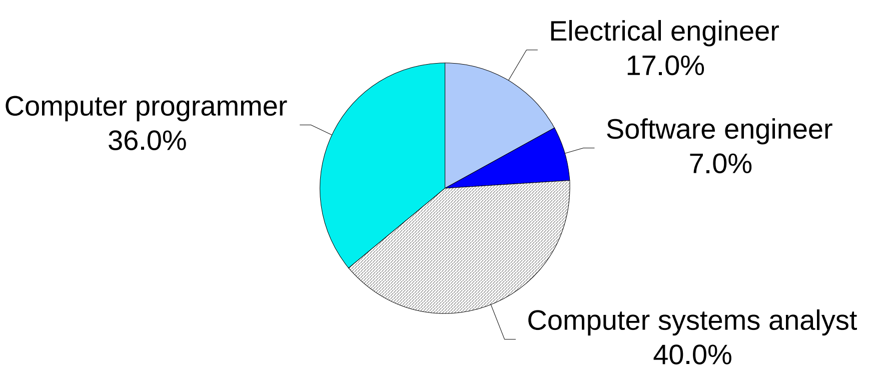
Table 15: Wage profile of software engineers, United States, 2004

Computer software engineer – systems software			
Industry ranking: by highest level of employment	Annual mean wage	Industry ranking: by top paying	Annual mean wage
1. Computer systems design/related services	$84,630	1. Scientific research and development services	$97,240
2. Software publishers	$87,820	2. Computer and peripheral equipment manufacturing	$92,030
3. Scientific research and development services	$97,240	3. Druggists' goods merchant wholesalers	$91,900
4. Computer and peripheral equipment manufacturing	$92,030	4. Magnetic media manufacturing and reproducing	$91,810
5. Data processing and related services	$76,950	5. Specialized design services	$91,700
Computer software engineer - applications			
1. Computer systems design/related services	$79,790	1.Computer and peripheral equipment manufacturing	$94,760
2. Software publishers	$83,390	2, Securities and commodity contracts brokerage	$90,950
3. Management of companies and enterprises	$76,210	3. Securities and commodity exchanges	$89,780
4. Management and technical consulting services	$81,950	4. Other financial investment activities	$89,100
5. Insurance Carriers	$73,330	5. Lessors of non financial intangible assets	$88,670
Source: MERIT, based on data extractions of BLS, www.BLS.gov September 2006.			

The U.S. is not alone in drawing talent away from Europe. Although on a much smaller scale, Canada too relies upon Europeans to meet short-term needs in key sectors. The figure below shows Europe is contributing to meet short-term needs for skilled workers in ICT occupations in Canada. Over three years, for example, the EU-15 provided more than 1,000 computer systems analysts.

From 2000 to 2002, more than 3,500 ICT skilled persons from the EU-25 landed in Canada for residency. Among these people, 40% were computer systems analysts and 36% were computer programmers.

Figure 41: EU-25 ICT skilled persons granted landed status in Canada, 2000 to 2002.



Source: tabulations of CIC, 2006.

7.4.3.1. Skills and employment

A study on ITEC[53] jobs reveals that among software engineers in the U.K., 57% of the men and 71% of the women in software occupations were graduates. Among the graduate-qualified software engineers, men were less likely to have graduated in an ITEC related specialization compared with women. It was among the graduate-qualified computer programmers/analysts, the greatest variation in specializations was evident. Among the graduate-qualified software engineers, at least two in five had specialized in mathematics and computing science, one in five in engineering and one in ten in physical sciences. The results of the U.K. study suggest a range of skill levels and specializations are needed to meet the demands of the ICT occupations.

Among the respondents in the FLOSS survey, about half reported having a university degree. The results of the U.K. study on ICT occupations and of the FLOSS survey suggests a range of skill levels and skill specialization are used to fill these occupations. The nature of ICTs means that Europe has to ensure it has an adequate supply of skilled workers with various levels and fields of expertise to fill these jobs.

7.4.4. FLOSS: retaining skills in Europe?

On a 2003 survey[54], businesses in Canada were asked to identify drivers for adoption of FLOSS — the major drivers were cost reduction (23%) and vendor independence (21%) and then skills with 11% (as stated elsewhere in this report cost and vendor independence have been cited as important reasons for FLOSS adoption by users in several surveys including the FLOSS 2002 survey of user organisations and the FLOSSPOLS survey of government organisations in Europe). Among the Canadian companies surveyed, if IBM Canada and CGI are excluded, 4 years was the average number of years in business and they had an average of 14 employees. Canada is an economy of small and medium-sized businesses and cost savings are important for their survival. The Canadian report suggests that while businesses may well adopt FLOSS to save money initially, other benefits soon become apparent – as described further in section 7.5.2, “The business case for SMEs supporting FLOSS”. In particular, there is a human resources dimension as well. FLOSS can be seen as a shift away from past trends of outsourcing skilled IT talent, back to reliance on internal ICT talent. Thus, in several ways, the adoption of FLOSS represents a shift away from the past trends of outsourcing skilled IT talent, back to greater user involvement and responsibility in ICT innovation, development and deployment. Like Canada, countries in Europe rely upon small and medium size firms for employment and job growth. Perhaps changing the way people work and the way firms work will change the way things are being done (e.g. outsourcing).

Given the that FLOSS is adopted by small firms to save money and skilled ICT workers are needed to develop FLOSS, there is an argument to be made that small and medium sized FLOSS enterprises in Europe will contribute to local employment and retain

---

[53] ITEC includes occupations of computing, software engineers, electronics and electrical engineering.

[54] *Open Source Business Opportunities for Canada’s Information and Communication Technology Sector, A Collaborative Fact Finding Study*, e-Cology Corporation, September 2004, a study carried out for the Government of Canada

ICT skilled workers in Europe through increased demand for in-house skilled labour. As shown in section 7.4.1, there is evidence to suggest that a significant among of skills training can emerge from participation within the FLOSS developer community itself, and indeed this may explain the willingness of firms to allow employees to participate in FLOSS communities during their time at work. Since the increase in FLOSS developer skills is a feature of using FLOSS – which provides a skills development growth path from user to power user to developer – the increase in FLOSS usage among SMEs and their resultant need for skills may be self-reinforcing. There is another point that needs further investigation, but suggests considerable potential for FLOSS: given that larger firms offshore to save on costs and access skilled ICT workers, is there not a scenario that small and medium firms can affect cost savings in another way, with FLOSS, and thus be able to afford and employ local ICT skilled workers?

We know that governments are adopting FLOSS and are a driver of FLOSS development. When the public sector demands products and services, employment and job creation can follow. Section 7.5.3 shows that FLOSS allows firms to locally retain a higher share of value added than proprietary software (except for the few firms that are leading proprietary software vendors themselves, and the handful of regions that host such firms). A corollary is that FLOSS could allow local firms, particularly SMEs, to save costs through FLOSS rather than outsourcing (which is more expensive for SMEs than for large firms), retain a higher share of local value added, and thus retain a higher share of skilled IT employment locally. This demand for skills can be met at least partially through the informal skills training provided through the use of FLOSS and the associated possibility to participate actively in the FLOSS developer community.

Moreover, as more and more FLOSS companies respond to regional and national needs for services and security, the potential for regional and national job growth will increase just as one might expect the off-shoring of ICT research and development to decline (in relative terms). It could thus be useful to explore employment policy scenarios based on public and private sector FLOSS adoption projections.

In terms of employment growth and retaining of skilled ICT workers in Europe, there is another employment consideration, the indirect employment affect of FLOSS. Worldwide government purchases of software totalled $17 billion in 2003 and an increase of 9% per annum was expected until 2007.[55] While government procurement may be a relatively small (though substantial) share of a large proprietary software vendor's revenues[56], an important consideration is that as governments and public institutions (e.g. health and education) adopt FLOSS, adoption by business and citizenry follows. The demand for FLOSS and ICT skills will be driven by the need for compliance (e.g. e-government services) and the example provided by government users, generally presumed to be conservative. However, it will also be driven by needs and curiosity of the general public, as they have the opportunity to see FLOSS in day-to-day practices. This in effect moves FLOSS into mainstream business and social lives, a next logical component of e-government and e-society.

---

[55] *Open Source Software, Microsoft at the power point,* The Economist, Sep 11th, 2003.

[56] According to an estimate by Dr Tech Kari Tilli, Director (telecommunications and electronics industries) of Tekes published in March 2006 by the European Commission, the public sector accounted about 20% of the ICT market in the EU – see http://europa.eu.int/information_society/research/vienna_process/vienna_documents/documents/k_tilli.pdf

One of the key challenges of retaining ICT skilled FLOSS workers in Europe is the global demand for ICT talent.

A 2003 study on factors that cause European scientists and engineers to go abroad identifies employment opportunities, career advancement, funding and networking among the top reasons, with salary identified further down the ranking. If the exit of European ICT workers is not about money but about the chance to do the work and work with leading edge technologies and networks, then FLOSS may well be the answer to slow the outflow of European ICT workers. Again, the empirical evidence is not sufficient to prove the effect on flows, but it is sufficient to suggest the strong possibility of such an effect, and the need for additional study on the effect of FLOSS and international mobility of European ICT skilled persons.

7.4.5. FLOSS and software employment

As the previous sections show, there is a strong support for the argument that FLOSS provides a novel method of developing IT-related skills and allows broad skills diffusion with low barriers for skills improvement and deepening simply through its use. Furthermore, it provides a way of reducing costs, as an alternative to outsourcing, which would lead to increased demand for skilled in-house employment.

This background allows us to explore the possibility of quantifying the employment generation effect of FLOSS, keeping in mind the potential for cannibalisation of proprietary software by increased FLOSS software use, which would have an effect on jobs.

First, it is important to recognise the relatively small share of packaged proprietary software in the total market. Packaged proprietary software accounts for 19% of the European software market and 16% of the US software market (see Table 24 on page 124).

Custom software (52% in Europe, 41% in the US) and in-house software (29% Europe, 43% US) cannot really be cannibalized by FLOSS — they do not involve software licence fees that are the only spending that FLOSS *necessarily* eliminates. Indeed, in-house software could all become FLOSS – and already follows the FLOSS principles of being free to use study, modify and distribute to all its users, because in-house software only has one user that is also its originator. Similarly, customised software is also typically available for users to freely modify and distribute (as work-for-hire) and even where this is not the case, the economics of customised software follow a support or service model rather than a per-unit licence fee model. The support and service model is completely compatible with FLOSS, as described in section 7.5 (with a discussion of FLOSS-related IT services in section 7.7.1).

Table 16 shows US software development and support jobs by sector, based on strict definitions of “software development” and “software support”[57].

[57] See footnote 58.

Table 16: US software development and support jobs by sector, 2005

		Development				Support		
Rank	Sector	jobs ('000s)	jobs (%)	average annual wage ($)	% of total jobs in sector	jobs ('000s)	jobs (%)	% of total jobs in sector
1	Computer Systems Design and Related Services	443	26.7%	77,356	37.45%	188	16.6%	15.92%
2	Professional services incl consulting	167	10.1%	78,543	2.92%	105	9.2%	1.82%
3	Financial sector incl insurance	159	9.6%	73,362	2.00%	116	10.2%	1.45%
4	Federal, State or Local government (incl Postal Service)	126	7.6%	67,697	1.32%	65	5.7%	0.67%
5	Retailers and Wholesalers (incl computer and software)	109	6.6%	74,857	0.65%	92	8.1%	0.43%
6	Software Publishers	92	5.5%	82,756	38.89%	33	2.9%	14.22%
7	Data processing, hosting and Internet services (including ISPs)	82	4.9%	74,302	17.69%	53	4.7%	11.56%
8	Management of Companies and Enterprises	80	4.8%	72,723	4.61%	70	6.1%	4.01%
9	Computer and communications manufacturing including semiconductors audio video equip and media	78	4.7%	87,843	8.86%	22	1.9%	2.52%
10	Office administration support services (incl Employment)	59	3.6%	74,115	0.87%	65	5.7%	0.64%
11	Education including Universities	48	2.9%	56,525	0.43%	113	9.9%	0.92%
12	Telecommunications services	41	2.4%	76,868	4.67%	45	3.9%	5.15%
13	Machinery including aerospace, auto, electric	38	2.3%	74,877	1.12%	17	1.5%	0.48%
14	Instruments, including medical supplies and maintenance of precision equipment	34	2.0%	80,388	4.02%	13	1.1%	1.53%
15	Health services including hospitals	28	1.7%	62,845	0.21%	41	3.6%	0.27%
16	other manufacturing	25	1.5%	66,380	0.28%	27	2.4%	0.30%
17	Publishing including broadcasters and cable distribution	19	1.1%	70,658	1.25%	28	2.4%	1.83%
18	Utilities and construction	14	0.9%	70,249	0.29%	13	1.2%	0.18%
19	transport and post/courier	7	0.4%	66,059	0.20%	8	0.7%	0.21%
20	civil society / non-profit	5	0.3%	62,182	0.38%	14	1.2%	1.06%
21	Other	3	0.2%	64,884	0.05%	9	0.8%	0.12%
	TOTAL: ALL SECTORS	1,656	100.0%			1,136	100.0%	

Copyright © MERIT. Compiled from US Bureau of Labor Statistics data for May 2005. Ranked by development jobs. Sector 1, 6, 8 are 4-digit NAICS, the rest are grouped by MERIT. Occupations are grouped[58].

Proprietary software vendors are classified as "software publishers", including firms such as Microsoft and Oracle. This sector is ranked sixth in terms of its share of software developer employment, accounting for only 5.5% of software development jobs and 2.9% of software support jobs. Firms in other sectors also develop proprietary packaged software, especially in sector #9, which includes computer manufacturers (such as HP). The top-ranking sector, with by far the highest share of employment is primarily composed of custom software developers, including computing facility management services (i.e. outsourced "in-house" software developers)[59].

[58] Occupations are grouped by 6-digit OCC code. Software developers: Computer and information scientists, research, Computer programmers, Computer software engineers (applications and system sw), Computer systems analysts. Software support: Computer support specialists, Database administrators, Network and computer systems administrators, Network systems and data communications analysts, Computer specialists (all other)

[59] This sector includes IBM, which also produces some proprietary software; however, most of IBM's software developers, like those of other firms in this category with a large number of employees *worldwide* such as Capgemini and Computer Sciences Corporation write custom software or provide software services, what we call outsourced in-house software. In terms of top *worldwide* (not US) employers in this sector it is notable that 3 of the top 11 are Indian firms: Tata Consultancy Services, Wipro and Infosys, which illustrates the nature of this sector.

What is most remarkable about this distribution is that IT developing sectors (#1, #6, #7, #9) account for 42% of software developer employment, but a majority (58%) of software developers are employed in IT user firms. This suggests that at least in the US, software *development* skills are widely distributed throughout the economy, putting firms across the economy in a good position to adopt FLOSS. In particular, it is notable that the US government employs almost 8% of all software developers.

Of course, software publishers (and computer systems design and related services, which is custom software development) have the highest share of software developers relative to total employment in their sectors, at about 38%. The next most developer-intensive sector is #7, data processing and hosting providers, with 17% of total employees being software developers.

It should also be pointed out that the average salary paid to software developers by software publishers, while higher than some other sectors, is not much higher (the difference with the professional services and consulting sector is less than 5%; with "civil society / non-profit" – the lowest-paying sector, the difference is only 24%). This suggests that software publishers, and other firms in the "IT developing sectors" do not have a dominant share of the highest-skilled employees, taking salary as a proxy for skill levels (though sectoral differences in profit margins is surely also a factor in determining salaries). Thus skills, not just employees, seem to be widely distributed outside the software publishing and IT developing sectors to user sectors.

Other than employees of software publishers, and perhaps a share of employees at computer equipment manufacturers, all other software developers could potentially be developing FLOSS – representing over 90% of software developer jobs in the US. Of course, they are not developing FLOSS, but the point is that FLOSS cannibalisation is unlikely to have a significant negative effect on the market for jobs outside the software-publishing sector. Another point that should be made is that more than 90% of software developers are engaged in forms of software development that while perhaps not FLOSS, are also not proprietary packaged software. Indeed, as noted above, this software development is more aligned to the essential characteristics of FLOSS as users tend to have the same rights as creators of the software, and are often the same (always, in the case of in-house software representing perhaps 58% of software developer employment).

If we take IDC's survey data showing 71% of developers worldwide who *are* using FLOSS[60] and apply it to the over 90% of US developers who *could* be developing FLOSS, we can estimate the number of FLOSS-related software development jobs in the US at just above 1 million, or 63% of all software development jobs in 2005. If we more conservatively apply the ratio of FLOSS to total software estimated in Figure 60 (page 178) as 16% for the US in 2005, we get 238 thousand US software development jobs that are FLOSS-related, or 14.4% of total software development jobs.

Another approach would link the number of FLOSS-related jobs more closely to individuals directly participating in FLOSS projects, about half of whom earn FLOSS-related income[61] for directly or indirectly administering, supporting or developing FLOSS. As there

[60] IDC 2006, "Open Source in Global Software: Market Impact, Disruption, and Business Models".

[61] According to the FLOSS 2002 surveys of developers 49% of all developers earn income from FLOSS activities; the FLOSS-US 2003 survey, which had a stronger US bias, showed a figure of 44%; according to the FLOSSPOLS 2005 survey, the figure is 54%. According to the FLOSSWorld survey of countries in Asia, Africa, Latin America and South Eastern Europe the figure is 51%, perhaps reflecting the fact that

were over 440 000 globally active FLOSS participants from the US in 2005[62], growing at 20% annually, this implies about 218 000 remunerated FLOSS participants in the US in 2005 (about 13% of total US software development jobs, although these FLOSS developers may actually have jobs in other occupations), rising to over 500 000 by 2010. This assumes a gradual rise in the share of FLOSS participants who are remunerated[63].

Unfortunately, similarly detailed occupational data is not available for Europe, so we cannot reproduce the above analysis and estimates for Europe. However, we can expect that employment of software developers in Europe will be even less concentrated in the proprietary packaged software industry, which in Europe is considerably smaller than in the US. On the other hand, the much higher ratio of custom to in-house software in Europe (52:29) compared to the US (41:43) suggests that software development jobs in Europe may be somewhat less widely distributed in user industries than in the US, and concentrated instead in custom software and IT consulting firms. This does not significantly change the argument regarding FLOSS and cannibalisation – there is clearly less employment in proprietary software sector in Europe to cannibalise – since custom software is compatible with FLOSS, as described previously. So, we could still estimate the share of FLOSS-related employment in Europe in a similar range as for the US, i.e. between 16% and 63%.

Following the alternative approach linking the number of FLOSS-related jobs more closely to individuals directly participating in FLOSS projects, about half of whom earn FLOSS-related income for directly or indirectly administering, supporting or developing FLOSS, we can consider the registered users of Sourceforge from Europe as a proxy for globally active European FLOSS developers. (In fact, since Europe has a number of its own development portals, this may under represent the total; but we cannot just add up registration figures from different portals, as it would be impossible to correct for possible double-counting errors.) As there were over 407 000 globally active FLOSS participants from Europe in 2005, growing at 20% annually, this implies about 204 000 remunerated FLOSS participants in Europe in 2006, rising to over 600 000 by 2010. As for the US, this assumes a gradual rise in the share of FLOSS participants who are remunerated.

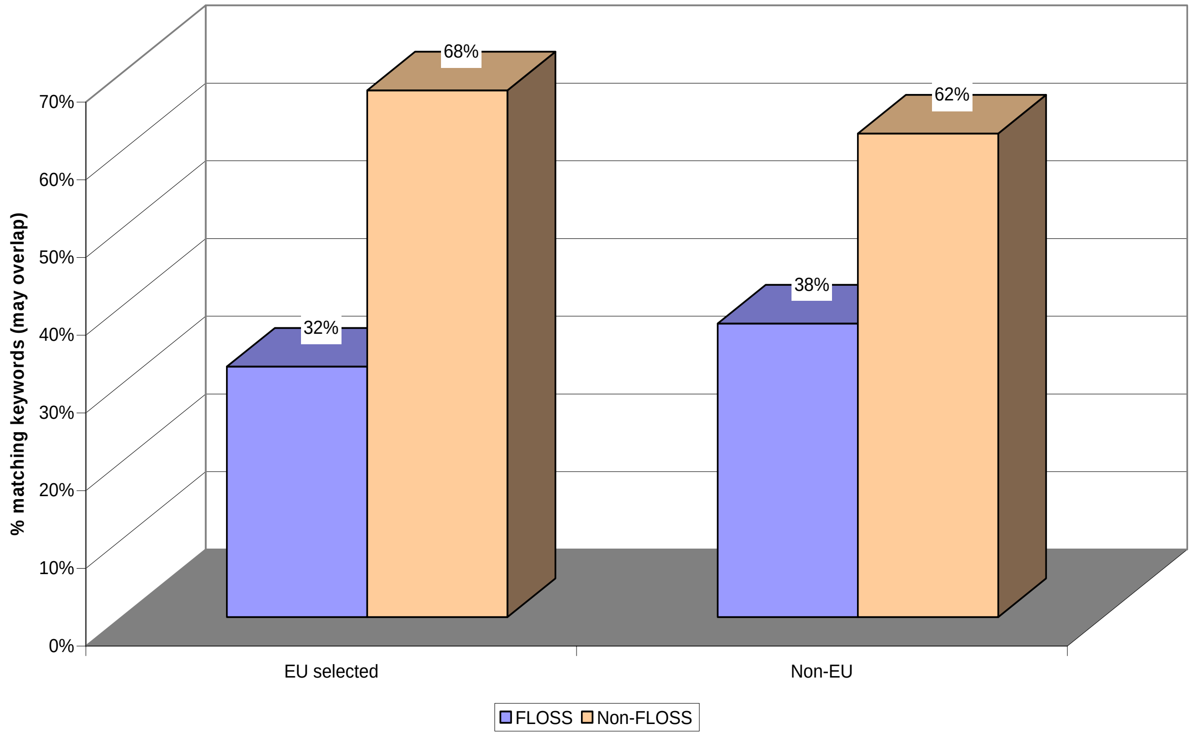
An indication (but certainly not definitive) of the interest in FLOSS-related jobs can be seen in Figure 42, which shows the shares of a sample of 379 thousand jobs postings matching selected “FLOSS-related” and “non-FLOSS-related” keywords, on over 50 job posting websites in 10 EU and 7 non-EU countries. The figure shows the shares of the total job postings matching any keyword that matched FLOSS-related keywords, and as a control, the share of postings matching a non-FLOSS keyword is also shown. As jobs may include keywords from both sets (e.g. “Linux” as well as “Oracle”) there is certainly a possibility of overlap, but the matches to the FLOSS-related keywords can be seen as a reasonable lower bound for the share of jobs postings for IT positions that require some FLOSS skills. Interestingly, this figure is higher for the non-EU sample.

these developers joined the FLOSS community about 3 years after those in the FLOSS and FLOSSPOLS sample.

[62] Registered users on Sourceforge.net, estimated by URJC based on registration data from Sourceforge/University of Notre Dame

[63] We conservatively apply the ratio of earners from the FLOSS 2002 survey to the current population, and the project a ratio based on the growth from 2002 to the the 2005 survey upon the projected 2010 population.

Figure 42: Job postings, FLOSS and non-FLOSS



Copyright © MERIT, from the FLOSSWorld project. FLOSS & non-FLOSS may overlap. Searches on jobs websites for FLOSS and non-FLOSS keywords[64], 379010 postings matched. Non-EU: Argentina, Brazil, Bulgaria, Croatia, India, Malaysia, South Africa. EU is for 10 selected EU25 countries. >50 jobsites were searched in July-Sept 2006.

[64] Such as Linux, MySQL, Apache, Perl (FLOSS) and Windows, Oracle, IIS, Java (non-FLOSS – although several Java jobs involve FLOSS).

7.5. New businesses, business models and benefits for SMEs

The FP5/IST FLOSS project coordinated by the University of Maastricht/MERIT conducted a study of business models in Free/Libre/Open Source Software, identifying a hierarchy of business models used which in general remains valid today (see Figure 43).

There are two fundamentally different groups of firms involved in FLOSS-related businesses. One group, with a significant knowledge of and involvement in FLOSS, tends to have considerable product and technology knowledge. This is used to build up what is essentially a services business. These include firms that provide training, support, consultancy and integration, as well as extending hardware sales that are enhanced by their expertise in terms of technical knowledge of FLOSS and participation in the FLOSS development community. Such firms range from the very big (IBM) to the very small (Linuxcare is a well known name, but there are innumerable small local FLOSS service firms). A subset of this group of firms works exclusively with niche FLOSS-only product development. For instance, MySQL, developer of one of the best-known brands for databases today is an originally Swedish company whose business model is based on FLOSS sales rather than services or integration.

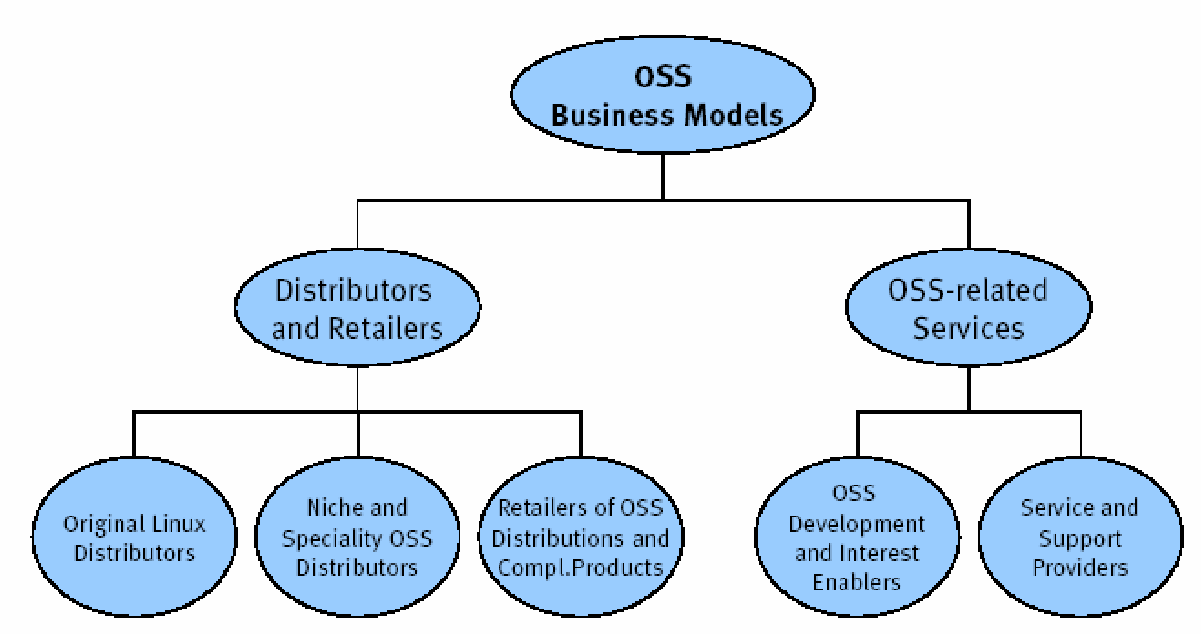
MySQL does this through a system of *dual licensing*. It makes its software available using the GPL licence, at no charge (though it offers paid support and services). The GPL licence prevents other firms from incorporating MySQL into their own *proprietary* software, as any work derived from the MySQL software must also be licensed under the GPL, i.e. must be available as FLOSS[65]. Due to the nature of MySQL's software – database infrastructure – there is a strong demand from firms to include MySQL in their own proprietary software solutions, as it saves them writing their own database software. MySQL's dual licensing strategy allows such firms to use the software under a separate, non-FLOSS licence, allowing them to make changes and incorporate MySQL in their proprietary software. When distributed under such non-FLOSS licences, however, the company charges for the software.

MySQL can maintain its strategy because it owns the copyright to the entire software it distributes. Of course, people can modify the GPL version, and MySQL would not ordinarily own the copyright to the entire modified version, and thus could not distribute it under its dual-licensing system. But MySQL ensures that it owns the copyright to modifications to the software it finds useful: by pre-empting modifications by others by making them in-house; by requesting external contributors to sign their copyright to changes over to MySQL; or, often, by employing individuals who make valuable external contributions and acquiring their copyright.

Similarly, JBoss, the US-based (but French led) SME that is the market leader for application servers and was bought by Red Hat earlier in 2006 develops its software internally. Somewhat different from MySQL, though, its business model is based on freely distributing software and charging for support, similar to a subscription model.

[65] For more details of FLOSS licensing see section 7.5.1.

Figure 43: Hierarchy of FLOSS-related business models



Copyright © 2002 Berlecon Research. Source: FLOSS User Survey (coordinated by MERIT)

As an example of a more recent development in business models, which could provide a future scenario for SMEs in general even beyond the FLOSS sector is the Orixo network[66] of mainly small and micro-enterprises in Belgium, France, Germany, Italy, UK and Switzerland specialising in massive mission-critical web server applications based around customising the FLOSS web server Apache and related Java/XML technology (such as Cocoon) for large users. Orixo works by each national SME member acquiring national customers and partners in other countries supporting each other's clients. E.g. UK-based Luminas and Italy-based Pro-netics worked together to build a solution for VNUNet, one of Europe's largest online newswire services, which now serves 100 000 articles with 6 million pages downloaded each month. The UK and Swiss firms were able to work together to implement a solution for BIOMED, a global database of thousands of medical images, and the national Orixo members are closely collaborating with the national TV/media organisations (BBC in the UK and RAI in Italy) to build FLOSS-based solutions for reliable broadband and mobile media content servers. This model allows each small firm to profit from its expertise for custom solutions, while drawing on a large base of pre-written software under FLOSS licences, and draw in addition on a large community of hundreds of individual developers spread around the world, including volunteers but also other similar small companies. Needless to say, skills levels in these niche firms are very high, with proportionately high salary levels and profits and potential for network-based employment generation on a large scale.

The revenue model of Orixo members is based on integration and customisation. They charge for building custom solutions based on FLOSS software for their customers' complex needs. The customisations they implement often involve improvements or feature additions to the FLOSS software that can be useful for others, including other Orixo members, so Orixo members can contribute back new features to the FLOSS codebase. Due

[66] www.orixo.com

to the essential needs of their (future) customers for special solutions, this does not necessarily reduce the revenue earning potential for Orixo members.

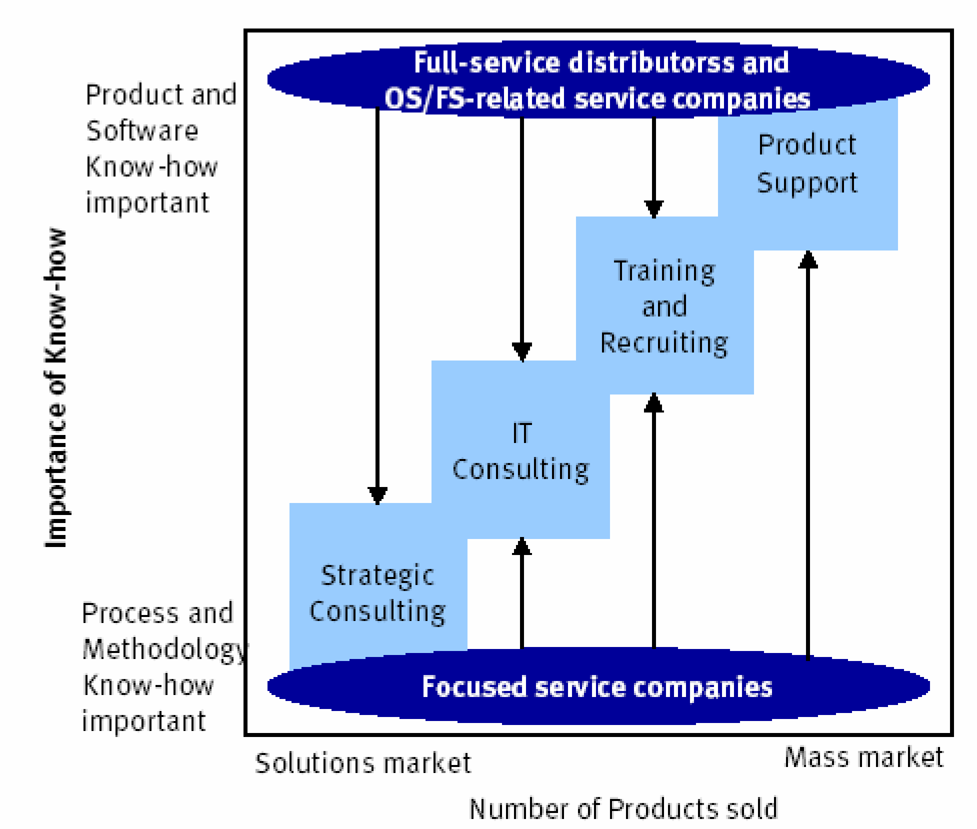
It should be noted that when firms such as those in the Orixo network provide custom solutions using FLOSS software, they rely on software contributed by thousands of individual developers. However, regardless of the specific licence used, such as the GPL, the customising firm provides an assurance to the customer about the quality and reliability for mission-critical applications. As the customising firm has full access to the source code, it can provide such an assurance as it can make any changes it believes may be required to meet the needs – quality or otherwise – of the customer. Of course, the customer may receive the resulting software under a GPL licence giving it the freedom to make changes on its own, but if it does so, it will probably violate its service agreement with the supplier. (Just as when you buy a computer, you are free to modify it – but you will probably void your service agreement or guarantee when you do so; although just as with FLOSS you can continue to modify it further if you have the skills, or pay someone else to do so.)

Similarly, ZEA Partners[67], formerly the Zope Europe Association, brings together SMEs from several countries that support or develop the FLOSS content management systems Zope, Plone and Silva. Founded by the founders of those FLOSS projects, ZEA includes 16 SMEs from nine European countries as well as four non-European partners: from the US, South Africa, India and Argentina.

ZEA provides promotion, training and coordination services as well as some degree of technical support to members, each of who specialise in their own application domains, and usually (but not necessarily) their own regions. Acting as a network makes it easier to approach large clients, and their collective client list is impressive, from eBay.com, Philips Research and London-based IMS, Europe's leading investment multi-manager, to Oxfam America, ETH Zurich and the Rotterdam Police Department.

[67] http://www.zope-europe.org/

Figure 44: Relationship between know-how, business models and market size



 Source: FLOSS User Survey (coordinated by MERIT)

At the other end of the knowledge spectrum, the there is a class of firms with considerable expertise in services and integration (e.g. KPMG consulting, Cap Gemini Ernst & Young – or smaller focussed firms such as Microconsult in e-learning or Monster.de in recruitment) that are broadening their service provision by adding FLOSS to their portfolio. This often allows firms to provide more complex services and integration possibilities while retaining a larger share of profits within their firm, as they essentially retain 100% of the profits and do not have to pay royalties to other (proprietary) firms for use of software, while building upon a large base of available software allows them to charge clients less for more advanced solutions.

Figure 44 shows the relationship between prior know-how of FLOSS and the sort of business model chosen, with the resulting market size for each individual product or service.

7.5.1. Collaboration and IPR sharing

Zea emphasises the role of associations of FLOSS SMEs in developing FLOSS businesses and leveraging complementary skill sets or market presences, as described also for the Orixo network in the previous section. When their work is mainly based on FLOSS, SMEs can cooperate with other SMEs that may be, in certain circumstances, potential competitors, without fear of IPR theft, without having to craft complex IP licensing arrangements that can be expensive for SMEs to maintain. This ability to cooperate with potential competitors may work especially well when firms use reciprocal licences such as the GPL. Such licences allow competitors to benefit from software that a firm releases, while ensuring the originating firm that competitors cannot exclusively appropriate the benefits. If the competitor improves or customises the software, reciprocal licences ensure that in most circumstances the original author can access the improvements. This is why most firms, like most individuals, prefer to use reciprocal licences to release their FLOSS software.

There are essentially two licensing models for FLOSS: permissive and reciprocal[68]. The *permissive* licensing model is fairly close to public domain. It allows licensees broad rights to use, study, modify, and distribute the software with few if any conditions. Most conditions relate to disclaimer of warranty issues. Examples of such licences include the Berkeley BSD licence, under which the popular operating system FreeBSD and its relations are distributed; the Apache Licence used for the market leader in web server software, Apache; and the MIT licence used for the X Window system of graphical user interfaces under Unix-like operating systems. As the names of some these licences indicate, they originated in universities and are often referred to as *academic* licences.

The other licensing model, accounting for a majority of FLOSS projects is *reciprocal*. Quite different from the public domain, this model forms a “protected commons”. Licensees have broad rights to use and study the software. If they distribute the software, they must provide recipients access to the source code (providing them the freedom to study). They must also provide recipients with the software under the same terms, allowing recipients to further use, modify or distribute it. Licensees can only modify the software if the modified software is also distributed under the same terms, ensuring that recipients of such a derived work can further modify it. This ensures reciprocity by forming a “protected commons” – authors are contributing their software into a commons with certain freedoms attached, and further modifications must be made available with the same freedoms protected.

The best known reciprocal licence is also the most widely used FLOSS licence, accounting for over 66% of FLOSS software projects (Freshmeat 2005[69]), the GNU General Public License (GPL), with a further 6% distributed under the closely related Lesser GPL. The GPL is the licence used by the Linux kernel and several other large software packages. Other widely used reciprocal licences include the Mozilla Public Licence[70], used for the popular web browser Firefox; the Lesser GPL[71], used by OpenOffice, the main competitor to the Microsoft Office productivity suite; and the Common Development and Distribution License used by Sun Microsystems for OpenSolaris, the open source version of its respected server operating system.

Among individual developers, 60% think[72] the role of a licence is “To prevent others from appropriating the software we've created” (FLOSS-US survey[73]), thus showing that they are not altogether (if at all) altruist and may frequently be choosing reciprocal licences with the selfish motive of ensuring their access to future improvements. According to a survey of 146 Italian firms, firms that release open source software prefer to use the GPL because “it

[68] The text in this section is drawn from Ghosh, R.A., 2007. “IPR, Law and FLOSS: building a protected commons”, *Journal of Intellectual Property Rights.* Forthcoming.

[69] http://freshmeat.net/stats/#license - 66% when accessed on July 17, 2006

[70] http://www.mozilla.org/MPL/MPL-1.1.html

[71] http://www.gnu.org/licenses/lgpl.html

[72] out of 1540 respondents: http://www.stanford.edu/group/floss-us/stats/q7.html

[73] David, Paul, Waterman, Andrew and Arora, Seema, 2003. “FLOSS-US: The Free/Libre/Open Source Software Survey for 2003”. *SIEPR/KNIIP Working Paper,* available at http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf

allows to keep the code open and forbids competitors to turn it into proprietary.” (Bonaccorsi & Rossi 2003[74]).

This has even been a concern for the public sector. For example, in a study conducted to examine the possibility of the European Commission releasing a software application it owns under an open source licence, a key condition was that “the Commission requires protection against appropriation of application by third parties” (Dusollier, Laurent and Schmitz 2004[75]). The recommendation, based on this requirement, was to use a licence with a reciprocity clause, i.e. a copyleft licence, such as the GPL.

It should be noted that some large vendors are particularly critical of reciprocal licensing as being “anti-business” and preventing commercialisation, while approving of “permissive” licences such as the BSD licence, which allow FLOSS software to be appropriated exclusively into proprietary software. In fact, reciprocal licenses only prevent competitors from indulging in IP theft, by taking software written by others and “commercialising” it as proprietary software, preventing the original authors to benefit from modifications to their work. This is why the argument against reciprocal licences rarely comes from firms that actually release FLOSS software, but rather from firms that want to incorporate software written by others into their own works, without providing anything in return. As this report shows in several sections, it is entirely possible to commercialise FLOSS software without exclusively appropriating it – indeed, most (about 70%) FLOSS is released under reciprocal licences, including Linux and MySQL.

While few firms that release FLOSS software advocate permissive licences, there are a number of individual voluntary developers that do so, the best known being Apache. This is perhaps the most important software using a permissive licence written after the creation of the GPL, the first reciprocal licence – the Apache licence was written in 1995[76]. This was written for the Apache web server, an open source application written by Internet professionals and website administrators. The GPL was already the dominant open source licence and the discussion among the Apache developers, about whether or not to require reciprocity, is something many subsequent projects have faced, with varying degrees of argument. Apache chose to maximize its user base, and to encourage contributions to the commons through gentle social pressure[77] rather than binding restrictions. Indeed, Apache’s user base was maximized – as shown in Figure 14 on page 33, it became the most used web server within a year of its release, and has held a steady two-thirds of the total web server market since 2000. Apache relies on goodwill – and a strong brand, thanks to its enormous popularity – to keep contributions to the software coming. But reliance on goodwill does not always work. Take the case of MMBase, a popular (especially in Western Europe) multimedia database and content management system. This was in fact developed mainly by VPRO, a Dutch publicly funded broadcaster, and further developed by a coalition of Dutch

---

[74] Bonaccorsi, A. and C. Rossi (2003). “Licensing Schemes in the Production and Distribution of Open Source Software: An Empirical Investigation”. MIT Open Source working paper series. Available online at http://opensource.mit.edu/papers/bnaccorsirossilicense.pdf

[75] Dusollier, S., Laurent, P., and Schmitz, P-E. 2004. *Open Source Licensing of software developed by The European Commission (applied to the CIRCA solution).* European Commission DG ENTR. Available online at http://europa.eu.int/idabc/servlets/Doc?id=19296

[76] The current version is 2.0, written in 2004 and available at http://www.apache.org/licenses/LICENSE-2.0

[77] See e.g. Apache Software Foundation, 2006. “Frequent Questions about Apache Licensing”. Available at http://www.apache.org/foundation/licence-FAQ.html

public authorities including the city of Amsterdam. When it was released as FLOSS, the organisation decided to use the Mozilla Public Licence, which is somewhere between a permissive and a reciprocal licence: it requires reciprocity for changes to the core code base of the copyright holder, but allows other derived works to be appropriated without reciprocal requirements. Like the Apache licence, the MPL was developed (for the Mozilla web browser, now better known as Firefox) to maximise use. For MMBase, the original developers "thought that take-up would be much faster if there was no [reciprocity] restriction at all on the use. So that every commercial vendor could just pick it up, wrap it up and sell it in a beautiful box", says Jo Lahaye[78], director of the MMBase foundation set up by the public authorities along with the business and developer community to steer MMBase after its FLOSS release. "But I think they didn't realise what the consequences of this would be or could be in the long run."

While a permissive or weak reciprocal licence may lead to rapid widespread use, as with Apache or Mozilla/Firefox, its benefits maybe outweighed by the threat of free-riding appropriators of the software, when the software is not a more-or-less complete package but a framework or collection, as with MMBase – which is frequently customised. The MMBase foundation now regrets the decision to use the MPL, rather than a "strong" reciprocal licence such as the popular GPL. "Because a lot of the companies that want to join MMBase, they have a problem with this MPL license, because they say, 'Well, of course I'm willing to give you something back that we made for this client, but then everyone has it and can use it, *but without them giving back their improvements on the thing I made*.' So this is really a great problem." The italics (our emphasis) succinctly summarise our argument that the main "free-riding" threat perceived by firms contributing to FLOSS is *not* that others are free to use their software, as end-users, without "giving back" – but that others could improve and further develop their software, without giving back those further contributions. Reciprocal licences such as the GPL prevent this from occurring.

More than relying on goodwill alone, reciprocity enforced through licensing changes the incentives involved in deciding whether to distribute software (or other information) under proprietary protections, or as open source. Instead of a binary choice between proprietary and public, implied by commentators such as Lerner and Tirole (2002)[79], contributors in fact face a more sophisticated choice, as illustrated in Table 17 below. Commentators have often assumed a choice limited to the contributor's own appropriation of a work, with the corollary assumption that a competitor (*B*) can benefit if the contributor (*A)* does not exclusively appropriate it. In fact, the existence and popularity of reciprocal FLOSS licensing shows that the profit to a firm's competitor of access to the firm's software may be much less than the profit to a competitor of exclusively appropriating the firm's software – with a corresponding loss for the original firm. As long as the benefits of providing access outweigh the lower losses of providing access while preventing the higher losses of exclusive appropriation by competitors, firms benefit by releasing software as FLOSS.

[78] Jo Lahaye Interviewed for the EC DG INFSO Project, "Study into the effect on the development of the Information Society of European public bodies making their own software available as open source", by MERIT, Unisys and Eurocities.

[79] "Why should top-notch programmers contribute freely to the provision of a public good?"

Table 17: Exclusive appropriation and modes of distribution

	Exclusive appropriation by others (competitors)	No appropriation by competitors
Exclusive appropriation by contributor	n.a.	Proprietary
No appropriation by contributor	Permissive licence	Reciprocal licence

In fact, Zope, the main software for Zea, is distributed under a permissive licence, like Apache. Note that both applications originated in voluntary (but professional) community development, unlike, say, MySQL, whose developers had business plans in mind from the beginning, and thus chose the reciprocal GPL licence as the most business-friendly one.

But the fact that Zope is FLOSS ensures that partners of the Zea network can contribute and share their IP in building better software without worrying too much that they their collaborators who may also be potential competition may exploit them. The IP contribution of each individual organisation in the network – just as with each individual developer in FLOSS in general – is small relative to the whole, so each member of the network benefits by staying a member and sharing rather than "cheating". And unlike with a primarily proprietary software model, complex legal arrangements are not required between the small firms that form a network.

The network structure allows SMEs to benefit from a formal joint marketing budget – beyond the brand building that forms an important reason for firms to release software as FLOSS, as the free distribution from FLOSS is also a free brand-building exercise. While the large firms that dominate proprietary software markets can afford extensive marketing, SMEs cannot; this may be one reason there are well known FLOSS SMEs that have flagships that are (near) market leaders, such as JBoss and MySQL, but few proprietary software SMEs in a similar situation.

7.5.2. The business case for SMEs supporting FLOSS

In addition to the possible cost savings of *using* FLOSS that can be essential for SMEs, and the ability to access a skills development environment (and more skilled personnel) at a lower cost, as described in section 7.4 following page 62, FLOSS provides a number of attributes that disproportionately favour SMEs.

Even if FLOSS is selected for its low cost[80], other benefits become quickly apparent. One of them is the ability to adapt software for local needs. Proprietary software companies are usually global, concentrated in a few parts of the world. This is the nature of the software market, which, thanks to network effects and proprietary standards tends towards natural monopolies. These global companies make investments on the basis of global returns, and may not pay sufficient attention to local needs.

The tendency of proprietary vendors to ignore local needs is especially marked in developing regions or regions with minority languages. As proprietary vendors are motivated

[80] This section is drawn from Ghosh, R.A. 2006, "Access to Knowledge and the Development Agenda: Emerging Issues from the Free/Libre/Open Source Software Debate", in Ghosh, Rishab Aiyer and Schmidt, Jan Philipp, 2006, *Open Source and Open Standards: A New Frontier for Economic Development?*, United Nations University/MERIT Policy Brief, No.1 2006, ISSN 1814-8026. Available at http://www.merit.unu.edu/publications/pb/unu_pb_2006_01.pdf

by global profit-maximisation strategies they often don't care about local issues and user needs – unless they matter in "a global context". For example, they may pay little attention to the needs of Nynorsk speakers in Norway, who form a relatively small market in global terms[81]. And since the software is proprietary, no local user or local business is in a position to add such support.

Many FLOSS developers too have absolutely no interest in software usability for Nynorsk speakers. However, unlike their proprietary counterparts, FLOSS developers allow and encourage those with locally relevant motives to adapt their software. So local users – and, importantly for building local ICT economies – small local businesses are entirely capable of providing services and adapting the software to local needs. In the case of Nynorsk this was done for several FLOSS applications before 2002, while Microsoft refused to provide local language support. Similarly, the government of Catalunya has recently started distributing a (partly locally developed) Catalan language distribution of GNU/Linux tailored for educational use to 120 000 desktops in schools.

Local adaptation can go well beyond language interfaces, however. In the well-known case of Extremadura, a poor region of Spain[82], a local version of GNU/Linux was developed, called GNU/LinEx. Uniquely, all the usual icons for common applications were replaced by images more familiar to locals (and easier to pronounce) than "Mozilla" and "GIMP" and "Browser". Instead, there were images of local painters and writers (to launch the paint and word-processing applications) and a bird known in legend to travel far and wide to search (web browser). As a result, this free software environment has been used to train over 70 000 housewives, unemployed and retired persons the use of computers for the first time, making the interface more approachable than that of the standard Windows (or the standard Mac or GNU/Linux).

7.5.3. Locally retaining a higher share of the value added

Such local adaptation supports the creation of new, local businesses, which are able to provide commercial support for and build upon free software thanks to its low entry barriers, in a way that would not be possible with proprietary software. This effect is heightened by any public support of the open source software sector. For example, the take-up by the Extremadura Region in Spain of open source through its support for the LinEx project has led to an economic regeneration in a relatively poor region of the European Union (receiving, in April 2004, the award of the European Regional Innovation Award). This has not just allowed the implementation of activities for a lower price, but activities especially in education and training which were simply not possible with proprietary software; it has also led to the growth of a number of small businesses to provide commercial support, since with free software there is no need to approach one sole vendor for support — approaching local entrepreneurs is possible and an obvious choice.

Of course, proprietary software also supports local businesses (excluding businesses who are *users*, who exist regardless of the type of software). What are the types of businesses that can be based upon proprietary software? Building new products and services above the

---

[81] Although, famously, after a threatened boycott of Microsoft products the company was forced to develop "expensive" support for Nynorsk (New Norwegian) which was already supported by free software applications in 2002; see http://dot.kde.org/972035764/ and http://news.bbc.co.uk/2/hi/technology/2615363.stm

[82] http://www.linex.org/linex2/linex/ingles/index_ing.html

platform is one, equally applicable to free software – 100% of this value is local. Sales commissions are another, rarely possible with free software, and of relatively low value. While 100% of the commissions are locally retained, they represent a small proportion of the total value added and every dollar of sales commission represents several dollars of imports. Finally, support, integration and customisation – this is where with proprietary software the local value added is limited by the proprietor's control of the software. "Deep" high-value support requires "deep" high-value access to the software, which only the proprietor has.

With free software, the "deep" support that can be provided by "deep" access to the code available to all local businesses can generate enormous value, all of which is retained locally. No royalties or licences have to be paid.

For local businesses producing their own software, rather than supporting other software too free software is often a better value proposition. Free software allows providers to reuse code rather than build from scratch, and to reuse a huge base of code written by others. Re-using (and modifying) allows the creation of much better end-user solutions for the same effort than writing from scratch, which local businesses are forced to do if they choose to develop proprietary software. Put together, this provides better value for money for customers (who benefit from software representing a large base of cumulative development) and better profit margins for local service providers (who can focus on adding new features faster rather than replicating basic ones, allowing them to charge more for less work).

It must be emphasised here that increased FLOSS use can allow regional economies, and SMEs in particular, to locally retain a higher *share* of the added value. It is clear that sales commissions related to proprietary software may lead to a higher *absolute* value retained locally, if proprietary software is much more widely used than FLOSS. A high added value in a small market can be less than valuable locally than low added value of a large market. Indeed, this makes FLOSS potentially more attractive, as it currently provides lower absolute added value locally than proprietary software, but provides a higher *share* of added value retained locally. This is because the market is currently dominated by proprietary software. Our analysis above suggests that if the share of FLOSS was increased relative to proprietary software – whether by market-driven demand, or by regional policies as described in the case of Extremadura, below – since the share of all value added that was retained locally would rise, the total value retained locally would also rise. In any case, when a high share of proprietary software leads to a high absolute value added retained locally in the form of, say, sales commissions, this only indicates the even higher absolute value that is *not* retained by local firms.

7.5.4. FLOSS can drive ICT use and innovation in SMEs

While there is insufficient data on the impact of FLOSS on SMEs across a large region such as Europe, except to suggest that among existing users of ICT, SMEs (and small government authorities) sometimes seem to lag behind larger organisations in FLOSS use (see section 6.2.1). This can be partly a statistical effect, caused by the highly skewed distribution of organisations – SMEs represent 80% or more of organisations in several sectors. It also appears to be due to a reluctance to migrate to new software, and lack of awareness (as demonstrated by the surprisingly high – 30% – share of "unaware users" of FLOSS among government authorities in Europe, mostly among small organisations, as shown in the FLOSSPOLS survey). In fact, SMEs have been found in a number of previous studies to show reluctance to adopt new ICT systems in general[83].

It should be noted that while SMEs in some areas may lag in FLOSS usage, this is by no means a uniform phenomenon. For example, even in terms of Linux Desktop adoption, Figure 9 on page 26 shows that small and large firms lead the way and it is middle-sized and very large firms that have a lower penetration.

Also, a number of areas show SMEs leading in terms of FLOSS adoption and service provision. The Zea network of mainly European SMEs dealing with content management systems and application service software Zope and Plone note, "the nature of the FLOSS services delivered in [their] field (task intensive and complex) is the base of a development of a thriving ecosystem of micro-SMEs." Most firms in this network have 3 to 8 employees, sometimes up to 30 persons. They also find that FLOSS adoption is driven by SMEs. Indeed, this reflects the nature of the IT market which, as IDC notes[84], is "mostly small businesses selling to small businesses", with SMEs and slightly larger firms (up to 500 employees) accounting for 55% of IT spending and 70% of IT employment.

In terms of specific benefits to SMEs, the data from FLOSSPOLS surveys show that smaller organisations, especially with very small ICT administration departments, tend to use FLOSS applications more often than larger ones particularly when they are unaware of open source. I.e. they use applications such as Linux or Apache or MySQL because they think they are "free of charge", while not realising (yet) that these are FLOSS applications. What we call "unaware users" appear to be driven by cost pressures, which indicates that for SMEs in particular the low-cost advantages of FLOSS may be significant, although awareness and ICT skills may be relatively low. However, increasing use leads to familiarity with the products and an awareness of their FLOSS properties beyond the zero licence fee – including the ability to customise and the existence of a large community of support. This may lead to skills improvements, as discussed above.

For SMEs who do not already have extensive ICT use, however – and this applies to significant sectors and regions of the economy – evidence from MERIT's initial study of the impact on local firms of the ICT/FLOSS policies of the regional government of Extremadura[85], Spain is instructive. There is a clear indication that while FLOSS use may not in itself drive economic growth, the availability of FLOSS drives ICT (not always FLOSS) take-up among SMEs.

A significant connection between ICT performance in firms and the role of FLOSS was found. There was strong evidence that effective ICT performance together with the role of FLOSS is what counts in terms of improving firm performance: above average performing firms with respect to ICT performance and FLOSS support exhibit above average scores with regard to market share, cash flow and return on investment.

[83] See e.g. Arundel, A; Sonntag V. Patterns of Advanced Manufacturing Technology (AMT) Use in Canadian Manufacturing:1998 AMT Survey Results. Report for SIEID, Statistics Canada, SIEID Research Paper 88F0017MIE, no. 12, November 2001.

[84] IDC, 2006. "The economic impact of IT, software and the Microsoft ecosystem on the European Union, Croatia, Norway and Switzerland", June.

[85] Dunnewijk, Theo and Garcia, Abraham, 2005. *The economic impact of ICT policies in Extremadura.* FUNDECYT/Junta de Extremadura, Badajoz, Spain.

This performance seems driven by the importance given to innovation, and a close relationship was found between ICT use *together with* FLOSS use and educated employees, and the degree of innovation. Thus, besides ICT importance in general (which is the most important indicator when compared to other firms with a lower ICT use), FLOSS support seems to be part of the explanation for the actual ICT performance together with the level of education of the employees. The conclusion was that ICT performance matters and FLOSS support and the level of educational attainment are equally important for its performance.

In particular, a number of local small businesses have arisen to support and develop FLOSS applications, sell hardware based on FLOSS (in particular, Extremadura's version of Linux called gnu/LinEx). Some of these also develop new software, such as FacturLinex, a FLOSS invoicing and billing system developed by a local micro-enterprise and used in many shops in Extremadura and increasingly elsewhere in Spain. In interviews with MERIT, small business customers in Extremadura have expressed a preference for using software which a small firm has developed (or helped to develop) as they feel they will get better support and personalised attention, whereas a large firm with a proprietary product may not be willing or able to attend to their specific needs.

It should be noted that the Extremadura model has already been duplicated in other regions, especially in Spain, such as the much larger Andalucia, where about 400 000 desktops are running a localised version of the FLOSS operating system GNU/Linux, which is also the standard platform – as with Extremadura – for libraries and digital inclusion centres. As pioneered by Extremadura, which used regional policy in support of FLOSS to encourage local SMEs to provide IT services, Andalucia is also developing a regional policy to induce economic development through SME firms retaining a higher share of value added locally.

Finally, smaller public administrations, like SMEs (and often with SMEs) working together to develop innovative FLOSS-based solutions. This sort of "pooling" among (small and large) public sector organisations can lead to significant service improvements and cost savings, as noted by the "Pooling Open Source Software" study published by the European Commission[86]. The *communesplone* project[87] is an example: it was an initiative within local governments (in Belgium). After a year existence, over 12 local governments are involved, publicly sharing FLOSS applications built upon the Plone content management platform (supported by members of the Zea SME network). The project has already attracted interest in several EU countries and abroad[88], and highlights the potential of innovation FLOSS can provide to public administration. Its impact is unusual in terms of opportunity for innovation, standardization and economy of scale while gaining independence from large IT providers.

7.5.5. FLOSS can drive growth in SMEs

There is little data available on the number of new businesses created to support FLOSS. FLOSS is not a specific category of business, most new FLOSS businesses are SMEs about whom data is not generally available, and most FLOSS businesses are likely to have previously supported other ICT systems (or may continue to do so).

---

[86] Schmitz, Patrice-Emmanuel & Castiaux, Sebastien. 2002. Pooling open Source software. *European Commission, DG ENTR (IDA).* http://europa.eu.int/idabc/en/document/2623#feasibility

[87] www.communesplone.org

[88] As reported on the European Commission's website: http://ec.europa.eu/idabc/en/document/5617/469

The collaborative network structure used by several FLOSS SMEs means that formalised groups link to a network of several times as many, less tightly connected, SMEs. For instance, the Plone/Zope Floss ecosystem is much larger than the Zea association itself. The SMEs and individuals involved in this ecosystem are thriving and growing. Based on available data – lists of Zope service providers – a conservative estimate would set a lower bound of about 500 SMEs worldwide exclusively providing services and consulting based on Plone and Zope. The real figure could be higher, as some SMEs do not advertise the FLOSS brand names, but focus on providing customised services. Customised software represent the largest share of the software in Europe (52%, compared to 27% in-house and 19% proprietary packaged software) and a large share in the US (41%, compared to 43% in-house software). As most IT service providers (and 70% of IT industry employment in Europe, according to IDC[89]) are in SMEs, there is a significant growth potential for SMEs as custom software service providers in line with the increasing growth of FLOSS use.

Thus, there are a number of ways to define a “FLOSS-related” business. One definition would count as a FLOSS-related business any business that provides any support for FLOSS, in terms of products or services. Since various surveys show that about 20% of organisations in Europe have a significant use of FLOSS, and estimates (Figure 60: Estimated "true" software investment, share of GDP) show the FLOSS share of total software investment rising to about 31% by 2010, we can estimate that the share of software businesses that are FLOSS-related is similar, rising from 20% (i.e. 97 000 firms, following the total software industry figure from Eurostat of 485 000 firms in 2005) to 225 000 firms by 2010. Given that 54% of firms employing software developers have FLOSS in production and that worldwide, 71% of developers use FLOSS[90], and 68% of firms from IT-intensive sectors in Europe incorporate FLOSS in their own software-based products[91], our estimate for the number of businesses that produce or build upon FLOSS is conservative.

Another approach would link the number (and definition) of “FLOSS-related” businesses more closely to individuals participating in FLOSS projects, a majority of who earn FLOSS-related income[92] for directly or indirectly administering, supporting or developing FLOSS. As there are over 407 000 globally active FLOSS participants from Europe, growing at 20% annually, this implies about 204 000 remunerated FLOSS participants in Europe in 2006, rising to over 600 000 by 2010. This assumes a gradual rise in the share of FLOSS participants who are remunerated[93].

---

[89] Firms with fewer than 500 employees account for 70% of IT industry jobs, so these include small non-SME firms with between 250 and 500 employees. IDC, 2006. “The economic impact of IT, software and the Microsoft ecosystem on the European Union, Croatia, Norway and Switzerland”, June.

[90] IDC 2006, “Open Source in Global Software: Market Impact, Disruption, and Business Models”.

[91] See Figure 50: Use of FLOSS for software products and operations, by sector on page 107

[92] According to the FLOSS 2002 surveys of developers 49% of all developers earn income from FLOSS activities; according to the FLOSSPOLS 2005 survey, the figure is 54%. According to the FLOSSWorld survey of countries in Asia, Africa, Latin America and South Eastern Europe the figure is 51%, perhaps reflecting the fact that these developers joined the FLOSS community about 3 years after those in the FLOSS and FLOSSPOLS sample.

[93] We conservatively apply the ratio of earners from the FLOSS 2002 survey to the current population, and the project a ratio based on the growth from 2002 to the the 2005 survey upon the projected 2010 population.

Given that about 36% of FLOSS-related income earning developers are self-employed[94], we can estimate about 73 000 employees in FLOSS-related SME (micro) businesses today rising to over 216 000 employees by 2010.

It is noteworthy that FLOSS allows SMEs to play a larger-than-life role. Although the primary value of FLOSS for SMEs may be allowing a greater share of value added to be retained locally by SMEs, this also allows SMEs to build products and services that reach a global audience. The FLOSS business model greatly reduces marketing costs for developing firms (MySQL achieved worldwide market recognition thanks to freely available and modifiable software, not through a marketing budget). This allows innovative SMEs to grow rapidly to service a global market, with a much lower requirement for capital due to lower marketing costs. This does not only apply just to the software itself. “JBoss follows the support model,” says a venture capitalist and early investor in the firm[95]. But JBoss does not really need to advertise itself as a support provider. While several (including several very large) firms provide JBoss support, it is only natural for people considering JBoss support to ask JBoss itself. The software application brand has been developed at little cost thanks to the FLOSS distribution and development model, and other revenue streams flow – from across the world – through that.

This property of FLOSS explains how FLOSS SMEs originating from European FLOSS developers have achieved global fame and market leadership, such as MySQL, JBoss and Trolltech – even though two of them to various degrees did need the US as a base for their companies[96].

[94] I.e. 17% of all developers including those with only income unrelated to FLOSS, according to the FLOSS 2005 survey. The FLOSS-US 2004 survey showed similar results, of 16% self-employed. However, the FLOSSWorld survey covering Latin America, Asia, Africa and South-Eastern Europe showed a slightly higher share of 23% self-employed, including income unrelated to FLOSS.

[95] David Skok, General Partner at Matrix Partners. Interviewed by Jeremy Geelan in the *Java Developer’s Journal*, vol 9, issue 4 (April 2004).

[96] Jboss Inc was founded and funded in the US, MySQL was founded in Sweden where it retains many operations but relocated its business headquarters to Silicon Valley. Trolltech, though, remains headquartered in Oslo, Norway.

7.6. User benefits: interoperability, productivity and cost savings

Several studies now exist of Total Cost of Ownership (TCO) for different software application domains, comparing specific proprietary software products with alternative open source solutions. In addition, a number of specific plans have been developed (typically in the public sector) identifying methods, risks and benefits of migrating from proprietary to open source solutions (in particular, the German Federal migration plan for open source, migration plans developed by the French ATICA/ADAE and the Union of Italian Provinces).

It should be noted that most debate on whether or not open source software provides any cost benefits over proprietary software is based on two flawed notions: that costs are short term rather than long-term over multiple product cycles; and that labour costs are necessarily high in relation to product license costs. A cost estimation model based solely on labour costs for support and maintenance together with one-time product license costs (an approach common to most TCO studies) ignores the high cost impact of the vendor lock-in resulting from the use of proprietary software which typically does not support (or discourages) the use of open standards, forcing consumers into a perpetual replacement cycle of new product versions and additional license fees. Besides this, there are various additional costs implicit in the usage of vendor-specific software and data formats, such as specific hardware requirements that become beyond the control of users. From the FLOSSPOLS survey of 955 public sector administrations we see that users are quite aware of long term issues and the relationship between interoperability (open standards), FLOSS and vendor lock-in, and that a sense of vendor-dependence drives a desire to increase FLOSS adoption (see Figure 45). This is neither limited to the public sector nor new – the 2002 FLOSS User Survey of firms and public sector organisations found that 30% "totally" and a further 26% "somewhat" agreed that independence from pricing and licensing policies of software companies was a reason to use FLOSS.

Figure 45: Long-term vendor dependency drives FLOSS adoption

	Useful to increase share of FLOSS in your organisation?		
Too dependent on vendors?	Yes	No	Average
Yes	**53%**	30%	44%
No	43%	**66%**	49%

Copyright © 2005 MERIT. Source: FLOSSPOLS Government survey

Related to the issue of vendor dependency, another driver of FLOSS adoption is the increasing demand for interoperable software and open standards. Table 18 indicates that, in the European public sector, aware users[97] of FLOSS show the strongest demand for interoperable software (defined here as "compatibility with other software from other

[97] In the 2005 FLOSSPOLS survey of 955 public authorities across Europe, we found that while about half of all public authorities had some use of FLOSS (whom we refer to as aware users), a further 30% reported using specific FLOSS applications but said they did not use open source or free software – we refer to this group unaware users.

producers", to contrast with "compatibility with the product family [from the same vendor]"). But even among the non-users 50% say demand for interoperable software. Thus, though interoperability and open standards are not features exclusive to FLOSS, they are nevertheless associated with FLOSS in the minds of users. The demand for FLOSS goes in line with a growing demand for interoperability.

Table 18: Interoperability as a driving force for FLOSS

		FLOSS usage in organisation	
		FLOSS users	Non-users
What is more important when you buy software?	Compatibility with software from the product family you already use:	25.9%	38.8%
	Compatibility with software from other families and other producers:	66.5%	49.5%
	Don't know:	7.6%	11.7%
	TOTAL	100%	100%

It should be noted that compatibility costs can be an important consideration while migrating to FLOSS. Migrating to a FLOSS application may reduce compatibility with applications with market dominance, *if those applications do not support truly open standards.* Indeed, the loss in terms of compatibility may be significant, initially. It is an important reason cited against migration to FLOSS, and may even result in a reverse migration.

An illustrative example is the case of the Central Scotland Police, which had migrated to Linux and StarOffice (a customised version of the FLOSS application OpenOffice) after 2000 and in 2005 decided to migrate back to Microsoft Windows and Office. While lower costs were cited, these were clearly linked to compatibility issues: "As the need for increased integration and compatibility with other criminal justice agencies and community partners grows, the value of similar infrastructures becomes more important," said the head of ICT David Stirling[98]. He noted that when his institution received applications from other police departments, they had to be customised in order to run in the FLOSS environment used by the Central Scotland Police, and said that the agency needed its systems to work smoothly with those at other agencies and criminal justice departments. Scotland's other seven police jurisdictions use Microsoft for their desktops and applications layer, he said[99]. If the software used by other organisations supported interoperability with open standards, this would not have been an issue.

This example clearly underlines how compatibility costs, while an important cost in migration, highlight one of the most negative aspects of proprietary software – the reliance on proprietary standards in order to achieve vendor lock-in. Although nothing technically

[98] Quoted in Simons, Mark. 2005. "Scottish police give open source the boot". *Computerweekly.com,* August 11. Available at http://www.computerweekly.com/Articles/2005/08/11/211337/Scottishpolicegiveopensourcetheboot.htm

[99] See Niccolai, James. 2005. "Scottish police pick Windows in software line-up". *Infoworld,* August 11. Available at http://www.infoworld.com/article/05/08/11/HNscottishpolice_1.html

prevents proprietary software vendors from adopting open standards, there are enough business reasons not to do so, as this would by definition reduce the losses to users in terms of compatibility. By reducing the vendor lock-in effect of proprietary standards, proprietary software vendors who adopt truly open standards make migration to other systems easier, which is of course a good thing for consumers but not necessarily (in the short run at least) good for the proprietary software vendor. So, on the one hand, when compatibility is lost the cost of this should be included in the cost of a migration to FLOSS – but this cost underlines the vendor lock-in strategies of proprietary software firms, and would reduce with greater adoption of FLOSS. On the other hand, if proprietary software vendors increase their adoption of open standards and support for interoperability, compatibility would not be lost upon migration and the cost of migration would thus reduce.

The issue of compatibility losses could be taken further. Arguably, the losses of compatibility, and the losses caused by migration in general, can be divided into two categories: those that would be incurred by migrating to a specific new system, and those that are incurred by migrating away from a currently used system. Compatibility losses are in the latter category – migrating away to *any* alternative system would result in the compatibility losses (assuming the use of proprietary standards, without which compatibility losses could not occur). Such *exit costs*[100] should be evaluated as part of the cost of the *current system*, not the future system. Any product that reduces the consumer's choice of switching to another product must be evaluated by balancing the future *exit cost* against any other benefit of the product. A system that is designed to provide users the choice of migrating away from a particular vendor thus has a much lower *exit cost* than a system that is designed to ensure that the user faces an increasingly expensive choice of migrating away. This exit cost can outweigh any other benefits of a given system, and should perhaps be taken into consideration

Moreover, the assumption of TCO models that labour costs (for support and maintenance) are far higher than license fees as a share of TCO may be true in high-cost markets, but in less wealthy regions of the EU and beyond this is not the case. Not only are labour costs for FLOSS software support likely to be lower in the long term due to lower barriers for entry and skills development (as discussed in the section above on skills and future economic potential), but where labour costs (and GDP/capita) are low to begin with, the total relative cost of license fees alone is much higher, and tilts the argument favourably towards FLOSS solutions[101].

Meanwhile, from the user perspective, FLOSS has the potential of providing cost savings and changing cost structures and productivity of *ICT use.* But this needs quantifying. Migrating to a new software solution may be a risky decision that may have a strong impact on the economy of a company. A deep analysis of risks and costs to transit to a FLOSS solution is a must for any organization. Unfortunately not all the companies may afford such speculations. Our study, therefore, has built an instrument for this decision making process, and helps draw conclusions on the basis of innovative studies of, among other costs, time-use productivity by directly comparing the work efficiency of end-users faced with specific proprietary and FLOSS applications.

---

[100] The term was probably first used in this context by Simon Phipps, now at Sun Microsystems.

[101] See Ghosh, R.A. 2003. "Licence fees and GDP per capita: The case for open source in developing countries". *First Monday,* volume 8, number 12 (December). Available at: http://www.firstmonday.org/issues/issue8_12/ghosh/

This is based on data and methodology from, among others, the COSPA project supported by the European Commission's FP6 research programme, which analysed time spent by end-users for performing similar tasks with different (FLOSS and proprietary) software applications, as well as detailed case studies of TCO and migration cost for a number of organisations. While the full results are included in Appendix 2: Report on user-level productivity and relative cost of FLOSS / proprietary software, the conclusions are summarised here.

The major novelty of this study consists in a detailed framework not limited to the parameters of TCO, comparing European organizations in different countries. Many of the known studies of FLOSS migrations – for example, several published TCO studies or the long list of case studies published by the European Commission's IDABC Open Source Observatory[102] – are single-organization studies. The advantage of using a comparative study is evident: methods and instruments have been validated in different contexts resulting more stable and reliable.

Another crucial perspective of this study is the focus on the use of tools for office automation. Many of the known studies refer mainly to operating systems[103] and often to back office migration. In our study we mainly monitored largely distributed software used on clients as the office suites. Thus, we were able to really test the impact of the new technology on non-expert end-users. More that 6000 PCs were migrated to OpenOffice.org. A comparative use of two Microsoft Office and OpenOffice.org suites has been performed after a period of training to reach to same knowledge of the previous office solution.

Moreover, the study has been conducted also with the use of a non-invasive automated tool to collect time and file use in order to respect and not to overwhelm the daily work of the end-user; TCO studies typically do not consider user productivity, certainly not at the very high level of detail of measurement that was used in this study. The objectivity of an automated tool work in background reduces the effects of subjectivity that questionnaires of other manual instruments of data collection might arise.

Another important aspect of our study refers to the precise division of migration costs and ownership costs, which tend to be confused by TCO studies. Migration costs have a volatile (and temporary, one-time) nature whereas ownership costs refer to a long term of use of a software solution. In particular, license costs have not been used to compare FLOSS and proprietary software in our study – factors like inflation and market demand may affect the comparison. We collected and report them anyway for completeness (Table 20).

Our analysis has also collected costs for training as in a migration this might be significantly higher. There are two considerations to report though. Formal training performed in a migration for an office suite is an added value. This is because training for

[102] List of studies on the adoption of OSS in the IDABC web sites. http://ec.europa.eu/idabc/en/chapter/470

[103] For example: Yankee Group, *2005 North American Linux and Windows TCO Comparison* (April 2005); Heidenhem District Office eGovernment Services Case Study (September 2005) (Windows chosen over Linux based on TCO). Gartner Research Group, *Costs and Benefits Still Favor Windows Over Linux Among Midsize Businesses*, (October 2005); Yankee Group, *Microsoft - The Dominant Vendor in the SMB and Mid-Market Applications and Platform Ecosystem* (December 2005). Microsoft report "Get the Facts on Windows Server and Linux"; Joel West and Jason Dedrick, *Scope and Timing of Deployment: Moderators of Organizational Adoption of the Linux Server Platform*, International Journal of IT Standards Research 4, 2 (July 2006)

office suites currently in use within an organisation is typically self-training. So very often no formal training for currently used office solutions are really provided to end-users. Initial formal training provided at the time of a migration is thus often the first time such training may be provided at all. Moreover, training at the time of migration reduces the need of training in long terms; reducing costs of ownership.

Our analysis has been performed on six organizations in different European countries. The majority of them are public bodies. The organizations have followed different types of migration on the base of their context.

We have investigated the costs of migration, and the cost of ownership of the old and the new solution differentiating them between the costs of purchasing and the costs of ownership of the software solutions. Special attention has been put on the intangible nature of the costs. Costs have been classified in categories defined trough existing studies and selected by a top down approach called Goal Question Metric. This instrument has been also used to define the questionnaires used to collect the data.

The model defined for each organization consists of a set of values of costs for the migration, for the initial purchasing, and for the ownership. The migration costs are labelled as tangible or intangible and they are subdivided in four macro categories. The ownership costs are computed on annual base and deduced by a monitoring or a prediction computation over a period of five years.

Our findings show that, in almost all the cases, a transition toward open source reports of savings on the long term – costs of ownership of the software products.

Table 19 displays the model of migration costs for each organization. The majority of the costs of migration concerns OpenOffice.org (OOo).

Table 19: Model of migration cost by category in each organization (KEuro).[104]

PA	Software (€K) Tang. | Intang.	Support (€K) Tang. | Intang.	Training/Learning (€K) Tang. | Intang.	Staffing (€K) Tang. | Intang.	Total (€K) Tang. | Intang.
SGV	€39.5K 82% | 18%	€82K 40% | 60%	€292.5K 92% | 8%	€246K 0% | 100%	€660K 51% | 49%
Extremadura	€0 -	€680K 26% | 74%	€180K 100% | 0%	€100K 100% | 0%	€960K 48% | 52%
PP	€99K 96% | 4%	€32.5K 77% | 23%	€61K 0% | 100%	€7K 0% | 100%	€199.5K 60% | 40%
SK	€0.01K 100% | 0%	€0.83K 28% | 72%	€3.07K 27% | 73%	€0.075K 0% | 100%	€3.985K 27% | 73%
TO	€20K 0% | 100%	€53K 62% | 38%	€233.5K 57% | 43%	€33K 0% | 100%	€339.5K 49% | 51%

Costs of ownership are compared in the pre and post software configuration to determine whether there have been savings (Table 19 and Table 20).

[104]BH and ProBZ have not supplied this data

Table 20: Model of ownership cost comparison in the organizations

PA	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost of purchasing	Annual cost over 5 years	Initial Cost of purchasing	Annual cost over 5 years
SGV	€240K	€170K	€800K	€179K
Extremadura	€1.140K	€270K	€6.0K	
PP	€7.1K	€3.4K	€25.6K	€2K
SK	€0.7K	€2.4K	€23.1K	€2.4K
TO	---	---	€31K	€11.3K
BH (phase 1)	€68K	€45K	€735K	€169.6K

Comparing these tables we find that:

- Costs of migration are significant and comparably higher than the annual costs of ownership – migration costs also occur in a shorter time frame. This means that the transition requires an exceptional monetary effort.
- Initial costs of purchasing are definitely higher for closed solutions
- Costs of maintenance are comparable in the two solutions although the FLOSS configuration is sometimes more expensive. This conclusion may be biased by the fact that costs for closed solutions are real whereas costs for open solutions are based on initial predictions. Initial prediction may still be influenced by the volatile costs of the transition.

Table 21: Savings due to the migration[105]

PA	Savings of the OSS migration	
	Savings on initial costs of purchasing	**Annual savings over 5 years**
SGV	√	√
Extremadura	√	√
PP	√	---
SK	√	√
TO	√	√
BH	√	√

For the new open solution savings have been computed by predicting on the initial first year of ownership and historical data - as the five-year period of ownership for FLOSS software has not been reached yet. All organizations report significant initial savings due to

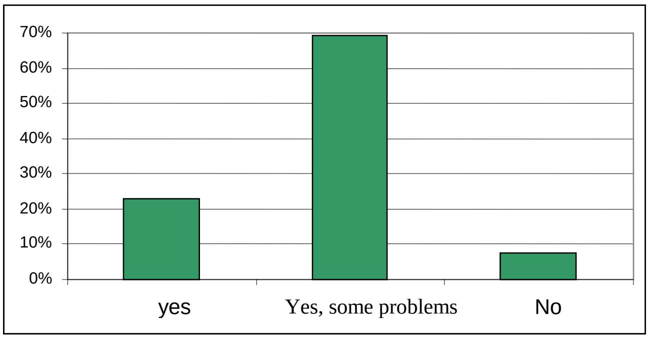
[105] ProBZ have not supplied this data. BH has migrated in two phases. Phase 1 is the most significant and we use it for our analysis. BH received a generous offer from Microsoft and in 2006 has gone back to Microsoft Office.

the zero cost of licenses. In the long term the profit is not that obvious: SGV, BH, Extremadura, TO predict to gain with the new solution. SK predicted equal costs for the new and the old solution whereas PP reports of higher costs with the new open solution in the long terms (see the section on the Province of Pisa).

We also investigated the productivity of the employees in using Microsoft office and OpenOffice.org. Office suites are widely used and are a good test bed and representative for a comparison on issues like effort and time spent in the daily routine of work. Delays in the task deliveries may have a bigger impact than costs on the organization's management. Our findings report no particular delays or lost of time in the daily work due to the use of OpenOffice.org.

The results of this time-use analysis has show that no statistically significant variation in productivity measured in number of document processed per day and average effort per document has been found over a period of 32 weeks between two randomised groups of users, one to whom OpenOffice was introduced, and one that kept using Microsoft Office. This is perhaps surprising, since users were not previously familiar with OpenOffice. Indeed, the group of users was also asked whether as a result of the experimentation they thought they could do with OpenOffice the same amount of work they could do with Microsoft Office. As shown in Figure 46, under 10% of respondents thought they could not, more than 20% thought they definitely could perform as well with OpenOffice, while almost 60% thought they could be as productive with OpenOffice (as confirmed by the time-use analysis) though with some problems. Given their previous lack of experience with OpenOffice the fact that problems were subjectively perceived is unsurprising, but the fact that objectively the productivity of users remained the same and did not reduce is important.

Figure 46: Can you be as productive OpenOffice as MS Office? User responses



Source: BICST.

7.7. Secondary production and services

Just as the impact of software goes well beyond software producers, the impact of FLOSS goes well beyond software production too. At least three areas are directly related to FLOSS software: hardware, purchased software and services.

IDC estimates the worldwide hardware and purchased software[106] market for Linux alone to increase from about $14 billion (hardware) and $5 billion (software) today to $22 billion (hardware) and $14 billion (hardware) by 2008 respectively.

7.7.1. FLOSS related services

As for services, Gartner Dataquest estimates[107] worldwide IT services revenue at $624 billion for 2005, of which Europe, the Middle East and Africa accounted for Euro 173 billion. A broad definition of FLOSS-driven services would include service revenue that relates to FLOSS capital investment, i.e. services for FLOSS systems and not necessarily only services using only FLOSS. We estimate notional value[108] of the capital investment in FLOSS at 20% of total investment in software today (in Europe) growing to 31% (see section 9.1, "Trends: use of software in Europe and the US" on page 176).

Assuming FLOSS use requires a similar amount of services as proprietary software (some claim FLOSS software requires a higher level of services[109]), we can conservatively estimate that the European market for IT services related or involving FLOSS is currently Euro 26 billion, rising to Euro 69 billion in 2010, or from 16% to 31%of the total IT services market. The worldwide FLOSS-linked services market can similarly be estimated at Euro 74 billion rising to Euro 215 billion in 2010, or from 14% to 32% of the total IT services market.

As seen in Figure 47, we estimate FLOSS-linked services to account for most of the growth in IT services in the next 5 years, with the FLOSS share of IT services growing slower outside Europe than inside until 2008, from when increased FLOSS-related demand from Asia in particular results in faster FLOSS-linked services growth than in Europe.

---

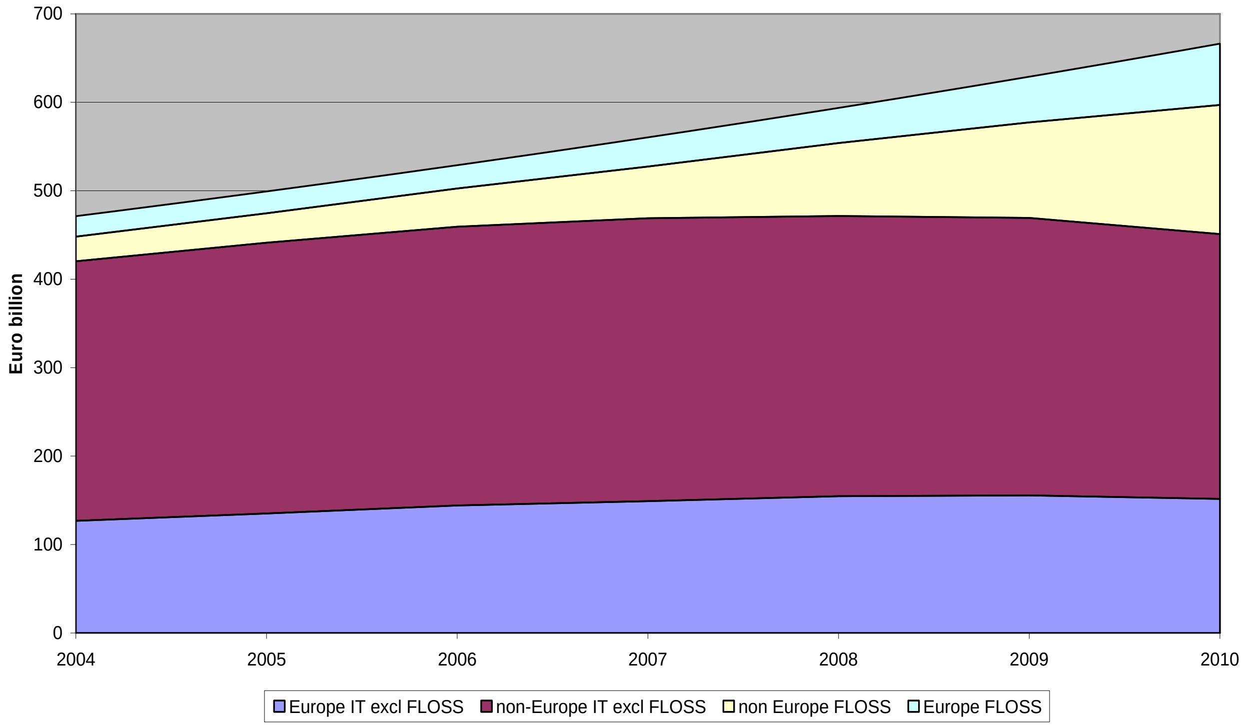
106 Purchased software such as JBoss and MySQL, which are FLOSS but for which major users who find them a suitable alternative to proprietary applications pay significant sums of money for services and support, often through a subscription model.

107 June 2006

108 Notional, since the FLOSS software is not purchased through licence fees.

109 Though for the current purpose we estimate FLOSS use generating service revenue similar to equivalent proprietary use, with a 2 year time lag in relation to the share of FLOSS-based PCs and servers in use and a growth rate higher than the FLOSS hardware growth rate.

Figure 47: FLOSS-related and IT services revenue, Europe and world



Copyright © 2006 MERIT. MERIT estimates and projections based on sources including Gartner (IT services market size); IDC (Linux server and PC sales); GGDC (software investment ratios). Shares add up to the total (Euro 667 billion in 2010).

Like ICT, the effect of FLOSS is not limited to software and services sectors alone. This is partly related to the distribution of ICT usage across sectors (Table 22), and partly to the distribution of FLOSS usage across sectors, which may not be the same (Figure 48, reproduced from the earlier section on market shares).

7.7.2. Secondary software sector

Although much of the recent FLOSS debate has focused primarily on desktop applications (Open Office, Mozilla Firefox, etc.), it is made clear in this study (especially in sections 8.1, "Competition, innovation and FLOSS" and 9.3, "Factors determining impact of FLOSS on the EU ICT market") its origins and strongest points have been in the tools and infrastructure underlying the Internet and Web services. Software like GNU/Linux, Apache, Bind, and the networking protocols for data transfer, email, the world wide web, file transfer and so on. This suggests that FLOSS may have a particularly important role to play in the secondary software sector; i.e. in domains where software is used as a component in other products, such as embedded software in the automotive sector, consumer electronics, mobile systems, telecommunications, and utilities (electricity, gas, oil, etc.)[110]. The method of software development and use varies greatly for each of these sectors.

[110] See the paper from the CALIBRE FP6 project, Ågerfalk, Pär J, Deverell, Andrea, Fitzgerald, Brian, Morgan, Lorraine. 2005. "Assessing the Role of Open Source Software in the European Secondary Software Sector: A Voice from Industry", *Proceedings of the First International Conference on Open Source Systems*

Figure 48: FLOSS usage in Europe by industry

Source: IDC's 2005 Western European Software End-User Survey (N=625))

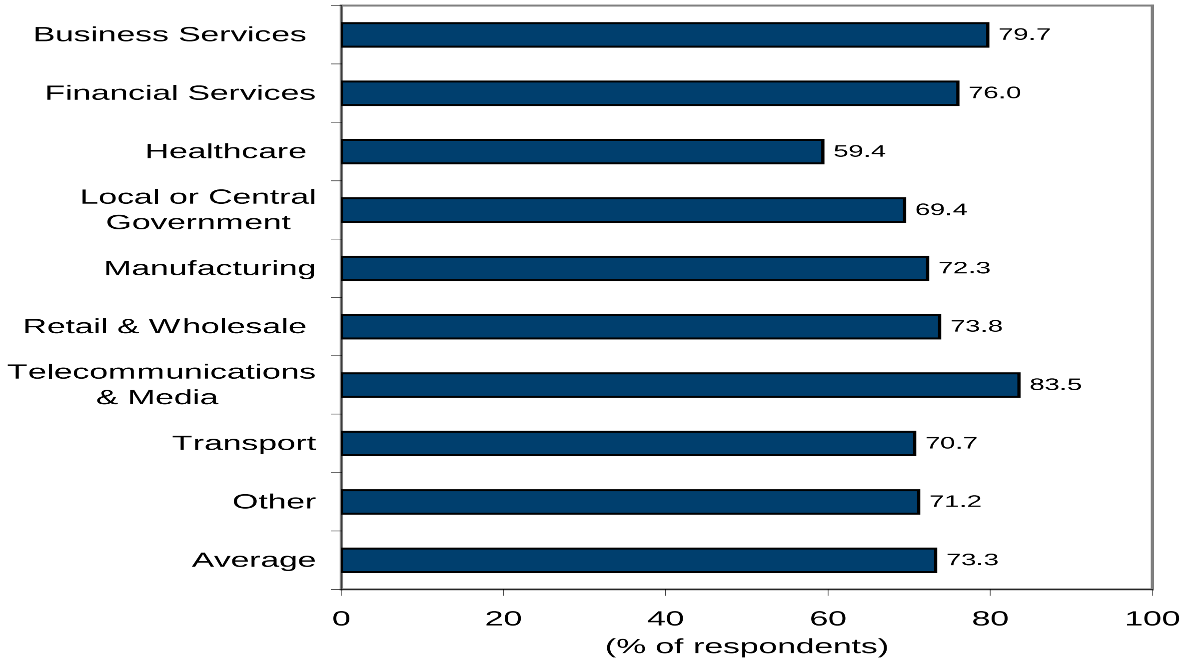


Table 22: ICT intensity by sector in terms of employment share, EU15

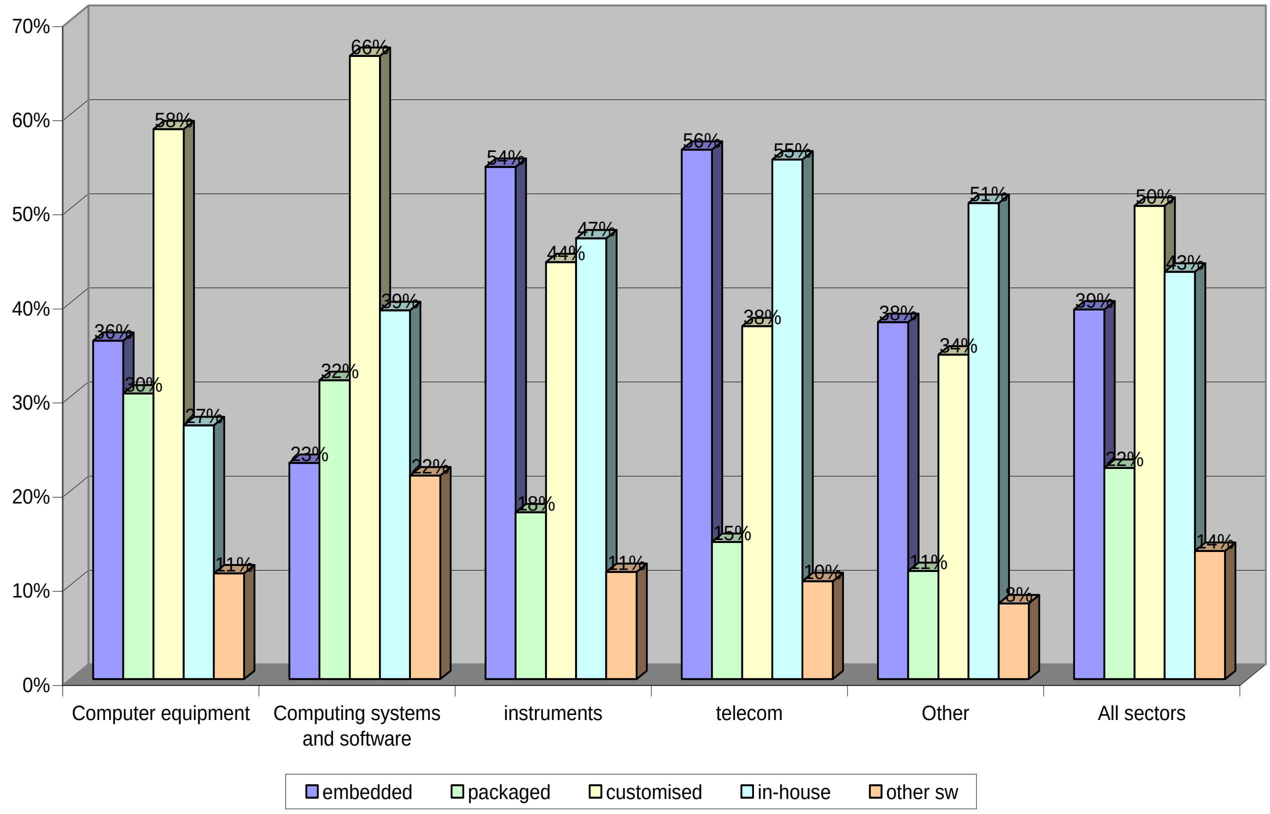
NACE	Sector	ICT employment as share of total (%)
72	Computer and related activities	84.2
66	Insurance and pension funding, except compulsory social security	74.8
65	Financial intermediation, except insurance and pension funding	69.2
67	Activities auxiliary to financial intermediation	67.9
30	Manufacture of office machinery and computers	57.1
74	Other business activities	50.4
70	Real estate activities	46.1
40	Electricity, gas, steam and hot water supply	45.1
32	Manufacture of radio, television and communication equipment and apparatus	44.8
51	Wholesale trade and commission trade, except of motor vehicles and motorcycles	42.7
11	Extraction of crude petroleum and natural gas; service activities incidental to oil and gas extraction excluding surveying	42.0
73	Research and development	41.3
24	Manufacture of chemicals and chemical products	35.9
31	Manufacture of electrical machinery and apparatus, n.e.c.	35.3
23	Manufacture of coke, refined petroleum products and nuclear fuel	35.2
64	Post and telecommunications	32.6
71	Renting of machinery and equipment without operator and of personal and household goods	31.4
33	Manufacture of medical, precision and optical instruments, watches and clocks	31.0

Source: OECD IT Outlook, 2004.

Some attributes of these methods are common across secondary software sectors:

1. Software as a cost centre, rather than a profit centre: in the secondary software sector, most software is a cost centre rather than a profit centre. This means that software may be used to reduce costs, but in itself costs money and is not directly a source of profit.
2. Some software may be a source of revenue, but most revenue is generated through activities for which software is not sold as a product in itself
3. A significant proportion of the software used provides an infrastructure rather than a competitive advantage, the costs for which could be shared with similar firms

Figure 49: Types of software produced, by sector



Source: MERIT, software innovation survey of firms, 2005. n=499. (responses overlap)

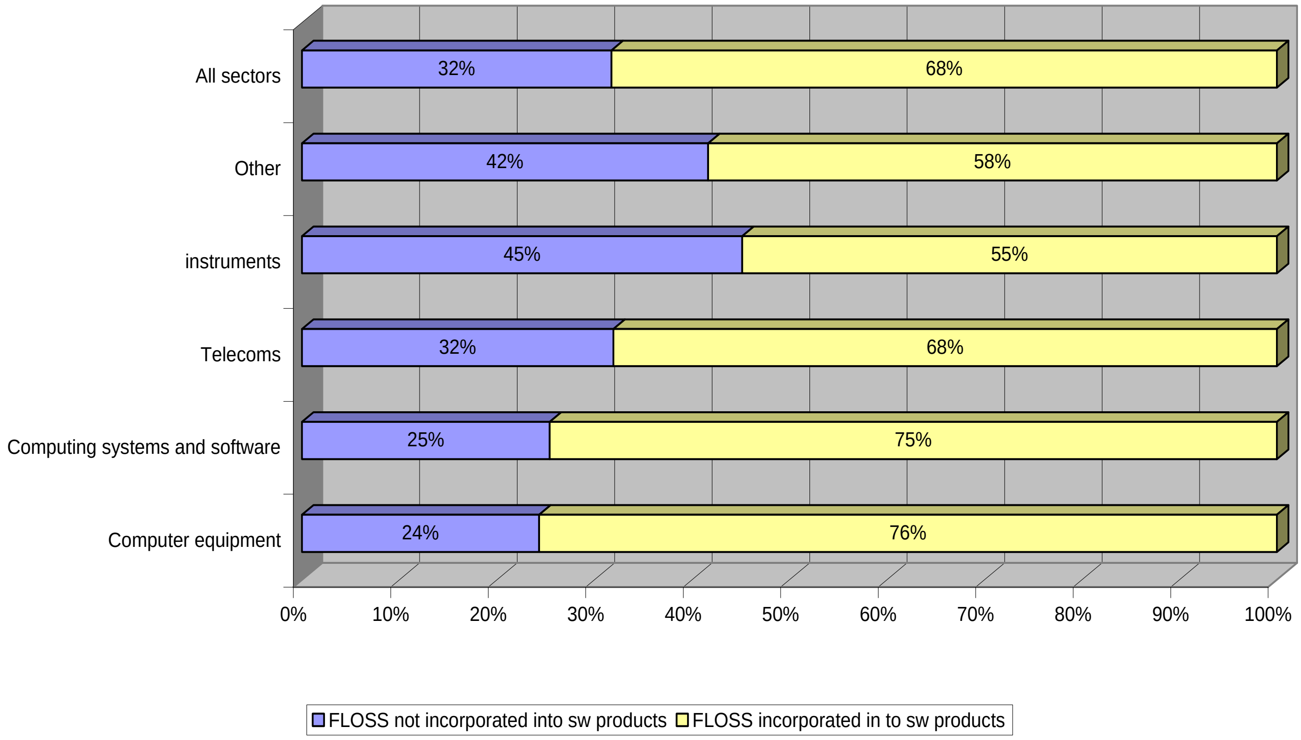
Figure 49 shows the types of software produced by sector, based on a survey of European firms conducted by MERIT for the European Commission DG INFSO's "*Study into the effects of patentability of Computer Implemented Inventions*". While the survey may over-represent software-producing firms, as well as computing equipment and software sectors, it should be representative of the type of software that is produced by firms in these sectors. It should be noted that except for computer equipment and software producers, the single largest type of software developed is for in-house use or, for hardware and telecoms manufacturers, embedded software (included in machinery, instruments or other equipment).

Figure 50 shows some remarkable findings: FLOSS is heavily used by firms in various sectors for their own software-related products or services[111]. The highest share of

[111] Respondents were asked: In the last three years, has your firm incorporated free Free or Open Source Software code in any of the following software products?

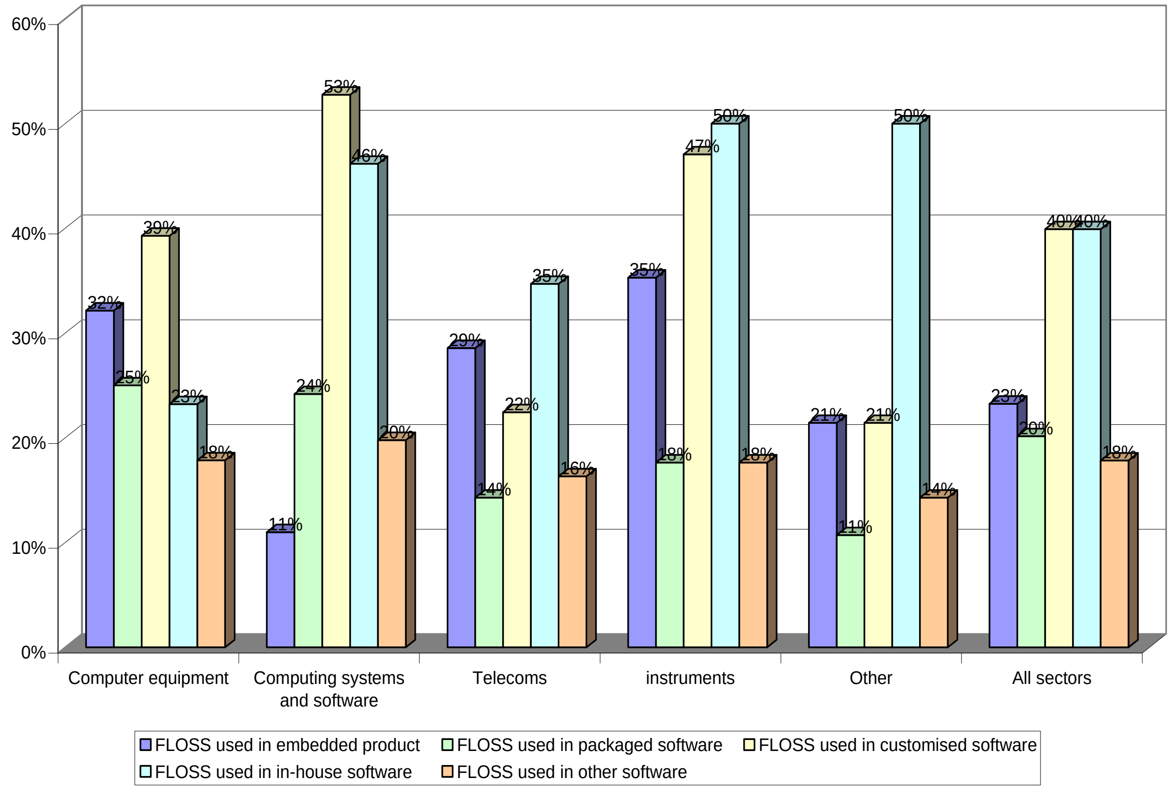
"no FLOSS use" is for the sector "Other", and even there 58% of respondents have incorporated FLOSS software in their own products or services. Across all high ICT intensity sectors surveyed, 68% of firms have incorporated FLOSS software in their own software products and services. As Figure 51 shows, FLOSS is incorporated into different types of software-based products – 40% each for customised or exclusively in-house software across all sectors. These results are particularly interesting as the topic of FLOSS was incidental to this survey, making it especially representative on this count, as there is no likely self-selection bias towards FLOSS-interested respondents.

Figure 50: Use of FLOSS for software products and operations, by sector



Source: MERIT, software innovation survey of firms, 2005. n=378.

Figure 51: Type of software products incorporating FLOSS



Source: MERIT, software innovation survey of firms, 2005. n=378.

Such high-intensity ICT users make up part of the wider ICT environment, as discussed in section 8.2, "The wider impact of FLOSS: beyond IT", following page 123. This wider ICT environment accounts for between 7.7 and 10% of European (and in general, OECD) GDP at the infrastructure layer, with a further 2.5% of GDP above for media and content. Of this, software products and services make up less than a fifth, at 0.63% and 1.73% of GDP respectively. For any given share of GDP devoted to software, almost five times as much again is accounted for in other ICT services built over software, or under it (in the case of hardware). In line with the estimated growing share of FLOSS in software investment (section 9.1, "Trends: use of software in Europe and the US" following page 176, especially Figure 60), we can thus credit between 1.5% (for infrastructure only) and 2.5% of GDP as "FLOSS related ICT" in 2005[112], increasing to between 2.4% (infrastructure only) and 4% of GDP by 2010.

This definition, while seemingly very broad (amounting to a lower bound of Euro 150 billion today rising to Euro 627 billion in 2010), may actually underestimate the potential FLOSS related share of GDP. This is because for information related services, network effects have a strong and cascading effect. FLOSS exhibits network effects stronger than

---

[112] Assuming FLOSS has a share of just under 20% in the value of software investment, rising to just above 30%, and applying this to the 7.7% of 2005 GDP estimated as the share of ICT infrastructure, we get 1.5% for FLOSS-related infrastructure; applying the same FLOSS share to the 12.5% estimate of the "wider ICT economy" share of GDP, we get 2.5% of GDP as the "FLOSS-related wider ICT economy" share. This assumes a multiplier effect for "FLOSS-related" similar to "software-related". This is not overly optimistic, as FLOSS is estimated to require a similar amount of ancillary support and services as proprietary software (some argue that FLOSS requires more support and services).

most technologies, as it can evolve and affect the market independently of vendor strategies (which – even for FLOSS vendors – invariably attempt to lock in customers through "differentiation" and thus reduce the benefits to them of network externalities). Thus, once FLOSS crosses a threshold that provides the "tipping point", whether through private business demands, public sector policy or individual "prosumer" action (as with the world wide web), it could have a snowballing effect in terms of enabling the shifting of products and services built to address proprietary platforms to interoperable – thus, frequently, FLOSS – platforms.

Much of this effect on the non-core ICT sector, or the secondary software sector, will be through R&D substitution. As an indirect effect, this is described in further detail in section 8.3, "FLOSS, R&D substitution and the impact of collaborative strategies" following page 140.

More direct effects come in two forms:

1. platform substitution – related to, but different from R&D substitution
2. maintenance cost sharing

7.7.3. Platform substitution

Platform substitution, for developers of services or products above a software layer, involves replacing the current layer they use with FLOSS. The current layer may be a complete proprietary software system, modifiable and reusable components licensed from third parties, or technology developed in-house. Platform substitution need not involve replacing existing product or service lines – it could take the form of platform augmentation, when a product or service line is replicated onto a new (FLOSS) platform. Platform substitution is fairly straightforward, and quite common. The ability to engage in platform substitution is one reason firms are increasingly concerned about interoperability and open standards, as the less interoperable their current platforms are, the harder it becomes to replace them with alternative platforms.

However, the possibility of platform substitution is one reason why many studies that credit to a platform vendor entire sections of the economy that is built above a particular platform tend to misrepresent the true picture. Services above a platform are valuable in themselves; such service providers are often in a position (and would usually like to be in the position) to switch platforms whenever they choose, if market conditions so suggest.

In the context of FLOSS, platform substitution is pretty simple: the Linux kernel and/or the GNU/Linux operating system is the platform that appears to substitute in one variety or another in several application domains.

A remarkable example is that of Internet networking devices: routers, switches, wireless access points. Once dependent on proprietary, usually in-house platforms, they almost all run GNU/Linux today, except for high-end products from vendors such as Cisco. (Though the consumer-oriented retail products from Cisco's subsidiary Linksys are mostly Linux-based.)

As an example of how platform substitution extends to new markets, the voice-over-Internet Protocol (VoIP) routers and consumer devices available today not only run Linux, but might have been too expensive to reach the level of popularity they have today if they were built on an expensive specially developed (or externally licensed) platform.

Another striking example of platform substitution, also one resulting in the exploitation of new markets is the near epidemic within the past two years of Linux-based smart phones. In a market growing at 85% annually, Linux has a 23% share, making it the second-placed in the market after Symbian (51%). The Linux share has more than doubled since 2004, and is poised to grow even further this year. What is striking is that Linux is barely visible in smartphones sold in Europe and the US. Most of the growth in Linux smartphones is in Asia, especially China, where on current growth trends Linux will soon become the dominant smartphone platform.

A number of large multinational firms drove growth, with Motorola as the main western manufacturer – which launched its first Linux phone in 2003, and had Linux on 25% of its new models in 2005[113]. Other global Asian firms have been involved too, such as LG, NEC, Panasonic and Samsung, all of whom have numerous Linux phones available in Asia. But the huge growth in Asian phones has been in large part due to the rapidly growing Chinese brands that are increasingly looking for markets abroad, such as Datang, Haier, Huawei and ZTE – all of whom rely on Linux as their platform.

Other manufacturers for whom the platform software is a cost rather than profit centre have been increasingly adopting Linux. E.g. Siemens Medical Solutions in September 2006 adopted Linux for its ultra-high-field magnetic resolution imaging (MRI) machines, because Linux “enable[s] Siemens to dramatically expand the processing capabilities of our measurement operating systems, without a costly overhaul of the technology.”[114] Firms such as Philips have been using Linux in several markets, including in digital media broadcasting devices such as set-top boxes[115], semiconductors and even as investors in embedded Linux firms.[116]

7.7.4. Maintenance cost sharing

A significant motivator for secondary software firms to release their own software as FLOSS is to share maintenance costs with other firms in their sector. As much software is developed for in-house usage, and this software is never directly generating profits, it operates as a cost centre. Profits are generated by the firm’s specific *use* of the in-house software, but as for the in-house software itself, the less expensive it is to create and maintain, the better.

Reducing creation costs is a reason to use FLOSS, allowing firms to build their in-house software from existing, freely available components, and only requiring them to spend effort on customisation and adaptation to their specific needs. Building upon existing FLOSS components ensures that the FLOSS component developers, rather than the firm’s in-house developers, maintain a sizeable share of the codebase. But to reduce costs further, organisations can choose to release the software developed for in-house use. When other organisations have similar needs, they may be encouraged to use it (saving on their own in-house development efforts), share modifications and thus also help maintain it. This is not

[113] See reporting on linuxdevices.com

[114] see press release at http://www.novell.com/news/press/item.jsp?id=1087&locale=en_CA

[115] see PRNewswire, September 17, 2001, “Philips and Lineo Partner to Provide MHP on Linux-based Digital Media Solution”, http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=104&STORY=/www/story/09-17-2001/0001573450&EDATE=

[116] Timesys, see http://www.philips.com/about/company/participations/index.html

just true for firms, but also in the public sector where increasing efforts have been made especially in Europe (but also in the US, see governmentforge.org) to bring together public authorities' software development skills and needs: a 2006 EU-wide survey of public authorities conducted by MERIT for the European Commission's DG INFSO[117] built upon a survey of Dutch government authorities to show that almost 8% of public authorities are releasing some software under FLOSS licences, and 12% own software that they may wish to release.

Maintenance cost sharing does not happen automatically, as firms cannot expect to publish in-house software as FLOSS and expect armies of volunteers (or even other peer firms) to start supporting them. They need to build communities of interest, following guidelines such as those detailed by MERIT for the European Commission's IDABC Open Source Observatory[118]. There are even firms that provide consultancy and support services to help other firms create such communities, and to help the development, release and maintenance of in-house software as FLOSS.

One example is how CollabNet, a consultancy and support firm created by the founder of the FLOSS web server Apache, helped the investment bank Dresdner Kleinwort (then Dresdner Kleinwort Benson) develop and release the OpenAdaptor toolkit as FLOSS.

OpenAdaptor provides a flexible toolkit to build interoperability between several applications, information systems, data sources and platforms that were not designed to interoperate. This is a typical situation at several large organisations that are heavy users of IT, especially the finance sector where legacy applications, hardware and software remain in use for decades and several streams of proprietary data sources designed for independent use could be usefully combined. Originally developed within Dresdner Kleinwort (DrKW) in 1997 as a set of tools based on XML and Java, OpenAdaptor was used for over 50 projects in the bank's various locations around the world, and the bank was distributing its source code informally to customers and partners. The firm recognised the value of sharing to reduce maintenance costs:" We certainly didn't want to become a software company," explained Jonathan Lindsell, Global Head of IT Business Development. "We shipped source code to our customers all along, inviting them to fix bugs and contribute enhancements, and they did. That's one of the reasons why it's become so incredibly successful."[119]

As described by CollabNet, which provided support for the formal "open sourcing" of OpenAdaptor, in 2000, to obtain even more open adaptor contributions from external sources, and expand its utilization, DrKW decided to formally release the software under a FLOSS licence since "open adaptor was already a big win for [them] and [their] customers, and [they] wanted to make it an even bigger win by enlisting help from a much larger community," according to Marc Eno, DrKW's Open Source Initiatives Manager. Although the firm knew they were helping competitors, which was not seen as a bad thing since "we were improving the lot of everyone involved, including our customers and ourselves." Such logic was reasonable since OpenAdaptor was never a direct source of profits – after all, the bank did not want to become a software company.

---

[117] As part of the PS-OSS project, see www.publicsectoross..info

[118] Rishab Ghosh, Ruediger Glott, Gregorio Robles, Patrice-Emmanuel Schmitz, 2005. **Guidelines for Public Administrations on Partnering with Free Software Developers –** published by the European Commission at http://europa.eu.int/idabc/servlets/Doc?id=19295

[119] Collabnet's case study of OpenAdaptor.

OpenAdaptor is now very widely used, initially within the financial sector – reflecting the peer community-based nature of dissemination in the FLOSS software model – but now well beyond.

It certainly saved DrKW large – if unknown – sums of money, time and effort, and added value to the economy and to peer firms without reducing value for DrKW. Indeed, it reduced costs for DrKW not only directly, as a useful application, but by distributing maintenance across several other organisations.

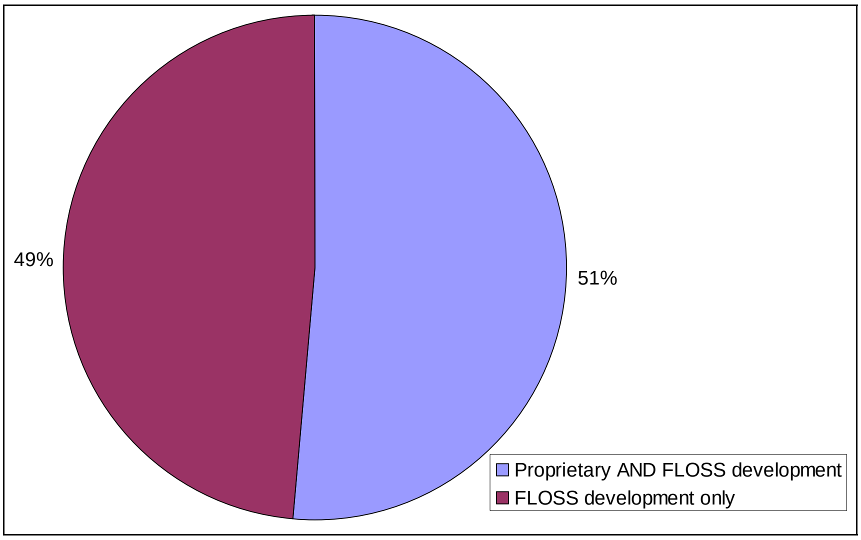
Given that in-house software already accounts for a large share of software developed as well as a large share of software developers in employment, and that it is inherently not in conflict with the FLOSS model – no revenue streams are threatened – we expect increasingly large amounts of the vast store of in-house software that is constantly being developed to be released as FLOSS. This would enrich the FLOSS community and the developing firms alike.

8. Indirect impact: FLOSS, innovation and growth

Software, and ICT in general, are characterized as "*cumulative" technologies* that gradually evolve through incremental innovation and integration of various inventions, algorithms and techniques (Merges and Nelson, 1990). The proper operation of software products requires also interoperability with operating systems and other applications that demand disclosure of technological information to assure an appropriate development of programs by other parties (David and Greenstein, 1990; Cohen and Lemley, 2001). Therefore, the accessibility to knowledge and to specifications of new standards advances is vital for the build-up of new inventions on the basis of present knowledge.

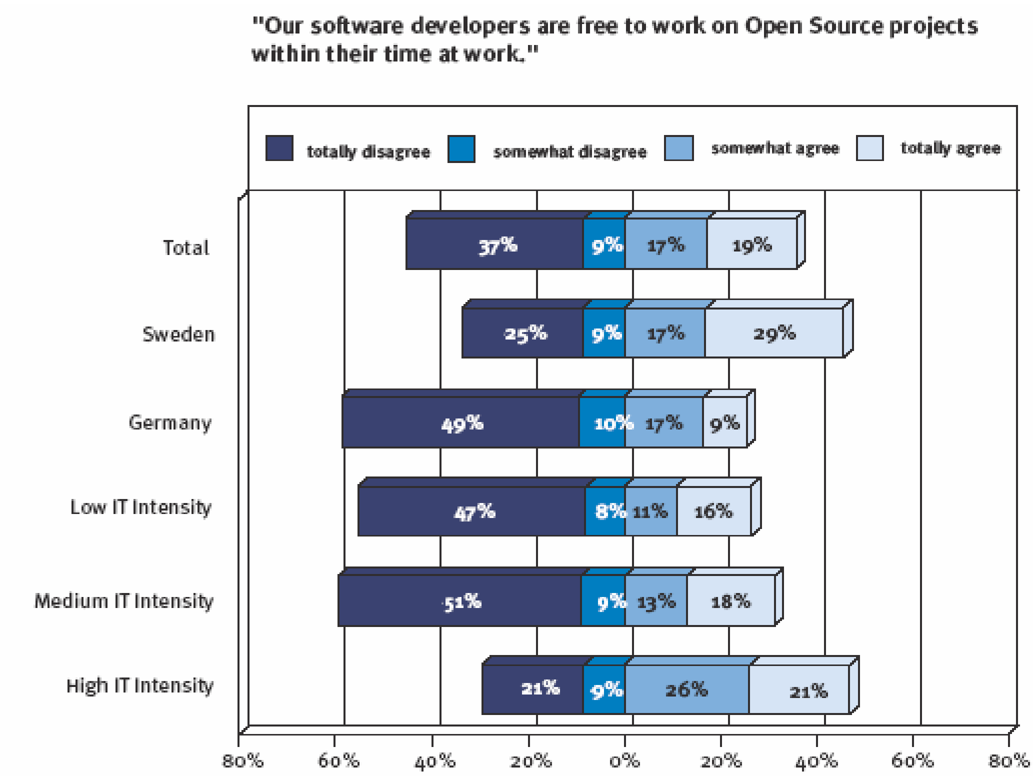
In general, the causality between the contribution of FLOSS to proprietary software and *vice versa* has not been fully clarified. However, theoretical studies on the significance of knowledge codification and the formation of knowledge externalities to innovation and technical change (Cowan and Jonard, 1999; Cowan *et. al.*, 2000) and considerable empirical and the evidence from the FLOSS developer survey and the FLOSSPOLS Skills Survey suggest that FLOSS and proprietary code co-evolve by establishing reciprocal exchange of technical know-how and enhancement of developers' skills by participating both in FLOSS and in-house projects (Figure 52).

Figure 52: FLOSS developers also develop proprietary software.



Moreover, research conducted suggest that a substantial share of programmers in commercial firms participate in FLOSS projects in their free time and even during their working day, as a part of their in-house training activities. One of the significant findings of the FLOSS User survey was that 47% of firms in high IT intensity countries (including ICT industries as well as other sectors such as banking and finance) agree that their software developers are free to work on FLOSS projects within their time at work (Figure 53). The corresponding figure for low IT intensity firms (sectors such as retail and tourism) was as high as 27%. A 2006 IDC survey[120] of 5 000 developers working in organisations in 116 countries worldwide confirmed that 71% use open source software.

Figure 53: FLOSS development allowed on employer time.



Copyright © 2002 Berlecon Research. Source: FLOSS User Survey (coordinated by MERIT)

There is considerable evidence on the relationship between FLOSS development, innovation and the ICT industry, of which some examples have been provided above. With regards to innovation, it may be useful to compare the role of FLOSS, which in its nature supports diffusion of knowledge, to patents that are justified for, among other things, promoting disclosure of knowledge and resulting innovation. In fact, patents have been found empirically to be a poor means of promoting disclosure.

Arora et al (2003)[121] find that "patent disclosures appeared to have no measurable impact on information flows from other firms, and therefore no measurable effect on R&D productivity". Arundel (2001)[122] finds that "a consistent result in survey research on the use

[120] IDC 2006, "Open Source in Global Software: Market Impact, Disruption, and Business Models".

[121] Arora, A. et al., 2003. "R&D and the patent premium", Nat'l Bureau of Econ. Research, Working Paper No. 9431. p17. Available at http://www.nber.org/papers/w9431

[122] Arundel, Anthony. "Patents in the Knowledge-Based Economy", *Beleidstudies Technologie Economie 67;*

of patent databases is that they are among the least important external information sources available to firms". His analysis of 12 445 firms' responses to the CIS survey results[123] shows that between 5% and 18% of small and medium-sized firms find patents to be a useful source of information[124].

In the case of software as a component of ICT innovation, there is interesting data comparing FLOSS and patents from the on-going study for the European Commission's DG Information Society. The surveys show (Arundel et al 2006[125]) that more firms think free software source code is an important source of new ideas (17%) than patent databases (5%). The firms are from a broad spectrum of ICT producing and ICT using sectors, from software to medical instruments and automobile manufacturing. The opinion of individual innovators (engineers) is perhaps more relevant as questionnaires on patents sent to firms are more likely to be answered by the legal department than by innovators. Far more innovators within firms[126] think source code (41%) is moderately or very important sources of new ideas, compared to patents (24%).

While we do not know how much of this software that is a source of new ideas is licensed under reciprocal terms, these data show that open source software is succeeding in providing disclosure to a much greater degree than patents. This is certainly at least in part due to reciprocal licenses such as the general public license (GPL) that provide a legal requirement to disclose (much as patents are supposed to do). If a legal framework is required to promote disclosure and follow-on innovation, there is therefore some evidence to justify an argument that reciprocal open source licensing provides a more effective framework than the current patent regime.

Studies by e.g. Meijers, 2004, carried out within the MUTEIS and NewKInd projects for the European Commission, show there are clear spillovers from the use of software and telecommunications that increase total factor productivity growth. This indicates a positive and strong relation between the use of ICT (and of software and telecommunication in particular) and the creation of knowledge across sectors of the economy.

The link between innovation and economic growth is widely recognised and confirmed by findings in the European Innovation Scoreboard (for which analysis is performed for DG Enterprise by MERIT's Hugo Hollanders and Anthony Arundel). In particular, as the European Innovation Scoreboard 2004 Report (EIS 2004) shows, "Computer and related activities" and "Business services" are the two most innovative service sectors in Europe, and "Electrical and optical equipment" the most innovative

---

[123] Arundel A. (2000), "Patent – the Viagra of Innovation Policy?", *Internal Report to the Expert Group in the Project "Innovation Policy in a Knowledge-Based Economy"*, Maastricht, MERIT. Figure 4, page 15. Available online at http://www.ebusinessforum.gr/index.php?op=modload&modname=Downloads&pageid=320

[124] The share is 34% for large firms, but even they find patents *less* useful than other sources of information, such as customers, suppliers, conferences and journals, trade fairs, and competitors.

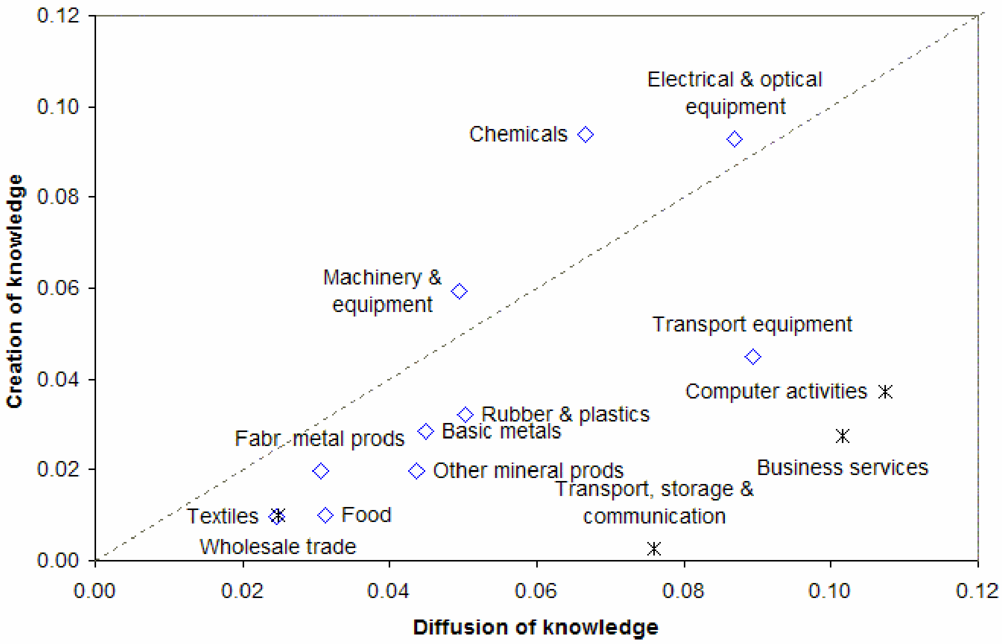
[125] Arundel, A., Bergstra, J., Feijoo, C., Ghosh, R.A., Glott, R., Hall, B., Klint, P., Martin, A., Thoma, G., and Torrisi, S. 2006. "Empirical Study of economic impact: Approach and preliminary findings". *European Commission DG INFSO*, unpublished. Part of the "Study of the effects of allowing patent claims for computer-implemented inventions".

[126] Arundel et al 2006 shows consolidated data for all respondents; figures included here are for individual innovators employed at private companies, i.e. excluding those employed at public organizations or research institutes. The latter are, as may be expected, even *less* likely to look to patents for new ideas.

manufacturing sector. As can be seen in Figure 54, these ICT-related sectors rely much more than other, less innovative sectors, on diffusion of knowledge as their mechanism of innovation – and less on the creation of knowledge, especially for the services sector[127]. This is indicative of the nature of innovation in ICT — it is incremental, relying less on proprietary in-house creation of new knowledge and more on building quickly and repeatedly upon knowledge diffused across the industry.

By definition, FLOSS provides far more diffusion of technology than proprietary software, especially to potential future innovators who are not faced with the search costs of locating sources of new innovation buried within proprietary software (they can search published source code, as indeed URJC does to identify types of new software for research purposes). Nor are potential future innovators faced with transaction costs of procuring licences to reuse and combine FLOSS software from different producers in order to build further incremental innovations, and of course the cost of licences itself is zero, at least with respect to copyright and proprietary trade secret acquisition.

Figure 54: ICT services sectors innovate through diffusion of knowledge.



Copyright © 2004 European Communities. Source: EIS Sector Scoreboard 2004 report prepared for the EC by MERIT's Hugo Hollanders and Anthony Arundel

The remainder of this section analyses:

- The impact of capitalisation and competition on innovation and the impact of FLOSS for ICT;

[127] For Electrical and optical equipment manufacturing, clearly, a lot of proprietary knowledge creation is also involved due to the nature of the innovation that is product-based, as well as to the definition of this sector that goes well beyond fast-moving ICT innovation.

- The wider impact of ICT beyond IT firms and the interaction between FLOSS and different parts of this wider ICT market;
- How FLOSS can substitute for R&D providing more cost-effective innovation, and how collaborative methodologies can extend to sectors beyond ICT
- The impact of ICT innovation on the wider economy
- A model for growth and innovation simulating the impact of FLOSS on the economy at large

8.1. Competition, innovation and FLOSS

In the last thirty years, major differences have arisen between companies with regard to stock capitalization and other forms of access to capital. In some branches, a number of key companies have acquired strong and increasingly global intellectual property monopolies on elements that can be reproduced at quasi-zero cost because they can be represented as information. This has led, in a number of branches or companies, to an unprecedented disjunction between costs (including R&D costs) and prices for the related goods. These monopolies can take the form of trademarks (in food and luxury goods), copyright (for software and media) or patents (IT, pharmaceutics and other bio-industries). Their effects are generally coupled with other market power factors such as promotion (luxury goods, food, pharmaceutics, media), control of distribution (media, telecom, some food products), and network effects (software, telecom).

Table 23 gives the ratio of capitalization to turnover for a number of major companies in various branches and the average value for companies listed in one branch[128]. One can see that, even without taking into account smaller or privately owned companies, there are extreme differences between branches (12:1 ratio between the car industry and pharmaceutics) and within a branch (37:1 ratio between Google and Cap Gemini, or 15:1 ratio between Microsoft and Cap Gemini, the example of Google being at the same time an example of the extreme importance of network effects and an "outlying" value). This ratio is classically interpreted by financial analysts as an indicator of the expectation of profitability, and is indeed strongly correlated with the profit margin of companies. Interpreting its relation with innovation and innovation targets calls for a much more in-depth analysis.

For companies that are not publicly quoted (smaller or privately owned), a similar ratio cannot be computed. However, the expectations of risk capital funds (or sometimes even public research programmes) somehow transport similar trends to these companies.

8.1.1. Innovation and innovation targets

It is clear that companies that generate strong profits have a strong market capitalization or strong means of access to innovation-related funds, and thus have stronger means of investment in R&D. More precisely, it is difficult for companies who lack these properties to mobilize resources for direct (in-house) R&D or even in less radical innovation. These gives rise to two radically different lines of interpretation:

The first interpretation identifies the generation of intellectual property (IP) titles and other market power mechanisms (for instance, network effects) with innovation, considers them as products and generators of innovation. Companies that exhibit high values for the ratio would be rewarded for past innovation by being provided with the means to continue to innovate.

The second interpretation considers that, on the contrary, there is a major failure of resource allocation to innovation. The companies who, because of their IP monopolies and other market power mechanisms such as network effects, have very strong margins and strong capitalization are not incited to invest in further innovation (and likely to cannibalise their existing business(es)) and are incited to choose research and innovation targets that are

[128] This is not the branch average, which is impossible to evaluate as many companies are not publicly traded in a branch

mostly protective of their existing profit lines or business models (for instance, investment in only superficially changed technology, or technology for the execution of IP rights or deployment of such technology). Companies who have a real business incentive to innovate because it is the only way for them to create or conquer markets or raise their margins of profitability are starved of investment means. The situation can be even worse in fields where the most important innovation is of a nature that does not lend itself to IP capitalization, because it is systemic in nature (public health) or arises from hard to predict usage (some non-voice communication usage, social software, self-production of contents by individuals). In these fields, innovation will simply be absent if other means are not put in place to support it.

The extreme differences in the figures presented in Table 23, in particular within the ICT and media industry, suggest it is critical to further progress in understanding to which degree each of the lines of analysis represents reality.

Table 23: Capitalisation and turnover by sector.

Sector or company (country)	Capitalization B€	Turnover B€	Staff (milliers)	Capit. / staff member K€	CA / staff member K€	Capit. / Turnover
Car industry	**295.9**	**854.3**	**2045**	**145**	**418**	**0.35**
Distribution	**291.2**	**587.9**	**3146**	**93**	**187**	**0.50**
Insurance	**268.1**	**315.8**	**607**	**442**	**520**	**0.85**
Reinsurance	**39.2**	**41.5**	**49**	**801**	**846**	**0.95**
Oil	**1,120.5**	**1,024.7**	**954**	**1,175**	**1,074**	**1.09**
Banks	**664.9**	**358.1**	**787**	**845**	**455**	**1.86**
Coca-Cola (US)	84.6	18.0	50	1,692	360	4.70
Pepsico (US)	76.2	23.9	153	498	156	3.19
Danone (FR)	24.5	13.7	89	275	154	1.79
Cadbury Schweppes (UK)	17.9	10.4	56	320	186	1.72
Kraft Foods (US)	42.8	26.4	98	437	269	1.62
Nestlé (CH)	95.5	59.5	253	377	235	1.61
Uniliever (UK/NL)	59.4	44.2	234	254	189	1.34
Food	**400.9**	**196.1**	**933**	**430**	**210**	**2.05**
Luxury goods (LVMH, F)	**26.9**	**12.6**	**60**	**450**	**211**	**2.13**
Google	82.4	4.9	5.7	14,456	860	16.82
Microsoft (US)	210.6	31.3	60	3,511	521	6.74
SAP	48.5	8.5	36	1,347	236	5.71
Oracle (US)	48.2	9.4	50	964	189	5.10
AOL-Time Warner (US)	70.4	33.7	85	828	396	2.09
Nokia (FI)	64.5	34.2	59	1,093	580	1.89
IBM (US)	129.8	77.0	329	394	234	1.68
Vivendi-Universal (FR)	26.3	21.4	38	693	563	1.23
Philips (NL)	31.5	30.4	164	192	185	1.04
France Telecom	42.8	47.2	203	211	233	0.91
Siemens (DE)	56.2	75.5	461	122	164	0.74
Cap Gemini	3.1	6.9	61	51	113	0.45
Software-IT-media	**814.3**	**380.4**	**1551.7**	**525**	**245**	**2.14**
Sanofi-Aventis (FR)	94.2	15.0	76	1,239	197	6.29
Novartis (CH)	108.1	23.1	79	1,368	292	4.68
GlaxoSmithKline (UK)	120.0	30.0	101	1,188	297	4.00
Pfizer (US)	156.5	43.0	264	593	163	3.64
Abbott Laboratories (US)	55.7	16.1	72	774	224	3.46
Bayer (DE)	23.4	29.7	115	203	259	0.78

Source: *Alternatives économiques* complemented by Ph. Aigrain / Sopinspace for the software, IT and media, luxury goods and reinsurance sectors. Based on company financial reports for 2005 or 2004.

8.1.2. Existing evidence

The economics of innovation have been primarily focussed on measuring the input into innovation (through R&D figures as accounted by firms or public funding) and its output measured from quantitative indicators in which IP ownership (and in particular patents) plays a key role. The part of the approach concerned with research and development (R&D) output is of course unfit to answer the question raised above, since it has a built-in assumed answer. It is also ill adapted to fields or regions in which patents play a limited role.

Despite these limitations to existing research, some evidence has started to emerge in the recent years both on the level of R&D spending and on innovation targets. Work by Bessen and Maskin[129] and Bessen and Hunt[130] on software patents has shown that the existence of patent mechanisms acted as a minor disincentive to R&D spending, and that this effect was stronger in firms that patented most. This work has been criticized on methodology by other researchers such as Robert W. Hahn and Scott J. Wallsten[131] and Bessen and Hunt have responded[132] to this criticism. In any case, these spending effects are by far less important than evidence that starts to emerge on innovation targets.

It is the innovation crisis of the pharmaceuticals industry that has given rise to pioneering work on explaining innovation behaviour based on existence of strong IP ownership. The leading innovation economist Frederic M. Scherer has demonstrated[133] that protecting and seeking continuation of its monopoly rents mostly explained the R&D behaviour of the pharmaceuticals industry. This finding corroborates the analysis on the benefits of new drugs that has shown a vastly predominant innovation targeted on “me-too drugs”, drugs with no new medical value but which replace existing drugs whose patents are due to expire.

Similar research has not been conducted on software innovation, and will be hard to conduct in a similar manner given there are no procedures similar to clinical trials to assess objectively the benefits or drawbacks of new software. In addition, network effects that exist independently of IP ownership play an important role in market power in the ICT industry, unlike pharmaceuticals. However, there is a long record of more qualitative findings by technology analysts. Already in the 1970s and early 1980s, the strategy of the then dominant IT company (IBM) to recruit key researchers and neutralize them in golden research retreats from which no practical innovation would emerge – not for IBM, but more importantly, not for competitors – was an object of humour. However, until relatively recently, IT remained a domain of widely open innovation. .

---

[129] James Bessen and Eric Maskin, *Sequential Innovation, Patents, and Imitation*, working paper, Research on Innovation, http://www.researchoninnovation.org/patrev.pdf

[130] James Bessen and Robert M. Hunt, An Empirical Look at Software Patents, Research on innovation working paper n°03/17R, http://www.researchoninnovation.org/swpat.pdf

[131] Robert W. Hahn and Scott J. Wallsten, A Review of Bessen and Hunt's Analysis of Software Patents, American Entreprise / Brookings Joint Center on Regulatory Studies working paper, http://www.researchineurope.org/policy/hahn_wallsten.pdf

[132] James Bessen and Robert M. Hunt, *A Reply to Hahn and Wallsten*, http://www.researchoninnovation.org/hahn.pdf

[133] Frederic M. Scherer, *Global Welfare and Pharmaceutical Patenting*, The World Economy, July 2004. See also his intervention at the International School of Economic Research on Intellectual Property, Siena, 2004.

The main trends that have been observed more recently in ICT innovation are:

- Algorithmic patents lead to frequent repetition of research without significant innovation and use of sub optimal techniques in commercial software[134].
- Licensing of all types of IP ownership for public research[135] provides little resources for public research, but has a major impact on selection of innovation targets[136] - for instance as many as 10% of firms in ICT producing and ICT-intensive sectors change or avoid lines of research because of concerns that others have patents in that area[137].

Innovation targets have also been affected by trends that are no longer related to patents but to copyright enforcement technology, in particular Digital Rights Management and trusted computing. A major reallocation of research efforts from user-oriented functionality in content processing (which increases competition and consumer benefits) towards supplier IP protection technology (which decreases competition and essential innovation) has been observed as early as ten years ago in the EU research programme in the field of multimedia and audiovisual networks, as well as in standardization. This process has recently accelerated (see for instance recruitment by Microsoft and Intel of leading content processing researchers to conduct work on copyright protection or enforcement).

Finally, a question arises regarding the difference in innovation behaviour for players who have acquired market power predominantly through network effects and winner take all effects rather than based on IP ownership allowing the arbitrarily pricing of information goods reproduction. These situations are rarely seen in isolation: Microsoft depends on network effects as well (mainly) on IP ownership, and Google depends on IP to some extent though mainly relying on network "winner-take-all" effects. It seems that network effects have much less bias in determining innovation targets as illustrated by the innovation behaviour of Google since becoming publicly traded (even though Google has started to exhibit the "innovative company acquisition" behaviour that is often associated with large stock capitalization). This finding is relevant to the discussion in the last section, as network effects are also at work in FLOSS and can even work at a faster degree in it.

8.1.3. Relationship with FLOSS

FLOSS-based innovation or innovation in a domain where FLOSS plays an important role is characterized by a situation where prior innovation is readily available for further

[134]See Philippe Aigrain, *11 questions on software patentability issues in Europe and the US*, Software and Business Method Patents: Policy Development in the U.S. and Europe meeting, organised by The Center for Information Policy, University of Maryland on December 10, 2001.

[135] See work published at the occasion of the 2àth anniversary of the Bayh-Dole act such as the set of papers gathered in Technology Innovation and Intellectual Property 2003(5), special issue on "Patents and University Technology Transfer", and work regarding more specifically algorithmic software patents in Philippe Aigrain, 11 Questions of Software Patentability if the US and in Europe, Software and Business Method Patents: Policy Development in the U.S. and Europe, Center for Information Policy, University of Maryland, 10 December 2001.

[136]See for instance study conducted by Philippe Aigrain on the R&D and technology transfer strategy of the IRCAM musical research institute. Note that technology transfer departments of universities often deny such effect when questioned in the studies quoted in the previous footnote, but it is definitely observed in the biotech, biomedical, software and other research fields.

[137] Arundel et al 2006, see footnote 125.

building upon, and in which new functionality can be disseminated extremely rapidly, with a low entry cost in trial or usage. FLOSS favours permanent innovation processes and rapid market or usage deployment. FLOSS is prone to network effects as much as proprietary software. FLOSS innovation can lead to capture of positive externalities by its originators, in particular using first to market and network effects, but in forms and to a degree that is not comparable to the capitalization based on IP ownership that monopolize the free reproduction of information artefacts. In particular, FLOSS based activity naturally supports interoperability through open standards. Defined in economic terms these are standards that move towards a natural monopoly in the technology (as with all standards) but ensure full competition in the market for supply of the technology, unrestricted by IP ownership[138].

On the basis of the evidence presented in the previous sections on the existence and likelihood of an increasing innovation resource allocation failure, it is worth considering how FLOSS-based innovation and a rebalancing of innovation incentives to make the innovation environment more equitable to all forms of innovation could act as a corrective.

One example of such policy measures is the disjunction that has been recommended by scientists and policy advisers in the field of medical R&D[139] between the market for innovation and the market for products. The idea being that there should be a market for innovation with rewards for successful innovation, but these rewards should not take the form of monopoly pricing on the resulting products or services. However, this attractive idea is difficult to implement in domains where one has little means of determining the degree of usefulness of an innovation at a sufficiently early stage, and is thus difficult to generalize to software, for instance.

Approaches more adapted to the situation of innovation that can only be assessed after significant deployment would have to directly tackle the comparative solvency of innovation models by decreasing the intensity or scope of IP ownership and/or by creating incentives for investment in commons-based and commons-producing innovations – which in the case of software mean FLOSS-based innovation. The latter can be achieved by tax credits measures, action on accounting rules for R&D and adaptation of funding rules of R&D subsidy and research programmes to support results being disseminated under commons regimes such as FLOSS.

Even measures that encourage FLOSS production or deployment by other means have a rebalancing effect on innovation models: the more FLOSS is available in a given domain, the less players using strong IP ownership capitalization strategies are able to impose arbitrary prices. This has been observed extensively for drugs for indications where generic drugs (even of different therapeutic effects) are available, and is presently arising for software in applications such as office suites, where, for instance, customers use the threat of competition from FLOSS software applications to achieve significant price reductions while purchasing proprietary applications.

[138] See Ghosh, R.A. 2005c. "An economic basis for open standards", FLOSSPOLS project report, *European Commission DG INFSO.* Available at http://flosspols.org/deliverables/FLOSSPOLS-D04-openstandards-v6.pdf#search=%22ghosh%20open%20standards%20economic%22

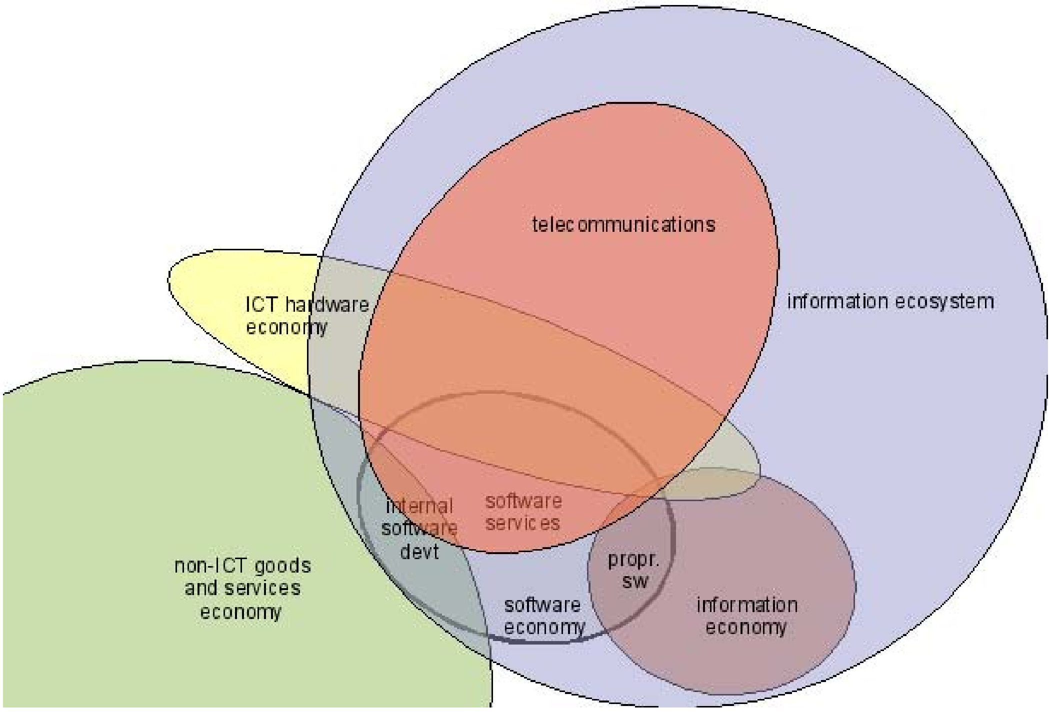
[139] Tim Hubbard and James Love, *A New Trade Framework for Global Healthcare R&D*, PLoS Biology 2(2): e52 DOI: 10.1371/journal.pbio.0020052

8.2. The wider impact of FLOSS: beyond IT

It is of course essential to analyse the impact of the development and deployment of FLOSS on the software economy and the ICT economy overall. A transformation in an essential generic technology such as ICT can deliver its possible benefits in Europe only if European economic players are truly able to develop business strategies around it, if new and existing companies can use it to their advantage. However, can one stop here, considering only this narrow reference universe?

In this section, macroeconomic figures are presented that illustrate the complex relationship of ICT innovation with the non-ICT secondary industry on one side, and the growing development of a sphere of non-commercial information creation and exchange activities. This analysis provides a background against which six questions on FLOSS' contribution to the European economy and the European information society are proposed and discussed.

Figure 55: The wider ICT ecosystem



Source: Philippe Aigrain, Sopinspace.

As shown in Figure 55, the software economy is only a part of software: a growing part of software development is done outside economic transactions[140]. This is the case not only for part of FLOSS software development and deployment but also for end-user development in all its forms (from spreadsheet formulas to macros, from Web page HTML

140 By “outside economic transactions”, what is meant is that the activity of software development or deployment is not the object of an economic transaction. Of course the means used for this activity such as computers, connectivity, consumables, printed documentation, sometimes travel are the object of economic transactions.

code to linking, forms and scripting). Recent schemes such as the "programming by configuration" used on content management systems and application servers expand this realm of end-user development.

Within the part of software that is in the monetary economy (in the sense of being the direct object of economic transactions though salary, services or sales of goods) – we call this the *software economy*; proprietary software license sales represent only a small part. Internal development of software by organisations[141] (companies and administrations) is greater than proprietary software licence sales by a factor of more than 1.5 in the EU and more than 2.5 in the U.S.[142], according to a study[143] by the FISTERA network (see Table 24).

Table 24: The software economy: sales, services and in-house.

Region	Proprietary software licenses	Software services (development and customisation)	Internal development
EU-15	19%	52%	29%
U.S.	16%	41%	43%
Japan	N/A	N/A	32%

Source: FISTERA thematic network.

Externally provided software services represent more than twice the figure of proprietary software licenses (conservative estimate since the higher FISTERA estimates include some IT services that are not software per se).

It should be made clear that the majority of software produced is not in software companies, or in the secondary software sector of producers of equipment that incorporates software (medical instruments, to stick to the same example). Other than the data collected by FISTERA, various official statistics show that one of the biggest software producers are in fact software users creating custom software for their own needs: in-house or "own account" software estimates vary between 20%-40% of the whole software production market in terms of value *as well as* employment, with a further 40-50% of the market in custom software (tailored to user needs by external businesses).[144] The software economy itself (which again

---

141 The share of internal development remains poorly measured, as it can be accounted for only when strict analytical accounting to this effect is done, or through specialised studies. Pioneering work was done with the specialised surveys conducted by the Norway and New Zealand statistical institutes at the end of the 1990s. The joint task-force of OECD and Eurostat (2001-2002) was able to demonstrate a gross underestimation of internal development in many countries and provided the first detailed global figures in *Measuring investment in software*, OECD/STI Working paper 2003/6. The figures provided by this report have confirmed the previous estimates in the pioneering surveys.

142 This ratio of internal software development to proprietary software package may be interpreted as an indicator of the maturity of skills within the workforce and of the strategic independence of companies towards major proprietary providers. If one does so, it provides a problematic assessment of the situation of the European organisations compared to the US.

143 Estimating the software activity in the European industry, report of the FISTERA thematic network on foresight in information society technologies in the European research area. The EU-15 figure for internal development is an estimate from main countries figures measured in 1999. Distribution between proprietary software and software services is an estimate for 2002.

144 See, e.g. Parker, R. & Grimm, B. 2000, "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98", US Bureau of Economic Analysis paper available at http://www.bea.doc.gov/bea/about/software.pdf; Parker, R., Grimm, B. &

is only part of a wider software activity) represents but a small part of the ICT economy overall. The two other main components, ICT hardware in the widest sense including digital media hardware, and telecommunications are much bigger. Actually, software represents only 3.3% of EU-15 GDP, *including* in-house “own-account” software development, while telecommunications services, strictly speaking, represent 2.6% of GDP, excluding media services. Hardware (excluding digital media equipment) represents 1.4%, providing ICT infrastructure with a 7.3% share of the European economy (Table 25). Overall, when one accounts for digital media equipment and distribution, retail services and ICT education, the total economy of the provision of means to the production, exchange and access to information represents around 10% of GDP in all developed countries[145], of which the software economy represents roughly one fifth (possibly a bit more if one considers that a very significant part of ICT education involves software).

Table 25: Valuing the EU15 ICT infrastructure economy, 2005.

Type of production activity	Approx turnover EU15	%GDP EU15
Software products	70.51	0.63%
Software / IT services	192.98	1.73%
In-house software	107.63	0.97%
Telecoms services	291.28	2.61%
Hardware	156.18	1.40%
Total	**818.58**	**7.34%**

Source: MERIT estimates based on OECD (GDP data); EITO ICT Markets 2006, FISTERA estimates for software market break-up.

How do the 10% of GDP compare with the sales of digital information and contents of all types, whether on carriers or on-line, whether unit or subscription-based (for the latter on needs to distinguish the provision of a telecommunication service with the provision of contents)? As seen in Table 26, total sales (turnover, not added value) for information (be it analogue or digital, on media or on-line) of all media and kinds is below 2.5% of GDP and does not go over 3.2% even when one adds physical distribution costs that are included for example in audiovisual content, information tools sold as information (packaged software) or intermediation services funded by advertising. It should be noted that the largest part of this figure is still constituted of physical carriers that means the value of the information sales themselves is probably grossly overestimated.

Wasshausen, D. 2002. “Information Processing Equipment and Software in the National Accounts”. National Bureau of Economic Research conference paper, http://www.nber.org/~confer/2002/criws02/wasshausen.pdf; Hermans, T., 2002. “Measurement of GFCF in software in the Belgian National Accounts”. Joint ECE/Eurostat/OECD Meeting on National Accounts paper. http://www.unece.org/stats/documents/ces/ac.68/2002/12.e.pdf

[145] Sopinspace estimates.

Table 26: Valuing the wider information economy (EU-15).

Media or type of information	Approx. turnover EU-15 (billion euros)	% GDP EU-15 (8531 billion euros for 2002)
Audiovisual (TV, video, film, music, radio) – figure includes some distribution services	100	1.17%
Books	27	0.32%
Magazines and journals	37	0.43%
Newspapers	47	0.55%
On line-information of all types not recorded above (max.)	8	0.09%
Other media (e.g. photographs and licenses)	2	0.02%
Proprietary software licenses	47	0.55%
Intermediators (Google, Yahoo, etc.) max. estimated from on-line advertising European market	1	0.01%
TOTAL	**269**	**3.15%**

Figures for 2002. Sources: Sopinspace, based on European Audiovisual Observatory, EITO, Publishing Watch, Forrester Research (for on-line advertising).

We are this faced with a situation in which the economy of providing means to an activity (creating, exchanging and accessing information) is about four times bigger than the economy of selling the by-products of this activity. For media in which self-production has recently exploded, such as digital photography, this ratio is at least 10:1. This clearly signals the fact that ICT (and software within it) is an infrastructure. In reality, it is an infrastructure for two different realms of activity. ICT is an infrastructure for an ecosystem of information exchange whose largest part escapes economic valuing because it does not use monetary transactions and is of an essentially non-commercial nature (for example, the photographs that are taken by digital cameras, which may result in indirect revenue for carriers or others but rarely for original producers, e.g. when the BBC relied on mobile-phone footage as the only source of images of the London Underground bombings of July 2005). The immense importance of this field of human activity can also be estimated directly, using time budgets, and indirectly, using Internet traffic statistics for instance[146].

ICT (and associated software) is also an infrastructure for the full economy of goods and services. This duality – the fact that ICT is an infrastructure for a non-commercial ecosystem of human activities as well as for the economy overall – is the main source of complexity for policy choices. In the next section, a number of questions are raised as a basis for further discussion. Their underlying common thread is that even from an economic impact of FLOSS perspective, it is worth paying full attention to the impact on the human information ecosystem, and that Europe's policy makers could miss many opportunities by underestimating the indirect returns of serving the information ecosystem.

[146] For an in-depth treatment of the relationship between the information ecosystem and the economy see chapter 6 of Philippe Aigrain, *Cause commune: l'information entre bien commun et propriété*, Fayard, 2005 (in French), or from a communication perspective, Andrew Odlysko, "Content is not king". First Monday, 6(2), February 2001.

8.2.1. Six questions on FLOSS' contribution to European information society

The questions raise issues on possible paths of impact of FLOSS on the wider universe that has been analysed in the previous section. Each is associated with a background description. This background should be interpreted in light of the tables on “Possible impact of FLOSS on selected markets and activities” and “Propagation paths between markets and activities”. These tables are reproduced after the discussion, in section 8.2.8, “Interaction between FLOSS and different markets” on page 135.

1. Innovation in new branches of activities and products vs. innovation in processes and productivity: Can FLOSS unblock the European paradox?
2. Can FLOSS provide a new European entrepreneurial model for innovators that are more adapted to the European culture?
3. Can FLOSS situate technology innovation in the perspective of a societal model that is endorsed by consumers and citizens?
4. Can FLOSS help the European industry to develop business strategies that address the wider universe of non-commercial creation and exchange of information? What are the conditions for deriving economic value from this universe?
5. Can a more neutral legal and regulatory environment in terms of business and technology models in the information and contents domain help the European ICT and related industries’ business strategies?
6. Can a better recognition of FLOSS in accounting rules, funding rules for R&D programmes, technology transfer and seed capital programmes help it deliver new innovation paths?

8.2.2. Unblocking the European paradox

No economy can afford to make a radical choice between innovation that aims at new products and possibly new branches of activities and innovation that aims at a better efficiency in existing processes or products and services. However, timing and priorities are crucial.

During the last fifteen years of the twentieth century, the European industry business strategies as well as the IST R&D strategy have been predominantly focussed on competitiveness within existing industry, processes, and products. Until 1995, this strategy was successful in a certain sense: it reached its target and productivity gains in Europe were higher by close to a percent by year to those in the U.S. However, these gains were obtained without major creation of new branches of industry or significantly new products[147], in particular in the field of general-purpose personal applications of ICT. Since 1995, the gains difference has been turned strongly in the opposite direction, with a difference of more than on percent per year in favour of the U.S. There are diverging explanations on the reasons for

[147] Mobile voice communication is often cited as a key counter-example. It is of course a true success and one that built on specific European assets for instance in the field of upfront standardisation. However it remains a niche activity strongly similar to a pre-existing activity. The defence of dominant positions in this domain has played a significant role in limiting the endorsement of open general-purpose IP-based applications in Europe.

this sudden change in performance, and the notion of productivity as measured in reference to the output corresponding to one hour of work[148] is also debated. However, there is an overall consensus that the difference in ability to invest in ICT and its usage has played a key role in this reversal.

Curiously, most explanations have stressed the lower level of investment in in-house use of ICT within European companies after 1995,[149] which is strange since such investment would have taken several years before delivering concrete results anyway. They fail to notice that this (relatively) low-level of investment came after a long period during which European industry invested heavily in productivity gains in traditional industries based on organisational techniques, replacement of work by machines and ICT-based cost-optimisation. During the same period in the U.S., after a period of wide upstream investment in generic ICT technology during the 1970s and 1980s, its deployment was encouraged by investment (by venture capital, personal investors and public policy during the first Clinton Presidency) in companies developing innovative functionality, even when the associated business models were uncertain.

One explanation for the productivity growth gap reversal must at least be considered is the hypothesis suggested by many economists of technical change[150]: investment in technology delivers growth and jobs in the long-run if it is sufficiently focussed on new branches of activity and new products. If it is too focussed on shorter-term productivity gains in existing branches, it of course encourages growth in the short term, but with a less favourable impact on jobs and a shorter time-span. By investing much more strongly in new industry and products, the U.S. have put themselves in a situation in which, when the incredible development of the Web took place in the mid-1990s, it could both cash on its past investment, and invest in productivity in existing branches in a context where the job destruction that is inevitably associated with such investment was bearable. Many other factors have of course played a role such as the job-intensive development of services in the U.S., the American real-estate boom, the strong availability of venture capital and other source of funding for deploying innovation in the U.S., the efforts in Europe to meet the criteria needed for the unique currency and the high cost of the reunification of Germany. Different kind of factors related to the entrepreneurial culture and the societal endorsement for technology are addressed in the next two sections

FLOSS plays a role in both innovation and competitiveness in processes and in setting the ground for new standards, functionality and societal usage (that later give rise to economic growth). However, the contribution in the later domain is by far more specific compared to proprietary software and other models that restrict follow-up innovation. FLOSS and proprietary software or patented algorithms (where they exist) are conducive to functional innovation and related new products. However they favour different types of functional innovation. Proprietary innovation mechanisms favour innovation in appropriable technology components. Leaving aside large installed proprietary software players whose innovation strategies are largely driven by their existing business models, proprietary

---

148 Some analysts use a measure per worker, which is also subject to caution.

149 See a typical example in *Is Europe Suffering from Productivity Paralysis?*, Business Week, 2 August 2004.

150 See for instance Marco Vivarelli, *The Economics of Technology and Employment: Theory and Empirical Evidence*, Elgar, 1995, or Chris Freeman, Luc Soete, *The economics of Industrial Innovation*, MIT Press, 1997.

innovation mechanisms are often used by start-ups and spin-offs. In the field of business applications, such components can have immediate demonstrable business models as the usefulness of the functionality they provide can be assessed within an existing frame of usage[151]. The success of these strategies in the field of general societal usage is much more doubtful: successful innovation in this domain depends on the existence of a great number of interdependent components, on a wide experimental deployment a low entry cost, and on the trust of advanced users in the independence of technology. As demonstrated by the success of the Internet, the Web, or more recent developments in the collaborative creation and sharing of information, this is much easier to achieve on a FLOSS basis and within a FLOSS-like culture.

8.2.3. Providing a new European entrepreneurial model

In his article “Why was Europe Left at the Station When America's Productivity Locomotive Departed”, Robert J. Gordon[152] has reviewed a wide range of explanations for the productivity gap observed since the second half of the 1990s. Most of the factors that he deems significant revolve around the innovation environment in the U.S. and the American social model (including components such as massive urban sprawl and the wide concentration of the distribution of goods). He particularly stresses the role of the U.S. culture of competitive innovation and entrepreneurship. Most commentators of his findings have stressed the need for Europe to imitate the U.S. This can be understood at various levels: if it is just a matter of stressing the need for a greater investment in ICT and its innovative adoption, there is little contention. However, if the message is that the means used to this effect have to mimic those used in the U.S., it might be a recipe for failure in Europe, at least outside the UK[153].

Young software innovators are not scarce in Europe. While many commentators lamented on the lack of risk-taking innovators and called for putting in place incentives to multiply their numbers, young software developers in Germany, France, the Netherlands, and the Scandinavian countries have put Europe at the forefront of global contribution to FLOSS, including for its most innovative components. Innovative projects such as KDE, VideoLan or a number of cryptography projects have originated and developed in Europe. It is also frequent for the most technology advanced U.S. company-led projects in the FLOSS domain to have their core developer teams in Europe (including Central and Eastern Europe). European developers also make a prominent contribution to FLOSS projects that are jointly supported by networks of small companies such as Zope/Plone. These European FLOSS

---

151 Contrarily to what is often assumed, ICT seed and venture capital investment in both the US and Europe has gone predominantly to firms developing business applications (in contrast to the situation for biotech, where most venture capital invests in firms engaged in research). The funding of bottom-up innovation for general usage has followed different paths such as personal investment (including by entrepreneurs turned rich in previous ventures, e.g. James Wales for Wikipedia) or buy-ins (a typical recent example is the acquisition of Writely by Google).

152 National Bureau of Economic Research Working Paper N° 16001, August 2004.

153 The UK has 2 traditions that ease its adoption of US-like strategies: a strong and demanding funding of universities and academic research, and a stronger culture of profit-minded entrepreneurship within the young innovators. It was for instance quicker to imitate the Bayh-Dole act than other European countries that often put in place similar incentives precisely when the US and the UK were starting to review its assessment in critical terms. For reviews of the effect of the Bayh-Dole act in the US, see work collated in the special issue of Technology Innovation and Intellectual Property, 2003(5), December 2003, http://www.researchoninnovation.org/WordPress/?m=200212.

software developers are neither risk-adverse nor business-adverse. Both in France and Germany, there is a high rate of creation of FLOSS SMEs, often with strong personal financial risk from their creators as they receive very limited public[154] or VC support.

FLOSS SMEs, NGOs and FLOSS activities in university research labs and by individuals are progressively creating a model of collaborative innovation where the players that collaborate to create usage and market compete for the related business and funding. Its greatest successes are measured by innovation in widespread usage. The exact scope in which this domain can produce results is not known, and it is clear that it will not replace firm-based proprietary innovation (in particular for material and complex systems innovation), large industry consortia, or academic research. It may be worthwhile for Europe to give this model due consideration because it is clearly endorsed by new generations of innovators, who are also strongly motivated by elements of societal usefulness addressed in the next section.

8.2.4. Ensuring technology innovation endorsed by citizens

The success in implementing technology for a society's benefit depends on the adequacy of a societal model that is endorsed by the population, technology that serves this societal model and enables it, and a policy environment that works for both. Today's Europe is characterised by a strong distrust of a significant share of citizens (among which many are young and potentially innovative individuals) for the underlying aims and motivations of technology deployment. This distrust or disaffection is reflected in the Europe-wide crisis of entry in scientific and technological studies, and in the fact that technology is often debated as “problem” rather than appropriated with enthusiasm. The policy answer has often been to assume that this was because of an insufficient understanding or knowledge of the reality of science and technology and thus focussed on a better communication. At European as well as national levels, one has now moved beyond this response, and the necessity of wide citizen debates on the orientation and possible risks of technology is gaining recognition.

There are complementary and promising roads to reconcile European citizens with technological innovation: acknowledge when they endorse it, and help them create it when they wish to do so. European youth has enthusiastically embraced ICT when it served the ability to express oneself (blogs), to share information with others (P2P file sharing, tagging-based image sharing, etc.), and to interact with others (SMS, chat, etc.)[155]. A segment of the youth has proved that it was able to invest significant efforts in the build-up of related technology or its innovative deployment. This is potentially a great asset, but it is presently a sleeping asset in the sense that there has been a very limited endorsement by either industry or policy of the underlying information society model[156]. The dominant models that were promoted during the 5th and 6th framework programmes were those of an inclusive information society based in ambient intelligence, where the latter was described as technology caring for the needs of consumers by delivering them contents and experiences,

---

154 With the exception of tax credits for R&D for young innovative firms, that are often difficult to use for the FLOSS SMEs due to their limited human and financial resources to conduct R&D.

155 The European youth has demonstrated a strong willingness to pay for the means to accomplish these activities. Surveys and studies such as those conducted in France for the Alliance Public-Artistes (http://www.lalliance.org) have shown that this willingness to pay extends to fees needed to ensure that some of these activities have returns for creators, when this can be done without intrusive or blocking technological control on usage.

156 For a detailed analysis of some properties of this model, see Philippe Aigrain, *Attention, Diversity and Symmetry in a Many-to-Many Information Society*, First Monday, 11(6), June 2006.

that is — if one excepts the word inclusive— the exact opposite of the appropriation of technology by people to create and exchange.

Europe is at a crossroads regarding how it promotes ICT. If it chooses the path of a many-to-many information society enabling a continuum of positions between receptors and professional producers, FLOSS can be a powerful tool to help it pursue this path. But how can European industry derive benefits from it? This is the subject of the next section.

8.2.5. Deriving economic value from non-commercial production

Despite successes at the infrastructure level, such as broadband deployment, mobile voice communication, or secondary benefits of digital photography in storage media and underlying electronic components, European industry has struggled to capture value from the ICT-based non-commercial information creation and exchange. One must first recognize that deriving business by capturing the positive externalities of non-commercial exchanges is intrinsically difficult. Providing the physical devices and telecommunication infrastructure needed for information exchanges has been successful, and we have seen above that this itself is a great source of economic value. However, one can't stop there, especially when one considers that the ICT innovation itself puts pressure on how much economic value can be derived from this information infrastructure perspective.

Already in 1973, Daniel Bell[157] has pointed that within a society where information plays a central role, and is in this sense[158] a society of abundance; there will remain a number of scarce resources and of activities that deliver them. Among them Bell lists the obvious “positions” (not everyone can be the most popular blogger) but also mediation functions: the animation of information exchanges, the filtering and editorial presentation of abundant information for users. Those businesses that have successfully captured positive externalities from non-commercial activities have done precisely that, from Slashdot to Flick'r, from Google to MySpace, from Ohmynews to Agoravox. They do it in many cases based on FLOSS software and FLOSS-enabled standards, and in all cases using FLOSS-like collaborative mechanisms between prosumers. However, there is an intrinsic difficulty in deriving economic value from these activities that arises from a classical aggregation problem and from transaction costs:

- The value delivered by a single provision of a mediation service to a given user is low, so one needs some aggregation mechanism so value can be extracted at a compound level.
- Transaction costs are strongly and rightly rejected in the ICT-enabled information exchange sphere, because the associated controls, needed user choices or monetary transactions defeat the very benefits of ICT[159].

---

157 *The Coming of Post-Industrial Society: A Venture in Social Forecasting*, Harper-Colophon Books, 1973.

158 Even in European societies, abundance is not the rule in all domains, and the environment, climate and energy crisis give strong incentives to the careful management of scarce and depletable resources.

159 See Yochai Benkler, Coase's Penguin, or Linux and the Nature of the Firm, Yale Law Journal, 112, 2002 and Clay Shirky, The Case Against Micro-Payments, OpenP2P, December 2000, http://www.openp2p.com/pub/a/p2p/2000/12/19/micropayments.html

The predominant response to these challenges has been to use advertising business models, but there is evidence that this model is constrained macro-economically (between 1% and 2% of GDP) and risk polluting or biasing the very ecosystem of information exchanges that the corresponding services intend to serve.

Recent papers[160] from active FLOSS authors have reviewed the various models that could be use to solved the challenge of deriving economic value for services mediating non-commercial information exchanges. They include consideration for value-added services assisting users to move across the continuum towards more professional roles, subscriptions to community memberships and non-conventional mutualisation schemes (legal fees-based, donations-based). The jury is still out on which models can work for which services and at which scale, but FLOSS and its applications are a major ground for experimentation. The creation of partnerships between FLOSS, information commons and collaborative media players on one side, and larger ICT industry on the other side, could play a key role in this respect. These partnerships can be successful only if the larger industry players acknowledge the constitutive properties of the open information communities and mobilise their innovation abilities to serve these properties.

8.2.6. Building a more neutral legal and regulatory environment

In the recent years, the ecosystem of non-commercial information creation and exchange has been subjected to contradictory trends. FLOSS is a major enabler for the corresponding activities and has received growing recognition in policy as well as in regulatory debates: support to FLOSS innovation in European and Member State R&D programmes, deployment in administrations, consideration of the impact of regulatory measures on FLOSS in the software patent ability debates as well as in the 2001/29/CE transposition debates, recognition of FLOSS as a relevant impacted domain for competition policy measures. In parallel, the more general domain of information commons has matured at a remarkable speed. The Creative Commons licenses and related movements have helped FLOSS derive explicit contractual foundations for the voluntary sharing of information that was often done on implicit terms before. Open Access scientific publishing is becoming the reference paradigm for the sharing and dissemination of scientific knowledge. Collaborative media, which were limited to specialised fields or political communities are now a mainstream phenomenon that receives attention from major media moguls. The Web-based sharing of photographs is an unprecedented explosion of diversity and quickly raises the quality level of prosumer contents. Similar trends are at a more preliminary stage in media such as music and video.

Meanwhile, the unauthorised non-commercial exchange of copyrighted works between individuals that has become a systematic practice among European youth and beyond, was a key driver for the deployment of broadband, but is clearly not sustainable in terms of the lack of associated remuneration and funding of creators and publishers[161].

---

160 Brian Behlendorf, How Sustainable Business Forms around Open Software and Lessons for Other Media", FM10 Conference, to appear in First Monday, and Philippe Aigrain, Attention, Diversity and Symmetry in a Many-to-Many Information Society, First Monday, 11(6), June 2006.

161 There is debate on whether the non-commercial exchange of information representations of copyrighted works results or not in losses in sales of carriers. However, in the long run, on-line digital content is bound to be the major scheme for distribution and access, which means that it must be coupled in a less indirect manner with revenue streams for creators and producers.

In the same period, the input to the evolution of the regulatory environment has been dominated by the requirements of the centralised commercial provision of information and contents as represented by the business strategies of media groups and of the providers of technology that implement access and usage control. European ICT industry players who have no direct interests in digital rights management (DRM) and similar technology have adopted, in most debates, a position of damage limitation, limiting liability risks and protecting the core operation of the technology from side effects of regulatory moves. For reasons of short-term pricing issues, most ICT players have taken a stand hostile to fees or levy-based remuneration schemes for producers and creators of contents that is exchanged non-commercially on-line between users or reproduced on carriers. In practice, the levies, fees or legal licensing schemes contribute an ever growing share of the income of creators and performers, no doubt because they have good properties in terms of the aggregation and transaction costs constraints that are so important in the information sphere. Despite this, the regulatory initiatives are often focussed on plans to phase them out in the name of their replacement by DRM. The collective societies (in particular for music), that are in some countries legal state-granted monopolies and in others de-facto oligopolies are refusing to authorise their members to give a creative commons status to their works for non-commercial exchanges, forcing them in practice to lose all the benefit of collective management and centralised redistribution if they want to explore the creative commons route.

The results of this situation are that the regulatory environment and part of policy are less and less neutral towards the exploration of business and funding models for on-line contents, and also lose neutrality in terms of the technology infrastructure. Some of the recently adopted legislation (for instance the French transposition of the 2001/29/CE directive) has a strong negative impact for FLOSS, due to the introduction of possible compulsory DRM for some types of software and to restrictions on the disclosure of source code for software that achieves interoperability with technical protection measures. The Internet neutrality debates in the U.S. are demonstrating that it is the basic fabric of the information infrastructure that is at stake at a more general level.

Without entering in a debate on precise legislation that is outside the scope of this study, an open “no-blinkers” discussion on the requirements for setting a neutral ground in terms of technology infrastructure, business models, and remuneration or funding schemes for creation could open new development paths for the European ICT industry.

8.2.7. New innovation through better accounting and recognition for FLOSS

Creation software that is freely usable, distributable and modifiable is an activity with specific constraints: it produces a common good that can be used by all, partners or competitors alike, in one's country or globally. In this sense, it is similar to the creation of open knowledge by pre-competitive basic research, though it can be very close to market activities. The very novelty of such an activity explains why it is difficult to find the proper incentives to encourage it when it is deemed useful.

Despite an OECD recommendation (1993), most EU countries do not account in-house software development or externally procured development of software as an investment. This creates difficulties for deriving a good knowledge of the software economy. Beyond that, it is a problem for FLOSS development by companies. In some cases, FLOSS companies or companies that develop some FLOSS may account for the corresponding activities under R&D and benefit from the R&D incentives. Schemes have been proposed that would generalise R&D-type tax credits to FLOSS software development, even when it does not meet the criteria of R&D. The practical implementation of these schemes requires one to sort out a number of issues, such as the ex-post assessment of the usefulness of the software. Other schemes such as “prize funds” that have been proposed in the area of pharmaceutical drugs[162] create similar incentives but raise different issues since they require a-priori identification of areas in which developments are targeted.

European and Member State research programmes have increasingly set a 50% shared cost funding[163] as their core rule. This choice is based on making sure, without too complex an individual assessment, that the limits set in the WTO protocol on state aids to R&D are respected. More generally, the indiscriminate application of competition oriented rules in the case of the production of a common good that is available for all competitors requires some review. In practice, the recipient of funding for R&D whose results are disseminated as FLOSS acquires some competitive advantage along the lines of the classical FLOSS-based business models such as “best-knowledge here” or because of having oriented the functional choices of the software. This justifies a participation in the costs, but it is doubtful the 50% rule is followed. In practice, most R&D actions that have successfully supported FLOSS projects (in particular when company participation was necessary) have used niche experimental actions to achieve a higher rate of funding. An explicit adaptation (for instance to the “rules for participation” for EU R&D) could provide the appropriate incentives for a higher dissemination of framework programmes results as FLOSS.

Beyond the initial dissemination stage, akin to the early high-failure stage common to start-up businesses and commercial software projects, FLOSS projects have a high rate of sustainability with the formation of a self-organised governance structure.[164]. FLOSS SMEs have survival rates that are as good as other SMEs. However, despite notable exceptions[165], technology transfer policy, publicly funded seed capital programmes, large company spin-off

---

162 See for instance: Creating Alternative Incentives for Pharmaceutical Innovation, Clinical Therapeutics, 18(1), 2006.

163 35% for demonstration actions.

164 See J. M. Gonzalez-Barahona and G. Robles. Unmounting the "code gods" assumption. In Proceedings of the Fourth International Conference on eXtreme Programming and Agile Processes in Software Engineering, 2003, available at http://libresoft.urjc.es/downloads/xp2003-barahona-robles.pdf#search=%22Unmounting%20the%20%E2%80%9Ccode%20gods%E2%80%9D%20assumption%22 ; also Michael K. Bergman, Open Source and the 'Business Ecosystem', AI3, 31 August 2005, http://www.mkbergman.com/?p=119;

165 Cf. the ObjectWeb consortium, for instance.

programmes are still ill at ease with FLOSS-centred business plans. This may be a case for a better communication of the related opportunities. The growing space devoted to FLOSS-based strategies in the communication of the IPR help-desk of DG RTD is an indication of change in this respect.

8.2.8. Interaction between FLOSS and different markets

The possible interaction between FLOSS and different markets, and the possible propagation paths of influence from one segment to another are illustrated in Table 27 and Table 28.

Table 27: Possible impact of FLOSS on selected markets.

Impacted market or activity (below) and FLOSS effect (right)	**Negative impact factors of FLOSS on the impacted market**	**Positive impact factors of FLOSS on the impacted market**	***Possible*** **outcome taking in account propagation paths between markets (to be checked at various time frames)**
Proprietary packaged software	Cannibalisation of market by competing FLOSS products, pressure on IP-based monopoly prices	FLOSS contribution to the take-up of new markets and branches of activity (ex. Web communication, digital photography) leads to a demand for higher-end proprietary software	Overall slightly negative
Software services including systems integration and solution providers	Prices forced down when software licenses are used to generate a captive service market and more generally because consumers are less dependent on vendors. Possible negative effects of greater fragmentation of products offered in the FLOSS world, though positive network effects also work here. FLOSS favours some re-internalising of software development (not truly a negative impact, rather a shift to in-house production)	As de facto standards with FLOSS implementations do not favour particular vendors, they promote interoperability with new services supported by new vendors and promote the development of new application domains. Relative competitive advantage for FLOSS-based service provision players	Moderately positive. Due to budget constraints on IT services procurement, value for money gains are absorbed to deliver better functionality of quality in the activity requiring the service. In the longer run, possibly stronger positive impact.
Information infrastructure services (IT and media equipment, telecom)	Resistance by proprietary software players to greater FLOSS use may lead to some temporary disruptions, as vendors try to retain market share by refusing to disclose interoperability information or prevent multi-platform use by consumers.	Market enlargement through a combination of lower prices and new usage domains (see below monetary transaction-free information exchange). Greater reliability and related cost gains.	Strongly positive (validated e.g. through take-up of FLOSS by telecoms and secondary software industry, network device manufacturers, embedded software etc)
Information economy content services	As FLOSS use is correlated with information sharing and diversity, it may have a negative impact on those publishers of content who drive prices up by controlling their market and artificially limiting supply.	In the longer run, possible development of business to serve demands that forms in free exchanges (for instance a-posteriori publishing of content whose value is increased through usage, e.g. Wikipedia).	Moderately negative at short-term, closer to neutral in the longer term

Table 27 cont'd: Possible impact of FLOSS on selected markets

Impacted market or activity *(below) and FLOSS effect (right)*	**Negative impact factors of FLOSS on the impacted market**	**Positive impact factors of FLOSS on the impacted market**	***Possible* outcome taking in account propagation paths between markets (Table 28, to be checked at various time frames)**
Information exchange without monetary transactions (interpersonal communication, commons-based creation and exchange of information)		Strong synergy between FLOSS development and usage and growth of information commons and inter-personal communication activity. FLOSS contribution to open standards and education also favourable.	Strongly positive (not directly an economic impact, but has effects on demand for information infrastructure and software services)
Non-IT economy / secondary users of IT (non IT-manufacturing, non-information services such as tourism, personal care, administrative, health, etc.)	Some initial disruptions due to inertia of business models that try to mimic information monopoly effects in other sectors, as well as initial (but one-time) migration costs involved in moving to FLOSS	Independence from suppliers, competence in the workforce, return of initiative, choice and control to user organisations, more investment ability thanks to visibility of future of technology and availability of trustable open standards not controlled by dominant vendors.	For administrative, health and education services, the productivity gains are likely to be absorbed to deliver better quality to users. For manufacturing and market services, strongly positive

Table 28: Propagation paths between markets and activities

How growth in some markets (below) may affect growth in others (right):	**Proprietary packaged software**	**Software services, systems integration, solution provision**	**Information infrastructure services (IT and media equipment, telecom)**	**Information and media content markets**	**Information commons-based non-economic exchange**	**Non IT-economy (manufacturing and services)**
Proprietary packaged software		Eat up some of the services market, limits investment because of vendor-dependency and uncertainty of future migration path. Facilitates captive market pricing for software services	Complex: adverse to open standards (thus negative in case of oligopoly), but may have positive effects when monopoly imposes de facto standard.	Favours greater concentration through supporting proprietary technology mechanisms. Negative market impact may be hidden by monopoly pricing (e.g. in music)	Negative impact, mostly because of contradiction between business models and related cultural inability to serve information commons	Limits investment because of lack of control over future investment due to dependence on proprietary suppliers.
Software services, systems integration, solution provision	No clear, direct impact		Enabler	No clear, direct impact	No clear, direct impact	*Proprietary* software-based solution provision has a detrimental effect on ability to invest in a middle term, due to vendor dependency.
Information infrastructure services (IT and media equipment, telecom)	No clear, direct impact; indirectly generates a demand for both FLOSS and proprietary software	Telecoms generate demand for software services and integration		Potential positive impact through market enabling.	Strong coupling (synergy)	Solow's paradox: productivity gains are slow to materialise and often hidden by enlargement of perimeter of activity.

Table 28 cont'd: Propagation paths between markets and activities

How growth in some markets (below) may affect growth in others (right):	**Proprietary packaged software**	**Software services, systems integration, solution provision**	**Information infrastructure services (IT and media equipment, telecom)**	**Information and media content markets**	**Information commons-based non-economic exchange**	**Non IT-economy (manufacturing and services)**
Information and media content markets	Concentrated media have synergy with proprietary software due to common reliance on information property monopolies	Generates specialised demand	Positive impact on some specialised equipment (DVD for instance)		Structural incompatibility of centralised media models with commons (competition for time budgets and attention).	No clear, direct impact
Information commons-based non-economic exchange	For communication complex: free-of-charge software is the rule, uncertainty on whether value can be sustainability captured through advertising or later pricing For collaborative work: probable negative impact on proprietary software market. For solitary work: unclear	Generates demand for services (example digital photography), however difficult to capture in concrete terms.	General new demands for infrastructure and an enlarged market for equipment (e.g. growth of world wide web, blogs, wikis…)	Information commons gives rise to new forms of media publishing and distribution, whose market potential is still uncertain		Effect interesting to explore, in particular role of the creation of more conscious customers.
Non IT-economy (manufacturing and services)	No clear, direct impact (generates a demand for both FLOSS and proprietary software)	Rapid positive effect on demand for software services	Generates demand for equipment and telecom	No clear, direct impact	No clear, direct impact	

8.3. FLOSS, R&D substitution and the impact of collaborative strategies

Lack of competition, in particular, provides market incumbents with higher scales of income that elevate the R&D expenditure of the dominant firms (usually, a fixed share of the revenues). Yet, various studies suggest that a more competitive structure enables entry of new firms to the market and increases product differentiation and results in higher scales of firm's learning and technological quality in the long term (Merges and Nelson, 1990; Harison, 2003).

The production and use of software in the FLOSS mode may be a complement or it may compete with the proprietary software model in which software is produced for sale. In its role as a complement, FLOSS development methods may extend the capacities of proprietary software and its production may enlarge the community of software developers. As a competitor, FLOSS methods may offer software that better meets user needs and/or that diverts developers from the production of proprietary solutions to the enlargement of a publicly accessible knowledge base of software solutions.

On the supply side, adoption of the FLOSS mode of development substitutes for internal R&D and enables firms to allocate R&D for advanced projects, rather than "to re-invent the wheel". The basic FLOSS principles of code reuse reinforce the optimal but rarely achieved goal of innovation reuse. It can be empirically demonstrated that code reuse actually occurs to a high degree in FLOSS projects. Table 29 shows code reuse in the Debian collection of FLOSS applications[166]. Another question is to what extent can firms reuse FLOSS software to substitute for their own internal R&D? Firms already substitute for internal R&D by buying or licensing products and components from others – but those costs and reuse of FLOSS can prove much more cost effective.

Table 29: Code reuse across FLOSS projects

Code base collection	Debian 3.1 FLOSS distribution (2006)
A. Source lines of code (raw count)	247 809 088
B. Source lines of code (adjusting for reuse)	157 434 545
Code reuse ratio (A / B)	1.57
Reuse share of total code ((A – B) / A)	36%

Source: MERIT estimates based on URJC data. Note: Debian 3.1 in 2006 is larger than in 2005 (Table 2).

Our study presents results of examining aspects of substitution of internal R&D and cost savings within firms associated with FLOSS implementation, estimated through existing "census-type" approaches – such as the substitution cost of development for specific FLOSS applications as described above. Such methods can identify the cost to firms of developing in-house software with the functionality of, for example, the Linux kernel. Using the Linux kernel instead of developing software in-house allows the firm to build upon the existing software base, save large monetary and time resources, and direct R&D spending to new innovations. (Using proprietary software is unlikely to save such costs since only the proprietor of the software can make modifications other firms cannot modify and build upon proprietary software components. Embedded systems are the exception, but firms must buy

[166] This table shows code reused by actual copying at the file level, i.e. including of files of code from one module in another module, with possibly changed filenames and minor modifications to the reused code. It excludes code that is reused at e.g. the individual function level, or substantially modified code, as well as code reuse through library linking rather than copying.

expensive and often unavailable modification and source rights in order to adapt proprietary software to their needs).

There are a number of examples that can be cited to illustrate. For R&D efficiency in software research, there is a significant reduction of time-to-market because of large-scale code reuse in the FLOSS development model. The time to market for a new compiler (either for a new architecture or for a new language) can be dramatically lowered. This occurred with the implementation by New York University (NYU) of a proof of concept compiling system for Ada95 (a programming language designed by the U.S. Department of Defense) with a grant from the U.S. Department of Defense Advanced Research Projects Agency. With 3 million U.S.dollars (USD), including indirect costs, NYU developed the Ada implementation by reusing code from the FLOSS compiler for the C++ language – GCC. NYU's resulting compiler, called GNAT, involved the spending of a very small fraction of what Ada compilers had cost others to develop – typically in the order of 20 million USD or more, and obviously a correspondingly increased time to develop and time to market. Indeed, a URJC study (unpublished, prepared for this study) estimates the total substitution cost of the GNAT compiler today – i.e. the cost to the developers if they had written it from scratch instead of reusing much of the FLOSS application GCC and continued development in the FLOSS mode – at 4 764 person-months, or about Euro 45 million.

It is important to note that with proprietary software, such improvements and R&D substitution effects are only possible *within* a company, since code reuse is only possible internally, while the example above shows that innovators working in the FLOSS model can reuse code from external sources, even combining several external sources with no transaction or search costs related to identifying owners of the past R&D results as they are all licensed to reuse them under FLOSS conditions.

We further illustrate the extent of R&D substitution possible with a study of a European industrial test case, that of Nokia and its Maemo platform.

8.3.1. Nokia and Maemo: a FLOSS R&D substitution case study

The Maemo Development Platform[167] is the application development platform for the Nokia 770 Internet Tablet[168]. As the Maemo web site acknowledges, "the Maemo development platform [...] is contributed and operated by Nokia". The software is available to third parties under the FLOSS licenses corresponding to each package, and Nokia encourages others to use and improve it, expecting that it "will result in a feature rich, mature and well-supported software base that could evolve as a de-facto standard for Linux handheld via the open source process". However, they also state "for the time being, Nokia controls the development of Maemo to keep it aligned with its product development. This may mean in practice some limitations in accepting contributions to the package repository. However, we are open to suggestions to improve opportunities for involvement". (Note that due to the licences, this "control" is notional – it does not prevent a third party from "controlling" its own version of the software with modifications not accepted by Nokia.) Maemo is composed of FLOSS software, including well-known subsystems such as the Linux kernel, the GCC compiler, the GNOME user interface, GNU utilities, etc. As a whole, it is a complete

[167] http://maemo.org

[168]http://nokia.com/770

developing environment with capabilities for cross-platform development, including applications offering feature-rich system, and user friendly interfaces for users of tablet computers.

Maemo is in fact a Debian-like software distribution of Linux and associated software. As such, its source code is organized in packages, each one usually corresponding to an "upstream" application (upstream meaning that it is maintained by its original authors). The main work of the Maemo developers is to integrate all the components taking into account the specific requirements of the target hardware systems. In short, they start with a reduced set of Debian packages (or their equivalents, sourced directly from the upstream project repositories), and adapt them to ensure that they run smoothly in the hardware for which it is intended, the Nokia 770 Internet Tablet.

Maemo is a product promoted by a single company (Nokia) to use in one of their devices (the 770 Internet Tablet). It is a good example of how a company can use a collection of FLOSS software for its own needs. It shows how a company benefits from the efforts of third parties (e.g. other parties contributed to create the software), and how a company can save in R&D costs (or in licensing, if the software was not purchased from a vendor).

8.3.1.1. Methodology

For this study, we have considered the "Mistral" release of Maemo, as available from the Maemo repositories in August 2006. Maemo packages are split in two collections, named "free"[169] and "non-free"[170]. For the purposes of this study, both have been considered together as a single collection. Its main characteristics are:

- 260 different source packages with FLOSS software (under different licenses)
- The top packages (by number of lines of source code) are the following (each of them with well above 1 million lines of code):
    - kernel-source: the Linux kernel, version 2.6.16
    - gcc: the GCC compiler system, version 3.4
    - glibc: the GNU standard C library, version 2.3
    - gdb: the GNU debugger, version 6.4
    - gtk+: the Gtk+ library, a part of GNOME, version 2.0
    - binutils: the GNU binary utilities, version 2.16
    - xserver: the X11 graphics server, version 6.6

Maemo is used to build the Internet Tablet 2006 OS which is the software actually running in the 770 Internet Tablet. The source code is available both from a Debian style repository (as a set of source packages) and from a Subversion repository[171]. For this study, we have analysed the code in the former repository.

[169]http://repository.maemo.org/dists/mistral/free/source/Sources

[170]http://repository.maemo.org/dists/mistral/non-free/source/Sources

[171] Subversion is another version control system. See details at http://maemo.org/downloads/download-source.html

Source packages have been downloaded and stored locally. They have been uncompressed and unpacked into directories with the source code. This source code has later been analysed with URJC's pyTernity system. This system uses several heuristics to analyze portions of the source code (such as copyright attributions) to estimate the ownership of each file in the source code.

The basic assumption behind this methodology is that owners of the code, and especially companies owning the code, generally include appropriate copyright attributions in the source code. We have found, by analyzing many different libre software codes, that this assumption is usually true.

8.3.1.2. Main conclusions

The main conclusion does not come as a surprise: Nokia did not develop most of the code in Maemo. In fact, only about 200 000 lines of source code of a total of more than 15 000 000 present in Maemo appear to have been attributed to Nokia (i.e. less than 1.5%). This was not surprising given Maemo is based on upstream FLOSS software packages usually produced by third parties (including the community of individual developers).

The companies with significant amounts of code claimed in Maemo are:

- Red Hat Corporation: more than 415 000 lines of code, mainly found in the Linux kernel and in GNOME-related packages.
- Silicon Graphics Corporation: more than 275 000 lines of code, found in the Linux kernel, X11-related, and other packages.
- IBM Corporation: more than 220 000 lines of code, in the Linux kernel and related packages, in X11, etc,.
- Nokia Corporation: more than 200 000 lines of code, found in the Linux kernel and related packages, in graphical user interface packages, in some Maemo-specific packages, and others.
- Intel Corporation: more than 160 000 lines, in the C library, Linux-related packages, and others.
- Sun Microsystems: more than 130 000 lines.
- Digital Equipment Corporation: almost 130 000 lines.
- Hewlett Packard Corporation: about 115 000 lines.

Contributing between 30 000 and 100 000 lines of code, many other companies are found: Montavista Software Corporation (a provider of Linux based solutions for mobile phones), QLogic (networking storage solutions), Adaptec, Axis Communications, TrollTech AS, VA Software, Ada Core Technologies, Cisco, and Broadcom. A long list of companies follows, including for instance Fujitsu, Ximian, Xerox, Motorola, MIPS, SuSe, Qualcomm, Ericsson, Google, etc.

Of course, not all the code produced by these companies, let alone all the code produced by individual developers and other authors, is fundamental to Maemo. Some parts of it (corresponding, for instance, to hardware devices or functionality to which Maemo is not targeted) will never be run in the Maemo context. But the list certainly provides a good view of the landscape of the contributors that have made Maemo possible, several of whom have contributed individually far more than Nokia, and all of whom have jointly contributed some orders of magnitude more than Nokia.

In addition to companies, other entities have contributed to Maemo with large shares of code. Among them, some can be highlighted:

- The Free Software Foundation is the largest single owner of code found, with a total of more than 2 785 000 lines of code.
- The Open Group has contributed with about 200 000 lines of code.
- The OpenSSL project has been found owner for about 75 000 lines.
- The Purdue Research Foundation has been found in about 55 000 lines.
- XFree was found in files amounting a total of about 22 000 lines.
- The Python Foundation claims ownership for about 15 000 lines.

If it were to be created internally within a single company, the total Maemo code base would cost a little under 12 000 person years to develop, or about Euro 870 million (using the COCOMO cost estimation model). This is clearly an overestimate of the actual value of the R&D substitution, since much of the Maemo code base may never be used – as it does not cost anything to include it, there is no reason to include only what is absolutely essential. Nevertheless, it is reasonable to assume that the R&D substitution value of the FLOSS code in Maemo is in the range of the above estimate.

Of course, Nokia need not have spent that money to develop the technology internally; it could have been externally sourced in exchange for a licence fee. That would limit Nokia's ability to adapt the software to its own needs and to maintain control, which is an important consideration for them.

Finally, it must be pointed out that the Internet Tablet 770 was created as an experimental product. Neither a smart phone nor a PDA nor a laptop, the tablet was an attempt at exploring the market for a versatile wireless device (it has full wireless Internet and Bluetooth functionality) with a very clear display and powerful features, but without directly threatening the PDA market[172]. The hardware was not really optimised and was based on off-the-shelf components. Nokia was reportedly surprised at the very high popularity of the device. It quickly sold out in the U.S. and Europe.

It is almost certain that such an experimental innovative product would not have been attempted if Nokia had to pay the full – high – R&D cost, given the small expected immediate returns. It is certain that such an experimental product could not have been made using off-the-shelf software licensed from others. So this example demonstrates the potential of FLOSS as an R&D substitution mechanism – Nokia could rely on R&D diffused across several other firms and individuals and made available through FLOSS. Nokia could focus its research on making the product more innovative, and could take a greater risk of market failure at a lower cost to itself. With the success of the 770, Nokia now has the opportunity to multiply the return on its relatively low R&D investment on the Maemo operating system.

[172] E.g. it was released with a web browser but no calendaring or communications software of the sort suitable for PDAs.

8.3.2. Collaborative R&D and efficiency

Clearly, it is not just Nokia who benefits if Maemo and the Internet Tablet 770 turn out to be successful. Nokia's own contributions to Maemo have directly benefited other FLOSS users as they have entered the main distributions of Debian. This is particularly true in areas that Nokia was best positioned – due to expertise as well as incentive – to work on such as improving Linux Bluetooth functionality. So neither the community of individual volunteer developers nor the other firms contributing code being used by Nokia for Maemo need feel cheated.

Moreover, if Nokia's experiment in creating a new type of user device turns out to be successful, as initial indicators suggest, then other firms benefit too. Software firms can build code and services for the new market, but even other hardware firms could build products to compete with the 770, and do so using the Maemo platform. If this happens, Nokia will not be able to ensure its market dominance of this new non-PDA device market through IP ownership, since Maemo is FLOSS software, and it will have to continue to innovate and experiment. However, as we have seen above, if Nokia did not build this experiment upon FLOSS, it would not have any chance to exploit this new market, as it would not have been worthwhile risking the enormous investment in software R&D to create it.

This form of pre-competitive collaboration among firms – and in FLOSS, extended to a broader community of individuals, foundations and others – is not unique and is growing in popularity due to FLOSS-inspired collaboration models. The idea can be summarised as follows: share the costs of risky early R&D by building upon a common knowledge base and giving up exclusive ownership rights − in return have a good chance of profiting from a new market or new product category that would not have existed otherwise. This is a strong driver for greater innovation.

Another motivator for Nokia to use FLOSS as a basis for Maemo, in addition to R&D substitution for the innovation in the platform development itself, was to encourage *further development* including user applications by a wider developer community. Building Maemo from Debian immediately plugged the Nokia 770 into the large FLOSS developer community around Debian. This illustrates another benefit of FLOSS for R&D, as supported by a study of innovation on handheld computer systems hardware[173]. Although the study found positive effects of closed development during the early phases of development in the case of *hardware design*, subsequently, increased openness (and devolved control) yielded higher rates of innovation in hardware feature introduction. Devolved control in the multi-supplier case was viewed as stimulating hardware design innovation by encouraging more R&D investment from device manufacturers (other than the originator) when the chances were reduced that a controlling supplier could hold-up or lock-in manufacturers. The positive effects of control in the earliest stage of development were attributed entirely to coordination of the initial design process[174] (rather than, for instance, increased incentives for innovation due to control of IPR). Subsequently, however, opening the platform generated faster releases and higher hardware innovation rates.

---

[173] Bourdeau, Kevin. 2006. "Does Opening a Platform Generate More Innovation? An Empirical Study". *MIT Sloan Working Paper* 4611-06, June 2006: available at: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=913402

[174] pp 22-24.

This evidence from quite a different field of hardware supports the view that once initial design has been developed (whether in a closed environment or using FLOSS for R&D substitution as with the Nokia 770 example), opening access to further development of that platform encourages investment from third parties and stimulates a continuing trajectory of design innovation.

One concern for originating firms may be that collaborating firms could 'free ride' on their innovations, rather than contribute value that is useful to the originating firm as well. Such issues have been addressed in industrial consortia through complex legal arrangements, and FLOSS licences provide a simpler alternative: reciprocal licensing, which ensures that derived works must be distributed under the same terms as the original work. Thus, when the originating work is distributed with the permissions to view the source code, modify it and redistribute at no charge, any further development by a collaborator is likely to be made available (and thus benefit) the originator. For pre-competitive research, as well as for platform development, the benefits of having new software may be greater than the loss of others freely using the software. These benefits may *not* be greater than the (quite distinct) loss of competitors exclusively appropriating improvements to the software, but reciprocal licensing prevents this, which is why most firms that produce FLOSS software prefer it. Indeed, such models have also been used for collaborative research in areas other than software, such as the HapMap consortium in bioinformatics[175].

Section 7.5.1, "Collaboration and IPR sharing", explores this further together with an elaboration in the context of SMEs who are particularly disadvantaged for more traditional forms of collaborative innovation (consortia, cross-licensing) as they are harder pressed to afford legal skills and expenses for the creation and maintenance of complex IPR licensing agreements.

Thus FLOSS has the potential to significantly change the software development process. Indeed, while there are many divergent views on the more political aspects of FLOSS, its qualities as purely a system of development are almost universally appreciated and several firms are trying to adopt its development model even for internally developed proprietary software (e.g. Philips' plans for "InnerSource", which involves the sharing of code among 1800 or so people within Philips Medical Systems).

As mentioned previously, FLOSS has been the origin of a number of innovative developments at the very basis of Internet technologies. In particular, the Internet protocol, the domain name system, Internet mail and the World Wide Web were, although originated through public funds in academic environments – sustained through the FLOSS developer community (then known simply as the free software community). Scripting languages that allow web sites to be interactive and powerful, such as Perl and Python, are entirely FLOSS in their origin – much before, and still a bigger market than industry-developed languages such as Java – and in their continued development. Not only innovative in themselves, these technologies have been the basis for much development in other fields of science (e.g. the Human Genome Project was made practically possible through the usage of BioPerl, a version of Perl that, thanks to its open source nature, could be modified to suit the needs of genetic studies).

175 International HapMap Consortium. (2003) "The International HapMap Project". *Nature*. vol. 426 no. 6968, pp. 789-96

The use of FLOSS in and the application of the FLOSS methodology to particular new technologies may have positive attributes from social, legal and economic standpoints. From a social welfare perspective, ubiquitous computing and ambient intelligence are examples of core technologies that establish novel technological trajectory and open new venues for innovation; therefore ensuring public access to their technical prospects is essential for technological diffusion and for their advance. The ability to restrict knowledge in early stages of the technology makes it difficult to access parts of knowledge that may be building blocks for future research and development. Further, many of the results of research are not merely discoveries of facts but creation of research tools and basic algorithms. To keep proprietary vital inventions, algorithms and standards may reduce the possibility of others to pursue research in the fields that need them and further the technology and its applications. (Historical evidence that patents hampered the development of wireless technologies and aviation in their early stages is shown in Nelson, 1994; much older historical evidence such as how patenting strategies hampered the development of steam engines is shown in Nuvolari 2005).

As those tools and scientific discoveries can be very basic and generic, appropriating them in early stages may pose threats to scientific progress, while collaboratively sharing access to knowledge in basic areas may improve scientific progress. Industries already recognise this. The SNP consortium was formed to publicly share knowledge on single nucleotide polymorphisms and includes major players in the pharmaceuticals, ICT and bioinformatics sectors such as AstraZeneca, Aventis, Bayer, Bristol-Myers Squibb, Hoffmann-La Roche, Glaxo Wellcome, IBM, Motorola, Novartis, Pfizer, Searle, and SmithKline Beecham[176]. A recently published book by MIT Press ("Collaborative Ownership and the Digital Economy," ed. R Ghosh, MIT Press 2005) discusses in much detail various cases of collaborative development in industry, from software to pharmaceuticals to physical and engineering sciences, showing the relationship to the collaborative development model of open source from which many examples in other disciplines (e.g. the Human Genome Project) have explicitly drawn lessons from FLOSS.

8.3.3. Cost savings and efficiency gains from collaborative R&D

It is not possible to quantify the extent of R&D substitution effect that takes place. Every application of software-based innovation has different cost structures based on the level of competition in the market for the specific application, the level and direction of technical innovation targeted, the market in supply of alternative commercially licensable technologies (substituting for internal R&D) and the financial cost of risk-taking innovation – as well as the level of innovative development offered by pre-existing FLOSS tools for the application. Some examples illustrate the variation:

- Sitecom, like the majority of manufacturers of DSL modems and WIFI access points and routers today no longer build operating systems to provide the user interface and network management aspects of their hardware devices. They use Linux and related FLOSS tools instead. In this case, almost the entire software R&D is eliminated, or substituted for a focus on improved hardware functionality.

[176] http://www.ornl.gov/sci/techresources/Human_Genome/faq/snps.shtml#whoare

- IBM's Workplace system is derived from the popular FLOSS office suite OpenOffice.org. OpenOffice.org is a very large application, which IBM has modified to be used as a service over the Internet rather than as an application on a standalone PC. The entire essential application functionality was taken from OpenOffice, with IBM providing the modifications needed to make it accessible as an online service.

- Daimler-Chrysler has used real-time embedded Linux in a portable automotive diagnostic tool for dealers and repair centres to analyse Dodge, Chrysler, and Jeep vehicles. Unlike with WIFI access points, considerable investment in software R&D would have been required here – but the use of FLOSS allowed this R&D to focus entirely on the automotive diagnostics aspects, rather than the operating system and other essential underlying functionality.

Each example above represents a very different situation where the existence of FLOSS components substituted for internal software R&D. But some general indications may be possible. In a survey of embedded Linux developers conducted by the Technical University of Munich and MIT[177], employees at hardware manufacturers represented over 50% of respondents and indicated that their organisations release to the public on average 45.5% of their own FLOSS developments. Given that the contribution of any one company to embedded Linux is very small (not more than 5%), the fact that almost half of their contribution is considered worth releasing indicates a high degree of dependence on the embedded Linux community and on contributions by others.

The code reuse share of the Debian collection (Table 29 on page 140) may indicate a baseline for the general potential for code reuse as it represents the extent of code reuse in a perfect free market with all else equal, i.e. when no specific commercial considerations depending on a particular market for a particular product interfere in the decision to reuse or create new code. This 36% of code reused (as a lower bound) within Debian represents the achievement of a self-organised "free market" for code reuse.

We can assume therefore that firms who approach the existing code base of FLOSS precisely in order to reuse code could significantly improve upon the share of code reused. In the Maemo case, Nokia achieved 98% FLOSS code reuse. Nokia thus achieved roughly a 98% budget reduction in R&D spending (on the Maemo code base) – or, rather, was able to direct this share of R&D spending towards other, more innovative and risky aspects.

Thus, we can reasonably argue that R&D substitution effect of FLOSS code is at least 36%. This means either that software R&D budgets could be reduced by 36% and substituted with FLOSS code, or, more likely, that R&D budgets could be made more effective by directing at least about a third of it towards more innovative, perhaps riskier developments, using FLOSS code to reduce the need for baseline R&D and FLOSS development models (such as code release, collaborating with potential competitors) to extend pre-competitive research to more applied areas.

[177] Henkel, J. and Tins, M. (2004) Munich/MIT Survey: Development of Embedded Linux, available online at http://www.inno-tec.bwl.uni-muenchen.de/forschung/henkel/MunichMIT-Survey_Embedded_Linux_2004.pdf,

8.4. Economic impact of ICT

As a point of departure on the economic impact of FLOSS we start with the impact of ICT, and of software in special, on economic and productivity growth. A common methodology for such analysis is the so called growth accounting framework, see e.g. Jorgenson (2001), Daveri (2002), van Ark et al. (2003) and Meijers (2004) for recent applications. The impact of ICT can be divided in two main categories: that which originates from the *production* of ICT and that, which comes through the *use* of ICT. Table 30 compares the development of labour productivity growth (a main component of competitiveness) between the EU and the U.S. in the first and second half of the 1990s. It shows an increase of labour productivity in the total economy in the U.S, from 1.1% to 2.5% whereas it decreased in the EU from 1.9% to 1.4%. . A more detailed look, however, reveals that Europe's productivity growth is particular strong in ICT producing services telecommunications and computer services[178]) and in ICT using manufacturing whereas U.S. productivity growth is strong in ICT producing manufacturing and ICT using services. Moreover, both industries have a higher share in GDP in the U.S. as compared to the EU that amplifies the differences and, in the end, explains the different pattern in labour productivity growth between the two regions.

Table 30: Productivity development in the EU and the U.S. (in annual percentages).

	Productivity Growth				GDP share	
	1990-1995		1995-2000		2000	
	EU[a]	US	EU	US	EU	US
Total Economy	1.9	1.1	1.4	2.5	100.0	100.0
ICT producing Industries	6.7	8.1	8.7	10.1	5.9	7.3
ICT producing Manufacturing	11.1	15.1	13.8	23.7	1.6	2.6
ICT producing Services	4.4	3.1	6.5	1.8	4.3	4.7
ICT using Industries	1.7	1.5	1.6	4.7	27.0	30.6
ICT using Manufacturing	3.1	-0.3	2.1	1.2	5.9	4.3
ICT using Services	1.1	1.9	1.4	5.4	21.1	26.3
Non-ICT Industries	1.6	0.2	0.7	0.5	67.1	62.1
Non-ICT Manufacturing	3.8	3.0	1.5	1.4	11.9	9.3
Non-ICT Services	0.6	-0.4	0.2	0.4	44.7	43.0
Non-ICT Other	2.7	0.7	1.9	0.6	10.5	9.8

a) EU figures shown here are for Austria, Denmark, Finland, France, Germany, Ireland, Italy, Netherlands, Spain, Sweden and the United Kingdom, representing over 90% of EU GDP. Notes: Productivity is defined as value added per person employed

Source: van Ark, Inklaar and McGuckin (2003)

The growth accounting analysis departures from the assumption that firms optimise profits such that each factor of input, like labour, ICT-capital, non-ICT capital etc., is used up to the point where the marginal product equals its cost. This implies that spillovers that come

[178] The contribution of both sub-sectors to total labour productivity growth is higher in the EU than in the US. Though telecommunications has a larger impact in both regions, computer services in the EU is nearly as important in the EU whereas it is much smaller in the US.

through externalities are ruled out in such a framework. In a MERIT study for the European Commission (MUTEIS project, IST-2000-30117) we extended this analysis following the work of Barro (1999) by allowing for network externalities, next to the direct effects that are captured by the growth accounting analysis presented above. Using a panel structure for the main European countries and the U.S., we found that both software and telecommunication investments lead to spillovers whereas hardware did not show any, or even slight negative, additional effects (Meijers (2004)). Table 31 takes these effects into account and shows, using slightly different data, network effects account for 0.85% additional labour productivity growth in the US and about 0.7% in the EU. Note that these effects outweigh the direct effect of ICT investments on productivity growth through capital deepening. This implies that network effects are important and that they have a considerable effect on productivity growth.

Although there are insufficiently detailed data on the use of FLOSS, which is rarely visible in national accounts and macroeconomic data, in order to construct comparable figures on the economic impact of FLOSS, we know that network effects and spillovers are stronger in the development of FLOSS as compared to proprietary software[179]. Additional to that, since FLOSS is publicly available in its most extensive form Tuomi (2005), knowledge spillovers from FLOSS production towards the proprietary software sector is substantial. Moreover, there are indications that the use of FLOSS is stronger in Europe as compared to the US[180] and certainly so in terms of the development of FLOSS, as seen in sections 6.3 and 7.1.1. This gives at least some lower bounds of possible impacts of FLOSS on productivity and these figures will be used in a simulation model that includes the main relations between final production, productivity, research and development and the use of ICT in general and of software and FLOSS in particular. The model explores the effect of FLOSS on innovation and competitiveness in the ICT sector and the effect on economic growth, and is described later in section 8.5, “Modeling the economic impact of FLOSS on innovation and growth”, on page 153 onwards. The model explores in particular whether increased use of FLOSS could help Europe compensate for its consistently low share of investment in software as a share of GDP, in comparison to the U.S. (see Figure 56 on page 152).

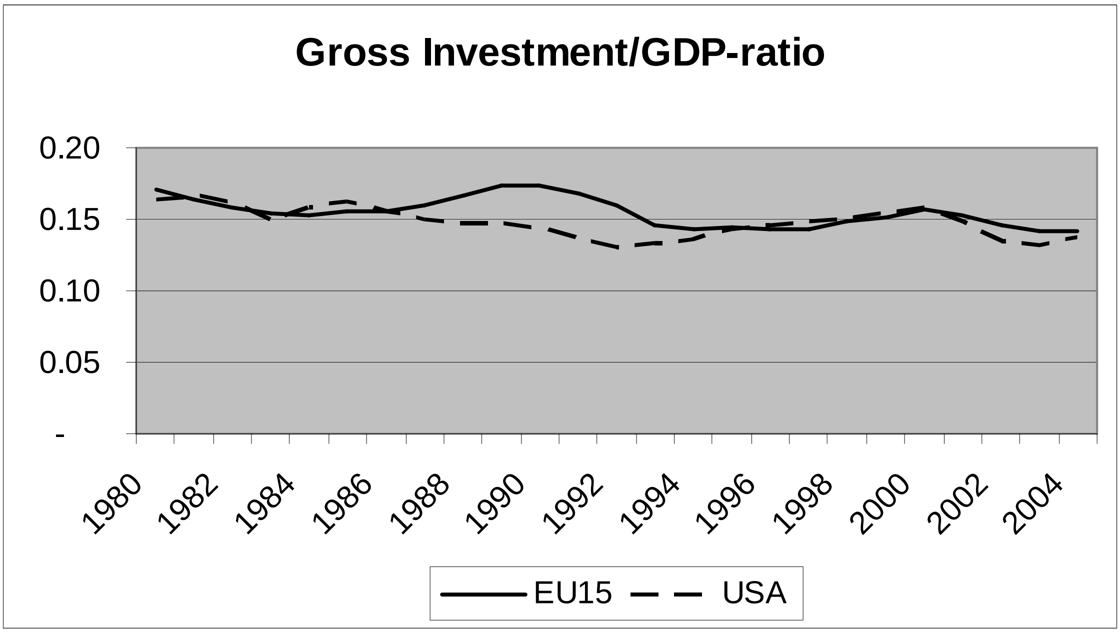
[179] See discussion of knowledge diffusion in section 8 including evidence on how FLOSS is valued by firms and innovators within firms more than patent databases as a source of new ideas (around page 115)

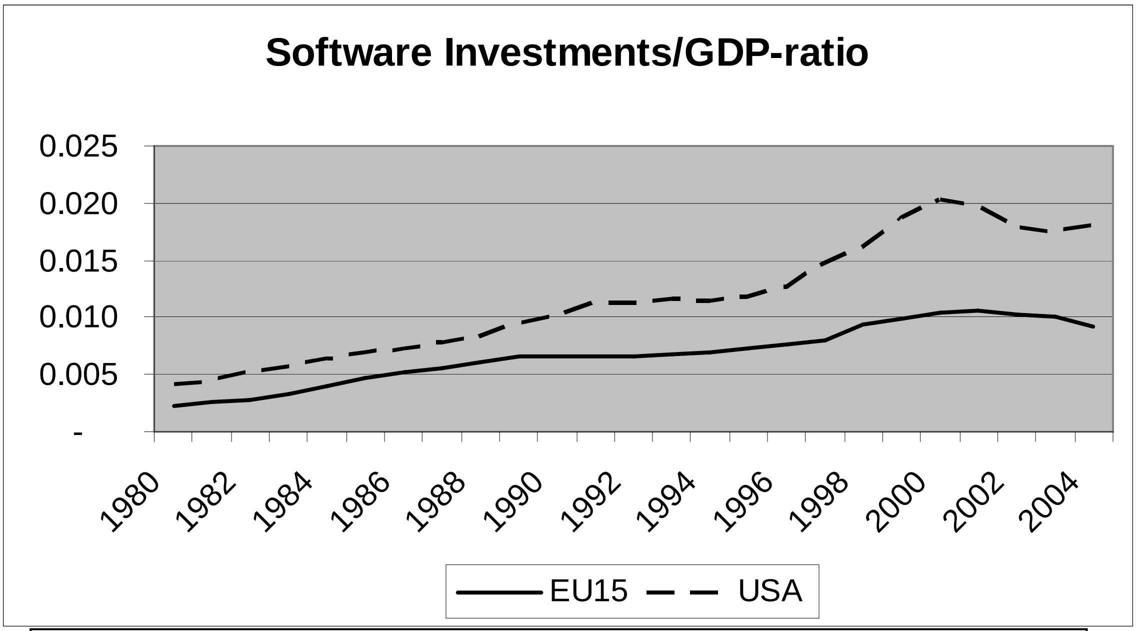
[180] See section 6.2.3.2, and figures from IDC showing the Linux share of PC sales about twice as high in Europe as in the US (IDC, 2004, “The Linux Marketplace - Moving From Niche to Mainstream”, prepared for OSDL)

Table 31: Labour productivity growth and sources (%), adjusted for network effects.

	1990-1995	1995-2001	1995-2001 over 1990-1995	
EU				
Labour productivity	2.43	1.37	-1.07	
Information Technology	1.05	1.16	0.11	
Capital deepening	0.29	0.42	0.13	
Network effects, software	0.69	0.66	-0.03	
Network effects, telecommunication	0.07	0.08	0.01	
Non information technology cap deepening	1.01	0.48	-0.53	
Utilisation rate effect	0.23	-0.15	-0.38	
Total factor productivity	0.14	-0.12	-0.27	
US				
Labour productivity	1.19	1.85	0.66	
Information technology	1.29	1.66	0.37	
Capital deepening	0.40	0.72	0.33	
Network effects, software	0.85	0.86	0.01	
Network effects, telecommunication	0.04	0.08	0.04	
Non information technology cap deepening	0.19	0.32	0.13	
Utilisation rate effect	0.19	-0.04	-0.23	
Total factor productivity	-0.48	-0.09	0.39	
Difference US-EU				
Labour productivity	-1.24	0.48	1.72	
Information technology	0.24	0.50	0.26	
Capital deepening	0.11	0.30	0.19	
Network effects, software	0.16	0.20	0.04	
Network effects, telecommunication	-0.03	0.00	0.03	
Non information technology cap deepening	-0.82	-0.16	0.66	
Utilisation rate effect	-0.04	0.11	0.15	
Total factor productivity	-0.62	0.03	0.65	
Copyright © 2004 MERIT. Source: Meijers (2004)				

Figure 56 Gross investment and software investment to GDP ratios, EU15 and U.S.





© MERIT, data source: Groningen Growth and Development Centre, www.ggdc.net

8.5. Modeling the economic impact of FLOSS on innovation and growth

Since the advent of new growth theory, the origin of economic growth has been closely linked to the arrival of new ideas and the accumulation of human capital at the expense of physical capital accumulation (Lucas (1988), Romer (1990), Aghion and Howitt (1998), Kaldor (1961)). While growth theorists have emphasised different aspects of the growth process, all of them agree on the central role of knowledge accumulation in explaining growth. The use and the effective generation and diffusion of knowledge have become the corner stone of new-growth theory. Inspired by the growth accounting literature, growth theorists have been focussing attention on the details of knowledge production and knowledge diffusion. In this section, the main mechanisms that translate the use of ICT, and in particular the use of FLOSS and proprietary software (PROPS) into growth are examined. To what extent can software contribute to macro-economic growth and productivity growth? What might the difference between FLOSS and PROPS imply in this context? Given the scope and coverage of the study, in this section the focus is on the effect of FLOSS.

The contribution of ICT to economic growth and productivity growth are traditionally measured in a growth accounting framework (Jorgenson (2001), Gordon (2004), Daveri (2002)). Although this is not without methodological problems (see e.g. Barro (1999) and Basu, Fernald et al. (2001) and Basu, Fernald et al. (2003)), growth accounting is a widely accepted method. One of the debates is whether ICT in general and software in particular lead to spill over effects and network externalities (see Stiroh (2002), O'Mahony and Vecchi (2005), and Meijers (2004b). If so, the growth accounting method underestimates the contribution of software to economic growth and productivity growth. Moreover, part of the growth that is attributed to total or multifactor productivity in growth accounting studies could be attributed to software investments.

Measuring the contribution of FLOSS in a growth accounting setting relies on factor shares of total income/total costs and since this type of software is, at least in principle, freely available the factor share is zero. FLOSS is likely to generate spillovers and network effects, even stronger than proprietary software so from that perspective the growth accounting framework is also not applicable. However, these difficulties do not arise if we model the contribution of software, both PROPS and FLOSS in an endogenous growth setting.

With a macro-economic growth perspective, the focus is on the user side of ICT (and software in particular), and one can construct an analytical growth framework in which one can distinguish between potentially different linkages between ICT use and economic growth. Production of software and its contribution to economic growth, being either through FLOSS or PROPS, is not taken into account. The framework will contain the two ultimate growth mechanisms, i.e. human capital accumulation and love of variety (LOV). LOV refers to specialisation through the ongoing division of (produced) resources. Because of the latter, the match between productive and consumptive needs and the specialised products resources used to fulfil these needs improves over time, as does productivity at the aggregate level, measured as produced final output/utility per unit of ‘raw’ resource.

In the formal growth model, there are three main channels through which ICT is thought to have a direct and significant impact on economic growth:

1. ICT use increases the productivity of the human capital accumulation process, including the 'shareability' of human capital. As in the Lucas (1988) model, the steady state growth rate depends positively on the productivity of the human capital accumulation process, and as this is widely regarded as a valid result, it follows immediately that investment in ICT in the context of human capital accumulation activities directly improves growth if it increases the productivity of the human capital accumulation process;[181] (Helpman (1998), van Zon (2001))
2. ICT use in (intermediate) output production enhances the productivity of the other inputs directly, through a better organisation of the internal production process on the one hand, but also through the embodiment of ICT in machinery and equipment, including 'general purpose' equipment such as micro-computers; (Harris (1998), Jorgenson (2001), Gordon (2000))
3. ICT use in (intermediate) output production allows the production sectors to become 'leaner and meaner' on the one hand (through an ICT-based improvement of the transparency of the market for (produced) inputs), while on the other hand it also allows firms to distinguish themselves more clearly from other firms by enabling them to improve the match between the characteristics of the products they produce and customer needs that become increasingly more specific. ICT allows firms to customize their products more easily and thus leads to more varieties (see Barua and Kriebel (1991), Becchetti, Londono Bedoya et al. (2003) and OECD (2004)).

So where and how does FLOSS as opposed to PROPS fit into this picture? First, FLOSS is freely available. This does not mean that it doesn't take resources to create or use FLOSS, but the costs are relatively low compared with PROPS (see section 7.6, "User benefits: interoperability, productivity and cost savings"). This then implies that the penetration of FLOSS is to some extent limited by the extent of the market, while the penetration of PROPS is limited both by the market and by the degree of monopoly power held by the PROPS-developers. Generally speaking, FLOSS software has a higher degree of penetration ex ante than pre-packaged PROPS, even though ex post the situation is less clear cut − the use of FLOSS software may be limited by the positive network effects and induced lock-in effects associated with using de facto standard software as Microsoft Word (e.g. Shapiro and Varian (1999), Klemperer and Farrell (2006)). Second, FLOSS allows users to tap into a software base that enables them to build dedicated software at relatively low cost, by stacking high-value software features on top of the 'standard features' already included in the FLOSS code, i.e. by tailoring the 'relatively-general-purpose' FLOSS code to firm specific needs (see Section 8.3, "FLOSS, R&D substitution and the impact of collaborative strategies"). Finally, support, in the form of debugging and the addition of new features happens generally more rapidly with FLOSS than with PROPS, as new features and new versions are produced by the users themselves who act primarily on their own behalf as they envisage specific high value-added uses of the improved software.

[181] The downside of ICT investment in human capital accumulation is of course that ICT investment has an opportunity cost, for example, in terms of physical capital investment, but also in terms of time available for the allocation to human capital accumulation.

Getting back to the question how FLOSS could be linked to points 1-3 above, one could think of extending the Lucas (1988) growth framework, by introducing an ICT-capital complex into the human capital production function, where the contribution of a dollar spent on FLOSS versus a dollar spent on PROPS has a higher impact on human capital production productivity a priori (due to the way FLOSS, unlike PROPS, leads to deeper skills learning effects, see Aection 7.4.1). The reasons for this would be both the higher ex ante degree of penetration, and the tailor-made character of FLOSS relative to PROPS. The output production side of the Lucas model can be made to accommodate the specialisation features implied by the tailor-made character of FLOSS, by integrating LOV in the production function. That allows for two classes of ICT users − those hooked (primarily) to FLOSS and those to PROPS. We can link the network and specialisation features by taking the extent to which the set of all firms taken together may take advantage of the maximum possible degree of specialisation (as generated by research activities as in the Romer model (1990)) to be a positive function of the use of FLOSS and PROPS.

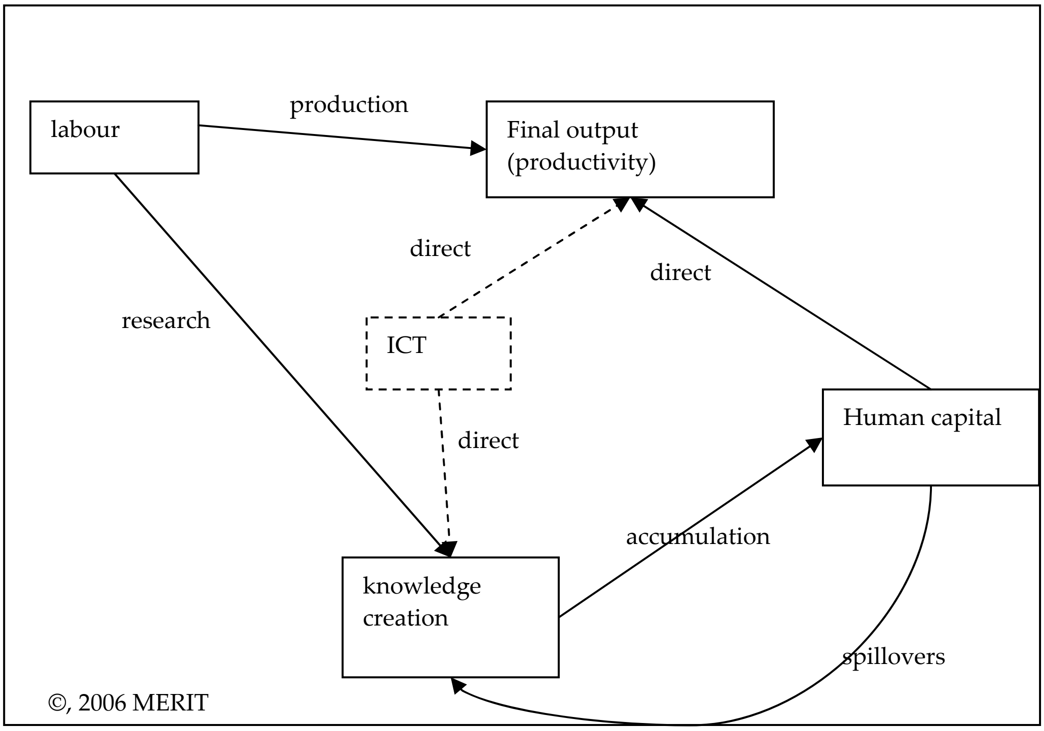
The organisation of the rest of this section is as follows. Section 8.5.1 describes the basic model and the extended model in a non-analytical way. Section 8.5.2 presents the analytical framework in more detail. Section 8.5.3 contains a number of illustrative simulations that provide some first rough estimates of the potential contribution of FLOSS to economic growth in a general sense. The estimates are meant to illustrate the principles involved, and to obtain the signs and orders of magnitude of the contribution of different speeds of diffusion of FLOSS to economic growth. To this end, the structural parameters of the underlying model are calibrated using the results from Meijers (2004a) and Meijers (2004b). Section 8.5.4 contains a short scenario analysis in which the investments in FLOSS in Europe are increased and the results are compared and contrasted with figures for the U.S. Section 8.5.5 presents the conclusions. The model is described in Appendix 1.

8.5.1. The Model: a schematic overview

We start from an endogenous growth model that includes the generation of knowledge through learning and R&D as described by Lucas (1988) to model and simulate the economic impact of FLOSS. Van Zon (2001) includes ICT in the general framework of Lucas. We first briefly describe the model as developed by van Zon (2001) and extend this model by introducing the software sector more explicit. Here we provide a brief, schematic overview of the basic model and the model including FLOSS ands PROPS. (Refer to Appendix 1 for detailed analytical description of the model).

The model structure as developed by van Zon (2001) is given in Figure 57. Labour can be used to produce final output or to perform R&D. R&D increases the stock of human capital, which increases the technological capabilities of the economy, i.e. final output production becomes more productive. The increased stock of human capital also enhances the R&D process such that the latter becomes more productive too. ICT has two effects. First it has a direct effect on the productivity of final output, similar to the findings of the growth accounting studies mentioned above. Second, it facilitates the R&D process directly, mainly due to the global and instantaneous availability of knowledge and information, and has hence also an indirect effect on final output production and productivity. By enhancing the production process of knowledge itself, ICT affects human capital accumulation and therefore has an indirect effect on final output production as well as on the knowledge creation process through spillovers.

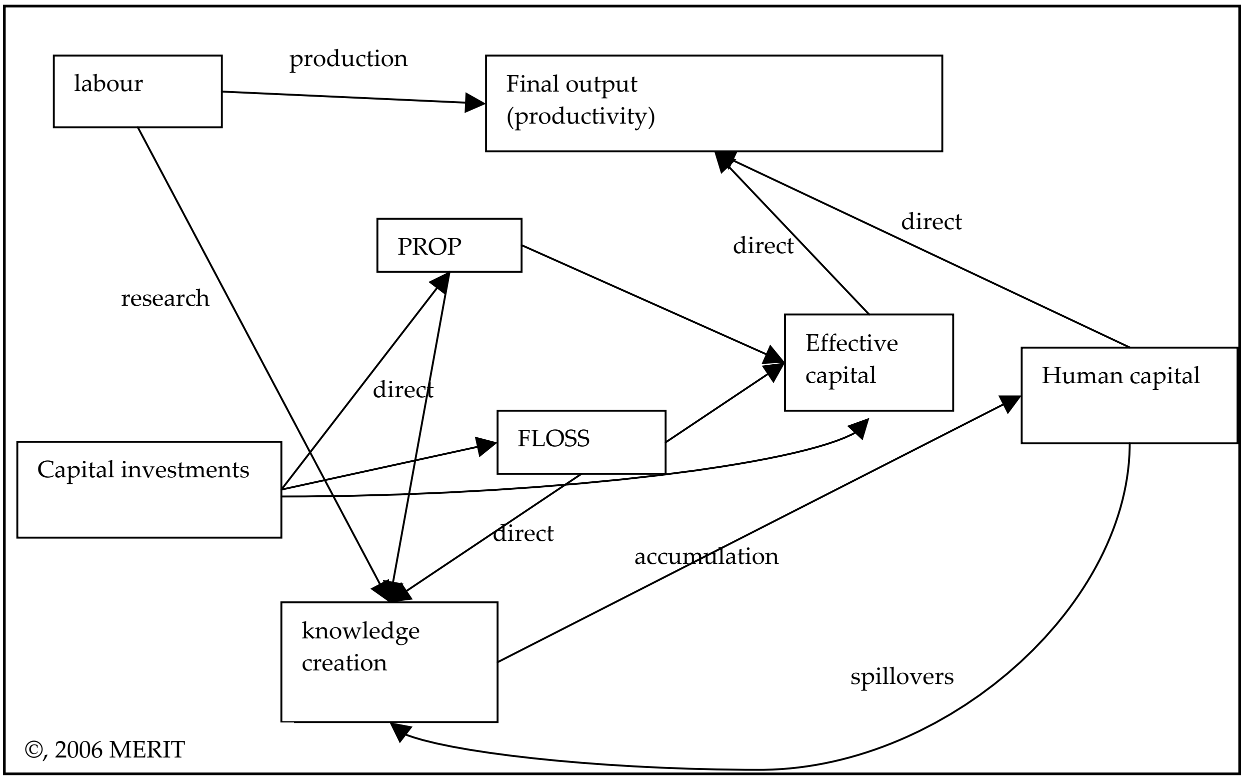
Figure 57: Endogenous growth model including ICT.



The framework described above can be extended to include the production and use of open source software next to proprietary software. Because of our focus on the use of software, we exclude other ICT capital goods to keep the model as simple as possible. As in van Zon (2001), we also exclude the production of software and the positive effects of FLOSS and PROPS in this process.

In the extended model, software production is similar to the accumulation of knowledge: it increases the productivity of the final output sector and it increases the productivity of human capital formation (Figure 58). The differences between PROPS and FLOSS is that the latter comes cheaper than the former (or at a higher quality) and that tailor-made characteristics of FLOSS can lead to more varieties in the production process and hence increases the overall productivity of that process. Basically the model runs as follows: labour can be used in final output production or in human capital creation, i.e. in the R&D sector. A fraction of the total labour force is employed in the final production sector and the remaining part in knowledge creation. Capital investments go either directly in non-ICT capital and thus add to the non-ICT capital stock or in it goes into software, which can be either PROPS or FLOSS. The stock of software capital has a positive impact on human capital creation process as in van Zon (2001) and as argued under point A in the introduction. Moreover PROPS and FLOSS also directly affect the final production process along the lines of argument B, which is basically also captured by growth accounting models. Finally, software also affects the final production process through more efficient customisation such that more varieties, or more versions, becomes available, as argued under point C in the introduction.

Figure 58: Endogenous growth model extended with software, PROPS and FLOSS.



8.5.2. The Model: description of key parameters

We will not derive a fully inter-temporal consistent allocation of resources for this particular setting, but instead use the behavioural approach strongly advocated by Solow (2000). Exogenously determined values for the various (investment and hence) saving rates are used rather than the fixed values normally implied by the use of a Constant Inter-temporal Elasticity of Substitution (CIES) utility function when in the steady state and a variable one outside of the steady state. Moreover, we keep these saving rates as constant fractions of final production, i.e. as constant fraction of GDP. Again, details of the model are presented in Appendix A; here we limit ourselves to a brief description of the relevant parameters that are used to calibrate the model and to carry out some sensitivity and scenario analyses in next sections.

8.5.2.1. The quality impact of FLOSS relative to PROPS

In order to accommodate the idea of the continuing division of economic activities to meet more and more specific needs that lead to productivity growth at the aggregate level, we use the Ethier function to define the effective capital stock in function of cumulative investment (i.e. the stock of 'raw' capital) and the degree of specialisation enabled by the ICT capital stock used in the final output sector. We distinguish different activities and a fraction of all activities is supported by PROPS and the other fraction by FLOSS. As pointed out earlier, FLOSS-based economic activities are likely to be of higher quality due to the greater customisation possibility of FLOSS[182] and due to the fact that FLOSS comes cheaper[183]. So

[182] Evidence provided in the FLOSSPOLS survey of European governments showed that users who value customisation are more likely to use – and increase their use of – FLOSS. The survey also showed that customisability is considered quite important, and thus an indication of quality.

we introduce a quality/price ratio in the model that indicates this difference between FLOSS and PROPS. This quality/price parameter can be set to 1 indicating no quality difference between PROS and FLOSS and it can be set to e.g. 1.1 indicating a 10% quality/price difference (in favour of FLOSS)[184].

8.5.2.2. Division of economic activities

Next to the (potential) difference in price (or quality) between PROPS and FLOSS as described above, the tailor-made characteristics of FLOSS can be included in the model by assuming that the growth rate of the number of different activities in the economy depends on the labour force measured in human capital units and also on the ICT-capital intensity in both the FLOSS based and the PROPS based group of activities. This implies that the growth rate of the number of activities depends positively on the size of the FLOSS based capital stock and the PROPS based capital stock.

8.5.2.3. Investment in Software and in 'raw' capital

The model satisfies the macro-economic budget constraint of a closed economy stating that total income is either consumed or saved and thus invested in new capital. As stated above, we did not include an explicit inter-temporal consumption function in which consumption and savings rates follow from optimising consumers. Instead, and in order to keep the model as simple as possible, we apply a fixed consumption rate and thus a fixed investment rate. Investment can be either in 'raw' capital (or non-ICT capital) or in software and we assume that a constant fraction of total income (i.e. of GDP) is invested in physical (non-ICT) capital and another fraction is invested in software capital. These investments add to the existing capital stock but this stock also decreases due to depreciation.

A fraction of the software capital stock is used in human capital formation and the remainder is used in final output production. This remainder is again subdivided in a fraction that is PROPS based and another (remaining) faction that is FLOSS based. So we include a measure of the use of ICT in human capital formation and a measure of the relative importance of FLOSS versus PROPS.

8.5.2.4. Production of human capital

The growth rate of human capital depends on the fraction of labour that is attributed to human capital formation (and thus is not used in final output production) and on the fraction of ICT-capital that is used in human capital formation. As for labour, the remaining fraction of ICT-capital is used in final output production. The relative contribution of ICT-capital in human capital formation is denoted by a parameter (called γ in the formal model) and the

[183] This is clear even if we avoid the entire discussion on TCO and other ways of evaluating cost (see section 7.6), since very large numbers of FLOSS applications are downloaded for no payment (see discussion in section 6). If some of these translate into commercial services, this drags the entire average price downwards.

[184] This is a reasonable, conservative assumption given the lower price (average unit price as used in this model) of FLOSS, and the fact that organisations that choose FLOSS cite among the top reasons for their choice various quality attributes such as "Better access protection", "Higher stability" and "better price to performance ratio" – see the 2002 FLOSS User Survey (flossproject.org/report/ - Part 1: "Use of Open Source Software in Firms and Public Institutions")

complement of this parameter denotes the relative importance of human capital itself in this process. In Lucas (1988) the growth of human capital depends on the size of human capital stock only and here we included ICT-capital as an additional factor. The relative contribution of both sources can be varied by changing parameter γ where $0 \le \gamma \le 1$.

By changing the values of the parameters described above and subsequently simulating the model numerically forwards in time, we can learn more about the corresponding level and growth-effects of such changes. We have to use such a numerical exercise, as it is impossible to obtain a closed form analytical solution, even with this simple ICT-expanded Solow growth model.

It should be noted that in our model human capital is just one of the sources of growth, next to LOV. Furthermore, as we have constant saving rates, it follows that in the long run the ratio of the ICT capital stock relative to the human capital stock will have a tendency to grow, leading to accelerating growth of human capital. If one doesn't want this, one simply sets the relative contribution of ICT-capital in the growth (or production) of human capital to zero and hence excludes a priori one of the channels through which ICT may generate growth. An alternative to such a measure would be to provide 'growth leakages' in the human capital generation process, for example by having decreasing returns to scale. As this complicates the analysis considerably, we will investigate the effects of shocks in the various system parameters by measuring the induced changes in the system variables relative to their base-run values (that may include accelerating growth therefore).

8.5.3. Model Simulations

8.5.3.1. Parameter Values and Calibration

In this section we present some simulation results that, of course, depend on the values of the structural parameters chosen. Indeed, there is relatively little empirical evidence available about the strengths of the various mechanisms on hand. Nonetheless, even without precise and reliable information one would like to know about the orders of magnitudes and the signs involved of the impact of changes in the penetration of FLOSS relative to PROPS on aggregate productivity growth. To this end, we have drawn up some rough a priori estimates of some of the structural parameters (refer to bold text in Table B.1 in Section 11.1 Parameter values), as well as less rough estimates of other parameters, mainly obtained from van Ark, Melka et al. (2002)[185]. The values for structural parameters we have used for our scenario analysis are presented in the table. Figure 56 shows that investment in software is about 2% of GDP in the U.S. whereas it is about 1% in EU15[186]. The total investment rate is about the same for both regions and fluctuates roughly around 15% of GDP. The difference in software investments is taken into account in all simulation exercises to emphasize the difference between the two regions and to investigate the potentials of FLOSS to close the productivity gap (at least to some extent). So the investment rate in

[185] Similar figures can be obtained from e.g. Jorgenson (2001), Colecchia and Schreyer (2002) and Daveri (2002) but van Ark, Melka et al. (2002) include software as a separate factor in their analysis (next to IT hardware and telecommunications) and they include data for both EU15 and the USA using one framework.

[186] Note that this measure does not include the value of the FLOSS software in a way equivalent to the investment in proprietary software licences. See our estimates of this in Section 9.

software is 2% and 1% for the US and EU15 respectively and investments rates in non-ICT capital is equal to 0.13 and 0.14, respectively.

All parameter values listed in Section 11.1, “Parameter values”, containing a subscript zero refer to initial values of the various stocks of capital, including human capital, as well as the labour force. The parameters listed in normal font (e.g. non-bold) are the most reliable, in the sense that they can in part be obtained from the National Accounts, such as the various investment rates and the value of the share of labour in the total distribution of income. The ratios of the various initial stocks have been obtained from Jorgenson (2001), van Ark, Inklaar et al. (2003) and are partly displayed in Figure 56. The value of the initial human capital stock has been roughly calibrated, given the values of the other parameters. The parameter values shown in bold font cannot readily be obtained from the National Accounts. For the parameters listed in bold and that are underlined, we will investigate the sensitivity of the productivity growth outcomes for various changes in the numerical patterns listed in Table B.1, while leaving the general order of magnitude of the parameters unchanged. Because the number of runs is increasing exponentially with the number of different combinations of parameter values, we did not include the parameters in bold font and only in sensitivity analysis, primarily to save on computer time and outcome processing time, and because we wanted to concentrate on the sensitivity of the outcomes for changes in the parameters that can be linked directly to differences in FLOSS and PROPS. As regards the latter, it should be stressed again here that the autonomous growth rates in the generation of new varieties have been calibrated for Europe and the U.S. in such a way that they reproduce the productivity growth results of Europe and the U.S., in combination with the other parameter values listed in Table B.1. They can therefore be thought of as playing the role of a growth residual. From the point of view of investment in ICT including software, this residual covers the growth coming from all other sources, apart from human capital formation, and influences productivity growth through variety expansion, as the main mechanism of growth next to human capital formation that has already been explicitly accounted for.

The parameters in bold are directly related to the impacts of FLOSS and PROPS on productivity growth, either through their effect on variety (through the ψ parameters), or though intrinsic differences between FLOSS and PROPS (through the quality/price parameter), or differences in the speed of diffusion of FLOSS and PROPS captured by changes in the $\overline{w}$ parameter. Finally, we also look at how changes in the impact of ICT on the accumulation of human capital would affect productivity growth (changes in the γ parameter).

8.5.3.2. Sensitivity Analysis

In this section we present the outcomes of the model simulations we have performed using different combinations of parameter values. We limit our discussion to the results for the U.S., as the models for the EU15 and the U.S. are structurally the same. The main difference between the two models lies in the values of the investment rates as well as in the autonomous terms used to calibrate the model for both regions (see also Table B.1 in Appendix 1). The various outcomes for different parameter constellations are discussed below, but only after we have presented the outcomes of the Base-Run.

All the growth rates calculated in the different runs refer to the ones that arise in the steady state, i.e. we concentrate on the long-term effects of changes in parameters (and policies). We obtain these growth rates by simulating the model for 200 time-periods, starting from time 1, using the different parameter constellations we want to investigate, and then picking the growth rates at the end of this simulation period. Because there are no outside (random) disturbances, the time paths for the various growth rates are generally converging smoothly through their long-term values.

The results for the U.S. are shown in Table 32 and the results for EU15 in Table 33. The results for both cases are very similar, except for the presence of the difference in the autonomous terms, as well as in the difference between the investment rates in software which is twice as large in the U.S. than in Europe (2% versus 1%). The overall rate of investment is the same in both countries, though.

Table 32: Simulation results for the U.S.

Parameter	Base-run	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7
Contribution of human capital in FLOSS based number of varieties(ψ_0^F)	1	1	1	1	0.9	1	1	1
Contribution of ICT-capital in FLOSS based number of varieties (ψ_1^F)	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3
Contribution of human capital in PROPS based number of varieties (ψ_0^P)	1	1	1	1	0.9	1	1	1
Contribution of ICT-capital capital in PROPS based number of varieties (ψ_1^P)	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1
Quality of FLOSS ($\overline{q}$)	1	1.1	1	1	1	1	1	1
PROPS based fraction in human capital formation ($\overline{w}$)	0.8	0.8	0.7	0.8	0.8	0.8	0.8	0.7
Relative contribution of ICT-capital in human capital formation (γ)	0.9	0.9	0.9	0.8	0.9	0.9	0.9	0.9
Simulation outcomes (all in growth rates):								
FLOSS based number of varieties ($\hat{A}^F$)	1.201	1.201	1.201	1.573	1.434	1.448	1.820	1.8255
PROPS based number of varieties ($\hat{A}^P$)	1.201	1.201	1.201	1.573	1.434	1.246	1.323	1.326
Human capital ($\hat{h}$)	0.423	0.427	0.423	0.756	0.454	0.443	0.475	0.476
Effective capital stock ($\hat{K}^e$)	3.427	3.443	3.427	4.548	4.055	3.874	4.712	4.730
ICT-capital stock ($\hat{K}_i$)	2.200	2.210	2.200	2.929	2.583	2.468	2.959	2.970
Output and labour productivity ($\hat{Y}$)	2.215	2.226	2.215	3.015	2.599	2.488	2.996	3.007

Table 33. Simulation results for the EU15.

Parameter	Base-run	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7
Contribution of human capital in FLOSS based number of varieties(ψ_0^F)	1	1	1	1	0.9	1	1	1
Contribution of ICT-capital in FLOSS based number of varieties (ψ_1^F)	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3
Contribution of human capital in PROPS based	1	1	1	1	0.9	1	1	1

number of varieties (ψ_0^P)								
Contribution of ICT-capital capital in PROPS based number of varieties (ψ_1^P)	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1
Quality of FLOSS ($\overline{q}$)	1	1.1	1	1	1	1	1	1
PROPS based fraction in human capital formation ($\overline{w}$)	0.8	0.8	0.7	0.8	0.8	0.8	0.8	0.7
Relative contribution of ICT-capital in human capital formation (γ)	0.9	0.9	0.9	0.8	0.9	0.9	0.9	0.9
Simulation outcomes (all in growth rates):								
FLOSS based number of varieties ($\hat{A}^F$)	0.714	0.717	0.714	0.896	0.838	0.86	1.066	1.069
PROPS based number of varieties ($\hat{A}^P$)	0.714	0.717	0.714	0.896	0.838	0.738	0.775	0.776
Human capital ($\hat{h}$)	0.356	0.359	0.356	0.519	0.369	0.366	0.379	0.38
Effective capital stock ($\hat{K}^e$)	2.159	2.172	2.159	2.704	2.501	2.41	2.824	2.836
ICT-capital stock ($\hat{K}_i$)	1.435	1.444	1.435	1.792	1.649	1.588	1.836	1.843
Output and labour productivity ($\hat{Y}$)	1.434	1.443	1.434	1.824	1.643	1.587	1.839	1.846

8.5.3.3. The Base-Run

It will not come as a surprise that the base-run reproduces the results for the U.S. almost perfectly (cf van Ark, Melka et al. (2002)). This was the aim of the calibration of the autonomous terms in the growth rate of number of varieties. Output per head[187] is growing by 2.2%, and it grows far faster than the growth of human capital per person (0.4%). The growth rate of the stock of ICT capital is equal to that of output as the investment rates are constant. However, the growth rate of the effective capital stock exceeds the one of the physical stock by far due to the LOV effect. As in the base-run both FLOSS and PROPS add symmetrically to variety by construction, the growth rate of physical capital (2.2%) and that of variety (1.2%) added together, equals that of the effective capital stock (3.4%).

8.5.3.4. Changes in the quality of FLOSS

If we raise the quality/price ratio of FLOSS (indicated by $\overline{q}$ in the formal model) by 10%, this implies that for the same spending of resources on FLOSS, the effect will be 10% higher as well. One would expect a faster accumulation of effective capital, both through the higher quality or lower price of the FLOSS contribution, and also because of secondary spillover effects that run from higher growth of the effective capital stock to higher growth of output to higher growth in investment in ICT capital and through the contribution of the latter to human capital formation; and also due to further increases in growth through faster human

[187] Since we keep the number of heads constant, the growth rate of the labour force is zero, labour productivity grows as fast as output itself.

capital accumulation. All these effects do indeed occur but they are relatively small to the base-run. For example, output growth increases by 0.011 percentage points from 2.215% to 2.226%, while the secondary effect on human capital growth is an additional 0.004%, raising the human capital growth rate from 0.423% to 0.427%. The growth rate of varieties is raised by the same amount from 1.201% to 1.205%. The order of magnitude of the results are in line with what one would expect on the basis of the study performed by van Ark, Melka et al. (2002) and Meijers (2004b). They conclude that the contribution of investment in software to productivity growth is limited to begin with, and for FLOSS particularly limited, as it represents a relatively small share of total investment in software (in the order of 20%[188], which is denoted by one minus the PROPS based fraction in human capital formation in the tables). In the growth accounting studies, the effect of software on labour productivity growth is 0.11percentage point for the EU15 and 0.26 percentage point for the U.S., measured over the period 1995-2000 as reported by van Ark, Melka et al. (2002). Similar figures are found by e.g. Jorgenson (2001) and Daveri (2002). As the FLOSS capital stock is 20% of the total software capital stock given our parameter constellation, raising the quality/price ratio of FLOSS by 10% would lead to an increase in the growth rate of output by 2% of 0.26 percentage point which is 0.0052 percentage point in a pure growth accounting setting. The fact that the effect is much higher in the simulation comes mainly through increased capital accumulation and spill over effects.

8.5.3.5. Changes in the relative importance of FLOSS versus PROPS in human capital formation

A drop of PROPS based fraction in human capital formation ($\overline{w}$) from 0.8 to 0.7 implies a fall in the share of PROPS based software users, and consequently a rise from 0.2 to 0.3 in the share of FLOSS based users. One would expect this to have a positive effect on growth in as far as there is a real difference between FLOSS and PROPS users, i.e. insofar as the quality/price parameter differs from one. This is corroborated in the outcomes, where we find no differences between base-run and a run where $\overline{w}$ falls from 0.8 to 0.7, for a given value of the quality/price of FLOSS being equal to one. When we increase the quality of FLOSS by 10%, i.e. by setting $\overline{q} = 1.1$, and compare the new results with those obtained earlier for $\overline{q} = 1.1$ and $\overline{w} = 0.8$, we find no differences either. This is due to the assumed symmetry between FLOSS and PROPS as regards their contribution to the change in variety, i.e. the four parameters at the top of Table 32 and Table 33. Only when these differ between FLOSS and PROPS, can one expect a change in the distribution of software investment over FLOSS and PROPS to make a real difference. We will come back to this later in the context of the sensitivity results with respect to changes in the parameters that reflect the relative contributions to changes in variety.

[188] As can be seen in the market share figures in Section 6.2. Note that investment in FLOSS is not usually accounted for in national statistics, as there are no licence fees bought as "investment". A number of statistical bodies (particularly the US and the OECD but also some in the EU including Belgium) have attempted to formulate methodologies for counting software investment for which no licence fees are paid, such as own-account "in-house" software, but these may not account for FLOSS software. Section 9 describes our estimate of FLOSS investment.

8.5.3.6. Changes in the importance of ICT capital in human capital formation

The parameter γ measures the relative contribution of ICT-capital in human capital formation process to the rate of human capital formation. Its complement 1-γ signals the relative contribution of the human capital stock itself. We have performed a number of experiments where we changed the relative strength of the contribution of human capital in favour of ICT capital. For example, if we raise 1-γ from its base-run value of 0.1 to a value of 0.2, we find a significant effect on aggregate productivity growth, equal to 0.8 percentage points, i.e. the growth rate rises from 2.215% to 3.015%. The rate of growth of human capital formation almost doubles, going from 0.423% to 0.756%. The large impact comes in part from the fact that we have actually *doubled* the value of 1-γ, but that shouldn't be too difficult in actual fact considering that we started out with a value of 1-γ equal to 0.1. Note that 1-γ defines the partial human capital formation elasticity of the ICT capital stock used in human capital formation. As such it is directly related to the marginal product of the ICT capital stock, which has now been modelled to fall with an increase in the ICT-stock, ceteris paribus. However, positive learning externalities through network effects could no doubt mitigate this drop in the marginal product of ICT capital or even reverse this drop. So, a policy of greater connectivity, or even a faster diffusion of FLOSS, that by its own nature brings about these learning externalities through the way its use and further development are organised, may be instrumental in raising 1-γ. Nonetheless, the absolute size of its impact, and certainly its relative size compared to the effects of the other parameter changes so far, indicates that further integration of ICT, and particularly FLOSS use, given its positive externalities with respect to building the skills to effectively use ICT in all kinds of activities, including learning, is something not to be underestimated a priori.

8.5.3.7. Changes in the parameters representing the contributions of human capital and ICT capital to changes in varieties

We have implemented two different types of changes in the parameters that reflect the contributions of changes in the stock of human capital to changes in variety and the contribution of changes in the stock of ICT-capital to changes in varieties, both to FLOSS based and PROPS based activities in final output. (the ψ parameters in the model). First, we changed the relative weights of the contributions of the growth of the human capital stock and the ICT intensity within the FLOSS and PROPS groups to the growth of diversity, while leaving the total weight in tact for both groups. A second group of experiments concerned the introduction of an asymmetry between the FLOSS and PROPS group, through an absolute (uncompensated) rise in one of these parameters.

The size of the effects of the compensated changes in the ψ parameters are in between those with respect to changes in the quality/price parameter and ($\bar{q}$) and of the importance of the ICT-capital stock in human capital formation (γ). In part, this is because we only experimented with the composition of the contribution of the growth of human capital in general and the ICT intensity in the FLOSS and PROPS groups. By relaxing the relative weight of human capital formation from 1 to 0.9 and raising the relative weight of the growth of the ICT capital intensity from 0.1 to 0.2, ceteris paribus, we find a rise in the rate of growth of output from 2.2% to 2.6%, i.e. a rise of almost 0.4 percentage points. This comes about mainly through a stronger "diversification" of capital services (i.e. a strong growth of the effective capital stock, rising from a growth of about 3.4% to about 4.1%), which is at least in an order of magnitude larger than the rise in the growth rate of the human capital stock (this rises from 0.4% to 0.5%). This indicates that the diversification channel too merits closer scrutiny, just like the literacy-channel in human capital formation, as at first sight the effects seem to outweigh by far those associated with changes in the qualitative differences between FLOSS and PROPS with respect to their direct use in the production of capital services. Their differences, with respect to their ability to create diversity, i.e. meet increasingly different and specific demand-requirements and generate additional utility/value because of that, seem to be particularly important relative to productivity/quality differences per se.

As regards the introduction of an asymmetry in the values of the ψ parameters between the FLOSS and the PROPS groups, we find that by setting $\psi_1^F = 0.2$ instead of 0.1, while leaving the other ψ parameters unchanged, we increase the rate of growth of the FLOSS group relative to that of the PROPS group, as well as raising the rate of growth of total diversity. The asymmetry between FLOSS and PROPS groups is immediately apparent, as the growth rate of the number of FLOSS based varieties (A^F) rises from 1.2 percent in the base-run to 1.4 in the present experiment, while the growth rate of the number of PROPS based varieties (A^P) rises from 1.201% to 1.246%, underlines two things. First, there are now definite asymmetries. Secondly the rise in the rate of growth of ICT capital accumulation as well as human capital accumulation also raises the growth rate of diversity in the PROPS group. The cumulative effect on the rate of growth of output makes that rise from 2.2% in the base-run to 2.5% in the present experiment, which is still about 0.3 percentage points.

8.5.3.8. Evaluation Sensitivity Results

As has been stated before, the lack of reliable information with respect to some of the system parameters limits qualitative conclusions, even though we feel that the parameters we have chosen are broadly correct. We can conclude that the model is most sensitive for changes in parameters that affect the rate of accumulation of human capital, but also of that of diversity. This was to be expected, as these are generally considered the most important productivity growth transmission channels in endogenous growth theory. We have sometimes referred to network-effects, underlining the potential non-linearity of the growth effects of equal changes in system parameters. This can be shown by comparing the outcomes of experiments 5 and 6 to the base-run outcomes. The importance of ICT-capital in the number of FLOSS varieties (ψ_1^F) rises from 0.1 to 0.2 to 0.3, respectively, in experiments 5 and 6, whereas the other parameters remain unchanged. We see that the growth rate of output rises from 2.2% to 2.5% and then to 3.0%, i.e. between experiment changes of 0.3 percentage points in between the base-run and experiment 5 and 0.4 percentage points between experiments 5 and 6, suggesting that there is indeed a non-linear (convex) relation between the size of the parameter change and the size of the growth response.

Finally, the effect of a combination of asymmetries between FLOSS and PROPS groups and a change in the relative investment shares of these groups (signalled by w), can be inferred by comparing experiments 6 and 7. We see that a rise in importance of the FLOSS group, indicted by a drop in w from a value of 0.8 to 0.7, does indeed bring about additional growth. However, the effect is very small, i.e. of the order of 1/100 of a percentage point, again underlining the importance of the way FLOSS may be expected to impact on the 'primary' growth channels known from endogenous growth theory, rather than on capital accumulation per se.

8.5.4. Scenario analysis

Given the limited quality of the data available, the only scenario that seems to make sense to us is to see what a rise in the investment share in FLOSS in Europe would bring as compared to the U.S. There are several ways of doing so, depending on the imposed reactions of PROPS investments; FLOSS investments can be increased leaving other investments unchanged; or, FLOSS investments can be increased and PROPS investments decreased at the same time. The latter can be thought of cannibalisation of FLOSS versus PROPS that can occur to the full extent or partially. Here, we take the rather cautious point of departure by assuming full cannibalisation of PROPS by FLOSS. Partial cannibalisation would increase total investments and thus would lead to larger effects on economic growth. So in that sense we present a lower bound of possible effects. One should note that if there are no differences in the parameters, i.e. if the effects of ICT capital and of human capital on the number of varieties are the same for PROPS and FLOSS and if the quality/price ration is equal to one, there will be no difference between FLOSS and PROPS so changing the shares of investment from 20% and 80% for FLOSS and PROPS respectively (the 20-80 case) to e.g. 30% and 70% (the 30-70 case) will have no effect, at least in principle. The only effect that can occur is due to the differences in the levels of stocks of (software) capital due to different investment levels in the past. If stocks are different, an effect that is equal in size in absolute terms will have different effects in relative terms and thus can have different effects on final growth rates of the economy.

The scenario analysis is carried out as follows: we run a base simulation given a set of parameters as described above and we run a simulation where the investments shares in FLOSS and PROPS change. By comparing both runs, we determine the effect of that change. This is done for differences in a single parameter, i.e. FLOSS and PROPS only differ in one dimension, and in combinations of these dimensions. Finally, we carried out two simulations for each parameter constellation to see whether changes in investments shares are linear in their effects. As above, the base run assumes a share of 20% for FLOSS and 80% for PROPS and we have changed these shares to 30%-70% and to 40%-60%. The latter is done to check whether effects are (nearly) linear or not.

Table 34 presents the results of the scenario analysis. Experiment 8a and 8b show the results where there are no differences between FLOSS and PROPS except initial investment shares. So a change from the 20-80 case to the 30-70 case increases the growth rate of final output (and thus of productivity) by 0.08%. This is due to the difference between the absolute increase on investments and relative increase if levels are different. Note that this difference does not depend on the change in shares of FLOSS and PROPS. In both the 30:70 case (experiment 8a) and the 40-60 case (experiment 8b) the effects are the same.[189] Experiments 9a and 9b shows the differences relative to their own base run where the quality/price difference between FLOSS and PROPS is equal to 10%, i.e. parameter *q* is equal to 1.1. For a relative small change from the 20-80 case to the 30-70 case there is no (visible) difference in

[189] It is easy to show that ratio of relative changes for a given absolute change, depends only on the initial levels and not on the size of the absolute change. (Example: A given absolute change on initial levels of e.g. 100 respectively 20 leads to a relative change that is a factor 100/20=5 times higher for the initial level of 20 as compared to the initial level of 100 and this factor does not depend on the size of the absolute change)

the sense that the effect on economic growth is 0.08%, which is the same as we had before. However, the difference becomes more apparent if we increase FLOSS further towards 40% of total software investments (and decrease PROPS to 60%). The difference in annual growth rates of production (and thus for labour productivity) now becomes 0.08% per year.

In the case the number of FLOSS based varieties is more sensitive to the ICT capital stock, i.e. in the case ψ_1^F is larger, the effect of an increase in FLOSS investments is clearly larger and the growth rate now jumps to 0.21% compared to its own base run (see experiment 10). This effect is also in the 30:70 case much larger than the level effect as shown in experiment 8a. The FLOSS based number of varieties ($\hat{A}^F$) shows a higher growth rate that turns into a higher growth rate of the effective capital stock. The effect of the 40-60 case on growth of output is double, i.e. 0.42% (not shown in Table). Finally, experiment 11 shows the effect of an increase in FLOSS investments if the relative contribution of ICT capital to human capital formation is changed. $1-\gamma$ denotes the contribution of ICT capital in the knowledge generating process so we should decrease γ in order to increase the contribution of ICT capital in knowledge generation. Experiment 11 shows that this has absolutely no effect on the growth rate of output (except the level effect as discussed in experiment 8a). This is, however, very natural since there is no difference between FLOSS and PROPS in this particular simulation experiment so increasing FLOSS and decreasing PROPS has no effect at all. This implies that –in case of full cannibalisation– the sensitivity of the growth rate of human capital on ICT capital has an effect on output growth if the nature of FLOSS differs from PROPS in more dimensions, i.e. if we combine experiments as described above.

Experiments 12 to 15 show such combined effects. If we combine experiment 9a and 11, which is done in experiment 12, we see basically no difference as compared to the initial experiments. The difference here again becomes only visible if we increase investments in FLOSS to 40%, as was the case in experiment 9b. Experiment 13 combines 9a and 10 and shows a slightly larger impact as was the case in experiment 10. Experiment 14 combines 10 and 11 and also here we see no (visible) additional contribution of the importance of ICT capital in human capital formation. The effects are simply too small to measure. Finally experiment 15 combines the three effects (9a, 10 and 11) and shows also here considerable effects on economic growth, comparable to experiment 10.

So the changes are very small, almost zero, if the relative contribution of ICT capital in human capital formation is increased. Since we did not impose any difference between FLOSS and PROPS in this respect and since we simulate the model under the assumption of full cannibalisation this result is expected. The effect is a bit larger if the quality/price difference of FLOSS compared to PROPS is introduced but also here we find small effects unless FLOSS investments increase to 40% of total software investments. Finally, if FLOSS has a larger effect of the number of varieties as compared to PROPS, the effect of a change in the investment share of FLOSS has a considerable effect on output growth. Combining the assumptions does not bring additional growth. Finally, one might consider such increases as displayed in Table 34 (in the order of e.g. 0.1%) as being minor but one should realize its size when translated in absolute figures. Moreover, the changes reported here are in growth rates that give rise to permanent effects on levels. If we translate this in output in Euro, an extra growth in output of 0.1% is equal to a bit more than Euro 10 billion per year in 2006. Since the effects of software on economic growth (and productivity growth) are in growth rates, the level effects are even more sizeable after some years, even if the extra impulse of extra FLOSS investments would be temporary.

Table 34: Simulation results for Europe in case of an increase in FLOSS at the cost of PROPS (one year after change)

	Exp 8a	Exp 8b	Exp 9a	Exp 9b	Exp 10	Exp 11	Exp 12	Exp 13	Exp 14	Exp 15
% of software investments in floss	30%	40%	30%	40%	30%	30%	30%	30%	30%	30%
% of software investments in props	70%	60%	70%	60%	70%	70%	70%	70%	70%	70%
Relative contribution of ICT-capital in human capital formation (γ)	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.9	0.8	0.8
Quality of FLOSS ($\overline{q}$)	1.0	1.0	1.1	1.1	1	1	1.1	1.1	1	1.1
Contribution of ICT-capital in FLOSS based number of varieties (ψ_1^F)	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2	0.2	0.2
Results: changes in growth rates compared to base runs:										
FLOSS based number of varieties ($\hat{A}^F$)	0.33%	0.33%	0.33%	0.67%	0.68%	0.34%	0.34%	0.69%	0.70%	0.70%
PROPS based number of varieties ($\hat{A}^P$)	-0.08%	-0.08%	-0.08%	-0.16%	-0.08%	-0.08%	-0.08%	-0.08%	-0.09%	-0.09%
Human capital ($\hat{h}$)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Effective capital stock ($\hat{K}^e$)	0.13%	0.13%	0.14%	0.28%	0.35%	0.13%	0.14%	0.37%	0.36%	0.38%
ICT-capital stock ($\hat{K}_i$)	0.00%	0.00%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.02%
Output and labour productivity ($\hat{Y}$)	0.08%	0.08%	0.08%	0.17%	0.21%	0.08%	0.08%	0.22%	0.21%	0.23%

Note: All changes are relative to the base run for that parameter constellation. This implies that the results of experiment 8a and 8b, where FLOSS investments increased to 30% and 40%, respectively, is compared with a base run where the floss investment rate is 20% and where all other parameters remained unchanged. So experiments 9a and 9b are compared with their "own" base run, experiment 10 with its own base run etcetera. Since all reactions appeared to be linear for the 30-70 share of FLOSS versus PROPS and the 40-60 cases, only the 30%-70% cases are presented as from experiment 10 onwards.

When comparing these results, the aggregate investment rate is still unchanged at 0.15 since we assume a full cannibalisation of PROPS by FLOSS, so for every increase in FLOSS we assume an equal decrease in PROPS. This is clearly overly pessimistic in terms of the effect of FLOSS on PROPS and reflects a conservative approach. And even though, according to Solow, a change in the investment rate normally brings about just a level effect that peters out in the end, the actual time at which this happens can be quite far away, whereas the short- and medium-term effects can be quite outspoken. Moreover, temporary changes in the growth rate relative to the steady state growth rate generally tend to disappear asymptotically. That implies that there is no tendency to overshoot the steady state, but rather to gradually approach the steady state from either above or below, never quite reaching it, but ever getting closer to it. This implies that the net positive growth effects that we get in this scenario, while being of the order of a tenth of a percentage point at maximum, must ultimately be a lower limit to the actual growth effects we can expect from raising the rate of investment in software for a given value of the rate of investment in ordinary physical capital. Nonetheless, the long-term effects remain small but nevertheless sizeable if we compare them with the effects of total investments in software. The order of magnitude has been the case in all experiments that we have performed (and there have been many more than we have reported here). Hence, we conclude that our experiment seem to point out that the strength of FLOSS may be primarily in its ability to support the various growth-engines of the economy, in the form of human capital formation and increases in product and service variety, rather than being a true source of growth of its own.

8.5.5. Conclusions

In this section we have described the construction of a growth model that allows for the standard motors of growth as distinguished by endogenous growth theory, and for FLOSS, PROPS and ICT in general as complementary factors to these growth engines. These growth engines are the accumulation of human capital, and the increased quality of the match between supply and demand for goods and services through increased variety in these goods and services. The latter generates either higher utility (hence value) in the case of final products/ services or greater productivity in the case of greater variety in intermediate goods and services. Generally speaking, ICT may be thought to influence the growth process through its impact on the productivity of the human capital accumulation process, but also through its impact on the creation of more varieties directly. Especially the latter seems to be a feature especially relevant to FLOSS, as it is its adaptability to the special needs and circumstances of its users as well as the speed of adaptation relative to PROPS that are a primary incentive for the adoption of FLOSS. But also the relatively low cost of adoption is important for its users.

We have constructed an endogenous growth model, based on multi-level nested Ethier production functions, while borrowing Solow's behavioural approach, implying that we have used exogenously fixed investment rates, rather than a fully consistent inter-temporal optimisation problem, as is standard practice in endogenous growth theory. Such a fully consistent inter-temporal optimisation framework forces one to economise on real content, in favour of analytical consistency, and in this case we preferred the real content over the consistency, as we needed to combine a number of different growth engines, and growth supporting FLOSS and PROPS activities, without endangering the ability of the model to come up with non-trivial results. Given the behavioural approach in this case did not allow us to obtain analytical solutions, we have used simulation exercises to illustrate the working of the model, and the kind of results one could expect to obtain.

To this end we have tried to guesstimate some of the lesser-known or even unknown structural parameters of the model, while calibrating others. Then we performed a sensitivity analysis, to find out about the signs and orders of magnitude of the growth effects associated with changes in the various system parameters. We found that the most important parameters influencing aggregate productivity growth results are those of the importance of ICT-capital in the human capital accumulation process and the parameters that reflect the importance of human capital and ICT-capital in the generation of new varieties. Because of the way in which FLOSS is organised, through communities that contribute to a joint FLOSS product, it is almost impossible **not** to have positive knowledge spillovers when using FLOSS. These spillovers may be software product specific, but they may also lead to a higher ICT-literacy, thus facilitating the absorption and sharing of new knowledge. The parameters that reflect the importance of human capital and ICT-capital in the generation of new varieties have to do with the link between software use and the ability to create variety in services and products. That ability is bigger with FLOSS than with PROPS, but it still depends primarily on the availability of human capital, aided by FLOSS and/or PROPS. In our sensitivity analysis, we have concluded that changes in these parameters do indeed lead to significant changes in growth performance, indicating that LOV as one of the main mechanisms of endogenous growth may indeed provide opportunities for FLOSS-based additional growth. Finally, we have performed an experiment in which we have changed the composition of investment in Europe in favour of FLOSS, and, pessimistically assuming full cannibalisation, at the cost of PROPS. While the results were positive with respect to growth in all cases, the effects become more sizeable the more important ICT is in human capital formation. In this respect, one can even argue that the impact of FLOSS might be larger since software development (as part of R&D) is more oriented towards development of new applications, based on common and public knowledge of existing applications. This would increase the productivity of human capital formation even further leading to a higher impact on productivity and output growth.

Our results are suggestive of the overriding importance of human capital formation in this set-up, and especially the way in which FLOSS can directly and positively influence the speed at which contributors to FLOSS communities can pick-up new knowledge and put that to good learning use, for themselves but also for the more down to earth users of FLOSS software, certainly if the latter would be geared at the design of free access ICT-based learning environments. Finally, it should be obvious from the above that more insights in the underlying processes are needed to give more precise and more conclusive figures on the economic impact of FLOSS versus PROPS. This, however, also requires more and better data.

9. Trends, scenarios and public policy strategies

MERIT's annual surveys in 2003 and 2004 of all Dutch public authorities and their attitudes towards open standards and open source software carried out for the Dutch government had some striking results. The results are corroborated by the FLOSSPOLS survey of 955 public administrations across 13 EU member states. In particular, the survey results showed a strong sense of protest among users against proprietary vendors charging licence fees that were seen as "too high" (by 61% of respondents) together with a sense of helplessness at being unable to control their own future technology choices due to vendor lock-in. This is neither limited to the public sector nor new – the 2002 FLOSS User Survey of firms and public sector organisations found that 30% "totally" and a further 26% "somewhat" agreed that independence from pricing and licensing policies of software companies was a reason to use FLOSS.

The strategic areas are numerous – in particular, issues relating to e-government, interoperability and transparency and public-sector support for open source. Many countries or regions have developed direct policies to encourage the usage or production of FLOSS, sometimes expressing them as preference in some application domains and for a variety of reasons (vendor independence, transparency towards citizens, costs, security, educational benefits and economic regeneration). Examples include, for instance, policy-decided migration to FLOSS office suites in France, commissioning and adoption of FLOSS security, communication and groupware software in Germany, or general adoption and support to business adoption in the Spanish region of Extremadura. The European Parliament has also voted (by 367 for, 159 against and 39 abstaining) resolution A5-0264/2001 that *"Calls on the Commission and Member States to promote software projects whose source text is made public (open-source software), as this is the only way of guaranteeing that no backdoors are built into programmes; Calls on the Commission to lay down a standard for the level of security of e-mail software packages, placing those packages whose source code has not been made public in the 'least reliable' category;*. There are several arguments relating to the specific nature of certain technologies, especially software for basic infrastructure, where it may be against the public interest to be dependent on specific vendors. In addition, many situations today, the public sector engages in its own software development (in-house, or externally commissioned), and there is a question whether such software should not be made, as a matter of principle, open source, for reasons of transparency and of fairness (so users – citizens – are not forced to use compatible software from specific vendors in order to interact with public sector software applications). Opponents to policies that explicitly support use of FLOSS have criticised such policies as a preferential technology choice, equating it to previous – and sometimes ill-fated – policy preferences regarding technology. However, FLOSS is not in itself a technology and exhibits, on the contrary, a wide variety of technology directions.

In several countries, governments have published policy frameworks and guides highlighting how administrations can, when it is judged useful, require that software is provided under a licence that includes the freedom to use, copy, modify and redistribute, just as administrations can define any other aspect of the software they procure. When arguing for procurement based on needs, it should not be assumed that needs are only technical; public tenders frequently include clauses on ownership of purchases and specify licensing conditions. This approach is followed in the section on procurement of "Guide for the choice and use of OS licences for public Administrations" published by the French government IT agency[190], and the procurement guide published by the ICT Unit of the Dutch government[191]. These frameworks make obvious procurement requirements specifying properties of the licence – without necessarily asking for FLOSS solutions. This does not represent a technology preference, but a requirement for activities that can be performed with the software. In addition, it is also clear that services connected to FLOSS software and solutions can also be tendered, just like it would be possible for proprietary software solutions. In the same countries and at European level, these initiatives have been complemented by the creation of competence centres and initiatives for pooling FLOSS developments among administrations, sometimes with the participation of local government associations (ADULLACT in France) or the private sector (BerliOS in Germany). In Germany, the government directly procured the development of FLOSS components whose absence limited the possible deployment of FLOSS within administrations. The German government procured FLOSS solutions in the field of cryptography integration with email and groupware[192]. For all these initiatives, the FLOSS-related initiatives of the IDA and IDABC programmes of the European Commission's DG Enterprise have been an arena of exchange of experience and coordination of best practice.

A further strategic area is in EC-funded research, where there is an open question regarding the nature of protections granted to IP, especially software, being developed as a result of public funds. The DG-Research Expert Group on "Strategic Use and Adaptation of Intellectual Property Rights Systems in Information and Communications Technologies-based Research" in its report recommended that open source software licenses be required in several cases where publicly funded research resulted in the development of software, especially in connection with the development of standards.

There are also broader issues related to the efficiency, innovative development and productivity of software (as described in previous sections).

From the policy-maker's standpoint, major advantages are associated with the FLOSS methodology of development and release. First, FLOSS can be obtained at (almost) zero-cost by the vast majority of users, through direct downloading of installation packages or as a "crude" source code from the Internet. Further, as the development of Open Source projects is continuous and its progress depend on attracting skilful programmers, preference for a

---

190 Le guide de choix et d'usage des licences de logiciels libres pour les administrations (2002), http://synergies.modernisation.gouv.fr/IMG/pdf/Guide_LLL-2.pdf

191 http://www.ososs.nl/attachment.db?6946

192 The Aegypten/Spinx and Kolab projects funded by the German government's BSI (cyber-security unit) paid for the development of software for secure communications and collaborative workspaces, released under GPL licences and further supported by the FLOSS developer community after the funding for development ended. See the case study on the European *Commission's Open Source Observatory*, Open Source Adoption of the German Federal Office for Information Security at http://ec.europa.eu/idabc/en/document/4492/470.

popular FLOSS application provides an ongoing support of the project community of developers and users.

Second, dominant software producers release their products to the market, by definition, at a monopoly price. FLOSS applications, on the other hand, can be acquired at no cost or can be purchased at a competitive price, lower than the cost of proprietary software, from FLOSS vendors (since, potentially, every user can download FLOSS programs and distribute them at any cost, their cost equals the competitive price). In terms of social welfare, FLOSS may eliminate the loss for users who purchase their application from a vendor and for others— it removes the need to invest directly in licensing/purchasing software products. Those savings may be substantial for SMEs and can be re-allocated to other operations of the firm.

Third, FLOSS may foster development of new software projects including proprietary applications on the basis of the FLOSS platform and, more broadly, supports the growth of ICT and service sectors that are established upon extensive use of data management and computer systems. Source code disclosure enables programmers to learn from it and to utilize it in advanced applications. The accessibility to the source code also provides essential information on interfaces and standards, thus enabling proper compatibility and interoperability among software platforms and applications.

Fourth, FLOSS software offers alternative technological solutions to those provided by proprietary software. While incumbent proprietary software players may perceive FLOSS as a threat, for the social policy planner and for the regulator it may encompass rather valuable merits. Following the evolution of the software market and the lessons learnt from anti-competitive behaviour of monopolies (e.g. the legal proceedings against Microsoft in the U.S. and in Europe), we suggest that the interpretation of “free market” should be geared towards free competition in order to lower monopolistic dominance and to a limited intervention of anti-trust agencies, when necessary. Anti-competitive practices by incumbent firms can lead to the survival of technically-inferior technologies and products (Gallini and Trebilcock, 1998), and given the rapid changes in the software market it is better to ensure full competition *a priori,* rather than attempt to regulate through anti-trust intervention *ex post facto* by which time it may be too late for competitive, innovative technologies and firms. .The development of ubiquitous computing and ambient intelligence, for instance, as FLOSS technologies, especially when undertaken by publicly funded consortia or by industry-funded committees (similar to the industry-funded, publicly accessible SNPS bioinformatics consortium referred to previously), can ensure that the new technology will become a “public common” that is accessible to all industry partners and to new entrants, which in turn furthers it for the benefit of industrial and technological development and for their users. Moreover, applying the FLOSS model could prevent scenarios in which essential components of the technology are “hijacked” and dominated by a single firm through proprietary development and strategic patenting (Granstrand, 1999).

Finally, public policy may target not only use and development of FLOSS within administrations and public bodies or to businesses through economic regeneration projects (Extramadura in Spain, West Midlands in the UK, Soissons in France, for instance), but also its widespread societal promotion through education and cultural policy. As illustrated in Section 8 on the indirect impact of FLOSS on innovation and growth, and Section 7.4 on skills and employment, it is those developments in the information society that have created massive usage by individuals and small organisations for the creation of documents (Web sites, blogs, wikis – e.g. Wikipedia, the free, volunteer-created and collaboratively edited online encyclopaedia available in 195 languages is now far larger than Britannica with 1.5

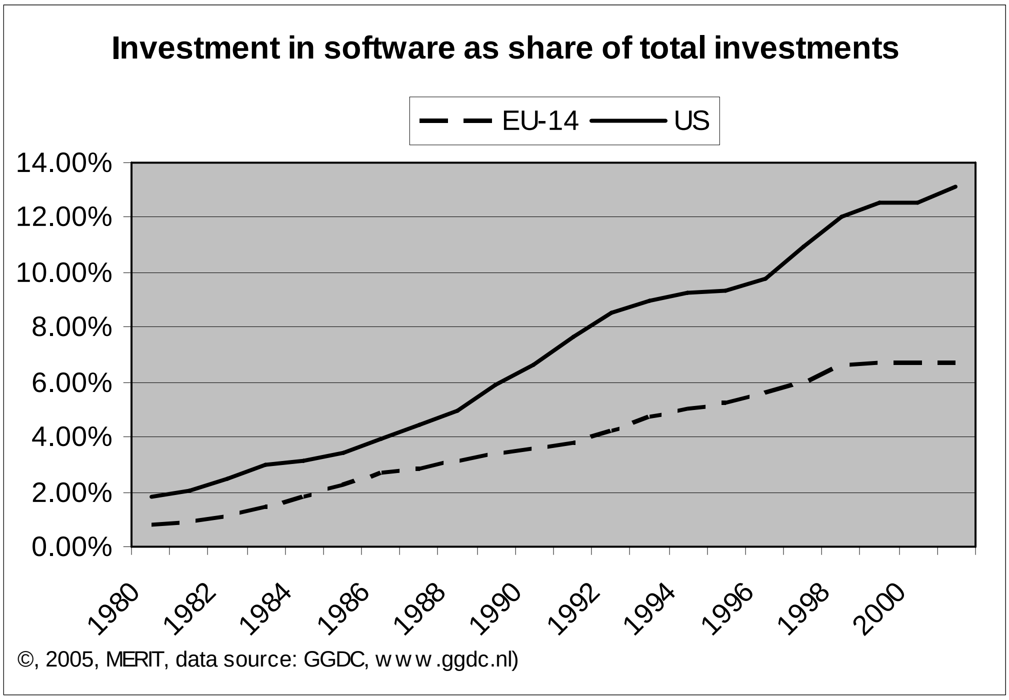
million articles) that seem to have a high potential of indirect economic impact on the ICT sector and general job creation, as well as the possibility of increasing and retaining highly skilled labour within Europe.

The production and use of FLOSS applications demonstrate how economic mechanisms underlying the software market are far more complex and distinct from the commonplace economic rationale. Several phenomena, such as the evolution and the dynamics of communities of developers, programmers that provide their programming skills to FLOSS projects without direct reward and increasing number of software firms that join the FLOSS movement, are rather striking. However, the decision of several commercial firms to strategically prefer FLOSS-based dissemination of their products and disclosure of their core assets (i.e. the source code) over the development and retailing of proprietary programs is not straightforward and, to a large extent, depends on the structure and the conditions of the market and on the economic peculiarities of software technologies (Behlendorf, 1999; Evans and Reddy, 2003).

9.1. Trends: use of software in Europe and the US

As is shown in Figure 59, US based firms invest much more in software as compared to their European competitors. In 2001, software investments accounted for about 13% of all US investments whereas the share of software investments was only 7% in Europe. However, as explained in e.g. Hollanders and Meijers (2001) (a MERIT report to the European Commission within the NewKInd New Indicators for the Knowledge Based Economy Project) only a limited fraction of own account software is accounted as software investments which underestimates actual investments. In the same report, we show by using Dutch figures that own account software accounts is about 55% of purchased software which implies that the official figures understate the actual use of software. Moreover, there are strong indications that own account and reproduced software is capitalized more in the official US statistics as compared to the European ones such that the gap between the US and Europe is smaller than the official figures reveal. (See e.g. Oulton (2001) and Ahmad (2003))

Figure 59: Software investments in the EU and the US (1980-2001)[193]



In a survey among 625 European firms, IDC finds that open source software has a significant level of diffusion. For instance, more than 30% of the respondents make significant use of open source database management systems and another 15% declare some use of these systems. In case of operating systems, the figures are 25% and 10% respectively. Similar figures were identified in the FLOSS User survey of 1492 European firms in MERIT's FP5 project and the FLOSSPOLS survey of European government users. This implies that the investment figures in software as presented in the official statistics understate the actual use of software, as FLOSS software use is not (yet) counted within national

[193] Both total investments and investments in software are in constant prices. Software prices are adjusted for quality changes according to hedonic price deflators. See e.g. Hollanders and Meijers (2001) for an analysis of quality adjustments on software investments.

accounts for software expenditure. Since some evidence indicates that European firms invest more in FLOSS, as compared to the US, the difference between the two regions becomes even smaller.

Based on the official figures as presented above, evidence originating from studies like Ahmad (2003) and Oulton (2001), data on Linux-based server and PC sales from IDC and Gartner, we provide some rough estimates on true software investment figures in the EU and the US in Figure 60, with projections until 2010. The baseline figures for US and EU investment (data points are circles) are extrapolations of the data on software investment as a share of GDP as depicted in Figure 56 on page 152. The lines plotted as triangles show the estimates for software investment if the installed base of FLOSS software is valued as an investment in the same way as proprietary software. Since investment measures do not usually include software for which licence fees are not paid[194], we estimate the notional value of FLOSS investment based on the relationship between software investment to hardware (servers and PCs), and the proportion of hardware running Linux. This may underestimate the notional value of FLOSS investment since it does not count the large (but essentially uncountable) number of free downloads of FLOSS software. The use of such downloaded software should ideally be included in notional software investment figures, especially for comparisons between the EU and US, as it would lead to productivity effects. However, since we use the notional value of FLOSS investment in order to estimate other related economic activities in this report (such as the FLOSS-related share of IT services), and much of the "freely downloaded" FLOSS may not result in spending on such things as IT services, we conservatively limit our estimates here to FLOSS use driven by Linux hardware sales.

Nevertheless, the increase in software investment as a share of GDP is significant, both for the EU and the US, amounting to over 0.2% of GDP (0.4% in the US)[195].

This amount is increasing with the increasing share of FLOSS servers and PCs sold, as can be seen in the solid lines in the figure (EU/US "incl FLOSS"). The estimated investment value of the FLOSS software currently in use (2006) is Euro 22 billion in Europe and Euro 36 billion in the US, representing 20.5% and 20% of total software investment, respectively. This is projected to rise by 2010 to Euro 39 billion in Europe and Euro 59 billion in the US (at current exchange rates) equivalent to 31% and 27% of total software investment, respectively.

Despite slightly greater FLOSS use in Europe on PCs, the software investment gap with the US does not decrease when including FLOSS, due to the higher rate of FLOSS server use in the US[196]. The dashed line ("EU: more FLOSS") projects what could happen if the share of FLOSS-based servers increased by a factor of 1.5 above current market share projections[197], and the share of FLOSS-based PCs increased by a factor of 2. Of course, we assume that US FLOSS usage rates continue to grow following current projections, thus

[194] But see footnote 186 on page 159.

[195] As the FLOSS investment value is not measured in accounts at all, it is not included in the GDP figures either; so this ratio requires adding the FLOSS value to GDP before calculating the ratios.

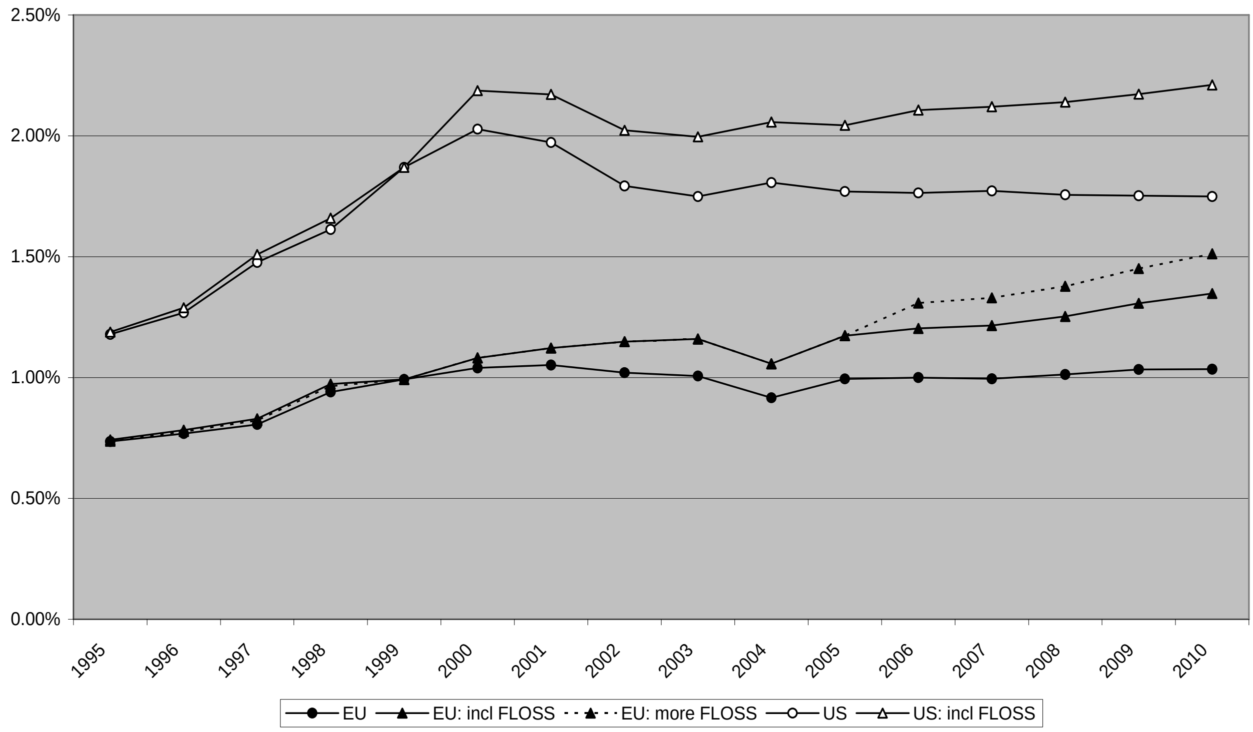
[196] This is partly a result of our estimation method, which values software on a server much higher than software on a PC. However, changing the PC/server effect in our estimation method does not significantly affect the EU-US investment gap.

[197] Extended projections by MERIT based on data from IDC, 2004, "The Linux Marketplace - Moving From Niche to Mainstream", prepared for OSDL.

allowing Europe to catch up somewhat. In this scenario, the value of FLOSS investments in Europe in 2010 will be 29% of that of other software investment, and will be in *addition* to such software investment.

It must be emphasised that the "non-FLOSS" software investment figures here do not refer exclusively (or even primarily) to packaged proprietary software. Proprietary packaged software accounts for a relatively small share of total software (see e.g. Table 24, "The software economy: sales, services and in-house" on page 124). However, we cannot easily conclude that proprietary packaged software accounts for shares of software investment *in official accounts, as shown here* in relation to its share of software spending. This is because different national statistics bodies have different methods of accounting for software investments, and one key area of difficulty – and difference – is accounting for the large share (in the US, the plurality) of software that is in-house or own account software. In the US, which may have the most complete accounting for in-house software, it is likely that the share of proprietary software in investment is similar to the share of proprietary software in spending, i.e. about 16% - similar to our estimate of the share of FLOSS in total software investment. In Europe, where in-house software may be less well accounted for, the share of packaged proprietary software in the figures here may appear higher, because in-house software is not fully included[198].

Figure 60: Estimated "true" software investment, share of GDP



Copyright © 2006 MERIT. Source: Investment data fromGGDC; Linux hardware data and projections from IDC "Linux Marketplace" study; GDP projections Eurostat and US Congressional Budget Office; Software and hardware sales data from EITO; MERIT estimates and projections.

The previous analysis has not assumed a reduction in the absolute value of the purchased software investment. Taking such a reduction into account is difficult, as it is not

[198] See footnote 144 for references to further discussion of in-house software and national accounts.

clear whether savings on software licence fees are going to reduce software expenditures. If so, this will be a gradual process - after all, the current EU software investment in FLOSS is in addition to the total estimated software investment of Euro 107 billion. But even if licence fee savings are not absorbed by investors they may be significantly diverted into services, or in-house software – especially in the US, with an already high share of in-house software and the in-house skills for software development among user organisations[199]. An increase of in-house software due to increased FLOSS use result in an apparent lowering of the software investment share of GDP, especially in Europe, as will an increase of software services as a result of greater FLOSS use – unless the value of FLOSS software is taken into account.

9.1.1. Implications of FLOSS investment

The increasing use of FLOSS has two significant implications for policy makers and governments. One is for national statistics bodies: unless a way is found to account for FLOSS software investments, the fact that it is increasingly substituting for investment in licence-paid software will result in an apparent, but false, decline in software investment figures. This has serious implications for growth and policy analysis, and may be particularly relevant for growth accounting studies that compare regions (such as the US and EU, but potentially much more importantly, high FLOSS use countries such as China).

A second implication is that the above analysis demonstrates that FLOSS can significantly drive economic growth. As Table 31 on page 151 shows, the network effects of software are a significant factor in labour productivity growth and explain in particular some of the difference between the growth patterns in the US and EU. Since FLOSS demonstrates very large network effects and greater FLOSS use in Europe can help narrow the software investment gap with the US, policy makers should seriously consider the positive implications of supporting – and not hampering – the trend towards FLOSS take-up in Europe.

[199] See section 7.4.5, “FLOSS and software employment”, especially Table 16: US software development and support jobs by sector, 2005 on page 78 and the associated discussion.

9.2. SWOT analysis of ICT industry and FLOSS

For this task of developing future trends and scenarios, we propose to highlight the particular dynamics of FLOSS products in the ICT market and the resulting impact on innovation and economic competitiveness. Following the previous techno-economic analysis of markets for software and information technologies, we construct a SWOT analysis that assesses the appropriateness of integrating FLOSS development and dissemination as a part of business strategies in the ICT industry. The analysis examines the strengths and weaknesses of the European ICT industry in adopting FLOSS-based strategies, as well as the opportunities and threats to such strategies, according to the following economic attributes:

1. Strong/weak degree of network externalities.
2. Monopolistic dominance or competitive market structure.
3. Demand for complementary products and services to non-proprietary software.
4. Rapid diffusion of the technology and formation of *de facto* standard adopted by the majority of the market.

Although many FLOSS applications are initiated as individual projects, ICT firms often choose to distribute their products as *hybrids* in which part of the source code remains proprietary and the other is made open (McKelvey, 2001). In particular, large sections of the ICT industry are increasingly using FLOSS for *embedded* applications (the Linux kernel is at the core of most routers and a large range of telecoms and industrial instrumentation equipment). In this case the hybrid model adopted involves free release with FLOSS software tightly coupled with proprietary hardware as a profit generator.

The FP5 FLOSS Final Report (part 3: “Open Source Software Markets and Business Models”) included a comparative study of business models practised by large players in the FLOSS market in the EU and abroad, which has been summarised in the previous section 7.5, “New businesses, business models and benefits for SMEs”, following page 82.

The SWOT analysis below relies on findings presented in the previous sections of this study. It focuses on strength and weaknesses of the EU economy in terms of its ability to use FLOSS and derive economic growth and innovation from it, with a particular attention to differences and similarities between the EU and the US. Based on the more detailed analysis presented in the sections “scenario analysis” and “wider impact of FLOSS”, it identifies opportunities and threats for the EU economy capacity to derive economic and innovation benefits from FLOSS. The policy strategies in section 9.5 are partly derived from this SWOT analysis.

The SWOT analysis is presented in two parts. First, the most important strengths, weaknesses, opportunities and threats are highlighted for the European ICT industry to achieve its full growth potential using FLOSS.

Then a more detailed SWOT table is presented.

9.2.1. Strengths for the European ICT industry adopting FLOSS

1. Europe’s ICT industry has a limited reliance on proprietary software licensing, with a much higher focus on services, customisation and integration, as well as a large secondary software sector. This puts Europe in a strong position to use and develop FLOSS.
2. Europe has a strong population of qualified contributors to FLOSS, and some unique experiences in the development of community projects (for instance KDE).
3. Europe's commitment to open standards and interoperability is a strength as this is a natural fit with FLOSS.
4. EU consumers and citizens tend to value political and ethical matters (fair trade, organic food, environmental quality). In the information economy FLOSS is often adopted in Europe for ethical reasons. This cultural characteristic of the EU domestic market is a clear strength for FLOSS-based ICT growth.

9.2.2. Weaknesses for the European ICT industry adopting FLOSS

1. Despite its strength in development of FLOSS code, Europe is relatively weak in leadership of FLOSS community projects, several of which are led by people from the US.
2. European ICT firms are not very closely involved with the FLOSS community, in contrast to many US firms.
3. Europe provides extensive R&D subsidies. While EU policy in theory supports SMEs (though not community participants), in practice these R&D subsidies disproportionately favour large ICT industry rather than smaller firms or FLOSS community participants, and discourage FLOSS-style business and licensing models[200].
4. Despite some exceptions, the European ICT industry has in general been slow to adopt FLOSS-based models for mutualisation of funding of development tools, much of which has been led by US firms.

9.2.3. Opportunities for European ICT industry adopting FLOSS

1. The skills improvement environment provided by FLOSS is an opportunity for the European ICT industry to draw on for human resources development, and supports the greater adoption and development of FLOSS.
2. The FLOSS contribution to the take up of new ICT activities (e.g. Web communication, blogging, digital photography) presents an opportunity for

[200] At least until the FP6 research framework programme, which prefers exclusive appropriation and exploitation of IPR resulting from publicly (partially) funded R&D.

the European ICT industry for higher value added software, hardware and services

3. Increased vendor independence resulting from FLOSS provides an opportunity for the European ICT industry to compete on a level playing field.

9.2.4. Threats for the European ICT industry adopting FLOSS

1. Software patents are perceived as a threat by FLOSS developers, and many small software firms in general. The threat is not specific to FLOSS, and has not prevented the development of FLOSS in the US – only since software patents have very rarely been enforced against FLOSS developers. A strong interoperability exception to enforcement would reduce much of this threat.
2. The current business models of large media publishers, relying on artificial restriction of supply by concentrating on a limited number of “star” titles, may result in continued legislative pressure on the ICT industry to adopt DRM and trusted computing technologies that do not interoperate with FLOSS. This can threaten the adoption of FLOSS by the ICT industry.
3. The value of FLOSS is often in its ability to encourage new services and new application domains from small providers (such as P2P/peer-to-peer); moves to erode network neutrality would threaten this.

Table 35: SWOT analysis of Europe and its ability to use FLOSS for deriving growth and competitiveness

Strengths	Weaknesses	Opportunities	Threats
The structure of the European ICT industry (strong internal development within systems integrators and device manufacturers, strong service industry, limited reliance on proprietary software licensing[201]) limits vulnerability to cannibalisation effects of FLOSS. This can give rise to a positive synergy between: - FLOSS approaches in devices, tools and platforms for services; - Industry provision of networked services, IT solutions for the non-ICT industry and interoperable devices. Europe's commitment to open standards and interoperability has a natural fit with FLOSS that often implement or contribute to the success of both. Europe has a strong population of qualified contributors to FLOSS, and some unique experiences in the development of community projects (for instance KDE). FLOSS encourages a reflexive attitude towards technology and collaborative approaches to problem solving. Europe's strong educational systems could further build on adopting FLOSS approaches, with a long-term benefit for the quality of the workforce and	Europe's earlier strong innovation in processes and productivity[202] was not accompanied with a similar investment in technology and services for personal ICT activities by end-users. Endorsement of ICT-based convergence was limited and the various media or device sectors remained relatively segmented. Process innovation for competitiveness reaches some limits in terms of social acceptability when the unavoidable disruptions are not compensated by a strong growth in new branches of activity (this weakness could turn into a strength if a strong adoption of FLOSS opened a European-specific path towards generic information society applications, that in turn would make possible a new round of productivity and quality-oriented innovation in goods and non-ICT services). Despite the strong European contribution to FLOSS development, the participation of European developers and companies in the strategic orientation of large FLOSS	A stronger development of FLOSS would ease the growth of secondary (non-ICT) service related economic activities by permitting innovation in these domains without too much investment in core software production or too much dependency on providers that capture an excessive share of the value chain (this might of course not be seen as an opportunity by these providers). In the context of a fast growing ecosystem of information exchange, FLOSS components open new business opportunities by promoting interoperability (open formats) and capturing the added value resulting from mass adoption (content tagging, rating, etc.). The emerging awareness of the potential for new commercial services serving non-	Established media companies appear reluctant to adapt to the emergence of non-commercial sharing usage. Their business model based on concentration of sales on a limited number of strongly promoted titles would require too disruptive an adaptation. The power exerted by these media companies on some information infrastructure companies (telcos, consumer electronics companies) and their influence on regulation expands the effects of their position. FLOSS-initiated technologies and forms of usage risk to be lobbied against by these players. FLOSS contributors perceive software patents as a major danger[203]. A heated debate

201 SAP and Business Objects are two significant exceptions. However the nature of their markets (software for organizations) and software could make possible for them to evolve in this respect. They value FLOSS tool platforms (SAP is one of the two “strategic users” of the Eclipse framework to which BO contributes). Nonetheless, the profit margins and capitalization associated with proprietary software licensing models make them very difficult to abandon voluntarily.

202 1980-1995.

203 With the exception of a limited number of very large companies who hold such large patent portfolios. In these companies (IBM being the key example, but Nokia providing a growing European example), the dual strategy of patent-based innovation in proprietary domains, and free licensing with retaliation clauses in the FLOSS

Strengths	**Weaknesses**	**Opportunities**	**Threats**
the competitiveness of the industry. In the field of emerging societal usage around content sharing (peer to peer and web 2.0 technologies), FLOSS is very well established and open source practices stands as the preferred way to efficient innovation (not EU specific, could turn to a weakness if Europe fails to invest in it). A growing part of the EU population has proven to be particularly attached to societal issues. As consumers and citizens they take into account these political and ethical matters (fair trade, organic food, environmental quality). In the information economy field FLOSS is often adopted in Europe for ethical reasons. This is also illustrated by surveys investigating individual FLOSS contributors' motivation. This cultural characteristic of the EU domestic market is a clear opportunity for FLOSS ICT-based growth. The past 20 years history of frequent failures in predefining which services will be successful for the wide creation and exchange of information calls for a more open experimentation approach. FLOSS products and methods are results of such an approach and now stand as good candidates to strengthen this open innovation strategy (not EU specific). One intrinsic advantage of FLOSS is that it provides a marketplace for buying services built around it. Clients are not limited to buy services from the product editor or the companies, which have been certified by that editor. The large European software service sector	projects remains limited (much lower than the share of code produced by European developers). Lack of large companies offering enterprise-level solutions, support and training on FLOSS prevent its adoption by large companies. This is particularly true for critical IT components such as database engines and information security, even though Europe holds strong positions in the corresponding FLOSS components. Limited availability and/or quality of IT devices and peripherals FLOSS drivers hinders the adoption of open-source technologies on these specific devices, in which Europe holds some strong positions. Innovation risks then to be limited by the lack of user initiative in fields where it appears crucial. Very few EU major ICT companies enter into strategic partnerships with FLOSS companies or community groups. In contrast the US companies (IBM, Sun, HP) have developed stronger policies in this respect. The diversity of open source licenses is perceived by non-ICT businesses and investors as giving rise to legal uncertainty and as a risk for interoperability. Although it	commercial information exchanges applications within telecommunications firms in Europe can lead to new adoption of FLOSS-based approaches. The primary business model built around FLOSS is service-based or result from the insertion of a FLOSS base into a physical product or service offer. New models are however possible such as the dual licensing adopted by European companies (Trolltech, MySQL) as well as in the US (Sleepycat, Red Hat). This can ease the business transition for some strong European software publishers. Europe has a strong tradition of putting in place schemes for sharing the funding of socially useful services. These schemes (legal licensing, flat-rate fees and levies, mutual funds for various types of insurance, mixed economy structures for NGOs and cooperatives active in the social or cultural domain, etc.) could enable a wide take-	has proceeded in Europe, where positions are now entrenched, without clear long-lasting outcome. There is likely to limit adoption of FLOSS by risk-adverse organisations and by some companies who are afraid of the need to manage a dual relationship to innovation incentives (patent-based in their physical inventions, FLOSS-based in software, when both activities are strongly linked). Hardware cryptography-DRMs represent a risk for FLOSS deployment. Compulsory DRM add the risk of a strong legal uncertainty and liability risks for FLOSS platforms. Wide DRM adoption would expose the market to a strong vertical segmentation by content providers or telcos that could lock-in user-driven innovation and the related FLOSS adoption. FLOSS has co evolved with a neutral, agnostic and

domain gains acceptance. However this strategy is more adapted to the opportunistic adoption or adaptation of existing FLOSS platforms (such as the GNU/Linux OS or the Apache Web server) than to new FLOSS innovation strategies.

Strengths	**Weaknesses**	**Opportunities**	**Threats**
could be more independent (of US proprietary software providers) and more innovative if it adopted FLOSS approaches. FLOSS-based innovation provides an entrepreneurial model that is more adapted to the European cultural environments than proprietary innovation. It could reconcile the strong humanitarian and ethical commitment in the European youth with technology innovation. Modularity, loosely coupled design, and strong adaptability are part of the very nature of FLOSS products. Hence, FLOSS components offer a good software groundwork for service related developments. This gives a strong position to a market such as Europe where customised solutions with strong and often local human support are valued.	has some ground today, this perception could change with the ongoing open source community work on licenses compatibility (The 2007 release of GPL version 3 license might address the long awaited Apache license compatibility). However the strategic involvement of large European ICT companies in these processes remains limited compared to the US. European and Member States research programs have growingly set for a predominant 50% shared cost funding. This funding model has up to now appeared to be insufficient in financing large strategic FLOSS projects developing a common basis for further innovation in services. FLOSS offers a model for the mutualisation of the funding of development tools among users. Europe has been shy in adopting this model, which is today mostly pushed by US-led consortia (even there are some strong initial European contributors such as Polarion or Jaluna, and companies like Ericsson were forerunners of such approaches).	up of a new generation of FLOSS-based information society applications. However the predominance of individual transaction and perfect market models in the standard economic reasoning limit the endorsement of these schemes, even though from a transaction cost viewpoint they are clearly superior in the information domain.	equitable to non-commercial contents Internet. This evolution is part of the synergy between FLOSS and innovation. Threats towards Internet neutrality (possibly reinforced by the ongoing IPv6 deployment) threaten the potential benefits of widespread FLOSS adoption. Part of the synergy between FLOSS and innovation is due to the technical excellence of FLOSS actors. The current crisis of entry in scientific and technological studies is then a serious threat to the continuation of production of innovative FLOSS products and related business opportunities. However this threat could turn into an opportunity if a strong endorsement of FLOSS ethics in science and technology took place.

9.3. Factors determining impact of FLOSS on the EU ICT market

FLOSS is not just about licenses. FLOSS licenses codify and enable processes of driving the orientation of technology from the periphery, from distributed innovators and users. But FLOSS also coevolves with architectures that are designed to empower actors at the periphery and that are generally based on FLOSS. The most notable examples are the architecture and protocol of the Internet, the standards and protocols of the Web, the architecture of new forms of peer-to-peer networks. This importance of the co-evolution between FLOSS and an information infrastructure that empowers innovators, users and economic players at the periphery has been emphasized for instance in Tim O'Reilly's keynote speech in the Georgetown Open Source Summit in 2002[204]. One key consequence is that one must not consider the development of FLOSS as a replacement of proprietary software by FLOSS within an unchanged information infrastructure, but as the enabling and development of a different information infrastructure for which few, if any, proprietary alternatives exist.

However, there has been little work on analysing the economic impact of this influence of FLOSS on the information infrastructure. This is paradoxical, since this impact has been strong for quite a while. The software for the Internet Protocol, all the reference implementations of the Web protocols, the availability of FLOSS reference implementations for a number of media codecs have played a key role in the major success of ICT-based growth. The Internet infrastructure itself, the core applications of email and file transfers and what they enable, the huge sphere of personal and business Web-applications are all FLOSS based. Every time an e-mail is sent, or a visit is made to an Internet website, FLOSS software-based protocols are being used. This is true even when the user interface (such as a web browser or an e-mail application) is not itself FLOSS – the underlying infrastructure almost always is FLOSS or, sometimes, a proprietary implementation of a tool originally implemented in FLOSS[205]. Even every day uses of commercial online services such as Google or Amazon are essentially uses of FLOSS – both services, like many other e-commerce websites make extensive use of FLOSS systems such as Linux and Apache.

A retrospective assessment of this impact (of the joint development FLOSS and open end-to-end general purpose information infrastructure) is not impossible. Reports have estimated up to 2.5 millions newly created jobs in the U.S. the early development of Internet/Web activities between 1994 and 1999[206] and similar estimates for Europe range from 1 to 2 million newly created jobs. It is difficult to single out the independent effect of the FLOSS aspects of the Internet /Web infrastructure, since the two are clearly interdependent .

The real question is: is there today a similar potential for future economic development based on the empowerment of a much greater number of information, service and technology providers, the development of associated ICT-related business (Web design

[204] See Tim O'Reilly, A Holistic View of Open Source, keynote speech at the Georgetown Open Source Summit, Georgetown Open Source Summit – Public Interest and Policy Issues (October 2002) report, pages 53-57, cf. http://opensource.georgetown.edu/report/osreport.pdf

[205] As with, for instance, the implementation of the Internet Protocol (TCP/IP stack) in Windows.

[206] University of Texas / Cisco study, http://newsroom.cisco.com/dlls/fspnisapia3b7.html

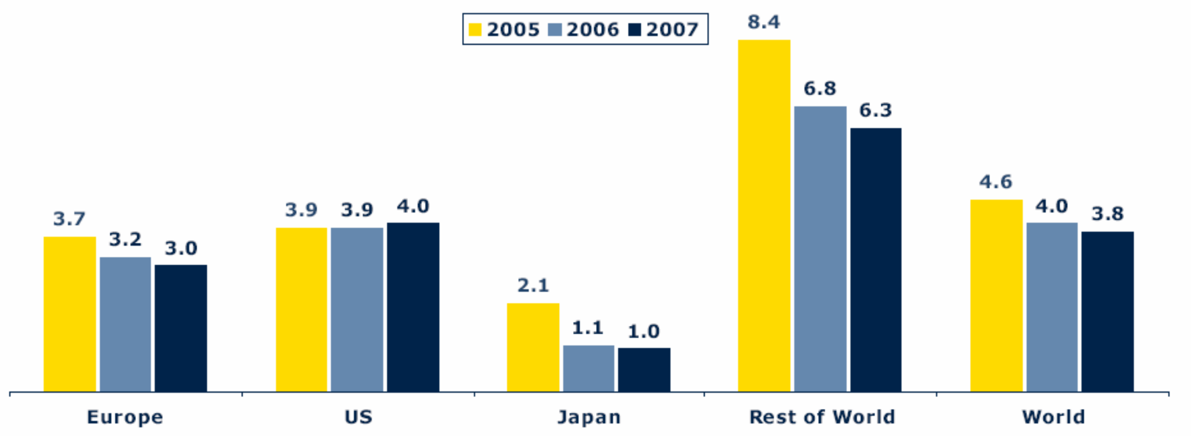
for instance in the past), non-ICT business (tourism, commerce, services to the person, B2B), and the providers of infrastructure (in the past: routers, Web software including proprietary) and infrastructure services (in the past: search engines, for instance)?

In a talk delivered at the European Internet Foundation just after the dotcom bubble crash, John Zysman of the Berkeley Roundtable on the International Economy declared that the “new economy” had not crashed, because it had never happened, most applications of dotcoms being of a substitutive nature in relation to the “old economy”, the core example being electronic commerce.

Can a new economy truly develop in the coming years? To give flesh to the technology, business, legal and cultural environment with which FLOSS and software will interact when trying to enable new applications and markets, we propose to chart it by identifying a number of key parameters each connected with 2 or 3 “options” that can be taken in a given domain. Each parameter is analysed in its relationship with FLOSS (both how it impacts FLOSS and how FLOSS impacts it) and with regard to differences between Europe and the U.S. The idea is that the analysis provided for each of these parameters will be immediately usable for reflecting on policy choices. A small number of scenarios can be designed from consistent subsets of options for each parameter. These scenarios represent major possible courses for the environment in which policy will take place. They are provided as an input to further discussion. The policy strategies that follow in the later section (9.5) of this report are not dependent on endorsing one scenario or the other.

To provide a geographical context, and as a reminder that further analysis is needed to consider more than just the U.S. and Europe, Figure 61 shows the trends for the ICT market worldwide. The key parameters for scenario design follow in Table 37.

Figure 61: Worldwide ICT market growth, by region



2006 Market value 2,027 Billion Euros. Source: EITO 2006, “ICT markets”

Table 36: Key parameters for scenario design.

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
Business strategy and organization			
"Software as a service" as business models	"Software as a service" is a buzzword in business models statements of ICT companies. However it can develop under models that correspond to different positioning with regard to the underlying software-licensing model. 1. Software as a service develops predominantly under a proprietary-software license rental model 2. Software as a service develops predominantly under an access to bundle of services associated to the provision and usage of software (which can be FLOSS or part-FLOSS/part undistributed or proprietary)	2. Can lead to a good synergy between proprietary of in-house undistributed software and FLOSS. This cohabitation would of course not be without tension (see e.g. debates on whether free software licenses should require or not distribution of source code of Web-based applications. 1. On the contrary can in some domains correspond to a de facto re-appropriation of software in application domains in which FLOSS tends to increasingly compete with dominant proprietary software. The important contribution of FLOSS to Web-based applications (see above parameter Web 2.0 services) can contribute to keeping 2. open.	Strong software service companies acting as solution providers and infocenters in Europe could use 2. To better resist the incoming US providers of such solutions. However, they have been up to know relatively timid in making it an explicit strategy.
Outsourcing and internalisation	Relative trends concerning software development, support and infocenters: 1. More outsourcing 2. Re-internalisation	FLOSS entertains a complex relation with outsourcing. By inducing a better modularisation of applications and reducing supplier dependency, it can favour partial outsourcing of IT for secondary users. However, due to the needed in-house competence and the fact	While the adoption of FLOSS-based strategies by large firms shows delays in Europe compared to the US, there has been a real blossoming of small to medium specialised FLOSS-based solution providers. They are growingly able to address larger scale applications through their own growth, networking between SMEs and some partnerships with larger service companies.

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
		that it opens new mutualisation and distribution potential, FLOSS also increases internal development and is often associated to re-internalisation processes. Rather than a choice between the 2., FLOSS acts as favouring different combinations of outsourcing and internalisation, whose key characteristic regards the (needed and induced) skills in user companies and organisations. The ICT market is still immature in responding to the corresponding needs in particular in terms of advice to contracting entities.	The key question regards the credibility of the "outsourcing followed by re-internalisation or transfer to another contractor" scenarios. Only under these scenarios does the theoretical potential of FLOSS for increased supplier independence gain real credibility.
Information services and communities business models	Providing means to information exchanges and creating synergy between them and physical goods or human-delivered services is by far the most important domain for assessing the impact of FLOSS on the economy. Whether or not this sphere will take off as a "quaternary" sphere of activities in economic terms is highly dependent on the type of coupling between free information exchanges and monetary-economy based goods and services. The corresponding "options" differ by the relative importance of :	There is high potential synergy between FLOSS and funding/business models 1, 3 and 6. On the contrary, the efforts to develop 4. are generally seen as adverse to the deployment and development of FLOSS. while 5 and 2. are compatible with both FLOSS and proprietary models for the underlying software. There is debate on whether 2. (advertising) which has often been used by FLOSS-based communities remains compatible in a longer-term perspective considering some side effects.	Europe has adopted systems based on legal licensing and fees in a variety of domains (2.) and there is growing support for them, though the deployment of DRMs is also accompanied by a trend towards considering levies-based systems as to be phased out. This is going to one the key policy choices in the coming years. Traditionally, 6. was considered a working model for the US but less so in Europe due to lack of donation culture. Recent trends seem to point out an evolution in this respect. 3 is equally important in Europe and the US, while 5 can have an important potential in Europe due to the quality of the telecom infrastructure on top of which it generally develops.

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
	1. Indirect funding by tax and fees redistribution 2. Indirect funding by advertising 3. Indirect funding by non-informational goods and services 4. Direct funding by information sales 5. Direct funding by information service subscription 6. Direct funding by voluntary contributions, donations, community membership fees		
Licensing of major development languages and environments	There is an intrinsic difficulty to sustain development tool and environment activities. 3 types of responses have emerged to this difficulty : 1. Continued trends towards development languages and environment becoming available as FLOSS : Java under a FLOSS license, Web 2.0 toolkits predominantly FLOSS, FLOSS middleware, Eclipse and similar IDE become reference solutions including for specialised devices 2. Predominance of a mixed OMA-type model with "open" APIs and proprietary implementations 3. Microsoft / Oracle (for general purpose computing), specialised device manufacturers are able to maintain a predominantly	1. and at a lesser extent 2. allow for a competition between proprietary and FLOSS implementations. How open is this competition depends largely on licensing for patents connected to the implementation of specs for industry standards and related functionality. The fact that key development environments are growingly FLOSS is a powerful factor in mainstreaming FLOSS-related skills. A long-awaited decision of Sun to license Java and its development kits under a FLOSS license would reinforce this trend.	Initially Europe (for instance Ericsson) was a pioneer in distributing development environments as FLOSS. However, recent trends seem to have been predominantly towards choice 2. which is often seen as dominant players in niche markets as a strategy top consolidate its position while resisting to the entry of general-purpose proprietary players.

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
	proprietary development universe (either directly through IDE licenses or indirectly through dependency on obtaining licences for needed IP).		
Security and privacy in a networked environment	1. Critical security software components are FLOSS implementations of open protocols 2. Security through obscurity (non open) becomes the standard security model for internet software components	Software security is often envisioned as a chain of components that must share common properties to be effective. This renders difficult a cohabitation of the two security paradigms. FLOSS is in synergy with the open science model and is dominant in web security components as far as protocols are concerned (SSL, PKI etc.). A critical parameter of user and corporate adoption of security systems is trust. Because they allow open assessment, only FLOSS systems or at least open standards allows a wide diversity of actors to compete for this trust.	

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
“Social networking” services	1. Social networking technologies based on proprietary software and formats tends to exclude any possible interoperability between services. End-users contributions to the design of “social networking” systems is reduced to a minimum. 2. Adoption of FLOSS orientates the technology towards interoperability and loosely coupled systems that permit user innovation at low entry cost.	“Social networking” systems were born around small technical communities using FLOSS as the base for all the network infrastructure software. Lately, the rise of web advertising income and the need for large and robust technical infrastructures to scale up to millions of users has opened a new market for all kinds of economical actors. Because of their very nature, the success of such systems can be based on benefits from mass adoption and “user enclosure” instead of innovation. A mid-term success of a monopoly based business model is a potential threat. In the field of web user-centric communication, early development and adoption of FLOSS is a “sine qua non” condition to sustain end-user innovation and prevent users’ data enclosure.	

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
Technology			
Web 2.0-enabled services	1. AJAX[207] and other Web 2.0 technology lead to re-centralizing (as services) applications and data that were previously autonomous on end-user machines or local networks 2. Centralization occurs for metadata but data itself and software remain largely decentralized. Only services that can be externalised without the need to centralize data and for which different providers exists truly become widely used web 2.0 applications (mapping services, financial services etc.)	FLOSS is an important enabler of Web 2.0 services and their rapid spread. However some key components are in an unstable state in this regard (for instance contrarily to the Yahoo User Interface Library, only parts of the Google AJAX toolkit are distributed under open source licenses). The general philosophy of FLOSS favours 2. (better user control) and FLOSS could be impacted negatively by a predominant deployment of 1. In addition, 2. can induce synergy between FLOSS and non-distributed software, while 1. could lead to re-appropriation of applications for which mature FLOSS solutions exist today.	There are some very innovative Web 2.0 companies in Europe, and some telecom players have demonstrated interest for a FLOSS-based development of Web 2.0 applications. However, these efforts lack critical mass, and they do not profit from the huge synergy that develops presently in the US West coast on these technology.
Scalable decentralized P2P[208]	1. Breakthroughs enable scalable P2P applications without centralised servers or with a very large numbers of peering servers	Most P2P innovation arises from the FLOSS world. Obtaining scalable search mechanisms in very vast P2P networks raises	There is strong innovation in Europe on these issues, in part due to a legal environment that is (was?) less hostile to P2P file sharing. There is some capitalisation on this innovation to offer services such as the growing

207 AJAX : shorthand for Asynchronous JavaScript and XML, is a Web development technique for creating interactive web applications

208 P2P : peer-to-peer is a technonology paradigm in which the bandwidth and the computing power is provided by each end-user participant instead of relying on centralised servers

Domains and parameters	Values of parameter	Impact of FLOSS on parameter and reciprocally	Possible differences between Europe and the US
	2. Most search, hosting and P2P application remain based on centralised servers or a limited number of peering servers	many hard challenges, and output is uncertain. A technological breakthrough in this respect would notably modify the balance between centralised service providers and distributed providers.	usage of Bittorrent P2P technologies to distribute media content (podcasts, vlogs, etc.). Though companies such as France Telecom have large teams working on P2P applications, the legal and business uncertainty limits deployment. In the connex grid-computing domain, Europe has failed, except in the UK, to make a decisive choice for FLOSS, and this limits its ability to drive global innovation.
Specialised and general-purpose information devices	1. Specialised IT devices interact (open interoperability) with general-purpose environments, and include user-production facilities (recording, shooting images, etc.) 2. Specialised IT devices are connected to vertically segmented markets and controls (DRM[209], limited interoperability with general-purpose equipment)	FLOSS generally enables high interoperability between general purpose IT platforms and specialised devices. However, open hardware and DRM can limit this effect (example: the Linux-based set-top boxes and MP3 players are not necessarily those that interoperate most easily will Linux-based PCs). See present discussion on GPL version 3. Scenario 2. would significantly hinder the take-up of FLOSS in general consumer usage, but the likelihood of its development is limited by consumer preferences.	Europe device manufacturers have been innovative in FLOSS -based specialised devices (both consumers and industry embedded systems). However this has been mostly an opportunistic scenario on non-core markets for companies, with limited impact. Compared to the situation in the US/Japan after the absorption of the embedded Linux consortium specs in OSDL[210] there is an "wait and see" position in Europe. There are some signs of Nokia/OMA adopting a more decisively FLOSS-oriented approach including for its core mobile phone activities, but it is yet to be confirmed. The representatives in companies in consortia of large European device integrators (such as ITEA[211]) are often relatively aged managers with a conservative approach to business and innovation models. They remain reluctant to adopt true FLOSS-based strategies.
Client side Operating System	1. Non-FLOSS OS and standard end-user applications remain the standard on all general-purpose	The client's Operating System has a key position in the technical infrastructure and thus in the	Scenario 3. corresponds to a possible positioning for the EU's device manufacturers (mobile devices, consumer electronics). The situation is unstable

209 DRM : Digital Rights Management referrs to any of several technologies used to enforce pre-defined policies for controlling access to digital data

210 ODSL : Open Source Development Labs, funded 2000, is an industrial consortium dedicated to advancement of the Linux operating system

211 ITEA : EUREKA programme consortium for Software-Intensive Systems and Services

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
	computing machines. The market concentration creates vertically segmented markets with high dependence between hardware and low level driver software. 2. FLOSS Operating Systems manage to gain significant adoption on end-users platforms. The high incentive for mobile devices manufacturers to create FLOSS drivers leads to an horizontally segmented market with low barriers for new entrants and low level innovation. 3. A fully networked environment becomes the standard for end-user computing by adoption of either thin clients (in specialised applications) or standard virtual machines for user interaction with remote systems and communications between systems themselves. The OS becomes a less essential component in the infrastructure.	economical ecosystem revolving around end-user platform. In a strongly interrelated technical and thus economical environment, FLOSS development is a way to prevent markets to be captured "from the ground-up" by monopoly on lower level of the software stack. 3. could provide an intermediate situation in which proprietary and FLOSS underlying OSes implementations of virtual machines would compete. Whether this constitutes a true "level-playing field" depends on whether the device manufacturers only provide limited APIs or true access to low-level performance and innovation.	regarding choices between proprietary and open strategies in this respect. Within the latter, there are also uncertainties (see "Specialised and general purpose information devices").
Policy environment			
Patentability of software	1. New laws codify rejection of software patentability in Europe;	Software patents are perceived as a major danger for FLOSS by all its	The existence of a difference in the legal framework (if not or at least not at the same extent in patent office

[212] EPO : the European Patent Organisation is a public international organisation with the task of granting European patents

Domains and parameters	*Values of parameter*	*Impact of FLOSS on parameter and reciprocally*	*Possible differences between Europe and the US*
	the US adopting a limited patent reform centred on raising requirements of inventiveness and prior art issues. 2. The present EPO[212]'s practice continues, without being codified as law (Community or EPC[213] revision). The US accomplishes a limited patent reform as in scenario 1. 3. A new effort at legitimising software patentability succeeds (either through a new directive, through SPLT[214] or through the Community patent regulation). The US does only a cosmetic patent reform.	players except a limited number of large firms holding large software patent portfolios in the US. The danger may be even higher for the adoption by secondary users (industry and administration) of FLOSS, and especially in terms of availability of open interoperable standards. Scenario 3 would have a major blocking effect. Under scenario 2, the adverse impact on FLOSS of software patentability is limited by legal uncertainty.	practice) is seen by some as a chance for a new innovation wave in Europe based on lower entry costs and absence of patent thickets, while pro-patent circles describe it as an impediment to competing based on patents in the US and global markets.
Internet neutrality	1. Rupture with Internet neutrality, creation of sub networks connected to forms of commercial provisions of media, reinforced by deployment of IPv6. Restrictive measures could also be enforced at state level in developing countries where no "internet neutrality" culture exists (China).	The "Internet neutrality" debate in the US is still undecided. It is superposed with a more global underground modification in Internet neutrality connected to the partial deployment of IPv6. FLOSS has coevolved with a neutral, agnostic and equitable to non-commercial contents Internet.	Paradoxically, Europe, in which some countries have long resisted the penetration of the Internet, or many specialised players have resisted or canalised IP-based convergence, is maybe now more aware of its benefit than the US.

213 EPC : European Patent Convention is a multilateral treaty instituting the European Patent Organisation and providing an autonomous legal system according to which European patents are granted

214 SPLT : Substantive Patent Law Treaty (SPLT) is a proposed international patent law treaty aimed at harmonizing substantive points of patent law

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
	2. Reaffirmation of Internet neutrality. Possible inversion of the respective situation of the Europe and US (Europe becoming more favourable to Internet neutrality than the US).		
DRM and P2P	1. Strong deployment of hardware-enabled DRMs without outlawing of circumvention and limited interoperability provisions (no force disclosure of interfaces of TPMs[215], no automatic permission of disclosing source code for those implementing FLOSS-based interoperability) 2. Extreme version of 1. with compulsory DRM clauses for classes of appliances or media (broadcast flag, Digital Transition Content Security Act, "VU" amendment to DADVSI[216] law in France) 3. Limited deployment of DRMs, DRM-free hardware and software platform are available, open competition between DRM-based and DRM-free content offers.	Hardware cryptography-DRMs represent a major risk for FLOSS deployment. A similar though less fatal risk exists for software-based DRMs when no protection of interoperability with free software exists. Compulsory DRM add the risk of a strong legal uncertainty and liability risks for FLOSS platforms Thus 1. and 2. could both severely hinder FLOSS. On the contrary, under 3., a more level-playing field for competition between content distribution models would create a strong demand for FLOSS. In the other direction, the already installed FLOSS-enabled applications represent one of the key sources of opposition to strong DRM deployment, together with consumer concerns.	There is a stronger endorsement of interoperability and a stronger consciousness of the benefit of non-commercial sharing. There used to be a much weaker definition of limitations of exceptions and fair use, but these rights have been severely limited in the US in the past 8 years. As a result, there could exist a greater willingness to explore 3. in Europe.

[215] TPM : Technical Protection Measure, technologies used to control access to information

[216] DADVSI : "loi sur le droit d'auteur et les droits voisins dans la société de l'information", French law on author's rights and related rights in the information society

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
Interoperability and open standards	Interoperability and open standards are not only a policy issue, they depend as well on the evolution of technology and economic players' strategies. However, they have given rise to a growing number of policy initiatives, both pro-active (promotion of interoperability frameworks and open standards) and corrective (limiting the effects of DRMs, correcting the effects of dominant position abuse, etc.). 1. Interoperability efforts come always too late: after the technology changes that displace the lock-in effects, after dominant positions are installed. They fail to prevent usage of patents or other intellectual property mechanisms and usage of DRMS, TCPA and other signed code techniques in order to make interoperability impossible or costly in practice. Interoperability legislation of policy does not enforce it under conditions that are compatible with FLOSS. 2. Definitions of open standards and interoperability policies take in account the requirements of FLOSS. They are sufficiently proactive and preventive to act at an early stage before strong	FLOSS is both a (the) key enabler for interoperability and gets a strong benefit from its existence. The evolution of technology enables a much greater interoperability through Web-based applications and virtual machines. This trend is however countered by some installed industry strategies.	Europe has a much stronger tradition of creating the conditions for interoperability through its decompilation and reverse engineering legislation. It has more recently developed strong policy in the field of FLOSS compatible definition of open standards and interoperability frameworks. However some factors limit thee benefits that Europe could obtain from this advantage: - the evolution of standards that are more and more encumbered with patents except at W3C) - regulatory policy: weak interoperability provisions in the transpositions of EUCD for instance - competition policy remaining corrective “after the irreversible”.

Domains and parameters	Values of parameter	Impact of FLOSS on parameter and reciprocally	Possible differences between Europe and the US
	network effects are installed. The Open Document Format and similar standards for generic ICT applications are recognised as the basis of ICT		
Education, skills and culture			
ICT education and related skills	1. Education policy promotes the distinction between the learning of ICT concepts and the usage of a particular software. 2. Education policy sticks to short-term market effectiveness and focuses on usage skills for dominant software products.	Option 1 helps the learners to abstract concepts from the tool. When learning with FLOSS they are given the opportunity to implement their own ideas and often improve existing functionality. VLC (a network video player) stands as a good example of a FLOSS product developed by an engineering student. Option 2 appears as a pragmatic and efficient approach to teach practical abilities. However, learning engineering skills, or even acquiring know-how that are long lasting and can adapt to changes in environment or technology requires using software with an open architecture and access to the source code. In general, users of FLOSS products tend to get a better idea of 'how' the service is rendered by the software (not only which functionality it offers).	
Consumer or prosumer	1. Software perceived mostly as a	Option 1 is clearly unfavourable to	Individuals in Europe have shown at least as great an

Domains and parameters	***Values of parameter***	***Impact of FLOSS on parameter and reciprocally***	***Possible differences between Europe and the US***
culture	black box delivering functionality. 2. Consumer awareness of information control and privacy issues, but with limited investment in mastering the underlying technical or legal issues. 3. Development of a prosumer culture for software and information applications and services. Strong involvement of consumer organisations in choices regarding the legal and contractual environment and the technical design of devices and services.	the spread of FLOSS outside the infrastructure and back-offices. Option 2 leads to a positive evaluation of FLOSS products or FLOSS-based services, without influencing the nature of applications and usage. Option 3 encourages a wider development of self-production and exchange of information artefacts: it favours FLOSS deployment and amplifies its wider economic and societal impact.	endorsement for prosumer type of activities than in the US, as exemplified by the explosive development of blogs, personal production and exchange of photographs and video, and home production. However, the offer of the related technology and services by the European ICT industry is not at the same level, if one excepts a wide development of community-based FLOSS projects.

9.4. Scenarios

In this section, we present three scenarios: CLOSURE, GENERIC, and VOLUNTARY. These scenarios highlight macroscopic trends in the business, technology and policy environments. Each one is associated with a consistent set of values for the various parameters identified and discussed in the previous section. They are provided as a basis for further discussion.

CLOSURE is a scenario in which the protection of some existing business models plays a key role in determining a legal and business context that is unfavourable to FLOSS development and deployment. The adoption of FLOSS strategies by the large European ICT industry remains limited to usage as a cost reduction scheme for existing infrastructure software and limited experiments in non-critical segments of business. In the IPR domain, the trajectory of scope extension (patentability of information processing methods, broadcast and netcasting rights, etc.) and harsher execution (DRM/TCPA deployment backed by unmitigated TPM legislation, criminal and civil sanctions for alleged IP infringements, etc.) for IPR is pursued. Competition policy is strictly corrective rather than pre-emptive and conducted as arbitration between installed businesses. FLOSS development continues to proceed, but is limited to niche hacker domains.

GENERIC is a scenario in which a dual software economy develops, where installed generic applications see a delayed but strong deployment of FLOSS software. New functionality appears in a dual mode: partly as proprietary software connected to the installed base of dominant players and partly as societal innovation where business models are not immediately apparent. However, this latter form of innovation remains weakly funded and its trajectories of economic valuation remain constrained by a conjunction of intellectual property-related restrictions and control on the installed base for the ICT infrastructure. Economic recognition of FLOSS does not extend beyond software to commons-based mechanisms for information (for instance geographic or scientific) and other creative works. Policy remains focussed on protecting interoperability ex-post for existing software and corrective competition measures, rather than pro-active encouragement of open standards. The name of the scenario is a reference to both the generic ICT applications and the situation prevailing for delayed generic drugs in the pharmaceutical industry[217].

VOLUNTARY is a scenario where policy develops to encourage a stronger investment potential and an easier legal environment for new FLOSS functionality. Interoperability and competition policy become more pre-emptive, taking into account the impact of planned products on future functionality and related-business models and avoiding the potential for vendor lock-in during (especially public) procurement. ICT-related education is centred on functionality rather than individual applications, as illustrated by FLOSS implementations. Research and innovation policies become FLOSS-friendly and more generally favourable to the production of information commons results. Cultural policy encourages self-production and non-commercial exchange of cultural information artefacts. Pre-competitive European industry fora for the creation of a FLOSS innovation basis (such as the ObjectWeb consortium in software, or

217 This analogy was used in Jean-Michel Dalle, Laurent Kott, Plaidoyer pour des logiciels génériques, La Recherche, janvier 2002.

the SNP consortium in pharma/biotechnology) are encouraged. Part of the large European ICT industry adopts an explicit FLOSS-based business strategy. Standardization policy implements strong definitions for the openness of standards based on their economic effect.

Table 37 gives the parameter values (as described in the previous section 9.3, "Factors determining impact of FLOSS on the EU ICT market") for the three scenarios.

Table 37: Values of key parameters for three scenarios

Parameter	***CLOSURE***	***GENERIC***	***VOLUNTARY***
"Software as a service" (SaaS) as business models	1: transfer of proprietary lock-in to Web-based applications	2: Competition between FLOSS and proprietary approaches to SaaS	2: Competition between FLOSS and proprietary approaches to SaaS
Outsourcing and internalisation	Neutral: CLOSURE favours forms of outsourcing with strong boundaries of control and competence (strong supplier control on users, limited transfer of competence to subcontractors) while VOLUNTARY induces competence build-up within user and subcontracting organisations.		
Information services and communities business models	**2**, **4**, 5: advertising, information sales, information service subscription	1, 2, **3**, **5**, 6: bottom-up, legally/policy-organized mutualisation mechanisms, advertising, indirect funding by goods and services, information service subscriptions	**1**, **3**, 5, **6**: bottom-up or legally/policy-organized mutualisation mechanisms, indirect funding by goods and services, information service subscriptions (community membership)
Licensing of major development languages and environments	3: predominantly proprietary development universe	2: open APIs, FLOSS implementations delayed compared to proprietary implementations	1: major languages and development environments (including recent) under FLOSS licenses
Security and privacy in a networked environment	2: security and privacy through obscurity and centralized control, resulting doubts, limitations on user information management know-how.	1: FLOSS-based security and privacy. Some user empowerment (e.g. in spam-filtering).	1: FLOSS based security and privacy, stronger user empowerment in personal information management
"Social networking" services	1: social networking technologies based on proprietary software and formats hinder interoperability between services	intermediate between 1 and 2: promotion of interoperability, but vertical integration using proprietary software locks in most users	2: adoption of FLOSS orientates the technology towards interoperability and loosely coupled systems that permits user innovation at low entry cost
Web 2.0-enabled services	1: re-centralization of services and user data	neutral	2: centralization only of metadata and indexing
Scalable decentralized P2P	2: search, hosting and P2P application remain based on centralised or peering servers	neutral	1: scenario eased if P2P applications become scaleable in a wider variety of fields
Specialised and general-purpose information devices	2: vertically segmented markets and controls for specialised IT devices	(1-2): Limited interoperability with general-purpose equipment. Some European ICT/ consumer electronics players bet on more open interoperability and user self-production.	1: open interoperability including with general-purpose environments. Specialised IT devices include user-production facilities
Client side operating	1: non-FLOSS OS and end-	2. Fully networked	3: FLOSS Operating

Parameter	***CLOSURE***	***GENERIC***	***VOLUNTARY***
System	user applications remain the standard. Concentrated vertical segmentation with high dependency between hardware and software.	environment by adoption of thin clients (for specialised applications) or standard virtual machines. Also compatible with (3)	Systems gain significant adoption on end-users platforms. Horizontally segmented market. Also compatible with (2)
Software design practices	2: usage of FLOSS development components limited to some innovative segments (due to lack of large European company adoption)	2: usage of FLOSS development components limited to some innovative segments (due to lack of large European company adoption)	1: software development-FLOSS components become mainstream and benefit from the adoption by large companies and institutions.
Patent ability of software	3: software patent ability codified in law. Cosmetic patent reform in the US.	2: EPO practice continues, without being codified as law. Limited patent reform in the US.	1: laws codify rejection of software patentability in Europe, limited patent reform in the US
Internet neutrality	1: sub networks connected to forms of commercial provisions of media.	2. Internet neutrality reaffirmed, possibly stronger in Europe than in the US.	2. Internet neutrality reaffirmed, possibly stronger in Europe than in the US.
DRM and P2P	1: strong deployment of hardware-enabled DRMs whose circumvention is outlawed and limited interoperability provisions with possibly: 2: compulsory DRM clauses for classes of appliances or media	3. limited deployment of DRMs, DRM-free hardware and software platforms are available, open competition between DRM-based and DRM-free content offers	4. same as 3. + strong development of P2P-based non-commercial exchange of self-published contents (or third party contents under legal licensing schemes)
Interoperability and open standards	1: interoperability efforts come always too late	2: definitions of open standards and interoperability policies take in account the requirements of FLOSS and are sufficiently proactive to act at an early stage	2: definitions of open standards and interoperability policies take in account the requirements of FLOSS and are sufficiently proactive to act at an early stage
ICT education and related skills	1: education policy sticks to short-term market effectiveness and focuses on usage skills for dominant software products.	2: Education policy promotes the distinction between the learning of ICT concepts and the usage of a particular software	same as 2. + explicit decision to use FLOSS software to achieve the aim.
Consumer or prosumer culture	1: software perceived mostly as a black box delivering a functionality	2: Consumer awareness of information control and privacy issues, but with limited investment in mastering the underlying technical or legal issues	3: development of a prosumer culture for software and information applications and services

The key points to note are:

The difference between CLOSURE and GENERIC relates to the balance between various approaches to software within a more or less constant perimeter of applications and of relationship between software and the overall economy and society, The main difference between GENERIC and VOLUNTARY relates to the nature of the software and information activities in which individuals (and groups) engage, and thus the perimeter of the information economy and society.

The options for many parameters are similar between GENERIC and VOLUNTARY. Those parameters for which the options differ are connected to the relationship between individuals and technology/information on one side, and the degree of voluntarism in policy choices related to FLOSS or information commons on the other side.

GENERIC can be seen as the centre of gravity of the European information society policy since 1991: commitment to interoperability, encouragement to standardisation within ICT[218] and to the widespread usage of ICT by non-ICT economic players, and a relatively narrow vision of ICT activities by individuals and groups, focussed on a vision of individuals as consumers rather than producers. The limits of GENERIC explain the strength with which CLOSURE has been growingly imposed from the outside both on information society policy and on company business strategies: neither policy nor dominant business strategies have made a strong and consistent endorsement in favour of an open prosumer information society. VOLUNTARY assumes a change in this situation, both in policy and in the strategy of some companies.

218 See discussion in section 9.5.1 and footnote 221 on page 206.

9.5. Policy strategies

We now turn to policy strategies. The strategies outlined in this section aim to enable Europe to make the most of its current situation relating to FLOSS, on the basis of findings in Sections 6 and 7, macroeconomic modelling and qualitative analysis in Section 8, the SWOT in Section 9.2 and the analysis presented when introducing the parameters for scenario design in Section 9.3.

We focus on policy that goes beyond interoperability-related actions, addressing topics such the recognition of voluntary commons schemes in the legal environment, the R&D environment, the innovation incentives, technology transfer, education, the facilitation of industry fora, the encouragement to partnerships between large ICT industry and emerging FLOSS SMEs, the conditions for the development of FLOSS approaches within the industry and administrations, actions that target anti-competitive situations arising from bundling hardware and software, and the creation of a common ground for exploring prosumer-based information society applications.

9.5.1. Background: The limits of interoperability-based policy

Interoperability-related policy has been a key component of keeping open the potential of FLOSS development and deployment in Europe. This has occurred in two stages.

First, at the end of the 1980s and during the first half of the 1990s, the predominant attention was focussed on reverse engineering and decompilation rights for the sake of interoperability. The 1991 European Directive[219] provisions in this respect remain a very important asset for Europe, even though at the time they were considered as a damage limitation compromise in face of the strong opposition of dominant IT players to stronger interoperability clauses. The 1991 directive provisions were relatively effective while the only possible restrictions to implementing software interoperability were in the realm of copyright[220]. In parallel to these legislative aspects, this first phase of interoperability policy was dominated by the ambiguous notion of open systems. Though there was an authentic desire to encourage openness and interoperability, and some significant developments for instance in the field of software engineering, the ambiguity came from the lack of explicit definition of the sense in which systems were to be open and of doubts on the governance of organisations promoting open systems in a period where the fragmentation of the various flavours of proprietary UNIXes

---

[219] European Council Directive of 14 May 1991 on the legal protection of computer programs (91/250/EEC)

[220] In presence of patents on software or information processing methods or in presence of technology restrictions to interoperability (possibly supported by legal outlawing of circumvention) the 1991 directive provisions no longer guarantee the possibility to develop an original software that interoperates with an existing software or device. During the legislative debates on software patentability, the way to adapt interoperability legislation to a possible existence of legally officialised software patents was the object of a lively debate, left to an uncertain conclusion by the rejection of the directive and the continued practice of the EPO of delivering software and information processing method patents. The directive on copyright in the information society left open for Member States the possibility to adopt partial exceptions to the outlawing of circumvention of "effective technical protection measures" for the sake of interoperability in its article 6.2b. The transpositions have been very variable but generally mark a regression in the enforcement of interoperability in this particular domain.

created much confusion. In particular, there was some confusion between the promotion of openness and the promotion of a particular type of operating systems (UNIXes).

The second, more recent phase of interoperability policy is focussed on open standards and formats and common interoperability frameworks. This second phase built partially upon a critique of the first phase, but even more was constructed as a reaction to challenges for public administration projects in an era of monopolistic or oligopolistic control on key platform and application software. Administrations were confronted with challenges in managing the interoperability of their own software due to the lack of interoperability between their various applications or end-user platforms, and even more to a growing lack of control on evolution of their own software. Common interoperability frameworks developed in most Member states and for which the European Commission's IDA (now IDABC) programme played an important coordination role addressed these challenges by trying to limit interdependency between component applications thanks to (XML-encoded) specifications of their interaction, by moving towards Web-based versions of most business applications, and by endorsing open standards, formats and protocols for communication. After some tuning, the need for explicit definitions of the sense in which standards, formats and protocols were open was strongly felt. After much debate, the definition of open standards published in IDABC's Interoperability Framework during the Dutch presidency of the EU in 2004 has become a reference point and its principles have been encoded in several Member states legislations or policies, for instance the French *Loi sur l'économie numérique, or the Dutch government ICT Unit (ICTU), where a closely related earlier definition originated.* The key element in this definition is that it requires that relevant intellectual property titles that have a bearing on the standard are licensed irrevocably, royalty-free and without discrimination, particularly in the domain of software[221]. This is justified on the basis of the economic impact of open standards as allowing full competition in the market for the supply of technology despite the possibility of a "natural monopoly" in the (open standard) technology itself[222]. FLOSS played an important role in this maturation of interoperability policy, by demonstrating the fact that when a reference implementation of the full chain of usage of a standard was available as FLOSS, the intended benefits of openness materialised truly (see for instance the giant domain of applications and markets opened by the first JPEG standard, or even the several standards for the Internet and World Wide Web).

Despite this maturation, there are indications that focussing on interoperability aspects alone are not sufficient to rebalance markets and innovation in presence of strong network effects, horizontal and vertical technology and usage interdependency, and bias of innovation incentives towards restrictive targeted proprietary approaches. This can be illustrated by a number of facts:

[221] Some standardization for a where industry is strongly represented such as W3C have adopted standardization policies that are adapted to effective interoperability including with FLOSS. A difference between the GENERIC and VOLUNTARY scenarios presented in section 9.4 regards where the line will be drawn between such policies and the "RAND" policies, many of which are not adapted to FLOSS-based interoperability, in particular when FLOSS development is not supported by large companies.

[222] See Ghosh, R.A. 2005c. "An economic basis for open standards", FLOSSPOLS project report, *European Commission DG INFSO.* Available at http://flosspols.org/deliverables/FLOSSPOLS-D04-openstandards-v6.pdf#search=%22ghosh%20open%20standards%20economic%22

The bundling of hardware and software (maintained through pressure on OEM suppliers) and the constraints of legacy platforms and applications suffice to maintain a monopolistic situation on operating systems software for personal computers, except where strong policies of escaping this situation are put in place. This is the case despite true breakthroughs such as the Ubuntu distribution that have made GNU/Linux easier to install on a machine, in practice, than Windows. But this is not relevant to end-users since they are still almost only ever offered pre-installed-Windows machines, and must then struggle to remove it totally or partially before manually installing GNU/Linux. There is now a limited offer of pre-installed GNU/Linux machines, though these often target the top end of the market, with differential prices sometimes leading to the same hardware costing *less* with commercially licensed Windows than with a free operating system[223]. The anomaly of this situation is demonstrated by the situation of servers and specialised IT devices (from set-top boxes to music players, PDAs and now mobile smartphones, where the Linux market share is just behind the market leader, Symbian, see section 7.7, "Secondary production and services"). There, FLOSS OSes are strong contenders or leaders. What the ICT industry does everyday (choosing GNU/Linux or other FLOSS OSes as the basis for their products) is still made very difficult for user organisations and end-users of general-purpose IT, at the expense of competition and ability to invest in other layers.

Even for generic applications for which there exists FLOSS software that are at least equivalent to (or better than) dominant proprietary applications such as office suites and Web browsers, pre-installed software applications with a dominant market share are able to retain around 90% of the total installed base. This prevents the introduction of requirements for document formats to be open in the above sense, and allows the promotion of formats that superficially appear open but are so complex to use for their competitors for legal or other reasons (in particular FLOSS) that they are in practice proprietary.

More generally, the inertia created by an installed dominant position easily perpetuates, and due to the strong interdependency between software components and applications in practical end-user activities, even if very limited number of components are dominated by proprietary software, most end-users end up being locked in to entire proprietary platforms. Examples of such key components have included calendar/scheduling applications, instant messaging, streaming media, or Web-based players for encapsulating streamed media players in Web pages for instance. Often, it is U.S. patents or various other IPR-related disputes and the simple inertia of the installed base that are hindering FLOSS competition in these domains.

For business applications, legacy software and legacy skills that have been narrowly focussed on proprietary software products rather than on underlying concepts and techniques play a key role in slowing down the evolution towards a more diverse software ecosystem.

The unprecedented ability of providers of monopolistic proprietary software to obtain huge profit margins and related capitalisation (see Section 8.1 on capitalisation and innovation in the ICT industry) results in their ability to invest huge sums in lobbying, promotion, and defensive innovation to protect their installed positions, as well as the acquisition of innovative

[223] See for instance the extended reportage by *The Register* on Dell's pricing of "Open Source PCs" without Windows: Vance, Ashlee, 2005, "Reg readers take the Dell 'Open-source PC' challenge", *The Register* October 7, available at http://www.theregister.com/2005/10/07/dell_linux_tough/

firms to neutralise their competition. In contrast, FLOSS and more generally industry with service-based business models (with the exception of centralised intermediators such as Google, Yahoo and the like) are relatively starved of capital and human resources.

Some cultural factors, that can be summed up by slogans such as “technology is a black box, I don't want to know how it works”, “nobody got fired for buying from the leading provider” and “intellectual property ownership guarantees future investment in innovation” are still powerful. Commons-based innovation has not yet received the level of official recognition that would set it as an alternative to be considered by decision-makers. This means that purchase decisions are in fact not made on the basis of “buying the best product for a need”, but typically made on the basis of “buying the best proprietary product for a need” or more commonly “buying what we/others have always bought” – as empirically demonstrated in the FLOSSPOLS EU-wide survey of 955 government authorities.[224]

The changes associated with relying on software produced by a complex ecosystem require a difficult adaptation in user organisations (including the ICT industry), to which the growing FLOSS economy has not yet provided sufficiently convincing answers: who provides minimal guarantees for future support, for time-frames for developments, or the governance of the roadmap for evolution of a software application. These questions receive unsatisfactory answers in many cases of proprietary software too, but at least there is an easy source of answers when there is a single proprietor.

The above discussion relates to the question that may be asked as to why we provide policy strategies in relation to FLOSS at all. A criticism might be that “if FLOSS is indeed better and cheaper, no active policy in favour of FLOSS would be necessary”. In fact, this criticism is quite wrong, in that it naïvely supposes there are no network externalities and installed base effects. The statement relates to the *private* costs (and benefits) of FLOSS, and even if it provides technical or other superiority over proprietary equivalents, and does not apply to the existing context in which proprietary software vendors have a dominant market position, and deliberately follow a strategy of locking in users to their software through the use of proprietary standards[225]. It is notable that the case of the return of the Central Scotland Police to Windows, after switching for a few years to Linux (see Section 7.6) illustrates exactly this situation – the costs of interoperating with other police departments using proprietary standards was cited as a significant reason for switching back to the proprietary standard[226].

[224] See examples of public procurement tenders and survey results of IT managers' purchasing preferences in Ghosh, R.A. 2005c. “An economic basis for open standards”, FLOSSPOLS project report, *European Commission DG INFSO.* Available at http://flosspols.org/deliverables/FLOSSPOLS-D04-openstandards-v6.pdf#search=%22ghosh%20open%20standards%20economic%22

[225] For example, the recent moves of Microsoft towards opening up its Office document format are clearly a result of pressure from customers and governments and moves, such as those of Denmark and the US state of Massachusetts, to support open standards in general and the Open Document Format (ODF) in particular. It remains unclear whether Microsoft’s new approach will actually be sustained enough to ensure the existence of viable competing implementations of its format.

[226] We are thankful to Microsoft for bringing this case to our notice.

We note that the policy strategies outlined below relate to FLOSS not only because this is a report about the impact of FLOSS on Europe, but because there is a justifiable case to be made, based on the evidence presented in this report, for differential public policy supporting FLOSS production. Actually, there are two arguments. First, past policies created a playing field that has led to market dominance of a few players whose position has been entrenched by public policy including public procurement methods[227]. The effects of this should be reversed because the effects are harmful. So it is reasonable to propose regulations that take into consideration the historical first mover advantages and installed base dominance established by proprietary software publishers.

The second argument, relates to the evolution of hybrid models in which dual licensing and other "mixed" models combining FLOSS and proprietary features exploit the complementarities between FLOSS platforms and proprietary products and services built with them[228]. In such a situation, the FLOSS components retain the features of public goods. But juxtaposed with private goods in a hybrid model, even where these FLOSS components may be cheaper and technically superior, under investment may be much more pronounced in the case of FLOSS production. FLOSS, here, is thus a *complement* to proprietary software, but its public good nature and resulting under investment can justify public policy approaches similar to those for basic research. Indeed, the more complementary FLOSS and proprietary software are – the more they are combined by users and included in hybrid models by businesses and developers – the less the criticism against public policy "crowding out private R&D" applies. (The argument concerning complements is the modern rationale for public funding of basic, exploratory science research, i.e. providing a platform that increases the long-run marginal social rate of return, as well as the private profitability, of commercial applications-oriented R&D.[229])

The policy recommendations that follow try to address the blocking factors and historical preferences for proprietary software, and to unleash the economic and innovation impact of FLOSS innovation. Specific support is given for each of them in their presentation. In addition, Table 38 provides a synoptic view of relationships between each policy strategy and analysis in previous sections of the report.

---

[227] While public procurement supporting FLOSS or even specific open standards is criticised with the argument that public procurement should be "technology neutral" – although FLOSS is not a technology – whilein practice public procurement is frequently not even *vendor neutral.* The vendors preferred – implicitly and often explicitly – so far, have been typically proprietary software producers. See Ghosh, R.A. 2005c. "An economic basis for open standards", FLOSSPOLS project report, *European Commission DG INFSO.* Available at http://flosspols.org/deliverables/FLOSSPOLS-D04-openstandards-v6.pdf#search=%22ghosh%20open%20standards%20economic%22

[228] Termed "OSS2.0" by Fitzgerald, Brian. 2005. "The Transformation of Open Source Software". *MIS Quarterly*, Volume 30, issue 3.

[229] David, P.A. 2003. "The Economic Logic of 'Open Science' and the Balance between Private Property Rights and the Public Domain in Scientific Data and Information: A Primer," in *The Role of the Public Domain in Scientific and Technical Data and Information: A National Research Council Symposium,* J. Esanu and P. F. Uhlir, eds., Washington, D.C.: Academy Press, 2003.

Table 38: Relationship between policy strategies and analysis in the report

Policy strategy	Sections built upon or responded to
Public R&D: dissemination rules and funding models	8.2.7
Innovation incentives: do not penalise FLOSS	7.5, 8.1, 8.2.7
Equitable tax treatment: FLOSS is a donation	8.2.7, 8.5
Education: avoid lifetime vendor lock-in for students	7.4, 8.5, 9.3
FLOSS in pre-competitive research and standardisation	7.5.1, 7.6, 8.2.2, 9.1.1, 9.2, 9.3[230]
Partnerships between large ICT firms, FLOSS SMEs and communities	7.5, 8.3, 9.2, 9.3[231]
Unbundling in the hardware / software domain	7.6, 8.5, 9.2, 9.3[232]
Working towards a prosumer-based information society	8.2.4, 8.2.5, 8.5, 9.2, 9.3
Recognise the legal approach of commons-based schemes	8.2.6, 9.2, 9.3[233]

9.5.2. Public R&D: dissemination rules and funding models

An increased use of FLOSS and more generally information commons mechanisms as dissemination schemes for publicly funded research has been proposed now for some years. While the U.S. has often followed the principle that publicly funded research should be available to the public (without specifying a particular licensing model), this has not been the practice in Europe. Indeed, in practice, the share of software developed from EU funded research being disseminated under FLOSS licenses is extremely small (an estimate of 3% was computed with the IST programme in 2002), though increasing. A DG Research expert group (see Kamperman-Sanders et al 2003)[234] suggested the promotion of open IPR licensing for publicly funded ICT-related research, and in particular, FLOSS licences for resulting software. This has resulted in minor changes in EU policy in the right direction – now, in several areas such as e-health and e-government, funding calls suggest (or occasionally require) the use of open source.

The two main policy decisions regard whether there should be default rules for some kinds of research results and whether the R&D funding environment should be adapted to make it easier for R&D participants to choose these schemes.

---

230 Web 2.0 enabled services
231 Software as a service, licensing of major development environment and languages,
232 Specialised and general-purpose information devices, client-side operating system.
233 All entries in policy environment section.

234 Kamperman-Sanders, A., Granstrand, O., Adams, J., Blind, K., Dumortier, J., Ghosh, R.A., De Laat, B., Kircz, J., Lindroos, V., De Moor, A. 2003. *Expert Group Report on S trategic Use and Adaptation of Intellectual Property Rights Systems in Information and Communications Technologies-based Research.* European Commission, DG Research. Available online at http://ec.europa.eu/research/era/pdf/ipr-ict-report.pdf

After many calls from scientists, the UK discussed, as early as in 2001, making FLOSS a default licensing scheme for research results that receive public funding, with a special meaning of “default scheme”, since it was actually meant as a fall-back scheme to be used when other proprietary schemes were not put in place in a relatively short period. Most discussion was regarding whether such dissemination should then occur under reciprocal or “copyleft” licences such as the GNU GPL or under non-reciprocal or “permissive” licenses. This debate was not truly concluded, and in most European countries, such a choice is generally left to the research teams themselves. In many cases, the provisions regarding existing software from which the software is derived or with which it must be linked impose the choice. A detailed discussion of the conditions and analysis that can determine the choice between copylefting or non-copylefting licenses in more general contexts can be found in the *Guide to choosing and using free software licenses for government and public sector entities*[235] published by the French government IT agency and in the paper *A framework for understanding the impact of GPL copylefting versus non copylefting licenses*[236]. As to the types of research results for which there should be an explicit rules giving priority[237] to FLOSS schemes, the scenario analysis and wider impact analysis suggest establishing such a rule for software that aims at setting a baseline for future innovation by third parties or even end-users, be it basic research, standard reference implementations and other standard-related tools, tool kits for creating services or contents, archiving and indexing software, etc.

Whether such a rule is put in place or not, the largest part of FLOSS dissemination of research results is likely to continue to result from voluntary decisions by the participants in the research actions. However, these decisions are largely dependent on the publicity given to the fact that this is considered a valuable scheme and on the adequacy of the funding schemes that are put in place, so publicity and non-mandatory encouragement may be significant drivers of FLOSS adoption.

European and Member States research programmes have increasingly used a 50% shared cost funding[238] as their core rule. This choice is mostly based on making sure without too complex an individual assessment that the limits set in the WTO protocol on state aids to R&D are respected. It is an unwanted side effect that this should lead to insufficient funding of the production of results that are readily available for use by all, competitors and foreign countries alike. (Public funding R&D of which results would be available to foreign countries would probably not violate WTO rules, as they would not form state aid to regional firms or industries). In practice, the recipient of funding for R&D whose results are disseminated as FLOSS acquires of course some competitive advantage along the lines of the classical FLOSS-based business

235 http://www.adae.gouv.fr/upload/documents/free_software_guide.pdf

236 Philippe Aigrain, *A framework for understanding the impact of GPL copylefting vs. non copylefting licenses*, Free / Open Source Software research community, http://opensource.mit.edu/papers/aigrain2.pdf, also reproduced in Linux User.

237 Meaning that a FLOSS scheme should be used in these cases except when duly justified constraints make it impossible or inadequate, and provided that the intended results can still be obtained despite the lack of FLOSS dissemination.

238 Or lower, e.g. 35% for “demonstration actions” in the EU Framework Programmes.

models such as "best-knowledge here" or because of having oriented the functional choices of the software.

This justifies supporting the development costs, but it is very doubtful that the 50% rule is suitable. In practice, most R&D actions that have successfully supported FLOSS projects (in particular when company participation was necessary) have used niche experimental actions to achieve a higher rate of funding. An explicit adaptation (for instance to the "rules for participation" for EU R&D) could provide the appropriate incentives for a higher dissemination of framework programmes results as FLOSS. A typical figure of 75% funding is a possible estimate that could be adapted depending on the type of research action.

Participants in research actions often hesitate before using FLOSS schemes because they are not sure these schemes would be considered as a convincing dissemination or exploitation plan in the evaluation for their proposals. This may be a case for a better communication of the related opportunities. The growing space devoted to FLOSS-based strategies in the communication of the IPR help-desk of DG Research is an indication of positive change in this respect.

It should be noted that the principle of public availability of publicly funded software development is much more straightforward in the context of 100% public funding[239]. This may be especially applicable to national funding programmes – for example, this is already the policy of a UK Joint Information Systems Committee (JISC) internal directive. On certain kinds of infrastructure software, the UK's Open Middleware Interface Infrastructure Initiate (established at Southampton University)promotes open standards and open source requirements..

Finally, it should be noted that in the neighbouring domain of scientific publication, a worldwide movement is in process towards strong incentives or obligations of publication in open access journals or at least with open access archiving of all papers arising from publicly funded research. The European Commission has recently published a report on this subject[240], and proceeded to a consultation on its recommendations.

9.5.3. Innovation incentives: do not penalise FLOSS

As indicated in the Section 8.1, "Competition, innovation and FLOSS" and in Section 8.2.6, "Building a more neutral legal and regulatory environment", various policy measures can be envisaged to rebalance the incentives to investment in innovation in a manner less unfavourable to FLOSS. Creating incentives for investment in commons-based and commons-producing innovations can be done by tax credits measures, prize funds and mutual funds. The three mechanisms present different constraints:

1. Tax credits require an assessment of the investment (similar to R&D tax credits but with some adaptation as one can not rely on diplomas nor job or department titles to judge of the allocation of resources to FLOSS development) and an assessment of the usefulness of the

[239] In contrast with most EU funded research that as noted previously involves private research with partial public support.

[240] Study on the economic and technical evolution of the scientific publication markets in Europe, http://europa.eu.int/comm/research/science-society/pdf/scientific-publication-study_en.pdf

results that could be done possibly by usage but with unpleasant delays. In many jurisdictions R&D tax credits are provided against potential financial losses due to the risk of commercial failure; this metric may penalise R&D where results are FLOSS.

2. Prize funds (allocation of a prize for provision of a FLOSS implementation of some functionality) require the ability to predetermine the need / usefulness of the functionality and the ability to evaluate ex-post that the provided implementation meets the requirements. One can consider that the tendering that was done in Germany for FLOSS implementations of some strategic components of the IT infrastructure (cryptography, cryptographic email interfaces, adapted groupware[241]) approximates prize funds when the covered functionality is of general interest.

3. Mutual funds permit the mutualisation of efforts between a set of players such as administrations so that they can share the cost of development of a FLOSS solution to a need in a given domain. The community of regional and local public authorities that support FLOSS in France, ADULLACT, as well as the Commonwealth of Massachusetts in the US have put in place such mechanisms. The development of a FLOSS solution is normally more costly than the implementation of a purely internal undistributed solution (the need for better written and documented code, back-office of distribution, etc.). There is a case for public funding to bootstrap this mutual fund. Mutual funds are not restricted to administrations. Some industry organisations such as Embedded Linux and the Open Source Development Lab or even ObjectWeb are similar to this proposal for mutual funds, and when the consortia have open membership and open governance rules, the case for public bootstrapping funding can also apply.

9.5.4. Equitable tax treatment: FLOSS is a donation

FLOSS software development may not be a charitable activity, although a majority of contributors remain independent individual volunteers (see Figure 28, “Distribution of code output by individuals, firms, universities” on page 50). However, when the software is released to the public, it is a charitable donation and treating it as such for tax purposes may be a simple and effective support mechanism – but is in any case, only fair.

It should be noted that IPR donations are commonly used for tax deductions by firms especially in high technology sectors in the U.S.. There has been considerable controversy resulting in a general investigation by the U.S. Internal Revenue Service on the somewhat arbitrary valuations placed by firms on such donations, particularly on donations of patents to universities[242]. *Consumers* can usually deduct donations of packaged software at the purchase price, while software producers are often allowed to deduct donations of software as a share of foregone profits for each copy donated. It should be noted that this is much more than the production cost of software.

[241] BSI’s support of Kolab groupware and Aegypten security systems is document in Nagler, Michael, 2005. “Open Source Adoption of the German Federal Office for Information Security”, *European Commission, IDABC Open Source Observatory,* July 22. Available at http://ec.europa.eu/idabc/en/document/4492/470

[242] See e.g. Feder, Barnaby J., 2002. “Patent Donations Are Novel Corporate Gift”, *New York Times,* November 17 (Finance News). Available at http://www.nytimes.com/ref/open/finance/17PATE-OPEN.html

However, with FLOSS software, a simple lower bound valuation could be the time spent on development. While there are means of evaluating this based on the size of the codebase[243], which could be used as a control on time claims, these "donations" could also be valued on the basis of actual time spent as documented by timesheets.

Two arguments may be made against such tax treatment of FLOSS development: one, a technical point, is that the donation is not made to a specific charitable body, and two, that firms make money out of FLOSS and so their FLOSS development is not charitable at all. The response to the first argument is simple: since the donation of FLOSS is made to *everyone,* it is indeed at least as much a donation as if it were made to a single recognised charitable body.

The response to the second argument, that firms make money from FLOSS and therefore should not be given favourable tax treatment ignores that firms are already given favourable tax treatment for charitable activities that may benefit them indirectly. Indeed, it is arguable that firms do not make donations for the good of society as much as for the good of their shareholders – charitable donations generate goodwill, and in many cases this is directly related to the marketing and public relations efforts of firms (e.g. the environmental and "good corporate citizenship" activities of oil firms). Similarly, if a firm makes money off FLOSS software that it develops by providing services, this in no way reduces the value to society of the firm's charitable donation of the software developed as a public good. Besides, there are a number of firms that develop software in-house to save costs, and choose to release it as FLOSS; they make no money off this, and instead, by providing software that others can use, save other similar firms money[244].

One caveat must be added to this proposal – since a large share of FLOSS software is written by individuals, it is essential that tax incentives recognise this contribution of individuals – rather than firms – to innovation, and allow tax deductions not just for corporate tax but for individuals as well. Since donations are typically tax deductible for personal income tax and not just corporate tax, this should not prove to be a problem, although it may not provide sufficient incentives for the considerable share of volunteer developers in their early stages of contribution who do not have a very high income. Of course, these developers perhaps do not need tax incentives – they are already motivated by a number of things, including the skills development potential of their FLOSS activities. Moreover, the main purpose of this proposal is equitable treatment for FLOSS compared to other voluntary donations, not an equitable treatment of voluntary and paid work.

This proposal comes in three versions, including one (the second) that probably needs, in several jurisdictions, publicity for the use of existing tax law rather than any policy changes:

1. Organisations and individuals should be allowed to treat as a donation for tax purposes the value[245] of the software released under a FLOSS licence. The

[243] Such as the substitution cost measures widely used in the software industry, and described previously in this study in the evaluation of the "primary production value" of FLOSS.

[244] E.g. the Enterprise Application Integration *OpenAdaptor* developed by Dresdner Bank and released as FLOSS, now widely used by a number of other banks.

[245] By "value" in this section we refer to the time spent on writing the software as a lower bound.

advantage of this version is that it would be the easiest to apply in practice. However, it would require significant changes in tax law in order to recognise the concept of donation to "the public" rather than to a specific charitable body. Moreover, a FLOSS licence, while irrevocable, is not legally the same as a transfer of ownership.

2. Organisations should be allowed to treat the value of the software of which the (copyright) ownership is transferred donated to a registered charitable foundation[246] as a donation for tax purposes. There are a number of ways by which the value could be measured, but in principle the (auditable) time spent by developers in writing the software should be treated as a lower bound for the value.

   For this proposal to work in practice, foundations would have to exist that would accept the donated software with the guarantee to the donor that it would always be made available under a specified FLOSS licence (this affects whether donors will want to make donations, it does not change the tax treatment[247]). Donor agreements could be on a rolling basis, so that each line of code does not require a new agreement but is automatically donated.

   Indeed, some such non-profit FLOSS repositories already exist mainly in the U.S. for specific groups of software applications, such as the Software Freedom Law Center and (for Apache web server software projects) the Apache Software Foundation. Existing foundations are selective about receiving software donations due to liability concerns, but similar organisations need not be so selective if donors indemnify the recipients against any legal claims resulting from the use or distribution of the donated software.

   This version is almost certainly consistent with current tax policy in the US[248] and would probably require no tax regulation changes, but only publicity and the existence of suitable recipient foundations and donor agreements to work in practice. In the US, IPR donations are common, especially of patents which are much harder to value. Currently US authorities are investigating widespread abuse of patent "donations" where IP is valued arbitrarily high for tax deduction purposes. Our proposal, on the other hand, provides an auditable means of measuring value. In the US, it is also possible for firms donating copies of packaged software – e.g. Microsoft Office – to deduct from their tax liability part of the foregone sale price of the software. Firms have long valued their proprietary software donations in relation to the retail (or dealer) price <u>for each</u>

[246] Obviously, the foundation would have to be registered with specific charitable goals in order to qualify for special tax treatment, as required by tax regulations.

[247] Thus it would of course be possible to claim such a tax deduction for donating proprietary software to a foundation but this is of course not likely to happen, as the proprietary software would then find it difficult to commercially sell licences to software it does not own.

[248] And to different extents in different European jurisdictions.

copy donated[249]*. Firms also make tax deductible cash donations resulting in purchases of their software. In our proposal, the deduction is much lower and stricter – the copyright to the software is transferred, and valued once; the software itself may then be freely distributed an infinite number of times with no resulting increase in the tax deduction.*

3. As above, but organisations *and individuals* should be allowed to treat as a donation for tax purposes the value of software of which the ownership is donated to a charitable organisation and distributed as FLOSS. This version should also be consistent with current tax policy, though either individuals or tax authorities may have to adapt somewhat. E.g. in countries where such donations cannot be deducted from personal income tax but require the creation of a business, individuals may have to create such businesses. This can be trivial in many EU countries, but impossible in some, such as Belgium. Given the significant contribution of individuals to FLOSS; authorities may need to allow deductions from personal income tax, or making it easier to set up one-person-businesses (which is a good idea for reasons unrelated to FLOSS).

It should be noted that the logic of equitable treatment for in-kind donations of FLOSS applies also to other non-software goods that are donated under such “information commons” schemes, such as music, text, scientific and other creative works distributed under (several, but not all) Creative Commons licences. A control for valuation may be somewhat more difficult for other artefacts where, unlike for software, substitution cost estimation metrics do not exist – but auditable time input at the opportunity cost of the donor’s time can always provide a lower bound for the value of the donation.

9.5.5. Education: avoid lifetime vendor lock-in for students

The reason it seems desirable to promote the use of FLOSS in education (ICT education and more generally all educational activities that have a bearing on the cultural relationship with information technology) is threefold:

1. It is obviously likely to have a strong impact on the future usage of FLOSS products and the build-up of the related skills.
2. It builds up essential ICT skills rather than the knowledge of specific applications from specific vendors (leading to the current locked-in-for-life situation, where vendor lock-in applies not only to organisations but to individuals who have typically not chosen their software but been provided it for free by schools[250]).

[249] For examples see e.g. Langley, Monica. 1997. “High-Tech Companies Battle Over the Value Of Donated Software”. *Wall Street Journal.* Tuesday, September 9; Cha, Ariana Eunjung. 2003. “Microsoft's Big Role on Campus”. *Washington Post,* Monday, August 25 (p A01). Available at: http://www.washingtonpost.com/ac2/wp-dyn/A40000-2003Aug24

[250] Bill Gates acknowledged this effect of software use in the context of the unauthorised use of Microsoft software in China, speaking to students at the University of Washington, “As long as they are going to steal it, we want them to steal ours. They'll get sort of addicted, and then we'll somehow figure out how to collect sometime in

3. It is likely to install an attitude towards information technology that favours the ability to create and actively participate rather than just consume – i.e. the scenarios under which FLOSS is most likely to deliver a strong positive economic and societal impact, by encouraging collaborative prosumer usage and a reflexive attitude on usage and the technology that supports it.

A non-mandatory policy to stimulate the usage of FLOSS in education would encourage students to:

- Have a critical point of view on the different software solutions;
- Understand better what is FLOSS and its differences with proprietary software;
- Possibly become active contributors at various levels to a FLOSS tool.

This policy could take multiple forms:

- Encourage (through European education and vocational training programmes and coordination actions, through input to the harmonisation of diplomas and curricula) Member States to develop policy under which the ICT support to education is FLOSS-based when possible and distinguishes between the concepts of tools and the particular implementation of a piece of software. For instance, learning to use software to edit text, programs or images should always distinguish between the functionality that is constitutive of any editor and the particular sequence of actions that is necessary to obtain a result in one particular piece of software. Of course, specific software applications have to be used in order to teach these skills, but FLOSS applications should be used at least as much as proprietary applications.
- When it has education value, encourage the usage of source code and other forms of public documentation to understand the way in which a difficulty has been solved or functionality is obtained. When it is a relevant to a pedagogical project, encourage modification or extensions to software, and distribution under FLOSS licenses of modified or extended software by schools and teachers. Note that such policy is in no way restricted to ICT or technology education, it is fully relevant to all disciplines including in primary and secondary education (using adequate languages or tools), with a particular attention to artistic disciplines, social and human sciences, maths, natural sciences, literature and foreign languages.
- Continue supporting the dissemination of valuable FLOSS products produced by the educational players themselves, by indexing, directories and repositories actions. These actions have been developed for quite a while within the IST programme and in European education programmes, but will deliver true value only if they are pursued over a significant period with a sufficiently clear positioning with regards to the support of FLOSS and of self-production by teachers, schools and students.
- Fund directly or encourage/coordinate the funding by National programmes of measures that facilitate and reward the self-production by schools, teachers and students of FLOSS

the next decade.” Corey Grice and Sandeep Junnarkar, 1998, “Gates, Buffett a bit bearish”, *CNET News.com*, July 2. Available at http://news.com.com/2100-1023-212942.html?legacy=cnet education resources of European relevance in fields such as European history, artistic education, maths and sciences, foreign languages. Note: though actions of this type were recommended by experts or educational players, the adoption of policy to this effect was light because of the vision defended by publishers that educational ICT is an information publishing market. This has led to concentrate actions on the sole metadata layers (indexing of educational resources). FLOSS-oriented policy can only develop if one acknowledges that beyond educational "content", educational value arises from practice using educational or information resources. In this respect, the availability of large public domain or voluntary-commons pools of documentary resources is a condition for the development of ICT-based education and media literacy.

9.5.6. FLOSS in pre-competitive research and standardisation

The European Commission (DG Research, Information Society and Media as well as DG Enterprise) has a strong tradition of supporting or facilitating industry and academic research fora for standardisation, coordination and support activities. Most of these actions make it possible for participants to invest in setting a future innovation or market creation by creating a FLOSS basis: FLOSS reference implementation of standards and full environments for using the standards, FLOSS libraries and development tools for the creation of experimental services and applications.

In the COST or IHP networks research actions there are a number of successful cases of building a FLOSS basis for future innovation and markets that have resulted in successful business creation: see for instance the DAFX COST action for digital audio effects and the later MOSART IHP network on music performance and instruments. However these remain niche activities and in particular, there are few cases of fora with strong industry participation that take this route. Obviously, industry can't be forced to embrace such initiatives, but the case for a FLOSS approach is so strong in domains where market creation depends on a wide experimentation of services by downstream innovators and even end-users that it would be worth trying to facilitate the creation of such actions by consultation and targeted funding.

In contrast to some standardisation activities, we emphasise that the fora do not have to be industry-wide, though of course participation to them must be open. It is likely that some players who have vested interests in closed or vertically segmented approaches will not be motivated to participate, but this should not be seen as a problem. The past 20 years history of failures in predefining which services will be successful for the wide creation and exchange of information calls for a more open experimentation approach.

9.5.7. Partnerships between large ICT firms, FLOSS SMEs and communities

One key weakness of Europe in terms of ability to derive economic benefits from its strong contribution to FLOSS lies in the weak investment of the large ICT industry in partnerships with FLOSS companies or community groups. In contrast, U.S. companies (IBM, Sun, HP) have developed strong policies in this respect. Just as with the initial development of the Internet and the World Wide Web, it could be an sadly ironic development that Europe, holding such strong assets in this domain (skilled and motivated contributors being the most important) could – in relative terms – fail to build technology and economic growth on these assets and leave it to U.S.-based companies to cash on the future growth of FLOSS-based applications and markets.

As in the previous recommendation, a more pro-active strategy of partnering with FLOSS companies and sponsoring community projects can only result from the large European ICT industry itself. But it can be encouraged and supported when it develops.

Note that we do not limit ourselves by “partnership” to only (or even mainly) formal arrangements here. We include the full range of ways of support that go from formal and financial when communities have legal bodies to represent them (Apache, Plone, and many others have registered foundations). FLOSS foundations (and foundation run educational programs) could enter partnership or joint trust agreements with sponsoring firms. In addition to training, and licensing of existing software, foundations might undertake to support work on the kernel of new projects that would address the need of SMEs in a certain branch of business in a given region—so they could contract with regional business associations, rather than individual firms.

There are also simpler arrangements without involving legal bodies representing communities, of firms hiring developers or community organizers who are allowed to keep contributing to the community on company time. For example, a growing set of FLOSS-specialised companies and even the non-FLOSS solution providers already support community projects (beyond those projects that are central to their own business) by letting their programming staff contribute to these projects. In general, Europe, this does not match the level of U.S. firms such as Google which hires key FLOSS developers with a day per week explicitly allocated to free contributions to FLOSS projects. Some European companies such as France Telecom have hired part-time developers or FLOSS community managers while agreeing to support their continued contribution to their projects during work.

Indeed, as shown in Figure 53, “FLOSS development allowed on employer time” on page 114, 26% of “high IT intensity” European firms that use FLOSS “totally agree” that employees could contribute to FLOSS projects during their time at work, as long ago as 2002. But this is rarely explicitly supported, by company policy or work contracts. One could imagine such schemes being explicitly supported by allowing ICT companies to account for the corresponding cost as a sponsorship to public interest actions, or by partially subsidising such contributions (if this does not give rise to excessive management costs). The interest of such schemes lies also in the cultural dissemination that occurs in the sponsoring company, which sets the ground for more strategic partnerships. Certainly, treating employee time donated to development of donated FLOSS software as deductible for tax purposes (Section 9.5.4) should provide a considerable incentive for such behaviour.

For true industry FLOSS-based strategies to develop, one will need much deeper investments where support to projects and partnerships are done on company-strategic technology or business segments. Experience shows that when this occurs, the companies no longer need public support and even do not desire it since such support requires disclosure obligations and time constraints that they can't accept in their strategic activities. But public action can prepare the ground for this stage, and for this, it must move to a more pro-active questioning of ICT-industry strategies. Organising domain specific seminars with the joint participation of ICT industry strategists and high-level managers, core teams of FLOSS developers and information society policy analysts could be productive in this respect, if the seminars are prepared by background papers that identify choices that lie ahead in each of these domains. Indicative examples of domains where such actions could be useful would be: end-user development, development tools for lightweight applications, media software for large groups of contributors, collaborative tagging and filtering of large information spaces, small business extensible applications, etc.

It should be noted that the recently concluded[251] CALIBRE project, funded through the EU's FP6 research programme under DG Information Society, has run a series of such workshops for "secondary software" industries[252] from communications to automobiles. This included the creation of the CALIBRATION industry forum, and the holding of intra-company workshops in organisations such as Vodafone, Eurocontrol and Philips Medical Systems. However, this project has focussed on mainly technical aspects and thus involved mainly technical personnel from ICT firms already involved with FLOSS development. We suggest supporting similar engagement at a strategic business level, to attract and generate awareness among large firms who are not yet aware of the potential of cooperating with FLOSS SMEs and communities.

Actions targeting partnerships between large ICT companies and FLOSS SMEs can be complemented by encouragement to networking between FLOSS SMEs that are active in a similar technological domain. One interesting proposal for such collaboration is "50+: The Lisbon Trust"[253] prepared by members of the Zea network of FLOSS SMEs, which suggests a specific set of activities including a "software conservancy" and a platform for informal and formal collaborations between large and small European firms, research institutions and the developer community.

Finally, public agencies at the EU, national, provincial and local levels follow the example of private firms and explore formal and informal partnership arrangements with FLOSS foundations and specific project communities to provide "localizations" and special purpose software systems for their own business management needs. The healthcare sector offers a good example of an under exploited FLOSS system that could be adapted for EU application: the US Veterans Administration has the largest FLOSS software system for management of outpatient and hospital records, which was created and expanded over more than a decade of work by over 100 programmers, and is serving 20 million patients. Studies on the interaction between public administrations and the FLOSS communities, including practical guidelines for governments and developers on how to form partnerships, were prepared for the European Commission's IDABC unit and are available on the EU's Open Source Observatory[254].

9.5.8. Unbundling in the hardware / software domain

Already in 2001, a consultation meeting of the main European and global FLOSS players identified the bundling of PC hardware together with operating systems as the main factor

[251] Ending September 2006.

[252] See http://www.calibre.ie/

[253] Prepared by Paul Everitt, reviewed by CALIBRE project members.

[254] http://ec.europa.eu/idabc/en/document/3879/471

blocking the deployment of FLOSS for home users[255]. The situation has only marginally evolved with regards to PC hardware: the availability of home user PCs (which are today increasingly laptops) sold with an option of pre-installed FLOSS operating systems remains limited to a token level, and there are still clear signs of strong pressure exerted through OEM agreements against the development of such offers. The situation is more open in the field of acquisition of PCs by organisations, though except where a specific effort has been made or where there was pressure from public opinion, it remains unusual to acquire PCs and operating systems separately. The main transformation that has occurred since 2001 is that bundling in the hardware / software domain is now an issue for a much wider set of devices: game stations, music players, set-top boxes and soon digital assistants, mobile phones and controllers in machines. Though a significant and growing share of the three first types of devices are running GNU/Linux or other FLOSS software, they often do it in concrete situations where the end users or third parties are denied the practical ability of exerting rights granted by the FLOSS licences of adapting / modifying the software.

The complexity of the overall situation calls for a careful choice of the policy mechanisms that could be used to limit the clearly anti-competitive lock-in effects of bundling between devices and operating systems. Priority could go to actions targeting general purpose IT devices and more generally devices that are used to acquire, create, and exchange information and media between end-users. The reasons for these priority choices are that these are the cases where a greater openness to end-user innovation has the clearest benefits (following the scenario analysis in Section 9.4, “Scenarios”).

As for the types of policy actions that could be used, the apparent obvious choice of an active competition policy does not in practice always work, because of its essentially corrective nature, when efficient policy measures have to be of a preventive nature or at least to have direct effects. However, the prospect of strong corrective competition measures remains an absolutely essential tool to prevent excessive pressure by monopolistic software providers against unbundling by device integrators. Similarly, procurement policy which has proved to be efficient within organisations whose buying power is sufficient, is of course very useful but does not impact sufficiently on the home user market, though the case of employer-provided employee laptops presents an interesting intermediate situation. Consumer policy could have a very significant impact by simply requiring the clear labelling of the share of the price that goes to pre-installed software. Moreover, as noted in the report of the FLOSSPOLS study funded by the European Commission’s DG INFSO under the FP6 programme, public procurement for software among European government authorities is anti-competitive, and often explicitly so[256].

Beyond these aspects, it is the coordinated action with consumer groups that would lead to the most significant impact. These groups are today increasingly aware of software proprietary control issues, and of their impact on price (superior to copyright related levies). The recent development of the Ubuntu GNU/Linux distribution has considerably simplified the

---

255 See *Public report on the consultation meeting on European perspectives for open source software*, ftp://ftp.cordis.europe.eu/pub/ist/docs/ka4/tesss_oss-report.pdf

256 See section 5 in Ghosh, R.A. 2005c. “An economic basis for open standards”, FLOSSPOLS project report, *European Commission DG INFSO.* Available at http://flosspols.org/deliverables/FLOSSPOLS-D04-openstandards-v6.pdf#search=%22ghosh%20open%20standards%20economic%22

installation of FLOSS platforms, both for professional providers and for end-users. For those users who remain locked in at platform level, the growing usage of FLOSS Web browsers and office suites provides an intermediate step that can be made without too much disruption of usage habits. It seems that simple inciting actions under the umbrella of "freedom of software choice" could have a powerful impact. The success of these actions is connected to the development of a more general change in the relationship between individuals (consumers, citizens) and information technology, which is the object of the next policy recommendation.

9.5.9. Working towards a prosumer-based information society

A key economic and societal asset of Europe lies in the demanding attitude of its consumers with regards to quality. Development such as fair trade, organic food, environmental quality in buildings and products, responsible consumption, etc. highlight that quality today is no longer perceived as a purely functional property of products but includes elements related to production modes, social aspects, ecology and the environment. While sometimes perceived as an annoyance by industry, such consumer activism has often led to entirely new lines of innovation and increasing economic potential in Europe (e.g. in alternative energy or organic food production). Providing quality in this wide sense to consumers is proving everyday to be a key competitive advantage in the internal market at least in fields where the diversity of offer and an adequate level of information among consumers exist. There is often a tension between quality and the focus on price, particularly in domains where large-scale distribution and TV-based commercial advertising play a role.

What is the corresponding situation in the field of information technology and software? There exists in this domain conflicting definitions of quality: between strong integration of components and decoupling that permits choosing the best offer for one job, between "we do it for you approaches" and "we give you power to choose who does it and how", between easy to learn and powerful. These tensions are here to last: the transformation of consumers into prosumers[257] (producers as well as consumer of information, with all types of intermediate situations such as prescription, evaluation, critical assessment, recommendation to others, amateur activities, etc.), that seems to be a key potential of the information society, will take time, require innovation at the technology as well as the social and cultural level. Such a transformation cannot be decreed, it can only be facilitated by the encouragement of an adequate experimentation ground.

There is a particular challenge in developing non-commercial information exchanges applications whose users are prosumers[258]: when these applications take off[259], they can generate significant economical growth, but before that stage, they are hard to predict, their business models are uncertain, and the fit between a particular technology and their development is also hard to predict. This signals a clear case for public policy actions when the creation of an

---

[257] The term was created bin Toffler, Alvin, 1980. *The Third Wave.* Bantam, New York.

[258] See analysis in the wider impact section of the impact analysis.

[259] Examples: Internet and Web applications, SMS, and more recently blogging (Technorati, Blogger, Skyblog), collaborative sites such as Flick'r or MySpace, collaborative media such as OhMyNews or Agoravox and emerging Web 2.0 services such as Netvibes.

experimentation ground does not result from the spontaneous strategies of industry and other stakeholders.

The basis for experimenting with new information exchange services is threefold:

- To support the creation of adapted underlying open technology infrastructure, where devices, development tools and libraries are available under FLOSS licenses;
- To support experimentation with services without the immediate requirement of economic sustainability, but with strong requirements of end-user relevance.
- To network participants in particular with the means to fund such services or derive economic value from them with a wide spectrum of possible approaches: added-value services assisting users to move across the continuum towards more professional roles, subscriptions to services and community memberships, non-conventional mutualisation schemes (legal fees-based, donations-based).

These actions could be developed in fields that are fit for prosumer approaches, such as grassroots and mixed publishing/collaborative media, collaborative sites for video interchange, ICT-based participative democracy, knowledge sharing, etc. They require funding of a nature that can proceed at the EU level from the R&D programmes, the structural funds, ICT-related actions and some parts of the e-Ten programme. In all these fields, there have been some activities of this nature, but without a clear structuring perspective.

9.5.10. Recognise the legal approach of commons-based schemes

Commons-based schemes for software and information have been formulated using available licensing possibilities within the existing framework of copyright law, where a permission notice or a contractual arrangement is used to implement in practice a commons status for the covered works. This approach was very successful, since it has proved to be possible to adapt it to different entities (software, information, media), and to radical changes in the scale of its usage and the nature of the players using it. However, such a definition of information commons by licensing remains but a simulation of a situation which it is important to make explicit: one contributes to the commons something that one has produced or created, possibly using other entities that already belonged in the commons. This is a unilateral gesture that does not require the agreement of anyone else. Then others, any others, are free to draw things from the commons, to use them or build modified or new entities. Possibly, those using the commons have responsibilities towards them, at least when this usage impacts the public sphere.

There are several reasons for which a better and more explicit recognition of commons-schemes in the foundations of the intellectual rights environment[260] would be useful or even necessary:

[260] By intellectual rights one means here the rights that regulate both permissions and restrictions of usage of intellectual entities, that is in the present material law: intellectual property titles and their enforcement mechanisms, limitations, exceptions or fair use doctrines, as well as the public domain.

Without such as recognition, the evolution of substantive law (in particular copyright and author rights, patents, some sui generis rights and their execution mechanisms) struggles to take in account the requirements of information commons. It has to proceed by discussing a possible impact of a change on some activities and players, without considering the substance of the scheme underlying all these activities. Recent examples where this has given rise to much confusion are the debates on software patent ability, on the protection of technical protection measures against circumvention, the database directive (with regards to open access publishing), etc.

An explicit recognition would help to clarify a number of elements linked to contract and consumer law (in particular for liability and warranty). It would make clearer that one can have a combination between an absence of liability and warranty[261] for contributing to the commons, and an adequate level of liability and warranty for products and services that incorporate elements from the commons. The confusion that surrounds these questions has acted to some extent as a disincentive to contributing to the commons for risk-adverse organisations.

An explicit recognition in the legal foundations would have a strong pedagogical dimension, and would act as a potential motivator to contribution.

Such recognition is potentially non-controversial as it does not require any change to the substantive definition of IP titles. It would create a better legal certainty for investment in commons-based approaches. Such recognition is most likely to proceed at the level of global international arenas. In particular in WIPO, a number of steps have been taken, at the initiative of NGOs[262] as well as WIPO's own initiative[263]. In other UN agencies such as WHO[264] and UNCTAD there have also been significant steps.

Up to now, the attitude of the EC and the EU presidencies have been relatively unsupportive of these initiatives, which is surprising considering the leading role that Europe has in contribution and some forms of deployment of the software commons. A positive recognition of commons schemes must be done with a true respect for the specific identity and motivation of commons-based innovation and creation: one must not artificially project concepts from restrictive property-like rights on commons-schemes that are based on a universal notion of ownership by humanity. There are some interesting precedents in other areas of international law that could serve as precedents, regarding geographic areas (Spitzberg, Antartic) and global public goods (climate, environment, biologic diversity, cultural diversity). The governance of the commons also requires different settings than the care for IPR: for information goods and when humanity is the stakeholder, the role of States at all levels becomes one of trustees caring for the existence conditions of the commons and societal governance is essential.

---

261 Except of course for intentional damage or unauthorised contributions of entities to which one does not hold the corresponding rights.

262 See for instance the Geneva Declaration on the Future of the World International Property Organisation, and the proposed draft treaty on Access to Knowledge.

263 Development Agenda.

264 See recently adopted resolution on a new framework for medical R&D.

10. Glossary

In order to provide a quick reference to the terms and names used in this report, we provide in this section a brief overview of the sectors of the software market, the main FLOSS products, and the major firms providing FLOSS support.

10.1. Software market terminology and overview

The software market is divided into three broad categories, on a dimension defined by the method of business and development.

According to business and development, the software market includes:

Packaged software

This is software that is acquired "off-the-shelf", and is the main area where proprietary software plays a role as a business model for vendors. Off-the-shelf software can be widely used and cheap – Microsoft Windows, or Mandriva Linux – or rarely used and expensive – enterprise applications from SAP, or geoinformatics software from ESRI. Either way, the software itself is not specially made for the customer. Thus, packaged software is also often proprietary, as it is easy (in business terms) to make the product, "own" and package it, and sell it. Of course, FLOSS can also be packaged software, whether packaged by a firm – as with Mandriva Linux – or downloaded by users of the Internet, as with Mozilla/Firefox or Debian GNU/Linux. In this case, too, users may use the software off-the shelf without modification – although unlike proprietary packaged software, users can modify FLOSS packaged software.

While much of the discussion of the software market revolves around packaged software – which includes most proprietary software – this sector accounts for only 19% (EU) and 16% (US) of the software market, by revenues. Packaged software publishers account for under 6% of employment of software developers in the US (although other firms employing software developers may also release some packaged software).

Custom software

This is software specially written for the customer, generally by an external firm that provides software development, customisation, integration or consultancy services. Custom software is in itself not proprietary nor necessarily FLOSS, though it often shares all the attributes of FLOSS – typically, customers own rights to the software developed for them, so they are free to use, study, modify or distribute it, just like customers of FLOSS software. Due to this, custom software is quite compatible with FLOSS, and custom software developers can reuse FLOSS software without paying any royalties.

Custom software accounts for 52% (EU) and 41% (US) of the software market by revenue, and firms providing software development, customisation, integration and consultancy services account for 37% of employment of software developers in the US (although such firms may also release some packaged software, and other firms employing software developers may also develop custom software).

In-house (own account) software

In-house software is developed by internal employees of an organisation and is thus hardest to account for. As with custom software, it is neither FLOSS nor proprietary, but has all the attributes of FLOSS as it is owned by its users, and is thus also suitable to be developed by customising or building upon existing FLOSS code.

In-house software accounts for 29% (EU), 43% (US) and 32% (Japan) of the software market by value, and for about 57% of total employment of software developers in the US (measured by counting all software developers employed at firms that are not software publishers, consultants, integrators or service providers; of course, some firms in other sectors may also create some packaged or customised software, and several software developers may be creating in-house software at software publishers, consultants, integrators and service providers).

*Apart from the above classification of software, there is also a classification of the "**secondary software**" market.*

The **secondary software sector** comprises firms whose main business is not software development or service provision, but something else – such as hardware manufacturing – but who nevertheless develop a lot of software. Most such software would be classified under *in-house* software above, while some is custom software, or even packaged software. There is also **embedded software**, which is incorporated into hardware and rarely (if ever) receives direct user interaction – for example, the software used in cars, for fuel injection control or anti-skid braking systems. More details about this, and the types of software usually developed, can be found in section 7.7.2.

Major FLOSS application areas include databases (MySQL and Postgres), operating systems (GNU/Linux, FreeBSD), web servers (Apache), scripting languages (Perl, PHP) and office applications (Firefox, OpenOffice). For brief descriptions of these applications see the glossary of technical terms, below.

Major firms providing substantial FLOSS products or services include large firms such as HP, IBM, Novell, Oracle, and Sun Microsystems (which provide extensive FLOSS software product, support and/or integration services, and support FLOSS platform for their proprietary packaged software). Smaller firms include Mandriva (Linux systems and services), MySQL (databases), Red Hat (Linux systems and services), and Trolltech (graphical user interfaces, embedded applications such as for cellular phones). Several other firms providing integration, support, business consulting and/or hardware offer extensive FLOSS-related support services, such as Cap Gemini and Unisys. A number of firms use FLOSS on their popular hardware offerings, including Cisco (Linksys routers run on Linux), Motorola (which sells lots of Linux-based phones, mainly in China) and Nokia (which has released Linux-based products, and also develops and releases FLOSS software).

10.2. Glossary of technical terms

Apache HTTP Server

A free software/open source web server for Unix-like systems, Microsoft Windows, Novell NetWare and other operating systems. Apache is notable for playing a key role in the initial growth of the World Wide Web, and continues to be the most popular web server in use.

Debian

A widely used and the largest distribution of FLOSS software, collated, quality checked and maintained by the Debian Project.

Digital Rights Management (DRM)

Any of several technologies used by publishers (or copyright owners) to control access to and usage of digital data (such as software, music, movies) and hardware, handling usage restrictions associated with a specific instance of a digital work.

Firefox, Mozilla.

A free, open source, cross-platform, graphical web browser developed by the non-profit Mozilla Corporation and hundreds of volunteers.

Free Software *see FLOSS*

FLOSS (Free/Libre Open Source Software)

Software whose users have the right ("freedoms") to use, study, change, and improve its design through the availability of its source code and the right to distribute the changed program. These rights are usually granted through a copyright licence. FLOSS, free software, libre software, and open source software are different terms to describe the same software, licenses and software development models, though there are some differences in ideological emphasis between supporters of the term "free software" and "open source". Some proponents of the latter believe that open source should be chosen for practical reasons, and some proponents of the former believe that free software should be chosen simply for the freedoms it offers.

Free Software Foundation (FSF)

FSF is a non-profit corporation founded in 1985 by Richard Stallman to support the free software movement ("free" as in "freedom"), and in particular the GNU project.

General Public License (GPL)

The GNU General Public License is a widely used free software license. The GPL grants the recipients of a computer program the following rights: the right to run the program for any desired purpose, the right to study how the program works, and modify it (access to the source code is a precondition for this), the right to redistribute copies, the right to improve the program, and release the improvements to the public (access to the source code is a precondition for this).

GNOME

A graphical user interface environment for GNU/Linux distributed under a FLOSS licence.

GNU

A free operating system consisting of a kernel, libraries, system utilities, compilers, and end-user applications. Its name is a recursive acronym for "GNU's Not Unix". The way most people use this is in the GNU/Linux distribution.

GNU/Linux

The preferred name, for many people, for the operating system more commonly known as “Linux”. Linux is in fact only the kernel of the operating system, and most of the libraries, system utilities and several other system applications required for Linux to be used are part of the GNU system.

KDE

A graphical user interface environment for GNU/Linux distributed under a FLOSS licence.

Kernel

The core of an operating system, which allows user applications to access devices, the file system, memory and other system resources.

Linux (also known as GNU/Linux)

A Unix-like computer operating system. It is one of the most prominent examples of open source development and free software; its underlying source code is available for anyone to use, modify, and redistribute freely.

Linux distribution

A Linux distribution is a version of a Unix-like operating system for computers comprising of the Linux kernel, all or part of the GNU operating system, and assorted libraries and applications. Linux distributions take a variety of forms from fully featured desktop and server operating systems to minimal environments for use in embedded systems or for booting from a floppy.

MySQL

A multithreaded, multi-user, standard query language (SQL) database management system (DBMS).

Network neutrality

The principle that network customers pay for a quality of service and not for a specific set of applications. Thus, network service providers provide different levels of service quality (bandwidth) but not restrict specific application domains (such as P2P or Voice over Internet Protocol, also known as Internet Telephony). Network neutrality has been credited with the historical success of the Internet, as it allows the development of innovative new protocols and application domains.

OpenOffice

OpenOffice.org is a FLOSS office suite application available for many different operating systems including Microsoft Windows, Linux, Solaris and Mac OS X. It is intended to be a compatible alternative to Microsoft Office. It supports the OpenDocument Format standard for data interchange.

Open Source Software *See FLOSS*

Open Source Initiative (OSI)

A non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community, specifically through the OSI Certified Open Source Software certification mark and program.

Operating system (OS)

A computer program that manages the hardware and software resources of a computer. Examples of operating systems for personal computers include Mac OS, Microsoft Windows, and GNU/Linux.

P2P – Peer to Peer

Technology that allows individual computers to act as both client and server, thus allowing the distribution of files without any centralized storage structure (without P2P, files are typically stored on a set of computers, known as servers – belonging to the data supplier – accessed by computers known as clients – belonging to customers). P2P technology replaces the hub-and-spoke architecture of a client/server architecture often leading to large efficiencies in bandwidth and storage use. It can also make the sharing of unauthorized copies of files (such as music) easier as it does not have a central point of control.

Perl, PHP

Perl and PHP are popular *scripting languages* – programming languages designed for rapid prototyping and customization. They are frequently used for applications running on websites. Development environments for both languages are released under FLOSS licences, as is much of the software written in these scripting languages.

Project (or FLOSS project)

A project is a (sometimes temporary) endeavor undertaken to create a specific software product.

Repository

A place where large amounts of source code are kept, either publicly or privately. They are often used by multi-developer projects to handle various versions and developers submitting various patches of code in an organized fashion.

RSS (syndication, syndicated blogs)

Allows visitors to a website to "subscribe" to an "RSS feed", which ensures that they automatically receive updates (or notifications of updates) whenever the content of the website changes. In addition to website visitors, other websites can also subscribe to an RSS feed, allowing them to automatically reproduce the content from the original website as it is updated. This is typically used by "syndicated blogs" – popular blogs that provide RSS feeds to allow other blogs to automatically include their content, as with news syndication.

Source code

Any series of statements written in some human-readable computer programming language. A computer program's source code is the collection of files that can be converted from human-readable form to an equivalent computer-executable form. The source code is either converted into an executable file by a compiler for a particular computer architecture, or executed on the fly from the human readable form with the aid of an interpreter. It is very difficult, if not impossible, to change a program, or understand how it works, without access to the source code.

Sourceforge

SourceForge.net is perhaps the largest globally accessed ***repository*** for software developers to control and manage open source software development, and acts as a source code repository.

Unix

A computer operating system originally developed in the 1960s and 1970s by a group of AT&T employees at Bell Labs. Today's Unix systems are split into various branches, developed over time by AT&T, as well as various commercial vendors and non-profit organizations.

11. Appendix 1: A Formal model description

In order to accommodate the idea of the continuing division of economic activities to meet more and more specific needs that lead to productivity growth at the aggregate level, we use the Ethier function to define the effective capital stock in function of cumulative consumption foregone (i.e. the stock of 'raw' capital) and the degree of specialisation enabled by the ICT capital stock used in the final output sector:

$$K^e = \left(\int_0^{AP} (x_i^P)^\beta \, di + \int_0^{AF} (q \cdot x_i^F)^\beta \, di \right)^{1/\beta} \quad (1)$$

In equation (1), K^e is the effective capital stock, while *A* is the total number of different economic activities. *AP* of these activities are supported by PROPS and *AF* by FLOSS, hence *A=AP+AF*. *q* is an index that represents the influence of quality differences in FLOSS and PROPS in turning physical capital into an effective input into the CES aggregator function. x_i^F is the amount of physical capital per FLOSS supported economic activity, while x_i^P is similarly defined but then for PROPS supported activities. β is a constant parameter.

Final output *Y* is produced according to the following Cobb-Douglas technology:

$$Y = B \cdot ((1-u) \cdot h \cdot L)^{1-\alpha} \cdot (K^e)^\alpha \quad (2)$$

In equation (2), u is the fraction of time spent on human capital formation as in Lucas (1988), while *L* is the size of the population. *h* is the average human capital stock per person. α is a constant parameter in between zero and one and *B* is a positive scale parameter. Note that equations (1) and (2) taken together define a slightly more general form of the Romer (1990) production function, first because we have subdivided the set of all activities into two sub-sets containing different types of productive activities, i.e. those supported by FLOSS and those by PROPS, and secondly because the elasticity of substitution between intermediates is independent of α.[265]

[265] In the standard Ethier function also used in Romer (1990) the elasticity of substitution between intermediates is given by $\sigma = 1/(1-\alpha)$. It follows that if intermediates are perfect substitutes (i.e. $\alpha = 1$), LOV

The macro-economic budget constraint must satisfy:

$$Y = c \cdot L + I + R \qquad (3)$$

where *I* represents current investment in physical capital and *R* represents current resources spent on aggregate ICT investment. *c* is consumption per head, and *L* is the number of heads (also the size of the labour force). The growth of physical capital is equal to net investment, which in turn equals gross investment minus depreciation. So the physical capital stock grows in accordance with $\frac{dKy}{dt} = I - \delta^y \cdot Ky$, where *Ky* represents the stock of physical (non-ICT) capital used in all final output producing activities taken together. δ^y is the corresponding rate of depreciation of physical capital. The aggregate ICT stock grows in accordance with $\frac{dKi}{dt} = R - \delta^i \cdot Ki$, where *Ki* represents the stock of ICT capital and in which δ^i is the corresponding rate of depreciation of ICT capital.

A fraction *v* of the stock of *Ki* is assumed to be used in human capital formation, while the remainder is used in final output production. A fraction *w* of that remainder is PROPS based and a fraction *1-w* is FLOSS based. Hence, *v* is an implicit measure of the use of ICT in human capital formation, while *w* is a direct measure of the relative importance, spending-wise, of PROPS use versus FLOSS use.

As stated above, we follow the behavioural approach and assume that:

$$R = s^R \cdot Y \qquad (4.A)$$

$$I = s^I \cdot Y \qquad (4.B)$$

$$u = \overline{u}\ ,\ v = \overline{v}\ ,\ w = \overline{w} \qquad (4.C)$$

where $\overline{u}, \overline{v}, \overline{w}, s^R, s^I$ are all exogenously given numbers in between zero and one. So the investments in ICT capital (*R*) and in non-ICT capital (*I*) are fixed fractions of final output. Although these assumptions are shortcuts compared to the endogenous growth models, actual data show fairly stable gross investments rates as fraction of GDP (s^R) as is shown in Figure 56 on page 152. Software investments as fraction of GDP (s^I) show and increase until the late 90s but seems to stabilize since then.

---

doesn't work, since, roughly speaking, $Y = L^{1-\alpha} \cdot \int_0^A x_i^\alpha di = A^{1-\alpha} \cdot L^{1-\alpha} \cdot K^\alpha$, indicating that in that case A doesn't influence Y. This is intuitively plausible, as a perfect substitute for some good or service, can hardly be regarded as another variety of that good or service. When α gets closer to zero, the elasticity of substitution approaches the value one, and the growth impact of an expansion of variety increases accordingly.

The fact that FLOSS comes cheaper than PROPS is in part reflected by the assumption that *q>1*, i.e. FLOSS can be thought to augment raw capital (more than PROPS). In addition, the tailor-made characteristics of FLOSS can be modelled by assuming that, ceteris paribus, the growth rate of the number of varieties depends positively on that of the labour force measured in human capital units as well as on the ICT-capital intensity in the FLOSS and PROPS groups (to express the notion that FLOSS enables the creation of (even) more variety (than PROPS)), next to an exogenous term that one can think of as being linked to R&D activity (not modelled explicitly). Hence, we postulate:

$$\hat{A}^F = \psi_0^F \cdot (\hat{h} + \hat{L}) + \psi_1^F \cdot (\hat{K}_i^F - \hat{h} - \hat{L}) + \psi_2^F \qquad (5.A)$$

$$\hat{A}^P = \psi_0^P \cdot (\hat{h} + \hat{L}) + \psi_1^P \cdot (\hat{K}_i^P - \hat{h} - \hat{L}) + \psi_2^P \qquad (5.B)$$

where a hat over a variable denotes its instantaneous proportional growth rate, and where $K_i^F = (1-w).(1-v).K_i$, $K_i^P = w.(1-v).K_i$. Furthermore, the ψ_j^i 's for *i=F,P* and *j=0,1,2* are constant and non-negative parameters. Equation (5) states that for equal sizes of cumulative ICT resources allocated to FLOSS and PROPS using sectors, the set of FLOSS using producers will be able to be more specialised than the PROPS using producers, for certain parameter constellations.

Finally, for the production of human capital, we postulate, as in van Zon (2001), that:

$$\frac{dh}{dt} = \pi \cdot (u.h)^{\gamma} \cdot (v \cdot K_i)^{1-\gamma} \qquad (6)$$

where π is a constant parameter reflecting the productivity of the human capital accumulation process. $0 \leq \gamma \leq 1$ is also a constant parameter. For $\gamma = 1$, (6) reproduces the human capital production function of Lucas (1988).

In fact (6) implies that:

$$\hat{h} = \pi \cdot u.^{\gamma} \cdot v^{1-\gamma} \cdot (K_i / h)^{1-\gamma} \qquad (7)$$

Hence, (7) states that the rate of growth of human capital accumulation will be constant, for constant allocations of time and ICT resources and if the ICT/human capital stock ratio is constant.

The purpose of the exercise is now to relate the conditional growth performance of this economy to the values of $\bar{u}, \bar{v}, \bar{w}, s^R, s^I, \psi^F, \psi^P$ and *q*, where *q* represents the difference in quality between FLOSS and PROPS use.

In order to simplify the model somewhat, we can first use the symmetry implied by (1), i.e. all users belonging to a certain group of software users would produce exactly the same amount of output, and hence use exactly the same amount of raw capital *x* per activity. In real terms, output would be maximised by requiring that $\frac{\partial Y}{\partial x_i^F} = \frac{\partial Y}{\partial x_j^P}$, which is, under the usual symmetry assumptions with respect to marginal intermediate cost as well as their contribution to output/effective capital, what the market would also prescribe when left on its own. Hence, using (1) and (2), we have that:

$$\frac{\partial K^e}{\partial x_i^F} = \frac{\partial K^e}{\partial x_j^P} \;\Rightarrow\; x_i^P = q^{-\beta/(1-\beta)} \cdot x_i^F \qquad (8)$$

As one would expect, equation (8) states that activity levels for the group of PROPS users would be lower the higher the (implied) quality difference (*q*) is between FLOSS and PROPS, as $0 \prec \beta \prec 1$. Using (8), and the full employment condition for 'raw' capital K (i.e. cumulative consumption foregone), we can solve for an individual value of x_i^P, and then using (8) again, we can simplify (1) again. Thus, we find that:

$$K = A^F \cdot x^F + A^P \cdot x^P = \left(A^F + q^{-\beta/(1-\beta)} \cdot A^P\right) \cdot x^F \qquad (9)$$

In equation (9), x^F and x^P represent the common levels of within group 'raw'capital use. Obviously, (9) can be used directly to find x^F in terms of *K*:

$$x^F = \left(A^F + q^{-\beta/(1-\beta)} \cdot A^P\right)^{-1} \cdot K = \varphi^F \cdot K \qquad (10)$$

where φ^F is implicitly defined by (10). Equations (10) and (8) then imply:

$$x^P = K \cdot q^{-\beta/(1-\beta)} / \left(A^F + q^{-\beta/(1-\beta)} \cdot A^P\right) = \varphi^P \cdot K \qquad (11)$$

where φ^P is implicitly defined by (11). Substituting (10) and (11) back into (1), we obtain:

$$K^e = K \cdot \left(A^F \cdot \left(q \cdot \varphi^F \right)^\beta + A^P \cdot \left(\varphi^P \right)^\beta \right)^{1/\beta} \qquad (12)$$

Output growth can now be seen to depend on the stock of ICT and the FLOSS and PROPS distribution, by first substituting (4) into (12) next to the definitions for φ^F and φ^P (as implied by (10) and (11)), and then plugging the result into (1) and solving the ICT-expanded Solow growth model defined by the expanded production function thus obtained, and the various saving and corresponding capital accumulation equations as given by (4). By shocking the values of the exogenous variables $\overline{u}, \overline{v}, \overline{w}, s^R, s^I, \psi^F, \psi^P$ and *q*, and subsequently simulating the model numerically forwards in time, we can learn more about the corresponding level and growth-effects of such shocks. We have to use such a numerical exercise, as it is impossible to obtain a closed form analytical solution, even with this simple ICT-expanded Solow growth model.

It should be noted that in our model human capital is just one of the sources of growth, next to LOV. Furthermore, as we have constant saving rates, it follows that in the long run the ratio of the ICT capital stock relative to the human capital stock will have a tendency to grow, leading to accelerating growth of human capital. If one doesn't want this, one simply has to set $\gamma = 1$ in equations (6) and (7), hence excluding a priori one of the channels through which ICT may generate growth. An alternative to such a measure would be to provide 'growth leakages' in equation (7), for example by having decreasing returns to scale in (6), or by decreasing *u* and *v* in such a way that *h* and *Ki* will not show accelerating growth. As this complicates the analysis considerably, we will investigate the effects of shocks in the various system parameters by measuring the induced changes in the system variables relative to their base-run values (that may include accelerating growth therefore).

11.1. Parameter values

Table B.1 presents the values of all parameters in model as described in 9.1. Note that the investments rates in software (as percentage of GDP) differs between the US and EU15 (2% and 1%, respectively) and that investment rates in non-ICT also differ (13% and 14%, respectively) since total investment rates are equal to 15% in both regions.

The parameter ψ_2^F, which denotes autonomous productivity growth through the creation of varieties in the production process, is used to calibrate the model for both regions as to replicate basic labour productivity growth rates as presented in e.g. van Ark, Melka et al. (2002). Unless stated otherwise in some experiments, all other parameter values are the same for both regions.

Param	Value	Param	Value	Param	Value	Param	Value	
							US	EU15
L_0	1	h_0	**2**	$\bar{u}$	0.1	ψ_2^F	0.6	0.25
$\hat{L}$	0	δ^i	0.05	$\bar{v}$	0.1	ψ_0^P	**1**	**1**
B	1	δ^y	0.05	$\bar{w}$	0.8	ψ_1^P	**0.1**	**0.1**
α	0.6	π	**0.025**	$\bar{q}$	**1**	ψ_2^P	0.6	0.25
$K_{i,0}$	1	γ	**0.9**	ψ_0^F	**1**	s^R	0.02	0.01
$K_{y,0}$	7	β	**0.5**	ψ_1^F	**0.1**	s^I	0.13	0.14

Table B.1. Parameter values

12. Appendix 2: Report on user-level productivity and relative cost of FLOSS / proprietary software

This document reports of the analysis of costs on some European organizations which have performed a migration to Open Source software. This work applies the framework defined in the deliverable "Report on the methodology used to collect data." It presents a comparative analysis of the costs incurred during a migration and a prediction of future costs of ownership.

This document presents the application of the framework of costs of a transition toward Open Source software (OSS) defined in the report "Report on the methodology used to collect data" on some European organizations. It reports of analogies and differences among different European organizations.

The deliverable first introduces the framework, it summarizes the results across the organizations, and then it discusses the results organization by organization.

12.1.1. A Brief Overview of the Framework

For a better comprehension of the results we briefly review the framework defined in the deliverable "Report on the methodology used to collect data."

Following existing literature ([Winslow, 2004], [Gartner, 2003], [Linux ROI]), costs are subdivided in four macro categories: 1) Software, 2) Support, 3) Learning/Training, and 4) Staffing. This categorization is the base for our cost model.

Cost models usually capture a specific perspective of cost analysis. Ours focuses on surfacing intangible (hidden) costs.

In our model costs are first divided into migration costs – the volatile costs occurred in a migration – and ownership costs - the costs to own a software product. Both types of costs are further described by the four categories above.

Table 39 summarizes the total costs of migration for the organizations we have monitored. We have broken down costs into the subcategories and highlighted their intangible part. Subcategories labelled as intangible include costs that are hard to be budgeted and computed. For example, in the category Software the item "pilot projects" is considered intangible. Often a pilot project requires the use of spike solutions and new technologies to be tested on the fly. Their costs are often not foreseen. Another example is in the category Support. The costs for searching alternative solutions or documentation are always considered intangible [Shapiro, 1999]. The category Training/Learning contains most of intangible costs. For example, except the annual regular training, in a transition there are other types of training, like peer support or ad hoc training. Also the category Staffing has often a large percentage of hidden costs. For example, when an employee, whose tasks concern regular maintenance, is used to migrate to or introduce new IT, part of his budgeted cost may be considered as hidden for a transition: if no migration was ongoing s/he would have performed different tasks. Also incentives are often not budgeted.

Table 39: Migration Cost Model266

Cost category	Cost	Intangible?
Software	Pilot projects	Y
	Data conversion tools	
	Interfacing to legacy software	
	Software add-ons	
	Security tools	
	Upgrades	
Support	Search for alternatives	Y
	Search for documentation	Y
	Data compliance	
	Search for new support contracts	Y
	External support fees	
Training/Learning	IT personnel training for the new solution	
	IT personnel self-learning	Y
	Employees’ training for the new solution	
	Employees' self-learning	Y
	Lack of productivity	Y
Staffing	Employees extra hours and bonuses	
	Salary of temporary employees	
	Installation and deployment	Y
	Overheads and bonuses	Y

Costs of a migration are volatile – either intangible or not. They are directly due to the dynamic process of migration as for example, hiring temporary personnel allocated to the transition. They give information on the monetary effort that managers need to invest to migrate to an Open Source solution. In our model, these costs do not include the costs of ownership of a software product that are related to the acquisition and maintenance of software after it is deployed.

Ownership costs are useful for a long term cost comparison. Unfortunately, such a comparison is hard to perform for several reasons. For example, costs of ownership are monitored over a period of five years. In the specific case of the open solutions, this time-frame of ownership is often not yet reached. Another reason concerns the costs of licenses. These are one-time costs occurring in instants distant in time; a comparison based on them would be biased by factors like inflation or market demand. Thus it is meaningless comparing software on the acquisition costs.Following our framework, we have compared the cost of ownership of the open and the closed solution, collecting the data in the following categories of costs (Table 40).

[266]The bonuses in the two categories differ. Depending on the cause of the cost the bonuses might be tangible (as the first is one) or intangible (the second). See also the examples on the previous page.

Table 40: Ownership Cost Model

Cost of ownership	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost of purchasing	Annual Cost over 5 years	Initial Cost of purchasing	Annual Cost over 5 years
Acquisition (licenses)				
Updates				
Upgrades				
Software add-on				
Security (explore vulnerability)				
Maintenance (internal)				
Maintenance (support contracts)				
Consultancy				
Salary of employees				
Employees' regular training				
IT staff regular training				
Lack of productivity				
TOTAL				

Note. There are substantial differences betweens migration and ownership costs. Migration costs are often underestimated because of their intangible part whereas ownership costs do not have a predominant intangible nature[267], but they are difficult to obtain especially for new emergent technologies that do not have reached yet a large adoption. We were able to compute costs of ownership for OS solution only on the base of predictions based on the first initial years of ownership. Our results are, therefore, more detailed and significant for the migration costs - giving a substantial contribution to the effort analysis of a migration - than for the ownership costs – difficult to get in a short time frame.

This study describes, analyses, and discusses a rich variety of migration data, which has been collected through different means: subjective (questionnaires and interviews), and objective (data gathered by the PROM tool[268]).

Administering the questionnaires was not straightforward. To get the best quality for our data we visited each interviewee explaining the meaning of the questions - in particular giving examples of hidden costs they might have incurred during the migration.

For every organization, we summarize the results with tables and charts describing the break down of the costs into the categories and highlighting the hidden component of them.

12.1.2. Type of Migration

In this section we report of the migration type of the organizations considered. Other organization may use this categorization to find the case study that best fit their characteristics.

---

[267] Namely, we do not classify them in intangible/tangible.

[268] PROMetric. A tool to collect software metrics, developed at the Center for Applied Software Engineering of the Free University of Bolzano-Bozen. http://www.prom.case.unibz.it/

Table 41: Types of migration in the organizations analyzed

Organization	Type of migration
SGV (Consorzio dei Comuni della Provincia di Bolzano)	Partial migration from proprietary software
PP (Province of Pisa)	Partial migration from proprietary software
SK (Public Administration of City of Skopje)	Migration from scratch
TO (Törökbálint Nagyközség Polgármesteri Hivatala)	Migration from scratch
ProBZ (Province of Bolzano-Bozen)	Trial migration. Partial migration from proprietary software
Estremadura (Fundecyt in Estremadura)	Migration from scratch

No representative of a partial migration from mixed software or a total migration has been analysed.

12.1.3. Type of Organization

The majority of the organizations we have monitored are public bodies – the level number four of our classification based on NACE[269]. Fundecyt in Estremadura and SGV are private organizations that supply support to public bodies. Both have been established with this reason, but if SGV remained a supplier for the local councils, Fundecyt in Estremadura extended its action ray creating a parallel organization supplying support for Open Source distributions to private enterprises. SGV and Fundecyt in Estremadura can be classified at level 1 as NACE K.

12.1.4. Synoptic Overview of the Findings

In this section we summarize our findings and conclusions on the impact of the transition towards OSS on the costs in the organizations analyzed.

The model defined for each organization consists of a set of values of costs for the migration, for the initial purchasing, and for the ownership. The migration costs are labelled as tangible or intangible and they are subdivided in four macro categories. The ownership costs are computed on annual base and deduced by a monitoring or a prediction computation over a period of five years.

Table 42 displays the model of migration costs for each organization. The majority of the costs of migration concerns OpenOffice.org (OOo).

[269]See the deliverable "Report on the methodology used to collect data"

Table 42: Model of migration cost by category in each organization (KEuro)270

PA	Software (€K) Tang. | Intang.	Support (€K) Tang. | Intang.	Training/Learning (€K) Tang. | Intang.	Staffing (€K) Tang. | Intang.	Total (€K) Tang. | Intang.
SGV	€39.5K 82% | 18%	€82K 40% | 60%	€292.5K 92% | 8%	€246K 0% | 100%	€660K 51% | 49%
Extremadura	€0 -	€680K 26% | 74%	€180K 100% | 0%	€100K 100% | 0%	€960K 48% | 52%
PP	€99K 96% | 4%	€32.5K 77% | 23%	€61K 0% | 100%	€7K 0% | 100%	€199.5K 60% | 40%
SK	€0.01K 100% | 0%	€0.83K 28% | 72%	€3.07K 27% | 73%	€0.075K 0% | 100%	€3.985K 27% | 73%
TO	€20K 0% | 100%	€53K 62% | 38%	€233.5K 57% | 43%	€33K 0% | 100%	€339.5K 49% | 51%

Costs of ownership are compared in the pre and post software configuration to determine whether there have been savings (Table 43 and 44).

Table 43: Model of ownership cost comparison in the organizations

PA	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost of purchasing	Annual cost over 5 years	Initial Cost of purchasing	Annual cost over 5 years
SGV	€240K	€170K	€800K	€179K
Extremadura	€1.140K	€270K	€6.0K	
PP	€7.1K	€3.4K	€25.6K	€2K
SK	€0.7K	€2.4K	€23.1K	€2.4K
TO	---	---	€31K	€11.3K
BH (phase 1)	€68K	€45K	€735K	€169.6K

Comparing Table 42 and Table 43 we can deduce that:

- Costs of migration are significant and comparably higher than the annual costs of ownership – migration costs also occur in a shorter time frame. This means that the transition requires an exceptional monetary effort.
- Initial costs of purchasing are definitely higher for closed solutions
- Costs of maintenance are comparable in the two solutions although the OSS configuration is sometimes more expensive. This conclusion may be biased by the fact that costs for closed solutions are real whereas costs for open solutions are based on initial predictions. Initial prediction may still be influenced by the volatile costs of the transition.

[270] BH and ProBZ have not supplied this data

Table 44: Savings due to the migration[271]

PA	Savings of the OSS migration	
	Savings on initial costs of purchasing	Annual savings over 5 years
SGV	√	√
Extremadura	√	√
PP	√	---
SK	√	√
TO	√	√
BH	√	√

For the new open solution savings have been computed by predicting on the initial first year of ownership and historical data - as the five-year period of ownership for OS software has not been reached yet. All organizations report significant initial savings due to the zero cost of licenses. In the long term the profit is not that obvious: SGV, BH, Extremadura, TO predict to gain with the new solution. SK predicted equal costs for the new and the old solution whereas PP reports of higher costs with the new open solution in the long terms (see the section on the Province of Pisa).

Finally, we monitored the productivity of the employees using the two office suites (OOo and MSO). Productivity is measured with the amount and the speed of work of the employees in their daily activities. The table below gives a summary of our findings.

[271] ProBZ have not supplied this data. BH has migrated in two phases. Phase 1 is the most significant and we use it for our analysis. BH received a generous offer from Microsoft and in 2006 has gone back to Microsoft Office.

Table 45: Conclusions from the comparison of the usage of OOo and MSO in the organizations

PA	Conclusions
SGV	*In SGV the migration to OpenOffice.org has been extensive. The adoption has been not uniformly accepted, but an increasing significant number of employees fully use the open solution. No extra costs and decrease of speed of work has been found with the use of OpenOffice.org. Tasks have been performed regularly.*
ProBZ	*No big difference in the use of the office suites has been reported. The good results of the trial installation motivate the instantiation of a more extended experiment aimed at studying, analyzing, and evaluating the introduction of OpenOffice.org in public institutions.*
PP	*The use of OpenOffice.org in the Province of Pisa was extensive; the application was more tried than deeply used though. But, it was tried to perform usual office tasks. Comparing individual usage, the use of OpenOffice.org does not impact on the overall workload and effort of the daily office routine. No negative attitude toward OpenOffice.org has been detected.*
SK	*The pilot project for migrating to OpenOffice.org in the City of Skopje showed very stable behaviour in the employees' work. Moreover, the absence of a drop of OpenOffice.org usage towards the end of the period suggests that OpenOffice.org was quite capable in substituting Microsoft Office in the appointed tasks, whatever their complexity might have been.*
TO	*The analysis of the software usage in TO show that the general pattern of use is similar for the two applications and that the productivity is also comparable in the two cases. Since there were a significant number of switchers– users that utilized both products within the period that is analyzed, meaning that users are actually participating in the experiment, we can also conclude that the use of OOo could not have a negative impact on the work of the organization.*
BH	*Adoption of Open Source software started well before the experimentation. Employees have gained some experience with open formats. The expert employees of BH work similarly and produce more documents with OOo than with MSO. Therefore no extra cost but perhaps an intangible return on the investment is experienced in BH. We found that Beaumont Hospital has still to maintain proprietary format for the purpose of document exchange. As top management decision Beaumont Hospital is considering though to partially migrate back to proprietary software.*

In our analysis no adverse attitude toward the use of OSS was found. There is also no evidence that using OOo may cause additional costs to the organizations as the pattern of use of OOo is similar to the one of MSO, in term of daily documents worked and average time spent on them. In Table 46 we report the percentage of users of only MSO (pure MSO), users of only OpenOffice.org (pure OpenOffice.org), and the users of both the applications (Switchers). The maturity of the transition in each organization can be easily deduced from this table. A high percentage of switchers denotes curiosity toward the new technology. Switchers become Pure OOo users when the transition has been successful. High percentage of Pure MSO users indicates a negative attitude toward the new software or an initial stage of the transition – this is the case of SGV.

Table 46: Percentage of users. Pure OOo users, Pure MSO users, and switchers

PA	Users		
	Pure OpenOffice.org*	Switchers*	Pure MS Office*
BH	82%	9%	9%
SGV	5%	57%	38%
SK	2%	42%	56%
PP	7%	92%	1%
TO	19%	57%	24%

12.2. Province of Bolzano-Bozen

The case of the Province of Bolzano-Bozen represents a different study from the rest of the analysis. Namely, in this case the results refer to a pilot project that has been run as a controlled experiment to understand the productivity of the employees. No data about ownership costs or budgeted migration costs are reported here.

The Consortium of the Townships of the Province of Bolzano-Bozen (Italy), in collaboration with the Centre for Applied Software Engineering of the Free University of Bolzano-Bozen, has performed a trial installation of OpenOffice.org in ten associate townships. Townships ranged from very small (five employees) to small-medium size (twenty employees). The activities performed are the usual office tasks: word processing, spreadsheet, *etc*. Microsoft Office was the only office automation tool used.

In the end, OpenOffice.org was installed on about one hundred desktop computers. The operating system was Microsoft Windows in all the cases.

A set of 16 PC computers uniformly distributed in the Townships was selected. The end-users volunteered for the experiment belong to four different departments. In eight PC computers was installed OpenOffice.org.

Transitions lasted from two to four working days and employed two instructors each. A personnel training was performed on-site and one-to-one. Instructors first went to the site for "exploring" the environment and for collecting the most used documents by offices' personnel. The instructors then returned the day after with all the documents converted to OOo format. They then installed OpenOffice.org and train the personnel by working on the very same documents they were usually working on.

The conversion of more than two hundred documents from Microsoft Word to OpenOffice.org was performed without any particular problem and with great efficiency: the size of an OpenOffice.org document was generally one third of the equivalent Word document.

Personnel do not generally look positively at the introduction of new or different technologies and at the abandon of those which is used to: a phenomenon called "hostility to change". The most reported reason is the refuse to use tools different from those of colleagues or from those used at home. However, during the transition to OpenOffice.org we found only a few employees showing hostility to change.

We instead have found an inefficient use of resources: the personnel routinely used only the very basic features of Office, and did not consider little more complicate features which would have lead to better use of resources.

Users with good knowledge of Office have not had any problem in switching to OpenOffice.org. Most of the problems have been caused by personnel with little Office knowledge. Personnel's training has been usually performed on-site and one-to-one, but it has turned out that instructors have had to frequently interrupt training because of incoming phone calls, urgent documents delivering, *etc*.

The pilot project was designed in few steps. First a picture of all the applications calling and called by the office automation tool used, and the macros used by each department was taken. Then it was monitored the use of the office automation tools for a period of seven weeks before the transition. Soon after the transition it was monitored the use of both the old and new solution for a period of fourteen weeks. Finally, the access to the documents to automatically opening them in OpenOffice.org was configured: opening the documents with Microsoft Office was still possible, but it required a more complex procedure. Again it was monitored the use of Microsoft and OpenOffice.org for a period of other three weeks.

12.2.1. Data analysis

Table 47 describes the calls between Word/Excel and the other Microsoft Office applications. We have selected the following applications

Table 47: Description of the applications considered

Called by	Description
CPCQM.EXE	Printer driver Canon
MSOHELP.EXE	MSO Help Menu
EXPLORER.EXE	Folder viewer
DW.EXE	MSO Error Reporting tool
EXCEL.EXE	MSO Excel
IEXPLORER.EXE	MSO web browser
MSACCESS.EXE	MSO Access
MSTORE.EXE	Microsoft Clip Organizer
IFRUN60.EXE	Oracle Forms (Runforms)
OUTLOOK.EXE	MSO Mail client
WINHELP32.EXE	MSO Help guide

Table 48. Top score applications calling Word and Excel

Calling	Word	Excel
EXPLORER.EXE	80.74%	94.38%
OUTLOOK.EXE	14.61%	3.42%
DW.EXE	0.64%	0.92%
IFRUN60.EXE	0.11%	0.00%
IEXPLORER.EXE	0.19%	0.00%
EXCEL.EXE	0.04%	0.00%
UNKNOWN	3.68%	1.28%

Table 49. Top scores applications called by Word and Excel

Called by	Word/Excel
CPCQM.EXE	71.05%
MSOHELP.EXE	13.16%
EXPLORER.EXE	7.89%
DW.EXE	5.26%
EXCEL.EXE	2.63%
IEXPLORER.EXE	2.63%
MSACCESS.EXE	0.00%
MSTORE.EXE	0.00%
IFRUN60.EXE	0.00%
OUTLOOK.EXE	0.00%
WINHELP32.EXE	0.00%

The numbers expressed in the tables indicate the interoperability of the desktop applications that needs to be taken into account in the transition: customization and adaptation of the office tools impact on effort and costs.

Namely, despite the difference in percentage of calls (Table 49), all the applications that are in Table 49 need to be considered in the transition. For example we need to customize the call to an oracle DB as there has been at lest one call to this DB (IFRUN60.EXE). For the same reason we analyze the existence of macros: for the accessible excel files it has been reported 43 macros for a total of 21,482 lines of code for 526 files inspected. No macros have been found in place for Word files.

We monitor the number of documents used and the daily time spent on the documents as a measure of productivity.

The histograms below report of the three productivities in the three periods in the two groups.

Each bar in the below histograms represents the productivity of a single employee in the three different phase of the monitoring. The pre-transition phase corresponds to the nr. 1, the transition phase to the nr. 2 and the post-transition to the nr. 3

Figure 62: Partial productivity in the three periods in the group transited to OpenOffice.org

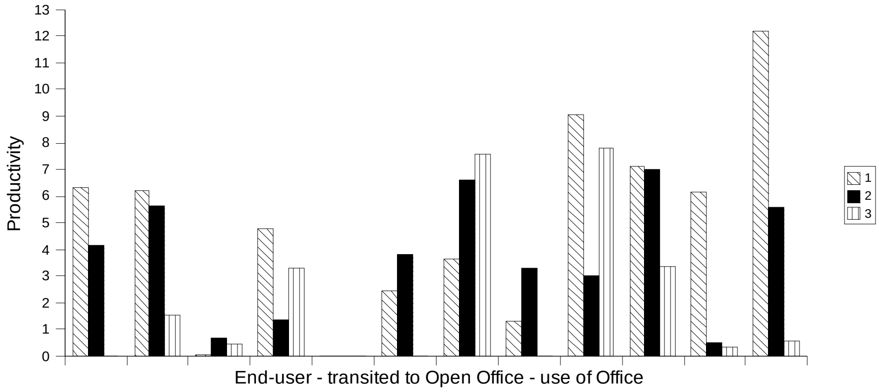


Figure 63: Partial productivity in the three periods in the group not using OpenOffice.org

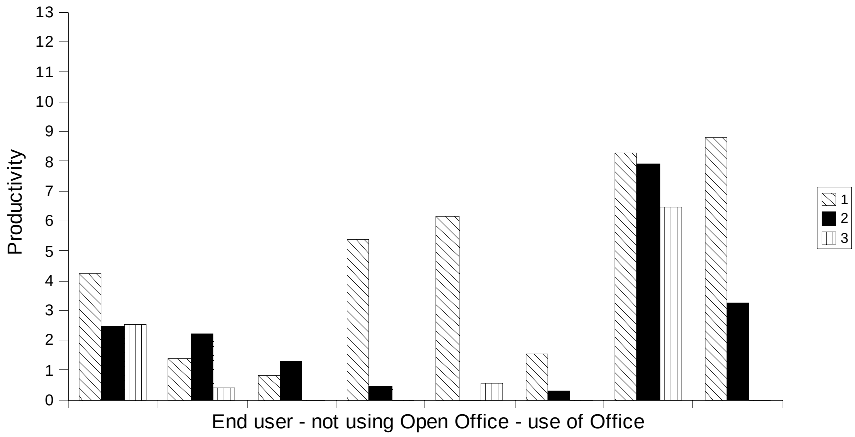
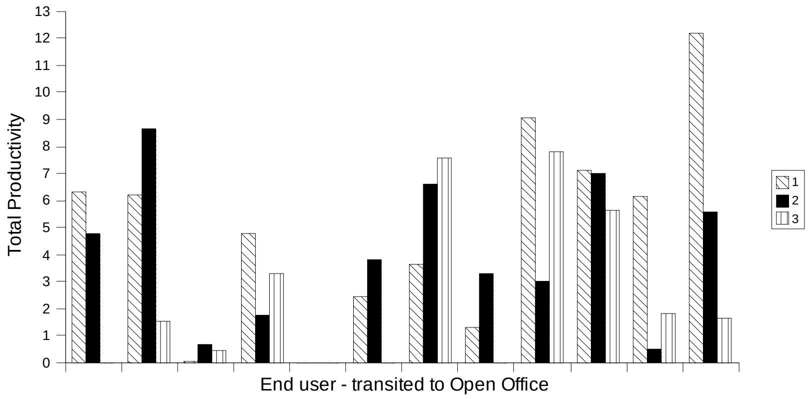


Figure 64: Total productivity in the three periods in the group transited to OpenOffice.org



The first picture represents the daily average productivity when documents are saved/modified as office documents (.doc, .xlm extension) disregarding the application used - within OpenOffice.org or Office. Therefore the productivity here is an upper bound of the productivity related to the solo use of Office: a user may have modified a file .doc with OpenOffice.org.

The second picture displays the trends in the control group. From the picture we deduce that the first period of monitoring is characterized by a bigger productivity.

The third picture reports the productivity compute as ratio of number of files – indifferently Office or OpenOffice.org – and time spent with OpenOffice.org or Office applications.

The preliminary analysis based on the groups comparison in the three periods of the experiment, indicates that there is no lack of productivity in the group transited to OpenOffice.org.

Even more some of the members of the group transited to OpenOffice.org present a higher productivity when working both with only Office documents (Fig. 1 and Fig. 3 period n. 1) and with any kind of document (Fig. 3, period n. 3). In the transition, when the choice to use one or the other application is even (Fig. 1 and Fig. 3, period n.2), no increase of productivity has been registered: no documents new or saved as OpenOffice.org files have been produced.

In the third period the use of OpenOffice.org has increased as the path to access to Office applications has become more complex.

To facilitate the transition we have performed an analysis on application interoperability and existence of macros. This has helped to customize the new solution in terms of the needs of the end-user.

12.2.2. Analysis of the Problems

Again, the fact that Microsoft Office is by far the most used office automation tool raises the problem of training the personnel for OpenOffice.org. To this end, part-time courses on OpenOffice.org have been organized. The courses are held off-site, to avoid the disturbing factors experienced in the trial installation. In these courses, offices' personnel have been taught the basic and most used OpenOffice.org features, with the possibility of suggesting some particular topic of interest.

Another problem which might occur is the hostility to change. In this case, in order to maintain the efficacy of the training action, we might think of motivating the personnel by a series of “bonuses for change”. Another solution is to train homogeneous groups of people, that is, personnel coming from the same of closely related offices.

The choice of introducing OpenOffice.org while maintaining the same client operating systems is motivated by the need to minimize the training load for the personnel. That choice allows also a smooth transition, minimum interruption of public services and limits any possible hostility to change.

It has also established: a hotline, a data base of success cases, a FAQ and a knowledge base. These services are aimed at organization personnel already trained and will offer user and technical support on various OS software of interest for the organization.Consorzio dei Comuni della Provincia di Bolzano - Südtiroler Gemeindenverband (SGV), Italy

SGV is an institutional but private Italian association providing support for IT to the councils of the province of Bolzano-Bozen. The organization consists of 50 employees. In our classification it represents a high IT organization. It supplies software support for local councils of various needs and levels of IT. SGV has seen OS software a possibility to free councils from continuous upgrading. SGV hired a person to follow the migration in all the councils of the province. SGV uses a software to nightly convert documents in the open format when it feasible. The strategy SGV adopts for their clients to convince them to migrate was the regularization of the licenses: no need of frequent upgrades and the freedom to distribute software to all the employees of the council. Despite this, not all the councils have adhered easily and some of them are still reluctant. The migration has started in a medium size council where on 10 PCs no licenses of the installed software were found. The migration was conducted with no real training as the majority of the employees were part-time. The lack of training and immediate support causes the failure of the migration. A well defined strategy was then put in place: a period of training and the uninstalling proprietary software to then replace it with the open one. Specifically, the migration was extensive for OpenOffice.org (OOo). SGV upgrades its office suite internally and for its clients till Microsoft office 97 in 1998 with the internal upgrade of 30 licenses. At the time of this report, the situation at the first council which migrated is different: 23 % of the documents created in July were with OOo. Smaller councils with more dependency from SGV have reported even better numbers in favour of OOo: the majority of the documents created in July was in OOo. The migration of office suites are an extended but easy task to perform. Few technical impediments have occurred. They were mostly connected to an adverse attitude. In SGV, the programs connecting to databases - as for example the software for accountancy – still remain a big issue. In this case, data are stored in an Oracle database and the migration of the database and its interfaces has resulted unfeasible at the current state.

12.2.2.1. General Overview of the Migration

The migration started in 1998 with a server side transition to the Linux Operating System. After a positive experience with Linux a wide migration of the existing office automation tools to Open Source alternative has started in 2003, namely OpenOffice.org was installed. In 2004 a general deployment was achieved. In fact, after 2 years of usage about ¼ of the documents are already in OpenOffice.org format.

Table 22 shows a summary of the migration costs and effort in SGV. Each category is afterwards discussed in more details.

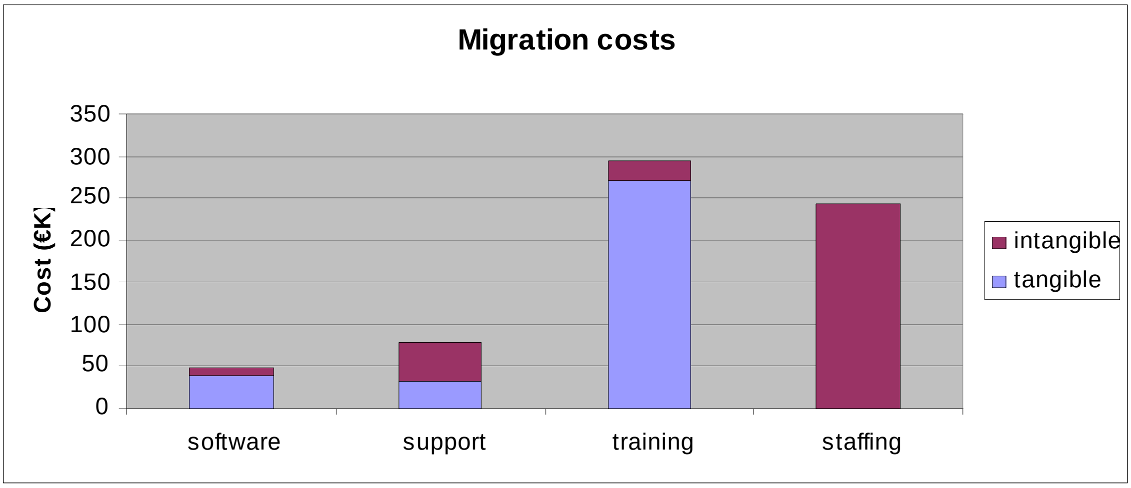
Table 50: Summary of the Migration Effort and Costs for SGV

Category	Intangible?	Effort (man/months)	Cost (€K)	Subtotal (€K)
Software	36.5			
Interfacing to legacy software		3.5	10.5	
Upgrades		9	26	
Support	49.5			
Search for alternatives	Y	10	29	
Search for documentation	Y	6	18	
External support fees		-	2.5	
Training/Learning	24.5			
IT personnel training for the new solution		5.5	16.5	
IT personnel self-learning	Y	0.75	2	
Employees' self-learning	Y	2	6	
Staffing	245			

12.2.2.2. General Overview of Hidden Costs

The figure below shows that most of the hidden costs occur in Support and Staffing, this is due to internal strategies for the migration in which the management has decided to allocate only internal personnel to manage the migration process and a considerable amount of time has been spent in searching documentation, suitable Open Source products, and external support.

Figure 65: Tangible and Intangible Costs in SGV



On the other hand, almost all the training costs are explicit. This happened because the management has recognized the high importance of training as a critical factor for a successful migration and has included this cost into the annual cost balance.

12.2.2.3. Analysis of the Cost by Category

Before a general migration, pilot projects were set-up for each of the three software categories. For the Operating System and the Groupware solution, the effort for the pilot project was about 0.5 person-months and € 1.5 K each. For OpenOffice.org effort was of 2 person-months and € 6K. There was no cost for data conversion and security tools. However both time and money were spent for interfacing legacy software as shown below. In the case of Group-E there was a need to configure Pegasus (an e-mail client) to access to Group-E via IMAP. For migrating to OpenOffice.org a conversion of existing applications based on Microsoft Office was required, especially the ones for the technical office and the decision management.

Only for the groupware (Group-E) additional costs appeared for add-ons – specifically €3K and 1 person-month of effort. While this was one-time expense regular costs are expected in order to cover upgrades. Generally, such costs will appear once every two years or two and a half years. It is calculated for 238 servers on which Linux will run and about 3,000 PCs that are needed for the SGV employees on which OpenOffice.org will be available.

Certainly, other costs appeared caused by the introduction of open source software. For books and other training materials €1K was spent for Linux and OpenOffice.org (€ 0.5K each).

The migration process started with the search for alternative Open Source solutions to the used proprietary software and comparison of the available options. This initial phase took in total 10 person-months and € 30K. To be more precise, for choosing the operating system for the server 3 person-month and € 3K were spent; for office automation 6 person-month and € 18K and for selecting the groupware 1 person-month and € 3K.

Searching for documentation for the chosen Open Source solution is sometimes quite time-consuming and, thus, implicitly influences migration costs. The estimated costs are as follows: the biggest share falls to searching documentation for OpenOffice.org - € 9K, followed by the information related to the Linux OS – € 6K and, lastly, € 3K for the Group-E.

Additionally, for achieving technical and data compliance and interoperability, expenses of € 29K were added to the migration costs. They were spent as 10 person-months for the conversion of the common documents (i.e. OpenOffice.org).

For some of the chosen Open Source packages external support was necessary, namely for the Linux and Group-E the reported cost is € 2.5K. Furthermore a 0.5 person-months and € 1.5K were spent for searching for new support contacts considering Linux OS and the final choice was a one year support contract for SUSE. For the groupware the support is provided by the Group-E developers of a local software house.

Summing up, support expenses were mainly for searching for alternative solutions to proprietary software used before and finding appropriate documentation. Relatively big seem to be also the costs for achieving technical and data compliance and interoperability of the OSS, as shown on Figure 4.

At the beginning of the monitoring, only a small part of the IT personnel (10 people) was prepared to the transition, i.e. properly trained, and to work with OSS. Partial training of the IT staff was provided before the transition. Additional three weeks training was given to 6 members of the IT staff during the transition. The cost of this external training is estimated in

€ 2K. The cost of the self training is much bigger as it involves 6 people that dedicate most of their time on the Linux servers. In total, training amounts in 5 person-months effort at the total cost of € 15K. Self learning was required for OpenOffice.org and Group-E and was estimated to be € 6K and € 1.5K respectively. For the groupware solution also an external training was required and was provided by a local software house for the cost of € 4K.

The training of the rest of the personnel of the municipalities to the new functionalities of the Open Source software was carried out using a combination of approaches. One-day training was provided to the personnel in the IT centre. This added up some costs for travelling, which, unfortunately, were impossible to trace. On-site help was provided whenever needed during the regular visits of the IT staff at the councils. At the beginning of the migration, a helpdesk was available for 1 hour a day and administrators were dedicating 2 hours a day for remote support via VNC. Products' documentation was made available on the Intranet. Summing up to about 2500 employees were provided training throughout a 2 month period. An IT expert has been hired to support the transition to OpenOffice.org and external consultancy was needed for some server side solutions (e.g., Linux). All this makes the biggest expense during the migration, which is €256K.

At the same time the lack of productivity of the employees, for both IT staff and all the other users, should be considered in the staffing expenses and is a crucial hidden cost.

As reported previously, the IT administrators were spending 2-3 hours a day during the first two months of the migration period for supporting users via help desk and remote VNC support. This time is calculated as lack of productivity for the employee that needed the help. The cost can be estimated to € 2K.

One of the biggest expenses during the migration period was for staffing. The personnel involved in the deployment of the software were estimated to cost about € 245K. For the client side deployment the installation was done by the administrators (IT staff) via remote installation scripts. The cost was that high because the number of client machines, which OpenOffice.org was installed on, was rather big (2,829 PCs).

During the migration one person was hired to follow the entire migration at the councils, clients of SGV. No temporary employees were hired for the internal migration. There was also no additional costs for regular employees' extra hours or bonuses caused by the migration.

12.2.2.4. Summary of the Costs by Software Category

The total cost for software and support has been estimated in € 124K and includes about 43 person-months for the deployment of three software groups we have monitored (i.e. server-side operating system, office automation, and groupware), not counting the training and staffing effort. The largest share of costs refers to office automation, followed by operating system for the servers.

Cost of Ownership

SGV started the transition with a heterogeneous software setting, which consisted of both Open and Closed Software solutions. However, almost all the Open Source products installed were server-side software packages. The migration consisted, mostly, in a massive installation of OpenOffice.org in all the municipalities of South Tyrol.

12.2.2.5. Previous Solution

Table 51 shows that all the initial costs refer to licenses and installations. In the long term, the majority of total annual costs are related to maintenance, updates, and upgrades of software. Training has only a limited impact on the overall expenditures and costs are equally distributed between IT staff and employees.

In general, such setting would require three people to manage Microsoft Windows 2003 server (€ 105K) and 1 person to manage Microsoft Exchange server (€35K). These activities include maintenance, updates, training, and internal consultancy. An extra €35K needs to be added for the upgrade to the Microsoft Vista software.

Table 51: Cost of Ownership of a Closed-Source Solution in SGV

Cost of ownership	Closed Source Solution					
	Initial Cost of Purchasing (€K)			Total annual costs (€K)		
	MSO Win2K3 Server	MSO Office	MSO Exchange Server[272]MSO Win2K3 Server	MSO Office	MSO Exchange Server	
Acquisition	605	136	112	-	-	-
Maintenance, updates, upgrades	-	-	-	289	25	5
Employees' regular training	-	-	-	-	10.5	-

12.2.2.6. Current Solution

Table 52 shows that most of the initial costs of purchasing are due to the installation of Open Source packages. In addition, other costs are added to buy ARKEIA, a backup software for Linux € 20K, subscribe a support contract for SUSE Linux (€ 0.5K), and train employees to use OOo.

Most of total annual costs are due to maintenance, upgrades, and updates. In addition, minor costs are necessary for a support contract (€ 2K) with a local software house and IT and personnel's regular training (€ 10K).

Such setting requires three people to manage Linux servers (€ 105K), one person to manage maintenance, upgrades, and training of OpenOffice.org (€ 35K), and, finally, one person that spend 20% of this time to manage Group-E (€ 7K)

[272] This cost has been computed also on the imminent purchasing of Microsoft Vista. 35KEuro have been allocated for maintenance of such Operating System.

Table 52: Cost of Ownership of an Open Source Solution in SGV

Cost of ownership	Open Source Solution					
	Initial Cost of Purchasing (€K)			Total annual cost (€K)		
	Linux	OpenOffice.org	Group-E	Linux	OpenOffice.org	Group-E
Acquisition	170	55	20	-	-	-
Software add-on	20	-	-	-	-	-
Maintenance, updates, upgrades	-	-	-	268	14.5	3
Maintenance support contracts	0.5	-	-		-	2
Employees' regular training	10.5	-	-	-	10.5	-

Table 53 identifies the OSS components adopted by SGV, and clearly shows that the actual cost savings in the transition to Open Source software were extremely significant. The once-off savings compared to proprietary alternatives are in the order of €600 thousands. Furthermore, annual maintenance costs, viewed over a five year period, show that savings are even more dramatic, leading to an overall saving of € 1.2 millions.

Table 53: Cost Comparison of OSS versus Comparable Closed Solutions for SGV

Application	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost of Purchasing (€K)	Total cost over 5 years (€K)	Initial Cost of Purchasing (€K)	Total cost over 5 years (€K)
Operating system (server-side)	170K (Linux)	991K	640K (MSO Windows 2003 Server and Vista)	1.97M
Desktop systems	55K (OpenOffice.org)	72.5K	136K (MSO)	125K
Groupware	20K (Group-E)	14K	77K (Microsoft Exchange)	26K

12.2.2.7. Cost of Use of OpenOffice.org

"In SGV the migration to OpenOffice.org has been massive. The adoption has been not uniformly accepted, but an increasing significant number of employees fully use the open solution. No extra costs and decrease of speed of work has been found with the use of OpenOffice.org. Tasks have been performed regularly."

In the Consorzio dei Comuni della Provincia di Bolzano we have analyzed a sample of data of a period of about two months. The period of time we have chosen is representative of the whole period of the experimentation (lasted about one year).

The number of users monitored has been very high (1525 PCs). One third of the users never used OpenOffice.org. The existence of such a big number of users that did not take part of the experimentation, but have been monitored might mean that there was a top management decision (the central IT department) for the migration. We might conclude that there was a partial resistance of the personnel. This might be related to the structure of SGV as association of several councils, some with little infrastructure and dependent from the central IT department and others with a very modern IT infrastructure and therefore independent from the central IT decision quarter.

83 users (5.45%) used only OpenOffice.org. Although this number is in percentage small, its absolute value is noticeable. As the central IT department is small, 83 people working only with OOo, means that several non-expert employees fully adopted this application for their daily routine.

Comparing usage in the two groups - excluding users of both the applications - we can see a great similarity. Table 54 displays the similitude in the use (in average) of the two applications.

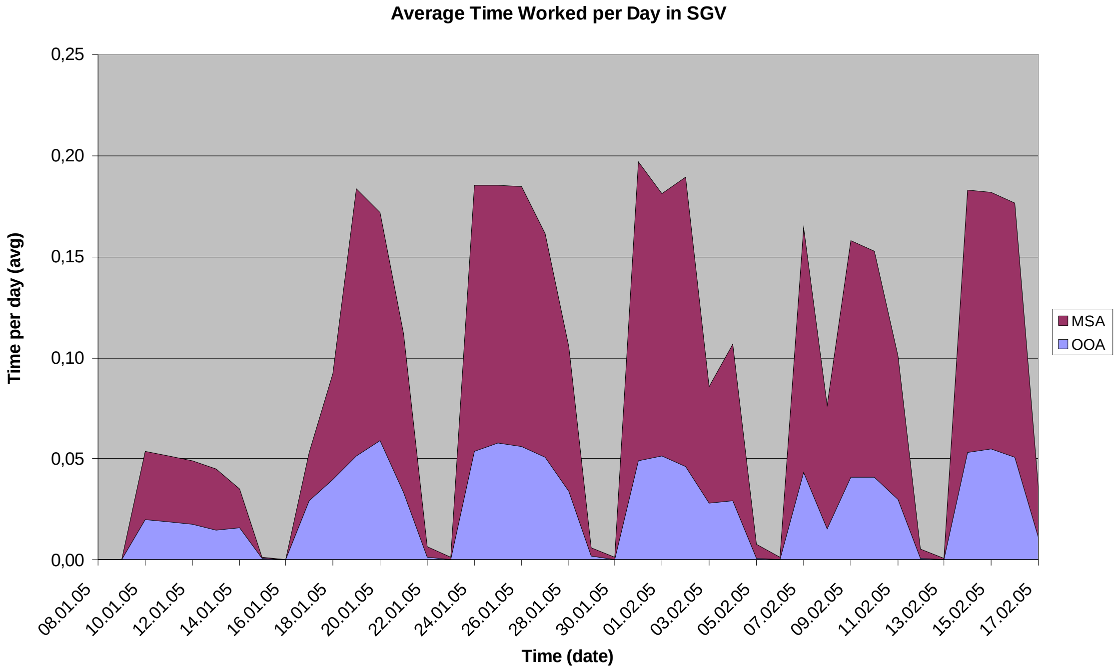
Table 54: Types and Number of Users for SGV

	Open Source	Microsoft
Average number of events per document	18.49	12.92
Average time spent on a document (seconds)	955.88	800.03
Average number of users working with a document	1.35	1.26

12.2.2.8. Effort and Productivity with the Two Applications

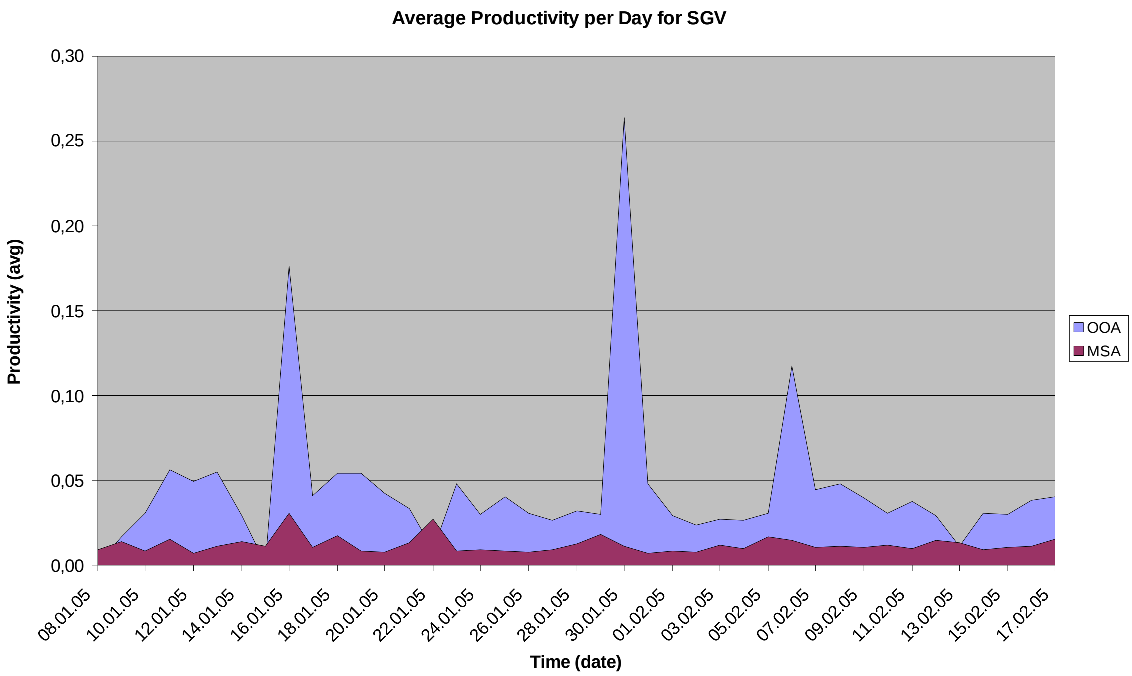
Now we perform our analysis excluding the pure MSO users. Almost two thirds used both the applications (about 900 people). For these users we found that the daily number of Microsoft Office documents per user is the biggest. The average time spent on documents per user by day is roughly 3-4 times higher for Microsoft Office than for OpenOffice.org. Only a small part of Microsoft Office documents were opened in OpenOffice.org. Instead, the common format for documents opened with OpenOffice.org was the native OpenOffice.org document format.

Table 55: Average Time Spent on Documents per User by Day for SGV



As we have already said, productivity is a measure of the "speed of working" (the number of documents produced divided by the time spent working). Daily productivity is higher when using OpenOffice.org documents proving at the first sight that OOo users work faster than MSO ones. In Figure 10, the productivity of OOo is somewhat twice as high as the productivity of MSO

Table 56: Average Productivity by Day for SGV



We can conclude that people are working faster with OOo than they are working with MSO.

The absence of any decrease of the usage of OpenOffice.org suggests that OpenOffice.org was quite capable in substituting Microsoft Office in the daily usual office tasks of the employees.

12.3. Fundecyt in Extremadura, Spain

The Spanish Region of Extremadura is the first case in the world of adoption of Open Source software in high schools and public offices.

12.3.1. General Overview of the Migration

Extremadura is the poorest region of Spain, lagging behind the rest of the country in both the economic and technological area. In the mid-90s, the regional government decided to invest in information technology as a means that could help the Region to overcome its historical peripheral situation. The strategy of the government was twofold: provide Internet as a public service to citizens and train people to use new technologies.

The LinEx (Linux Extremadura) project is a Linux distribution created to provide universal access of the regional IT services to all the citizens. The main goal of LinEx is not software innovation, but rather the specific aspects of translation and customization. To avoid any kind of technical problem during the initial phase of the project, a Spanish company (Andago, Madrid) was hired to take an existing set of Linux software from the web and customize it.

LinEx is specifically designed for use in regional administration and schools, but the software is distributed for free on a much larger scale than public bodies.

Table 57 presents an overview of costs needed to introduce LinEx in the Public Administration of the region of Extremadura.

Table 57: Summary of Migration Effort and Costs for Extremadura

Cost	LinEx		
	Effort	Cost (€K)	Subtotal (€K)
Support	680		
Search for alternatives		400	
Search for documentation		100	
External support fees		180	
Training/learning	180		
IT personnel training for the new solution		90	
Employees' training for the new solution		90	
Staffing	100		
Employees extra hours and bonuses			

12.3.2. Analysis of the Costs by Category

The adoption of the Linux distribution did not require expenses for data conversion, interfacing with legacy software. All the necessary upgrades have been provided by the local software house Andago and they are included in the costs of external fee support (€ 180K). The only additional costs caused by the introduction of Linux have been:

Hardware for € 150K

Graphical Design for € 30K

The first phase of the project was related to the evaluation of different alternatives for the adoption of Open Source software in schools. To cope with this problem, expert staff has been hired: five people for the first two years and, then, three people for the next two years. Each member of the team costs € 25K gross per year. Thus, the total cost of this operation was € 400K. After this initial phase, all the remaining work has been provided by the external company Andago at the fee of €180K. They provided:

- installation and deployment;
- support during the adoption.

A pilot project started before the official adoption of Linux. 14 schools were involved and they continued their regular activity (study and lessons) during the transition.

For the training of personnel (both IT and administrative), the expenses were € 45K per year for four years. The training was organized as explained above (courses, seminars, etc.). There has been no lack of productivity in the schools. For the adoption of Linux, two temporary staff members have been hired for 1 year with a total cost of € 100K. In this way no extra working hours have been necessary for the permanent staff (extra performed time would have been repaid as extra spare time).

12.3.3. Cost of ownership

Costs of ownership for Extremadura were not completely available. Table 58 reports mainly of initial costs of purchasing.

Table 58: Cost of Ownership of Extremadura

Cost of ownership	Open Source Software Solution	
	Initial Cost (€K)	Total Cost over 5 years (€K)
Acquisition, Updates and Upgrades	860	
Maintenance support contracts	Included in acquisition, updates, and upgrades	
Consultancy	Included in acquisition, updates, and upgrades	
Salary of employees	100	Average 42

12.4. Province of Pisa (PP), Italy

12.4.1. General Overview of the Migration

Since 2003, the Province has adopted a local law (L.P. N°. 186 of 21/10/2003) that recommends the use of ODS and OSS. Before the beginning of the transition the province has analyzed a possible OSS migration with two pilot projects:

"Gare di appalto" software. In 2003, the Province decided to re-engineer a piece of legacy software in response to the new law. The strategy applied was to replace the old proprietary version and use in-house skills and resources to develop a new version. At the same time, the Province has modified the contract to have the ownership of the code of the new in-house version and the right to freely distribute it to other organizations. The resulting software is released under GPL.

The GIS area. In this case, the Province has adopted OSS only for the whole area and at the same time signing a yearly contract with five external consultants, due to lack of availability of in-house IT staff. The Province has saved money on licenses as well as training and wages for in-house IT staff, even though the external consultancy incurs additional costs. Two internal employees act as work coordinators and the cost of each external consultant is more or less the same as that of the in-house technical staff.

From this experience the Province has adopted the following strategy:

- Before transitioning to a new solution, it analyses the availability of OSS. OSS is both a challenge and an opportunity to reduce costs;
- It analyses the transition towards OSS in all the software products developed in-house. Often these products need re-engineering, since they began without a clear overall design.
- It extensively adopts the Open Document Format. Where possible, the Province requires its software suppliers to produce solutions that work with the Open Document Format.

When the migration began, the province had already a good experience of OSS and ODS. Within the migration, PP has migrated 120 workstations to OpenOffice.org and the Mozilla suite. Red Hat was installed on 11 servers. To evaluate OpenOffice.org a group of employees volunteered to evaluate the suite. To facilitate the use of OpenOffice.org a web application "Doc transformer" was developed internally to convert MSO into OpenOffice.org documents.

Table 59: Summary of Migration Effort and Costs for PP

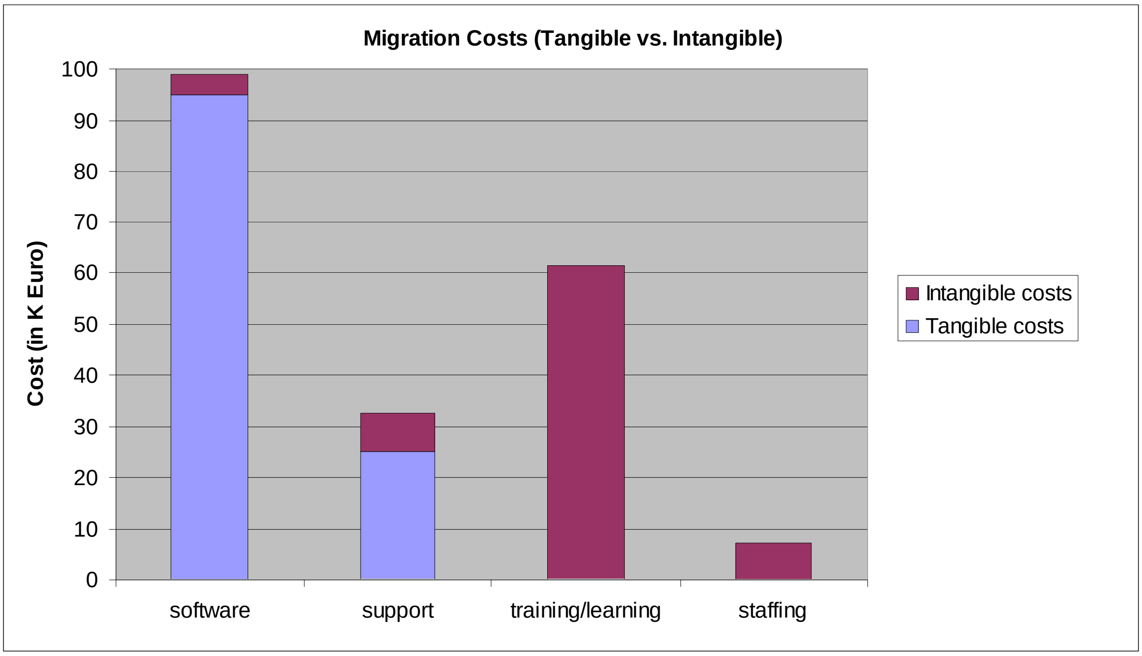
Category	Intangible?	Effort (man/months)	Cost (€K)	Subtotal (€K)
Software	99			
Pilot projects	Y	-	4	
Data conversion tools		-	25	
Software add-ons		-	5	
Security tools		-	45	
Upgrades		-	20	
Support	32.5			
Search for alternatives	Y	-	7.5	
External support fees		-	25	
Training	61			
IT personnel self-learning	Y	5	11	
Employees' self-learning	Y	22	50	
Lack of productivity	Y		-	
Staffing	7			
Installation and deployment	Y	3	7	

12.4.2. General Overview of Hidden Costs

Figure 14 shows the comparison of intangible and tangible costs in the four categories of costs. The histogram displays a high value for software costs, with little percentage of intangible costs. The percentage of hidden costs is due to the pilot project run for the introduction of OpenOffice.org. The high value of costs for software is related to several tools implemented as add-ons, plug-ins, and security tools.

The little cost for staffing (all hidden) is unexpected at the first instance. Namely, PP implemented lots of in house software, but the effort declared is not so high. This apparent contradiction may be due to the previous degree of knowledge and skills of the IT staff. This conclusion is also upheld by the high percentage of intangible costs in “support”: “search for alternatives and documentation” is the real hidden expenditure together with “ad hoc and peer to peer training” – as it shows the bar of training/learning costs.

Table 60: Tangible and Intangible Costs for PP



12.4.3. Analysis of the Costs by Category

The transition in PP has focused on three types of software migration. Table 61 presents the break down of costs and effort per type of software. The category support is omitted as it was not possible to retrieve this information.

Table 61: Summary of Migration Effort and Costs for PP by Software and Category

Cost category	Software type	Costs (€K)	Effort (hours)
Software	Operating System desktop (Linux)	4	-
	Office Automation (conversion tools and integration tools, Thunderbird)	30	-
	Mozilla	-	-
Training/ Learning	Operating System desktop (Linux)		800
	Office Automation (OpenOffice.org)	50	3600
	Mozilla	0.3	25
Staffing	Operating System desktop (Linux)	1.5	152
	Office Automation (OpenOffice.org)	2.5	240

The migration has involved three major migrations: to the mail application Thunderbird, from Microsoft Office to OpenOffice.org and from MSAccess to MySQL. Costs for software is mostly due to in house implementation of conversion tools (Microsoft Office/ OpenOffice.org, MSO, Access/MySQL), plug-in for OS tools for file conversion (Thunderbird), spike solutions, and upgrades due to proprietary tools still in use in the PP. No hardware costs due to the migration have been reported. No costs for legacy systems are reported. No costs for security tools and upgrades have occurred for the migration.

The major cost for support concerns two temporary employees that have provided consultancy on the migration. The total cost in this case is € 25K for two consultants temporarily hired for 6 months. Two members of the IT staff and the consultants have also worked in searching for alternatives. The internal staff sums up to € 2.5 K in this case. There was no cost for searching for new support contracts as external consultants enrolled in this project have already had several contracts with PP.

Training for non-IT personnel is provided in three different ways:

- For generic requests, technicians try to solve users' problems. If the problem is due to inadequate use of software, the technician explains the problem using examples in a one-to-one training session. The cost of this training is estimated at one day per month;
- Internal manuals and short references guide are written by IT staff. For example, to manage the introduction of digital signatures in some business processes, the IT staff wrote 10 different references guides to help employees to use smart cards with certificates;
- The training for software developed by external suppliers is provided by an external instructor. The maximum cost allowed is between € 500-800 a day (plus national sales tax of 20%).

PP was able to quantify internal self-training with a total of 62K Euro. The majority of the costs are due to OpenOffice.org self-training estimated with 30 hours of effort per person. No lack of productivity due to the introduction of the three types of software has been detected.

Cost for staffing is low and limited in time. This is because PP has a skilled internal IT staff that supported the migrations with little effort. No costs for bonus or extra office hours have been reported. Costs are mainly for deployment and installation. OpenOffice.org has been self-installed by users in about a total of 30 days. The total time for this installation has been estimated (considering the average salary of the employees) to be about € 2.6K. Thunderbird has been installed in about 450 PCs by the IT staff, requiring 30 minutes per installation with a total cost of € 3K. Linux desktop has been installed in 5 PCs in two days by the IT staff in collaboration with the external consultants. The cost for internal staff corresponds to € 1.5K.

PP has no cost to report for the transition to Linux server. This operation was supported by software vendors, as part of the contract for maintenance, which was dated before the transition. This cost concerns the periodical update of the servers of PP.

12.4.4. Cost of Use of OpenOffice.org

“The use of OpenOffice.org in the Province of Pisa was extensive; the application was more tried than deeply used though. But, it was tried to perform usual office tasks. Comparing individual usage, the use of OpenOffice.org does not impact on the overall workload and effort of the daily office routine. No negative attitude toward OpenOffice.org has been detected.”

In the Province of Pisa the collection of data with PROM lasted for a rather long period of 8 months.

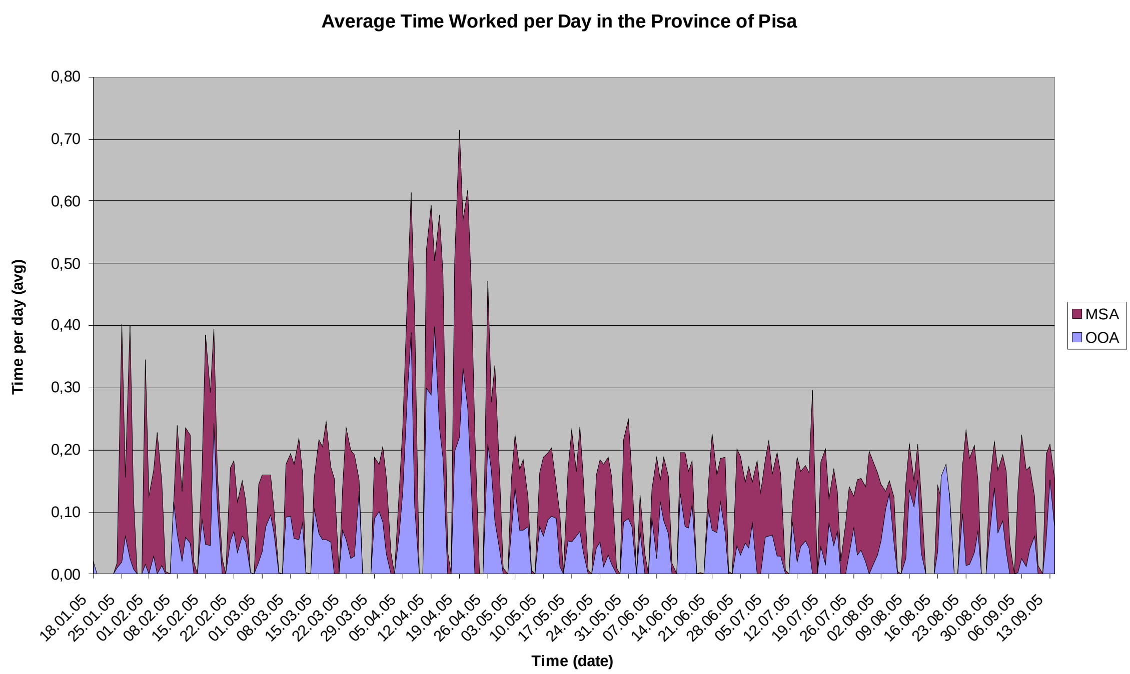
During this period there was just one user which used only MSO. There were only 8 users (possibly comprising the 6 IT members) that used only OOo during the whole period. The rest (more than 100 people) was switching from one application to the other. No adverse attitude toward OOo has been recorded.

By the heterogeneity of the group of employees and the similar pattern of documents' workload (omitted) we can also say that the users perform similar tasks both with OOo and MSO.

12.4.4.1. Effort and Productivity with the Two Applications

"The average time (effort) worked per day is always higher for MSO than for OOo, which was to be expected as MSO was always more used than OOo. The proportion of the average time worked per day is more than double."

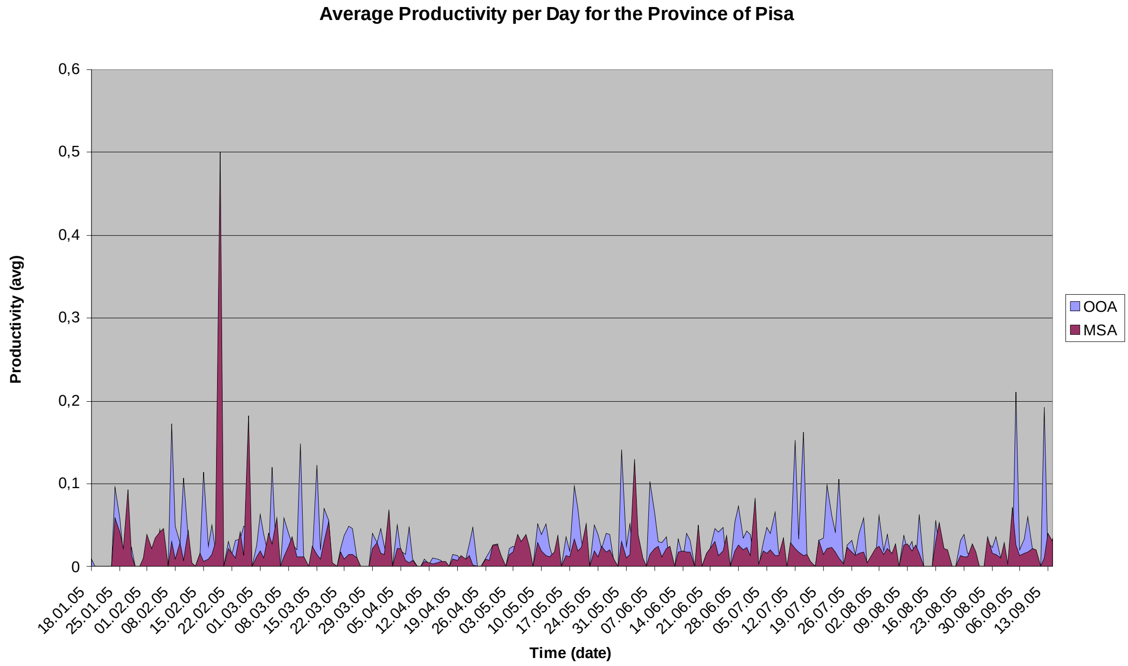
Table 62: Average Time Spent on Documents per User by Day for PP273



The OpenOffice.org productivity – the number of documents opened divided by the time spent on them – is generally higher than the MSO productivity or very close to this one.

[273]MSA is the Application Microsoft Office, OOA is the application OpenOffice.org

Table 63: Average Productivity by Day for PP



Trends of peaks show that users are getting more experienced in using OpenOffice.org than they were at the beginning. People are working faster with OpenOffice.org than with MSO.

The use of OpenOffice.org does not impact on effort and speed of work.

There are some considerations to add at this point. The productivity is higher for OpenOffice.org while the time spent on the OpenOffice.org files is very low. This might mean that the users just tried OpenOffice.org, opening lots of files for small amounts of time. This would increase their productivity and imply a smaller daily effort for the OpenOffice.org users (both pure OpenOffice.org users and switchers). But this is not really the case here, as we have seen that OpenOffice.org was used to perform office tasks similar to the ones performed with MSO and no negative attitude toward OpenOffice.org has been reported.
Public Administration of City of Skopje (SK), Macedonia

12.4.4.2. General Overview of the Migration

In the year 2005 a three month pilot project took place within the Public Administration of the City of Skopje with the goal to test the possibility for migration to Open Source Software. For the experimentation OpenOffice.org was installed on about sixty employees' computers together with Microsoft Office. The IT staff (7 people) also participated in the testing. During the first days of the experiment certain problems were encountered with some computers' configurations (i.e. small amount of RAM memory on old machines that triggered unexpected crashes of the software). As a consequence this part of the participants was dropped from the project and OpenOffice.org and the PROM tool were uninstalled. The rest of the personnel were happy to switch to Open Source products, as they were aware that such migration will solve many problems with software licenses and related costs. The users were aware that with OpenOffice.org they can open and save documents also in MSO format, while the vice-versa is not possible.

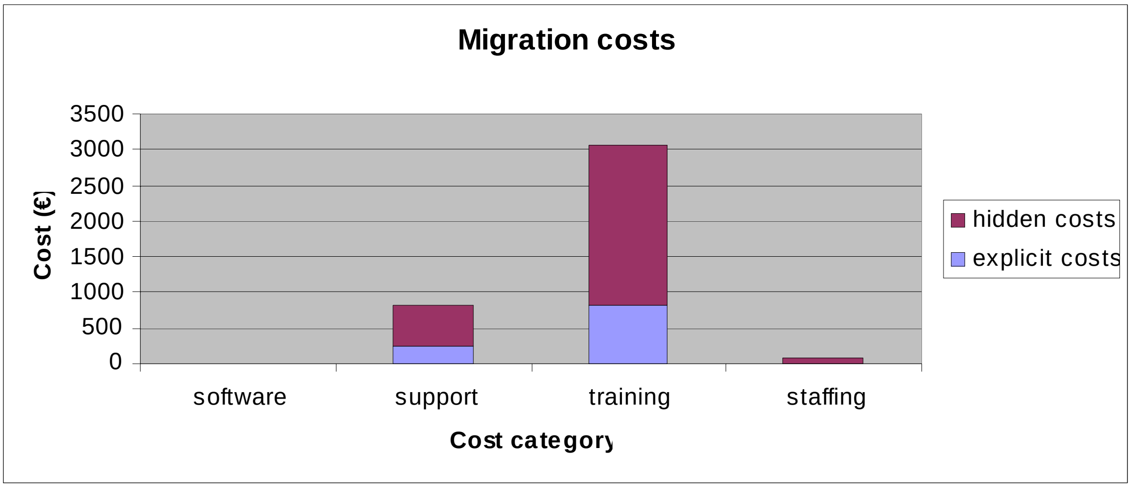
Table 64: Summary of the Migration Effort and Costs for SK[274]

Category	Intangible?	Effort (man/months)	Cost (€)	Subtotal (€)
Software	10			
Interfacing to legacy software		0.05	10	
Support	830			
Search for alternatives	Y	2	400	
Search for documentation	Y	1	200	
Data compliance		1	200	
External support fees		-	30	
Training	3,070			
IT personnel self-learning	Y		1,400	
Employees' training for the new solution			825	
Employees' self-learning	Y		125	
Lack of productivity	Y		720	
Staffing	75			
Installation and deployment	Y	0.38	75	

General Overview of Hidden Costs

In SK hidden costs are significant (Table 65). SK has not performed an extensive migration. By the explicit intention of the management, SK has not allocated a specific budget for the transition. Mainly costs are due to unforeseen small support needs or training. Costs are in any case limited.

Table 65: Tangible and intangible costs at SK



12.4.4.3. Analysis of the Costs by Category

Table 66 describes the costs of ownership at SK by software category. As we already mentioned, costs are low because of the limited availability of budget for IT.

[274] Costs are reported in € as they are not

Table 66: Summary of the Migration Effort and Costs for SK by Software and Category

Cost		Effort (man/months)		Costs (€)	
		Linux	Open Office	Linux	Open Office
Software	Interfacing to legacy software	0.05		10	
Support	Search for alternatives	1	1	200	200
	Search for documentation		1		200
	Data compliance	1		200	
	External support fees			30	
Learning/ Training	IT personnel self learning				1400
	Employees’ training for the new solution				825
	Employees' self learning				125
	Lack of productivity				720

Almost no costs were introduced for software, as no conversion tools or add-ons were needed. A one-day work of one of the IT department members was spent for writing some scripts for interfacing some legacy software, but the estimated costs is negligible.

The migration process usually starts with a search for alternative software solutions and comparison of the available options. For the IT department of the City of Skopje this initial phase was facilitated by couple of factors. More concrete for choosing the operating system for the server side only one person-month was spent which is equal to € 200 (i.e. the average monthly salary of an IT staff member). The chosen alternative was proposed by an external firm which cooperated with the organization in a previous project. Only some functionalities and versions of the same product were tested by the IT staff which can be estimated to a one person-month effort.

Searching for documentation and other sources of information for the chosen Open Source software was needed only for the OpenOffice.org, as the previously mentioned external firm took care of the installation and further support of the server products. The total of one person-month and € 200 were spent. Additionally, for achieving technical and data compliance and interoperability on server-side another person-month (i.e. € 200) were added to the migration costs.

External support was needed only for the server software. As mentioned general support was provided by an external company, but was free of charge in the form of donation to the PA. Nevertheless after a certain period the company started to charge the visits in case of problems. Such cases were very rare and in fact happened only once for the whole period.

Summing up for SK the support expenses were mainly for searching for alternative solution to the proprietary software used, as such cost appeared for each product, as shown on the Figure 20.

The specific situation in the Macedonian software field is that Microsoft based products are used by almost all the population. Although in the recent years there is an increased effort to promote Open Source products they are not utilized in practice. At the beginning of the migration neither the IT personnel (7 people), nor the employees participating in the tests (initially 60 people) were prepared, i.e. properly trained, to work with OSS, namely with OpenOffice.org. As mentioned, the server-side support was fully provided by external experts so almost no training of the IT staff was needed.

The training for the OpenOffice.org work was done fully within the PA. The IT staff spent a rather long period for in-depth self-learning – one month per person, before installing the software to the other employees. The training consisted in reading documentation and forums plus testing the available functionalities of product.

The training of the rest of the personnel was with a combination of approaches. One day training was provided to the personnel in each participating department. In total this training caused a 0.5 person-month (10 days) for the IT staff. During the whole test period and even afterwards (in total 4-5 months) also a helpdesk was available. The help-desk was available before the introduction of Open Source Software. However, after the deployment of OOo one IT department member was spending one hour a day to help the employees deal with OpenOffice.org issues.

At the same time the lack of productivity of the employees, which is an important hidden cost, was estimated by the IT members to 10-20% for the first month. This is estimated to € 720. All these costs make the training/learning factor with the biggest share in the total costs for migration.

The expenses of SK during the migration period for the staffing for the installation of the new software solutions were estimated to € 75. The number of client-side machines on which the OpenOffice.org and PROM were installed was 60, however, on some of them both software were removed due to problems. During the migration no temporary employees were required. Also there were no additional costs for regular employees extra hours or bonuses caused by the migration to open source software.

12.4.4.4. Cost of Use of OpenOffice.org

"The pilot project for migrating to OpenOffice.org in the City of Skopje showed very stable behaviour in the employees' work. Moreover, the absence of a drop of OpenOffice.org usage towards the end of the period suggests that OpenOffice.org was quite capable in substituting Microsoft Office in the appointed tasks, whatever their complexity might have been."

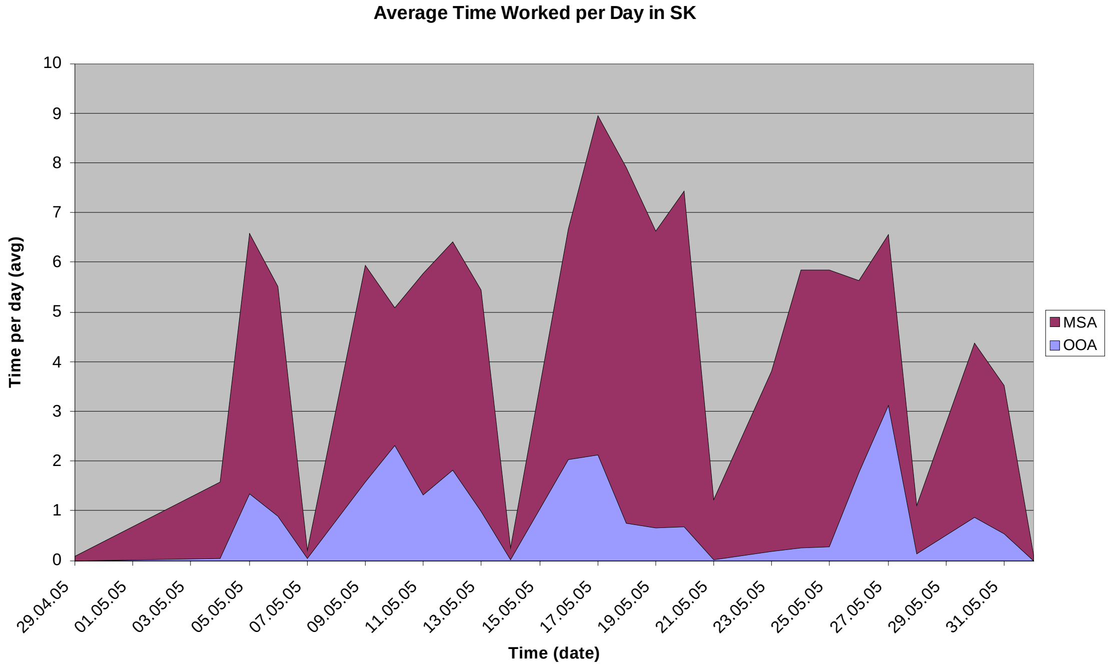
The period of experimentation lasted about two months. However, only the second half of this period has been used for comparing the behaviour of the users of the two platforms as data before was rather scarce, probably due to the fact that the tracking tool was not installed on all the machines.

In total 48 users were monitored, but more than half of them never used OpenOffice.org programs. Only a user was using only OOo during the whole analyzed period, while the others (20) were switching from one application to the other.

Effort and Productivity with the Two Applications

The average time spend (effort) on working with documents each day was generally bigger for MSO than for OOo. Figure 24 shows that the average value is about three times higher for MSO.

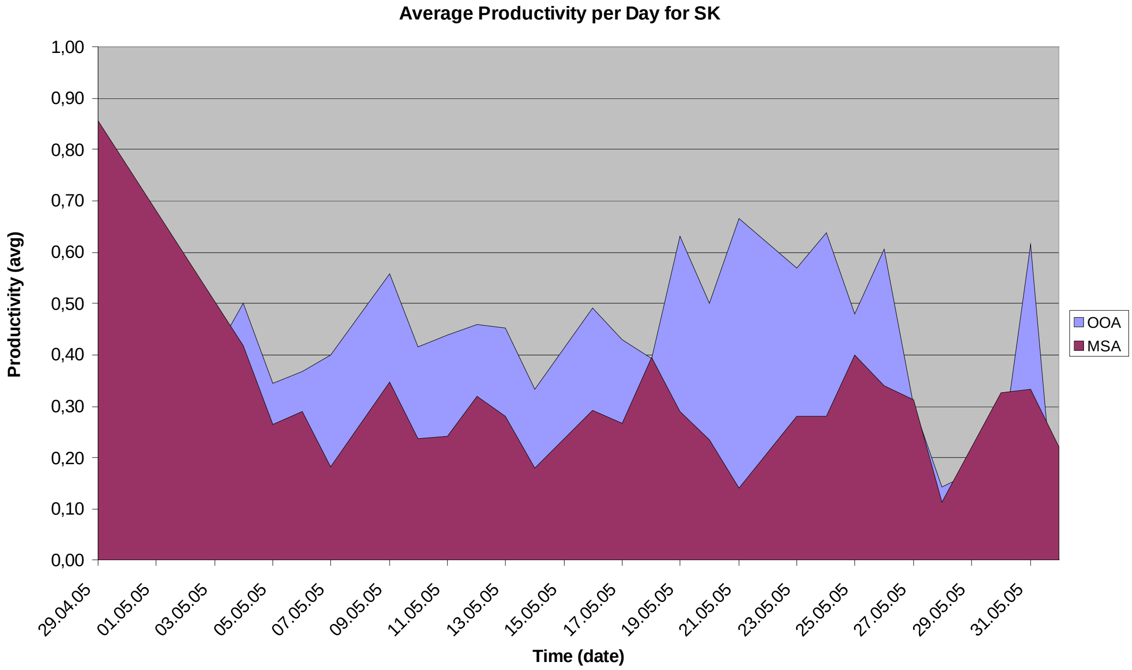
Table 67: Average Time Spent on Documents per User by Day for SK



We can see that the effort was generally higher for MSO than for OOo, however the shapes of the two graphs are very similar. This, together with the omitted analysis for the number of users and documents per day, suggests that both applications were used in similar manner for doing everyday work.

The productivity - the number of documents produced divided by the time spent working on them – shows the "speed of working" with each application.

Table 68: Average Productivity by Day for SK



The productivity for OOo is higher than that for MSO (with few exceptions). In the figure zero productivity means that on that particular day no events were captured. From this we can conclude that people are working faster with OOo than they are working with MSO.

Note. The fact that the productivity is higher for OOo while the effort is very low can be explained with the supposition that the users were merely trying the OOo while relying on the MSO for the actual work. The existence of a big number of users that did not take part of the experimentation, but being monitored may depend on the fact that part of the personnel was using quite old PCs. The IT staff has reported that OpenOffice.org was crashing on these machines, so it was uninstalled short after the installation. The crashes in those earlier versions of OpenOffice.org might be the reason for employee's to switching often to MSO.

Nonetheless OOo do not show a negative impact of the daily work even if it has not conceived as the major application to use.

12.5. Törökbálint Nagyközség Polgármesteri Hivatala (TO), Hungary

12.5.1. General Overview of the Migration

TO is a small Hungarian organization of nearly 40 employees, organized in few departments.

Employees are using old computers based on Microsoft Windows. The network is present but its use is mainly for incoming and outgoing communication, not for internal communication within the PA. For instance, files are exchanged between computers using floppy and not via the network; there are two servers but users do not have access to a network file system.

TO decided to migrate for economical motivations. The fact that European funding would cover most of the costs was a very important decision factor.

Several desktops have been already migrated to use OpenOffice.org instead of the previously used proprietary tool. There has not been much resistance to the change from the users. The users do not have complex requirements and OpenOffice.org is covering them. Users can call the technical support when needed, but no extensive need for such support has been experienced. No external support is needed at the moment, and all the work related to the transition can be done by the personnel of the PA.

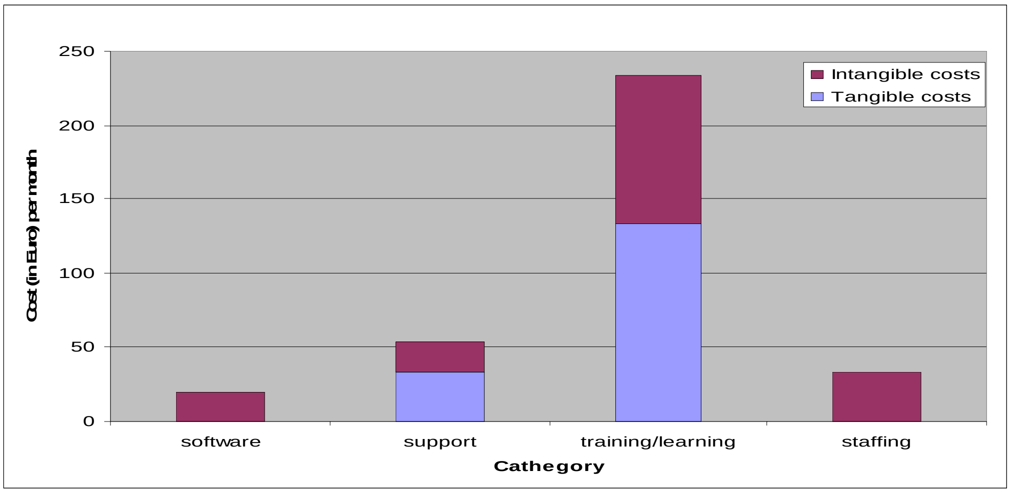
Table 69: Summary of the Migration Effort and Costs for TO

Category	Intangible?	Effort (man/months)	Cost (K€)	Subtotal (K€)
Software	20			
Pilot projects	Y	-	20	
Support	53			
Search for alternatives	Y		13	
Search for documentation	Y		7	
Data compliance			33	
Training	233.5			
IT personnel self-learning	Y		100	
Employees' training for the new solution			133.5	
Staffing	33			

12.5.2. General Overview of Hidden Costs

Figure 26 displays the intangible part of the costs at TO. Total costs were very low and almost all relates to internal personnel. Costs for internal personnel are mainly intangible.

Table 70: Tangible and Intangible Costs in TO



12.5.3. Analysis of the Costs by Category

Cost of software is very low. This includes the purchasing of Fast Ethernet for network of different site of the council, for upgrade of the memory, replacing of old PC. No costs of security tools, upgrades etc. has been reported.

Support for the transition in TO is done mainly internally. In Figure 27 the percentage of internal support is reported. The total time spent is not high.

No external contract has been established and consequently no search for such contracts or external support has been performed.

There has been only self-learning. The IT staff spends about 15% of its work time. There has been internal teaching and help desk run by IT staff. The IT staff spends about 10% of the work time to supply the service, whereas the non-IT staff has dedicated 2% of its time to use help-desk.

Cost for staffing consist only of the internal IT staff (3 people) costing 2 K Euro per month. There were no costs for external personnel temporary hired for the transition, neither bonus or extra hours have been caused by the transition.

12.5.4. Cost of Ownership

Table 71 describes the cost of ownership in the two software settings. The right column reports of the software migrated.

Table 71: Costs of Ownership in the Settings for TO

Software	Open Source Software Solution		Comparable Closed Source Software Solution		Notes *(optional)*
	Initial Cost	Total Cost over 5 years	Initial Cost	Total Cost over 5 years	
OpSys	free	free	€330 per PC	€560 per PC	Windows -> Linux
Office program	free	free	€290 per PC	€490 per PC	MSO -> OpenOffice.org
Virus defensive	do not need,	but Clamav	-	€80 per PC	McAffee -> (Clamav included in UHU-Linux)
Firewall			There was not before		OpenBSD
web server	free	free	There was not before		Apache
mail server	free	free	There was not before		Cyrus

12.5.5. Cost of the Use of OpenOffice.org

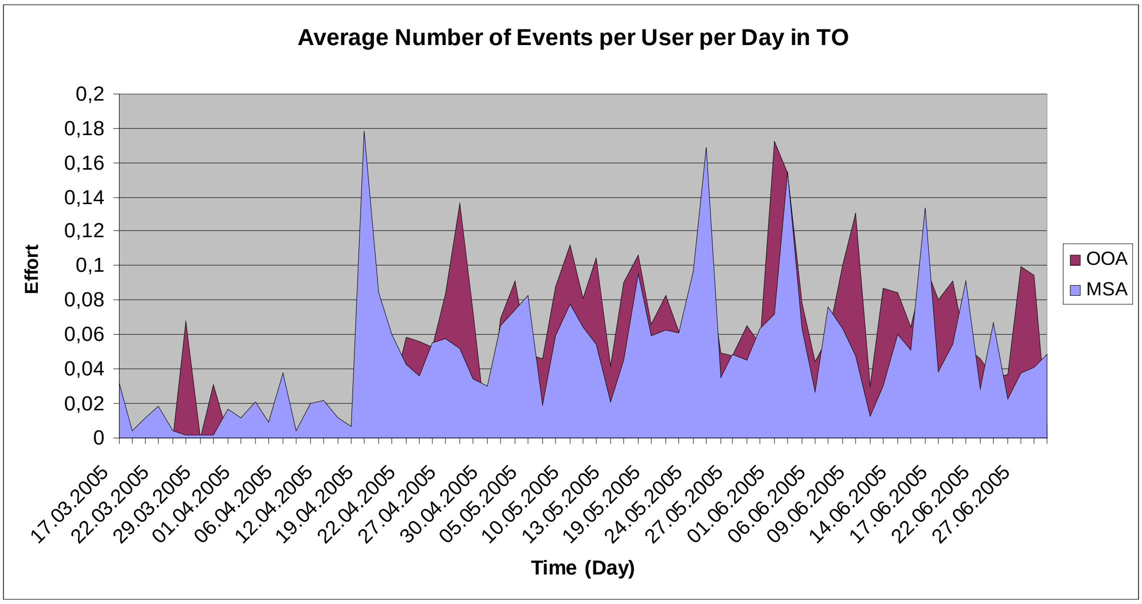
"The analysis of the software usage in TO show that the general pattern of use is similar for the two applications and that the productivity is also comparable in the two cases. Since there were a significant number of switchers, meaning that users are actually participating in the experiment, we can also conclude that the use of OOo could not have a negative impact on the work of the organization."

This analysis is performed on data collected in a period of three months. During the first days of the period, the number of OOo users increases fast and, afterwards it stabilizes to a given average. The number of MSO users suffers a small decrease during the whole period. The significant number of switchers – users that utilized both products within the period - demonstrates no adverse attitude toward the new technology as users are similarly using both platforms.

12.5.6. Effort and Productivity with the Two Applications

The average effort on documents per day is generally slightly higher for OOo than for MSO. Excluding one-event documents does not change the picture significantly.

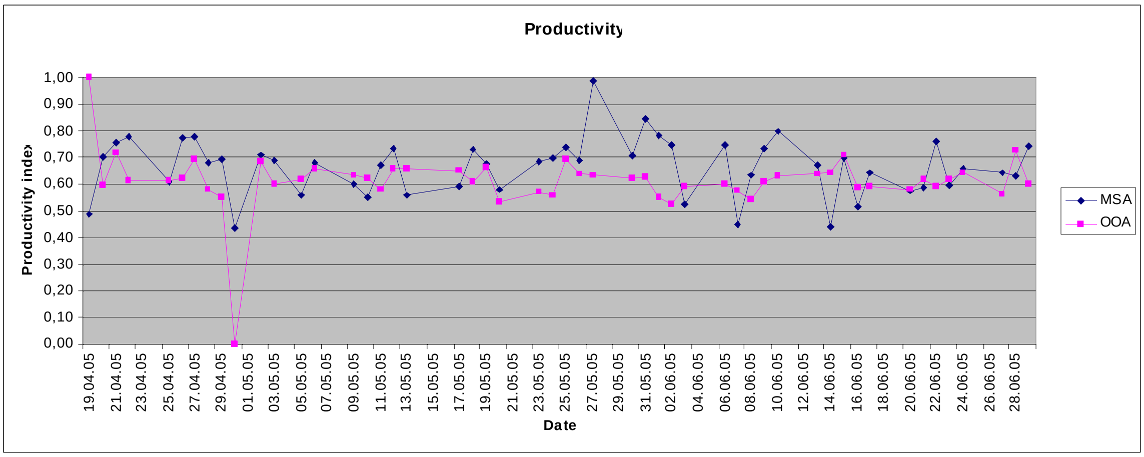
Table 72: Average Number of Events on Documents per User by Day for TO



Considering that the number of events could be used as a proxy for the time spent on documents we can conclude that the effort of working with OOo is slightly higher than MSO.

The users' productivity – the number of documents produced divided by the number of events – gives us the idea of the "speed of working" with each application.

Table 73: Average Productivity by Day for TO



The analysis show that the productivity associated with the use of OOo is higher than the productivity of MSO. Throughout the whole period the productivity of both platforms is regular, with the exception of a high peak on the MSO productivity in the beginning of May that is simultaneous to the absence of activity associated with OO.

We have found that for both applications there were similar number of users, similar number of documents, similar workload and productivity and only some difference in the documents lifespan.

Altogether we might conclude that the way of working with the two applications is comparable and OOo does not have negative impact on the way of work. No generally negative attitude towards the use of OOo was found.

Note. The number of one event documents was quite big in TO (36-40%). Considering the one-event documents, the productivity of OOo was much higher than that of MSO while after having excluded the one-event documents, the productivity of MSO got slightly higher than that of OOo. One reason might be that in this case, the large number of one-event documents compromises the correlation between the number of events for a file and the time spent on that file – as we have used the number of events as proxy of time.

12.6. Beaumont Hospital (BH), Ireland

Beaumont Hospital is one of the largest major general hospitals providing acute hospital care services for the Dublin area and is located on the North side of the city.

The hospital provides acute care services across 54 medical specialties and is the National Referral Centre in Ireland for the specialties of Cochlear Implantation, Neurosurgery and Renal Transplantation. Beaumont Hospital offers both in-patient, day patient, out-patient and casualty services to the population which it serves. Currently, it has a complement of 690 beds.

There are 117 consultants associated with the hospital; 49 of them with admission rights to the hospital, i.e., they have access to in-patient beds. A further 32 consultants have attachments to the hospital but do not have admission rights. In addition to the above, there are 36 consultants with major attachments to the hospital in the areas of Anaesthesia, Laboratory Medicine and Radiology. These specialties, by their nature, do not admit patients but rather provide specialist services to patients of other consultants.

In particular BH has a well organized department for IT. The overall IT environment is characterized by a high heterogeneity of application platforms and associated servers. The Intel-based servers are running Linux and Windows NT, the HP 3000 mainframe runs its own operative system and is used for primary clinical applications (mainly, the radiology information system), whereas the HP Unix system is used for financial applications. The Sun Fire v880, provided by Sun, is used for digital images thank to the 1 TB disk storage.

To introduce IT innovation and overall maintenance in various departments of the hospital, BH has created a specific position of IT expert - an IT manager and IT super user. The cost of this position needs to be spread over each BP.

BH develops in-house solutions or customizations. The annual cost of in-house software development is about € 30,000, involving a ¾ time job for a junior developer and ¼-time for a project manager. For example, the “Nurse Scheduling/Rostering” software cost € 100,000 for 4 people over a year (one manager and three programmers). The cost of maintenance in this case is calculated on the cost of the “Super user”.

12.6.1. Open Source opportunities investigated so far

At the BH, lock-in situations have been analyzed with respect to Open Source solutions. Table 74 lists the OS software investigated. Many of these applications have been tested against BH's software and hardware environment, and some have been abandoned or not adopted. In particular, Vista was long investigated since it could have replaced the core healthcare information system. Other examples are explained in more detail in section 2.2.7.

Many of the OS packages have only been partially adopted, except for Star Office (i.e. the OSS part) and OpenOffice.org. OSS software has been widely adopted in the back-office architecture, the IT departments, or and the Nursing/Rostering service in-developed in-house. The extensive deployment of Star Office has been the major impact on hospital end-users.

Table 74: Open source opportunities investigated at BH

Open source opportunities investigated
PostGres
Junit – unit testing
Jamelon – System testing
Jboss – J2EE Application server
Apache struts – Web application
Framework
Potential Vista system replacement. (then abandoned) Vista is the healthcare information system deployed by the US Department of Defence Veterans Administration (VA). It is available as OSS for a nominal fee
JasperReports - reporting
Eclipse – Integrated development
Cruise Control – Continuous Build tool

12.6.2. Software in use

The categories of software used at Beaumont range broadly from office productivity tools to security applications. There are 3 different kinds of software in use:

The BHI software covers the most important business process actually in use. BHIS (Beaumont Hospital Information System) is a mainframe-based hospital information system. The BHIS was purchased in 1988 and the implementation began in 1989. The product was initially acquired on a 7-year license from a US corporation. This company has since been taken over many times and the current owner is another US corporation HBOC & Co. The initial software license was granted for a period of 7 years (1989–1994). The contract provided for procedures for a license extension on payment of a further fee. However neither party entered into a license renewal arrangement at the time or subsequently. In 2004, HBOC informed BH that they intended to withdraw support for the product from the end of 2005 (initially September 2005).The BHIS application runs on a HP 3000 platform. This is a mainframe computer which has been one of the mainstays of the HP product line for many years. HP announced some years ago that they were withdrawing support for this product from the end of 2006. But they have indicated a willingness to provide a lower degree of support (probably next day support) after that date. Furthermore a range of third party support organisations based in the UK are already seeking to service this market. The HP 3000 itself is an extremely reliable machine. For that reason, the IT manager is confident that we can continue to use it for a further 3 to 5 years.

Single software developed in different years and by different vendors. All these software applications cover business processes not included in the initial structure of BHIS. In-house development mainly focuses on customizing and deploying software, except in the case of nursing /rostering support where in-house products have been developed.

Generic software like browser, word processor etc.

Due to a high budgetary shortfall, Beaumont Hospital started to deploy OSS solutions in 2002, ranging from desktop applications such as Star Office and SuSE mail to Web browsers. For example, like many large organizations, Beaumont has been using email for internal and external communications, and held an 800-user license for Lotus Domino. (CITE article Kenny Fitzgerald) In 2002, there was a demand from the organization to expand the coverage of email to the 3,000 employees, but it was to cost far more than the tight budget available, so SuSE Mail was selected as an alternative email solution. This solution proved inadequate and another open software solution had to be adopted (Section 2.2.7)

The core system relies on the (mainly proprietary) database management systems (in particular, Oracle)

12.6.3. Cost of Ownership in the First Phase

Managers in Beaumont considered the savings in adopting OSS a possible strategic investment option. Table 75 identifies the OSS components implemented in Phase 1, and shows the actual cost savings. The once-off savings over proprietary alternatives are in the order of € 667K. Given that annual maintenance costs are typically about 20% of purchase price, when viewed over a 5-year period, the savings are even more striking, leading to an overall saving of more than €1.5 million for operating system, utilities and desktop applications. In the following table we report of the costs of ownership and purchasing in the two software settings at BH.

Table 75: Cost Comparison of OSS versus Comparable Closed Solutions for BH - Phase 1

Application	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost (€)	Total cost over 5 years (€)	Initial Cost (€)	Total cost over 5 years (€)[275]
Operating System	- (Linux)	150K	77K	384K
Desktop Systems	28K (StarOffice)	35K	120K (e.g. MSO)	289K
Content Management	20K (Zope)	32K	126K (e.g. Lotus Notes)	140K
Application Server	10K (JBOSS)	61K	302K (e.g. Websphere)	595K
Email	10K POSTFIX	15K	110K (e.g. Lotus Domino)	175K

12.6.4. Cost of Ownership in the Second Phase

Table 76 shows the estimated initial costs of purchasing and the ownership costs that would occur over a five-year period from the deployment of the OSS solutions in Phase 2. Again, the initial savings of €6.45 million and the overall savings over a five-year period of € 11.34 million are very significant. Even in the scenario where proprietary financial systems are implemented, the savings from Vista adoption alone would be almost €10M over 5 years. Unfortunately, the adoption of Vista is still under discussion.

[275] Including purchasing costs.

Table 76: Cost Comparison of OSS versus Closed Solutions for BH - Phase 2

Application	Open Source Software Solution		Comparable Closed Source Software Solution	
	Initial Cost (€)	Total cost over 5 years (€)	Initial Cost (€)	Total cost over 5 years (€)
Vista (Based on 1,000 concurrent users)	1.7M	2.5M	7.4M IDX	12.4M
Compiere	10K	60K	760K ISOFT	1.5M
TOTAL	1.71M	2.56M	8.16M	13.9M

12.6.5. Cost of maintenance

In the case of BH we were able to get information on costs of maintenance and initial acquisition costs for three crucial software products in the four major internal services: Data storage, Management reporting, Nurse scheduling, and Patients master indexing.

The maintenance cost depends on the agreement with external providers and the related contractual statements.

The following table summarizes the most widely-used software in BH business processes, and the maintenance costs per single package.

Table 77: Initial acquisition and maintenance of software in the major internal services of BH

Name or functions	Operating. system	Database	Investment cost	Annual maintenance cost
Data repository	HP-UX	Oracle	66000	6880
Management reporting	HP-UX	Oracle	18500	6000
Nurse Scheduling	Linux	Oracle	100000	6000
Patients master index	MPE	Turbo image	1500000	115000

In particular, the sophisticated database for x-rays, TurboImage, has a significant impact on the initial acquisition and maintenance.

12.6.6. Cost of Use of OpenOffice.org

"Adoption of Open Source software started well before the monitoring. Employees have gained some experience with open formats. The expert employees of BH work similarly and produce more documents with OOo than with MSO. Therefore no extra cost but perhaps an intangible return on the investment is experienced in BH. We found that Beaumont Hospital has still to maintain proprietary format for the purpose of document exchange. As top management decision Beaumont Hospital is considering though to partially migrate back to proprietary software."

The migration to Open Source software (Star Office, in fact) at Beaumont Hospital started long before the data collection with the PROM tool that in turn lasted more than five months. We selected a representative period of two months. The top management has then taken the decision to migrate back to proprietary solutions.

The total number of employees monitored is 210. Very few participants have used MSO more than one time during the whole period and about half of those have used only MSO. Less then 10% of all the participants was switchers – users that utilized both products within the period that is analyzed. The analysis of the daily use of the applications reports of a constant increase of the daily use of OO.

Table 78: Users at BH

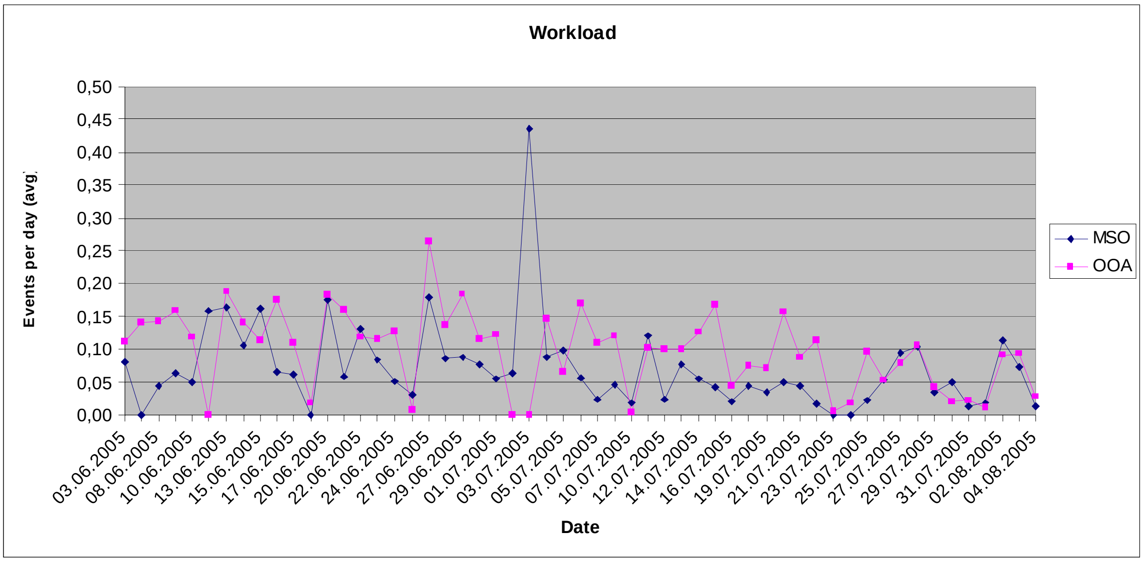
Type of users	Number
Users that used MSO at least once during the whole period	37
Users that used OOo at least once during the whole period	192
Users that used only MSO during the whole period	18- 8.57% of all; 48.64% of MSO
Users that used only OOo during the whole period	173 (82.38%)

Using events on the documents (office activities performed to work on the document, like "save as," "print," etc.) as proxy of time we could trace the trend of use of both the application. We have deduced that the usage of both the application is comparable, that is complexity of the use and time spent are similar with a little predominance of OOo.

12.6.7. Effort and Productivity with the Two Applications

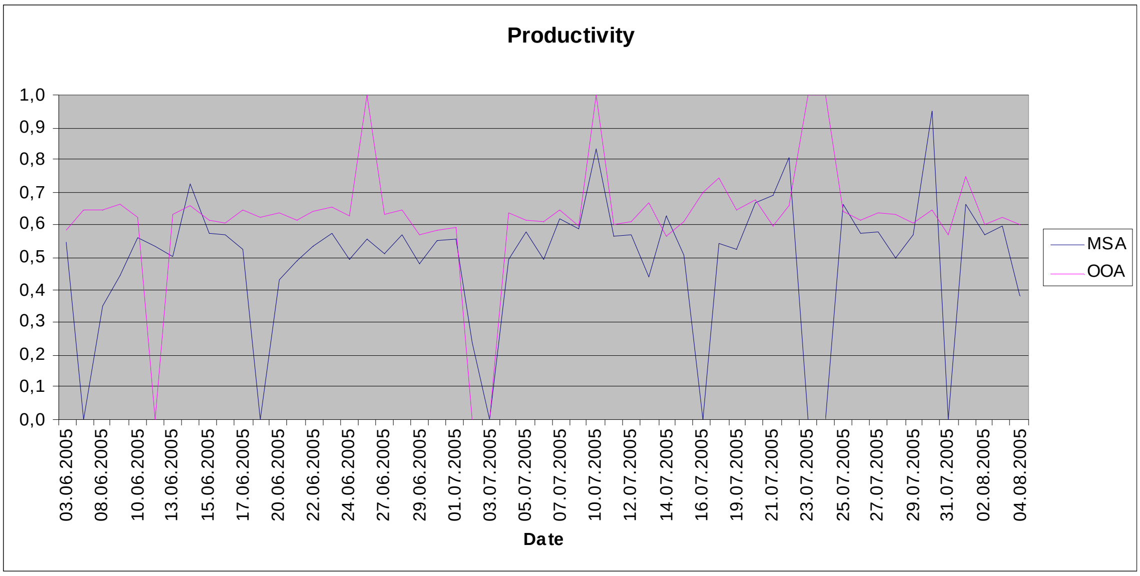
Figure 30 shows that measuring the productivity, that is the "speed of working", as the number of documents produced divided by number of events, there us a predominance of OOo.

Table 79: Average Number of Events on Documents per User by Day for BH



In conclusion the expert employees of BH work similarly and produce more documents with OOo than with MSO.

Table 80: Average Productivity by Day for BH



To check that users of different samples had similar tasks we performed an analysis on the distinct documents opened per user per day analysing the way to work on a single document. We have found that the way to work is pretty similar with a higher number of activities performed with OOo. As employees are experienced in using OO, they use OOo on documents requiring more activities (like print, save as, print all, etc.). To support the claim that users have similar tasks we restricted the analysis on document that are shared by two or more employee. We found that the trend of events is similar in the two applications.

We also found that a substantial number of MSO documents are opened in OOo. A possible explanation is that those files are meant to be exchanged between the two platforms. This may mean that BH has to exchange documents with external organizations which do not use open format.

12.7. Conclusions

Our analysis has been performed on six organizations in different European countries. The majority of them are public bodies. The organizations have followed different types of migration on the base of their context.

We have investigated the costs of migration, and the cost of ownership of the old and the new solution differentiating them between the costs of purchasing and the costs of ownership of the software solutions. Special attention has been put on the intangible nature of the costs. Costs have been classified in categories defined trough existing studies and selected by a top down approach called Goal Question Metric. This instrument has been also used to define the questionnaires used to collect the data.

Our findings show that, in almost all the cases, a transition toward open source reports of savings on the long term – costs of ownership of the software products.

Costs to migrate to an open solution are relevant and an organization needs to consider an extra effort for this. However these costs are temporary and manly are budgeted in less than one year. The major factor of cost of the new solution – even in the case that the open solution is mixed with closed software – is costs for peer or ad hoc training. These are the best example of intangible costs that often are not foreseen in a transition. On the other hand not providing a specific training may cause and adverse attitude toward the new technology. Fortunately those costs are limited in time and are not strictly linked to the nature of the new software adopted.

We also investigated the productivity of the employees in using Microsoft office and OpenOffice.org. Office suites are widely used and are a good test bed and representative for a comparison on issues like effort and time spent in the daily routine of work. Delays in the task deliveries may have a bigger impact than costs on the organization's management. Our findings report no particular delays or lost of time in the daily work due to the use of OpenOffice.org.

12.7.1. Considerations

With our analysis we achieve a good level of understanding of the costs, benefits and productivity of a transition. The following are the considerations we have drawn upon.

1. Before buying, upgrading proprietary office software one needs consider that:

OpenOffice.org has all the functionalities that public offices need to create documents, spreadsheets, and presentations

Upgrading office programs is time-consuming and expensive. It requires installation time, potential document conversions, and new training. It also poses a risk because some documents containing code or macros may not be readable anymore

OpenOffice.org is free, extremely stable, and supports the ISO Open Document Standard.

2. In our study the motivations to transit to OSS are: the exchange of documents in an open shared format (ODS), reuse of old hardware in some cases, and being independent of software vendors even when creating a distribution or an application for local needs.

Employees may perceive that their work is under-valued using 'cheap' OSS products or changing operating model to OSS is problematic.

To overcome these pre-conception it is recommended to adopt a policy of both ad hoc and periodic training to fill the lack of knowledge/experience in relation to what OSS products are appropriate and how they might be deployed.

3. It is not always justified to base the migration on the promise of lower license costs, although in our study initial purchasing costs are lower for the OSS (they includes deployment and customization for the first run of the configuration). This is because these costs are too much influenced by factors like inflation and market flow. .

4. A model that differentiates between cost of migration and costs of ownership better respond to the managers' needs. The former involves high investment for a shorter period, while the latter foresees expenditure for maintenance over a period of at least five years

In the model drawing a fundamental factor is the intangible part of costs that often is neglected. Intangible costs might be a substantial factor

Another good crucial reason of costs is training. Although training costs are a substantial part of the migration costs their benefits can be realized over time. The migration process is also an opportunity to provide users with formalized training on the software applications they use, improving productivity in a significant and measurable way. A deeper knowledge of the software infrastructure gives more power to IT managers in negotiation with external consultants or vendors.

5. There are no extra costs due to lack of productivity arising from the use of the OOo.

Acknowledgements

Data collection has been partially supported by the COSPA project - VI European Framework Contract no.: IST-2002-2164. We also thank all the organizations participating in this study.

12.8. References

[Gartner, 2003] Distributed Computing Chart of Accounts.

http://www.gartner.com/4_decision_tools/modeling_tools/costcat.pdf

[Linux ROI] Return on Investment on Linux Migration/Purchase. www.linuxvalue.com/linux.xls

[Shapiro, 1999] Shapiro, C. and H.R. Varian, “Information rules: a strategic guide to the network economy”, Harvard Business School Press, Boston, 1999

[Winslow, 2004] Winslow M., Evaluating the ROI of Open Source on the Desktop (Linux and OpenOffice are ready for your business, 2004, - web site: http://linux.syscon. com/read/43720.htm

13. References

Aghion, P. and P. Howitt (1998). Endogenous growth theory. Cambridge and London, MIT Press.

Barro, R. J. (1999). "Notes on Growth Accounting." Journal of Economic Growth **4**(2): 119-37.

Barua, A. and C. H. Kriebel (1991). Information Technologies and Business Value : An Analytic and Empirical Investigation., Carnegie Mellon University, Graduate School of Industrial Administration, GSIA Working Papers.

Basu, S., J. G. Fernald, et al. (2003). The Case of the Missing Productivty Growth: Or, Does Information Technology Explain why Productivity Accelerated in the US but not the UK?, National Bureau of Economic Research, Inc, NBER Working Papers: 10010.

Basu, S., J. G. Fernald, et al. (2001). Productivity Growth in the 1990s: Technology, Utilization, or Adjustment?, National Bureau of Economic Research Working Paper: 8359.

Becchetti, L., D. A. Londono Bedoya, et al. (2003). "ICT Investment, Productivity and Efficiency: Evidence at Firm Level Using a Stochastic Frontier Approach." Journal of Productivity Analysis **20**(2): 143-67.

Colecchia, A. and P. Schreyer (2002). ICT Investment and Economic Growth in the 1990s: Is the United States a Unique Case?: A Comparative Study of Nine OECD Countries. Review of Economic Dynamics. **5:** 408.

Daveri, F. (2002). "The New Economy in Europe, 1992-2001." Oxford Review of Economic Policy **18**(3): 345-62.

Gordon, R. J. (2000). Does the "New Economy" Measure Up to the Great Inventions of the Past? Journal of Economic Perspectives. **14:** 49-74.

Gordon, R. J. (2004). "Two Centuries of Economics Growth: Europe Chasing the American Frontier." Mimeo, Northwestern University and CEPR.

Harris, R. G. (1998). The Internet as a GPT: Factor Market Implications. General purpose technologies and economic growth. Cambridge and London, MIT Press**:** 145-66.

Helpman, E. (1998). General Purpose Technologies and Economic Growth: Introduction. General purpose technologies and economic growth. Cambridge and London, MIT Press**:** 1-14.

Jorgenson, D. W. (2001). "Information Technology and the U.S. Economy." American Economic Review **91**(1): 1-32.

Kaldor, N. (1961). Capital Accumulation and Economic Growth. in: The Theory of Capital. F. A. Lutz and D. H. (eds.), International Economic Association, MacMillan, London**:** 177-222.

Klemperer, P. and J. Farrell (2006). Co-ordination and Lock-in: Competition with Switching Costs and Network Effects, Economics Group, Nuffield College, University of Oxford, Economics Papers: 2006-W07.

Lucas, R. E. J. (1988). "On the Mechanics of Economic Development." Journal of Monetary Economics **22**: 3-42.

Meijers, H. (2004a). "ICT and network effects in Europe and the US." MERIT, Maastricht, Paper presented at the ENCIP EURO CPR Conference Barcelona, 28-30 March 2004.

Meijers, H. (2004b). "Productivity in Europe and the U.S.: ICTs and the Role of Network Effects." Communications and Strategies **0**(54): 161-88.

O'Mahony, M. and M. Vecchi (2005). "Quantifying the Impact of ICT Capital on Output Growth: A Heterogeneous Dynamic Panel Approach." Economica **72**(288): 615-33.

OECD (2004). "The Economic Impact of ICT; Measurement, evidence and Implications."

Romer, P. (1990). "Endogenous Technological Change." Journal of Political Economy **98**(5): S71-S102.

Shapiro, C. and H. R. Varian (1999). Information rules: A strategic guide to the network economy. Boston, Harvard Business School Press.

Solow, R. M. (2000). Growth theory: An exposition. New York and Oxford, Oxford University Press.

Stiroh, K. J. (2002). "Are ICT Spillovers Driving the New Economy?" Review of Income and Wealth **48**(1): 33-57.

van Ark, B., R. Inklaar, et al. (2003). "The Contribution of ICT-Producing and ICT-Using Industries to Productivity Growth: A Comparison of Canada, Europe and the United States." International Productivity Monitor **0**(6): 56-63.

van Ark, B., J. Melka, et al. (2002). "ICT Investments and Growth Accounts for the European Union." GGDC Research Memorandum GD-56 **(revised 2003)**.

van Zon, A. H. (2001). ICT-Investment, Knowledge Accumulation and Endogenous Growth, MERIT, University Maastricht.