Project acronym: OSEPA

Project name: Open Source software usage by European Public Administrations

Project code: INTERREG IVC, 0918R2

Document Information:

Document title: Report on the economic foundation of open source software.

Date of Delivery: 24.06.2011

Component: CP3

Component Title: Exchange of experiences

Component Leader: University of Sheffield

Distribution (Restricted/Public): Restricted to the consortium

Nature: Report

History Chart

Date	Changes	Cause of change	Implemented by
24.06.2011	Initial Document	N/A	Computer Technology Institute & Press CTI "DIOPHANTUS"

Authorisation

No.	Action	Partner	Date
1	Prepared	Computer Technology Institute & Press CTI "DIOPHANTUS"	24.06.2011
2	Approved		
3	Released		

Disclaimer

The information in this document is subject to change without notice.

Table of Contents

Abbreviations **3**

Executive summary **4**

1. Introduction **6**

1.1. Scope and context *6*

1.2. Issues addressed *7*

2. What defines FOSS? Features and properties **9**

2.1. FOSS features and properties *9*

2.2. FOSS use rights and licence types *11*

3. What drives the growth of FOSS? Actors, mechanisms and motivations **14**

3.1. Demand for FOSS solutions *14*

3.2. FOSS supply: enterprises, developers and communities *18*

3.3. Commercial and non-commercial mechanisms in developing and distributing FOSS *23*

4. What makes FOSS economically efficient? Software market strategies **25**

4.1. Competing or converging? Open and closed source licensing regimes *25*

4.2. Open source business strategies *26*

4.3. The added value of FOSS: managing complexity *30*

5. The economic assessment of FOSS: conclusions for public organisations **32**

5.1. The economic assessment of FOSS: beyond cost analysis *32*

5.2. FOSS as a public good and competitive advantage *33*

6. Annex **35**

6.1. Research methodology *35*

6.2. References and resources *36*

Abbreviations

BSD	Berkeley Software Distribution
EC	European Commission
EPA(s)	European Public Administration (s)
EULA	End-user licensing agreement
EUPL	European Union Public Licence
FSF	Free Software Foundation
FOSS	Free/ Open Source Software
GPL	General Public Licence
IDABC	Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens
Iaas	Infrastructure as a Service
ICT	Information & Communication Technologies
ISV(s)	Independent Software Vendors
IT	Information Technology
LGPL	Lesser General Public Licence
OSEPA	Open Source Software Usage by European Public Administrations
OSI	Open Source Initiative
OSS	Open Source Software
PaaS	Platform as a Service
CTI	Computer Technology Institute & Press CTI "DIOPHANTUS"
R&D	Research and Development
SaaS	Software as a Service
SME(s)	Small and Medium Enterprise(s)
TCO	Total Cost of Ownership

Executive summary

This report investigates the actors, properties and demand and supply mechanisms that define the economic dimension of Free and Open Source Software (FOSS) and drive its increasing growth within the broader market environment it is produced and distributed. It provides an overview of both "user" and "producer" roles, motivations and business strategies that make FOSS work in economic terms. The aim of the report is to contribute to building up knowledge resources that can help European Public Administrations (EPAs) assess FOSS solutions based on economic efficiency and competitive advantages.

This document comes as an output of the OSEPA (Open Source Software Usage by European Public Administrations) EU-funded INTERREG IVC project aiming to assess and promote the uptake of Free and Open Source Software (FOSS) in public administrations.

FOSS is developed, supported and promoted by both enterprises and communities driven by various economic, technological and social motivations. Within the market software context, FOSS is used as a vehicle to reduce entry barriers and disrupt the dominant position of market leaders in specific market segments.

The growth of FOSS has come as a response to endogenous market needs. It responds to the need for optimised efficiency in managing software complexity, this being its main competitive advantage. FOSS is developed, tested and evaluated by wide user and developer communities on a scale and rate beyond the capacities of most IT companies or organisations. As a result, it offers stable and customisable solutions with a higher penetration in contexts where increased component interactions and advanced implementations are required (e.g. middleware).

In this context, FOSS seems to operate complementary as an extension and not necessarily a competitor of proprietary software. In market reality, "hybrid" business strategies blending elements of both open source and proprietary licensing regimes harvest the competitive advantages of both models in order to achieve flexibility and economic efficiency.

FOSS also brings particular challenges and opportunities for governments and public administrations rising from its source code availability, its re-use and customisation capabilities and its special character as a complex public good. By integrating FOSS solutions in their IT policies and infrastructures, public administrations can improve their strategic position in the software sector in terms of ensuring data openness and interoperability, identifying economically efficient solutions tailored to their specialised needs and achieving long-term independence from vendors and suppliers. In this sense, public organisations are encouraged to assess FOSS not merely on a cost analysis basis, but also considering the long term benefits and competitive advantages it has to offer.

1. Introduction

1.1. Scope and context

This report investigates the economic foundations of Free and Open Source Software (FOSS) providing a basis for its overall economic assessment. The report is a foreseen output of the OSEPA (Open Source Software Usage by European Public Administrations) project aiming to assess and promote the uptake of Free and Open Source Software (FOSS) in territorial public administrations.

One of the main objectives of the OSEPA project is to provide economic assessment insights of open source software, not only in comparison to proprietary software solutions but also in the light of major market trends and technological developments in the software sector. Within this scope, this report contributes in building up knowledge resources that can help public administrations understand the underpinning concepts and the major mechanisms that drive the production and distribution of FOSS. It is also relevant to developers, users and communities who are either directly or indirectly involved in FOSS demand and supply processes.

More specifically, this report targets three main recipient groups:

1. **European Public Administration representatives**, senior administrative staff, IT and procurement managers.

2. **FOSS providers and entrepreneurs:** individuals involved in developing and supporting, distributing and marketing FOSS solutions.

3. **FOSS users, developers and communities:** FOSS user and developer communities, workgroups and collaboration teams maintaining and supporting FOSS projects.

Providing a full comparative assessment of FOSS vs. proprietary software solutions is beyond the scope of this document. This report focuses, instead, on setting a broader context for assessing the underlying properties, factors, and mechanisms [1] that define the economic dimension of FOSS within the market environment that it is being produced and distributed.

1.2. Issues addressed

The issues covered in this report relate to the properties, demand and supply mechanisms and marketing and distribution strategies of FOSS. Three main issues are addressed:

1. ***What defines FOSS?***
   What are the properties of FOSS and how do they relate to the ways in which it is produced and distributed?
2. ***What drives the growth of FOSS?***
   What are the main actors, motivations and demand / supply mechanisms for producing and distributing FOSS?
3. ***What makes FOSS economically efficient?***
   What are the business strategies that make FOSS efficient in economic terms?

Within these three main areas certain questions are examined:

1. *What is the specificity and added value of FOSS?*
2. *Who are the main users and producers of FOSS? What are the user needs and how are they covered by FOSS? What motivates FOSS producers, developers and communities?*
3. *What are the different open/closed source licensing regimes that operate in the software market? How are they combined?*
4. *Does FOSS constitute an economically efficient solution for public organisations? What are the critical aspects to consider?*

[1] defined throughout this report as the "economic foundation(s)" of FOSS.

Chapter 2 of this document outlines the defining features and specific properties of FOSS. Chapter 3 investigates the main actors and mechanisms that define FOSS demand and supply and drive its growth. Chapter 4 examines the main business strategies that make FOSS an economically effective solution. Chapter 5 sums up key points and highlights critical aspects to be considered by public organisations in assessing FOSS as an economically efficient, strategic choice.

2. What defines FOSS? Features and properties

2.1. FOSS features and properties

Control over the use and distribution of software is the differentiating factor between the open source and proprietary model for software production. Proprietary software is based on the value of a fixed and exclusively owned property right that is protected by trademarks or patents and its usage can only be transferred under certain conditions that are usually stated in end-user licence agreements (EULAs). FOSS, on the other hand, relies on use rights that in most cases are non-excludable, non-rival and limitlessly transferable. [2] In this sense, FOSS sets, from the very beginning, the foundation for a different perspective on generating and harvesting software value.

Free and / or open source software is software that can be freely run, distributed and modified by accessing its source code. Although there are different definitions of FOSS, there are some basic principles and properties[3] on which they all rely. These refer to:

- *the freedom to run a software program for any purpose*
- *the freedom to study and modify a software program by accessing its source code*
- *the freedom to distribute copies of a software program, whether modified or not*

Some additional requirements and specifications for open source software have been defined by the Open Source Initiative:

[2] See: Daffara C., "The basis of OSS business models: property and efficiency", July 26, 2010, http://carlodaffara.conecta.it/category/oss-business-models/ and Bessen J., "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005).

[3] These "freedoms" and principles are defined by the Free Software Foundation: http://www.gnu.org/philosophy/free-sw.html and the Open Source Initiative: http://www.opensource.org/osd.html

- *licence compliance of the derived works with the original software*
- *integrity of the author's source code*
- *no discrimination against persons, groups or endeavour fields in open source contributions.*
- *licence distribution*
- *licence not tied to a specific product or restricting other software*
- *licence must be technology neutral*

The inherent features and properties of open source software set a different paradigm for software development and use than that of proprietary software. Contrary to closed-source software that is based on proprietary licensing, open source does not restrict but promotes the ability to use, copy, distribute and modify the software. As a result, it is decoupled of exclusive rights of use and acquires the features of a public good: it is a mutually non exhaustive resource in the sense that its use by a user or an organisation does not reduce its utility for other potential users. It also employs an open and participatory software development scheme based on ongoing contributions that build up a pool of open, accessible software code and related resources for anyone to use and draw from. Code contribution in open source repositories that are usually moderated by FOSS communities and non-profit foundations[4] is not coming merely from volunteers but also, to a critical extent, by IT firms and enterprises.[5] This open and collaborative model of software production shifts the conventional linear supplier-to-customer pattern to more complex schemes, based on user involvement, multiple producers and continuous feedback and community support.

[4] such as the Apache Software Foundation, or the Mozilla Foundation.

[5] Recent research points out that 50% to 90% of the programming effort invested in FOSS development is directly or indirectly contributed by businesses. See Bessen, J. "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005).

It should be noted, however, that the business growth of FOSS, could not be fully explained based on its collaborative software development model as it also relates to market competition practices and rival business strategies of IT firms.[6]

2.2. FOSS use rights and licence types

Open source software licensing schemes are different than proprietary licensing regimes in the sense that they do not pose restrictions on the scale and extent of software usage. Instead, they promote and encourage software distribution, copying and modification under certain conditions, the most common of which is to release any modified software under a same licence type in order to maintain free code availability.

Open source licences should not be associated however, with the public domain as they retain intellectual property rights for software and define certain obligations for the licensee rising from the licence agreement.

Open source software licence types also vary in terms of provisions defined and permissions granted regarding the freedom to reuse, distribute and integrate code in new software products. OSS licences could be divided into:

- "permissive', or attribution-style licences (e.g. Berkeley Software Distribution / BSD, MIT licences) allowing developers to reuse software and re-release derivatives under any other licence type whether open-source or closed-source (proprietary).

[6] Daffara (2009) makes a distinction between a "software model" associated with control, a "development model" defining the scale of collaboration and a "business model" linked to strategies for revenue streams. See: Daffara C., "Economic Free Software perspectives," April 5, 2009, http://carlodaffara.conecta.it/category/oss-business-models/page/3/

- "copyleft", share / share-alike licences (e.g. GNU General Public Licence / GPL) requiring that any software modifications or redistributions are released under a same licence type and are freely and openly available.[7]

Taking a closer look at software licence families, three licence groups are presented here: the Berkeley Software Distribution (BSD) licences, GNU General Public Licence (GPL) and the European Union Public Licence (EUPL). Although EUPL is considered a "copyleft", licence type, it is presented separately due to its specific relevance for public administrations in the EU context.

The BSD licences

The BSD licence family[8] also includes the MIT X licences and therefore is also referred to as BSD/MIT, MIT X, X or MIT licences. Under the BSD licensing regime, source code distribution is allowed but not required for derivatives. Therefore any software released under a BSD licence can be combined with or integrated in proprietary software systems and components. BSD licensed software has been integrated in major proprietary software systems such the MacOSX operating system.

The GNU General Public Licence

Unlike BSD licences, GNU General Public Licence[9] explicitly prohibits software redistribution or modification without including the source code. GPL requires that all source code modifications are released under GPL as well ("viral licence" model). Those simply wishing to use a GPL licensed software application without modifying the source code have no further obligations. In case, however, of a task or project involving source code modification and redistribution, the obligation of releasing under GPL should be taken into account. It should be

[7] There also "hybrid" licence types such as the Mozilla Public licence (MPL) that combines features of both GPL and BSC licensing models. MPL is available at: http://www.mozilla.org/MPL/MPL-1.1.html
[8] For a description of BSD licence, see http://www.linfo.org/bsdlicence.html
[9] http://www.gnu.org/licences/gpl.html

noted that GPL does not oblige an end-user or an organisation to release any source-code modifications but requires, instead, that in case a decision is made to release any modifications or redistributions, GPL will be used. A variation of GPL is the GNU Lesser General Public Licence (LGPL).[10]

The European Union Public Licence

The European Union Public Licence (EUPL)[11] is a licence issued by the European Commission that aims to promote the use and distribution of software by European institutions under a Free/Open Source Licence in compliance with the European law requirements. This licence, specifically developed by and for the European Commission could be used as a standard in case of FOSS redistribution releases by European public agencies and organisations.

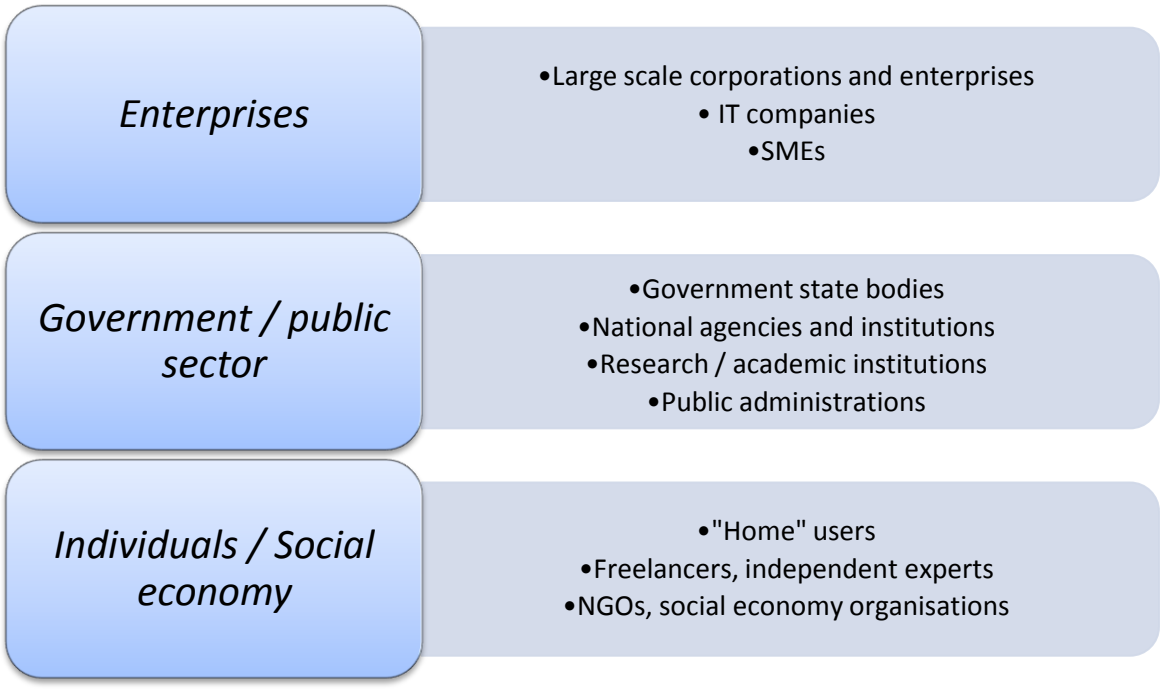
[10] http://www.gnu.org/licences/lgpl-2.1.html
[11] http://www.osor.eu/eupl/

3. What drives the growth of FOSS? Actors, mechanisms and motivations

3.1. Demand for FOSS solutions

There is a wide range of enterprises, organisations, communities and groups of individuals that collectively define and articulate the demand for FOSS solutions. FOSS users and stakeholders have different sizes and profiles, priorities and fields of activity varying from global corporations to small and medium enterprises or from governments to single home users.

Figure 1. FOSS users and stakeholders.



Enterprises

Software and IT companies are among the biggest software users themselves heavily relying on application stacks not only to meet their internal operational needs but also to support the provision of services to their clients. [12] Enterprises and IT companies have certain economic incentives in adopting free and open source software as a strategic component enabling them to:

- reduce maintenance and R&D costs that in the case of FOSS are shared among several actors and stakeholders.[13]
- achieve independence as software users from software vendors and competitors.

Enterprise software solutions (e.g. ERP, CRM, Project Management, Knowledge Management etc.) occupy a critical share of the software market and have attracted an increasing competition on applications, system administration platforms and middleware (e.g. application servers and application management tools).

Government / public sector

Public organisations collectively represent a critical, mass-scale software consumer and end-recipient of associated IT services with significant influence on software product specification and licensing agreements. Depending on their scale, organisational profile and the specialised administrative and operational tasks they have to undertake (e.g. e-government services, tax administration, human resources management), public organisations often seek custom developed FOSS-based services and solutions that can be tailored to their specific needs. FOSS,

[12] Daffara C., "The basis of OSS business models: property and efficiency", July 26, 2010, http://carlodaffara.conecta.it/category/oss-business-models/

[13] According to Ghosh (2006), savings in R&D investment can potentially range over 36%. See: Ghosh, R. A (ed.), *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 (European Commission, November 20, 2006).

allowing for maximum customisation, constitutes an attractive choice for public stakeholders that can also help them minimise licence purchasing costs. With cost cutting currently being a priority in the public sector, it is expected that FOSS will retain its attractiveness for public organisations.

Moreover, due to their public service orientation, public agencies and administrations have further incentives to integrate FOSS in their IT strategies and infrastructures as they have to comply with open access and transparency requirements not just in software procurement but also in daily operation. The software related needs and prerequisites that are specific to public organisations could be summed up as follows:

1. ***Open standards / interoperability:*** public bodies, being obliged to facilitate the access of citizens to public data and to support information exchange, are expected to adopt open standard requirements and specifications. The interoperability of software systems and applications through the use of open standards is a prerequisite for any IT strategy in the European Union and has been defined as such in the European Interoperability Framework[14] providing guidelines on the implementation of open standards among public agencies and organisations.

2. ***Flexibility:*** Organisational needs and operational requirements in the public sector change over time often requiring large-scale adaptation, fine-tuning or re-structuring of entire IT architectures. This poses the need for flexible, highly scalable and customisable systems and applications that can be directly adjusted according to organisational needs.

3. ***Transparency:*** Software procurement, selection and integration procedures should be kept open and transparent at all stages in order to promote competition fairness, public information accessibility and accountability. Software system architectures, features and

[14] http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529

functionalities should be also as visible as possible so they can be benchmarked, evaluated and modified if needed to meet the particular needs of public organisations.

4. ***Vendor independence***: Avoiding vendor lock-in is a top priority for any public organisation planning to acquire IT systems and applications. Not being tied up to proprietary trademarks and single vendors for support and maintenance is critical if supplier independence is to be achieved. Several public agencies and organisations opt for FOSS solutions as an effective way to achieve such and independence.

It should be noted that the government / public sector demand on FOSS solutions still remains largely unfulfilled mainly due to: a) poor supply in terms of quality regarding purpose built applications and specialised solutions b) lack of awareness on available solutions and competitive offerings.

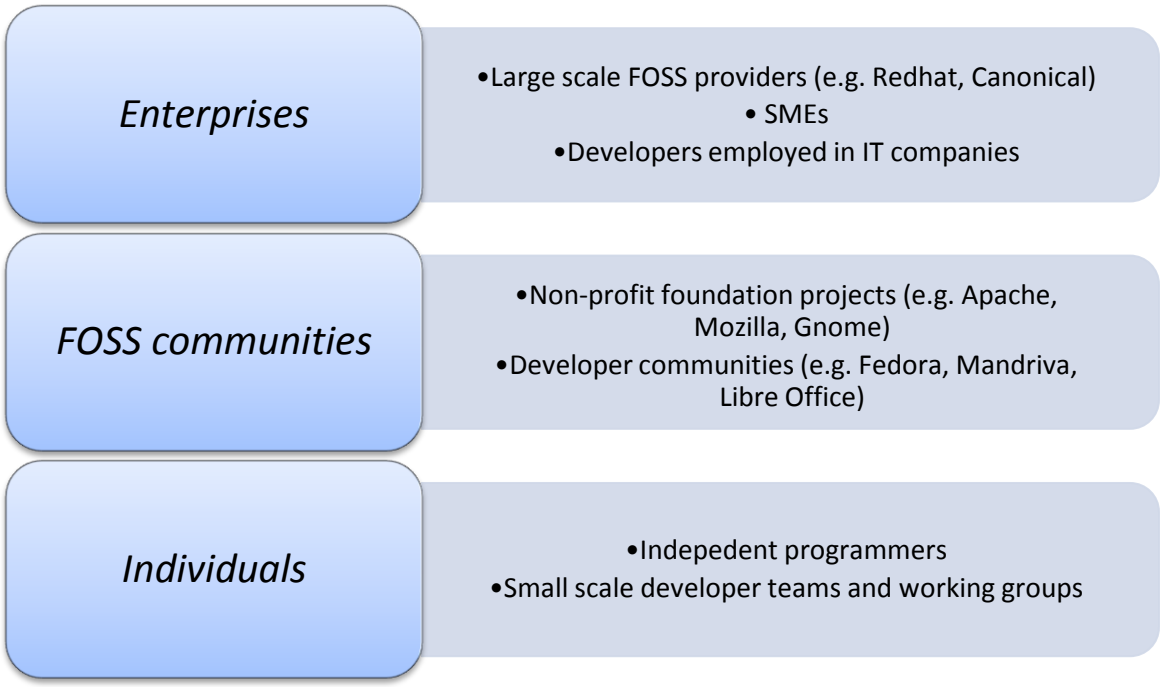
Individuals / Social economy organisations

FOSS also serves the needs of individuals or small scale organisations ranging from mainstream applications (e.g. Mozilla Firefox, Libre Office, Gimp) and operating systems (e.g. Ubuntu, OpenSUSE) to more sophisticated solutions (e.g. MySQL, Apache). Individual FOSS users are often FOSS contributors, whether in terms of code development or as simple members of FOSS communities and discussion forums. This reciprocal mode of both benefiting from and giving back to the FOSS community is also widely adopted in the non-profit, social economy sector. It should be noted that, even in the case of commercial open source, non-profit clients (e.g. NGOs, volunteer networks, and independent agencies) are usually treated under a special status regarding software application offerings, license agreements and pricing policies.

3.2. FOSS supply: enterprises, developers and communities

FOSS is produced by enterprises, FOSS non-profit foundations and communities and individual developers that are driven by heterogeneous economic, technological and social motivations. The main roles and motivations are described in the next sections.

Figure 2. A categorisation of FOSS producers and developers



FOSS developers and communities

Open source software development relies on communities of users and developers. Whether based on large communities with hundreds or thousands of members and volunteers (e.g. the Ubuntu, Debian, Fedora communities) or on a team of developers working on a single project, the development, maintenance and availability of open source software depends on the cumulative and combined work of motivated individuals and IT firms. Although often difficult to

quantify, open source communities constitute the most critical player of the open source share within the software market in the sense that they coordinate efforts, validate achieved results and mobilise resources.

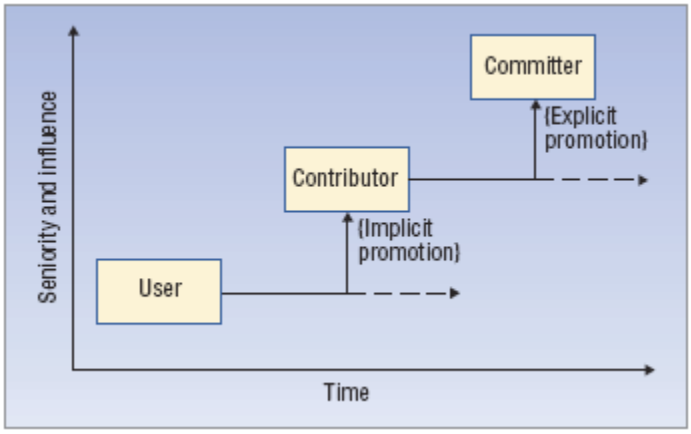
There has been a lot of discussion on the motivation of open source developers and volunteers. Getting personal gratification from contributing to the free and open source movement, maintaining a high status based on technical skills among peers or investing on future careers as a personal marketing strategy are some of the incentives attributed to FOSS developers. A thorough analysis of the motivations of FOSS developers is well beyond the scope of this report. Two important aspects however, should be highlighted:

a) A modelisation of the involvement of a FOSS devotee would pinpoint the evolution over time within the dynamics of a community of peers from a simple user to that of a "committer".[15]

b) A leadership status for an individual in a FOSS community acknowledges an increase in skills quality or a value creation for a specific open source project. This acknowledgement, usually capitalised through a business initiative or a career opportunity, seems to be a strong incentive for FOSS developers.

[15] A "committer" holds a high position on the decision making hierarchy of FOSS community project See: Riehle, D., "The Economic Motivation of Open Source Software: Stakeholder Perspectives". *IEEE Computer*, vol. 40, no. 4 (April 2007). Page 25-32.
http://dirkriehle.com/computer-science/research/2007/computer-2007.pdf

Figure 3. Contribution status in open source projects.



Source: Riehle, 2007.[16]

Although code writing is the key activity in FOSS projects, members of FOSS communities contribute in a variety of forms and tasks:

- *documentation*
- *testing and quality assurance*
- *troubleshooting and technical support*
- *graphics and artwork resources*
- *training*
- *distribution / awareness raising*

Major FOSS communities are usually sponsored and moderated by non-for-profit corporations such as the Mozilla Foundation, the Apache Software Foundation or the Gnome Foundation. Open source communities are based on their internally defined roles, mechanisms and operational frameworks. Within this context, all produced software and related resources are collectively owned by the community.

[16] Riehle, D., "The Economic Motivation of Open Source Software: Stakeholder Perspectives". *IEEE Computer*, vol. 40, no. 4 (April 2007). Page 25-32.
http://dirkriehle.com/computer-science/research/2007/computer-2007.pdf

Enterprise-driven FOSS supply and support

Some of the most successful commercial open source projects such as Ubuntu / Canonical, Fedora / Redhat and Suse / Novell are owned and maintained by large IT companies that, contrary to independent software vendors (ISVs) that sell packaged proprietary software, they provide custom software solutions (e.g. enterprise servers, applications management etc.) and certified support for advanced needs of specific clients. Although such IT companies invest a lot on the interaction with open source communities, their for-profit strategies aim to generate revenue streams for trusted and reliable open source solutions and services that they exclusively own and provide. Due to its inherent properties (offered in source code form, free to modify and distribute) FOSS provides and opportunity for IT companies wishing to switch from an unsuccessful proprietary model or aiming to enter the market at a minimal initial investment.

Moreover, several IT companies indirectly support FOSS by encouraging, permitting or tolerating employee involvement in open source code development. Enterprises invest and contribute in the development of open source software as a means to reduce their overall costs, capitalise on community generated code and resources and shape or extend new market segments in which they can operate as service and solution providers. FOSS communities and user networks also offer the ability to get instant feedback, testing results and user experience insights that would otherwise require time consuming procedures based on internal enterprise communication infrastructure and networks of registered clients.

Software companies adopt elements of or comprehensive pro-open source approaches for a variety of vital strategic reasons:

- New entrants or firms aspiring to enter a market segment aim to disrupt established software market leaders by mobilising individual open source developers.
- Software companies may compete indirectly by raising the stakes and required resources for their competitors in other market segments than the ones in which they operate.

- Firms often encourage the personal motivations of their employees in embracing open source in an effort to build team spirit with their own staff or to recruit new talent.
- Firms may promote open source as a public good in the context of a wider corporate responsibility approach. In this case they support FOSS much in the same way they work to reduce CO_2 emissions or to provide equal opportunities for minorities.
- Offering or supporting open source entry level or complementary open source products and services may constitute an effective marketing strategy to expand market share for core offerings.[17]
- Embracing open source is often a way to outsource demanding tasks and components while avoiding a strategic dependency to a single critical supplier.
- Adopting a pro-FOSS approach provides the opportunity to outsource software complexity at the hands of a world-wide community that can mobilise far more resources than most companies based on a shared and distributed investment on research and development.

Bonaccorsi and Rosi (2004)[18] have identified, in their study, various, economic, technological and social motivations of FOSS producing or contributing firms as shown below:

[17] Indeed empirical evidence shows that many companies participating in open source projects do offer complementary products and services.

[18] Bonaccorsi A. and Rossi C., "Altruistic individuals, selfish firms? The structure of motivation in Open Source software," *First Monday* 9, no. 1 (2004): 1–9.

Table 1. Firms' motivations for producing / contributing to FOSS.

Ranking	Motivation	Motivation area
1	Because Open Source allows small enterprises to afford innovation	*Economic*
2	Because contributions and feedback from the Free Software community are very useful in fixing bugs and improving software.	*Technological*
3	Because of the reliability and quality of Open Source Software	*Technological*
4	Because we want to be independent of the price and licence policies of large software companies	*Economic*
5	Because we agree with the values of the Free Software movement	*Social*
6	Because we wish to place our source code and skills at the disposal of the Free Software community and hope that others will do the same	*Social*
7	Because good IT specialists are easy to find in the field of Free Software	*Economic*
8	Because we want to study code written by other programmers and use that code in developing new programs and products	*Technological*
9	Because opening our source code allows us to gain a reputation among our customers and competitors	*Economic*
10	To get products that are not available on the proprietary software market	*Technological*
11	Because we think that software should not be a proprietary asset	*Social*

Source: Modified from Bonaccorsi and Rosi (2004)

3.3. Commercial and non-commercial mechanisms in developing and distributing FOSS

According to the aforementioned enterprise-driven and community based modes of developing FOSS, two main types of open source software occur: community open source and commercial open source. Although community-based open source projects can also be commercialised, the key differentiating factor is that of ownership and decision-making control over an open source project.[19] It should be noted that commercial open source also makes full use of the incremental software code base, feedback inputs and resources contributed by several users or FOSS communities.

[19] See: Riehle, D. "The single-vendor commercial open course business model," *Information Systems and E-Business Management* (November 23, 2010): 1-13.

Community open-source

Community open source software is developed managed and supported by communities. A community of users, volunteers and contributors determines source code integrations, software releases, fixes and updates. The community also owns and manages all related resources (e.g. documentation, troubleshooting resources, graphics and artwork) and defines distribution and dissemination strategies. Community-based open source is the main setting under which a potential vendor lock-in is prevented by retaining software and related support services open to market competition and out of the control of a single vendor or of oligopolistic software supplier groups.

Commercial open source

Commercial open source software is owned, developed and supported for-profit by an enterprise that maintains the copyright and determines software development and implementation strategies.[20] Riehle (2010) has described this type of FOSS development and distribution as the "single-vendor commercial open source" model.[21] This model of single corporations sponsoring and controlling open source projects is expected to increase its share in the following years. Gartner has recently estimated that by 2012 more than 50% of all revenue generated from open source software projects will come from projects controlled and sponsored by a single vendor.[22]

[20] Some typical examples of this model are Redhat's Enterprise Linux, Novell's Suse Linux or MySQL database.
[21] Riehle, D. "The single-vendor commercial open course business model," *Information Systems and E-Business Management* (November 23, 2010): 1-13.
[22] Gartner, Inc. Predicts 2009: The Evolving Open Source Model. Gartner, Inc. (2008)

4. What makes FOSS economically efficient? Software market strategies

4.1. Competing or converging? Open and closed source licensing regimes

Software provision and marketing is defined by licensing / pricing schemes that regulate the scale and conditions for software use, distribution and modification. Licensing regimes are usually differentiated on a restricting / granting permissions corresponding to proprietary / open source schemes.

In proprietary licensing, the type, scale and extent of software use is restricted by the software vendor. Proprietary licences define maximum numbers of software installations or users, restrict copying (e.g. only for back-up purposes) and distributing and prohibit decompiling or reverse engineering of software applications.

Open source software licensing schemes do not pose restrictions on the scale and extent of software usage. Instead, they promote and encourage software distribution, copying and modification under certain conditions, the most common of which it to release any modified software under a same licence type in order to maintain free code availability.

In practice, however, such a categorisation proves to be over-simplistic since IT companies employ far more complicated and combined methods of open source and closed source / proprietary licensing regimes, according to their defined revenue model, strategic objectives and market competition. According to Bessen (2005)[23] FOSS provides a further enabling and

[23] Bessen J., "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005).

complementary market mechanism that allows for a higher level of software complexity and efficiency to meet business objectives.

Open and closed source licensing regimes do not necessarily compete but rather converge or complement each other as IT companies are trying to meet their clients' needs in more flexible and effective ways by combining open source accessibility and added proprietary features or components.

4.2. Open source business strategies

There is a wide range of "hybrid" business strategies that combine open source and proprietary features and blend licensing regimes, delivery channels and revenue models. This hybridisation of business models has come partly as a result of the increased market penetration of open source solutions particularly in advanced, enterprise systems and applications. The term "open source business strategies" is used here to describe licensing regimes and revenue models that exceed the conventional proprietary model by making full use of FOSS properties and capabilities. Although not always compliant with official FOSS definitions, these business strategies comprehensively integrate distinctive open source features that enrich variations and blur the lines between FOSS and proprietary licensing regimes.

1. *Dual licensing*

Dual licensing is applied in cases where the same software code is released and distributed under both free software (e.g. GPL) and a proprietary licenses. This model offers users an exception from the "copyleft" requirement to release derivatives under the same licence type by providing a second, proprietary licence option. In this way, licensees (e.g. companies, developers) are able

to decide whether they want to adopt an open source or more restrictive approach on controlling software use, distribution and sub-licensing.

The benefit of dual licensing is that it can work both ways. Open source licensing allows external contributions under the same license type while proprietary licensing helps to fund and commercially promote a software product. It has been shown, however, that this model may also lead to limited external contributions by developers, due to the "same licensing regime" requirement.

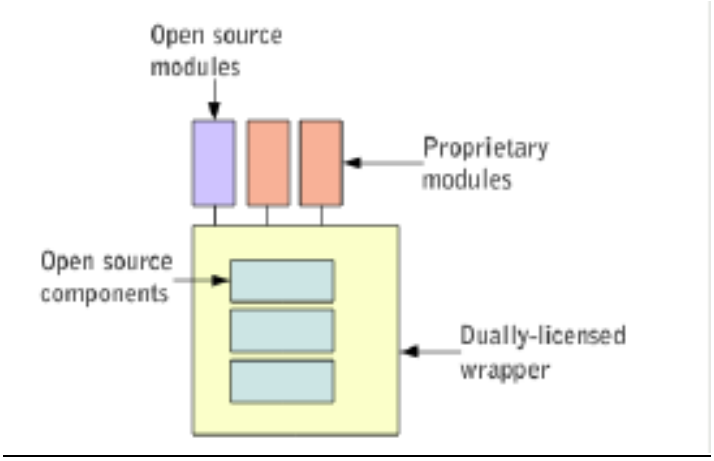
A typical example of a dual licensing business model is that of MySQL. MySQL[24] provides the option of choosing between GPL and a commercial licence. Within this scheme, those producing and distributing FOSS under a "copyleft" licence can use the GPL licensed MySQL code. On the other hand, developers or companies who wish to use the MySQL code but are not willing to release the source code of their own software products may acquire the commercial licence.

2. *Open Core*

The "Open Core" model is based on a dual offering of a basic, free software version and a proprietary version that extends the functionality of the basic version with proprietary components and plug-ins. In order for the model to be successful, however, a certain balance of attractiveness between the basic, free software version and the value-added proprietary version should be achieved (Daffara 2009). Firms applying the "open core" model usually either apply dual licensing or adopt the Mozilla Public Licence[25] that offers greater flexibility than GPL.

[24] http://www.mysql.com/
[25] http://www.mozilla.org/MPL/MPL-1.1.html

Figure 4. The "Open Core" model.



Source: Daffara 2010.[26]

The Zimbra[27] email and collaboration software provides a typical example of employing an open core business model.

3. *Certified solutions and customer support*

Major commercial open source companies employ a business strategy that despite variations, is based on offering community based software for free while selling versions of tested, certified and fully supported open source software, usually on a subscription basis. This subscription-based "product plus support" model has been employed by Red Hat, leading one of the largest and most successful commercial open source projects.[28] A similar approach has been adopted by

[26] Daffara C., "The relationship between Open Core, dual licensing and contributions," *Carlo Daffara*, July 21, 2010.
http://carlodaffara.conecta.it/the-relationship-between-open-core-dual-licensing-and-contributions/

[27] http://www.zimbra.com

[28] Red Hat offers Fedora for free through the Fedora Project, while selling Red Hat Enterprise Linux and related support services on a subscription basis. Similarly, Novell offers openSUSE for free through the openSUSE Project, while selling SUSE Linux Enterprise.

Canonical that supports the Ubuntu project while also offering technical support and training for businesses. By adopting such a strategy, FOSS providers are shifting expected revenue sources from the non-exclusive use rights of FOSS to the efficiency in certified technical support that they exclusively provide.

4. *Training and consulting services*

Several IT companies, while not developing FOSS products themselves are providing training and consulting services on open source solutions. Although not directly involved in FOSS production, such firms respond to an increased demand for expert training, strategic consulting and evaluation services on various open source systems and applications, particularly in business environments.

5. *Software as a service*

FOSS lies at the core of new software provision modes such as Software-as-a-Service (SaaS) and is expected to have an increased impact and penetration on these, still forming, market segment. The SaaS and cloud computing paradigm facilitates the deployment of open source software stacks[29] and favors web-based revenue models (e.g. online subscriptions, pay per instance or usage over time, storage limit, online advertising etc). The compelling, competitive advantage of the SaaS[30] model is its multi-tenant architecture allowing to serve multiple clients through one shared instance or application.

It should be noted, however, that software offered as a service is not necessarily tied to FOSS or to particular business models as various proprietary software offerings can also be deployed on cloud-based infrastructures applying various combinations of revenue streams. The value-added

[29] Some open source cloud infrastructures are already currently offered (e.g. Openstack, Eucalyptus Open Source).
[30] Or Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) model.

ability of serving on-demand the needs of multiple clients through a shared instance shifts significance from open source vs. proprietary licensing to reaching a critical mass of users.

4.3. The added value of FOSS: managing complexity

Complexity is a critical aspect by which software development cost and software value are defined. On an etymological approach, complexity refers to the interconnections of things or components bundles together. In terms of software development, complexity describes the effect of software components that bundled together tend to generate increased interactions that are hard to predict or to control. The un-predictability of these interactions shifts the cost of software production from coding itself to testing, debugging and maintenance of software applications. Software companies invest most efforts and resources on pre-release testing and post release fixes, updates and customer support for their software products.

FOSS, however, developed and maintained by world-wide developer communities brings the competitive advantage of mass-scale, outsourced testing, evaluation feedback and debugging at a rate that no software company could compete with. This ensures that code quality is constantly monitored and improved by an extended network of testers and contributors.

Moreover, the FOSS development model, based on constant contributions, integrates fixes, updates, and third-party add-ons, thus resulting in a growing code base with multiple features, and more complex or specialised applications that can meet the unique needs of individual users. According to Bessen[31] this unrivaled ability to constantly combine and improve new features constitutes a competitive advantage in terms of efficiency of software provision, compared to pre-packaged software.

[31] J. Bessen, "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005).

This explains, to a large extent, the fact that FOSS has an impressive market penetration in certain market segments such as web servers, [32] server operating systems and middleware (e.g. application servers, application management platforms) that require managing a higher level of complexity in terms of quality monitoring, testing, performance and security fixes.

In other words, FOSS allows a more efficient management of software complexity in terms of combining functionalities and component interactions while at the same time retaining a shared and distributed required investment.

[32] The widespread use of the Apache server, a direct competitor of Microsoft's IIS server is a clear example of this.

5. The economic assessment of FOSS: conclusions for public organisations

5.1. The economic assessment of FOSS: beyond cost analysis

In trying to assess FOSS as an economically effective solution for public organisations, it is often viewed through a strictly cost-based perspective. Such an approach either leads to the misconception that FOSS is cost-free or to cost analysis[33] studies that tend to ignore qualitative, strategic aspects and long-term benefits.

It is true that although FOSS is based on a free use and distribution licensing model, getting the software and the services associated with it can have considerable cost. All direct, indirect or hidden costs associated with integrating open source solutions in public sector IT infrastructures should be made visible and analysed in order to reach an informed decision on how to spend available resources. The Total Cost of Ownership (TCO), in all approaches and its complex estimation methodologies, if often mentioned in discussions of software procurement. TCO is estimated based on various cost categories such as: licence purchases, required hardware, required upgrades and extensions, technical support, training and maintenance fees, exit cost.

Such a cost analysis remains restricted, however, to measurable costs excluding long-term benefits that are hard to quantify. A full economic assessment should include both quantifiable costs and expected benefits or strategic advantages projected on a long-term scale. This is particularly true in the case of open source that can provide higher flexibility and vendor independence in the long-term compared to third-party dependent proprietary systems.

Projecting not just the costs but also the expected benefits for the full life-cycle of systems and applications both on a mid-term and long-term horizon in relation to the IT policies and strategic

[33] Mostly focusing on Total Cost of Ownership (TCO).

planning of public organisations, is essential in order to fully assess FOSS as an economically effective solution.

5.2. FOSS as a public good and competitive advantage

Based on non-excludable, non-rival use rights and properties, FOSS is considered a public good. The main differentiating factor between public and private goods is that public goods can be simultaneously used by several users (Kooths et al. 2003).[34] According to Bessen, [35] however, FOSS is not a simple public good but rather a complex public good, since it offers a higher level of efficiency in delivering combined features and customised functionalities to meet the specialised needs of heterogeneous users.

In this sense, FOSS is an ever-accessible and incremental resource that brings competitive advantages to public organisations, particularly relating to their strategic mission of service provision to citizens. By integrating FOSS solutions in their IT policies and infrastructures, governments, national agencies and public administrations can improve their position as strategic players in the software and IT sector. In uptaking FOSS, public organisations can be in a better position to:

- *ensure data openness and interoperability*
- *tailor software solutions to their changing organisational needs and operational requirements*
- *reduce licence purchasing costs and achieve strategic vendor independence on a long-term scale*

[34] S. Kooths, M. Langenfurth, and N. Kalwey, "Open-Source Software: An Economic Assessment" Muenster Institute for Computational Economics (MICE), University of Muenster (2003).
[35] J. Bessen, "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005): p4.

- *attract more technically and economically efficient offerings through competition in the software market.*
- *contribute to open source code quality and supply of reliable open source solutions.*
- *achieve higher efficiency and independence as producers of own, in-house software solutions.*

6. Annex

6.1. Research methodology

This report has been based on desk research of recent literature and existing evidence on the software market and the economics of free and open source software. Various documents and information sources have been used and reviewed as indicatively grouped below:

1. *EU official documents, expert group reports and guidelines.*
2. *Academic / research papers and empirical studies.*
3. *Independent reports from the software industry sector.*
4. *Software market indicators and statistical data from online resources.*
5. *Online FOSS projects, communities and repositories (www.osor.eu)*

The desk research did not apply an extensive literature review but selectively focused on key documents and critical resources that particularly relate to the European context and the public sector. Information resources, interest groups and online communities were partly identified through the knowledge base and the communication network of the OSEPA project partnership.

Previous surveys, reports and all relevant knowledge resources of the OSEPA project regarding the technological, economic or social aspects of open source software were also taken into account in order to highlight issues and priorities set by the OSEPA partnership through the exchange of experience and information.

6.2. References and resources

1. A. Aumasson et al., *Economic and Social Impact of Software & Software-Based Services. D5 – Final Report*. Pierre Audoin Consultants (PAC), August 2010. http://cordis.europa.eu/fp7/ict/ssai/docs/study-sw-report-final.pdf

2. Bessen J., "Open source software: Free provision of complex public goods," *The Economics of Open Source Software Development* (2005).

3. A. Bonaccorsi and C. Rossi, "Altruistic individuals, selfish firms? The structure of motivation in Open Source software," *First Monday* 9, no. 1 (2004): 1–9.

4. Daffara C., "The basis of OSS business models: property and efficiency", July 26, 2010, http://carlodaffara.conecta.it/category/oss-business-models/.

5. Daffara C., "The relationship between Open Core, dual licensing and contributions," *Carlo Daffara*, July 21, 2010 http://carlodaffara.conecta.it/the-relationship-between-open-core-dual-licensing-and-contributions/

6. Daffara C., "Economic Free Software perspectives," April 5, 2009, http://carlodaffara.conecta.it/category/oss-business-models/page/3/

7. Gartner, Inc. Predicts 2009: The Evolving Open Source Model. Gartner, Inc. (2008)

8. Ghosh, R. A (ed.), *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 (European Commission, November 20, 2006).

9. Giron F. et al., *Economic and Social Impact of Software & Software-Based Services. D2. The European Software Industry.* (Pierre Audoin Consultants (PAC), July 30, 2009), http://cordis.europa.eu/fp7/ict/ssai/docs/20090730-d2-eu-ssbs-industry_en.pdf

10. Kooths S., Langenfurth M., and Kalwey N., "Open-Source Software: An Economic Assessment". Muenster Institute for Computational Economics (MICE), University of Muenster (2003).

11. Krishnamurthy S., "An analysis of open source business models," *Perspectives on free and open source software* (2005): 279–296.

12. Riehle, D. "Industry Perspective - The Economic Case for Open Source Foundations," *IEE Computer*, Vol. 43, no. 1 (2010): 86.

13. Riehle, D., "The Economic Motivation of Open Source Software: Stakeholder Perspectives". *IEEE Computer*, vol. 40, no. 4 (April 2007). Page 25-32. http://dirkriehle.com/computer-science/research/2007/computer-2007.pdf

14. Riehle, D. "The single-vendor commercial open course business model," *Information Systems and E-Business Management* (November 23, 2010): 1-13.

15. Rönkkö M. et al., *Software Industry Survey 2010* (Aalto University. School of Science and Technology, November 19, 2010).

16. The European Commission, Industry expert group. *Playing to win in the new software market. Software 2.0: winning for Europe. Report of an industry expert group on a European Sotware Strategy.* June 2009. ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/ssai/European_Software_Strategy.pdf.

Web resources

1. The Free Software Foundation.
   http://www.fsf.org/

2. The Open Source Initiative.
   http://www.opensource.org/

3. The Mozilla Foundation
   http://www.mozilla.org/foundation/

4. The Apache Software Foundation
   http://www.apache.org/

5. The General Public Licence
   http://www.gnu.org/licences/gpl.html

6. The Gnome Foundation
   http://foundation.gnome.org/

7. Lesser General Public Licence
   http://www.gnu.org/licences/lgpl-2.1.html

8. The European Union Public Licence
   http://www.osor.eu/eupl/

9. The European Interoperability Framework
   http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529

10. Open Source Observatory and Repository
http://www.osor.eu/

11. Redhat Inc.
http://www.redhat.com/

12. Canonical / Ubuntu
http://www.canonical.com/about-ubuntu

13. MySQL
http://www.mysql.com/