

Valuation of open source for governments

Gabor LASZLO

Gabor Laszlo, Budapest Polytechnic, Information Society Research Centre, Budapest, E-mail: laszlo.gabor@ittk.hu

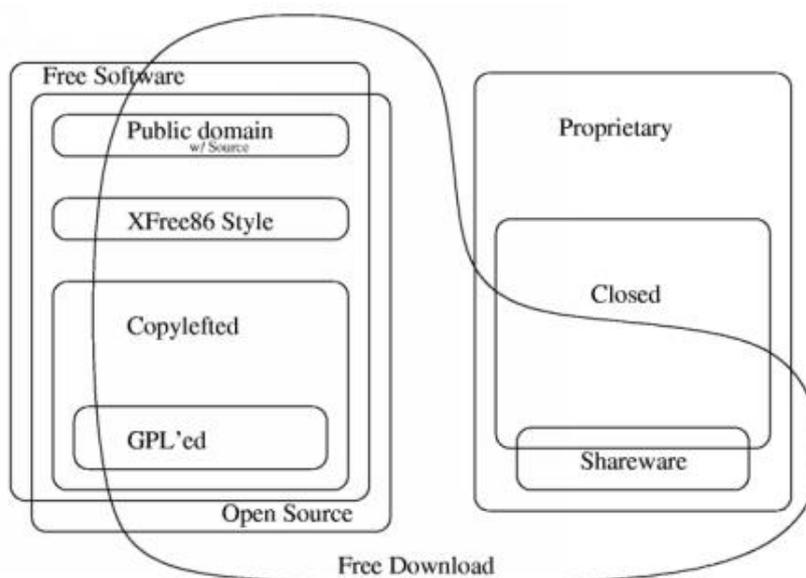
1 ABSTRACT

Keywords: open source, government, strategy consideration

Worldwide many governments recognized and occupied the power and advantage of open source software and nowadays the number of these governments is growing. Open source is a viable alternative against proprietary software products. Time is over the theological controversy and the market proved that in many fields open source software were equal or better than commercial software products. Free and open source software gives the user the freedom to use, copy, distribute, examine, change and improve the software. Free /open source software has already a long history, that is still unknown to many people. This paper shows what kind of benefits can gain the governments from using open source software and why is it so attractive for the governments.

2 THE KEY TERMS OF OPEN SOURCE AND FREE SOFTWARE

In the first instance is important to clarify some of the key terminology used in the free and open source environment. [1],[2],[3],[4] What are the differences between Free Software and Open Source Software? The Free Software movement and the Open Source movement are like two political camps within the free software community. Richard Stallman says that “The fundamental difference between the two movements is in their values, their ways of looking at the world”. The difference between them lies in the motivations that are emphasised. The term “Free Software” stresses freedom from control by another. “Free software” is a matter of liberty, not price. To understand the concept, one should think of “free” as in “free speech”, not as in “free beer”. The term “open source” software is used by some people to mean more or less the same thing as free software and it is often used by people who wish to stress aspects such as high reliability and flexibility of the resulting program as the primary motivation for developing such software.



Graph 1: Category of software products, Source: <http://www.gnu.org/philosophy/categories.htm>

Free software¹ is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbour.
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

The Open Source Definition²

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

¹ <http://www.gnu.org/philosophy/free-sw.html>

² <http://www.opensource.org/docs/definition.php>

- **Free Redistribution** The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- **Source Code** The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
- **Derived Works** The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
- **Integrity of The Author's Source Code** The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- **No Discrimination Against Persons or Groups** The license must not discriminate against any person or group of persons.
- **No Discrimination Against Fields of Endeavor** The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
- **Distribution of License** The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- **License Must Not Be Specific to a Product** The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
- **License Must Not Restrict Other Software** The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
- **License Must Be Technology-Neutral** No provision of the license may be predicated on any individual technology or style of interface.

In the rest of the document, both (FS and OSS) will be referred to as F/OSS.

2.1 Copyright and intellectual property

The copyright automatically and implicitly protects all intellectual creation, including computer software. „The issues related to intellectual property are quite important for open source software. Among the four mechanisms that international legislation provide for protecting intellectual propriety, only three (copyright, patents and trademarks) could be suitable for software. The fourth one, trade secret, is obviously not adequate for open source software, since it requires obscurity (by not having access to source code, in the case of software) or restraint on modifications or resell and redistribution of derived works. None of these requirements are possible with open source software.”[5],[6]

Copyright is the most usual method of protection for software products. In fact, open source licenses are enforceable because they use, in one form or another, copyright law. Most open source licenses were designed according to the United States law. Just recently some research has been done about the applicability of some of them in particular countries. [6]

2.2 F/OSS licenses

„The copyright law, by default, do not allow for redistribution (nor even use) of software. The only way that redistribution can be done is by granting specific permission in a license.” [5] The license is the contract between the User and the Licensor.

Open Source licenses³ are more permitted than Free Software licenses⁴. F/OSS licenses have two types, non-permissive and permissive. The Free Software licenses do not allow “closing” the source code while the permissive licenses permit the creation of proprietary development. “There are dozens of OSS/FS licenses, but nearly all OSS/FS software uses one of the four major licenses: the GNU General Public License (GPL), the GNU Lesser (or Library) General Public License (LGPL), the MIT (aka X11) license, and the BSD-new license. Indeed the Open Source Initiative refers to these four licenses as the classic open source licenses. The GPL and LGPL are termed “copylefting” licenses, that is, these licenses are designed to prevent the code from becoming proprietary.”

³ <http://www.opensource.org/licenses/index.php>

⁴ <http://www.fsf.org/licenses/license-list.html>

2.2.1 Copyleft

Copyleft is just a specific way to apply copyright. As the FSF declares, “Proprietary software developers use copyright to take away the users' freedom; we use copyright to guarantee their freedom. That's why we reverse the name, changing “copyright” into “copyleft.”” The major impact of copylefting is the prevention of changes to the license terms.

	GPL	LGPL	BSD&MIT	Apache	Public Domain	Microsoft MIT ⁴ EULA
a. Can be stored on disk with other license types	✓	✓	✓	✓	✓	(bans FOSS) ⁵
b. Can be executed in parallel with other license types	✓	✓	✓	✓	✓	(bans FOSS) ⁵
c. Can be executed on top of other license types	✓	✓	✓	✓	✓	(bans FOSS) ⁵
d. Can be executed underneath other license types	✓ ¹	✓	✓	✓	✓	(bans FOSS) ⁵
e. Source can be integrated with other license types	✓	✓	✓	✓	✓	(bans FOSS) ⁵
f. User decides if and when to publish derived code	✓ ²	✓	✓	✓	✓	✓
g. Software can be sold for a profit	✓	✓	✓	✓	✓	✓
h. Binary code can be replicated by users as desired	✓	✓	✓	✓	✓	
i. Binary code can be redistributed as desired	✓ ³	✓	✓	✓	✓	
j. Binary code can be used as desired by users	✓	✓	✓	✓	✓	
k. New users always receive source code of derived works	✓	✓ ⁶				
l. New users receive full source modification rights for derived works	✓	✓ ⁶				
m. New users receive full redistribution rights for derived works	✓	✓ ⁶				
n. Binary code can be released without source code			✓	✓	✓	✓
o. Derived code can have a different type of license		7			✓	
p. Original source can be incorporated into closed source products					✓	
¹ Provided that both programs are fully and independently usable in other unrelated contexts.						
² Provided that the binary code has not been previously released to the public.						
³ Provided that source code is always redistributed along with the binary code.						
⁴ The proprietary Microsoft MIT EULA is not related to the similarly named MIT (X/MIT) license.						
⁵ Specifically bans use of: GPL, LGPL, Artistic, Perl, Mozilla, Netscape, Sun Community, and Sun Industry Standards.						
⁶ The rights granted by LGPL do not necessarily extend to the applications linked into an LGPL library.						
⁷ The LGPL does permit re-licensing under GPL as a special case, but not re-licensing under any other license type.						
License Acronyms:						
GPL –GNU General Public License (Microsoft) MIT –Mobile Internet Toolkit LGPL –GNU Lesser General Public License (X/MIT) MIT –Massachusetts Institute of Technology			BSD –Berkeley Software Distribution EULA –End-User License Agreement MPL –Mozilla Public License FOSS –Free and Open-Source Software			

Table 1: A Comparison of F/OSS and Related Licenses, Source: MITRE <http://www.egovos.org/dodfoss.pdf>

3. ADVANTAGES OF F/OSS SOFTWARE

It is as recent as the last two years that the public sector in Europe has become interested in free and open source software. The European Commission has released several reports and open source software has been the subject of interest in many different ways. In Europe, Germany is the country that has the highest rate of open source software usage. Many governments have encountered budgetary tightening and principally in the developing countries they can save money. But in the developed world the hardware have been changed more often and this change usually accompanies with software upgrade. This cost can be decreased by using open source software in every case. The licensing questions are in close connection with the financial consideration. (Usually) there is no fee for every single license; this makes easy the inventory of software registration and of course it is a much cheaper way for software procurement.

3.1. Economical impacts

The existence of F/OSS software has a complex and important impact on the economy. Recent years the information technology has become part of our daily life. Daily activities are unthinkable without services of information technology. Many-many new hardware have been developed, but the hardware could be operational by software. Software costs can result high amount of bills. These are some reasons for governments to support the dissemination of open source software. Software costs can be cut, instead of government jobs. Replacing some proprietary software products with F/OSS software would eliminate the licensing costs of those products. Many good examples exist around the world, where the primary aspect were the cost cut.

New products, new services can develop on open source basis. Another important impact is the job creation while moving government systems to open source software model that means there will be more local, high-paying IT jobs for integrators and

consultants, that has a spin-off economic multiplier effect. Spread of F/OSS software would make stronger the local IT sector, the number of small and medium size enterprises could growing. This resulting higher rate of employment.

In specific fields, such as education using F/OSS software and open source development model work out some problem, such as the cost of software and interoperability, reduce the total cost of ownership.

The reality of the macro-economic effects is well-documented, their quantification, and their comparison is not possible today, because of the lack of adequate detailed statistical data on free and open source usage and its detailed links with economical and non-economical activities.

3.1. Total Cost of Ownership (TCO)

Total cost of ownership (TCO) is an important measure. TCO is extremely sensitive to the set of assumptions. It contains “hidden” costs, such as administration costs, upgrade costs, technical support, end-user operation costs etc. F/OSS software has many cost advantages in various categories that result the smallest TCO in many cases.

3.1.1. Acquire

F/OSS is not cost-free, but costs far less to acquire than proprietary software. F/OSS programs has initially less cost at the purchase or usually it can be download from the Internet withot any fee. The most users in some cases prefer the printed manual, support and they are ready to pay a small fee to a distributor for an integrated package with CD-ROMs, documentation, and support. There are cost for documentation, support, training, system administration etc just as in case of proprietary systems.

3.1.2. Upgrade/maintenance costs

Long-term upgrade costs are far less for F/OSS systems. F/OSS systems can be (free) downloaded, or simply re-purchased and the single upgrade be used on every system. This does not include technical support, but the technical support can be also purchased and it can be competed because there are more than one supplier (not in case in proprietary software), and you can switch among them. While upgrading a Microsoft system will typically cost around half the original purchase. This factor was that made the Munich public administration move towards open source software.

3.1.3. License management costs

In case of using F/OSS software there is no license management cost, in contrast proprietary software vendors make money from the sale of licenses and the charge the license management to customers. This means later the customers have to prove that it was paid for every installed software. Who cannot later prove that it was paid for every installed copy of proprietary software risk stiff penalties. Organizations must keep careful track of license purchases that prevent any legal actions by proprietary software vendors (or the organisation that represents their members interest, such as Business Software Alliance). It demands huge efforts from customers both in time and in cost.

3.1.4. Other factors

OSS/FS can often use older hardware more efficiently than proprietary systems, yielding smaller hardware costs and sometimes eliminating the need for new hardware.

3.2. Security

Computing is crucial to the infrastructure of advanced countries. The F/OSS software are efficient and secure, however quantitatively measuring security is very difficult. The F/OSS software advantages are based on software standard (e.g. W3C, ISO) more than proprietary software products so they meet demand to build real interoperability systems. The European Commission working paper [7] advises the open source software for governments to build their own e-government services.

Examining the government information system the most important and relevant question is the security. In case of open source development the comprehensive audit is assured by the members of open source community and the patch for security vulnerability is usually improved faster. The user can be sure that this software has not got any backdoor and other malicious and unwanted codes, from where unauthorized and unwanted persons can gain sensitive data. There are good examples for this development and trend in USA the National Security Agency [8] (that developed the Security Enhanced Linux) and Department of Defense (that advocates the open source programs in security environment). [9]

Another issue that have to discuss about is computer viruses. “Most of the world’s computers run Microsoft’s operating systems, thus most of the world’s computers are vulnerable to the same viruses and worms at the same time.” [10] The viruses are more and more dangerous and illegal attacks’ number is also growing. Viruses can disable large organisation and also governmental offices. Last year was a good example when could be seen some viruses during hours spreaded around the world. The viruses make major impact for the economy. Data loss, interrupted economical processes, etc. These incidents could be usually preventable even using proprietary software products, but it requires consideration by users. F/OSS software have less vulnerability and the viruses have not impact on them, yet.

3.3. Dependency

In case of using proprietary software the user usually can not choose a supplier or software vendor. Best example is the operating system market where Microsoft increasingly threatens to dominate market share. The strongest potential rival to its dominance is no longer its traditional commercial rivals, but the F/OSS software vendors. In case F/OSS the users have not only the mentioned

freedom as in Free Software, but they have the freedom of the choice of software vendors. That is why more and more governments choose the F/OSS software, beyond the security consideration.

The European Commission IDA programme made and published the “Open Source Migration Guidelines” [11] that helps the public administration migration to F/OSS software.

3.4. Other factors

3.4.1. Healthcare

Healthcare is one of the most costly segment within the governmental “services” around the world. The F/OSS software is occupied comprehensive spectrum in the healthcare not even more the cost consideration. During the therapy is important the protection, controll and utiliozation of patient’s data because their life can depend on it. The F/OSS softwares ensure the patients’ privacy.

The F/OSS software can improve the performance of the healthcare services, while they ensure the interoperability and patients’ privacy.

3.4.2. Education

In educational system there are two major expenses on software: in case of proprietary software the school has to buy every single computer a license that want to use the software, on the other hand the school has to ensure the possibility to use these software the students after the class. At present in most cases (no matter what side of the Earth) the informatics education cover up lower grade knowledge of the proprietary software products. Because of that the students leaving school do not have knowledge on standards and they do not know more software product. This also serves using the pirated programs, because the students usually do not get a legal copy their own computer. In education sector is a most significant development that can be seen.⁵

3.4.3. Research and Development

Nowadays the innovation is more important than earlier. From R&D must not missing the software. Every R&D project is supported by software and more R&D software are based on open source basis. This can support best way of the projects because this software can customize, usually without any legal violation. F/OSS support the share of the scientific results and disseminations.

4. CONCLUSION

No software is perfect. Both proprietary and F/OSS software made by people. The most relevant differences are between them, that F/OSS is developing and assured by group of users and the source code is available for everybody. This makes this software so desirable for governments and government agencies. The source code is verifiable and modifiable so it is secure. Recent years the governments tasks expand to build Information Society and e-government services. The secure and trustworthy services must be based on reliable software systems. Since the proprietary software can not be verified according to unavailable closed source code by security, these conditions can be accomplished only by F/OSS software. In the public sector many software handle and forward the citizens’ personal data because in this case 100% security is needed to prevent citizens’ privacy.

Governments have special obligations to protect the integrity, confidentiality and accessibility of public information throughout time like no other entity in society. Therefore, storing and retrieving government data through secret and proprietary data formats tied to a single provider is especially problematic. The trends show that many governments choose the F/OSS software different strategy consideration. F/OSS software can be considered both a great opportunity and an important resource. F/OSS software has already started to modify the rules in the information technology industry, which will produce enormous changes in further years. Europe is in a good position to take early advantage of open source, and can also help the open source movement to get stronger, that explain many government moving towards of open source last year.

Michel Sapin, the Minister of Public Services in France, stated, “Next generation e-Government has two requirements: interoperability and transparency. These are the two strengths of open source software.”

REFERENCES

- Free Software Foundation <http://www.fsf.org>
 Opensource Initiatives <http://www.opensource.org>
 European Commission Information Society Free Software portal http://europa.eu.int/information_society/activities/opensource/
 David A. Wheeler: Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers! http://www.dwheeler.com/oss_fs_why.html
 [Downloaded: Dec. 15. 2003]
 Free Software / Open Source: Information Society Opportunities for Europe? <http://eu.conecta.it/> [Downloaded: April 5. 2003]
 Pooling Open Source Software <http://europa.eu.int/ISPO/ida/export/files/en/1115.pdf> [Downloaded: April 6. 2003]
 Linking-up Europe: the importance of interoperability for e-government services <http://europa.eu.int/ISPO/ida/export/files/en/1523.pdf>
 Defense, NSA move on 'open source' software development
<http://www.govexec.com/news/index.cfm?mode=report&articleid=25151&printerfriendlyVers=1&> [Downloaded: March 18. 2003]
 Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense <http://www.egovos.org/pdf/dodfoss.pdf> [Downloaded: April 5. 2003]
 CyberInsecurity: The Cost of Monopoly <http://www.ccianet.org/papers/cyberinsecurity.pdf> [Downloaded: Dec. 15. 2003]
 IDA Open Source Migration Guidelines <http://europa.eu.int/ISPO/ida/export/files/en/1618.pdf> [Downloaded: Dec. 15. 2003]
 FLOSS <http://www.informics.nl/FLOSS/> [Downloaded: June 10. 2003]

⁵ <http://schooltool.sourceforge.net/>