VALUATION OF OPEN SOURCE SOFTWARE: HOW DO YOU PUT A VALUE ON FREE?

AVALIAÇÃO DE SOFTWARE ABERTO: COMO VOCÊ COLOCA UM VALOR SOBRE ELE?

Jesús García-García
Professor Doutor do Departamento de Contabilidade da Universidade de Oviedo
Endereço: Avenida del Cristo, s/n – Facultad de Economía y Empresa
33071 Oviedo – España
E-mail: jesgar@uniovi.es

María Isabel Alonso de Magdaleno
Professora Titular do Departamento de Administração de Empresas da Universidade de Oviedo
Endereço: Avenida del Cristo, s/n – Facultad de Economía y Empresa
33071 Oviedo – España
E-mail: ialonso@uniovi.es

ABSTRACT

The aim of this study is to assess the financial reporting framework applied to open source software. Open sourcing software developments means sharing technology and resources with communities worldwide to help eliminate the digital divide, create economic opportunity, and support equal access to technology. Therefore, a methodological approach is needed to assess properly the performance and the value generation potential and to put such measure into organizational reports. International financial reporting framework is checked over conditions to allow value recognition of open sourced assets. Linux kernel development value is estimated to reflect worth of open source developments despite absence of book value due to inexistence of a single cost source. Several attempts to estimate a valuation of open source software have been performed previously. However, this study is the first to judge suitability of accounting framework to report on this value. The main finding is that open sourced assets do not fully accomplish conditions to be included in financial reports. We seek to stimulate academic and professional debate about the pursuit of valuation of a large and efficient ecosystem of software innovation, freely available to society.
Keywords: Financial reporting. Accounting standards.Open source software.

RESUMO

O objetivo deste estudo é avaliar a estrutura de relatório financeiro aplicada ao software de código aberto. O desenvolvimento de software aberto significa compartilhar tecnologia e recursos com as comunidades em todo o mundo para ajudar a eliminar a exclusão digital, criar oportunidades econômicas, e apoiar a igualdade de acesso à tecnologia. Portanto, é necessária uma abordagem metodológica para avaliar corretamente o desempenho e o potencial de geração de valor e colocar essa medida em relatórios organizacionais. Relatos internacionais financeiros abordam as condições para permitir o reconhecimento de valor de ativos de origem de softwares abertos. No Linux o valor do desenvolvimento do kernel é estimado para refletir no valor de desenvolvimento de código aberto, apesar da ausência de valor contábil devido à inexistência de uma fonte de custo único. Várias tentativas de estimar uma

Recebido em 26.11.2012. Revisado por pares em 16.01.2013. Reformulado em 12.02.2013.
Recomendado para publicação em 18.03.2013. Publicado em 12.04.2013.

valorização de software de código aberto foram realizadas anteriormente. No entanto, este estudo é o primeiro a julgar a adequação do quadro contábil para informar sobre este valor. A principal constatação é que códigos abertos não cumprem plenamente as condições para que sejam incluídos nos relatórios financeiros. Procuramos estimular o debate acadêmico e profissional em busca da valorização de um amplo e eficiente ecossistema de inovação de software, disponível gratuitamente a sociedade.
Palavras-chave: Relato Financeiro. Normas de Contabilidade. Software de código aberto.

1 INTRODUCTION

Under the standard framework of accounting standards for financial reporting used in any given jurisdiction, general volunteer activity is not reflected on financial statements. As a result, there is not value of volunteer contributions and there is also no single source for cost estimates of how much it has taken to develop an open source software (OSS). This volunteer activity encloses not only individuals but also corporations contributing software into the open source movement.

There is a large body of literature in economics, finance and accounting concerned with the valuation relevance of intangible investments on financial reporting. Among the studies on the valuation of expenditures on intangibles are those that examine R&D costs and provide evidence consistent with the notion that investment in intangibles enhances the value of the firm: Chan, Martin and Kessinger (1990), Sougiannis (1994), Lev and Sougiannis (1996), Lev and Zarowin (1999), Shi (2003) and the literature review made by Anagnostopoulou (2008). Furthermore, when related to accounting and financial reporting framework Aboody and Lev (1998), Mohd (2005) and Givoly and Shi (2008) show that capitalization of technology development costs provides relevant information to investors and reduces the information asymmetry between insiders and outsiders.

There are many and varied reasons why a business might choose to value its open sourced knowledge, providing as reliable and accurate valuation as possible, even if they have no limit to distribution and reuse for anybody. Some of the most commonly basis could be:

- To raise finance, using it as a way of attracting investment.
- To put a realistic value on a business to be purchased or sold.
- For taxation purposes, when transferring rights or fully exploiting tax reliefs.
- For ensuring an appropriate licensee fee when granting a special leave to or from a third party.
- To enable effective decision-making when managing assets and planning business operation and strategy.
- To report accurately on the value of the business to investors and other interested third parties by returning accounts that provide a true and fair view of the business.

Knowledge is often one of the most valuable assets of a business and it is often the key objective in mergers and acquisitions, even when it have been open sourced, e.g. Oracle-Sun acquisition. One of the main reasons for the growing disconnection between market values and financial information is the legal framework for financial reporting. Great value of OSS is missing from financial statements and management decisions because accounting framework ignore activities carried out by users and outside of formal productive contexts. OSS whole ecosystem of crowdsourced innovation shouldn´t be left aside from companies' valuation.

2. OPEN SOURCE SOFTWARE

Wealth, growth and welfare are driven nowadays by intangible intellectual assets. The structure of companies has changed dramatically, shifting the focus of value creation from tangible based activities to intangible based value creation. On this way of transformation, some kind of intellectual assets as software are fast becoming commodities being freely available to society through open source methodologies.

Open source describes practices in production and development that promote access to the end product's source materials. The principles and practices are commonly applied to the peer production development of source code for software that is made available for public collaboration. The result of this peer-based collaboration is usually released as OSS. Open source is an approach to the design, development, and distribution of software, offering practical accessibility to software’s source code. Some consider open source as one of various possible design approaches, while others consider it a critical strategic element of their operations. Before open source became widely adopted, developers and producers used a variety of phrases to describe the concept, mostly used term was "free software"; the term "open source" gained popularity with the rise of the Internet, which provided access to diverse production models, communication paths, and interactive communities. The open source model of operation and decision-making allows concurrent input of different agendas, approaches and priorities, and differs from the more closed, centralized models of development (RAYMOND, 1999).

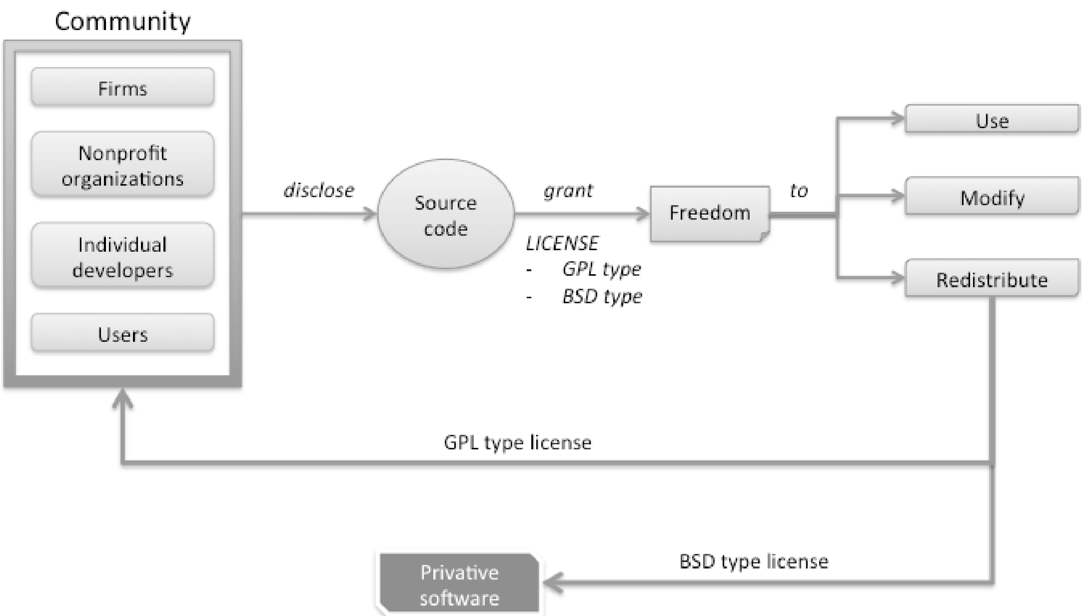

Figure 1. OSS concept map

OSS is software that can be used, studied, and modified without restriction, and which can be copied and redistributed in modified or unmodified form either without restriction, or with minimal restrictions only to ensure that further recipients can also do these things and that manufacturers of consumer-facing hardware allow user modifications to their hardware. In practice, for software to be distributed as free software, the human-readable form of the program (the source code) must be made available to the recipient along with a legal license granting the above permissions. OSS could be under GPL-type license, which means that derived works can only be distributed under the same license terms, or under the more permissive BSD-type license, which places minimal restrictions on how the software can be redistributed even just allowing privative redistribution. We believe it doesn´t make a difference for valuation purposes whether OSS license is GPL-type or BSD-type.

Nearly all OSS is free software. The two terms describe almost the same category of software, but they stand for views based on fundamentally different values. Open source is a development methodology; free software is a social movement (STALLMAN, 2007). For the free software movement, it is an ethical imperative, because only free software respects the users' freedom. By contrast, the philosophy of open source considers issues in terms of how to make software better in a practical sense only. It says that non-free software is an inferior solution to the practical problem at hand. For the free software movement, however, non-free software is a social problem. Increasingly, the consensus term "Free Libre Open Source Software" (FLOSS) is used to describe the common ground between free software and open source. It emphasizes the loose component of the free software with the Spanish term "libre", avoiding confusion with the no-payment meaning of free.

The open source movement has been the inspiration for increased transparency and freedom in other domains. It has also been applied to media other than computer programs, e.g. cultural and entertainment industry (free culture) or political organization (open government). It also constitutes an example of shared innovation. Often, open source is an expression where it simply means that a system is available to all who wish to work on it. Indeed, open source is a frictionless technology transfer agreement.

3. THE CONCEPT OF INTANGIBLE ASSET AND ITS VALUATION ON ACCOUNTING FRAMEWORK

3.1. CONCEPTUAL FRAMEWORK FOR FINANCIAL REPORTING

Every country has is own accounting legal framework, but all of them relies on standards fixed by private not-for profit organizations. International Accounting Standards Board (IASB) and Financial Accounting Standards Board (FASB) are main independent, privately funded accounting standard-setters. In order to establish accounting principles, IASB and FASB issue pronouncements publicly, each addressing general or specific accounting issues.

The IASB was founded on 2001 as the successor to the International Accounting Standards Committee (IASC) founded on 1973. It is responsible for developing International Financial Reporting Standards (IFRS), called International Accounting Standards (IAS) before 2001, and promoting the use and application of these standards. IFRS are considered a "principles based" set of standards in that they establish broad guidelines as well as dictating specific procedures. IASB is a European organization based on London so IFRS are used in many parts of the world but mainly on European Union where IFRS reporting is required or inspires local accounting framework. In addition, increasing worldwide acceptance of financial reporting using IFRS is leading USA towards IFRS. IASB work is directed also at convergence with USA financial reporting framework.

The FASB is an American private organization, founded on 1973 and settled on Norwalk (Connecticut), whose primary purpose is to develop generally accepted accounting principles (GAAP) within the USA in the public's interest. GAAP are accounting rules used to prepare, present, and report financial statements for a wide variety of entities, including publicly-traded and privately-held companies, non-profit organizations and governments. USA government does not directly set accounting standards, in the belief that the private sector has better knowledge and resources. GAAP are not written in law although the Securities and Exchange Commission (SEC) requires that it be followed in financial reporting by publicly-traded companies.

The most important difference between IFRS and GAAP is that the IFRS are based on principles, whereas GAAP are based on rules. GAAP suffers from the complexity of trying to set rules for all situations, a complexity that often masks economic reality. According to IASB, in the absence of a standard or an interpretation that specifically applies to a transaction, management must use its judgement in developing and applying an accounting policy based on accepted principles that results in information that is relevant and reliable.

3.2. RECOGNITION OF ASSETS IN ACCOUNTING FRAMEWORK

Assets are the fundamental concept in accounting. Assets, also called economic resources, are the lifeblood of both business enterprises and not-for-profit organizations. Since resources or assets confer their benefits on an enterprise by being exchanged, used, or otherwise invested, changes in resources or assets are the purpose, the means, and the result of an enterprise’s operations, and a business enterprise exists primarily to acquire, use, produce, and distribute resources (FASB, 1985).

On IFRS an asset is defined as a resource controlled by the enterprise as a result of past events and from which future economic benefits are expected to flow to the enterprise. An item that meets the definition should be recognised if (IASB, 2001):

1. It is probable that any future economic benefit associated with the item will flow to or from the entity; and
2. The item has a cost or value that can be measured with reliability. Measurement, as defined by IASB, is the process of determining the monetary amounts at which the elements of the financial statements are to be recognised and carried in the balance sheet and income statement. This involves the selection of the particular basis of measurement.

According to FASB (1985) an asset has three essential characteristics to come into existence:

1. It embodies a probable future benefit that involves a capacity, singly or in combination with other assets, to contribute directly or indirectly to future net cash inflows.
2. A particular entity can obtain the benefit and control others’ access to it.
3. The transaction or other event-giving rise to the entity’s right to or control of the benefit has already occurred.

Both definitions include future economic benefit as the main feature of an asset. The most obvious evidence of future economic benefit is a market price. Anything that is commonly bought and sold has future economic benefit. Similarly, anything that creditors or others commonly accept in settlement of liabilities has future economic benefit, and anything that is commonly used to produce goods or services, whether tangible or intangible and whether or not it has a market price or is otherwise exchangeable, also has future economic benefit. Incurrence of costs may be also significant evidence of acquisition or enhancement of future economic benefits. It must be clear that we must firmly reject the popular argument that costs are assets. Although an entity normally incurs costs to acquire or use assets, costs incurred are not themselves assets. The essence of an asset is its future economic benefit rather than whether or not it was acquired at a cost. The ultimate evidence of the existence of assets is the future economic benefit, not the costs incurred (FASB, 1985). The practical problems are in determining whether future economic benefit is actually present and in quantifying it, especially if realization of benefits is far away. In practice, most companies write off as an expense of the present period costs made with the expectation of benefiting future periods. This policy mismatches revenue/expense relationship and cannot be justified on the grounds of accounting principles to show the true and fair view of the company. Furthermore, precluding capitalization removes from financial statements what may be a company's most valuable asset (GORNIK-TOMASZEWSKI; MILLAN, 2005).

IASB definition of an asset is wider than FASB's one. FASB requires not only probable future benefit but also exclusive control of resources and the existence of a transaction in the past. The absence on IASB's definition of a transaction in the past can be justified on the ground that it was superfluous; anything that exists must have come into existence at some time in the past. But it also can be interpreted as a possibility to report real world economic phenomena that have no origin in past transactions, thus allowing the recognition, at fair value, of elements of internally generated goodwill that until this time have not been regarded as suitable for recognition in financial reports.

An entity must control an item's future economic benefit to be able to consider the item as its asset. The classical view of control over assets is based on scarcity. To enjoy an asset's benefits, an entity generally must be in a position to deny or regulate access to that benefit by others. Hence, an asset of an entity is the future economic benefit that the entity can control and as a consequence can, within limits set by the nature of the benefit or the entity's right to it, use as it pleases. The entity having an asset is the one that can exchange it, use it to produce goods or services, exact a price for others' use of it, use it to settle liabilities, hold it, or perhaps distribute it to owners. An entity usually gains the ability to control an asset's future economic benefits through a legal right. However, an entity still may have an asset without having an enforceable legal right to it if it can obtain and control the benefit some other way, for example, by maintaining exclusive access to the asset's benefits by keeping secret a formula or process.

3.3. ATTRIBUTES OF INTANGIBLE ASSETS

Intangible assets can be defined as identifiable non-monetary assets that cannot be seen, touched or physically measured, which are created through time and/or effort and that are identifiable as a separate asset. There are two basic forms of intangibles:

- Legal intangibles: such as trade secrets, copyrights, patents and trademarks.
- Competitive intangibles: such as know-how, knowledge, collaboration activities and goodwill.

Legal intangibles are frequently called intellectual property and generate legal property rights. Competitive intangibles, while legally non-ownable, have an impact on effectiveness, productivity, wastage, and opportunity costs within an organization. Therefore, they have also impact on costs, revenues, market value, and share price. Competitive intangibles are the biggest source of competitive advantage for organizations.

The three critical attributes of an intangible asset are, according to IAS 38 (IASB, 1998):

- Identifiability
- Control (power to obtain benefits from the asset)
- Future economic benefits (such as revenues or reduced future costs)

These attributes are similar to the generic attributes of an asset except for the requirement of identifiability as the main feature of an intangible. Requirement is obvious due to lack of physical entity. IAS 38 states an intangible asset as identifiable when:

- It is separable (capable of being separated and sold, transferred, licensed, rented, or exchanged, either individually or together with a related contract) or
- It arises from contractual or other legal rights, regardless of whether those rights are transferable or separable from the entity or from other rights and obligations.

These requirements apply whether an intangible asset is acquired externally or generated internally. The probability of future economic benefits must be based on reasonable and supportable assumptions about conditions that will exist over the life of the asset and is always considered to be satisfied for intangible assets that are acquired separately or in a business combination but not for internally generated intangible assets. If an item does not meet both the definition of and the criteria for recognition as an intangible asset, it must be expensed when it is incurred without possibility to be reinstating as an intangible asset at a later date.

R&D and software development costs are capitalised only after technical and commercial feasibility of the asset for sale or use have been established. This means that the entity must intend and be able to complete the intangible asset and either use it or sell it and be able to demonstrate how the asset will generate future economic benefits. Resources to complete the project and ability to measure cost are also required.

Open source generated assets face the problem of the control over them. According to open licenses there is one organization that keeps control over the asset. But it doesn't hold any control about its uses and economic exploitation. However, any organization that freely receives the asset can use it as a competitive intangible to generate income. Furthermore, an OSS is developed across a continuous process. There is never a finished product, as with privative software. Accordingly, there is not a clear time to finish capitalization of development costs. However, this fact should not prevent to know real monetary value of an OSS.

4. AN ESTIMATION OF THE VALUE OF THE LINUX KERNEL

The most prominent example of OSS is the Linux operating system kernel. It has been developed using an open source development model. As OSS, Linux is developed collaboratively, meaning no one company is only responsible for its development or support. Companies share research and development costs with their partners and competitors.

Since 2005 over 7800 individual developers from nearly 800 different companies have contributed to the kernel (Linux Foundation, 2012). There is not booking value for the Linux kernel as a consequence of not being developed by an only firm, but development costs should be estimated. Obviously, putting a traditional cost model on the Linux kernel doesn't quite make sense but it helps to illustrate the enormous value in an OSS project. We must note that the Linux kernel represented a $21 billion ecosystem in 2007, expected to more than double in revenue by the end of 2012 (IDC, 2009).

4.1. METHODOLOGY

Parametric models are amongst the most suitable to estimate OSS costs, due to the lack of an only and whole data source. Resource estimates are developed using prediction models, which mathematically relate effort and duration to the parameters that influence them. These models are built up using regression analysis on available data and are highly dependant on good understanding of what the parameters used in the model mean, their flawless measurement and periodic recalibration.

Contructive Cost Model (COCOMO81), developed by Boehm (1981), is one of the most popular and most elaborated parametric effort models for software. Although it provides a rough estimate of the effort needed to develop software of a given size, it had been applied by several authors estimating OSS distributions such as Red Hat (WHEELER, 2001), Debian (GONZALEZ-BARAHONA *ET AL.*, 2001; AMOR-IGLESIAS *ET AL.*, 2005) and Fedora (LINUX FOUNDATION, 2008). Results it gives when applied to Linux Kernel should be viewed with caution because COCOMO81 was designed for classical and large software projects.

We employ the sum of code lines added, modified or deleted as measure of effort and output. Physical source lines of code (SLOC) measure, disclosed by Linux Foundation

(2010), is used as the primary measure of output in the estimation. It is defined as a line ending in a newline or end-of-file marker, and which contains at least one non-whitespace non-comment character. Park (1992) recommends the use of physical SLOC measure because is much easier to implement than logical SLOC, which must be redefined for every programming language being measured, making inter-language comparisons more difficult (e.g. count of all terminating semicolons in a C file). Using physical SLOC also implied that COCOMO81 have to be used; more advanced versions of the model, such as COCOMO II, require logical SLOC instead of physical SLOC.

COCOMO81 consists of three submodels: Basic, Intermediate and Advanced. It can be written as:

$$E = KS^{\alpha}\prod_{i=1}^{15} C_i \qquad (1)$$

Where *K* and α are parameters dependant on how the software system was developed, *S* is the measure of the output and C_i are fifteen cost drivers (Boehm, 1981; Hu et al. 1998).

Parametric cost model “Intermediate COCOMO81” is used to get better accuracy on our estimation of effort, considering the Linux Kernel to be a “semidetached” application; and according to the intermediate model and previous literature (Wheeler, 2004) parameters are estimated as K=3, α=1.12 and $\prod C_i$=1.55; size of the project (*S*) is measured using physical SLOC as output. Effort results in total person-months.

Then, development time in months can be obtained from:

$$T = cE^{d} \qquad (2)$$

Where *c* and *d* are parameters given in Intermediate COCOMO81 as 2.5 and 0.35.

The number of developers required to code the software associated to an effort E and in time T is calculated as:

$$P = \frac{E}{T} \qquad (3)$$

Data was taken from Linux Foundation (2010). We use kernel development history from version 2.6.11 to 2.6.35 (released from 02.03.2005 to 01.08.2010), with main variables: SLOC (total, added, modified and removed) days of development, number of developers and employers. To calculate the cost for the Linux kernel, a base salary was estimated from EUROSTAT Labour Market Database, taking data from Structure of Earnings Survey 2006 by Educational Attainment tables (the most recent year that data are available). According to EUROSTAT, the average annual salary for a developer in 2006 was 31,040€, measured as the mean of annual earnings in high-tech manufacturing and knowledge-intensive high-technology services for upper secondary and tertiary education. Most OSS development is global, so making use of a EU-average salary number inserts some bias into the model. An overhead factor value is necessary to estimate the costs of office space, equipment, overhead staff, and so on. We use 2.4 as an estimate, which is used on previously cited literature applying COCOMO models to estimate OSS value.

4.2. RESULTS

Estimations of effort and development costs for Linux kernel are reported in Table 1. Days between versions, number of developers and number of employers are real data obtained from Linux Foundation (2010). An estimated full cost of development from very beginning is calculated for each kernel version, according to the Intermediate COCOMO81 model described by the equation (1) and SLOC of each version (Figure 2). These figures represent the cost that would be incurred if at the present moment any entity tried to initiate the development of this OSS. We use equation (1) to obtain the total estimated development effort in person-months. Then, time of development measured in months and number of developers needed is calculated based on equations (2) and (3). Incremental cost from previous version has been calculated as the difference between full cost of last and previous versions. They should be considered as R&D or intangible asset value created by volunteers. Results are validated against linear regression of developers observed on developers according to model (R^2=81,65%; figure 3).

Linux kernel version 2.6.35 released on August 2010 contained near to 13.5 million physical SLOC, compared to 6.6 million SLOC in version 2.6.11 released on March 2005. Using Intermediate COCOMO81 model, had Linux kernel version 2.6.35 been developed by traditional privative means, it would have cost over 1,200 million EUR to develop in European Union and it is estimated to have required about 1,100 developers working full-time during 15 years. Compare this to estimated development costs for version 2.6.11 of almost 550 million EUR, it results a differential cost of 700 million EUR over five years. However, this figure could be even higher if we took into account typical features of crowdsourced OSS production as the amount of effort associated to modified and deleted SLOC by community of developers.

Linux kernel versión	Date Released	Days between versions	Number of developers	Number of employers	Incremental cost from previous versión (mil. EUR)	Full cost of development (mil. EUR)
2.6.11	02/03/05	69	389	68	549623	549623
2.6.12	17/06/05	108	566	90	14310	563934
2.6.13	28/08/05	73	545	94	19693	583627
2.6.14	27/10/05	61	553	90	14463	598090
2.6.15	02/01/06	68	612	108	13786	611877
2.6.16	19/03/06	77	709	111	17887	629764
2.6.17	17/06/06	91	726	120	10188	639953
2.6.18	19/09/06	95	815	133	15589	655542
2.6.19	29/11/06	72	801	128	21190	676733
2.6.20	04/02/07	68	673	138	12014	688747
2.6.21	25/04/07	81	767	143	13722	702469
2.6.22	08/07/07	75	870	180	24171	726641
2.6.23	09/10/07	94	912	181	6437	733078
2.6.24	24/01/08	108	1057	193	28146	761224
2.6.25	16/04/08	83	1123	232	35974	797199
2.6.26	13/07/08	88	1027	203	17354	814554
2.6.27	09/10/08	88	1021	187	21182	835737
2.6.28	24/12/08	76	1075	212	47641	883379
2.6.29	23/03/09	89	1180	233	80143	963522
2.6.30	09/06/09	78	1150	249	62030	1025553
2.6.31	09/09/09	92	1166	227	40736	1066289
2.6.32	02/12/09	84	1248	261	56281	1122570
2.6.33	24/02/10	84	1196	238	38191	1160761
2.6.34	15/05/10	80	1150	243	33365	1194127
2.6.35	01/08/10	78	1187	209	22711	1216839

Table 1. Information about Linux kernel development

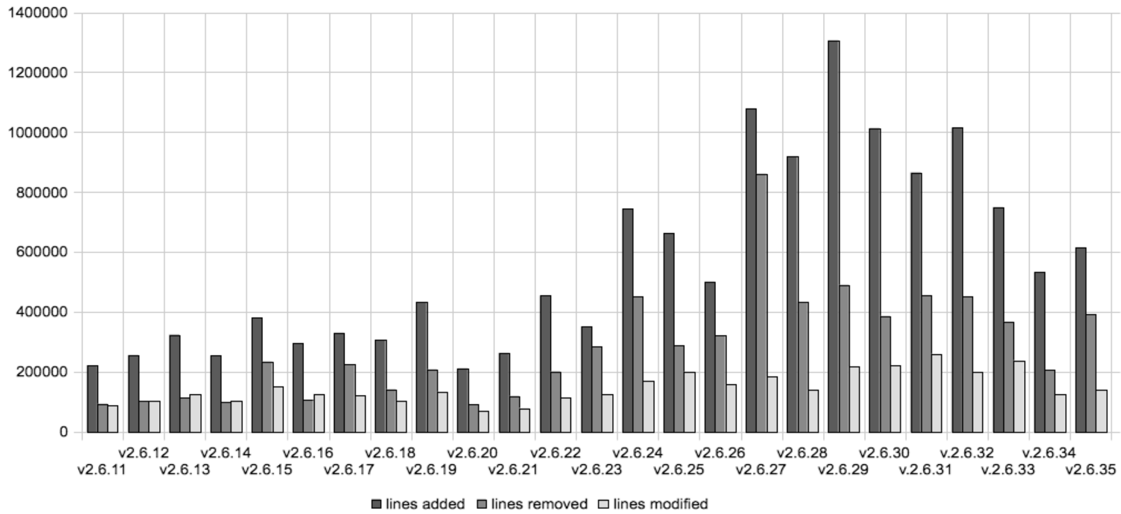

Figure 2. Kernel Source Lines of Code variation

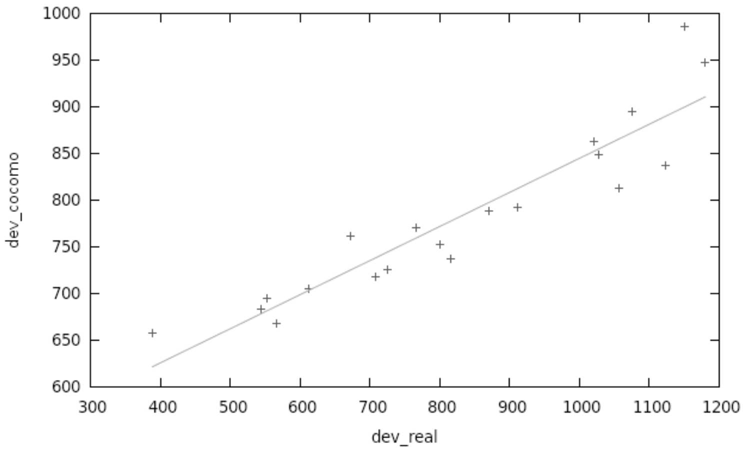

Figure 3. Number of developers observed and predicted

4.3 LIMITATIONS TO THE APPROACH

There has been little work on modelling the effort in collaborative and incremental development. COCOMO model was designed from research on privative software development. Because of that, it may undercount the complexity inherent in collaborative developed software, being a model that takes SLOC as output focus on net additions to software code. In a collaborative development model, code is developed and then changed and deleted and this effort is not reflected in the values associated with this estimate. We have tried to solve this issue by using addition of modified and deleted SLOC, since they also involve effort and resource consumption. We must remark that privative developments of software doesn´t need to deal with this matter; consequently methodological treatment of modified and deleted SLOC has been omitted on parametric models. Nevertheless, SLOC methods equate value to quantity. But the impact in innovation is not just measured in lines of code. The numbers obtained from COCOMO represent how much it would cost to develop the software from fresh start; therefore this estimates the costs but not the value to the industry ecosystem.

In addition, collaborative development means there often will be multiple agents working on different procedures to solving the same technical problem. Since only one approach is included in the delivered software, the effort invested in the alternatives isn't included in SLOC estimates. Furthermore, an enhanced COCOMO-style effort model will be needed to compute the gain in productivity in working with smaller incremental components as done in collaborative development.

Future research will be needed about costing models on OSS. In the present state of the art, they undercount the complexity inherent to open methodologies, because in a collaborative development model code is continuously added, but also deleted and modified. We also need to re-think models in order to include reuse, evolution and interfertilization of code by first creators and follow-on innovators.

5. CONCLUSIONS

Actual accounting systems are based on transactions, but in the current knowledge-based economy much of the value creation precedes, sometimes by years, the occurrence of transactions. This is a major reason for the growing disconnect between market values and financial information.

There is a great value on open sourced innovation. Because open source assets are developed collectively, there is no single source for cost estimates of how much it has taken to develop the technology. Despite absence of book value, we think they must receive a higher level of official recognition that would set it as an alternative to decision-makers. Legal and regulatory framework should allow companies participating on open sourced models to generate intangible assets for their contribution to successful projects. Otherwise, expenses must have an equitable tax treatment as a donation to social welfare as argued by Ghosh (2006).

In both cases, financial measurements are needed. Tax treatment may be favoured by the fact that many communities of developers are organized on foundations, which represent the community of developers (individuals and firms), serve as the representative of the projects under its responsibility and provide financial support and legal certainty. This phenomenon in open source development has been studied by Riehle (2010) and Riehle and Berschneider (2012). Foundations can obtain legal coverage for tax-exempt activities and donations and could be the fair accounting entity to centralize data about value of open sourced developments.

However, accounting framework for financial reporting is not ready to estimate OSS value, neither in cost, income or market valuations. Development costs are not capitalized as assets; the great value of a community crowdsourcing the development is not reported neither on fundamental statements (e.g. balance sheet) neither on complementary documents to financial reports. Nevertheless, we caution that is possible that value should not be reduced entirely to monetary terms. Several technical and social advantages cannot be communicated or decide upon effectively as financial issues.

The effect of OSS over social welfare could be also subject of disclosure in corporate social responsibility report. A number of reporting guidelines or standards have been developed to serve as frameworks for corporate social responsibility reporting. Despite none of them make mention of open sourced projects or commons creations, reporting of corporate social responsibility may be followed as a model to disclosure a community value report with quantitative and qualitative indicators.

REFERENCES

ABOODY, David; LEV, Baruch. The Value Relevance of Intangibles: The Case of Software Capitalization. Journal of Accounting Research, n. 36 (Supplement), p. 161-191, 1998.

AMOR-IGLESIAS, Juan José; GONZÁLEZ-BARAHONA, Jesús María; ROBLES-MARTÍNEZ, Gregorio; HERRAIZ-TABERNERO, Israel. Measuring Libre Software Using Debian 3.1 (Sarge) as a Case Study: Preliminary Results. UPGRADE The European Journal for the Informatics Professional, vol. VI, n. 3, p. 13-16, 2005.

ANAGNOSTOPOULOU, Seraina. R&D expenses and firm valuation: a literature review. International Journal of Accounting and Information Management, vol. 16, n. 1, p. 5-24, 2008.

BOEHM, Barry W. Software Engineering Economics. Prentice Hall: Englewood Cliffs, NJ, 1981.

CHAN, Su Han; MARTIN, John D.; KESSINGER John W. Corporate Research and Development Expenditures and Share Value. Journal of Accounting and Economics, n. 26, p. 255-276, 1991.

FASB. Statement of Financial Accounting Concepts No.6. Norwalk (CT): Financial Accounting Standards Board, 1985

GHOSH, Rishab Aiyer. Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector. Maastricth: European Communities, 2006.

GIVOLY, Dan; SHI, Charles. Accounting for software development costs and the cost of capital: Evidence from IPO underpricing in the software industry. Journal of Accounting, Auditing & Finance, n. 23, p. 271–303, 2008.

GONZÁLEZ-BARAHONA, Jesús María; ORTUÑO, Miguel Angel; DE LAS HERAS, Pedro; CENTENO, José; MATELLÁN, Vicente. Contando patatas: el tamaño de Debian 2.2. Novática, n. 154, p. 30-37, 2001.

GORNIK-TOMASZEWSKI, Sylwia.; MILLÁN, Miguel Angel. Accounting for Research and Development Costs: A Comparison of U.S. and International Standards. Review of Business – Saint John's University of New York, vol. 26, n. 2, p. 42-47, 2005.

HU, Quing; PLANT, Robert T.; HERTZ, David B. Software Cost Estimation Using Economic Production Models. Journal of Management Information Systems, vol. 15, n. 1, p. 143-163, 1998.

IASB. IAS 38 Intangible Assets (revision July 1st, 2009). London: International Accounting Standards Board, 1998.

IASB (2001). Framework for the preparation and presentation of financial statements. London: International Accounting Standards Board.

LEV, Baruch; SOUGIANNIS, Theodore. The Capitalization, Amortization and Value Relevance of R&D. Journal of Accounting and Economics, n. 21, p. 107-138, 1996.

LINUX FOUNDATION. Estimating the Total Development Cost of a Linux Distribution. Technical Report, 2008. Available on: http://www.linuxfoundation.org/publications/estimatinglinux.php Accessed in: April, 2011.

LINUX FOUNDATION. Linux Kernel Development. How Fast It is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It. Technical Report, 2010. Available on: http://www.linuxfoundation.org/publications/whowriteslinux.pdf Accessed in: April, 2011.

MOHD, Emad. Accounting for Software Development Costs and Information Asymmetry. The Accounting Review, n. 80, p. 1211-1231, 2005.

PARK, Robert E. Software Size Measurement: A Framework for Counting Source Statements. Technical Report CMU/SEI-92-TR-020, 1992. Available on: http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.020.html Accessed in: June, 2010.

RAYMOND, Eric S. The Cathedral and the Bazaar. Sebastopol (CA): O'Reilly, 1999.

RIEHLE, Dirk. The Economic Case for Open Source Foundations. IEEE Computer, vol. 43, n. 1, p. 86-90, 2010.

RIEHLE, Dirk; BERSCHNEIDER, Sebastian. A Model of Open Source Developer Foundations. In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON OPEN SOURCE SYSTEMS, Hammamet (Tunisia), 2012.

SHI, Charles. On the Trade-off Between the Future Benefits and Riskiness of R&D: A Bondholders' Perspective. Journal of Accounting and Economics, vol. 35, n. 2, p. 227-254, 2003.

SOUGIANNIS, Theodore. The Accounting Based Valuation of Corporate R&D. The Accounting Review, vol. 69, n. 1, p. 44-68, 1994.

STALLMAN, Richard M. Why Open Source misses the point of Free Software, 2007. Available on: http://www.gnu.org/philosophy/open-source-misses-the-point.html Accessed: May, 2010.

WHEELER, David A. More than a Gigabuck: Estimating GNU/Linux's size, 2001. Available on: http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html Accessed in: December, 2010.