Contributing to the common pool resources in Open Source software. A comparison between individuals and firms

Andrea Bonaccorsi, Cristina Rossi[*]

Sant'Anna School of Advanced Studies

Institute for Informatics and Telematics (IIT-CNR)

Pisa, Italy

SECOND DRAFT

Abstract

This paper studies the contributions to Open Source projects of software firms. Our goal is to analyse whether they follow the same regularities that characterize the behaviour of individual programmers.

An exhaustive empirical analysis is carried out using data on project membership, project coordination and contribution efforts of 146 Italian firms that do business with Open Source software.

We follow a meta-analytic approach comparing our findings with the results of the surveys conducted on Free Software programmers. Moreover, the availability of the data gathered by Hertel et al. (2003) on 141 developers of the Linux kernel will allow direct comparisons between the two sets[1].

[*] Corresponding author. Address: P.zza Martiri della Libertà 33, 56127, Pisa; Tel.: +39 050 883343; Fax: +39 050 883344. *E-mail address:* cristina.rossi@iit.cnr.it

[1] We thank Professor Hertel for making available his own data for direct comparison.

1. Introduction

Open Source software is gaining momentum. Two facts witness its astonishing diffusion. On one hand, the demand for Open Source solutions is rising very fast; nowadays thousands of individuals and organisations are running Open Source programs on their systems. On the other hand, there are more and more Open Source projects and an ever-increasing number of programmers contribute to them.

In the mid of September 1991, Linus Torvalds released the first version (0.01) of the Linux Operating System to the Free Software community (Diamond and Torvalds, 2001).

After slightly more than a decade, the approximate number of users is estimated to be around eighteen million (Schweik and Semenov, 2003). Citing the Gartner Group, Shankland (2003) reports that the *sales of Linux servers increased 63 percent from 2001 to 2002, from $1.3 billion to $2 billion*. Torvalds' operating system is spreading all around the world: in Japan 49.3% of the IT solution vendors support Linux (Mitsubishi Research Institute, 2003). Moreover several surveys state that the use of Linux by corporate users is now booming (Mortali, 2002; Dignan, 2002).

Some Open Source programs play a leading role in supporting Internet infrastructure and have been the *killing applications* on their markets.

The Web Server Apache entered the market in 1995. In 1996 it gained reputation supplanting the server software of the National Centre of Supercomputing Applications (NCSA) at the University of Illinois. In May 2003, the Netcraft Web Server Survey (Netcraft, 2003) reported that Apache is used on 62.53% of the Internet connected computers and powers 66.52% of all the active Web sites all around the world[2].

Also Sendmail, the OSS/FS e-mail server invented by Eric Allman more than 20 years ago is leading its market (Sendmail Inc., 2003, Wheeler, 2002). In a survey conducted between September 2001 and October 2002, Bernstein (2001) found that Unix Sendmail had the largest market share powering 42% of all e-mail servers. The market share of its leading competitor, Windows Microsoft Exchange, is around 18%.

Almost 95% of the domain name servers, which take human-readable machine names and translate them into IP numeric addresses and vice-versa (Wheeler, 2002), used some version of the Open Source program BIND.

Moreover, PHP recently surpassed Microsoft's ASP to become the most popular server-side Web scripting technology on the Internet. PHP is running on over 9 million sites out of the 37.6 million surveyed worldwide and over the past two years has averaged a 6.5% monthly growth rate (Wheeler, 2002; Hughes, 2002).

The unremitting programming activity of an increasing number of Free Software developers pushes forward the diffusion of the Open Source software.

Several metrics account for the size of the Open Source community, first of all the number of Open Source projects and developers working on them. The Free Software projects posted on the Internet are rising very fast. On 12th May 2003, SourceForge[3], one of the largest Open Source repositories, hosted 61,485 projects and numbered 621,950 registered users. In a year, projects have almost doubled. Lerner and Tirole (2002b) report that on May 2002 SourceForge contained approximately 39,000 projects.

The projects hosted at the SourceForge are only a subset of all existing ones. Much smaller competing sites come alongside this repository. According to Kim (2003), in March 2003, Freshmeat[4] contained 27,514 projects. Two months later this databases was hosting 28,310 projects, numbering 230,055 registered users[5].

[2] The same survey reported that Microsoft IS, which is the leading competitor of Apache, is used on 27.17% of the Internet connected computers and powers 24.64 % of all the active Web sites.

[3] http://www.sourceforge.net, accessed on 12th May 2003.

[4] http://freshmeat.net, accessed on 13th May 2003.

[5] On 13th May 2003. On 11 February 2002, the Freshmeat's statistics listed 18,540 Open Source projects and 109,191 registered users (Schiff, 2002).

During the last year another repository, Savannah[6] experienced a similar growth, although at a smaller scale. At present the projects hosted at Savannah Web site are 1,610 (registered users: 16,758**).** They were 790 in May 2002.

The Free Software foundation maintains another important repository, the GNU Free Software Repository[7]. This database hosts only projects released under the GPL licence. In March 2003 it hosted 2,077 projects (Kim, 2003). At present[8] they are 2,209.

Finally projects maintaining their own Web site have to be added to the ones hosted on the repositories. This is the case of many high profile projects e.g. Apache, Perl, Sendmail, Linux (Freeh et al., 2002).

This decentralized structure of the Open Source production mode makes impossible to calculate how many developers form the Open Source community.

Empirical analyses (Knoch and Schneider, 2002; Mockus et al., 2000) highlight that most successful Open Source projects are far from being anarchical communities and have a strong hierarchical organisation. Nevertheless scholars have highlighted that the involvement in Open Source projects resembles the participation to social movements (Coleman et al., 2003; O'Mahony, 2003). People, in fact, contribute to Open Source projects on their own free will.

Agents can freely join the community of the developers of an Open Source project. The admittance to the programmers' group is subject only to the demonstrated contribution of valuable code (von Krogh et al., 2003). Developers are not employees of the project and project relations are not guided by employment relations (O'Mahony, 2003). In general, nobody is forced to perform a particular task and the agents choose to focus on problems that best fit their own interests and competencies. At the same time they can resign the project whenever they want and whatever the reason.

Estimations of the size of the Open Source community can be computed in various ways. From 23th April to 1st May 2002, Krishnamurthy (2002) collected data on the *top 100 most active*

[6] http://savannah.gnu.org/, accessed on 12th May 2003.
[7] http://www.gnu.org/directory/, accessed on 13th May 2003.

projects in the mature [stage] class[9] hosted by SourceForge. He found out that there are on average 6.61[10] developers working on each project. Since the total number of developers per project is likely to decrease over the life cycle of the project, a lower bound for the size of the Free Software community can be roughly computed multiplying this mean value by the number of projects contained in the above-cited repositories. In this way a population of 618,788 programmers is obtained.

Several surveys assess the number of developers involved on samples of Open Source projects. In *The Orbiten Software Survey*, Ghosh and Prakash (2000) have taken into account a fairly representative sample of the software projects released under the GNU Public Licence and its variants, using a code base of 1,067 MB. They found 12,706 identifiable authors involved in 3,149 projects. In 2002, Ghosh et al. used a larger code base (*almost five billion of bytes of software source code*) gathering data on 31,999 developers.

Moreover the Free Software community is growing fast, especially with reference to GNU/Linux applications. In November 2001, Evans Data published the results of a survey conducted on a sample of 400 developers representing over 70 countries[11]. It was found that *48.1% of international developers and 39.6% of North Americans plan to target most of their applications to GNU/Linux* (Wheeler, 2003). The same survey was repeated in October 2002 (Evans Data, 2002). It reported that *59% of developers expect to write Linux applications in the next* year (Wheeler, 2003).

Since the year of creation of the Free Software Foundation by Richard Stallman, the developers' community has been supplying an astonishing volume of source code. For example, the size of the Linux kernel grows exponentially. The first version of Linux (version 0.01, September

[8] On 13th May 2003.

[9] The mature stage of an Open Source project is the final stage of the project's development, when it is almost fully functional and distributed.

[10] The top 100 projects are likely to have more developers than the others. Taking into account only the projects hosted at the repositories counterbalances this bias. This, in fact, leaves out the projects having their own Web site that are in general the largest ones.

[11] November 2001 edition of the Evans Data International Developer Survey Series.

1991) was formed by 10,239 lines of code (0.2 MB) while the one released on September 2002 (version 2.5.37) was 5,100,081 lines of code (152 MB) (Brouwer, 2003).

Many scholars have tried to measure this new and pervasive phenomenon. Nevertheless most of the empirical analyses *gather data on individual programmers* (Dempsey et al., 1999; Ghosh, 2003; Ghosh et al., 2002; Ghosh and Prakash, 2000; Freeh et al., 2002; Healy and Schussman, 2003; Hunt and Johnson, 2002; Kim, 2003; Koch and Schneider, 2002; Krishnamurthy, 2002; Mockus et al., 2000). Little is know about firms that base their business on Open Source software.

The analysis of their behaviour is a very challenging issue for economic theory. Two trends, in fact, are affecting the software market. On one side, several large software companies are releasing the source code of their programs to the Open Source community. On the other side, new firms are entering the software market with business models based on the supply of Open Source products and services (Hawkins 2002; Hecker, 2000; Mustonen; 2002). They play a leading role in promoting the diffusion of the Open Source software among an increasing number of users by providing support services and implementing more user-friendly solutions.

Several studies address the involvement of large software companies, such as HP, Compaq, Dell and even Intel and IBM in well established Open Source projects, such as Linux and Apache (Ahmed, 2000; Babcock, 2001; Foley, 2000; West and Dedrick, 2001; Wichmann 2002a, 2002b). Nevertheless at present we are not aware of surveys that have gathered data on the participation of firms supplying Open Source solutions to Free Software projects.

In order to fill this gap, during 2002, we conducted a large-scale survey on 146 Italian firms whose business models are based on Open Source software. We collected information on a large set of variables dealing with the characteristics of the firms, their attitude towards the Open Source phenomenon and their linkages with the Free Software community. In particular we gathered data on the number of projects in which each firm has been engaged from the very start of its Open Source activity or is currently participating, the percentage of line of codes that it contributes on average to each project and the number of its contributions (patches and modules) that have been included into the project official versions. We asked also the number of projects that they have been coordinating.

The aim of this paper is to use these data to study the contribution to Open Source projects given not by individual programmers but by business organisations. Our goal is to analyse whether it is shaped by the same stylised facts that characterize individual developers' contributions. We follow a meta-analytic approach comparing our findings with the results of the surveys conducted on Free Software programmers. Moreover, the availability of the data gathered by Hertel et al. (2003) on 141 developers of the Linux kernel will allow direct comparisons between the two sets[12].

The paper is organised as follows.

Section II surveys the literature on individual developers' contributions to Free Software projects. The methodologies of the most important empirical analyses and the main stylized facts emerging within Open Source projects are described.

Section III contains a description of our sample and variables. Data on firms' participation within Open Source projects are analysed. In order to address the relationships between Open Source involvement and firms' motivations correlation analyses and regression models have been used.

Section IV compares our findings with the main results of the surveys made on individual programmers. The role played by different classes of incentives (economic, technological and social motivations) in determining the extent of the Open Source engagement of firms and developers is examined.

Section V summarises the main conclusions of the paper.

[12] We thank Professor Hertel for making available his own data for direct comparison.

2. *Contributing to common pool resources. A survey of stylized facts form the literature*

Open Source projects share some common characteristics with respect to the pattern of contribution of individual programmers. Several surveys (Bates et al., 2002; Centeno-Gonzales et al., 2003; Dempsey et al., 1999; Ghosh et al., 2002a, 2002b; Ghosh and Prakash, 2000; Freeh et al., 2002; Hars and Ou, 2002; Healy and Schussman, 2003; Hertel et al., 2003; Hunt and Johnson, 2002; Kim, 2003; Koch and Schneider, 2002; Krishnamurthy, 2002; von Krogh et al., 2003; Mockus et al., 2000) have addressed this issue shading light on the inner structure of the Free Software community. In particular they identify some empirical regularities in the number of projects joined by developers and in the distribution of programmers' efforts.

Data collection is organised according to two main methodologies. Some authors (Bates et al., 2002; Lakhani et al., 2003; Ghosh et al. 2003b) have submitted questionnaires to samples of Open Source developers. Sample selection is carried out following different criteria. Several studies focus on only one project, usually chosen among the most successful ones (Hertel et al., 2003; Evans Data Corp., 2003). Other analyses, instead, target the whole Free Software community (Hars and Ou, 2002). The familiarity with the Internet of Open Source programmers is exploited by asking developers to fill on line questionnaires or posting the announcements of the surveys on newsgroups and mailing lists.

The main advantage of this methodology is that it allows to gather information not only on contributions to the software development process but also on programmers' characteristics. Data dealing with gender, age, place of origin, educational qualification and professional status are usually collected shading light on the demographics of the Open Source community.

Another methodology uses the project itself as a source of information. Given the decentralized fashion of the Open Source production mode, all the activities carried on within a project *leave detectable traces* in mailing lists, newsgroups, control version systems and problem reporting databases (German and Mockus, 2003). They represent important data sources. Nevertheless, in order to draw out the contributing authors and determine the extent of the coding effort of each programmer these data need to be cleaned and validated. These databases, in fact, are not designed as measurement tools.

It could be done for a given project by reviewing many discussion lists but this an extremely time-intensive procedure (Lanzara and Morner, 2003; Schweik and Semenov, 2003). Such lists, in fact, usually contain many posts that have not a standard structure allowing to read the data in an automated fashion (Ghosh, 2003). In other to overcome this problem, automated procedures of data extraction have been implemented on the other project sources of information (Fielding et al., 2002a, 2002b; Mockus et al., 2000; German 2002; German and Mockus, 2003).

Following this approach, some authors (Dempsey et al., 1999; Koch et al., 2002; Centeno-Gonzales et al., 2003) run automated procedures on version control and problem tracing databases of selected projects. Version control systems perform tasks that favour the coordination among developers[13]. In particular they record the changes that each developer has made in the project files. Problem tracing databases, instead, contains bug reports. These data are very useful to account for the characteristics of Open Source projects (German and Mockus, 2002).

In order to target larger samples of developers other scholars (Ghosh and Prakash, 2000; Krishnamurthy, 2002) apply automated procedures on large code bases downloaded from the Open Source repositories. According to Ghosh et al. (2002a), in fact, the source code of a program includes pure hard data that can be gathered through automated analysis. In particular it contains documentation that provides information on *the authorship of the software* (Ghosh, 2003). In 1998, Ghosh and Prakash implemented a software application that scans the code automatically. They called it CODD that is the acronym for *Concentration of Developer Distribution*, choosing this name probably because of the main result that they obtained using this procedure.

[13] For an exhaustive definition of VCS see Haley and Schssman (2003) note 5, page 10.

CODD extracts data on authorships, concentration and diversity of the contributions, degree of intersection between projects and code sharing, participation of developers in different projects; volatility of changes to the code base, size and integrity of the files, and groups of authors who work together *at a sub-package/ component level*. The automated fashion of this procedure allows to explore quickly a large code base and leads to results far more objective than any sample-base interactive survey (Ghosh, 2003).

Despite the differences in the methodologies of information retrieval, all thee surveys detect a set of stylized facts that shape the structure of Open Source projects.

Empirical investigations on developer demographics show that women are very rare. Hars and Our (2002) surveyed 79 Open Source programmers finding out that 95% of the respondents are males. About the same percentages have been obtained in surveys conducted on larger samples (Bates et al., 2003; Hertel et al., 2003; Ghosh et al., 2002a and Robles et al., 2001)[14]. As a rule programmers are young. Most of the respondents to the questionnaire submitted by Hars and Ou (2002) are between 20 and 40 years old[15]. Other surveys[16] highlight a similar age structure (Ghosh et al., 2002a). It has been detected that many students and young IT specialists take part in Free Software movement (Bates et al., 2003).

Moreover these studies bring several well-known myths about Free Software back into perspective. The movement is not so geographical dispersed as Open Source evangelists claim. Developers do not come from all over the world. Most of them live in the European Union or in the United States (Bates et al., 2002; Robles et al., 2001)[17]. There are programmers that do not work for free (Hertel et al., 2003) and many of them do not have a degree or a Ph.D. (Hars and Ou, 2002) in computer science.

[14] Bates et al.: sample size= 695, male percentage: 98%. Ghosh et al.: sample size= 2,784, male percentage: 98.9%. Hertel et al.: sample size: 141, male percentage: 95.7%. Robles et al.: sample size= 5,875, male percentage: 98.60%.

[15] Robles et al. (2001) have found that *the average developer is 27 years old*.

[16] Bates et al.: average age = 30. Ghosh et al.: average age = 27.1. Hertel et al.: average age = 30. Hars and Ou: average age = between 20 and 40 years old.

[17] For instance, it has been found that only 0.16% of the Debian developers live in Africa (Robles et al, 2001).

Several metrics are available to measure the level of activity within the Open Source community (Healy and Schussman, 2003), first of all project size (number of developers per project), project membership (number of projects joined by each developer) and contribution effort of each participant (Freeh et al., 2002) in terms of man/hours.

The analysis carried out by Krishnamurthy (2002) on the top 100 mature projects hosted at SourceForge have shown that only 29% of all the projects *had more than five developers* while on the other extreme, 22% of projects had only one developer associated with them. One is both the modal and median number of developers.

Another study on SourceForge data corroborates the hypothesis that the Free Software development community is dominated by single developer projects (Kienzle, 2001)[18]. About 18,700 projects out of 27,918 (67%) have a single developer while only 7 can *be classed as large*. The average project size is about 2[19].

The vast majority of the Open Source developers get experience from a rather small number of projects. Ghosh et al. (2002a) report that more than 70% of the respondents to their survey work on less than 6 projects while a very small group of programmers (3%) have been developing more than 20 Free Software programs. Robles et al. (2001) have found that 1,527 programmers out of 5,478 (27.87%) are involved in a single project and only 60 developers take part in more than 10 projects. A similar result has been obtained by Ghosh and Prakash (2000)[20.].

Another participation metric, the amount of time that individuals spend developing Free Software, displays a similar pattern. The empirical analyses belie another Free Software myth: the idea championed by the Open Source evangelists (Raymond, 2001) that developers spend all their spare time writing code. Only few programmers set apart whole days for developing Free Software, most of them choose to spend only few hours a week working on projects. Robles et al. (2001) have found that *only 278 programmers out of 5,233 develop more than 20 hour a*

[18] The data relates to projects hosted on SourceForge before 9 October 2001 (Kienzle, 2001).

[19] Mean: 1.94, Std. dev .: 2.52.

[20] The authors surveyed 12,706 participants in Open Source projects finding out that 9,617 work on one project and only 25 authors take part in more than 25 projects.

week. Similar results have been obtained by Ghosh et al. (2002a). Moreover, these authors have highlighted also that *the development of Open Source/Free Software is not at all a matter of leisure "work" at home*: 95% of their sample claim that they use OS/FS at work, school, or university.

Besides the number of participated projects, it is possible to rank developers by the extent of their contribution effort.

Running CODD on about 5,000 MB of Open Source code downloaded from the Internet, Ghosh et al. (2002b) have determined how many bytes are authored by each developers and have ranked the authors by the size of their contribution. They have discovered that contribution efforts are *unevenly distributed* among programmers: few of them provide the bulk of the code. *The first decile, that is the top 10% developers, makes up almost three quarters (74%) of the whole amount of the scrutinized software code*. Ghosh and Prakash (2000) have detected a similar contribution pattern on a smaller code base (1,067 MB). The top 10% developers (1,271 out of 12,706) have written more than 72% of the code while the top 10 authors account for 20% of the contributions. Apart from these very active developers, the contributions are a suddenly decrease. The second and third decile develop respectively 8.928% and 4.062% of the code while the last one accounts only for 0.060% of the contributions.

Also several cases studies of successful Open Source projects underline such unequal distribution of participants' contribution efforts. In general these studies make reference to a well know unit of measure of software size: the *Line of Code (LOC)*[21]. They report for the percentage of overall LOCs written by each author (Centeno-Gonzales et al, 2003; German, 2002; German and Mockus, 2003; Herman et al. 2000; Hertel et al., 2003; Koch and Schneider, 2002; Mockus et al., 2000).

[21] According to Conte et al. (1986), *a line of code is any line of program text that is not a comment or a blank line, regardless of the number of statements or fragments of statements on the line. This specifically includes all lines containing program headers, declarations, and executable and non-executable statements.*"

Mockus et al. (2000) have highlighted a high degree of concentration of the programming activity within the Apache project. The *top 15 developers*, forming the so-called core development group, account for the most of the code changes. They *contribute 88% of added lines and 91% of the deleted lines.* Koch and Schneider (2002) have identified 301 *programmers who currently work* on GNOME project*, or have worked upon this software*. Such programmers *differ significantly* with respect to the efforts that they devote to this software. Only 52 individuals are responsible for about 80% of the added LOCs. According to Hermann et al. (2000) a similar distribution of the contributed LOCs emerges in the community of Linux kernel developers.

In short, empirical analyses have detected three main stylized facts dealing with the leading role played by skewed distributions across a wide range of Open Source activities.

Fist of all the size distribution of projects, as measured by the number of programmers taking part to each project is highly skewed. A small number of projects attract the most of the development activities. Secondly. another skewed distribution governs project membership (number of projects joined by each programmer). Few agents take part in a large number of projects. Nevertheless looking at the data of Robles et al. (2001) it is worth to notice that project membership distribution displays a peculiar behaviour at the tails. The membership function is *monotonically falling* in the range 1-9 but it has a peak at 10 or more projects. Only 26 programmers are involved in 8 projects and only 5 are involved in 9 projects but there are 60 programmers that engage in 10 or more projects[22]. The same happens bearing reference to the hours spent developing Free Software. The programmers that spend more than 40 hours a week carrying on Open Source activities are more than the ones that declare to set apart for them between 20 and 40 hours. This sheds light on the presence of a *most experienced elite within the community of OS/FS developers* (Ghosh et al., 2002a).

[22] The data collected by Ghosh et al. display the same pattern. 0.2% of the programmers belong to the bracket 51-75 projects and 0.1% of them belong to the bracket 76-100 projects. Nevertheless 0.5% of the developers work on more than 100 projects.

Finally the distribution of the contributions within the projects is *spectacularly skewed*. A small number of programmers bear the most of the programming efforts.

This *challenges the common image of the Open Source community as a relatively flat network of interacting peers* (Healy and Schussman, 2003). The model of the Open Source production mode as a *bazaar* where thousands of developers exchange the source code of their programs does not fit most of the activities of the community. It is truly effective only within successful projects. Then more attention has to be paid to characteristics of such projects, with respect in particular to developers' skills and organisation.

Several scholars have addressed the stylized facts that shape Open Source projects making reference to power law distributions (Freeh et al., 2002; Hunt and Johnson, 2002, Healy and Schussman, 2003). That is distributions whose density functions takes the form:

$$f(x) = C \cdot x^{a}$$

with $a \leq 1$.

Empirical regularities of this sort shape a wide range of natural and social phenomena. They include earthquakes' intensity (Beirlant et al., 1999; Kagan et al., 1996; Richter et al., 1958), city population (Krugman, 1996; Gabaix, 1999a, 1999b), firm size (Axtell, 2001; Hart and Outlon, 1997), incomes of individuals and companies (Okuyama et al., 1999). Bearing reference to the Internet, power law distributions characterize links (Barabasi et al., 1999), pages and visits (Adamic and Huberman, 1999, 2000) of the Web sites; size of the files posted on line (Gong et al., 2001); response time of the Internauts to new pieces of information (Johansen, 2001) and other Web surfers' behaviours (Huberman et al., 1998).

Healy and Schussman (2002) focus on a peculiar power law distribution, the Zipf law, named after Harvard linguistic professor George Kingsley Zipf (1949) that have detected it in the word frequency usage in English texts. This law relates the rank and the size (rank-size property, Reed, 2001) of an event stating that the size S (or the frequency of occurrence) of some event, as a function of the rank (r), when the rank is determined by the size of the event itself, is a power law function:

$$S \approx r^{a}$$

with *a* close to –1.

Using data collected from SourceForge, they have found rank-size distributions for all project activity measures that they gathered, that is size, downloads and Web site visits, mailing list posts and CVS commits.

Also Freeh et al. (2002) have fitted power law models on SourceForge data[23] gathered at from January 2001 through March 2002. They have detected that that both project size (number of developers per project) and project membership (number of projects joined by each developers) display the classical power law plot: a straight line on a log-log scale (Urzua, 2001)[24].

Focusing on project downloads, Hunt and Johnson (2001) have obtained a *very heavily skewed distribution with a tail that extend to more than 600,000 downloads*. Plotting the number of projects vs. the number of their downloads on a log-log scale, a *characteristic* power law distribution has emerged. Among the explanations proposed in literature (see for instance Bak, 1997), the authors make reference to the *winner-take-all processes* (Cook and Frank, 1995). The emergence of small number of top performers that gather almost the all the available resources[25] characterizes these processes .

When a project grows in popularity it becomes more and more attractive so that developers are likely to join it. The opposite happens to unpopular projects that do not succeed to attract a large base of programmers. The decision to join successful projects relies on the structure of the motivations that lead developers to carry on Open Source activities. According to Lerner and Tirole (2001, 2002a), expected reputation gain among peers and talent signalling to software houses that lay at the basis of programmers' involvement in the Free Software movements.

[23] The authors collected data on 33,000 Open source developers participating in over 39,000 projects.

[24] The authors plot on a log-log scale the number of projects vs. the number of developers taking part in them and the number of projects joined by each developer vs. the number of developers joining them.

Individuals are more likely to increase their reputation if they write valuable code that is under the eyes of a large community. Moreover at present several large software companies are entering the community of the most successful projects. As a consequence, developers that contribute to these projects have more chances to attract their attention.

3. An empirical investigation of the project activity of firms supplying Open source products and services.

In order to study the extent of the involvement of firms supplying Open Source based solutions in the Free Software movement, we gathered data on several metrics of the level of activity within a project.

Six variable have been collected, dealing with:

– The number of projects which firms have been participating since the very start of their Open Source activity: *overall project membership, ALL_A_PM*[26]

– The number of projects in which firms are currently involved: *current project membership, C_ PM*

– The number of projects which firms have been coordinating since the very start of their Open Source activity: *overall coordination activity, ALL_A_CP*

– The number of projects which firms are currently coordinating: *current project coordination, C_ CP*

– The percentage of LOCs contributed on average to each project*: %_LOCs*

– The number of firms' contributions (patches, modules) inserted in project official versions during 2002: *N_C_OV.*

Table 1 reports descriptive statistics of these variables.

[25] For instance Adamic and Huberman (2000) refer to winner-take-all processes for analysing the skewed distribution of Web site visits.

[26] We do not define the overall involvement in Open Source projects conditional to the age of the firm, because most firms entered the field in the last 3-4 years.

Variable	*Acronym*	*Min*	*Max*	*Mean*	*Std. Dev.*	*Median*	*Skewness*
Number of projects joined from the very start of the Open Source activity	*ALL_A_PM*	0	50	3.8	7.8	1	3.5
Number of projects joined during the last year	*C_PM*	0	20	1.6	2,8	1	3.7
Number of projects coordinated from the very start of the Open Source activity	*ALL_A_CP*	0	28	1.1	3.4	0	5.9
Number of projects coordinated during 2002	*C_CP*	0	7	0.5	1.2	0	3.5
Percentage of LOCs contributed on average to each project	*%_LOCs*	0	99	10.56	23.5	0	2.5
Firms' contributions (patches, modules) accepted into project official versions	*N_C_OV*	0	300	6.9	36.9	0	6.7

Table 1: Descriptive statistics.

Our data show that the firms' level of activity within Open Source projects is quite limited. On average, the agents have been taking part in less than four projects since their first step in the Free Software community. About 68% of the firms have been participating in no more than two projects while only 7.7% have been involved in the development of more than 10 programs (table 2).

The median value of ALL_A_PM is 1. It is worth to notice that almost half of the sample (49.6%) is not currently engaged in Open Source projects and many firms (46.2%) have never joined one. These findings are consistent with firms' business models. Most respondents, in fact, offer services such as installation (80.1%), support (82.9%), maintenance (76%), consultancy (84.9%) and training (64.4%). As a consequence they are likely to carry on developing activities only to adapt Open Source programs to their customers' requirements without placing back these ad hoc solutions at disposal of the community.

Variable	*Acronym*	*No. of projects*	*%*	*Cum %*
Number of projects joined from the very start of the Open Source activity	*ALL_A_PM* *N = 117*	0	46.2	46.2
		1	6.0	52.1
		2	15.4	67.5
		3 - 5	17.1	84.6
		6 - 10	7.7	92.3
		> 10	7.7	100.0
Number of projects joined during last year	*C_PM* *N = 123*	0	49.6	49.6
		1	15.4	65.0
		2	13.8	78.9
		3 - 5	15.4	94.3
		6 - 10	4.1	98.4
		<10	1.6	100.0
Number of projects coordinated from the very start of the Open Source activity	*ALL_A_CP* *N = 121*	0	72.9	72.9
		1	10.2	83.1
		2	6.8	89.9
		3 - 5	5.9	95.8
		6 - 10	2.5	98.3
		> 10	1.7	100.0
Number of projects coordinated during 2002	*C_CP* *N = 118*	0	78.5	78.5
		1	11.6	90.1
		2	4.1	94.2
		3 - 5	4.1	98.3
		6 - 10	1.7	100.0
		> 10	0.0	100.0
		% of LOCs		
Percentage of LOCs contributed on average to each project	*%_LOCs* *N = 104*	0	59.6	59.6
		≤ 10	23.1	82.7
		11-30	5.8	88.5
		31-50	3.9	92.4
		51-80	3.8	96.2
		≥ 81	3.8	100.0
		No. of contributions		
Firms' contributions (patches, modules) accepted into project official versions	*N_C_OV* *N = 101*	0	72.3	72.3
		1	6.9	79.2
		2	5.9	85.1
		3-5	5.9	91.0
		6-10	5.0	96.0
		> 10	4.0	100.0

Table 2: Firms' level of activity within projects.

As we expected, the firms in our sample have gotten little experience from project administration. On average they have been coordinating a single project since they entered the Free Software community. The large majority of the firms (72.9%) have never carried out coordination tasks. Only 26 firms (21.5%) are currently coordinating a project, mostly (53.8%) just a single one. This is due to several factors. Like individual developers, firms probably prefer to join large, successful projects whose leadership has been already settled down. These projects provide largely diffused software products that give the firms more chances to supply services to their users. Moreover these programs are released in stable official versions. As a consequence it is easier for the software firms to adapt them to their needs. At the same time contributing to successful projects improves the corporate image, hitting favourably costumers and possibly venture capitalists. The newness of the phenomenon must be taken into account too. Most firms, in fact, entered the Open Source arena just four years ago[27]. As a consequence, they have not yet gained enough reputation among Open Source developers to be appointed to the coordination of a project.

The scanty involvement of the respondents in Open Source projects is witnessed also by the percentage of the contributed LOCs. If we class as leading authors the firms providing more than 50% of the overall LOCs of a project, then only 7.6% of the firms play this role. This has repercussions on another measure of the developing efforts within Open Source projects: the number of contributions included in official releases. Few firms have written pieces of code that have been accepted for project official versions. The mean value of variable N_C_OV is 7 but it is deeply affected by three outliers providing hundreds of accepted contributions[28]. After having excluded these agents the mean becomes one.

Mirroring the results of the surveys made on individual developers, all the observed variables display skewed distributions.

Our surveys collected data on the motivations of the decision to adopt business models based on Open Source software[29]. Following the taxonomy proposed by Feller and Fitzgerald (2002), we group these motivations into different classes (economic, social and technological motivations) [30] and have explored their weight within the agents' decisional process. It is now of interest to

[27] 64.5% of firms have been adopting the Open Source technology since 1999.

[28] The contributions of these firms included in the project official versions are respectively 100, 200 and 300. The agents ranking fourth has 12 accepted contributions.

[29] Each incentive is measured at a Likert scale ranging from 1 (not at all important) to 5 (very important).

[30] *Economic motivations*: because Open Source software allows small enterprises to afford innovation; because we want to be independent on the price and licence policies of large software companies; because in the field of Free Software we can find easily good IT specialists; because opening our source code allows to gain a reputation among our costumers and competitors. *Social motivations:* because we

study how these incentives relate to firms' active involvement in Free Software projects. Table 3 reports the significant correlations between motivational components and the metrics of project participation.

Motivations	*Acronym*	*Area*	*C_PM*	*C_CP*	*N_C_OV*
Because Open Source software allows small enterprises to afford innovation	M4	E		-0.15^{*}	
Because we want to place our source code and skills at disposal of the Free Software community and we hope that others do the same thing	M5	S			0.18^{*}
Because we conform to the values of the Free Software movement	M6	S	0.15^{*}		0.25^{**}
Because we think that software should not to be a proprietary good	M7	S			0.27^{***}
Because we want to study the code written by other programmers and use it for developing new programs and solutions	M9	T			0.17^{*}

Table 3: Motivations and project participation. Correlation analysis.

*: p value < 0.10; **: p value < 0.05; ***: p value< 0.01.

The number of firms' contributions accepted for project official versions is positively correlated with the learning incentive and all social motivations.

The Open Source production mode places a huge volume of source code at disposal of whoever likes to modify, debug or just study it. Firms that exploit this *immense learning opportunity* have greater chances to improve their skills in Open Source programming. As a consequence they are more likely to write pieces of source code that the community holds enough valuable to be accepted into the official versions of the programs.

Moreover, the performance within Open Source projects is positively affected by social motivations. On one side, firms that gift the code, conform to Open Source values and fight for the software freedom have stronger linkages with the Free Software community than agents who entered the Open Source arena only to exploit new business opportunities. These firms want to actively take part in the Open Source movement. They therefore devote much effort in Open Source activities and this has positive repercussions on the number of accepted contributions.

conform to the values of the Free Software movement; because we want to place our source code and skills at disposal of the Free Software community and we hope that others do the same thing; because we think that software should not to be a proprietary good. *Technological motivations:* because contributions and feedbacks from the Free Software community are very useful to fix bugs and improve our software; because of the reliability and quality of the Open Source software; because we want to study the code written by other programmers and use it for developing new programs and solutions; for having products that are not available on the proprietary software market.

On the other side, individual developers trust these firms and support their Open Source involvement. They think of them as members in every respect of the community and pose no problem to include their contributions into official releases.

One could observe that social motivations are not correlated with the other measures of the level of activity. However firms that want to cultivate social ties with the Open Source community may choose to take part in a small number of projects devoting to them the bulk of their programming efforts without performing coordination tasks.

The negative correlation between M4 and the number of projects that firms are currently coordinating is more difficult to explain. It is possible that firms that emphasize the potential for innovation by small firms see a role for participating to projects without necessarily taking the duty to coordinate them.

In order to examine closely the link between motivations and level of Open Source activity, Mann-Whitney tests have been run. With reference to each incentive, firms have been divided in two groups on the basis of the scores they assigned to the appropriate variable. The first group clusters firms that declared a low score (1 or 2) while the second one includes the ones that assigned a high score (4 or 5). Table 4 summarizes the results. Only mean values displaying statistically significant differences in the two groups are reported.

The metrics of the Open Source level of activity show different patterns in the two groups depending on the incentive area (economic, social, technological).

Taking into account both the cumulated and current Open Source activity, firms that assign high scores to economic motivations have made much less experience in project coordination.

Motivations	Acronym	Area	Metrics	Group	N	Mean	Dev. Std.
Because we want to be independent of the price and licence policies of the large software companies	M1	Economic	C_CP**	LOW	20	0,7	1,1
				HIGH	76	0,4	1,4
Because Open Source software allows small enterprises to afford innovation	M4	Economic	ALL_A_CP**	LOW	13	2,0	2,4
				HIGH	85	0,9	3,3
Because we want to place our source code and skills at disposal of the Free Software community and we hope that others do the same thing	M5	Social	ALL_A_PM**	LOW	28	3,4	8,0
				HIGH	61	4,9	9,2
			N_C_OV*	LOW	24	0,6	2,1
				HIGH	51	1,5	3,0
Because we think that software should not to be a proprietary good	M7	Social	%LOC**	LOW	38	8,4	23,0
				HIGH	38	14,2	26,3
			N_C_OV-***	LOW	38	0,4	1,8
				HIGH	38	1,8	3,1
Because contributions and feedbacks from the Free Software community are very useful to fix bugs and improve our software	M8	Technological	%LOC*	LOW	14	6,6	24,0
				HIGH	70	10,1	22,8
			N_C_OV*	LOW	15	0,1	0,3
				HIGH	65	1,2	2,7
Because we want to study the code written by other programmers and use it for developing new programs and solutions	M9	Technological	ALL_A_PM*	LOW	28	4,1	10,9
				HIGH	59	4,5	7,9

Table 4: Open Source activities and motivations. Mann Whitney tests for firms assigning high and low scores to incentives.
*: p value< 0.1; **: p value<0.05; ***: p value< 0.01.

Firms that adopted the Open Source technology in order to promote innovation and emancipate from the licence and price policies of the large software companies are mainly acting out of extrinsic motivations. They are likely not to attach great importance to social links within the Free Software movement. These firms want to sustain cooperation with Open Source developers in order to obtain the feedbacks and contributions that allow them to lower down developing costs. However, these firms can win developers' trust just by gifting their code and avoiding hijacking the one that has been written by other programmers. Our data show that, in fact, those firms that attach more importance to these purely economic considerations, are less likely to be involved in coordination activities.

Other findings witness the leading role played by code gifting in promoting cooperation. The percentage of contributed LOCs and the number of accepted patches are higher for firms that attach much importance to the feedbacks from the community.

The results of the Mann- Whitney tests corroborate the findings of the correlation analysis on the role played by social motivations in shaping the level of activity of Open Source firms. As we explained above, firms that assign high scores to social incentives are more likely to win the trust of the community. As a consequence more contributions of them are accepted into official releases.

Firms that attach greater importance to social motivations also display a higher project membership and produce many more lines of code.

Firms that value very much the learning opportunities provided by the Open Source mode of production and give high scores to technological motivations clearly behave in the same way.

In particular firms that want to fight for software freedom, try hard to contribute LOCs to Open Source projects (14.2% vs. 8.4%). In this way they increase the code base that is released under Free Software licence schemes.

Finally, following the approach proposed by Hertel et al. (2003), we have run exploratory stepwise regressions using the metrics of the level of activity as dependent variable and motivational components as independent variables. The correlation matrix shows that independent variables are correlated between each other. Moreover they are categorical variables measured with a Likert scale. In order to overcome these problems, a factor analysis[31] has been run on each group on incentives. Two factors were extracted from the group of economic and technological motivations while one has resulted from the set of social motivations. We used these factors as independent variables of our regression models.

Only the model having the number of firms' contributions accepted into project official versions (N_C_OV) as dependent variable (table 5) displays coefficients that are statistically different from zero (p value = 0.001). It is proved once again that social motivations have a positive effect on the number of firms' contributions that are included in project official releases.

[31] Principal components methodology, varimax rotation.

Model	Dependent variable	Independent variables	Beta	t	Sig.	R2
1	N_C_OV	Costant	0,901	-1,908	0,000	0,11
		SM_FAT	0,328	1,804	0,001	

Table 5: Motivations and Open Source activity. Regression models.

Where SM:_FAT is the factor summarizing the group of social motivations (M5, M6, M7).

4. *Project activity of firms and individual developers. A comparison.*

In order to compare our data with the main findings of the surveys on individual developers, we follow a meta-analytic approach. The target and the methodology of all published studies are reported (table 3).

The samples of some of the empirical analyses are much larger than ours. However, if we take into account all the people working in the firms that we surveyed, it can be estimated that our data account for more than 2,300 developers[32]. This is clearly an upper bound because the staff is very likely to include non-programmers.

All the surveys but 2 and 3 gather data on current project membership. We compare these findings with overall and current project membership of the firms in our sample. As it emerges from the data illustrated above, the level of Open Source activity of the respondents to our survey is quite low. As a consequence this double comparison may be of interest.

In order to account for skewed distributions, we report the percentage of firms taking part in no more than two projects (PM_≤2), in more than five projects (PM_≥5) and in more than 10 projects (PM_≥10). The value (PM_≥5) allows to compare most of the surveys. In fact all the studies but the one conducted by Bates et al. (2002)[33] compute this variable.

[32] Our sample include total staff at around 2,388 developers. The data on firms' staff collected by the questionnaire include free lance, employees and partners.

[33] According to these authors about 20% of the developers take part in more than four projects.

		DEVELOPERS										FIRMS	
Target		Open Source community							Single projects				
Methodology		Q					AP		Q	AP		Q	
References		*Robles et al. (2001)*	*Bates et al. (2002)*		*Ghosh et al. (2002a)*		*Ghosh Prakash (2000)*	*Ghosh et al. (2002b)*	*Hertel et al. (2003)*	*Koch Schneider (2002)*	*Mockus et al. (2000)*	*Our survey*	
Survey ID		1	2		3		4	5	6	7	8	9	
Variable	*Acronym*	CA	TA	CA	TA	CA	CA	CA	CA	CA	CA	TA	C A
Sample size	*N*	5,478	2,221		2,784		12,706	12,584	141	301	388	146	
Average project membership	*PM_AV*	2.87	4.9	2.6	6.09	2.49	1.679	1.619				2.8	1.2
% of agents participating in no more than 2 projects (%)	*PM_≤2*	63.08	28.5	60		55.30	90.831	89.709				67.5	78.9
% of agents participating in over 5 projects	*PM_≥5*	4.93			28.10	5.40	1.857	2.900				15.4	5.7
% of agents participating in over 10 projects (%)	*PM_≥10*	1.24			9.40	1.00		0.858				7.7	1.6
% of LOCs contributed on average by each agent	*%LOC_AV*								0.1	0.33	0.26	10.43*	
% of LOCs contributed by the leading authors	*%LOC_LA*						72.320	73.645		80	88	61.9	
No. of contributions accepted into the project official versions	*N_C_OV*								2.3			6.9; 1**	

Table 6: Level of activity of firms and developers within Open Source projects. Meta-analysis.

Note: Q.: questionnaire; A. P.: automated procedures; C. A. current Open Source activity; T. A. cumulated Open Source activity. *: comparison non appropriate, **: without outlier values.

All the values of the variable AV_PM but the one of survey 2 have been estimated. The authors, in fact, do not report directly this figure. They define several classes of project membership and account for the number or the percentage of developers in each of them. In order to determine the average project membership we refer to the central value of each class[34].

[34] The highest classes are always open intervals. The values corresponding to them (for instance more than 25 project) were computed on the basis of the width of the other classes. This does not affect

Surveys 6, 7 and 8 do not account directly for the percentage of LOCs contributed on average by each programmer. It had to be computed. Project case studies pose no problem, since they report the total LOCs of the project and the number of developers. Hertel et al. (2003) gathered only the number of LOCs added by each developer. As a consequence, we have estimated %LOC_AV dividing this figure by the total number of LOCs in the version 2.2.0 of the Linux kernel that was released in January 1999 (Brouwer, 2001). This is consistent with the time period during which the authors collected their data: between February 15th and April 12th 2000[35]. At the same time they grouped the number of contributions inserted in the projects official versions by classes. In order to make our data comparable with these ones the central value of each class was computed[36].

It is worth to notice that the definition of leading authors used to compute %LOC_LA varies across the surveys. It may indicate the *first decile* (surveys 4 and 5) or the core developer group[37] (surveys 6 and 7).

In order to compare our survey with 4 and 5, we classed as leading authors the first decile of our firms ranked by the percentage of contributed LOCs[38]. Given that we asked the firms for the percentage of LOCs contributed to the projects in which they take part, the value %LOC_LA reported in table 6 for our survey represents the percentage of LOCs contributed on average by the leading authors.

The meta-analysis highlights that firms are less engaged than individual developers in Open Source activities. This is witnessed across most of the studies by all the metrics of the Open Source level of activity that we have taken into account[39]. However the properties that

significantly the final result given that very few developers belong to the highest classes of project membership.

[35] Another version of the Linux Kernel (2.5.3) was released in September 2002. It consist of 5,100,081lines of code.

[36] We approximated the highest class (> 10) with 15.

[37] The top 15 and 52 authors respectively.

[38] After ranking the agents by their contribution effort, contributions from the first 10% of them are taken into account. Given the sample size, the top 15 firms compose the first decile in our study.

[39] As regard to %LOC_AV, as we stated above, the comparison our survey with the other studies is not allowed.

characterize the structure of contributions of individual developers also shape firms' contributions.

Result I: in general firms join less projects than individual programmers do.

All the empirical analyses witness that on average firms' project membership is lower than individual developers' one. This holds for both the current and the overall Open Source activity of the agents.

Focusing on this latter, survey 2 and 3 show that, in comparison with individuals, firms are more numerous in the lowest project membership class (PM_≤2) and less numerous in both the highest ones (PM_≥5 and PM_≥10). These findings should be carefully weighted, in particular comparisons with further analyses are needed. Data collected through automatic procedures (survey 4 and 5), in fact, show opposite results. However, the differences in methodology and sample size are likely to affect the comparison.

Moreover the adoption timing of the Open Source technology needs to be taken into account. Many firms (64,5%), in fact, entered the Open Source market no more than four years ago. As a consequence, they might have not had enough time to get experience from a large number of Open Source projects. However the fact that each firm may hire a group of developers relaxes such time constraint. These programmers, in fact, can join different projects at the same time.

With regards to current project membership, all the surveys that have been conducted by submitting questionnaires to individual developers have obtained lower values of PM_≤2. No significant difference between the two sets of agents emerges with reference to PM_≥5 and PM_≥10[40]. This is probably due to another time constraint that in turn is experienced both by firms and individual developers. On one side, firms in order to be on the market need to carry on many activities besides the Open Source ones. On the other side, writing Open Source software often is not among the work duties of individual programmers who develop Free Software *covertly* during the working hours or in their spare time. Bates et al. (2002) report the

[40] It is worth to notice that PM≥5 and PM≥10 are slightly higher in our survey.

lack of sleep as the most important cost of participating in Open Source projects. It is worth to notice that analyses carried on through automatic procedures have found opposite results also with reference to current project membership.

Result II: in general firms devote less programming efforts to the projects in which they take part.

The case studies of single successful projects display similarities in programming efforts as measured by the percentage of LOCs contribute on average by each developer. A much larger value has been instead obtained in our study.

Actually such comparison is not correct in a methodological view. The strictly hierarchical organisation of the large successful projects, in fact, makes often very difficult for the developers to add line of codes to the projects official versions.

Most of the programmers contribute no line and this affects significantly the mean value of this metrics. With reference to our firms, %LOC_AV does not deal with a single successful project but refers to project participation in general. Together with a large group of firms contributing no LOC (59.6%) we have found out firms providing almost the overall code base of the Open Source projects in which they are involved. This is likely to happen for small projects that have been started by the firms themselves and have not been able to attract a large base of developers.

Both the surveys addressing the whole Open Source community and the project case studies witness that firms classified as leading authors contribute a lower percentage of LOCs with respect to developers holding the same position[41].

One could think that this is not due to the presence of fewer firms devoting large programming efforts but to the lower concentration the contributions. However this is not the case, given that above 88% of the agents contributes no more than 30% of the LOCs of the projects in which they are involved.

[41] The different definitions of *leading author* in the analyses that focus on single projects need to be taken into account.

Another metric witnesses the poor performances of the firms within Open Source projects: the number of contributions included in the official releases of the Open Source programs. Excluding the outlier values, it results that firms succeed in putting a smaller number of contributions into project official versions than individual developers do.

In order to go further ahead in our analysis we make reference to the data collected by Hertel et al. (2003) on 141 developers of the Linux kernel.

Two of the metrics gathered by the authors are comparable to our ones: percentage of contributed LOCs and number of contributions inserted into official releases. However, as we explained in the meta-analysis above, methodological problems allow to use only the latter.

Table 7 reports about the distribution of contributions in the two groups. In comparison with individual developers, there is a higher percentage of firms in the three lowest classes (92.1% vs. 86.4%). However no significant difference emerges in the percentage of agents that do not succeed in placing contributions in project official versions. The two highest classes display an interesting pattern. About 5% of the firms in our sample have inserted in the official releases a number of contributions ranging from 7 to 10 while only 1.6% of the programmers performed in the same way. Nevertheless 12% of the developers have had more than 10 accepted contributions versus 4% of the firms. This sheds light on the presence of an elite of individual programmers very well performing in Open Source developing activities. Firms that supply Open Source solutions do not seem to include such a group.

	Linux Kernel developers (Hertel et al. 2003)		*Firms (our survey, 2003)*	
No. of contributions	*%*	*Cum. %*	*%*	*Cum. %*
0	73,6	73,6	72,3	72,3
1-3	8,8	82,4	14,9	87,2
4-6	4	86,4	4,9	92,1
7-10	1,6	88	4,0	96,0
>10	12	100	4,0	100,0

Table 7: Distributions of the accepted contributions of individual developers and firms.

The following empirical investigations exclude the agents placing no contributions.

Mann Whitney test shows that on average firms place fewer contributions into project official versions than individual developers do (4.1 vs. 8.7, p value = 0.005).

In section III we have analysed how the motivations that lay at the basis of firms' engagement in the Open Source movement, affect their level of activity within Open Source projects. Now in order to analyse as heterogeneity in motivations between firms and individual programmers affects their Free Software performances, the comparable motivational components of the two surveys are taken into account. Table 8 reports correlation analyses for these items. The data shed light on the different role played by social and technological motivations in determining the extent of the Open Source involvement of the two groups. On one side, the number of accepted contributions of the individual developers is positively correlated with an incentive that has been classified as social. On the other side, a positive correlation has been found between firms' contributions and the learning incentive. Clearly further investigations are needed.

Motivations	*Area*	*Correlation*
Personal exchange with other software developers^{+}	S	0,3
Because we want to place our source code and skills at disposal of the Free Software community and we hope that others do the same thing		.
Code should be free	S	.
Because we think that software should not to be a proprietary good		.
Improving one's own programming skills	T	.
Because we want to study the code written by other programmers and use it for developing new programs and solutions^{+}		0,4

Table 8: Comparable items in the two surveys. Correlation analyses. $^{+}$ p value < 0.1.

Conclusions

These findings raise an interesting problem. It seems that the level of contribution to Open Source projects is a function of the presence of social and technological motivations over purely economic ones.

There are firms that exploit the low cost, large availability and good quality of Open Source software to build up a sustainable business model without contributing in the same proportion.

More generally, these findings shed light on an interesting evolutionary property of the Open Source communities (Bonaccorsi and Rossi, 2003), namely robustness.

The behaviour of contributing to a common pool resource must not be equally distributed among contributors in order to be self-sustaining. Open Source communities permit some member to take much more than they give, provided they do not violate minimal membership rules. By exploiting existing code more than they contribute, they still enlarge the bases of the Open Source users, indirectly enhancing the motivations of active producers. The literature on CPR, public good provisioning and free riding has probably over estimated the potential destructive role of a small number of non contributors, assuming that their behaviour should inevitably self-propagate. This is not necessarily true.

Nevertheless, because these firms are accepted in the Open Source community as legitimate partners, it is almost certain that they comply with the rules of membership and legal obligations coming from Open Source licensing schemes. This is to say, for example, that they do not hijack Open Source software, but rather adapt and redistribute it under an appropriate scheme.

At the same time, it is clear that these firms take more than they give. It seems that the new organisational mode of software production is robust to a distribution of contributing behaviour that include cases in which the contribution effort is limited to the minimum entry level.

References

Adamic L. A., Huberman B. A. (1999) *The growth dynamics of the World Wide Web.* Nature, 401, 131.

Adamic L. A., Huberman B. A. (2000) *The nature of the market in the World Wide Web.* Quarterly Journal of Electronic Commerce, 1, 5-12.

Ahmed P. K., Graham G., Hardaker G. (2000) *Non-linear innovation across the Internet*. Integrated Manufacturing Systems, 11(7), 484-491.

Axtell R. L. (2001) *Zipf's distribution of U.S. firm size*. Science Reprint, 293, 1818-1820.

Babcock C. (2001) *Open Source code: a corporate building block.* Interactive Week, http://zdnet.com/intweek/stories/news/0,4164,2717905,00.html, accessed on May 14th 2003.

Bak P. (1997) *How nature works: the science of self organized criticality*. Oxford University Press, Oxford, UK.

Barabasi A. L., Albert R. (1999) *Emergence of scaling in random networks.* Science, 286, 509-512.

Bates J., Di Bona C., Lakhani K., Wolf B. (2002) *The Boston Consulting Group hacker survey.* http://www.osdn.com/bcg/BCGHACKERSURVEY-0.73.pdf, accessed on August 1st 2003.

Beirlant J., Caers J., Maes M. A. (1999) *Statistics for modelling heavy tailed distributions in geology: part II applications.* Mathematical Geology, 31, 411-434.

Bernstein (2001) *Internet host SMTP server survey*. http://cr.yp.to/surveys/smtpsoftware6.txt, accessed on May 12th 2003.

Bonaccorsi A., Rossi C. (2003) *Why Open Source can succeed*. Research Policy, 32(7), 1243-1258.

Brouwer A. (2003) *The Linux kernel*. http://www.win.tue.nl/~aeb/linux/lk/lk.html#toc1, accessed on May 13th 2003.

Centeno-Gonzales J., Gonzales-Barohona J. M., Mattelan-Olivera V., Rodero-Merino L., Robles-Martinez G. (2003) *Studying the evolution of Libre software projects using publicly available data*. In Proceedings of the 3rd Workshop on Open Source Software Engineering, May, 3-11, Portland, Oregon, USA.

Coleman E. G., Hill B. M. , Michlmayer M. (2003) *The social production of productive freedom: Debian and ethical volunteerism.* Proposal accepted for publication in Koch S. (Ed.) Free/Open Source Software Development. Idea Group, Inc., Hershey, PA, USA:

Conte S. D., Dunsmore H. E., Shen V. Y. (1986) *Software engineering metrics and models*. Benjamin/Cummings, Menlo Park, CA, USA.

Cook P. J., Frank R. H. (1995) *The winner-take-all society*. Free Press, New York, NY, USA.

Dempsey B. J., Greenberg J., Jones P., Weiss D. (1999) *A quantitative profile of a community of Open Source Linux developers.* SILS Technical Report TR- 1999-05.

Diamond D., Torvalds L. (2001) *Just for fun: the story of an accidentally revolutionary*. Texere Publishing, New York, NY, USA.

Dignan L. (2002) *Survey: Linux growing; CRM in doubt*. http://news.com.com/2100-1001-956496.html, accessed on May 12th 2003.

Evans Data Corp. (2001) *Study by Evans Data Corp. Reveals That International Developers Twice as Likely as North Americans to Primarily Target Linux*. http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.111301/213170209, accessed on May 12th 2003.

Evans Data Corp. (2002) *New Survey of International Developers Shows Web Services Now Focused Inside Businesses*. http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.112602/223300066, accessed on May 12th 2003.

Evans Data Corp. (2003) *Linux development survey.* http://www.evansdata.com/n2/strategic_developer_studies.shtml#linux, accessed on May 14th 2003.

Feller J., Fitzgerald B. (2002) *Understanding Open Source Software Development.* Addison Wesley, Boston, MA, USA.

Fielding R. T., Hann I. H., Roberts J., Slaughter S. (2002a) *Delayed returns to Open Source participation: an empirical analysis of the Apache HTTP server project.* In Proceedings of the Open Source Software: Economics, Law and Policy Conference,. June 20-21 Toulouse, France.

Fielding R. T., Mockus A., Herbsleb J. (2002b) *Two case studies of Open Source software development: Apache and Mozilla*. In proceeding of the 2nd Workshop of Open Source Software Engineering, 19-25 May, Orlando, Florida, FL, USA.

Foley M. J. (2000) *IBM "Vikings" seek more Linux conquest*. Interactive Week. http://www.zdnet.com/intweek/stories/news/0,4164,2609727,00.html, accessed on May 14th 2003.

Freeh V., Madey G., Tynan R. (2002) *Understanding OSS as a self-organizing process*. In proceeding of The 2nd Workshop on Open Source Software Engineering at ICSE 2002, Orlando, Florida, FL, USA.

Gabaix X. (1999a) *Zipf's law and the growth of cities*. American Economic Review, Papers and Proceedings, LXXXIX, 129-132.

Gabaix X. (1999b) *Zipf's law for cities: an explanation*. Quarterly Journal of Economics, 8, 739-765.

German D. M. (2002) *The evolution of the GNOME Project.* In proceeding of the 2nd Workshop of Open Source Software Engineering, 19-25 May, Orlando, Florida, USA.

German D., Mockus A. (2003) *Automating the measurement of Open Source projects*. In Proceedings of the 3rd Workshop on Open Source Software Engineering, May, 3-11, Portland, Oregon, OR, USA.

Ghosh A. R. (2003) *Clustering and dependencies in Free/Open Source software development: methodology and tools.* First Monday, Peer-reviewed Journal on the Internet, http://firstmonday.org/issues/issue8_4/ghosh/index.html, accessed on May13th 2003.

Ghosh R. A., Glott R., Krieger B., Robles G. (2002a) *Survey of developers*. Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Ghosh R. A., Glott R., Robles G. (2002b) *Software source code survey.* Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Ghosh R., Prakash V. V. (2000) *The orbiten Free Software survey*. First Monday, Peer-reviewed Journal on the Internet, http://firstmonday.org/issues/issue5_7/ghosh/, accessed on 13th May, 2003.

Gong W., Liu Y., Misra V. (2001) *On the tails of the Web file size distributions*. Proceedings of 39th Allerton Conference on Communication, Control, and Computing, October, Monticello, IL, USA.

Hars A., Ou S. (2002) *Working for free? Motivations for participating in Open Source projects.* International Journal of Electronic Commerce, 6, 25-39.

Hart P.E., Oulton N. (1997) *Zipf and the size distribution of the firms*. Applied Economic Letters, 4, 205-206.

Hawkins R. E. (2002) *The economics of the Open Source Software for a competitive firm*. MIT Working Paper, http://opensource.mit.edu/papers/hawkins.pdf, accessed on August 1st 2003.

Healy K., Schussman A. (2003) *The ecology of Open Source software development*. MIT Working paper, http://opensource.mit.edu/papers/healyschussman.pdf, accessed on August 1st 2003.

Hecker F. (2000) *Setting up shop: the business of Open-Source software*. http://www.hecker.org/writings/setting-up-shop.html, accessed on March 26th 2003.

Hermann S., Hertel G., Niedner S. (2000) *Linux study homepage.* http://www.psychologie.uni-kiel.de/linux-study, accessed on May, 16th 2003.

Hertel G., Niedner S., Hermann S. (2003) *Motivation of software developers in the Open Source projects: an Internet-based survey of contributors to the Linux kernel.* Research Policy, 32(7), 1159-1177.

Huberman B. A., Pirolli P.L.T., Pitkov J. E., Lukose R. M. (1998) *Strong regularities in the World Wide Web surfing*. Science 280, 95-97.

Hughes F. (2002) *PHP: most popular server-side Web scripting technology*. http://lwn.net/Articles/1433/, accessed on May 12th 2003.

Hunt F., Johnson P. (2002) *On the Pareto distribution of Sourceforge projects*. In Proceedings of the Open Source Software Development Workshop, 122-129, Newcastle, UK.

Johansen A. (2001) *Response time of Internauts*. Physica A, 296, 539-546.

Kagan Y. Y., Knopoff L., Sornette D, Vanneste C. (1996), *Rank-ordering statistics of extreme events: application to the distribution of large earthquakes*. Journal of Geophysical Research, 101(B6), 13883-13894.

Kienzle, R. (2001). *Sourceforge preliminary project analysis*. http://www.osstrategy.com/sfreport, accessed on May 18th, 2003.

Kim E. E. (2003) *An introduction to Open Source communities*. http://www.blueoxen.org/research/00007/index.html, accessed on August 1st 2003.

Koch S., Schneider G. (2002) *Effort, co-operation and co-ordination in an Open Source software project: GNOME*. Information System Journal, 12, pages 27-42.

Krishnamurthy S. (2002) *Cave or community? An empirical examination of 100 mature Open Source projects*. First Monday, Peer-reviewed Journal on the Internet, http://firstmonday.org/issues/issue7_6/krishnamurty/, accessed on May13th 2003.

Krugman P. (1996) *The self-organizing economy.* Backwell, New York, NY, USA.

Lanzara G. F., Morner M. (2003) *The Knowledge Ecology of Open-Source Software Projects*.
Lerner J., Tirole J. (2001) *The Open Source movement: key research questions*. European Economic Review, 45, pages 819-826.

Lerner J., Tirole J. (2002a) *Some simple economics of the Open Source*. The Journal of Industrial Economics, 2 (L), 197-234.

Lerner J., Tirole J. (2002b) *The Scope of Open Source Licensing*. MIT Working Paper. http://opensource.mit.edu/papers/lernertirole2.pdf, accessed on August 2nd, 2003.

MIT Working Paper, http://opensource.mit.edu/papers/lanzaramorner.pdf, accessed on 1st August 2003.
Mitsubishi Research Institute (2003) *Linux white paper 2003*. http://oss.mri.co.jp/, accessed on May 12th 2003.

Mockus A., Fielding R., Herbsleb J. (2000) *A case study of Open Source software development: the Apache server*. In Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland, 263-272.

Mortali M. (2002) *Market Opportunity Analysis For Open Source Software Management Summary.* http://www.openforumeurope.org/research/market_analysis_for_open_source_software/, accessed on May 12th 2003.

Mustonen M. (2002) *Why do firms supported the development of substitute copyleft programs*. MIT Working Paper,. http://www.valt.helsinki.fi/fppe/GS%20fellows/mustonen_paper4.pdf, accessed on August 2nd 2003.

Netcraf (2003) *May 2003 Web Server Survey*. http://news.netcraft.com/archives/2003/05/05/may_2003_web_server_survey.html, accessed on May 12th 2003.

O'Mahony S. (2003) *Guarding the commons: how do community managed software projects protect their work*. Research Policy, 32(7), 1179-1198.

Okuyama K., Takayasu M., Takayasu H. (1999) *Zipf's law in income distribution of the companies*. Physica A, 296, 125-131.

Raymond E. (2001) *The cathedral & the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Sebastopolous , CA, USA.

Richter C. F. (1958) *Elementary seismology*. W. H. Freeman, San Francisco, CA, USA.

Robles G., Scheider H., Tretkowski I., Weber N. (2001) *Who is doing it? A research on Libre Software developers.* Fachgebiet für Informatik und Gesellschaft TU-Berlin, http://widi.berlios.de/paper/study.html, accessed on May 15th 2003.

Schiff A. (2002) *The economics of Open Source software: a survey of the early literature*. Review of Network Economics, 1(1), 66-74.

Schweik C. M., Semenov A. (2003) *The institutional design of Open Source programming: implications for addressing complex public policy and management problems*. First Monday, Peer-reviewed Journal on the Internet, http://firstmonday.org/issues/ /issue8_1/schweik/index.html, accessed on May14th 2003.

Sendmail Inc. (2003) *A search for new heroes*. http://www.sendmail.com/company/awards, accessed on May 12th 2003.

Shankland S. (2003) IBM, Dell win in losing server market. http://news.com.com/2100-1001-985769.html, accessed on May 12th 2003.

Urzua C. M. (2000) *A simple and efficient test for Zipf's law.* Economic Letters, 66, 257-260.

von Krogh G., Spaeth S., Lakhani K. R. (2003) *Community, join, and specialization in Open Source software innovation: a case study*. Research Policy, 32(7), 1217-1241.

West J., Dedrick J. (2001) *Proprietary vs. open standards in the network era: an examination of the Linux phenomenon.* In Proceedings of the Hawaii International Conference on System Science (HICSS-34), Maui, Hawaii, USA.

Wheeler D. A. (2003) *Why Open Source Software/ Free Software (OSS/FS)? Look at the numbers.* http://www.dwheller.com/oss_fs_why.html, accessed on May 13th 2003.

Wichmann T. (2002a) *Basics of Open Source software markets and business models*. Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Wichmann T. (2002b) *Firms' Open Source activities: motivations and policy implications*. Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Zipf G. K. (1949) *Human behaviour and the principle of least effort*. Addison-Wisley, Cambridge, MA, USA.