Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2008 Proceedings

Mediterranean Conference on Information Systems (MCIS)

10-2008

FIRMS'INVOLVEMENT IN OPEN SOURCE PROJECTS: A CONTROVERSIAL ROLE

Eugenio Capra
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy, eugenio.capra@polimi.it

Chiara Francalanci
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy, francala@elet.polimi.it

Francesco Merlo
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy, merlo@elet.polimi.it

Cristina Rossi Lamastra
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy, cristina1.rossi@polimi.it

FIRMS'INVOLVEMENT IN OPEN SOURCE PROJECTS: A CONTROVERSIAL ROLE

Capra, Eugenio, Politecnico di Milano, Dipartimento di Elettronica e Informazione, via Ponzio, 34/5, 20133 Milano, IT, eugenio.capra@polimi.it

Francalanci, Chiara, Politecnico di Milano, Dipartimento di Elettronica e Informazione, via Ponzio, 34/5, 20133 Milano, IT, francala@elet.polimi.it

Merlo, Francesco, Politecnico di Milano, Dipartimento di Elettronica e Informazione, via Ponzio, 34/5, 20133 Milano, IT, merlo@elet.polimi.it

Rossi Lamastra, Cristina, Politecnico di Milano, Dipartimento di Ingegneria Gestionale, P.za Leonardo da Vinci, 32, 20133 Milano, IT, cristina1.rossi@polimi.it

Abstract

In this paper, we focus on Community OS projects with the goal of understanding whether the involvement of firms through their professional developers has an impact on the quality of the software product and on its overall success. We distinguish between two main typologies of firms' involvement: Development firms contributions and Non-development firms contributions. The paper posits that a higher percentage of code contributed by paid developers has a positive impact of project success and size. However, it also puts forward a negative impact of non-development firms contribution on software design quality. Hypotheses are tested on a sample of 643 applications from the SourceForge.net repository, corresponding to 5,335 versions. Data were collected by means of an online questionnaire and a tool developed ad hoc to calculate software design quality metrics. Empirical findings support our hypotheses. Overall, our data confirm that firms are significantly investing in OS projects and that they can play a crucial role in determining projects' success when they also take active part in code development. However, most of them are taking a short-term perspective that does not focus on quality. This may lead to higher costs and a lower user satisfaction in the long term.

Keywords: Open Source; firms' involvement; software design quality; software project success..

1 INTRODUCTION

Even though Open Source (OS) was born as a cultural movement mainly supported by volunteer developers, nowadays the largest IT and software firms are investing in OS and an increasing number of smaller firms are building their business on OS (Wasserman and Capra 2007; Riehle 2007). Eclipse has been released under an OS license, MySQL has been acquired by Sun Microsystems, and the number of projects hosted by online software repositories is steadily increasing. There exist two main categories of OS projects: *i)* Commercial OS projects, and *ii)* Community OS projects (Wasserman and Capra 2007). Firms participate in each category of projects with different business goals.

Commercial OS projects (e.g., former MySQL, EnterpriseDB, SugarCRM, Jaspersoft, Zimbra, Alfresco, Funambol) produce software released on open licenses according to the OSI standard[1] and encourage participation from the community. However, they are entirely led by a firm, which controls the access to the code base, defines the evolution strategy of the project, and sets the implementation roadmap (Capra and Wasserman 2008; Capra 2008; Fitzgerald 2006; Goldman and Gabriel 2005). These firms view OS as a new business model that leverages the community as a source of feedback, testing, and marketing, and aim at selling the whole range of services built around their software, such as customization, maintenance, training, and hosting.

[1] http://opensource.org

Community OS projects are usually hosted by online software repositories, such as *SourceForge.net*, *Apache Foundation*, or *Tigris.org*. Although there is no single firm owning the project, some or most developers can be employed by a firm and paid to take active part in the community and develop specific components of the software program according to their firm's needs (Wasserman and Capra 2007). Code contribution is a common and effective strategy for firms to contribute to OS projects, as it allows them to directly impact on the final product. However, there is a number of different typologies of involvement other than code development activities. Firms may test the software, report bugs, contribute to packaging, suggest functional requirements, write documentation, or simply participate and animate forums dedicated to the project (Goldman and Gabriel 2005). In addition to that, firms may provide financial, logistic, and marketing support, or generally coordinate the project (Fogel 2006).

In this paper, we focus on Community OS projects with the goal of understanding whether the involvement of firms, which we distinguish in development and in non-development contribution, has an impact on the quality of the software product and on its overall success.

Intuitively, the participation of a firm in a software project is likely to have an impact on the software product. Community OS projects are usually informally governed, there are no roadmaps, and development is guided by the most active contributors (Fogel 2006). If all developers are volunteer, it is almost unfeasible to set deadlines or to commit to develop specific functionalities. People contribute to fulfill personal objectives, which include a sense of satisfaction with producing a high-quality software artifact and a sense of belonging to a community of independent developers, and no formal management levers are available. On the contrary, the developers hired by a firm work because they are paid to do it and commit to the specific objectives set by their employers. Consequently, their personal motivation is completely different from that of volunteer developers. If a higher number of professional developers is involved, the project is more likely to have an explicit strategy and to be more efficient in translating strategic objectives into operating tasks. This can be expected to have an impact on the overall success of the project. In turn, when a firm is significantly involved in a project, it is likely to impose constraints, schedules, and deadlines to reach its business objectives. These constraints may have a negative effect on volunteer developers, whose objectives are usually different than those of the firm, and on the overall spirit of the project. This may have an impact on the quality of the software.

The results presented in this paper are based on data on 643 *SourceForge.net*[2] Java projects. We have conducted an online survey to measure the extent of firms' involvement in each project. We have developed an *ad-hoc* tool to analyze the Java bytecode and extract a set of standard design quality metrics widely accepted in the software engineering community. This data set has allowed us to analyze the relationship between different typologies of firms' involvement, quality, and success within a large sample of Community OS projects.

2 RELATED WORK

2.1 Firms' involvement in OS software projects

The expression "OS firm" indicates companies that have included OS products in their product mix and have integrated an OS development model in their work practices, thus reducing the distance between OS communities and business (Bonaccorsi et al. 2006). The integration of OS into a company's business model has been subject to extensive research (Koski 2007; Gehring 2006; West 2007). While a number of studies have focused on the motivations of firms' involvement in OS, only a few studies have considered the impact of firms' involvement on the output of projects. Dahlander and Wallin (Dahlander and Wallin 2006) have studied the network of relationships within a specific project, by distinguishing between hired and volunteer developers. They have discovered that hired developers have specific behaviors that often generate an initial diffidence among volunteer developers. Lerner et al. (Lerner et al. 2006), studying a sample of 100 OS projects, have found that

[2] http://www.sourceforge.net

the share of code contributed by paid developers is larger in the case of projects with a larger code base and with higher growing speed (calculated as the difference between the number of lines of code between a version and the previous one). West (West 2007) has discussed the importance of value networks and popularity, but his research has only a marginal focus on firms' involvement. Robles et al. (Robles et al. 2007) have studied corporate involvement in the Debian project.

Although a direct control over the final software product can be achieved only by performing code-writing activities, other non-developing types of involvement can be considered by firms as well in order to influence a project. Overall, the relationship between firms' involvement and project success is still an open research subject.

2.2 Software design quality

This paper focuses on the quality of software design, i.e. on the internal quality of software. Previous literature suggests that higher values of software design quality metrics represent drivers of a number of external quality variables, such as testability, correctness, and reliability (Boehm et al. 1976; Brito e Abreu and Melo 1996; Barbey et al. 1998; Marinescu 2005). In turn, these external quality variables affect user satisfaction and can influence adoption and actual usage (Bevan 1995, ISO/IEC 2004). However, the direct analysis of external quality variables, i.e. of software effectiveness variables, is outside of the scope of the present paper.

Software design quality can be measured by analyzing design properties of source code. There exists a consolidated body of literature focusing on code-based design quality metrics.

With the advent of the object-oriented programming paradigm, coupling, cohesion, inheritance, and information hiding have been identified as the basic properties of software design quality (Emerson 1984; Symons 1988; Chen and Lu 1993; Sharble and Cohen 1993). Based on these four basic properties, a number of metrics have been proposed to evaluate the design quality of object-oriented software. The most widely known metrics have been first proposed by (Chidamber and Kemerer 1994) and by (Brito e Abreu 1995). These milestone contributions have started a lively debate within the software engineering community on the consistency and generality of such metrics (Basili et al. 1996; Chidamber et al. 1998; Harrison et al. 1998; Hitz and Montazeri 1996; Rosenberg 1998). As a matter of fact, metrics addressing complexity, inheritance and coupling represent a standard and are included in most development environments, such as Eclipse and Visual Studio .NET. This paper focuses on these standard metrics.

3 RESEARCH METHOD

3.1 Research Hypotheses

Firms participate in OS projects with a number of different goals. However, it is reasonable to hypothesize that it is more likely that firms decide to contribute code to a project if they regard it as successful and, on the other hand, when they decide to be involved, they work towards success with a clear business goal. How to assess the "success" of an OS software project is highly controversial and depends on the point of view of specific classes of stakeholders (Crowston et al. 2003). However, a very common indicator of success is the *ranking* measure, as defined by *SourceForge.net*. The ranking of a project measures the web traffic generated by the project page, its communication activities (e.g., forum and bug tracking activities), and its development activities. This proxy of success is widely used as it seems reasonable to assume that a project is successful if its website counts numerous visits and downloads, if it raises the community's interest and generates lively debates, and, ultimately, if the community produces a significant quantity of code at a steady pace. Other repositories, such as *FreshMeat.net*[3], define similar indicators of success (vitality, popularity and rating). Based on these measures of success, when a firm contributes code to a project it should positively affects its ranking. Professional developers work according to a precise schedule, contrary to volunteers who work in

[3] http://freshmeat.net

their spare time and do not guarantee a continuous and consistent production of code. In addition to that, professional developers must fulfil the business objectives set by the firms that pay them, which typically include the growth of the code base and of the project's community. Code, web traffic, and communication are likely to grow, accordingly. These considerations lead to our first research hypothesis.

H1: *A higher percentage of code contributed by paid developers leads to greater success.*

In addition to making new profits, firms contribute to OS projects to improve their image and to leverage the online repositories as marketing channels (Wasserman and Capra 2007). Larger projects are more likely to have sufficient momentum to attract new contributors and new users. A firm that contributes by means of code-writing activities to a project is interested in increasing the size of the project: consequently, the code contribution of professional developers to an OS project is likely to further increase the size of the project. Professional developers write code on a continuous basis, according to the schedule agreed upon with their employer. They feel in charge of the tasks that have been assigned to them and are often eager to complete their assignments to obtain incentives or gain further contracts (Austin 2001). This leads us to our second research hypothesis.

H2: *A higher percentage of code contributed by paid developers leads to larger projects.*

OS code is commonly perceived to be of higher quality compared to proprietary code (Boulanger 2005; Fitzgerald 2004). The common perception is that the looser governance approach typical of open contexts removes the pressure of deadlines and encourages the individual motivation of developers towards the production of a unique artifact (MacCormack et al. 2006; Howison and Crowston 2005; Austin 2001). Community OS projects, i.e. projects hosted in online repositories such as SourceForge.net, are generally developed by a geographically distributed team (Monga 2004). Developers usually live and work in different parts of the world with different time zones and rarely meet in person. A recent stream of literature states that a higher design quality is necessary to enable the cooperation among distributed and looser groups of developers (MacCormack et al. 2006; Slaughter et al. 2006). For example, a modular structure of code facilitates the job allocation and the coordination among geographically distributed programmers, increases the project's transparency, and lowers the knowledge barriers to new contributions (Baldwin and Clark 2000; MacCormack et al. 2006). In addition to that, corporate structures can compensate a lower design quality and still enable development in a distributed context. For example, projects led by firms can count on professional full-time project managers who assign tasks, coordinate development, and integrate different contributions into a coherent application (Fogel 2006). According to these considerations, it is reasonable to assume that when a firm is significantly involved in a project it is likely to impose constraints, schedules, and deadlines to reach its business objectives, also if it is not directly performing code-writing activities. Such constraints may have a negative effect on volunteer developers, leading them to neglect design quality under the pressure of reaching business goals, since their personal objectives are usually different from the firm's ones. This leads us to our third research hypothesis.

H3: *A higher firms' involvement in non-development activities leads to lower software design quality.*

3.2 Variable Definition and Operationalization

This section defines the variables involved in testing our hypotheses and explains how they have been operationalized.

Software design quality. We have measured a set of traditional software design quality metrics: CBO, DIT, NOC, WMC, RFC, AIF, MIF and CC.. Two of the most referenced suites of object-oriented design metrics have been included in our metrics' set, as suggested by (Harrison et al. 1998): the MOOD metrics set for the evaluation of quality at the software system level (Brito e Abreu 1995), and the Chidamber and Kemerer metrics suite for the evaluation of quality at the class level (Chidamber and Kemerer 1994). In addition, the Cyclomatic Complexity metric (McCabe 1976) has been included as a standard reference measure for complexity.

The metrics of software quality have been evaluated by analyzing the Java bytecode of all available versions of each project in the application sample. The bytecode has been analyzed with a tool developed ad-hoc.

Firms' involvement. The evaluation of firms' involvement in the projects is operationalized with a questionnaire to be submitted to project administrators. Specific questions aim to determine whether one or more firms are involved in a project or not. In particular, we distinguished between two different typologies of firms' involvement: *Development Firms' Involvement* and *Non-Development Firms' Involvement*.

We define as *Development Firms' Involvement* the percentage of code developed by hired developers, as opposed to the code developed on a voluntary basis. We consider code to be contributed by hired developers whenever a firm: releases proprietary code to the community with an OS license; formally authorizes its employees to develop code (including writing patches and fixing bugs) for the OS project during working hours and/or asks them to do so; hires community members or external people to develop code for the project.

However, as discussed in the previous sections, firms' involvement in an OS project may not be limited to code contribution. Specific multiple choice questions have been included in the questionnaire to clarify and analyze all the possible levels and typologies of firms' involvement. From these questions, we have defined a Boolean variable labeled *Non-Development Firms' Involvement* whose value is one when an at least one firm performs or has performed in the past any of the following activities within an OS project: sponsoring the project by providing financial and logistic support; distributing the project's software; providing marketing support of the project; formally authorizing its employees to work for the OS project during working hours and/or asks them to do so; hiring community members or external contributors for testing the software, reporting, packaging, suggesting functional requirements, writing documentation, participating in forums dedicated to the project, supporting users, training, planning and designing, coordinating and managing.

Project success. Project success is operationalized as the *ranking* metric of projects as defined by the *SourceForge.net* repository. Low ranking values are an index of high success, and conversely. As explained by (Crowston et al. 2003; Howison and Crowston 2004; Crowston et al. 2006), ranking is calculated as a function of several indices, namely the number of downloads, the project page views, and the development activity of team members.

Project size. Project size is operationalized as the number of methods. Note that this metric is consistent throughout all the projects in our sample, as they are all written in Java (see the following section).

3.3 Data Sample

The data set used for this study has been built by measuring our variables on a sample of OS community applications from the *SourceForge.net* repository. Since mining online repositories (such as *SourceForge.net*) can lead to controversial results because of the varying quality of available data (Howison and Crowston 2004), the application sample has been obtained by refining an initial set of applications. The initial application set has been selected according to the following criteria: active and beta status or higher, according to the status reported by the home page of each application on *SourceForge.net*; at least 2 members, listed either as administrators or developers; written in Java. We are aware that our sample is not fully representative of the OSS world, which is indeed very wide and inhomogeneous, but we had to focus on a set of comparable projects. We chose only one online software repository to have homogenous measures for the ranking and the number of developers, and only one language in order to have homogenous design quality metrics.

Such application sample was comprised of 8,308 applications, and constituted the widest possible sample of Java applications that could be obtained from *SourceForge.net* satisfying the selection criteria listed above. The initial sample has been refined by removing the applications for which we did not receive coherent answers or were not able to compute the quality metrics. The final application sample is constituted by 643 applications, corresponding to 5,335 versions. Table 1 presents the summary statistics of the final application sample. Data on all the applications refers to June, 12th 2007 to guarantee the temporal consistency of the data set. The number of project administrators and developers that we have used is the one officially listed by *SourceForge.net*.

In order to evaluate the firms' involvement in the projects of our sample, we have conducted an online survey addressed to all the administrators of the initial application sample. We have conducted a *pre-test* with to a casual and stratified sub-group of our sample, constituted by 195 administrators. The *pre-test* phase allowed to tune the questionnaire and make sure that there were no non-response bias. The survey was conducted starting from November 27th 2007. After the first e-mails, two follow-ups were conducted, starting from December 5th 2007 and December 18th 2007, respectively.

All quality metrics were measured on the final application sample by means of an ad-hoc tool.

Table 1. Descriptive statistics of application sample.

Variable	Average	St.Dev.	Min	Max
Projects	643			
Versions per project	8.30	13.97	1	157
Administrators	2.23	1.63	1	15
Developers	4.53	11.00	0	132
Team members	6.76	11.90	2	141
Project age (months)	40.24	23.42	0.10	90.40
Project ranking	22,078	23,171	25	114,427
SF Activity percentile	87.89	12.51	52.91	99.99
Months per version	15.44	18.20	0.05	89.03

4 EMPIRICAL RESULTS

4.1 Qualitative analyses of firms' involvement in Open Source projects

216 out of 643 projects (34%) have declared that one or more firms somehow involved in their project. This confirms the great importance of the role played by firms not only in Commercial OS projects, but also in Community OS projects, such as those hosted on *SourceForge.net*.

On average, when firms participate in an OS project, they contribute more than half of the code (57%). If the same percentage is computed for the code of all the projects in the sample (i.e. including the projects with no firms involved), approximately 20% of the code is contributed by hired developers.

4.2 Measurement Model

The measurement model has been defined to verify the validity of the metrics used to assess software design quality. In particular, a principal component analysis (PCA) has been performed on the set of metrics discussed in the Variable Definition and Operationalization section to verify that our metrics address different software quality attributes, namely complexity and inheritance.

Figure 1 presents the measurement model of the complexity metrics. One factor has been extracted and labeled as *Complexity*. This means that, according to our data, WMC, RFC, and CC can be interpreted as the observable components of the same latent (i.e., underlying and unobserved) variable. Figure 2 presents the measurement model of the metrics of inheritance. In this case, one factor has been extracted, and labeled as *Inheritance*.

Table 2 shows the results of PCA along with the standardized regression weights of the relationships between latent and observed variables. Results show that all the factorizations can be accepted, since all the values of the composite factor reliability are greater than the threshold value of 0.70, as suggested by (Bagozzi and Yi 1988; Fornell and Larcker 1981). All the relationships considered between observed and latent variables are significant with $p < 0.001$. This confirms that the factorizations in the measurement model are meaningful. In this particular case, the results of factorization are consistent with the literature on software design quality (Chidamber and Kemerer 1994; Brito e Abreu 1995), which indicates complexity and inheritance as two important measures of software design quality.

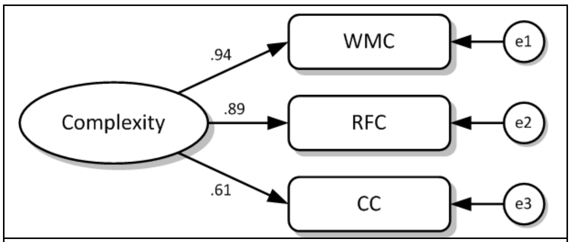

Figure 1. Measurement model of complexity.

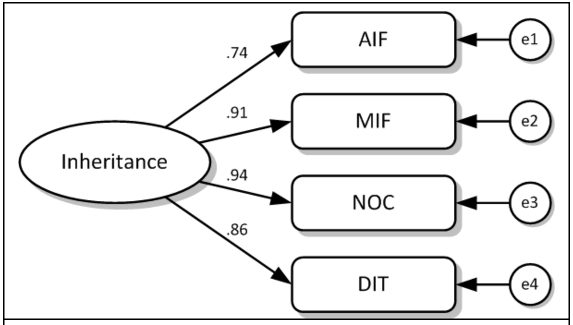

Figure 2. Measurement model of inheritance.

Table 2. PCA results and estimates of regression weights for the measurement models of Figure 1 and Figure 2.

Dependent Variable	Independent Variable	Standardized Regression Weight (b)	Standard Error	p-value	Composite Factor Reliability
WMC	Complexity	0.942	-	-	0.925
RFC	Complexity	0.895	0.071	< 0.001	0.917
CC	Complexity	0.608	0.171	< 0.001	0.783
AIF	Inheritance	0.736	-	-	0.827
MIF	Inheritance	0.907	0.069	< 0.001	0.933
NOC	Inheritance	0.939	0.059	< 0.001	0.938
DIT	Inheritance	0.861	0.124	< 0.001	0.885

4.3 Structural Model Testing

We have used AMOS 7 (Arbuckle 2006) to analyze the research model by means of Structural Equation Modeling (SEM). Structural Equation Modeling is a common statistical technique for testing and estimating causal relationships (Kaplan 2000; Kline 2004). SEM allows latent variables (*Complexity* and *Inheritance* in our case) within a model and can verify research hypotheses that involve latent variables. In contrast, simple multivariate regression requires to compute actual values for all variables.

Figure 3 presents the research model that we have used to test our research hypotheses. The estimation results of the research model of Figure 3 are reported in Table 3. For sake of simplicity, the model in Figure 3 does not show the factorizations leading to the latent variables *Complexity* and *Inheritance* discussed in the previous section. Project age and size have been taken into account as controlling variables, as suggested by (Banker and Slaughter 2000). Although Figure 3 does not show the controlling variable *Age* and the controlling relationships of *Size*, they have been considered in the model, as reported in Table 3. Please note that *Age* does not significantly affects *Complexity*, as shown by the *p*-value ($p = 0.441$). Further, *Development Firms' Involvement* and *Non-Development Firms' Involvement* are only very weakly correlated. This shows that several firms are involved in OS projects without directly participating to code development activities.

All the relationships hypothesized between model variables are significant with $p \leq 0.05$, that is, can be accepted at a significance level $\alpha = 95\%$. In order to assess the overall fit of the model, several indices have been considered, as suggested by (Kline 2004). All goodness-of-fit indices of the SEM estimation indicate that the overall model fit is satisfactory: IFI is 0.835, close to 0.90 as suggested by (Bollen 1989), NFI is 0.807, close to 0.90 as suggested by (Bentler and Bonnet 1980), CFI is 0.833, close to 0.90 as suggested by (Bentler 1990), and RMSEA is 0.08 thus satisfying the criterion suggested by (Browne and Cudeck 1993). Alternative models that reversed causal relationships were non-significant or less significant than the model in Figure 3. We also tested the relationships between *Development Firms' Involvement* and *Complexity* and *Inheritance*, and between *Non-Development Firms' Involvement* and *Size* and *Ranking*, but they all resulted non-significant.

The results of the statistical analyses thus confirm all our hypotheses.

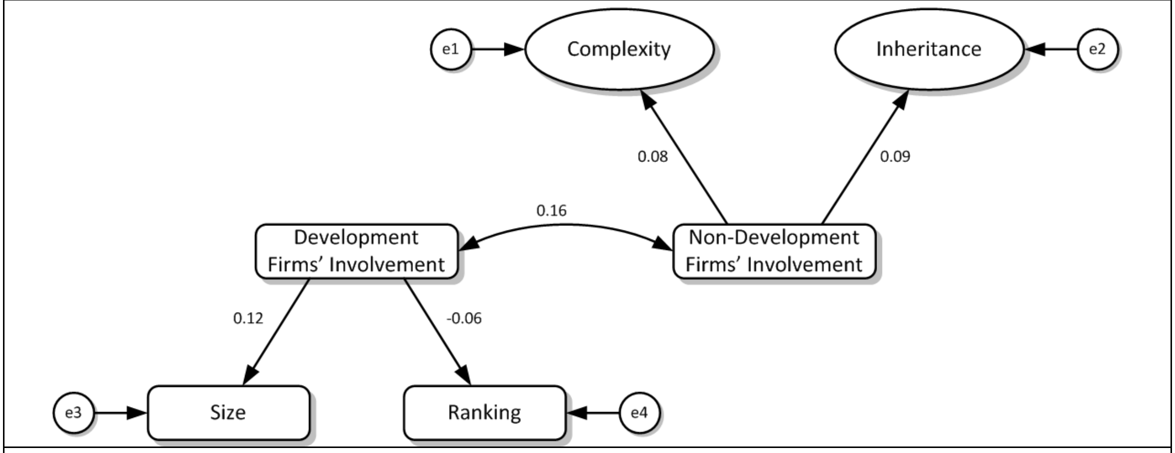
e1
Complexity
Inheritance
e2
0.08
0.09
0.16
Development Firms' Involvement
Non-Development Firms' Involvement
0.12
-0.06
e3
Size
Ranking
e4

Figure 3. Structural model for the verification of research hypotheses.

Table 3. Estimates of regression weights for the research model of Figure 3.

Dependent Variable	Independent Variable	Standardized Regression Weight (b)	Standard Error	*p*-value
Ranking	Dev. Firms' Involvement	-0.064	1,775.748	0.040
Size	Dev. Firms' Involvement	0.121	438.431	0.055
Complexity	Non-dev. Firms' Involvement	0.078	0.327	0.049
Inheritance	Non-dev. Firms' Involvement	0.091	0.010	0.017
Complexity	Age	-0.030	0.000	0.441 (n.s.)
Inheritance	Age	0.111	0.000	0.004
Size	Age	0.089	0.159	0.024
Complexity	Size	0.267	0.000	< 0.001
Inheritance	Size	0.327	0.000	< 0.001

5 DISCUSSION AND CONCLUSIONS

Empirical results support hypothesis H1, thus confirming that firms' involvement in code developing activities leads to more successful OS projects. Whereas previous literature has already shown the impact of firms' involvement on social networking metrics, bug fixing speed, number of developers and popularity (Grewal et al. 2006; Crowston et al. 2006), no clear empirical evidence has yet been proposed to analyze its impact on the overall success of the project.

Results also support hypothesis H2, thus confirming firms' involvement as a fundamental lever to reach a critical project size, which is essential to gain visibility and to fully exploit OS communities as a new and broad marketing platform (Wasserman and Capra 2007).

Findings also support hypothesis H3, thus indicating a negative effect of firms' involvement in non-development activities on software design quality. The involvement of a firm supporting the evolution and development of an OS project can influence governance models, missions and objectives, shifting from a "community-driven" project to a more traditional approach (Capra 2008). As a consequence, a higher working pressure is put on developers, deadlines become more strict and development activities are performed within a shorter time frame to meet corporate goals and preserve the project's leading position.

Our results hint that firms take part in OS projects to reach corporate objectives. These objectives include the success of the project and its visibility. Our results suggest that these objectives can be reached only if firms actively participate to development activities, as this is the most direct and effective way to influence and impact on the final product. On the other hand, these corporate objectives seem to not include quality.

Overall, our data confirm that firms are significantly investing in OS projects and that they may play a crucial role for projects' success. However, most of them are taking a short-term perspective that does not focus on quality. This may lead to higher costs and loss of user satisfaction in the long term. Based on our findings, we recommend that firms *i)* actively participate to code-development activities to directly impact on the success of the final product, and *ii)* put into action all the possible measures to equal the high software design quality favored by OS contexts.

References

Albert, R. and Barabàsi, A. L. "Emergence of scaling in random networks", Science, vol. 286, no. 5439, pp. 509-512, 1999.

Arbuckle, J.L. Amos version 7, Small Waters Corp., Chicago, Illinois, 2006.

Austin, R. D. "The effects of time pressure on quality in software development: an agency model", Information Systems Research, vol. 12, no. 2, pp. 195-207, 2001.

Bagozzi, R. P. and Yi, Y. "On the evaluation of structural equation models", Journal of the Academy of Marketing Science, vol. 16, no. 1, pp. 74-94, 1988.

Baldwin, C. Y., and Clarck, K. B. Design Rules vol. 1 – The power of modularity. MIT Press, Cambridge, MA, 2000.

Banker, R. D., and Slaughter, S. A. "The Moderating Effects of Structure on Volatility and Complexity in Software Enhancement", Information Systems Research, vol. 11, no. 3, pp. 219-240, 2000.

Barbey, S., Buchs, D. and Peraire, C. "Modelling the production cell case study using the fusion method", Technical Report 98/208, EPFL-DI, 1998.

Basili, V. R., Briand, L. C. and Melo, W. L. "A validation of object-oriented design metrics as quality indicators", IEEE Transactions on Software Engineering, vol. 22, no. 10, pp. 751–761, 1996.

Bentler, P.M. "Comparative fit indexes in structural models", Psychological Bulletin, vol. 107, pp. 238–246, 1990.

Bentler, P.M. and Bonnett, D.G. "Significance tests and goodness of fit in the analysis of covariance structures", Psychological Bulletin, vol. 88, pp. 588–606, 1980.

Bevan, N. "Usability is Quality of Use", Proceedings of International Conference on Human Computer Interaction, 1995.

Boehm, B., Brown, J. R. and Lipow, M. "Quantitative evaluation of software quality", Proceedings of International Conference on Software Engineering, pp. 592–605, 1976.

Bollen, K.A. "A new incremental fit index for general structural equation models", Sociological Methods and Research, vol. 17, pp. 303–316, 1989.

Bonaccorsi, A., Giannangeli, S. and Rossi, C. "Entry Strategies Under Competing Standars: Hybrid Business Models in the Open Source Software Industry", Management Science, vol. 52, no. 7, pp. 1085-1098, 2006.

Bonaccorsi, A. and Rossi, C. "Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business, Knowledge", Technology and Policy, vol. 18, no. 4, pp. 40-64, 2006.

Boulanger, A. "Open Source versus proprietary software: is one more reliable and secure that the other?", IBM Systems Journal, vol. 44, no.2, p. 239, 2005

Brito e Abreu, F. "The MOOD metrics set", Proceedings of ECOOP Workshop on Metrics, 1995.

Brito e Abreu, F., Melo, W. "Evaluating the Impact of Object-Oriented Design on Software Quality," Proceedings of METRICS, p. 90, 1996.

Browne, M.W. and Cudeck, R. "Alternative ways of assessing model fit", in Bollen, K.A. and Long, J.S. (eds.) Testing structural equation models, Sage, Newbury Park, CA, pp. 136–162, 1993.

Capra, E. "Software design quality and development effort: an empirical study on the role of governance in Open Source projects", Ph.D. Dissertation, Dept. of Electronics and Information, Politecnico di Milano, Italy, 2008.

Capra, E., Wasserman, A.I. "A framework for evaluating managerial styles in Open Source projects", Proceedings of Open Source Systems Conference, 2008 .

Chen, J. Y. and Lu, J. F. "A new metric for object-oriented design", Journal of Information Systems and Software Technology, vol. 35, no. 4, pp. 232-240, 1993.

Chidamber, S. and Kemerer, C. "A metrics suite for object oriented design", IEEE Transactions on Software Engineering, vol. 20, no.6, pp. 476–493, 1994.

Chidamber, S. R., Darcy, D. P., and Kemerer, C.F. “Managerial use of metrics for object-oriented software: An exploratory analysis”, IEEE Transactions on Software Engineering, vol. 24, no.8, pp. 629–639, 1998.

Crowston, K., Annabi, H. and Howison, J. “Defining Open Source Software Project Success”, Proceedings of International Conference on Information Systems, Seattle, 2003.

Crowston, K., Howison, J. and Annabi, H. “Information systems success in free and open source software development: Theory and measures”, Software Process: Improvement and Practice, vol. 11, no. 2, pp. 123-148, 2006.

Dahlander, L. and Wallin, M. W. “A man on the inside: unlocking communities as complementary assets”, Research Policy, vol. 35, no. 8, pp. 1243-1259, 2006.

Emerson, T. J. “A discriminant metric for module comprehension”, Proceedings of International Conference on Software Engineering, pp. 294-431, 1984.

Fitzgerald, B. “A critical look at Open Source”, IEEE Computer, vol. 37, no. 7, pp. 92-94, July 2004.

Fitzgerald, B. “The transformation of Open Source software”, MIS Quarterly, vol. 30, no. 2, pp. 587-598, 2006.

Fogel, K. Producing Open Source software, O’Reilly, Sebastopol (CA), 2006.

Fornell, C. and Larcker, D. F. “Evaluating structural equation models with unobservable variables and measurement errors: Algebra and statistics”, Journal of Marketing Research, vol. 18, no. 3, pp. 383-388, 1981.

Gao, Y. and Madey, G. “Network analysis of the SourceForge.net community”. In Open Source Development, Adoption and Innovation, Springer Boston, 2007.

Gehring, R. A. “The institutionalization of Open Source”, Poiesis & Praxis: International Journal of Technology Assessment and Ethics of Science, vol. 4, no. 1, pp. 1-76, 2006.

Grewal, R., Lilien, G. L., and Mallapragada, G. "Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems", Management Science, vol. 52, no. 7, pp. 1043-1056, 2006.

Harrison, R., Counsell, S., and Nithi, R. “An evaluation of the MOOD set of object-oriented software metrics”, IEEE Transactions on Software Engineering, vol. 24, no. 6, pp. 491-496, 1998.

Hitz, M. and Montazeri, B. “Chidamber and Kemerer’s metrics suite: A measurement theory perspective”, IEEE Transactions on Software Engineering, vol. 22, no. 4, pp. 267–271, 1996.

Howison, J. and Crowston, K. “The perils and pitfalls of mining SourceForge”, Proceedings of International Workshop on Mining Software Repositories, pp. 7–12, 2004.

Howison, J. and Crowston, K. “The social structure of free and open source software development”, First Monday, vol. 10, no. 2, 2005.

Kaplan, D. “Structural Equation Modeling: foundations and extensions”, Advanced Quantitative Techniques in the Social Sciences series, vol. 10, 2000.

Kline, R. B. Principles and practice of structural equation modeling, Second Edition, Guilford Press, NY, 2004.

Koski, H. “Private-collective software business models: coordination and commercialization via licensing”, Review of Economic Research on Copyright Issues, vol. 4, no. 1, pp. 47-61, 2007.

Lerner, J., Pathak, P. and Tirole, J. “The Dynamics of Open-Source Contributors”, American Economic Review, vol. 96, no. 2, pp. 114-118, 2006.

Lin, Y. W. Understanding social dynamics in the collaboration between firms and FLOSS communities: conceptual and methodological issues, in Economic perspectives on Open Source software. Intellectual property, knowledge-based communities, and the software industry, pp. 174-187, Franco Angeli, Milan, 2006.

MacCormack, A., Rusnak, J., and Baldwin, C.Y. “Exploring the structure of complex software designs: an empirical study of Open Source and proprietary code”, Management Science, no. 52, pp. 1015-1030, July 2006

Marinescu, R. “Measurement and Quality in Object-Oriented Design”, Proceedings of International Conference on Software Maintenance, 2005.

McCabe, T. J. “A complexity measure”, Proceedings of International Conference on Software Engineering, p. 407, 1976.

Monga, M. “From Bazaar to Kibbutz: how freedom deals with coherence in the Debian project”, Proc. Workshop Open Source Software Engineering, pp. 71-75, 2004.

Riehle, D. “The Economic Motivation of Open Source Software: Stakeholder Perspectives”, IEEE Computer, vol. 40, no. 4, pp. 25-32, 2007.

Robles, G., Dueñas, S., and González-Barahona, J.M., “Corporate Involvement of Libre Software: Study of Presence in Debian Code over Time”, Proc. Of Open Source Systems Conference, pp. 121-132, 2007

Rosenberg, L. H. Applying and interpreting object oriented metrics, Technical Report, Software Assurance Technology Center NASA SATC, April 1998.

Sharble, R. C. and Cohen, S. S. “The Object-Oriented brewery: a comparison of two Object-Oriented development methods”, ACM SIGSOFT Software Engineering Notes, vol. 18, no. 2, pp. 60-73, 1993.

Slaughter, S.A., Roberts, J. and Hann, I. “Communication networks in an Open Source software project”, Proc. Conf. on Open Source Systems, Italy, June 2006.

Symons, C. R. “Function Point analysis: difficulties and improvements”, IEEE Transactions on Software Engineering, vol. 14, no. 1, pp. 2-11, 1988.

Wasserman, A.I., and Capra, E. “Evaluating software engineering processes in commercial and community Open Source projects”, Proc. Int’l Workshop Emerging Trends in FLOSS Research and Development, 2007.

West, J. “Value Capture and Value Networks in Open Source Vendor Strategie”, Proceedings of Hawaii International Conference on System Sciences, 2007.

Xu, J., Christley, S., and Madey, G., “Application of Social Network Analysis to the Study of Open Source Software” in The Economics of Open Source Software Development, Bitzer, J. and Schröder, P. J. H. (eds.), Elsevier B.V., 2006.