Paper to be presented at the DRUID 2011

on

INNOVATION, STRATEGY, and STRUCTURE -
Organizations, Institutions, Systems and Regions
at
Copenhagen Business School, Denmark, June 15-17, 2011

ORGANIZING FOR COLLABORATING WITH COMMUNITIES OF USERS AND DEVELOPERS: THE CASE OF FIRMS DOING BUSINESS WITH OPEN SOURCE SOFTWARE COMMUNITY

Cristina Rossi-Lamastra

Management, Economics, and Industrial Engineering
cristina1.rossi@polimi.it

Massimo Colombo
Politecnico di Milano
Dipartimento di Ingegneria Gestionale
massimo.colombo@polimi.it

Evila Piva

evila.piva@polimi.it

Abstract
This note presents our investigation of how firms organize to collaborate with communities of users and developers. We studied the antecedents of the adoption by software firms of a specific but widely diffused organizational practice, i.e., authorizing programmers to participate autonomously during working hours in Open Source software (OSS) projects of their choice. We claim that the probability of the adoption of this practice is higher when the innovation inputs that programmers obtain from the OSS community are more valuable to the firm and when the firm has less experience with collaborations with the community. The probability is lower when the opportunity costs of programmers? time are greater. Econometric results using data from 319 European OSS firms support our hypotheses.

Jelcodes:L29,-

ORGANIZING FOR COLLABORATING WITH COMMUNITIES OF USERS AND DEVELOPERS:
THE CASE OF FIRMS DOING BUSINESS WITH OPEN SOURCE SOFTWARE COMMUNITY

Abstract. This note presents our investigation of how firms organize to collaborate with communities of users and developers. We studied the antecedents of the adoption by software firms of a specific but widely diffused organizational practice, i.e., authorizing programmers to participate autonomously during working hours in Open Source software (OSS) projects of their choice. We claim that the probability of the adoption of this practice is higher when the innovation inputs that programmers obtain from the OSS community are more valuable to the firm and when the firm has less experience with collaborations with the community. The probability is lower when the opportunity costs of programmers' time are greater. Econometric results using data from 319 European OSS firms support our hypotheses.

JEL: O31, D8, L17

Keywords: firm-community collaborations, Open Source software, organizational practices, external knowledge sourcing.

1. Introduction

Communities of users and developers are currently a relevant source of new technological knowledge (Hargrave and Van de Ven, 2006). As is widely acknowledged, the Open Source software (OSS) community is the largest of these communities. Although the OSS community was born out of an ideological movement (Stallman, 1984), its economic importance is clear. OSS programs such as the Linux operating system or the Apache Web server successfully compete with similar proprietary solutions. Moreover, more and more firms in the software sector provide their customers OSS-based solutions (*OSS firms*, Bonaccorsi et al., 2006). These firms often sponsor existing OSS projects[1] (Ghosh et al., 2008; Capra et al., 2010), pay their employees to work on them (Hars and Ou, 2002), launch and coordinate new OSS projects (O'Mahony and West, 2008), and release the code under OSS licenses, making the code available to the OSS community (Henkel, 2009). Firms also increasingly in-source knowledge produced by OSS projects in their own innovation activity (Pisano, 2006, p. 1122; Stam, 2009).

Firm-OSS community collaborations are being given increasing attention by management scholars. A growing body of studies has analyzed how members of OSS projects internally organize OSS software development processes to collaborate with firms (e.g., O'Mahoney and Bechky, 2008). However, to the best of our knowledge, the issue of how firms design their organization to take advantage of collaborations with the OSS community has remained largely unexplored.[2] This note makes a step in this direction.

For reasons that will be explained in the following section, collaborations with the OSS community pose serious organizational challenges for firms, challenges that require the adoption of suitable organizational practices to overcome. We focus attention here on a specific organizational practice and study the antecedents of its adoption. The practice consists of authorizing a firm's programmers to contribute[3] autonomously during working hours to OSS projects other than those to which firms contribute on their behalf. Hereafter, we refer to the former projects as *external OSS projects*. This practice is widely used by

[1] An OSS project is "any group of people developing software and providing their results to the public under an Open Source license" (Evers, 2000). OSS projects are generally made available online in so-called repositories. Repositories are dedicated Web sites that also provide a suitable environment for software development and interaction between programmers.

[2] Another stream of literature has analyzed the strategic issues that firms face in creating and mobilizing communities. For instance, Miller et al. (2009) have examined the participation of firms in online communities of users as a means of enhancing demand for their products.

[3] We use the phrase "contribute to OSS projects" to include the following activities: releasing code to projects, modifying community-developed code and integrating it back into the projects, writing documentation, debugging code and answering technical questions raised by users through the mailing lists associated with the OSS projects.

OSS firms, as will be documented in the following sections. The rationale for firms to adopt this practice is to identify and assimilate from the OSS common pool (Bonaccorsi and Rossi, 2006) technical artifacts (i.e., software code) and programming competences, inputs that are valuable to the firm's own innovation activity. Obtaining these inputs would be impossible without granting programmers this freedom. The reason is that a firm's programmers possess specific knowledge (Jensen and Meckling, 1992) of the OSS community. Authorizing programmers to contribute autonomously to external OSS projects allows them to use this knowledge in collaborations with the community, making the collaborations more effective.[4] We hypothesized that the probability of the adoption of this organizational practice is higher when the value to the firm of the inputs that firm's programmers can in-source from the OSS community is greater and when the firm has less experience in collaborating with the community. Conversely, this probability is lower when the opportunity costs of the programmers' time are greater. In the empirical section of the paper, we used data from a sample of 319 European software firms to provide empirical evidence in support of the above arguments.

The paper proceeds as follows. In the next section, we present the conceptual framework and develop the theoretical hypotheses. Then we describe the dataset used in the empirical analysis. In the subsequent section, we describe the econometric methodology and variables used in the estimates. The presentation of the econometric results follows. In the concluding section, we discuss these results and indicate directions for further research.

2. Conceptual background and research hypotheses

2.1. The organizational challenges of collaborating with the OSS community

Collaborating with the OSS community is of great importance to the innovation activity of OSS firms. Through this collaboration, firms pick out software code and modules from the OSS common pool and integrate them into their own solutions (see Bonaccorsi and Rossi, 2003; Dahlander and Magnusson, 2005; Dahlander, 2007; among the many others). Firms also find new uses for their own software code and talent-

[4] This idea dates back to Hayek's (1945) seminal work on the use of knowledge in society. "If we … agree that the economic problem of society is mainly one of rapid adaptation to changes in particular circumstances of time and place, … decisions must be left to the people who are familiar with these circumstances, who know directly of the relevant changes and of the resources immediately available to meet them. …We must solve it by some form of decentralization" (Hayek, 1945, p. 524; see also the extensive discussion of this issue in Jensen and Meckling, 1992). In our specific case, decisions relate to the selection of the OSS projects to which programmers autonomously contribute, the choice of the technological problems they tackle, and of the OSS project members with whom they collaborate.

scout the best OSS programmers to collaborate with or to recruit (Eilhard, 2008). Nonetheless, collaboration with the OSS community poses major challenges for OSS firms because of the peculiar institutional framework of the community.

First, the OSS community is by definition *open*: everyone can join and freely contribute to the OSS common pool. Currently, thousands of OSS projects receive contributions from a large number of members from all around the world (Raymond, 2001). The technical skills of these individuals and the quality of their contributions are highly variable. Moreover, a large body of literature has acknowledged that these contributors have heterogeneous motives (see von Krogh et al., 2009, for a comprehensive review). Some OSS project members may be eager to signal their talent through their OSS coding activity to obtain better jobs (Lerner and Tirole, 2002), while others may develop software code just as a hobby (Hertel et al., 2003, Lakhani and Wolf, 2005). Programmers' motivations are likely to influence the quality of the software code that is produced, but these motivations are unobservable to the OSS projects' outsiders.

Second, software development in OSS projects is not governed by contracts. No enforceable agreement among project members specifies what the final output of a project should be. Project members autonomously decide the amount and content of their efforts within the project. In some cases, they may receive monetary compensation for their OSS development activities, as frequently happens in OSS projects sponsored by companies (Hars and Ou, 2002). However, the project members are not *employees of the project* (O'Mahony, 2003, p. 1179). Hence, it is difficult to determine whether and when they will carry out their software development activities. Project discontinuation and departure from the initial specifications are concrete risks (Feller and Fitzgerald, 2002). Finding trustworthy OSS development teams that are committed to reliably carrying out software development is quite difficult.

Third, although each OSS project has its own administrators who lead the project, being in contact with them might not be enough to become aware of the quality of the software produced within the project and of its future development directions. In OSS projects, leadership usually emerges from the bottom and is frequently challenged by the most active participants. These participants play a leading role in the project and may possess relevant information on the software development process that is unknown even to the project's administrators. Detecting who these lead participants are is not an easy task for project outsiders.

Fourth, the OSS community was originally shaped by the ideological concerns of fighting for software freedom (Raymond, 2001). If these concerns are still prominent among the members of an OSS project, they will not be keen on collaborating with firms. To *guard their commons* (O'Mahony, 2003, p. 1179) from external meddling by for-profit entities, OSS project members may adopt an esoteric style of software documentation, refuse to answer questions posted by firms on project mailing lists, or prohibit other project members from participating in software development projects unless they are released under the GNU General Public License (GNU GPL). If these decisions are replicated over time, they may acquire the status of *unwritten norms,* i.e., un-codified rules that govern the behaviors of project members (Ågerfalk and Fitzgerald, 2008). Awareness of and compliance with these unwritten norms is mandatory in collaborating with the OSS community. Firms ignoring these norms may waste time and effort in an attempt to participate in projects that do not welcome firms. Moreover, a firm that violates these norms in contributing to an OSS project jeopardizes further contributions to the project and may even experience difficulties in contributing to other projects in which members of the focal project participate.

Because of the aforementioned challenges, an OSS firm that aims to collaborate successfully with the OSS community must adopt suitable organizational practices. An organizational practice that can render OSS community collaborations more effective consists of granting a firm's programmers the autonomy to contribute to *external OSS projects* of their choice during working hours. Firms' programmers possess context-specific non-reproducible knowledge of the OSS community, as they act as *insiders* within that community. The key benefit for firms in adopting this organizational practice is that it allows the firms to use this *specific knowledge* (Jensen and Meckling, 1992) in decisions relating to community collaborations.

As part of their daily activities, programmers work on OSS projects to which their firm contributes on the firm's behalf. Moreover, plenty of anecdotal evidence indicates that these programmers frequently participate in OSS projects in their spare time. Because of their frequent interactions with the OSS community, they likely know the community better than their boss, who usually does not directly join OSS projects because his or her time is devoted to running the firm's operations (Cabiddu et al., 2010). Hence, programmers are in an advantageous position to locate external OSS projects from which software code that is valuable to the firm can be in-sourced, to single out talented programmers and link them to the firm, and to make sense of projects' internal dynamics and future evolution. In addition, because of their inside experience of OSS projects, programmers develop an awareness of the presence of projects' unwritten norms of behaviors and learn how to cope with them. Letting programmers autonomously contribute to external OSS projects allows firms to leverage their information advantage, thereby effectively *bridging* the firm to the OSS community (Henkel, 2009, p. 436).

In principle, OSS firms could ask their programmers to transfer the information they possess about the OSS community to their boss, who could then give them orders on how to collaborate with the OSS community. In this situation, programmers have no autonomy. In practice, this organizational arrangement is not viable. The knowledge of the OSS community possessed by programmers has been developed through their context-specific experience and daily practice. It is primarily tacit and thus difficult to transmit (Allen, 1977). Therefore, the most effective way of using this knowledge to in-source valuable innovation inputs from the OSS community is to authorize programmers to contribute autonomously to external OSS projects during working hours.[5]

Obviously, this organizational practice has a cost: the opportunity cost of the programmers' time. While contributing to external OSS projects, programmers cannot carry out their usual tasks. With everything else being equal, labor costs per unit of output will increase. We assume herein that an OSS firm will adopt the organizational practice under investigation if the expected (marginal) benefits of such a choice exceed the expected (marginal) costs.

2.2. Theoretical hypotheses

In this section, we will consider factors that influence the extent of the benefits and costs of granting firms' programmers autonomy to contribute to *external OSS projects* of their choice during working hours. In particular, we will focus on the following: i) the diversification of the OSS activities undertaken by the firm; ii) the firm's experience in collaborating with the OSS community; and iii) the type of tasks carried out by OSS firms.

The diversification of the OSS activities undertaken by the firm. The benefit of adopting the organizational practice under investigation is greater the higher the probability that, while collaborating with the OSS

[5] The above reasoning echoes arguments made under modern delegation theory (e.g., Dessein 2002, Alonso and Matouscek 2008). In this literature, it is argued that when an individual employee enjoys an information advantage over his or her boss and the underlying information is "soft" (i.e., it cannot be verified by third parties, for instance because of its tacit nature), the delegation of decision authority (in our case, over collaboration with the OSS community) is generally more efficient than its centralization.

community, a firm's programmers will identify software code, modules, and programming competences that the firm can profitably use in its own innovation activity. This probability, in turn, increases with the diversification of the OSS activities that are undertaken by the firm. OSS firms substantially differ from one another in this respect (Bonaccorsi et al., 2006). Some of them commercialize pre-packaged OSS products and provide complementary services (e.g., installation services, training on advanced functionalities, and compilation of user manuals). Other firms adapt OSS code freely downloaded from the Web to their customers' needs. They may also integrate different OSS modules to develop customized solutions. Last, some firms develop innovative software products from scratch and release them on the market under OSS licenses. Of course, any combination of these activities is possible. If a firm is involved in a more diversified set of OSS activities, it is more likely that its programmers, while contributing to external OSS projects, will absorb innovation inputs that are valuable to the firm. Therefore, we put forward Hypothesis H1:

> *H1: All else being equal, the probability that an OSS firm authorizes its programmers to contribute autonomously to external OSS projects during working hours is higher if the firm is involved in a more diversified set of OSS activities.*

A firm's experience in collaborating with the OSS community. We argued above that the specific knowledge of the OSS community possessed by firms' programmers is the key driver of the decision of an OSS firm to authorize them to contribute freely to external OSS projects during working hours. This knowledge is more valuable when the firm has less experience collaborating with the OSS community and when the information advantage of a firm's programmers over their boss is greater. Conversely, a firm that has already contributed to many OSS projects is likely to face lower obstacles in collaborating with the OSS community. The firm is likely to have already identified the best OSS programmers and the most valuable projects. Moreover, the firm is also likely to have already become aware of the most widespread unwritten norms of the OSS community and to be able to cope effectively with them (Osterloh and Rota, 2007). Under these circumstances, the use of the specific knowledge of the OSS community possessed by a firm's programmers does not render the collaboration of the firm with the OSS community more effective. Hypothesis H2 follows:

> *H2: All else being equal, the probability that an OSS firm authorizes its programmers to contribute autonomously to external OSS projects during working hours is higher when the firm has less*

experience collaborating with the OSS community.

The type of tasks carried out by OSS firms. The drawback of authorizing firms' programmers to contribute to external OSS projects of their choice during working hours is that the time the programmers devote to this activity obviously cannot be used for other tasks. The greater the opportunity cost of programmers' time, the greater the cost of granting them this freedom. The opportunity cost of programmers' time depends on the type of tasks carried out by firms. The opportunity cost is greater for OSS firms that carry out high-value-added tasks such as software code development than it is for firms involved only in writing documentation and fixing software bugs. We then derive Hypothesis H3 as follows:

H3: All else being equal, the probability that an OSS firm authorizes its programmers to contribute autonomously to external OSS projects during working hours is lower if the firm carries out high-value-added tasks.

3. The sample

The theoretical hypotheses illustrated in the previous section were tested using data from the ELISS database. This database provides information on the structural characteristics, OSS activities, and OSS community collaborations from 361 OSS firms.[6] Data were collected through a structured questionnaire administered to the owner-managers and chief technology officers of a random sample of 6,000 firms operating in the software sector (NACE code 72.2) and located in five European countries (Finland, Germany, Italy, Portugal, and Spain). The sample is stratified by firm size and geographical area (NUTS2 level). Among many other questions, firms were asked whether they contribute to OSS projects. This section of the questionnaire also included the following question: *Does the firm authorize its programmers to contribute autonomously during working hours to OSS projects other than those to which the firm contributes on its behalf?*[7]

[6] ELISS was developed in 2004 by the Laboratory of Economics and Management at the Sant'Anna School of Advanced Studies within the CIPR project of the PRIME Network of Excellence, which was funded by the European Community within the Sixth Framework Program (see Bonaccorsi et al., 2005, for a comprehensive description). The authors are indebted to Andrea Bonaccorsi for allowing them to use the database.

[7] Note that an OSS firm can decide not to contribute to OSS projects (Bonaccorsi et al., 2006). In this case, the firm simply downloads the open code freely available on the Internet and adapts it to its customers' needs without releasing the code and integrating the technical knowledge back into the community (i.e., the OSS firm adopts a *taker* rather than a *giver strategy*, Bonaccorsi and Rossi, 2006). The last question was not posed to taker firms. As these firms do not contribute to OSS projects, it is very unlikely that they allow programmers the freedom to do so autonomously during working hours.

The questionnaire benefited from a long preparatory phase. First, it was discussed in depth with OSS project members and owner-managers of OSS firms (pilot testing). Then it was pre-tested on a number of OSS firms that were not included in the final sample. The questionnaire resulting from this phase was then administered by e-mail and through a dedicated Web site. Answers were checked for internal coherence by trained research assistants. In several cases, phone or face-to-face follow-up interviews with respondents were conducted to obtain missing information and ensure that the data that were collected are reliable. The sample considered in this paper included the 319 OSS firms for which we were able to build a complete dataset relating to the variables of interest (see Section 4 for a description of these variables).

Our study focused on a hidden population (Bonaccorsi et al., 2006). Indeed, no census exists of the firms offering OSS solutions located in the five European countries we considered. The absence of a precise definition of the population did not allow us to extract a representative sample. Notwithstanding this limitation, our sample exhibits several important strengths. First, it includes a large number of OSS firms located in different countries, while samples used in prior quantitative studies on OSS firms were smaller and only included firms from one country (Bonaccorsi et al., 2006; Gruber and Henkel, 2006; Dahlander, 2007; Stam, 2009). Second, our sample exhibits substantial heterogeneity with respect to the variables of interest, notably the type of OSS activities in which firms are involved, the tasks firms perform while contributing to OSS projects, their experience with interactions with the OSS community, and the adoption of the organizational practice under consideration.

[Table 1]

Table 1 reports descriptive statistics for the sample firms. On the survey date, 141 sample firms (44.2%) were found to have contributed to OSS projects.[8] About half of these firms authorized their programmers to collaborate autonomously on external OSS projects during working hours. There were no significant differences between these latter firms and the whole sample with respect to country and size distribution ($\chi^2(4) = 7.07$ and $\chi^2(3) = 5.46$, respectively).

[8] Within OSS projects, firms not only write documentation (60.5% of the firms that participate in OSS projects) or report and fix bugs (75.6%) but also contribute code (75.8%) and provide assistance to users by answering the questions that users post on projects' mailing lists (55.8%). These data are in line with recent findings of Capra et al. (2010), Eilhard (2008), and Ghosh et al. (2008) on firms' engagement in the OSS projects hosted on the SourceForge repository.

4. Methodology of the econometric analysis

The aim of the econometric analysis was to study the antecedents of the decision by firms to authorize programmers to contribute autonomously to external OSS projects during working hours. The dependent variable (*DAUTONOMY*) is a dummy that is equal to 1 for firms that adopted this organizational practice and 0 otherwise. As mentioned above, this question was posed only to the firms that contributed to OSS projects. Thus, the presence of *non-ignorable missing data* (Allison, 2002, p. 5) may engender a selection bias problem (Heckman, 1976). To address this problem, we used the two-stage correction of the probit model, as proposed by Heckman (1981). In the first stage, we estimated a probit selection model for the entire sample of OSS firms that explains the probability that a firm has ever contributed to an OSS project. In the second stage, we focused on the subsample of firms that contributed to OSS projects. We estimated a probit model that explains the probability that a firm adopts the organizational practice under consideration and inserted in this equation the inverse Mill's ratio obtained from the estimates of the selection equation. This ratio controls for unobservable factors that influence both the probability of contributing to OSS projects and of authorizing programmers to contribute autonomously to external OSS projects during working hours.

[Table 2]

A summary of the explanatory variables is reported in Table 2. Diversification of the firms' OSS activities is proxied by the number of different OSS activities in which firms are involved (*NActivities*). We predicted a positive coefficient for *NActivities*: the greater the number of OSS activities, the more likely it is that, while autonomously collaborating with the OSS community, programmers will absorb technical artifacts and programming competences that are valuable to their employer (Hypothesis H1). We distinguished five activities: i) offering of high-level services (e.g., personalized training on advanced functionalities and compilation of user manuals) on pre-packaged OSS products; ii) adaptation of pre-existing OSS programs and solutions to suit the needs of firms' customers; iii) integration of OSS modules and programs into new solutions that are released under OSS licenses; iv) design and development of solutions on order for firms' customers that are released under OSS licenses; and v) design and development of new solutions from scratch that are released into the market under OSS licenses.

We tested Hypothesis H2 by including in the set of regressors the logarithm of the number of OSS projects to which the firm has contributed since the beginning of its OSS activity (*LnOSSProjects*) as a measure of the experience of collaboration with the OSS community gained by the firm. We predicted a negative coefficient for this variable.

According to Hypothesis H3, if OSS firms are involved in high-value-added tasks, it is less likely that they authorize programmers to contribute to external OSS projects during working hours, as the opportunity cost of their time is greater. We know whether sample firms primarily offer OSS solutions (*DOSSBusinessModel* = 1) as opposed to proprietary solutions. Among the firms that primarily offer OSS solutions, those that develop OSS code (*DOSSCode* = 1) are involved in higher-value-added tasks than those that only write documentation or fix software bugs. Therefore, the opportunity cost of programmers' time is greater for the former type of firms. Conversely, for firms that primarily offer proprietary solutions (*DOSSBusinessModel* = 0), *DOSSCode* is an unsatisfactory proxy for the opportunity cost of programmers' time. Even if programmers do not develop OSS code, they may develop high-value-added proprietary code. We then inserted into the model specification, in addition to *DOSSCode*, the interactive term *DOSSCode* x *DOSSBusinessModel*. *DOSSBusinessModel* was also inserted as a control. While we have no prediction for the coefficient of *DOSSCode*, in accordance with Hypothesis H3, we expect the sum of the coefficients of *DOSSCode* and the interactive term *DOSSCode* x *DOSSBusinessModel* to be negative.

We also inserted several controls in the model specification, namely firm size (*LnSize*), calculated as the logarithm of the total number of the firm's owner-managers, employees, and freelancers (in full-time equivalents) in 2004, firm age in 2004 (*Age*), four product category dummies and three country dummies (being located in the Iberian peninsula is the benchmark in the estimates). We also included the inverse Mill's ratio obtained from the estimates of the selection equation. The explanatory variables inserted in this latter equation are inspired by the extant literature on firms' contributions to OSS projects (Capra et al., 2010). They include a dummy variable (*DOSSAgreement*), which is equal to 1 if agreement with the values of the OSS movement (i.e., championing exchange of knowledge and sharing of code among the OSS community members, see Zeitlyn, 2003) was indicated by respondents as one of the three main motives for their firm's involvement in OSS activities; the (log of the) the number of OSS products (*LnOSSProducts*) and services (*LnOSSServices*) offered by firms; firm age; and country dummies.

Table 3 gives descriptive statistics and the correlation matrix of the explanatory variables included in the models. Correlations among variables were low, so we excluded the presence of multi-collinearity problems.

[Table 3]

5. Econometric results

The results of the econometric analysis are illustrated in Table 4. Model 1 included only the control variables. In Model 2, we added the explanatory variables aimed at testing our theoretical hypotheses. For this latter model, we also showed the marginal effects of a one-unit change in any explanatory variable on the probability of adopting the managerial practice under investigation. Marginal effects are calculated with all other covariates evaluated at the sample means.

[Table 4]

Let us first focus on Model 1. Among the controls, only *DFin* was significant. Firms located in Finland were more likely to authorize programmers to contribute autonomously to external OSS projects during working hours. The other controls were not significant.

Let us now consider the estimates of Model 2. The signs of the coefficients of the control variables did not differ from those of Model 1. However, in Model 2, *LnSize* and *InverseMills* were (weakly) significant. More interestingly, the insertion into the model specification of the covariates that reflect our theoretical arguments substantially improved the explanatory power of the model, as is documented by the increase of the Pseudo R^2 (from 0.13 to 0.39). As was predicted by Hypothesis H1, the more diversified the set of firms' OSS activities, the higher the probability that their programmers were authorized to collaborate autonomously on external OSS projects during working hours. *NActivities* had a significant positive (at 5%) coefficient. The addition of one activity to a firm's set of activities led to a 13.2% increase in the probability of adopting the organizational practice under consideration. Conversely, the greater the number of OSS projects in which firms were previously involved, which denotes greater experience with interactions with the OSS community, the lower this probability. The (negative) coefficient of *LnOSSProjects* was significant (at one percent), and the marginal effect was quite large (-21%). Hence, Hypothesis H2 is supported.

Last, let us consider Hypothesis H3, which claims that firms involved in high-value-added tasks are less likely to authorizing programmers to collaborate on external OSS projects during working hours. As is apparent from the (weakly) significant value of the Wald χ^2 test reported at the bottom of Table 4, OSS firms that predominantly offer OSS solutions as opposed to proprietary solutions (*DOSSBusinessModel* = 1) are less likely to adopt this organizational practice if they release OSS code (*DOSScode* = 1) than if they do not. For OSS firms that predominantly offer OSS solutions, releasing OSS code leads to a 25-percentage-point decrease in the likelihood of the adoption of the organizational practice under investigation. This result confirms the prediction made by Hypothesis H3. No such effect was found for OSS firms that predominantly offer proprietary solutions, which is in line with the view that, for these firms, *DOSSCode* is not correlated with the value-added nature of the tasks that they perform.

7. Discussion and conclusions

This note extends our understanding of the antecedents of the adoption of a specific organizational practice by firms that collaborate with communities of users and developers; the organizational practice considered is aimed at successfully in-sourcing from communities technological artifacts and competences that are valuable for the innovation activities of the firms. We focused on the collaboration between OSS firms and the OSS community and examined a popular organizational practice that consists of authorizing a firm's programmers to contribute autonomously to *external OSS projects* during working hours. Our findings show that the adoption of this practice is more likely if firms have limited experience with previous collaborations with the OSS community. When firms allow these collaborations, the firms' programmers are likely to have specific knowledge of the OSS community beyond that possessed by their bosses. Granting programmers autonomy allows the firm to use this knowledge when collaborating with the OSS community, thereby making this collaboration more effective. We also found that adoption was more likely if a firm is involved in a more diversified set of OSS activities, as it is more likely that the innovation inputs that programmers absorb from the community will be valuable to the firm. Last, taking into account the cost side of the organizational practice under investigation, our results provide (admittedly weak) evidence that the likelihood of the adoption of this practice decreases with the opportunity cost of programmers' time.

In spite of the importance of these results, this note has some limitations that provide directions for future research. First, our analysis did not take into account the micro-level nature of individuals and, in particular the individual-specific characteristics of firms' programmers. These characteristics clearly influence the extent of programmers' information advantage over their bosses and thereby the probability that they will be authorized to contribute to external OSS projects of their choice during working hours. Shedding light on the micro-level nature of individuals would be of high relevance in the context of firm-OSS community collaborations. The specific knowledge of the OSS community is likely to be unevenly distributed among firms' personnel. For instance, this knowledge may depend on programmers' individual behaviors, such as the habit of contributing to OSS projects during spare time on a voluntary basis. The opportunity cost of programmers' time also depends on their individual skills. Consequently, consideration of programmers' individual characteristics is crucial for allowing firms to grant programmers autonomy on a selective basis.

Second, we could not take into account the influence of inertia on the adoption of the organizational practice in this investigation as we did not have access to longitudinal data. As prior studies have shown (e.g., Ichniowsky and Shaw, 1995; for a review, see Colombo and Delmastro, 2008, ch. 8), a firm may use a given practice simply because it was adopted in the past, despite changes in employees' skills, the firm's activities or environmental changes that have made the practice suboptimal. For instance, an OSS firm with a long-lasting tradition of successful centralization of decisions may be unlikely to allow programmers to collaborate autonomously with the OSS community, even if this organizational arrangement would lead to a more effective collaboration.

Third, herein we considered a specific organizational practice. To effectively in-source technical artifacts and programming competences from the OSS community, OSS firms may need to change other aspects of their organization. Possible changes include the adoption of suitable compensation packages for programmers (including non-monetary compensation) and re-designing the organization of internal software development projects. Jointly exploring the antecedents of (some of) these organizational aspects would be a very interesting extension of this research.

In spite of these limitations, we think that our work contributes to the literature on firm-community collaborations. This research set aside the community-centered perspective adopted by the bulk of the OSS literature and put the firm and its organization at the core of the analysis. More precisely, our work advanced the argument that to benefit from this new external innovation source, firms have to implement a suitable organizational design. The decision to grant autonomy to programmers is an important aspect of this design. Our note also contributes to the organizational design literature. The adoption of new organizational practices granting greater autonomy to employees has attracted increasing scholarly interest. Most previous studies (e.g., Osterman, 1994; Ichniowski et al., 1996) have analyzed the adoption of workplace practices, such as quality circles, total quality management practices and job rotation among employees. Recently, scholars have devoted attention to organizational practices aimed at proficiently collaborating with third parties (e.g., with users, Foss et al., 2010). Our note extends this line of research to collaborations with communities, which so far is an unexplored field.

We envisage that our work is of great interest for practitioners. Our results indicate that designing a suitable organization for the sourcing of technical artifacts and competences from the OSS community is a serious managerial challenge. There is no single best way to collaborate with the community. Programmer autonomy is necessary to gain the most from the collaboration if a firm is a newcomer in the OSS arena. This holds even truer if the firm is involved in a diversified set of OSS activities and is therefore likely to in-source inputs valuable to its innovation activities. However, firms must also realize that granting autonomy to their programmers has drawbacks; it diverts their talent from internal software development projects, thereby possibly hampering productivity.

References

Ågerfalk and Fitzgerald. 2008. Outsourcing to an unknown workforce: exploring opensourcing as a global sourcing strategy. *MIS Quartely* **32**(2): 385-409.

Allen TJ. 1977. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization*. MIT Press, Cambridge.

Allison PD. 2002. *Missing Data*. Sage Publications: Thousand Oaks (CA).

Alonso R, Matouschek N. 2008. Optimal delegation. *The Review of Economic Studies* **75**(1): 259-293.

Bonaccorsi A, Giannangeli S, Rossi C. 2006. Entry strategies under competing standards: Hybrid business models in the Open Source software industry. *Management Science* **52**(7): 1085–1098.

Bonaccorsi A, Rossi C, Scateni A. 2005. "ELISS Report. Rapporto del progetto ELISS", European Libre Software Survey. MIMEO.

Bonaccorsi A, Rossi C. 2006. Comparing motivations of individual programmers and firms to take part in the Open Source movement. From community to business. *Knowledge, Technology and Policy* **18**(4): 40-64.

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

Cabiddu F, Piva E, Rossi Lamastra C. 2010. The allocation of decision making power in firms doing business with OS communities: A case study analysis, *working paper*.

Capra E, Francalanci C, Merlo F, Rossi Lamastra C. 2010. Firms' participation in Open Source projects: which impact on software quality and success. *Journal of System Software*, forthcoming.

Colombo MG, Delmastro M. 2008. *The Economics of Organizational Design: Theoretical Insights and Empirical Evidence*. Palgrave Macmillan.

Dahlander L. 2007. Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source communities. *Industrial and Corporate Change* **16**(5): 913-943.

Dahlander L, Magnusson MG. 2005. Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* **34**(4): 481-493.

Dessein W. 2002. Authority and communication in organizations. *Review of Economic Studies* **69**(4): 811-838.

Eilhard J. 2008. Firms on SourceForge. MPRA Paper No. 7809

Evers S. 2000. *An introduction to Open Source Software development*. Working paper. Technische Universität Berlin, Fachbereich Informatik, Fachgebiet Formale Modelle, Logik und Programmierung (FLP).

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

Foss NJ, Laursen K, Pedersen T. 2010. Linking costumer interaction and innovation: the mediating role of new organisational practices. *Working Paper* Copenhagen Business School.

Ghosh R, Haaland K, Hall BH. 2008. Which firms participate in open source software development? A study using data from Debian. Paper presented at "DIME - DRUID Fundamental on Open and Proprietary Innovation Regimes", May 17th, Copenhagen, Denmark.

Gruber M, Henkel J. 2006. New ventures based on open innovation -an empirical analysis of start ups firms in embedded Linux. *International Journal of Technology Management* **33**(4): 256-372.

Hargrave TJ, Van de Ven A. 2006. A collective action model of institutional innovation. *Academy of Management Review* **31**(4): 864-888.

Hars A, Ou S. 2002. Working for free? Motivations for participating in Open-Source projects. *International Journal of Electronic Commerce* **6**(3): 25-39.

Hayek F.A. 1945. The use of knowledge in society. *American Economic Review* **35**(4): 519-530.

Heckman JJ. 1976. The common structure of statistical models of truncated, sample selection and limited dependent variables, and a simple estimator of such models. *Annals of Economic and Social Measurement* **5**: 475-492.

Heckman JJ. 1981. The incidental parameters problem and the problem of initial conditions in estimating a discrete time–discrete data stochastic process. In *Structural Analysis of Discrete Data with Econometric Applications*, Manski CF, McFadden D. (eds.) MIT Press: Cambridge (MA): 179–195.

Henkel J. 2009. Champions of revealing - The role of Open Source developers in commercial firms. *Industrial and Corporate Change* **18**(3): 435-471.

Hertel G, Niedner S, Herrmann S. 2003. Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel. *Research Policy* **32**(7): 1159–1177.

Ichniowsky C, Shaw K. 1995. Old Dogs and New Tricks: Determinants of the Adoption of Productivity-Enhancing Work Practices. *Brooking Papers on Economic Activity: Microeconomics*. Brookings Institute: Washington, DC.

Ichniowski C, Kochan T, Levine D, Olson C, Strauss G. 1996. What works at work. *Industrial Relations* **35**: 299-333.

Jensen M, Meckling C. 1992. Specific and General Knowledge, and Organizational Structure. In: *Contract Economics*, Werin L, Wijkander H. (eds.) Cambridge, Mass. and Oxford: Blackwell. 251-274.

Knudsen T, Levinthal DA. 2007. Two faces of search: Alternative generation and alternative evaluation. *Organization Science* **18**(1): 39-54.

Lakhani KR, Wolf RG. 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In *Perspectives on Free and Open Source Software*, Feller J, Fitzgerald B, Hissam S, Lakhani KR. (eds) MIT Press, Cambridge, MA.

Lee G.K, Cole R. 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science* **14**(6): 633-649.

Lerner J, Tirole J. 2002. Some simple economics of Open Source. *Journal of Industrial Economics* **50**(2): 197-234.

Miller K.D., Fabian F., Lin S. (2009) Strategies for online communities. Strategic Management Journal,

O'Mahony S, Bechky BA. 2008. Boundary organizations: Enabling collaboration among unexpected allies. *Administrative Science Quarterly* **53**(3): 422–459.

O'Mahony S, West J. 2008. The role of participation architecture in growing sponsored Open Source communities. *Industry & Innovation* **15**(2): 145-168.

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

Osterloh M, Rota S. 2007. Open source software development- Just another case of collective invention. *Research Policy* **36**(2): 157-171.

Osterman P. 1994. How common is workplace transformation and how can we explain who does it? *Industrial and Labor Relations Review* **47**(1): 173-188.

Pisano G. 2006. Profiting from innovation and the intellectual property revolution. *Research Policy* **35**(8): 1122-1130.

Raymond ES. 2001. A brief story of hackerdrom. In *The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary*. O'Reilly & Associates, Sebastopol, CA.

Stallman R. 1984. The GNU Manifesto. http://www.gnu.org/gnu/manifesto.html.

Stam W. 2009. When does community participation enhance the performance of open source software companies? *Research Policy* **38**(8): 1288–1299.

von Krogh G, Rossi Lamastra C, Haefliger S. 2009. Phenomena- based research in management and organization science. The case of Open Source Software. Paper presented at Academy of Management Meeting 2009, Chicago.

Zeitlyn D. 2003. Gift economies in the development of open source software: Anthropological reflections. *Research Policy* **32**(7): 1287-1291.

Tables

Table 1: Distribution of sample firms by country and size

	No. of firms	%
Country		
Finland	55	17.2
Germany	34	10.7
Italy	160	50.2
Portugal	14	4.4
Spain	56	17.6
Total	*319*	*100.0*
Total number of owner-managers, employees and freelancers		
0-10	193	60.5
11-50	92	28.8
51-250	21	6.6
> 250	13	4.1
Total	*319*	*100.0*

Table 2: Definition of controls and explanatory variables and expected signs of their coefficients.

Variable	*Description*
NActivities	Number of different OSS activities in which the firm is involved
LnOSSProjects	Logarithm of the number of OSS projects joined by the firm since the start of its OSS activity
DOSSCode	1 if the firm develops OSS code; 0 otherwise
DOSSBusinessModel	1 if the firm mainly offer OSS solutions; 0 if it mainly offer proprietary solutions
LnSize	Logarithm of firm size measured by the total count of owner-managers, employees, and freelancers
Age	Firm's age at survey date
DServer	1 if the firm develops Web servers or other kinds of servers; 0 otherwise
DWebProducts	1 if the firm develops solutions for e-commerce, e-mail clients, instant messaging or Web browsers; digital signature or content management systems; or e-learning tools; 0 otherwise
DNetworkInfrastrProducts	1 if the firm develops back-up systems, firewalls, antispam or antivirus solutions or user and identity management software; 0 otherwise
DOther	1 if the firm develops management or data management software, workflow systems or office automation packages; 0 otherwise
DFin	1 for Finnish firms; 0 for remaining firms
DGer	1 for German firms; 0 for remaining firms
DIta	1 for Italian firms; 0 for remaining firms
InverseMills	Inverse Mill's ratio obtained from the estimates of the selection equation

Table 3: Descriptive statistics for the independent variables included in the econometric models and the correlation matrix.

Variables	*No. of firms*	*Mean*	*S.D.*	*Min.*	*Max.*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
NActivities	109	3.963	1.113	1.000	5.000	1.000												
LnOSSProjects	109	1.517	0.892	0.000	5.017	0.296	1.000											
DOSSCode	109	0.780	0.416	0.000	1.000	0.142	0.151	1.000										
DOSSBusinessModel	109	0.596	0.493	0.000	1.000	0.395	0.190	-0.031	1.000									
LnSize	109	2.199	1.126	0.000	5.991	0.022	0.333	0.121	-0.167	1.000								
Age	109	5.596	5.032	0.000	22.000	-0.228	0.026	0.050	-0.272	0.322	1.000							
DServer	109	0.807	0.396	0.000	1.000	-0.037	0.172	0.133	0.262	-0.075	0.012	1.000						
DWebProducts	109	0.853	0.356	0.000	1.000	0.197	0.180	0.092	0.293	0.015	-0.220	0.455	1.000					
DNetworkInfrastrProducts	109	0.798	0.403	0.000	1.000	0.128	0.249	0.009	0.285	-0.078	-0.009	0.508	0.373	1.000				
DOther	109	0.862	0.346	0.000	1.000	0.035	0.093	-0.084	0.106	0.082	0.095	0.345	0.211	0.330	1.000			
DFin	109	0.165	0.373	0.000	1.000	-0.075	-0.078	-0.121	-0.087	-0.096	-0.255	-0.159	-0.025	-0.207	-0.109	1.000		
DGer	109	0.037	0.189	0.000	1.000	0.095	0.239	-0.014	0.061	0.174	0.250	-0.028	-0.057	-0.023	-0.064	-0.087	1.000	
DIta	109	0.486	0.502	0.000	1.000	-0.067	-0.207	-0.148	-0.023	-0.334	-0.013	-0.083	-0.219	-0.105	-0.144	-0.433	-0.190	1.000
InverseMills	109	0.741	0.315	0.252	1.645	-0.246	-0.274	-0.014	-0.411	0.161	0.514	-0.448	-0.391	-0.443	-0.298	-0.204	0.005	0.250

Table 4: Determinants of the adoption of the managerial practice under investigation.

		Probit equation Dependent variable: *DAUTONOMY* *Model 1*		*Model 2*	
a_0	*Constant*	-0.808 (0.813)		-1.381 (1.249)	
a_1	*NActivities*	-		0.330 (0.142) [0.132]	**
a_2	*LnOSSProjects*	-		-0.525 (0.195) [-0.209]	***
a_3	*DOSSCode*	-		0.489 (0.581) [0.190]	
a_4	*DOSSBusinessModel*	0.135 (0.245)		0.736 (0.734) [0.285]	
a_5	*DOSSCode x DOSSBusinessModel*	-		-1.378 (0.757) [-0.506]	*
a_6	*LnSize*	0.146 (0.112)		0.281 (0.150) [0.112]	*
a_7	*Age*	0.028 (0.030)		0.052 (0.037) [0.021]	
a_8	*DServer*	0.095 (0.351)		-0.147 (0.484) [-0.058]	
a_9	*DWebProducts*	0.157 (0.346)		0.266 (0.469) [0.105]	
a_{10}	*DNetworkInfrastrProducts*	-0.039 (0.340)		0.272 (0.436) [0.107]	
a_{11}	*DOther*	-0.014 (0.324)		-0.159 (0.476) [-0.063]	
a_{12}	*DFin*	1.154 (0.406)	***	1.164 (0.484) [0.418]	**
a_{13}	*DGer*	0.540 (0.464)		1.402 (0.881) [0.445]	
a_{14}	*DIta*	0.471 (0.310)		0.381 (0.360) [0.151]	
a_{15}	*InverseMills*	-0.568 (0.536)		-1.381 (1.249) [-0.487]	*
Number of observations		138		109	
Pseudo R^2		0.13		0.39	

Wald χ^2 test: $a_3 + a_5 = 0$	-	3.58(2)*

Legend. *Significance level greater than 90%; **significance level greater than 95%; ***significance level greater than 99%. Robust standard errors and number of restrictions are in parentheses. Marginal effects are in square brackets. McFadden Pseudo R^2.