ELSEVIER

Long Range Planning 46 (2013) 236–257

LRP long range planning

http://www.elsevier.com/locate/lrp

Authorising Employees to Collaborate with Communities During Working Hours: When is it Valuable for Firms?

Massimo G. Colombo, Evila Piva and Cristina Rossi-Lamastra

This paper advances our comprehension of how firms organise to in-source technical knowledge from communities of users and developers. Specifically, the paper focuses on software firms doing business with the Open Source Software (OSS) community (OSS firms). It explores the antecedents of the adoption by OSS firms of a quite popular organisational practice: authorising firm programmers to contribute autonomously during their working hours to OSS projects to which their employers do not contribute on their own behalf. We argue that this practice serves the purpose of scouting the OSS community for new knowledge by leveraging the individual-level absorptive abilities of programmers. Accordingly, we expect the likelihood of its adoption is higher for the OSS firms that: i) must rely on the individual-level abilities of their programmers to acquire and assimilate new knowledge from the OSS community as they have smaller *firm-level potential absorptive capacity*; ii) are able to transform and exploit effectively the new knowledge in-sourced by their programmers as they have greater *firm-level realised absorptive capacity*. The econometric results based on data from 293 European OSS firms provide support to our hypotheses.

Introduction

Scholars and practitioners agree that firms can fruitfully rely on knowledge generated outside of their boundaries to obtain valuable inputs for their innovation processes (Chesbrough, 2010). Nevertheless, we know little about how firms should organise internally to effectively absorb knowledge

0024-6301/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.lrp.2012.05.004

from external sources (Volberda et al., 2010). Because absorbing external knowledge poses major challenges to firms, suitable organisational structures and practices must be designed to effectively perform this activity (Colombo et al., 2012). This argument holds especially true for the in-sourcing of external knowledge from communities of users and developers, which have become an important source of technological knowledge (von Hippel, 2006). As will be explained in greater detail in the following section, communities of users and developers produce knowledge in a decentralised manner and have peculiar institutional characteristics. Thus, firms face severe challenges in scouting these communities for valuable knowledge.

The Open Source Software (OSS) community is a prominent example of a community of users and developers that produces technological knowledge suitable for commercial applications (von Hippel and von Krogh, 2003). Although the OSS community was born out of an ideological movement (Stallman, 1984), its importance to firms operating in the software industry is evident (Fitzgerald, 2006). The OSS community constitutes a large and varied common pool of knowledge, in terms of programming competences and freely available software code (Bonaccorsi and Rossi, 2006). The OSS community comprises thousands of software development projects (OSS projects)[1] performed according to the principles of the OSS production mode (e.g., openness of the code and decentralisation of production) by developers from all around the world who are driven by a plethora of different motives (von Krogh et al., 2012).

Currently, many software firms take advantage of the OSS pool of knowledge and provide their customers with OSS-based solutions. We use the term *OSS firms* to refer to these firms (Bonaccorsi et al., 2006). In some cases, OSS firms simply adapt to their customers' needs the OSS code freely available on the internet. In other cases, OSS firms *contribute* to OSS projects *on their own behalf*. Namely, they may launch and co-ordinate new OSS projects (O'Mahony and West, 2008) or sponsor existing OSS projects and make their own code available within these by releasing it under OSS licences (Henkel, 2009). OSS firms that contribute to OSS projects on their own behalf pay their employees to write code on these projects (Hars and Ou, 2002), debug the code produced by other project members, write documentation and answer technical questions raised by users through the projects' mailing lists (Lakhani and von Hippel, 2003). In most cases, these firms are also interested in scouting the OSS community for new and relevant knowledge that resides in *other OSS projects* (i.e., OSS projects to which the firms *do not* contribute on their own behalf; we will hereafter refer to these as *external OSS projects*). This scouting activity is highly beneficial to OSS firms. They can pick out new software code and modules from the OSS common pool and integrate them into their own solutions (Dahlander, 2007). Moreover, OSS firms can benefit from serendipity processes and find new and unexpected uses for their own software code. Finally, they can scout skilled OSS programmers to collaborate with or recruit (Eilhard, 2008).

Despite these advantages, scouting the OSS community for new knowledge poses serious challenges to OSS firms, which can be overcome through the adoption of suitable organisational practices. In particular, we focus here on a specific but widely diffused organisational practice that may be used by OSS firms for this purpose and study what renders its adoption more likely. The practice consists of *authorising firm programmers to contribute autonomously to external OSS projects during working hours*.

In this work, we argue that firm programmers are in a privileged position to *acquire* and *assimilate* new knowledge from external OSS projects. Programmers have *individual-level absorptive abilities*, as they possess *personal specific knowledge* (Jensen and Meckling, 1992) of the OSS community. The autonomy given to programmers serves the purpose of using their personal specific knowledge of the OSS community to make decisions about knowledge in-sourcing from external OSS projects. This argument is inspired by Hayek's seminal work, which states that decision authority (in our case, decision authority on how to in-source knowledge from external OSS projects) must be co-located with

[1] An OSS project is "any group of people developing software and providing their results to the public under an Open Source license" (S. 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.

the knowledge that is relevant to the decision (i.e. the knowledge of the OSS community possessed by firm programmers).[2] By adopting this practice, firms can leverage the individual-level absorptive abilities of their programmers (Matusik and Heeley, 2005) to generate *firm-level potential absorptive capacity* (Zahra and George, 2002). The practice is more valuable to a firm, and thus its adoption is more likely, depending on two contingencies. First, the practice is especially beneficial to firms that, in the absence of programmers' individual-level absorptive abilities, would have limited *firm-level potential absorptive capacity*. This situation occurs if programmers enjoy an information advantage over their boss (i.e., firms' owner-managers in small entrepreneurial firms and salaried managers in charge of software development in larger firms) in acquiring and assimilating new knowledge from external OSS projects. Second, the knowledge acquired and assimilated by programmers from external OSS projects needs to be transformed and exploited by the firm. Hence, the firm will obtain greater benefits from this practice if it has greater *realised absorptive capacity*.

The paper proceeds as follows. In the next section two, we present the conceptual framework and develop hypotheses on the contingencies that influence the probability that this practice will be adopted. In the empirical sections of the paper, we use data from a sample of 293 European OSS firms to test these hypotheses using econometric models. Specifically, we first illustrate the dataset used in the empirical analysis. In the subsequent section, we describe the econometric methodology and the variables used in the estimates. The econometric results are presented afterward. In the concluding section, we discuss the results and their managerial implications and provide directions for further research.

Conceptual background

The challenges of scouting the OSS community for new knowledge

Scouting the OSS community for new knowledge that resides in external OSS projects poses major challenges to firms because of the peculiar software production mode and institutional framework of the OSS 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 project members from all around the world. The technical skills of these individuals and the quality of their contributions are highly variable. A large body of literature has acknowledged that these contributors have heterogeneous motives (von Krogh et al., 2011). Some OSS project members may be eager to signal their talent through their OSS coding activities in order to obtain better jobs (Lerner and Tirole, 2002), whereas others may develop software code simply for fun (Hertel et al., 2003). These diverse motivations are likely to impact the quality of the produced software code, but they are unobservable to project outsiders.

Second, software development in OSS projects is not ruled by contracts. No enforceable agreement among project members specifies what the final output of an OSS project should be. In some cases (e.g., OSS projects sponsored by companies), project members receive monetary compensation for their OSS development activities. However, project members *are not employees of the project* (O'Mahony, 2003). Consequently, it is difficult to determine whether, when and how they will perform their software development activities. Project discontinuation and departure from the initial specifications are concrete risks (Feller and Fitzgerald, 2002; Shah, 2006). Finding trustworthy OSS project members who are committed to reliably executing software development may be 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 its future development directions. In OSS projects, leadership usually

[2] "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" (F. A. Hayek, 1945, p. 524).

emerges from the bottom up (O'Mahony and Ferrero, 2007; Dahlander and O'Mahony, 2011) and may be challenged by the most active participants. These participants may possess relevant information regarding the software development process that is unknown to the project's administrators. Detecting these key participants 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). To *guard their commons* from external meddling by for-profit entities, OSS project members may follow different strategies (O'Mahony, 2003). For instance, project members may adopt an esoteric style of software documentation or prohibit participation in other OSS projects that are not released under the GNU General Public License (Lerner and Tirole, 2005). If these strategies are replicated over time, they may acquire the status of *unwritten norms*: un-codified rules that govern the behaviours of project members (Ågerfalk and Fitzgerald, 2008). Awareness of these unwritten norms is mandatory when scouting external OSS projects for new knowledge. Ignorance of these norms may drive an outsider to make mistakes when interacting with project members and undermine further knowledge transfer relations.

The following section describes why the organisational practice of authorising firm programmers to contribute autonomously to external OSS projects during working hours can help OSS firms to overcome these challenges.

The practice

The practice of authorising firm programmers to contribute autonomously to OSS projects during working hours echoes the Google "20 per cent time" rule that enables Google engineers to spend one day a week working on projects that are not necessarily in their job descriptions.[3] As will be documented later in the paper, this practice is widely adopted by OSS firms. The case of a small OSS firm based in Pisa, Italy, which we name Alfa for privacy reasons, is a telling example. *Alfa* contributes on its own behalf to the development of the *Plone* and *Python OSS* projects. However, an employee observed:

> *"In our firm, there is great flexibility about the OSS projects that a programmer can contribute to during working hours. A colleague of mine is now contributing to a small OSS project that she found through the network of her personal social contacts…"*

The practice under consideration here can serve different purposes. First, a well-established tradition in the human resource management literature acknowledges that granting autonomy to employees increases their job satisfaction and thus positively impacts their productivity (Foss et al., 2009). This holds particularly true for talented employees who generally value a job environment in which people are empowered and can freely express their creativity (Finegold and Frenkel, 2006). In this respect, Dahlander and O'Mahony (Dahlander and O'Mahony, 2011) noticed that highly skilled technical experts often do not seek to become managers but seek alternative rewards, such as *greater autonomy* (p. 963). Second, the practice can be a way for OSS firms to show their compliance with the values of independence and free organisation of work championed by the *hacker culture*, which has strongly inspired the OSS movement. By adopting this practice, an OSS firm may earn a reputation as a good "OSS citizen", which fosters feedback from the OSS community and nourishes a collaborative attitude from OSS developers (Osterloh and Rota, 2007).

Although we recognise the importance of the aforementioned purposes, we argue that OSS firms authorise their programmers to work on external OSS projects during working hours for an additional crucial reason: to scout external OSS projects for new knowledge that is relevant to and can be profitably used by the programmers' firms.

[3] On the Google "20 percent time" rule see, e.g., the Google official blog: http://googleblog.blogspot.com/2006/05/googles-20-percent-time-in-action.html, accessed on March 14th 2012. See also A. Sutherland, 2012.

In principle, programmers may use the autonomy they are given to contribute to external OSS projects that generate private benefits for them but are unrelated to their employers' activities. Under these circumstances, firms experience loss of control problems (Aghion and Tirole, 1997). For instance, firm programmers may decide to contribute to external OSS projects that pose intriguing algorithmic challenges or involve talented developers. These projects may be "pet" projects that provide no useful knowledge for the programmers' firms, even though contributing to these projects may boost the programmers' intrinsic motivations (Lakhani and Wolf, 2005). In the worst case, decisions that maximise the programmers' private benefits may be detrimental to the firm. In particular, the firm's competence pool may be drained. Relying on their autonomy, firm programmers may choose to contribute to external OSS projects simply to signal their talent (Lerner and Tirole, 2002). As their OSS activities are visible to everyone, these programmers may be noticed by other firms or by talented OSS developers. In the former case, the programmers may find a better job. In the latter case, they may join forces with others and start their own enterprise.

However, in practice, loss of control problems associated with the adoption of this practice by OSS firms are likely to be fairly limited for several reasons. First, internal career prospects, the desire to obtain a salary increase and "high-powered" incentives that link a portion of the programmers' salary to their individual performances usually drive firm programmers to scout external OSS projects for knowledge that is (also) relevant to their employers.[4] Second, by choosing to contribute to "pet" projects, programmers run the risk of having their autonomy revoked or, in the worst case, of being fired (Baker et al., 1999). Finally, most OSS firms are young and small entrepreneurial ventures (Dahlander and Magnusson, 2005). Hence, firms' (owner-)managers can closely monitor their programmers. Mutual monitoring by peer colleagues also helps to reduce deviant behaviours (Knez and Simester, 2001). In such a context, programmers' interests are largely aligned with those of their firms (Gottschalg and Zollo, 2007).

The qualitative evidence that we collected from interviews with OSS firm programmers and managers confirms that the adoption of the practice under investigation here is driven by the OSS firms' willingness to in-source relevant knowledge from external OSS projects. A successful Italian OSS firm that is based in Turin and has branches in Milan and Rome (henceforth: *Beta*) contributes on its own behalf to several OSS projects, such as *Plone*, *Linux-Ubuntu* and *Open Office*. The firm also authorises its programmers to contribute to external OSS projects of their choice during working hours. One of *Beta*'s programmers explicitly refers to this practice as a way to scout external OSS projects for new knowledge that is valuable to the firm:

> *"During working hours, we are free to contribute to whatever OSS project we like. The firm favours this behaviour, so one can provide the knowledge he has acquired by contributing to OSS projects for his/her colleagues' disposal... New ideas and solutions can stem from contributions to OSS projects that are different from those in which Beta is involved on its own behalf… novel possibilities for business or new developments for existing solutions can arise…"*

One of the owner-managers of a small but active OSS firm based in Bologna (henceforth: *Gamma*) confirms this view:

> *"All (Gamma's programmers) can contribute autonomously during working hours to OSS projects of their choice. This organisational arrangement boosts the improvement and the reuse of our software… What I mean is that if any OSS programmer around the world develops a software module that is useful for us and if we become aware of this module, we can find bugs and propose modifications to this module… If these modifications are accepted, we have a more robust and stable software to propose to prospective customers…"*

[4] On the fact that employees interacting with the OSS community usually do not put in place *commercially harmful behaviors* that may cause damage to their firms see Henkel, 2009.

By granting programmers the autonomy to contribute to external OSS projects during working hours, OSS firms take advantage of their programmers' individual-level absorptive abilities, which stem from their *insider roles* within the OSS community (Dahlander and Wallin, 2006). As part of their daily activities, firm programmers work on OSS projects, to which their employers contribute on their own behalf. In addition, anecdotal evidence indicates that programmers frequently contribute to OSS projects in their spare time. Because of these frequent interactions with the OSS community, firm programmers possess *personal specific knowledge* of the OSS community. Accordingly, they are in a privileged position to acquire and assimilate new knowledge relevant to their firms from external OSS projects. Firm programmers can locate external OSS projects that develop high-quality code to be integrated into their firms' solutions, single out talented OSS developers and link them to the firm, and make sense of OSS external projects' internal dynamics and future evolution. Moreover, because of their inside experience with OSS projects, firm programmers have an in-depth acquaintance of the fundamental values of the OSS community. Consequently, they can easily learn the specific norms of each OSS project that is inspired by these values and easily cope with them. As the aforementioned programmer working in Beta observes:

> *"I totally share the OSS ethic… the freedom, the possibility of taking something (from the OSS common pool), using it, modifying it… I like the fact that nobody is forced to buy licences… free software for everybody, irrespective of their financial situation… We (Beta's programmers) are closely integrated within the OSS community, we can easily move (within OSS projects)…"*

In brief, firm programmers can act as *gatekeepers* who bridge the firm and the OSS community (Chan and Husted, 2010).

In principle, an OSS firm might ask its programmers to transfer their personal knowledge of the OSS community to their boss. The boss would then have the authority to determine how knowledge should be in-sourced from the OSS community. Accordingly, the boss would give orders to firm programmers clarifying which OSS projects they should contribute to and which types of activities they should undertake within these projects. In this scenario, programmers would not have the autonomy to contribute to external OSS projects of their choice during working hours. In practice, this organisational arrangement is hardly viable. Stemming from an insider role, the knowledge of the OSS community possessed by firm programmers is not only largely *tacit* and thus difficult to transmit (Allen, 1977) but also hardly *verifiable* by the boss (i.e., this knowledge is *soft* Stein, 2002). Consequently, firm programmers would strategically communicate this knowledge (Dessein, 2002; Alonso and Matouschek, 2008) to the boss to induce her to select the OSS external projects that maximise the programmers' private benefits instead of the firm's profits. Granting autonomy to programmers eliminates the loss of information problems generated by the communication of distorted information. This argument is in line with the economic theory of delegation (Alonso et al., 2008), which states that if agents' objectives are sufficiently aligned with those of their employer, the loss of control problems incurred by a firm that grants decision autonomy to the agents are less detrimental to the firm than the loss of information problems that arise in the absence of autonomy.

In the following section, we will argue that the practice under investigation is more valuable to OSS firms (i) if the firms have limited involvement in and experience of collaboration with the OSS community and (ii) if the firms are large and have diversified OSS-based activities. OSS firms that have limited involvement in and experience of collaboration with the OSS community have limited *firm-level potential absorptive capacity*. This limited capacity makes reliance on the personal specific knowledge of the OSS community possessed by programmers and the programmers' associated *individual-level absorptive abilities* fundamental to acquire and assimilate knowledge from external OSS projects. Likewise, OSS firms with larger and more diversified operations are more capable of transforming and exploiting the knowledge that their programmers acquire and assimilate from the OSS community (i.e., these firms have greater *realised absorptive capacity*).

Research hypotheses

We claim that the benefits of authorising firm programmers to contribute autonomously to external OSS projects during working hours depend on the value of programmers' individual-level absorptive abilities to their firms. In turn, this value depends on the extent to which the programmers enjoy an information advantage over their bosses when scouting the OSS community for new knowledge. We mentioned earlier that the individual-level absorptive abilities of firm programmers result from their *insider roles* within the OSS community. Following a similar reasoning, we claim that an OSS firm that is highly embedded and plays an insider role on its own within the OSS community has developed firm-level potential absorptive capacity that substitutes for the individual-level absorptive abilities of its programmers. Accordingly, such a firm is able to acquire and assimilate new knowledge from the OSS community on its own and does not need to authorise programmers to contribute autonomously to external OSS projects during working hours.

We argue that OSS firms are highly embedded in the OSS community when they i) base their entire business on OSS by providing exclusively OSS-based (as opposed to proprietary) solutions to their customers *and* ii) write code within OSS projects on their own behalf (i.e., the code is authored by the firm). To run their businesses, these firms closely and frequently interact with OSS users and developers and so develop an in-depth familiarity of the OSS common pool of knowledge (Stam, 2009). In addition, by reciprocating code within the OSS projects, these firms earn good reputations within the OSS community[5] and receive more feedback and contributions from OSS developers than their non-coding peers (Shah, 2006). Relying on a wide and trustworthy network of contacts within the OSS community, these firms receive more information on which OSS projects are producing relevant software and with which OSS developers it is worth collaborating. As a result, these firms have a better capacity to acquire and assimilate new knowledge from the OSS community. Thus, the practice under investigation here is less valuable to these firms. From the above arguments, hypothesis H1 follows.

H1: The likelihood that an OSS firm authorises its programmers to contribute autonomously to external OSS projects during working hours is lower if the firm provides exclusively OSS solutions and writes code within OSS projects on its own behalf.

Insider status being equal, OSS firms' abilities to scout the OSS community for knowledge depends on the *experience* that these firms have developed in collaborating with the OSS community by contributing to OSS projects on their own behalf.

A wide range of studies acknowledge that experience is a prominent antecedent of absorptive capacity.[6] By contributing to OSS projects on their own behalf, OSS firms progressively aggregate and store relevant knowledge about the OSS community in their routines (Argote, 1999) and thus develop potential absorptive capacity at the firm level. Firm-level potential absorptive capacity substitutes for the individual-level absorptive abilities of the firms' programmers. Consequently, experienced OSS firms can fruitfully scout the OSS community for new knowledge without relying on the personal specific knowledge of the OSS community possessed by programmers. *Ceteris paribus*, one would then expect OSS firms that have more experience in collaborating with the OSS community to be less likely to adopt the practice under consideration here.

However, for this practice to be beneficial to an OSS firm, the external knowledge in-sourced by the firm's programmers must be transformed and exploited by the firm. In other words, OSS firms must possess firm-level *realised* absorptive capacity to leverage the individual-level absorptive abilities of their programmers. Inexperienced firms may be so new to the OSS arena that they cannot transform and exploit the new knowledge that their programmers in-source from the OSS community. Under these circumstances, the practice is virtually useless.

Combining the above arguments, we predict a curvilinear, inverse U-shaped relation between the experience that an OSS firm has obtained in collaborating with the OSS community and the

[5] On the reputation gains from writing code within OSS projects see e.g. Baldwin and Clark, 2006.

[6] For a comprehensive survey of the literature on absorptive capacity see H.W. Volberda, N.J. Foss and M.A. Lyles, Absorbing the concept of absorptive capacity: How to realize its potential in the organization field, Organization Science 21(4), 1–21(2010).

likelihood that the practice will be adopted. In other words, the likelihood of adoption first increases and then decreases as firm experience increases. Hypothesis H2 follows.

H2. The relationship between the likelihood that an OSS firm authorises its programmers to contribute autonomously to external OSS projects during working hours and the experience that an OSS firm has in collaborating with the OSS community is inverse U-shaped.

An OSS firm's capacity to transform and exploit the new knowledge that its programmers in-source from the OSS community depends not only on the firm's experience in collaborating with the OSS community but also on the *scope* and *scale of the firm's activities* (Chandler, 1994).

As illustrated in the previous section, the quality of the software code dispersed within the OSS common pool and the competences of OSS project members are heterogeneous. Therefore, the knowledge in-sourced by firm programmers from the OSS community is quite unpredictable. The likelihood that an OSS firm can fruitfully transform and exploit the new knowledge acquired and assimilated by its programmers increases with the diversity of the firm's OSS-based activities.

Some OSS firms commercialise pre-packaged OSS products and provide complementary services (e.g., installation services, training on advanced functionalities and compilation of user manuals). What matters for these firms is the availability of downloadable high-performance software products within the OSS community. Other OSS firms adapt OSS code freely downloaded from the web to their customers' needs. These firms may also integrate different OSS modules to develop customised solutions. In this case, in addition to the quality of the software code, its adaptability and integrability with other software modules is fundamental. Lastly, some OSS firms develop software products from scratch and release them under OSS licences. For these firms, a critical factor is the availability of talented developers within the OSS community who are willing to collaborate with them. Of course, any combination of these OSS-based activities is possible. The wider the scope of a firm's OSS-based activities, the higher the probability that its programmers will in-source knowledge that is valuable to the firm while contributing autonomously to external OSS projects. Hypothesis H3 follows.

H3: The likelihood that an OSS firm authorises its programmers to contribute autonomously to external OSS projects during working hours is higher if the scope of the firm's OSS-based activities is wider.

Finally, the scale of OSS firms' activities (i.e., firm size) matters. Provided that firm programmers have acquired and assimilated relevant knowledge while autonomously contributing to external OSS projects during working hours, larger OSS firms can more profitably transform and exploit this knowledge than their smaller counterparts. Accordingly, larger OSS firms obtain larger benefits from the practice and are thus more likely to adopt it. First, to effectively transform and exploit the knowledge in-sourced by their programmers from external OSS projects, OSS firms must possess complementary assets. To "earn a living" out of the OSS code that is freely available on the internet, OSS firms must add value to the code. They can do so by combining the OSS code with complementary technological and commercial assets.[7] Similar to any large firm, large OSS firms possess more complementary assets, especially commercial assets (e.g., sales force, customer service personnel, distribution channels and brand) (Teece, 1986). Thus, the knowledge in-sourced by programmers from external OSS projects is more valuable to these firms than to smaller OSS firms. Second, larger firms usually have more slack resources (George, 2005) and are less financially constrained (Carpenter and Petersen, 2002) than smaller firms. Hence, larger OSS firms can more easily reallocate part of their employees' time to adapt to their customers' needs the OSS code that firm programmers have in-sourced when contributing autonomously to external OSS projects. Likewise, these firms have more financial resources to hire new employees for this purpose. Third, pre-packaged software benefits from economies of scale (Shapiro and Varian, 1999). Delivering the first copy to the market is usually expensive because most production costs are fixed and sunk.

[7] For instance, OSS firms can offer support or professional consulting services on freely available OSS solutions, develop proprietary modules that add new functionalities to these solutions, and sell hardware devices or appliances that incorporate OSS software and provide customers with after-sale services. In this regard, see e.g., J. Henkel, 2006 and Hecker, 1999.

Conversely, it is cheap to reproduce and distribute subsequent copies to customers[8] (i.e., marginal costs tend to be low). Pre-packaged software solutions that are based on the open code developed by the OSS community are no exception. To prepare the first copy of an OSS-based pre-packaged solution, an OSS firm must access and download the OSS code available on the internet. Then, the firm must package this code (e.g., by adding graphical interfaces or drivers for controlling peripheral devices) in a software solution that customers can easily install on their computers. These operations engender costs that are fixed and sunk for the most part. Once the first copy is ready for the market, the reproduction and distribution costs are low. Thus, the average costs incurred by an OSS firm offering pre-packaged OSS-based software solutions (i.e., the unit cost per copy sold) decreases with the scale of its activities. According to the same line of reasoning, for OSS firms that offer pre-packaged software solutions, the unit cost per copy sold of the practice under consideration here decreases with firm size. Based on the three arguments above, we propose hypothesis H4.

H4: The likelihood that an OSS firm authorises its programmers to contribute autonomously to external OSS projects during working hours is higher if the size of the firm is larger.

The sample

The theoretical hypotheses put forward in the previous section were tested using data from the ELISS database (Bonaccorsi et al., 2005). This database 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 Commission within the Sixth Framework Program. ELISS provides information on 361 OSS firms. Data were collected through a structured questionnaire administered to the owner-managers or chief technology officers of a random sample of 6,000 firms operating in the software industry (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 provided their customers with OSS solutions (i.e., whether they were OSS firms). The firms that stated they did so were further asked whether they contributed to OSS projects on their own behalf. The questionnaire also included the following question: *does the firm authorise its programmers to contribute autonomously during working hours to OSS projects other than those to which the firm contributes on its own behalf?* Note that OSS firms can decide to not contribute to OSS projects. In this case, OSS firms simply download the open code freely available on the internet and adapt the code to the needs of their customers without releasing code and reciprocating technical knowledge back to the community. The aforementioned question was not posed to these OSS firms. OSS firms that do not contribute to OSS projects on their own behalf are unlikely to allow their programmers the freedom to contribute autonomously to external OSS projects during working hours.

The questionnaire benefited from a long preparatory phase. First, the questionnaire was discussed in depth with OSS project members and (owner-)managers of OSS firms (pilot testing). Then, it was pre-tested on a set of OSS firms that were not included in the final sample. The questionnaire resulting from this phase was then sent out by e-mail and through a dedicated website. The answers were checked for internal coherence by research assistants. In several cases, telephone or face-to-face interviews with the respondents were conducted to obtain missing information and to ensure that the data were reliable. The sample in this paper includes the 293 OSS firms for which we were able to build a complete dataset related to the variables of interest (see the next Section for a description of these variables).

Our study focused on a hidden population. There is no official census of the OSS firms located in the five European countries considered in this study. The absence of a precise definition of the population of OSS firms did not allow us to extract a representative sample. Despite this limitation, our

[8] Pre-packaged software can be reproduced by just copying it on a CD Rom or making it available on the Internet. Evidence exists that software distribution through the Internet has significantly reduced software distribution costs.

sample exhibits several important strengths. First, it includes a large number of OSS firms located in different countries, whereas the samples used in previous quantitative studies on OSS firms were smaller and only included OSS firms from an individual country (Gruber and Henkel, 2006). Second, our sample exhibits substantial heterogeneity with respect to the variables of interest (in particular the variables that measure whether the firms exclusively offered OSS solutions and wrote OSS code within OSS projects, the firms' experience in collaborating with the OSS community, and the number of OSS-based activities in which the firms were involved). Table 1 reports some descriptive statistics on the 293 sample OSS firms. The sample firms were not equally distributed across the five countries. Some 51.2 per cent of these firms were located in Italy. Most of the sample firms were small: 61.2 per cent of them had 10 or fewer employees, freelancers and owner-managers.

Table 1. Characteristics of the sample OSS firms

	No. of firms	%
Country		
Finland	52	17.1
Germany	28	9.6
Italy	150	51.2
Iberian Peninsula	65	22.2
Total	*293*	*100.0*
Size[a]		
0-10	178	61.2
11-50	83	28.5
51-250	20	6.9
> 250	10	3.4
Total	*291*	*100.0*
OSS-based activities in which the sample firms are involved[b]		
Offering of high-level services complementary to pre-packaged OSS products	205	70.0
Adapting pre-existing OSS programmes and solutions to suit the needs of the firms' customers	232	79.2
Integrating OSS modules and programs into new solutions that are released under OSS licences	193	65.9
Designing and developing solutions on order for firms' customers that are released under OSS licences	196	66.9
Designing and developing new solutions from scratch that are released into the market under OSS licences	136	46.4
Collaboration of firms with the OSS community		
The firm did not contribute[c] to OSS projects	159	54.3
The firm contributed to OSS projects	134	45.7
- The firm authorised its programmers to contribute autonomously to external OSS projects during working hours	66	22.5
- The firm did not authorise its programmers to contribute autonomously to external OSS projects during working hours	68	23.2
Total	*293*	*100.0*

[a] Total number of owner-managers, employees and freelancers. Data are missing for two firms.
[b] Firms may be involved in multiple OSS-based activities.
[c] We use the phrase "contributes to OSS projects" to refer to 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.

The sample firms were involved in different OSS-based activities. We considered the following five activities: i) offering of high-level services (e.g., personalised training on advanced functionalities and compilation of user manuals) complementary to pre-packaged OSS products; ii) adapting pre-existing OSS programmes and solutions to suit the needs of the firms' customers; iii) integrating OSS modules and programmes into new software solutions that are released under OSS licences; iv) designing and developing solutions on order for firms' customers that are released under OSS licences; and v) designing and developing from scratch new solutions that are released in the market under OSS licences. The figures in Table 1 clearly indicate that many firms were involved in more than one OSS-based activity. The most common activity was adapting pre-existing OSS solutions to suit the customers' needs (232 firms; 79.2 per cent of the sample). In contrast, fewer than half of the sample firms (136 firms; 46.4 per cent of the sample) developed new solutions from scratch and released them under OSS licences.

On the survey date, 134 sample firms (45.7 per cent) were found to have contributed to OSS projects on their own behalf. Approximately half (66 firms) of the 134 firms that contributed to OSS projects on their own behalf authorised their programmers to collaborate autonomously on external OSS projects during working hours. There were no significant differences in terms of country and size distributions ($\chi^2(3) = 5.33$ and $\chi^2(3) = 4.82$, respectively) between the OSS firms that adopted this practice and those that did not.

Methodology of the econometric analysis

The aim of the econometric analysis was to study the firm-level antecedents of the OSS firms' decisions to authorise programmers to contribute autonomously to external OSS projects during working hours. The dependent variable (*Adoption of the practice*) is a dummy that is equal to 1 for the firms that adopted this organisational practice and 0 otherwise. As mentioned above, the question regarding the adoption of this practice was posed only to the OSS firms that contributed to OSS projects on their own behalf. Thus, the presence of *non-ignorable missing data* (Allison, 2002) may engender a selection bias (Heckman, 1976). To address this problem, we used the two-stage correction of the probit model, as proposed by Heckman (Heckman, 1981). In the first stage, we estimated a probit selection model for the entire sample of OSS firms to explain the probability that an OSS firm has ever contributed to an OSS project on its own behalf. In the second stage, we focused on the subsample of firms that contributed to OSS projects on their own behalf. We estimated a probit model that explains the probability that a firm adopted the organisational practice under consideration, and we inserted into 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 the probability of authorising programmers to contribute autonomously to external OSS projects during working hours.

A summary of the explanatory variables is reported in Table 2. According to Hypothesis H1, if OSS firms exclusively offer OSS solutions and write code within the OSS projects to which they contribute on their own behalf, these firms are less likely to authorise their programmers to contribute to external OSS projects during working hours. We inserted into the model specification a dummy variable (*Insider*) equal to one for these firms. We predicted a negative coefficient for this variable. We tested Hypothesis H2 by including in the set of regressors the logarithm of the number of OSS projects to which the firm contributed on its own behalf since the beginning of its OSS activities as a measure of the firm's experience in collaborating with the OSS community (*Experience*). Because we expected a curvilinear relationship between *Experience* and the likelihood of adopting the practice, we also included the squared value of this variable. We predicted a positive coefficient for *Experience* and a negative one for its squared value. The scope of the firms' OSS-based activities was proxied by the number of different OSS-based activities in which the OSS firms were involved (*Scope of OSS-based activities*). We considered the five OSS-based activities described in Table 1. We predicted a positive coefficient for this variable (Hypothesis H3). Finally, we tested Hypothesis H4 by including a measure of the size of firms' operations (*Firm size*) in the

Table 2. Definition of explanatory variables and controls

Variable	Description
Insider	1 if the firm exclusively offers OSS solutions (i.e., it does not offer proprietary solutions) and develops code within OSS projects; 0 otherwise
Experience	Logarithm of the number of OSS projects to which the firm contributed on its own behalf since the start of its OSS activities
Squared value of experience	Squared value of the logarithm of the number of OSS projects to which the firm contributed on its own behalf since the start of its OSS activities
Scope of OSS-based activities	Number of different OSS-based activities in which the firm is involved. We considered the following five activities: i) offering of high-level services on pre-packaged OSS products, ii) adapting pre-existing OSS programmes and solutions to suit the needs of the firm's customers, iii) integrating OSS modules and programmes into new solutions that are released under OSS licences, iv) designing and developing solutions on order for firms' customers that are released under OSS licences, and v) designing and developing new solutions from scratch that are released into the market under OSS licences
Firm size (ln)	Logarithm of firm size measured by the total number of owner-managers, employees and freelancers
Firm age	Firm's age on the survey date
Server products	1 if the firm develops Web servers or other types of servers; 0 otherwise
Web products	1 if the firm develops solutions for e-commerce, e-mail clients, instant messaging, Web browsers, digital signature or content management systems, or e-learning tools; 0 otherwise
Network infrastructure products	1 if the firm develops back-up systems, firewalls, antispam or antivirus solutions or user and identity management software; 0 otherwise
Other products	1 if the firm develops management or data management software, workflow systems or office automation packages; 0 otherwise
Finland	1 for Finnish firms; 0 for remaining firms
Germany	1 for German firms; 0 for remaining firms
Italy	1 for Italian firms; 0 for remaining firms
Inverse Mill's ratio	Inverse Mill's ratio obtained from the estimates of the selection equation

regressors. *Firm size* is calculated as the logarithm of the total number of the firm's owner-managers, employees, and freelancers (in full-time equivalents) in 2004. We expected *Firm size* to be positively related to the adoption of the practice.

We inserted several controls in the model specification: firm age in 2003, four product category dummies and three country dummies (being located in the Iberian Peninsula, i.e., in Spain or Portugal, 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). These variables are as follows: firm age, a dummy variable that is equal to one if agreement with the values of the OSS movement (i.e., championing the exchange of technical knowledge and sharing of code among the OSS community's members Zeitlyn, 2003) was indicated by the respondents as one of the three main motives for their firms' involvement in the OSS arena, the percentage of the firms' sales generated by OSS in the year 2003, and country dummies. The estimates of the selection equation are reported in the Appendix (Table A1).

We are aware that implementing the Heckman sample selection procedure is susceptible to identification problems. Identification only occurs if the residuals of the selection and outcome

equations are distributed with a bivariate normal distribution and is essentially possible because of the non-linearity in the selection equation. However, if the selection function is roughly linear in parts of its domain, it will be difficult to obtain precise estimates in the outcome equation because of multicollinearity problems and large standard errors. Hence, because of these identification issues, it is recommended that the selection equation includes at least one independent variable that satisfies the following two conditions: (i) the variable explains selection; and (ii) the variable is unrelated to the outcome (Sartori, 2003). Our selection equation contains a variable (i.e., the percentage of sales generated by OSS) that satisfies both conditions. This variable is related to the probability that an OSS firm has ever contributed to an OSS project on its own behalf (see Table A1). Conversely, this variable is unrelated to the adoption of the organisational practice under consideration. If we include the percentage of sales generated by OSS in the outcome equation, the coefficient of this variable is not significant (see Table A2).

Table 3 provides the descriptive statistics and the correlation matrix of the explanatory variables included in the models. Because the correlations among the variables were low, we ruled out the possibility of multicollinearity problems.

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 to test our theoretical hypotheses. For this latter model, we also calculated the marginal effects of a one-unit change in any explanatory variable on the probability of adopting the organisational practice under investigation. The marginal effects were calculated based on the following: the number of OSS-based activities in which the OSS firms are involved at the sample medians; firm experience in collaborating with the OSS community, firm size, firm age and the Inverse Mill's ratio at the sample means; the *Insider* dummy, the four product category dummies and the dummy Italy at one; and the remaining country dummies at zero.

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

With regard to the estimates of Model 2, the signs of the coefficients of the control variables did not differ from those of Model 1. The exception was the dummy Germany, which was significant at 1 per cent and exhibited a greater coefficient with respect to Model 1. More interestingly, the insertion of the covariates that reflect our theoretical arguments into the model specification substantially improved the explanatory power of the model, as documented by the increase of both the McFadden's R^2 (from 0.055 to 0.193) and the proportion of correct predictions (from 60 per cent to 67 per cent).

Let us now examine the results related to the individual hypotheses. First, let us focus on Hypothesis H1, which claims that the OSS firms that offer exclusively OSS-based solutions and write code within the OSS projects to which they contribute on their own behalf, being insiders within the OSS community are less likely to authorise their programmers to contribute to external OSS projects during working hours than the remaining OSS firms. In line with this prediction, the coefficient of the *Insider* dummy variable was negative. However, the coefficient was only close to statistical significance. To obtain further insights into this, we then replaced the *Insider* dummy variable with another dummy that equals 1 if the firm offered *mainly* (rather than exclusively) OSS solutions and wrote code within OSS projects. In line with H1, the coefficient of this latter variable was negative and significant at 5 per cent. Hence, Hypothesis H1 was (weakly) supported.[9]

Conversely, the experience that an OSS firm had in collaborating with the OSS community was found to significantly influence the probability that the firm's programmers were authorised to

[9] In these estimates the remaining results are very close to those that will be illustrated below. For the sake of synthesis, they are not reported in the paper and are available from the authors upon request.

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
Insider	104	0.221	0.417	0.000	1.000	1.000												
Experience	104	1.520	0.893	0.000	5.017	0.178	1.000											
Squared value of experience	104	3.099	4.143	0.000	25.173	0.090	0.955	1.000										
Scope of OSS-based activities	104	3.962	1.123	1.000	5.000	0.226	0.315	0.256	1.000									
Firm size (ln)	104	2.210	1.128	0.000	5.991	−0.101	0.324	0.351	0.011	1.000								
Firm age	104	5.808	5.028	0.000	22.000	−0.202	0.034	0.051	−0.246	0.323	1.000							
Server products	104	0.808	0.396	0.000	1.000	0.025	0.159	0.142	−0.017	−0.068	0.035	1.000						
Web products	104	0.788	0.410	0.000	1.000	0.049	0.258	0.206	0.130	−0.076	0.013	0.524	1.000					
Network infrastructure products	104	0.846	0.363	0.000	1.000	0.035	0.186	0.148	0.200	0.019	−0.208	0.468	0.366	1.000				
Other products	104	0.865	0.343	0.000	1.000	0.007	0.073	0.081	0.012	0.076	0.086	0.379	0.348	0.222	1.000			
Finland	104	0.154	0.363	0.000	1.000	0.158	−0.069	−0.051	−0.033	−0.057	−0.234	−0.198	−0.236	−0.040	−0.066	1.000		
Germany	104	0.038	0.193	0.000	1.000	−0.107	0.244	0.279	0.096	0.176	0.248	−0.029	−0.019	−0.053	−0.068	−0.085	1.000	
Italy	104	0.500	0.502	0.000	1.000	−0.070	−0.211	−0.234	−0.086	−0.344	−0.035	−0.098	−0.094	−0.213	−0.169	−0.426	−0.200	1.000
Inverse Mill's ratio	104	0.725	0.374	0.116	1.675	1.000	−0.350	−0.316	−0.276	−0.359	0.152	0.505	−0.078	−0.206	−0.269	−0.097	−0.160	−0.083

Table 4. Determinants of the adoption of the organizational practice under investigation

		Probit equation Dependent variable: Adoption of the practice			
		Model 1		**Model 2**	
a_0	Constant	−0.986 (0.570)	*	−1.381 (1.249)	***
a_1	Insider	−		−0.558 (0.352)	
a_2	Experience	−		0.919 (0.536)	*
a_3	Squared value of experience	−		−0.353 (0.119)	***
a_4	Scope of OSS-based activities	−		0.314 (0.146)	**
a_5	Firm size (ln)	−		0.270 (0.142)	*
a_6	Firm age	0.027 (0.030)		0.016 (0.039)	
a_7	Server products	0.043 (0.351)		0.010 (0.436)	
a_8	Web products	0.292 (0.364)		0.328 (0.451)	
a_9	Network infrastructure products	0.156 (0.324)		0.367 (0.402)	
a_{10}	Other products	0.078 (0.320)		0.250 (0.432)	
a_{11}	Finland	1.072 (0.372)	***	1.426 (0.487)	***
a_{12}	Germany	0.687 (0.442)		2.400 (0.685)	***
a_{13}	Italy	0.370 (0.293)		0.474 (0.361)	
a_{14}	Inverse Mill's ratio	−0.137 (0.361)		−0.174 (0.502)	
Number of observations		134		104	
Wald χ^2 test: $a_2 = a_3 = 0$		−		16.65(2)***	
McFadden's R^2		0.055		0.193	
% correct predictions		60.4		67.3	

*Significance level greater than 10%; **significance level greater than 5%; ***significance level greater than 1%. Robust standard errors and the number of restrictions are in parentheses.

contribute autonomously to external OSS projects during working hours. The null hypothesis that the coefficients of *Experience* and its squared value were jointly equal to zero was rejected at conventional confidence levels by a Wald test ($\chi^2(2) = 16.65$). The relation between firm experience and the probability of adopting the practice under consideration was inverse U-shaped, as expected, with *Experience* and its squared value exhibiting a positive coefficient and a negative one, respectively. The probability of adopting the practice first increased with firm experience until approximately four OSS projects and then decreased as the number of OSS projects to which the firm had contributed on its own behalf since the beginning of its OSS activities grew larger. Hence, Hypothesis H2 was supported.

Moreover, the probability of adopting the practice was higher the wider the scope of the firms' OSS-based activities and the larger the firms' size. The scope of the firms' OSS-based activities had a significant (at 5 per cent) and positive coefficient. Hence, Hypothesis H3 was supported. The addition of one activity to a firm's set of OSS-based activities led to an 11.3 per cent increase in the probability of granting the firm's programmers the freedom to contribute autonomously to external OSS projects during working hours. The coefficient of *Firm size* was positive and weakly significant (at 10 per cent). The marginal effect of this variable, which roughly corresponds to an increase equal to the standard deviation, was 9.7 per cent. Therefore, we found (weak) support for Hypothesis H4.

Discussion and conclusions

Because of the rapid pace of technological change and knowledge creation, modern firms cannot rely solely on internally generated knowledge to compete and innovate within the current knowledge-based society. Accordingly, a firm's ability to in-source knowledge from outside of its boundaries plays a crucial role. Relevant knowledge is dispersed across many different *loci* and may reside with competitors, customers, suppliers, universities and communities of users and developers. These communities are unanimously considered by both practitioners and scholars (Jeppesen and Frederiksen, 2006) to be rich and varied knowledge pools that firms can access at no (monetary) cost. Nevertheless, communities have peculiar characteristics that make it challenging for a firm to in-source valuable knowledge from them. Communities produce knowledge in a decentralised fashion, and their members usually share specific behavioural norms while strictly adhering to ethical values (O'Mahony and Bechky, 2008).

In this paper, we focused on a renowned community of users and developers (i.e., the OSS community Lerner and Schamkerman, 2010) and on the firms collaborating with it (OSS firms). We studied the adoption of a widespread organisational practice that allows OSS firms to in-source relevant knowledge from the OSS community. The practice consists of authorising firm programmers to contribute autonomously to *external OSS projects* (i.e., OSS projects in which the firm does not contribute on its own behalf) during working hours. We relied on the absorptive capacity literature and argued that the practice under investigation is intended to leverage the individual-level abilities of firm programmers to acquire and assimilate knowledge from the OSS community. In turn, these abilities stem from the programmers' personal specific knowledge of the OSS community. Accordingly, we predicted that the adoption of this practice is more likely under the following conditions: i) an OSS firm lacks *firm-level potential absorptive capacity* and is therefore forced to rely on the individual-level absorptive abilities of its programmers; and ii) the firm has *realised absorptive capacity* that allows it to transform and exploit the new knowledge allegedly in-sourced by its programmers while contributing autonomously to external OSS projects.

Our results largely support our predictions. We find that the practice is less likely to be adopted if an OSS firm has substantial experience in collaborating with the OSS community and thus has adequate firm-level potential absorptive capacity. The same applies to the firms that operate as "insiders" within the OSS community as they mainly offer OSS-based software solutions (as opposed to proprietary solutions) and develop code within OSS projects. Moreover, the likelihood that the practice under investigation will be adopted is higher for larger OSS firms, though the evidence in support of this argument is weak, and for the OSS firms that have a wider portfolio of OSS-based activities. These firms are more capable of transforming and exploiting the new knowledge in-sourced by programmers from external OSS projects (i.e. they have larger realised absorptive capacity).

This paper makes several contributions to the literature. First, it adds to the absorptive capacity literature. Specifically, few studies have investigated the role of individual employees and firms' organisational design in developing absorptive capacity. Our paper helps to fill these gaps. We focus on the role of programmers' individual-level abilities to acquire and assimilate external knowledge from the OSS community, and we study an organisational practice that is intended to leverage these abilities. Our findings suggest that the individual-level absorptive abilities of employees may substitute for the lack of firm-level potential absorptive capacity and complement firm-level realised absorptive capacity.

Second, our work contributes to the organisational design literature. The adoption of new organisational practices that grant greater autonomy to employees has attracted increasing interest from organisational design scholars. Most previous studies (Ichniowski et al., 1996; Osterman, 1994) have analysed the adoption of workplace practices, such as quality circles, total quality management and job rotation among employees. Recently, scholars have devoted attention to organisational practices aimed at fostering proficient collaborations with third parties, particularly customers

(Foss et al., 2011). Our paper extends this line of research to collaborations with communities of users and developers.[10]

Third, this paper provides interesting additions to the literature on OSS. We set aside the community-centred perspective adopted by the bulk of this literature (O'Mahony and Ferraro, 2007) and put the firm and its internal organisation at the core of the analysis. Moreover, we examine an interesting aspect within the overarching question of how software firms take advantage of the OSS community. Previous contributions have focused on the definition and description of OSS business models or have investigated whether and how firms actively participate in the production of OSS. The analyses of how firms organise to leverage the knowledge generated by the OSS community are scant and mainly phenomenon-based. The work by Dahlander and Magnusson published in this journal in 2008 is a prominent contribution to this research stream in that it examines how firms adopt *tactics* for *accessing, aligning and assimilating* knowledge created by the OSS community. Our paper moves a step further and uses arguments from the absorptive capacity literature to provide original insights into how firms organise to in-source this knowledge. In addition, whereas most of the studies on knowledge sourcing from the OSS community are based on case studies, our work provides quantitative results generated by rigorous econometric work.

We acknowledge that the paper has some limitations that open avenues for future research. First, our analysis did not account for the individual characteristics of firm programmers. These characteristics influence the benefits of the practice and thus impact the probability of its adoption. Because different programmers may possess specific knowledge of the OSS community to a different extent, they may have heterogeneous individual-level absorptive abilities. Programmers' specific knowledge of the OSS community depends on their individual skills and behaviours. For instance, the case studies we conducted documented that some programmers voluntarily contribute to OSS projects during their spare time as a habit. These programmers are likely to possess superior specific knowledge of the OSS community and therefore have greater individual-level absorptive abilities. Thus, firms should consider their programmers' individual characteristics and grant programmers autonomy on a selective basis.

Second, our analysis did not capture *to what extent* firm programmers were allowed to contribute to external OSS projects during working hours. There might be substantial heterogeneity across the sample firms with respect to the amount of time that firm programmers were allowed to devote to contributing to external OSS projects. It would be interesting to extend our work by investigating whether the antecedents of the decision to adopt the organisational practice differ from the determinants of the extent to which it was used.

Third, we did not have access to longitudinal data. Thus, we could not account for the influence of *inertia* on the adoption of the organisational practice under investigation. As prior studies have shown (Colombo and Delmastro, 2008; Ichniowsky and Shaw, 1995), firms tend to use a given organisational practice because it was used in the past, even if changes in internal and external contingencies have rendered the practice suboptimal. In the setting of our study, inertia may have led us to underestimate the negative impact of the experience obtained by OSS firms in their collaborations with the OSS community on the adoption of the practice. Specifically, a rather inexperienced (but not completely novice) OSS firm might have granted autonomy to its programmers to participate in external OSS projects in the past to leverage their abilities to acquire and assimilate knowledge from the OSS community. As time passes, the OSS firm obtains experience in collaborating with the OSS community and thereby develops firm-level potential absorptive capacity. As a consequence, the individual-level absorptive abilities of the firm's programmers become less valuable to the firm. Although the firm would benefit from revoking the autonomy granted to its programmers, it may fail to do so because of inertia.

Fourth, we focus on OSS firms' adoption of one specific organisational practice. For these firms to effectively absorb knowledge from the OSS community, the practice under investigation should

[10] See e.g. Alexy, 2011 for a recent exception.

be accompanied by other complementary organisational arrangements. Notably, OSS firms should design suitable knowledge-sharing practices so that the programmers can effectively transmit the knowledge that they have in-sourced from external OSS projects to their colleagues. Jointly exploring the antecedents of these organisational arrangements would be an interesting extension of this research.

These limitations do not diminish the managerial relevance of our work. The practice under investigation reverberates in the business and technical press and in practitioners' conversations. It has been adopted both by established companies (e.g., Google and 3M) and small entrepreneurial ventures, as documented by our case studies. In sum, our work is of great interest to the managers who face the challenge of designing how to organise their firms to in-source external knowledge from outside of their boundaries.

The key message offered by the present paper to practitioners is that a *one-size-fits-all* recipe does not exist because of the peculiar characteristics of the OSS community, which combine with firm-specific characteristics in idiosyncratic ways. Granting programmers the autonomy to contribute to external OSS projects during working hours may enable the firm to leverage its programmers' individual-level absorptive abilities when scouting the OSS community for relevant knowledge. For an inexperienced (but not totally novice) OSS firm that is not embedded within the OSS community but has a large scale of operations and a diversified set of OSS-based activities, the adoption of this practice is likely to provide substantial benefits. Conversely, for small OSS firms with a narrow set of OSS activities that either had the opportunity to obtain experience in collaborating with the OSS community or that are *insiders* in the OSS community, adopting this practice risks wasting their programmers' time without providing any concrete advantage. However, when evaluating this risk and deciding whether to adopt the practice, a manager should also account for the fact that this practice can serve purposes different from those considered here. Although we focused on the firm's ability to scout the OSS community for new knowledge, the practice may also have positive effects on employees' job satisfaction and may have a positive signalling effect on the OSS community. Both of these dimensions are likely to result in an increase in the firm programmers' productivity that (partly) balances the opportunity cost of the time spent freely scouting the OSS community.

Finally, it is worth noting that managers can learn from the results presented in this paper when designing how to organise their firms for sourcing external knowledge from third parties that share commonalities with communities of users and developers. Collaborations with universities are a telling example. The principles ruling open science have similarities with those of the OS movement. Open science is inspired by the values of knowledge sharing and dissemination, whereas university researchers' *taste for science* mimics the *fun to programme* (Diamond and Torvalds, 2001) of OSS developers. In addition, firms (especially those in high-tech industries) frequently hire employees who have previously worked as university researchers and therefore have personal knowledge of the scientific community. According to our findings, to effectively absorb knowledge from universities, firms that have limited experience with university collaborations but a high potential for transforming and commercially exploiting the technical knowledge produced by universities should grant their employees who have previously worked as university researchers the autonomy of freely scouting university research projects for new knowledge.

Acknowledgements

The authors are indebted to Andrea Bonaccorsi for allowing them to use the ELISS database. Moreover, they wish to thank Oliver Alexy, Nicolai Foss, Keld Laursen, Francesco Rullani, and the participants in the DRUID Summer Conference 2012 and in seminars held at D-MTEC ETH and University of Catania for their insightful comments. The suggestions of the two anonymous reviewers and of the associate editor were particularly helpful in preparing the final version of the paper.

Appendix

Table A1. Determinants of firms' contribution to OSS projects

		Probit selection equation Dependent variable: DPROJECTS	
a_0	Constant	−0.373 (0.242)	
a_1	Firm age	−0.040 (0.016)	**
a_2	Agreement with OSS values	0.407 (0.233)	*
a_3	Percentage of sales generated by OSS	1.575 (0.254)	***
a_4	Finland	0.074 (0.261)	
a_5	Germany	0.563 (0.321)	*
a_6	Italy	−0.318 (0.220)	
Number of observations		293	
McFadden's R^2		0.176	
% correct predictions		72.01	

*Significance level greater than 10%; **significance level greater than 5%; ***significance level greater than 1%. Robust standard errors are in parentheses.

Table A2. Determinants of the adoption of the managerial practice under investigation

		Probit equation Dependent variable: Adoption of the practice	
a_0	Constant	−3.604 (1.008)	***
a_1	Insider	−0.524 (0.348)	
a_2	Experience	0.919 (0.539)	*
a_3	Squared value of experience	−0.348 (0.119)	***
a_4	Scope of OSS-based activities	0.320 (0.147)	**
a_5	Firm size (ln)	0.266 (0.142)	*
a_6	Firm age	0.011 (0.035)	
a_7	Server products	−0.002 (0.437)	
a_8	Web products	0.346 (0.452)	
a_9	Network infrastructure products	0.389 (0.400)	
a_{10}	Other products	0.266 (0.436)	
a_{11}	Finland	1.448 (0.485)	***
a_{12}	Germany	2.443 (0.666)	***
a_{13}	Italy	0.483 (0.369)	
a_{14}	Percentage of sales generated by OSS	−0.006 (0.485)	
Number of observations		104	
McFadden's R^2		0.192	
% correct predictions		67.3	

*Significance level greater than 10%; **significance level greater than 5%; ***significance level greater than 1%. Robust standard errors and number of restrictions are in parentheses.

References

Ågerfalk, P.J., Fitzgerald, B., 2008. Outsourcing to an unknown workforce: exploring opensourcing as a global sourcing strategy. MIS Quarterly 32 (2), 385–409.

Aghion, P., Tirole, J., 1997. Formal and real authority in organizations. Journal of Political Economy 105, 1–29.
Allen, T.J., 1977. Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization. MIT Press, Cambridge.
Allison, P.D., 2002. Missing Data. Sage Publications, Thousand Oaks CA.
Alonso, R., Matouschek, N., 2008. Optimal delegation. Review of Economic Studies 75, 259–293.
Alonso, R., Dessein, W., Matouschek, N., 2008. When does coordination require centralization? American Economic Review 98, 145–179.
Argote, L., 1999. Organizational Learning: Creating, Retaining, and Transferring Knowledge. Knulwer, Norwell, MA.
Baker, G., Gibbons, R., Murphy, K.J., 1999. Informal authority in organizations, Journal of Law. Economics & Organization 15, 56–73.
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., Scateni, A., 2005. ELISS Report. Rapporto del progetto ELISS. European Libre Software Survey MIMEO.
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.
Capra, E., Francalanci, C., Merlo, F., Rossi-Lamastra, C., 2010. Firms' involvement in Open Source projects: A trade-off between software quality and success. Journal of System and Software.
Carpenter, R.E., Petersen, B.C., 2002. Is the growth of small firms constrained by internal finance? The Review of Economics and Statistics 84, 298–309.
Chan, J., Husted, K., 2010. Dual allegiance and knowledge sharing in Open Source software firms. Creativity and Innovation Management 19 (3), 314–326.
Chandler, A.D., 1994. Scale and scope: the dynamic of industrial capitalism. Harvard University Press, Cambridge, MA.
Chesbrough, H., 2010. Business model innovation: Opportunities and barriers. Long Range Planning 43 (2–3), 354–363.
Colombo, M.G., Delmastro, M., 2008. The Economics of Organizational Design: Theoretical Insights and Empirical Evidence. Palgrave Macmillan.
Colombo, M.G., Foss, N.J., Rossi-Lamastra, C., 2012. Organizational design for absorptive capacity linking individual and organizational levels. MIMEO.
Dahlander, L., Magnusson, M.G., 2005. Relationships between open source software companies and communities: Observations from Nordic firms. Research Policy 34 (4), 481–493.
Dahlander, L., O'Mahony, S., 2011. Progressing to the center: Coordinating project work. Organization Science 22 (4), 961–979.
Dahlander, L., Wallin, M., 2006. A man on the inside: Unlocking communities as complementary assets. Research Policy 35 (8), 1243–1259.
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.
Dessein, W., 2002. Authority and communication in organizations. Review of Economic Studies 69, 811–838.
Diamond, D., Torvalds, L., 2001. Just for fun: The story of an accidental revolutionary. HarperBusiness, New York, NY.
Eilhard, J., Firms on SourceForge, MPRA Paper No. 78092008 (2008).
Feller, J., Fitzgerald, B., 2002. Understanding Open Source Software Development. Addison Wesley, Boston, MA.
Finegold, D., Frenkel, S., 2006. Managing people where people really matter: the management of human resources in biotech companies. The International Journal of Human Resource Management 17 (1), 1–24.
Fitzgerald, B., 2006. The transformation of Open Source software. MIS Quarterly 30 (3), 587–598.
Foss, N.J., Minbaeva, D.B., Pedersen, T., Reinholt, M., 2009. Encouraging knowledge sharing among employees: How job design matters. Human Resource Management 48 (6), 871–893.
Foss, N.J., Laursen, K., Pedersen, T., 2011. Linking customer interaction and innovation: The mediating role of new organizational practices. Organization Science 22 (4), 980–999.
George, G., 2005. Slack resources and the performance of privately held firms. Academy of Management Journal 48 (4), 661–676.

Gottschalg, O., Zollo, M., 2007. Interest alignment and competitive advantage. Academy of Management Review 32 (2), 418–437.
Gruber, M., Henkel, J., 2006. New ventures based on open innovation - an empirical analysis of start-up firms in embedded Linux. International Journal of Technology Management 33 (4), 256–372.
Hars, A., Ou, S., 2002. Working for free? Motivations for participating in Open-Source projects. International Journal of Electronic Commerce 6 (3), 25–39.
Heckman, J.J., 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, J.J., 1981. The incidental parameters problem and the problem of initial conditions in estimating a discrete time–discrete data stochastic process. In: Manski, C.F., McFadden, D. (Eds.), Structural Analysis of Discrete Data with Econometric Applications. MIT Press, pp. 179–195.
Henkel, J., 2006. Selective revealing in open innovation processes: The case of embedded Linux. Research Policy 35 (7), 953–969.
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.
Ichniowski, C., Kochan, T., Levine, D., Olson, C., Strauss, G., 1996. What works at work. Industrial Relations 35, 299–333.
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.
Jensen, M., Meckling, C., 1992. Specific and general knowledge, and organizational structure. In: Werin, W.H.L., Wijkander, H. (Eds.), Contract Economics. Blackwell, Cambridge, Mass. and Oxford, pp. 251–274.
Jeppesen, L.B., Frederiksen, L., 2006. Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. Organization Science 17 (1), 45–64.
Knez, M., Simester, D., 2001. Firm-wide Incentives and mutual monitoring at continental airlines. Journal of Labor Economics 19 (4), 743–772.
Lakhani, K.R., von Hippel, E., 2003. How open source software works: "free" user-to-user assistance. Research Policy 32 (6), 923–943.
Lakhani, K., Wolf, R., 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In: Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K. (Eds.), Perspectives in free and open source software. MIT, Cambridge.
Lerner, J., Schamkerman, M., 2010. The comingled code, MIT Press, Cambridge, MA.
Lerner, J., Tirole, J., 2002. Some simple economics of Open Source. Journal of Industrial Economics 50 (2), 197–234.
Lerner, J., Tirole, J., 2005. The scope of OSS licenses. Journal of Law, Economics, and Organization 21 (1), 20–56.
Matusik, S.F., Heeley, M.B., 2005. Absorptive capacity in the software industry: identifying dimensions that affect knowledge and knowledge creation activities. Journal of Management 31 (4), 549–572.
O'Mahony, S., Bechky, B.A., 2008. Boundary organizations: Enabling collaboration among unexpected allies. Administrative Science Quarterly 53 (3), 422–459.
O'Mahony, S., Ferraro, F., 2007. The emergence of governance in an open source community. Academy of Management Journal 50, 1079–1106.
O'Mahony, S., Ferrero, F., 2007. The emergence of governance in an Open Source community. Academy of Management Journal 50 (5), 1079–1106.
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.

Raymond, E.S., 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.
Sartori, A., 2003. An estimator for some binary-outcome selection models without exclusion restrictions. Political Analysis 11, 111–138.
Shah, S.K., 2006. Motivation, governance, and the viability of hybrid forms in Open Source software development. Management Science 52 (7), 1000–1014.
Shapiro, C., Varian, H.L., 1999. Information Rules: a Strategic Guide to Network Economics. Harvard Business School Press, Cambridge, MA.
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.
Stein, J.C., 2002. Information production and capital allocation: Decentralized versus hierarchical firms. Journal of Finance 57, 1891–1921.
Teece, D.J., 1986. Profiting from technological innovation: implications for integration, collaboration, licensing, and public policy. Research Policy 15, 285–305.
Volberda, H.W., Foss, N.J., Lyles, M.A., 2010. Absorbing the concept of absorptive capacity: How to realize its potential in the organization field. Organization Science 21 (4), 1–21.
von Hippel, E., von Krogh, G., 2003. Open source software and the "private-collective" innovation model: Issues for organization science. Organization Science 14, 209–225.
von Hippel, E., 2006. Democratizing innovation. MIT Press, Cambridge, MA.
G. von Krogh, C. Rossi-Lamastra and S. Haefliger, Phenomena-based research in management and organization science. The case of Open Source Software, *Long Range Planning*, forthcoming (2011).
G. von Krogh, S. Haefliger, S. Spaeth and M.W. Wallin, Carrots and rainbows: Motivation and social practice in open source software development, *MIS Quarterly*, forthcoming (2012).
Zahra, S.A., George, G., 2002. Absorptive capacity: A review, reconceptualization, and extension. Academy of Management Review 27 (2), 185–203.
Zeitlyn, D., 2003. Gift economies in the development of open source software: Anthropological reflections. Research Policy 32 (7), 1287–1291.

Biographies

Massimo G. Colombo is Full Professor of Economics of Technical Change at Politecnico di Milano. His current research interests are mainly in economics of innovation and entrepreneurship. He serves as Associate Editor of the Journal of Small Business Manager and he is part of the editorial board of the Journal of Management Studies and the Journal of Technology Transfer. He has edited several special issues of international journals. He has published more than 60 papers in refereed international journals and four books. He is the author of *The economics of organizational design*, published by Palgrave MacMillan. E-mail: massimo.colombo@polimi.it

Evila Piva is assistant professor at Politecnico di Milano, Department of Management Economics and Industrial Engineering. Her research interests are in the area of Entrepreneurship and Economics and Management of Innovation. Her research activities focus mainly on the analysis of new technology-based firms. Her current areas of research include: firm strategic alliances and collaborations with communities; the characteristics of academic start-ups; the re-organisation after mergers and acquisitions. She has published several papers in international peer-reviewed journals (*Research Policy, Strategic Entrepreneurship Journal*, and *IEEE Transactions of Engineering Management*, among others). E-mail: evila.piva@polimi.it

Cristina Rossi-Lamastra is assistant professor at Politecnico di Milano, Department of Management Economics, and Industrial Engineering. She holds a PhD in Economics of Innovation from Sant'Anna School of Advanced Studies in Pisa. Her research interests are in the area of Entrepreneurship and Economics and Management of Innovation. At present, her research activities focuses mainly on the analysis of young entrepreneurial ventures operating in high-tech industries. Her studies in Open Source Software deal mainly with firms' involvement in the OSS movement (business models, incentives, project participation, innovation processes). She has published on these topics in *Management Science*, *Long Range Planning, R&D Management*, *Research Policy*, among others. E-mail: cristina1.rossi@polimi.it