TITLE: Dilemmas within Commercial involvement in Open Source Software.

Structured Abstract

Purpose – This paper seeks to contribute to the literature on commercial involvement in open source software, the levels of this involvement and the consequences of attempting to mix various logics of action.

Design/methodology/approach – This paper uses a case study approach based on mixed methods: literature reviews and news searches, electronic surveys, qualitative interviews and observations. It combines findings from several research projects as well as previous publications to present the scope of commercial options within open source software projects and their consequences.

Findings – Our findings show that higher levels of involvement in open source software communities poses important questions about the balance between economic, technological and social logics as well as about the benefits of being autonomous, having access to collaborative networks and minimizing risks related to free-riding. There are 6 levels of commercial involvement in open source and each of them is characterized by a different dilemma.

Originality/value – The paper sheds light on the various level of business involvement in the open source movement and emphasize that the popular 'open innovation' concept is only the first step in real involvement and paradigm change.

Keywords – commercialization, open source, open innovation, contradictory institutional logics

Paper type - Research paper

Introduction

Software development began in the U.S.A. in the 1950s and over the following decades a global organizational field emerged: a community of organizations and individuals that subscribes to a common meaning system in which participants interact with one another more than with actors external to the field (Scott, 1994, pp. 207-208). The field came to consist of a meaning system, which perceived software as a technical device. Participants saw it as natural that software development took place within open innovative communities in which professional developers and users shared knowledge about software products across private and public organizational boundaries. In the 1970s private companies began to introduce a different meaning system, which found it natural to understand software development as a commodity subject to proprietary rights.

Within these two organizational fields or meaning systems different software products were developed, and the fields engaged in fierce political debates over how knowledge should be shared and money earned within software (Weber, 2004): on the one hand was an organizational field that perceived the development of software within open communities focusing little or not at all on earnings as natural, and on the other an organizational field that continued to consider it natural to commercialize and patent software. In popular terms the two original fields may be called the 'copyleft' and the 'copyright' fields (Gehring, 2006).

Since the middle of the 1990s, companies have begun to get involved in open source software communities - combining commercialization and copyleft-modes of innovation (see for example Lerner and Tirole, 2002; Grand *et al.*; 2004; Langdon and Hars, 2007; Fosfuri, Giarratana and Luzzi, 2008; Ciesielska, 2010; Westenholz, 2012; Ciesielska and Petersen, 2013). As this leads to crossing private and collective models of agency (Ulhøi, 2004), these companies have been facing a dilemma: if the company builds a strong relationship with an open source software community, it needs to respect the norms in the community, and that may limit its room for maneuvering. On the

other hand, the company will get the benefits of support and innovation from the community. If a company moves away from the community, it may be easier to make a profit, however it will not obtain support and innovation from the community. The term "open innovation" was coined by Chesbrough and extensively described in his publications (2003; 2006; 2011) as only the first step in real firm involvement and paradigm shift towards a more open model. The 'open' here means that there are many ways for ideas to flow into the process and out into the market, for example by duplicating incentives, knowledge sharing within the firm or alliances. The path to open innovation leads through redesigning business model and very strong intellectual property management (Chesbrough, 2006). The community can both bring external ideas to the company ("outside-in") and take their unused ideas outside ("inside-out") (Chesbrough, 2006). Another very particular form of open innovation is involvement of the (lead) users in enhancing or creating new products and services (von Hippel, 1976, 2005; von Hippel *et al*., 2011).

Companies have been dealing with the control vs. openness dilemma in different ways. In the paper we develop a six-step graduation of business involvement in open source software (OSS) communities. We start by reviewing the existing literature on how companies have been dealing with the "opening up" strategy. We systematize the literature into six degrees of business involvement in OSS. We present our methods, then illustrate the three lower degrees of involvement with secondary data, focusing on the three higher levels for which we examine longitudinal case studies. In the concluding section we review the empirically observed levels in the context of the existing literature and conclude with a discussion of the key dilemmas faced by commercial organizations.

Literature review

OSS is identified as an increasingly important business model (Feller and Fitzgerald, 2002; Bonaccorsi *et al*., 2006). Lerner and Tirole (2002) described how since the turn of the century numerous major corporations, including Hewlett Packard, IBM, and SUN have launched projects to develop and use open source software. Other companies such as Red Hat and VA Linux have specialized in commercializing Linux, and yet another group of open source software companies have received venture capital financing. They argue that companies may employ two different strategies to deal with the interface between open (copyleft) and closed (copyright) source software development: they either imitate some aspects of open source processes, or mix open and closed processes. If companies follow the first strategy they do not get involved with open source software communities, as they will not allow users to modify their code. But these companies may, to some extent, duplicate the incentive by letting core developers work with both proprietary and open source. They may also imitate the idea of open source code sharing within the company, and/or they may involve their customers in the development of their product (Langdon and Hars, 2007). In the second strategy, companies become involved with open source software communities in various ways. They may choose a reactive relationship with open source communities by allocating programmers to an open source software project. Or they may choose a more proactive strategy by releasing code and creating a governance structure for the resulting open source process. For example Fosfuri, Giarratana and Luzzi (2008) discuss how firms are responding to open source software by adjusting to, resisting, or supporting it, dependent on the stock of patents and trademarks that firms have accumulated before commercializing their first OSS product.

Capra *et al.* (2009) substantiate the second strategy defined by Lerner and Tirole in a survey on companies' participation in open source community projects. Capra *et al.* argue that a distinction should be made between a) open source projects, which are led entirely by a company, and b) open source projects, which are led by community members. The researchers are primarily interested in

the latter situation where companies become involved in a community project. They argue that companies are profit-oriented agents and that the primary goal is to get tangible benefits from participation in the open source project. Companies may follow different models of participation in open source communities: they may participate in creating codes; they may support the project that creates codes in other ways, e.g. they may provide financial and logistical support; and they may engage in administrative or managerial work within the project in order to orient the product in a competitive way for the company.

Simultaneously, different ways of combining business and open source software community projects and new business models have been created (Krishnamurty, 2005; Weber, 2004). For example Fitzgerald (2006) makes a distinction between business strategies within OSS 2.0 and Free & Open Software (FOSS). OSS allows to view the source code, but not necessarily use it free of charge. FOSS development processes are characterized by value-added service-enabling and loss-leader/market-creating. In the latter strategy the open source product is sold for free, but with the end goal of enlarging the market for a closed sourced products and services. Grand *et al.* (2004) further investigate the question of how companies engage in the creation of open source software, developing a four-level model of companies' resource allocation. At level 1, companies primarily use rather than develop the software. However, this is not completely costless, as companies need technological expertise for installation and integration into the existing IT environment. Hoppenbrouwers (2007) mentions that these companies – which he calls 'community customers' – sometimes become engaged with open source software communities by donating efforts around the product to the community. At level 2, companies like IBM and Sun Microsystems sell their products with open source software as a complementary asset. This may require major investment, as development may be needed to adapt code developed for other purposes. At level 3, open source software becomes a design choice for the way the companies develop specific new software. At

level 4, open source software moves from being the design choice for a specific project to the design choice for the companies' overall business model. The researchers argue that the four levels have a gift economy logic; the greater the resource investment managers have to make, the greater the potential benefits.

Grand *et al.*'s (2004) four levels deal with companies' *resource allocation* in the creation of open source software, going from minimalcosts to a total allocation of resources to open source software as the overall business model of the companies. In this paper we focus on the *involvement of companies* with open source software communities. The question addressed here is how relationships between companies and open source communities are created.

The literature on firms' involvement with open source software communities shows great variations, which we systemize in figure 1 (Westenholz, 2012, p.28)[1]. The levels 1-3 were largely described in the literature, providing various businesses as successful implementations of those strategies. Here, the authors selected the secondary cases to illustrate the point about lower levels of involvement.

Insert figure 1

A good example of the *Level 1 - "open innovation"* is General Electric's (GE) initiative on renewable energy, which was considered as a new, neighboring market to the existing core business, or so-called "adjacency" (Idelchik and Kogan, 2012). As a first step they partnered with venture capitalists who support new ventures. After GE announced two investment funds and the Ecomagination Challenge they received 4,000 ideas from entrepreneurs in 160 countries. Several of those were funded by GE directly, and those in need of further development before the commercialization stage were supported by a further 20 million USD Innovation Fund. In a second

[1] Copenhagen Business Press kindly agreed to use the figure from the book: The Janus Face of Commercial Software Communities – An investigation into institutional (non)work by interacting institutional actors.

stage the company developed a network of business scouts to tap innovation networks in Israel, Japan, and Russia, India, China, Germany, and the United States. Lastly, they applied a structured invention platform to capture the highest value from their ventures, search for the best innovations, moderate their risk, and accelerate business growth. In the opinion of the practitioners involved in the implementation of those changes in GE, Idelchik and Kogan (2012, p. 31): "tearing down the lab walls, so to speak, and collaborating with these new partners has allowed GE to see around the corner, spot new technology and business trends, and make informed strategic decisions in growing adjacencies." GE was the primary beneficiary of their investment, as the open collaboration allowed an acceleration of the scaling and commercialization of innovations and entrepreneurs helped GE to challenge assumptions about business that became institutionalized in a big corporation, while the community around the GE eco-business supported them with suggestions and best solutions (Winston, 2011).

There are also companies, which become *community customers* (Level 2). For many years Open Office was a key end-customer OSS product, which was widely adopted in governmental and local agencies, educational institutions and some private sector companies such as hotel chains, retail shops, insurance companies, or even big manufacturers (Table 1).

Insert Table 1 here

Most Open Office users have migrated from proprietary Microsoft Office products, and this happened for various reasons. For example American Health First Inc 2004 decided to implement Open Office because of lack of initial investment, which afforded 2 million dollar in savings, and similar reason are given by others (Morgan, 2012; Fitzgerald and Kenny, 2004). Moreover Health First Inc was able to save even further on other licensed products such as Adobe Acrobat and macromedia Flash, because of additional functionalities of Open Office in comparison to Microsoft's package (Stafford, 2004). However it seems that in a public domain, organizations are

also often driven by democratic values such as independence and self-determination (Cassell, 2008). At the same time implementing open source software is not problem-free. Common issues are related to unfamiliarity with the software, potentially blocked functionalities (especially in documents exported from proprietary software), difficult support for proprietary applications and finally the potential cost of implementation (Karjalainen, 2010). Therefore once initiated, further migration to other open source software packages is likely to happen (Stafford, 2004).

Level 3 are companies which are *mixing proprietary and Open Source Software solutions*. A successful and highly regarded business at this level is Open Xchange, Ltd (http://www.open-xchange.com/). This company was founded in May 2005 to continue its previous incarnation as part of the SUSE Linux Openexchange Server. The founders of the company were involved in a range of software projects – both open and closed sourced – until they decided to implement a mixed strategy (Brodkin, 2007). Open Xchange offers a collaboration platform allowing its users to share e-mails, calendars, tasks, and documents generated both by proprietary and open source software. As an integration tool it allows IT administrators to migrate systems to an open source environment as well as to create and implement applications without having to change their existing infrastructure components (Galli, 2005). Before Open Xchange the world of e-mail server software had long been dominated by Microsoft's Exchange and IBM's Domino packages, but that has changed since 1and1 Internet, the world's largest Web hosting company, decided to roll out a million e-mail accounts running on Open Xchange's open-source software (Hamm, 2007). Today the company has acquired a good reputation on the market, receiving highly positive reviews from its current customers. From 2006 the project was awarded several times, including the Best Linux Groupware Server in the first Enterprise Open Source Readers Choice Awards. Casadesus-Masanell and Llanes (2011) speculate that incompatibility between systems makes it more likely that more

firms will adopt a mixed-source business model in the future instead of a pure open source business model.

In section 4 below, the higher degrees of companies' involvement in Open Source Software (Levels 4 to 6) communities will be illustrated with empirical case studies, followed by an analytical section. But first, we describe our research methods.

Methodological notes

This paper is based on a mixed-methodology, where primary and secondary case studies are compared to formulate a framework for business involvement in open source software communities. We were primarily interested in the higher levels of involvement, which pose additional challenges for the business, and in those cases empirical data was collected. Data about the other three cases was sourced from existing publications and re-analyzed. The combination of data sources was used to widen the spectrum of discussed levels. This strategy of comparison between primary and secondary data has not been frequently used before, mostly due to potential problem with data interpretation (Gillies and Edwards, 2005). The reason for this is twofold. On the one hand, re-use of existing qualitative research for secondary analysis may be limited by a lack of fit between the original and secondary data. On the other, lack of access to the original context may further impede the analysis (Gillies and Edwards, 2005).

However, Hammerslay (2010, p.6) points out that "the fact that data are being re-used does not signal the presence of problems that never arise in the context of primary research, nor does it automatically imply that 'fit' and 'context' are going to be impossibly troublesome" although those problems may be more likely to occur. Also, accepting the interpretation risks involved, we argue

that the way we recontextualize the data (Moore, 2007) enabled us to develop a more holistic framework. Detailed strategies and methods employed are described below.

Secondary data

The cases describing levels 1 to 3 are based on secondary data: literature and news searches, companies' websites and interviews with employees and managers. The selection of cases was dictated by the clarity of relation between open and closed strategies in their business models. The choice of the cases and the accuracy of their description and representativeness were confirmed by informal consultations with the Open Source Software developers.

Primary data

The actual empirical studies were conducted to understand firms' involvement in Open Source, as this was identified as an understudied area. The cases describing levels 4 to 6 are based on longitudinal research conducted by the authors. This was part of a larger research project on Institutional entrepreneurs, but was run as independent investigations of the particular cases. The choice of cases was dictated by their accessibility and high level of company's involvement in the open source universe. The cases were brought together at the end of the Institutional Entrepreneurs project and re-discussed to reach a closer understanding of what it means for a business to be highly involved in open source and what kind of consequences may be involved. This strategy also resulted in the use of variety of methods depending on the access gained to the particular community and company, as well as the specificity and size of the project studied. For example the TYPO3 study was focused on the community of companies and contributors, while studies of GNOME and Maemo were conducted primarily from Nokia's point of view. This affected not only the varied methods used but also a specific coding sheets used for analysis. As such the cases do not claim to be representative for any industry or sector, nor to be extrapolated, but provide close

accounts of the potential scope of business involvement in open source and how this may affect the company itself.

The cases of Nokia's involvement in Maemo and GNOME are based on a three-year project involving Open-ended interviews, direct observations and documentary material (Ciesielska, 2010; Westenholz *et.al.,* 2012). In total 20 formal interviews were conducted - 10 Nokia employees and four subcontractors working on Nokia's open source and tablet activities, as well as six independent open source software contributors. Many informal communications followed and were documented in the form of field notes. Interviews were semi-structured (Fontana and Frey, 1994; Kostera, 2007), with the preliminary list of questions prepared only to initiate the conversation rather than follow them strictly (Spradley, 1979). In addition, unstructured direct observations (Agar, 1980/1996) were made on seven separate events – conferences and meetings between 2007-2009, as well as during a visit to the Nokia Research centre in Helsinki. Websites and discussion forums concerning Maemo and GNOME served as a source of netnographic data (Kozinets 1997; 1998; 2002; Langer and Beckmann, 2005) together with existing publications and public statements about Nokia's involvement in OSS documentary material (see Ghosh, 2006; Dittrich, 2007). The analysis of this material was constructed around themes of knowledge management, trust development/collapse and identity struggle. The detail code sheet can be found in Ciesielska (2010, p. 51-53).

The TYPO3 case illustrating level 6 is based on two electronic surveys, thirteen interviews with managers of contributing companies and one direct observation of a community meeting. The surveys (Marsden and Wright, 2010) were conducted in 2005 and 2006 by the authors of this article and two colleagues at the University of Copenhagen, Peter Gundelach and Benedikte Brinker. The aims of the surveys were to gain knowledge of the community and how the companies became involved with the community. The 2005 survey was carried out electronically (Sheehan 2001) to all 5.155 members of the community defined as the participants on all TYPO3 mailing lists and

newsgroups around the world. 1,675 (32.5 %) of the questionnaires were returned, which is a fairly high response rate compared to other e-mail surveys. The 2006 questionnaire was electronically mailed to 1.110 TYPO3 firms listed on TYPO3 homepages. Half of all the companies that have been approved as consultancy companies by the TYPO3 Association participated in the survey; self-listed companies, however, had a relatively low response rate. Open-ended (Spradley, 1979; Fontana and Frey, 1994; Kostera, 2007) and elite (Stephens 2007) interviews were conducted between 2005 and 2008 with the founder of the community and with 13 managers in companies in Denmark, Germany and Holland. We also conducted direct observation (Agar, 1980/1996) participating in a three-day TYPO3 conference in Germany in 2006, and gathered material from the Internet about the community (Westenholz *et.al.*, 2012). The analysis of this material was constructed around themes of sharing and not sharing knowledge in relation to the specific practice in which companies developed software for specific customers. The detail code sheet can be found in Westenholz (2012, p.103 and 110).

Higher levels of involvement

Level 4: Companies creating and leading their own open source software community

The level 4 companies create and manage OSS project to support their business. In recent years, the famous, although unsuccessful story, is Nokia with its Maemo community. The Maemo.org domain was originally registered in February 2005 by Nokia Corporation and has since then remained in its assets. Not coincidentally this launch occurred the same year that Nokia released its N770 device and its involvement with many upstream open source software projects. The first hackers' activities on the website are dated May 2005. Although the project websites state that the "Maemo community is a non-profit organization sponsored by Nokia, which is an active and equal member", it was clear that Nokia's position in it is far more privileged. From the project's beginning Nokia

had an overall control and ability to support only selected developers. Maemo users and developers – if they decided to collaborate – had to accept Nokia's dominance (Ciesielska, 2010; 2012; Ciesielska and Iskoujina, 2012).

In general, Nokia offered very few incentives for developers and it did not enable a payment facility for software downloads from Maemo.org by the end users. The biggest problem with Maemo.org applications was that very few of them were ever fully completed to be released for the mass market. As a result the website's community remained truly engaged – albeit at a hobbyist level, not a professional level. Maemo.org attracted "the cloud of not so serious developers" (Ciesielska, 2010) and remained mostly a support website for users of Nokia devices. From Nokia's point of view it was not a successful undertaking, as it has not managed to provide a set of high-quality applications for Nokia's mobile phones. Many Maemo.org contributors felt the same way. One of the Nokia engineers also commented that Nokia:

> *have not seen anything really useful come out of that exercise ever there are some tools that would have been developed anyway completely without having this sort of community there.*

Level 5: Companies participating in OSS projects led by the community

At the same time as launching Maemo, Nokia was collaborating on other projects led by the communities. One of them was GNOME – the free desktop project. The GNOME community is primarily focused on software development and attracts a wide group of contributors, including many bigger and smaller companies, like Red Hat, Google, IBM, Motorola, Oracle, Collabora, Igalia, SUSE, Code Think, Free Software Foundation, Mozilla Foundation, and many independent software developers. Although Nokia eventually became the sponsor of the GNOME Foundation, it

has never been able to gain control of the project and therefore had to participate, benefit and contribute as any other member (Ciesielska, 2010; 2012).

From 2005 Nokia was open about its interest in the GNOME code repository, although they had been working on GStreamer-related technologies much earlier than that. In this initial stage they were purely a 'community customer' with no contributions (Level 2 involvement). Soon they realized that this strategy had a serious flaw - it created a code fork, which prevented them from taking full advantage of the OSS development process. This problem was related to the fact that a complex system like software is very sensitive to changes and sub-optimizations. In other words, a small change may have tremendous consequences on how the software works (or stops working). Since summer 2005 Nokia tried to fix this by aligning its version of the code repository with the official GNOME one. It took them about two years to catch up (Ciesielska, 2010; 2012).

Soon after, Nokia became a cornerstone sponsor of the annual GNOME Users' And Developers' European Conference, where the firm regularly makes presentations. Nokia also hired OSS engineers and subcontractors chosen among the original GNOME/GTK+ developers, who for several years became involved in the Maemo project. The first task was to reintegrate Nokia's with GNOME's code. This was quite difficult and required not only technical knowledge, but also strategic decision making of what was appropriate for submission to the upstream. Later on, one of the Nokia OSS engineers declared that in order to simplify the technical process, Nokia tried to integrate with the upstream wherever possible:

> *So we do the design and [use] open source instead of sit down and write it ourselves. We send the patches to an open mailing list, get reviews and feedback […] and when it's done, it's merged up with upstream and then we can pull it back from there. So, in a way, rather than doing it within Nokia and pushing it out, it sort of comes back through open source, which is nice.*

At the same time the expectations and needs of the open source community were growing, beyond what Nokia was offering. The basic difference between Nokia's approach to GNOME and a truly OSS-originated company was the lack of any altruistic element. Despite engineers' declarations and good intentions, truly open collaboration was not exactly in line with the overall corporate logic. A lot of formal and informal internal rules restricted actual OSS contributions. The protection of a possible competitive advantage was the most significant factor that created problems. The competitive advantage rule stated that any piece of code that is not a simple fix or bug report, but presents a solution never implemented before, represents Nokia's competitive advantage and is an internal knowledge asset. This affected, in particular, GNOME Nokia contributors whose code was taking a long time reaching upstream, and some not at all. A Nokia employee commented on this:

> *The problem is that quite often the things we are asked to do are either patented or what they call 'a competitive advantage'. So then they don't really want that to be shared because it is an advantage.*

If Nokia open sourced anything, it was because it was better for Nokia's development process, not necessarily for the good of anyone else. Therefore over a couple of years Nokia managed to create a highly negative impression of a company which took more than it gave. At the same time Nokia kept parts of the Maemo operating system as a closed component[2] (Ciesielska, 2012).

The initial GNOME excitement turned into distrust. In the meantime, Nokia managed to lose its GNOME/GTK team – the majority of which resigned in 2008 due to job burnout. But the turning point in Nokia-GNOME relations was in June 2008, when they announced their acquisition of Trolltech, their Qt application development framework – a competitive solution to GNOME/GTK+ package. The next decision was to replace GNOME/GTK+ with Nokia-owned Qt. Most GNOME contributors couldn't hide their dissatisfaction with Nokia's decision and found it very insulting that

[2] Similarity to Level 3 of involvement – mixing open and proprietary solutions

Nokia advertised Qt jobs at their conference, showing no respect for how much the GNOME community had worked to make GTK+ a useful tool (Ciesielska, 2010).

Abandoning GNOME without any warning proved that Nokia had a lack of respect for OSS work and achievements and fuelled even more distrust towards the company. What Nokia did not realize for many years, was the importance of trust within open source software projects. As a result, Nokia started to be perceived as located at the opposite end of the scale to companies like Red Hat, which were considered truly open source, very much immersed in the open source logic. Nokia's agenda, on the other hand, was unclear and changeable, because it has never been able to transform its strategy from closed to open software development (Ciesielska and Iskoujina, 2012).

Level 6: Companies becoming members of open source software communities

This last level illustrates the highest degree of company involvement. This level is not well described in the literature. We use the case of TYPO3 to illustrate this level (Westenholz, 2012).

TYPO3 is a content management system created by a young Dane - Kasper Skaarhøj - in the late 1990's. During 1½ year he devoted most of his time developing the software, which he released in 2000 on a GPL license[3]. After it was released a network of very diverse actors established itself around the product, which grew to thousands over the years. Some are private users of TYPO3. Others are employed in companies and use the software to develop a homepage for their own company/organization. A third group are freelancers and web bureaus selling services connected to the implementation of TYPO3. The customers don't pay for the software, which is released within the community but they pay for the specific applications and services requested by them from the company. These groups are primarily users of TYPO3, but many also take part in its development by identifying errors/bugs and developing extensions, specifically directed at the customer's needs. The core development of TYPO3 is a fourth group comprising relatively few people. They also

[3] Why he did not establish a business is another story – not told here.

evaluate and incorporate the suggestions they deem useful into the official version of the program. Almost all of them are freelancers loosely affiliated with a web bureau. The official TYPO3 Association was created in 2004 as a non-profit organization, and was founded to provide funds for long-term development goals that would not be possible otherwise. Since it started, the association has attracted about 600 members, companies and freelancers alike. The funds are generated by membership fees and donations.

Companies play an important role within the TYPO3 community. Compared with non-commercial TYPO3 actors, the 'commercial' TYPO3 actors devote more man-hours on developing TYPO3: they interact more frequently with others about the software, they participate in social community events and to a higher degree, consider themselves to be part of the TYPO3 community on a local, national, and international level. They also feel more known in the community and often think that one should pay attention to the proper socialization into the community. The conclusion does not mean that TYPO3 can be understood as a community of commercial actors alone. The non-commercial contributors also take part in the network around TYPO3, but the commercial ones play a central role in this network.

The diversity of actors as well as a clear business model emerging around this community allowed for a hybridization of technological, economic and social logics. The following citation from a managing director and programmer can illustrate this:

> *We have done a lot of work for TYPO3. What we develop, we like to share with others...what we do, others enjoy and we don't think that we lose anything by publishing it. We still have the expertise and know-how when it comes to the new system... we earn our money by getting something that other people have developed. So in this way, we enter some kind of community where we draw on [the work of] a large number of other people,*

and then we give back to the community as best we can. In our case, it is not just financial, but more about investing our time and publishing some of what we have created.

Firms participate in the development of TYPO3 software in many different ways. Some firms, developing software for specific customers, release their new knowledge to the TYPO3 community. Some software developers, employed in a TYPO3 company, contribute to the development of the core TYPO3 software on a voluntary basis, together with volunteer programmers and programmers from other companies. Some of them also contribute outside of their usual work time. Several companies, working together to develop specific elements of the software, release knowledge directly to the community, and many firms, which primarily develop TYPO3 software internally, also share their code with the community (Westenholz, 2012).

Not all companies contribute back everything they have created. It happens in particular when their customers have paid for development of specific knowledge and do not want it to be public or when knowledge is so customer-made that it is of no interest to the community. However this type of non-sharing behavior is accepted in the community. But sometimes the companies get into dilemmas between the different logics. It may happen if they have spent a lot of time in developing knowledge and they want to increase their profit by non-sharing behavior. On the other hand they know that technical advantages are achieved if the knowledge is shared and they may also get sanctioned by the community if they are defined as free-riders.

Discussion and Conclusions

As showed in the example cases, there are (at least!) six levels of business involvement in the open source field. Now, we will discuss those empirically observed levels in the context of the existing

literature and conclude with key dilemmas commercial firms may face if they decide to adopt the open source logic.

The lowest degree of contact is where companies imitate/translate ideas from open source (duplicate incentives, knowledge-sharing within the firm, user-involvement), but are not directly involved with the communities. Level 1 companies simply imitate the open source way of developing software but maintain it within a clearly controlled environment. In the literature this new way of boosting creativity in business is covered under the term 'open innovation'. However, in this case the whole process of innovation is not truly open, but only those elements that are beneficial for the company, and the exchange is often one way (the 'outside-in' part of open innovation), for example using existing or potential customers to suggest improvements. On the contrary, inside-out open innovation requires organizations to allow ideas to go outside to the market, but these cases are rarer, less research reports them and they seem to be less understood by business (Chesbrough, 2012).

The next step sees companies becoming 'community customers' as they use open source software and sometimes support the community financially. Level 2 companies acknowledge the value created in the open source area, but benefit from it directly, integrating or simply using OSS in their businesses[4]. 'Community customers' to a large extent are one-side beneficiaries, but some may donate or support OSS projects in a minor way.

Real involvement in the development of open source software happens in the third step, where companies sell their combination of proprietary software and open source software. Firms may deliberately try to marry multiple software modules with clear distinctions as to which are open and which should be keep proprietary. There are two potential ways of mixing software development

[4] The examples of commonly used Open Source software includes: Linux Operating System, FireFox Browser, Mozilla Browser, Apache Web server, Perl programming language, OpenOffice Suite, and MySQL Database.

solutions. Companies can either be based on open software, but allow closed-source extensions or despite a proprietary core, allow for open source extensions (Casadesus-Masanell and Llanes, 2011)[5].

In the fourth step, companies release codes and try to build an open source software community around the project. In most cases the company has a clear lead and/or control over the project, which contributions to accept or reject and which developers to support. The OSS project serves as a supplementary or sole R&D project with clear links to the company's primary business (Ciesielska, 2010).

The fifth step is characterized by firms which participate in open source software developments lead by an independent community. In this case the firm acts as co-developer and potentially financial supporter, but has relatively little say in terms of project management.

The last step, which illustrates the highest degree of involvement, is characterized by firms becoming members of open source software communities: firm employees write code, support the community and participate in its management. This level is not well researched in the literature.

The three lower levels of involvement are fairly well described by the literature, especially that which focuses on strategy, innovation, R&D. In all cases the primary companies' logics reside in the economic spectrum, with elements of technology-related structures. Levels 1 and 2 do not pose much difficulty for commercial organizations. Both of those levels are embedded in business activities, with a clear agenda of one-sided benefits from the open source software or ideas behind it. Level 1 firms may capture innovation spill-outs from other technology areas. Their main dilemma is the choice between absolute autonomy at the expense of potential innovation or relying on external contributions to expand their creative potential. Level 2 firms benefit simply by using

[5] The examples of the former, 'Open Core' solutions are SugarCRM, Zimbra, JasperSoft, Mac OS X, while MSFT Net, Mathematica, Stata, Facebook fall in the latter 'Open Edge' category.

open source software instead of purchasing a proprietary solution. The choice between waiting for others to innovate ”for free” versus buying innovation seems to be the key dilemma here. None of the choices are obviously better at providing more advanced solutions. Although the open source solutions are perceived as cheaper, the initial cost savings may be spent later on maintenance and other adaptations. Level 3 firms can employ either purely commercial or technological logics. In both cases they remain consistent in their talks and actions. Commercially-focused ones see a business opportunity in mixing open and closed source software, while technology-driven ones would aim at allowing open source software to be combined or compatible with the proprietary components.

Insert Table 2 about here

Firms with higher levels of involvement (4 to 6 on our scale) find themselves in much more complicated situations. This primarily comes from the fact that at those stages firms face hard to commensurate logics of action and various stakeholder pressures. Level 4 and 5 firms are guided primarily by economic and technological logics, but within social structures created by open source software communities. As a result the commercial actors need to develop and keep successful trust relationship with voluntary and business partners in order to secure valued and continuous contributions. While companies leading open source projects (level 4) are the primary beneficiary, companies participating in open source projects lead by an open source community are the secondary ones (level 5), with the community interest taking the first place. In both cases the main dilemma for the commercial actors is how to balance autonomy and the value generated by collaboration. Also, the movement from Level 4 to Level 5 means shifting the main beneficiary from the company to the community. Level 5 firms are in a difficult position where they have little control over the open source software communities’ directions, while on Level 6 there is a high risk of providing competitors with free knowledge. As a result businesses becoming involved in open

source at Level 5 and 6 become vulnerable, unless they adjust their strategies and reformulate value propositions to accommodate the fact that the work they do in the open is easily available to anyone. Failing to do so result in luck of trust and legitimacy, and may lead, as in the case of Nokia, to project closure (Ciesielska, 2010; 2012).

Alternatively, if companies succeed in building trust and legitimacy they become involved in fast-paced innovative processes, which they would not have been able to create on their own. Furthermore software engineers actually consider their work to be fun (Hunter *et al.*, 2010) and the involvement in open source may become of personal importance and a career progression opportunity. This will create a strong pro-open source coalition inside the organization. This strong cultural basis combined with a balance between exploration and exploitation of knowledge are considered to be potential success factors (Hemetsberger and Reinhardt, 2006). TYPO3 is an excellent example of how combining commercial and community interest lead to advanced technological innovation and sustainable business (Westenholz, 2012).

Theoretically our study contributes to the growing literature within institutional organizational theory focusing on different organizational responses to pluralistic, widespread institutional field logics (Boxenbaum 2006; Feldman 2003; Oliver, 1991; Reay and Hinings, 2009; Tilcsik, 2010; Westenholz, 2012). As mentioned in the introduction software development used to be developed within either an organizational field of companies applying an institutional logic of 'copyright', or an organizational field of open source communities applying an institutional logic of 'copyleft'. Since the middle of the 1990s firms have begun to get involved in open source software development and our contribution is to show how software companies have responded to this development in six different ways. We also show that they are facing different types of dilemmas according to their responses.

Finally, we would like to mention some ideas for future research. In the paper we identify 6 graduations of involvement. However the number of cases, and their timescale, limit the generalizability of our analysis. A more systematic taxonomy may show new global trends in commercial open source software solutions. One may also develop a new taxonomy by combining the work by Grand *et al*. (2004) focusing on four levels of companies' resource allocation and our work focusing on six degrees of companies' involvement with open source software communities. Further empirical studies may reveal even more graduations or combinations of graduations. Obtaining empirical examples of successes and failures at each of the levels so a more extensive comparison could be made would be worthwhile. Finally it would be interesting to study the dynamics between the graduations to understand how companies move from one level to another. The exact patterns are yet to be uncovered and properly theorized. Empirical evidence is not coherent in this matter and does not support a single movement throughout the scale. Some organizations are operating at different levels, using differentiated strategies (Ciesielska, 2010; 2012). Other companies may regress in their open strategies due to difficulty in managing the open-closed dilemma (Ciesielska, 2010, Ciesielska and Iskoujina, 2012). Yet other businesses are more static, like the ones created within an OSS paradigm offering business support and bespoke solutions based on open source software (Westenholz, 2012; Ciesielska and Petersen, 2013).

Literature

Bonaccorsi, A., Giannangeli, S. and Rossi, C. (2006), "Hybrid business models in the open source software industry." *Management Science*, Vol. 52, No. 7, pp. 1085–1098.

Boxenbaum, E. (2006), "Lost in translation? The making of Danish diversity management." *American Behavioral Scientist*, Vol. 49, No. 7, pp. 939-948.

Brodkin, J. (2007), "Pinpointing open source start-ups", *Network World*, Vol. 24, No. 33, pp. 16-44.

Casadesus-Masanell, R. and Llanes, G. (2011), "Mixed source", *Management Science*, Vol. 57, No. 7, July 2011, pp. 1212-1230.

Cassell, M. (2008), "Why governments innovate: adoption and implementation of open source software by four european cities", *International Public Management Journal*, Vol. 11, No. 2, pp. 193-213.

Capra, E. et al (2009), "A survey on firms' participation in open source community projects", in Boldyreff, C. *et. al., Open Source Ecosystems: Diverse Communities Interacting*, Proceedings from 5th IFIP 2.13 International Conference on Open Source Systems, OSS, Skövde, Sweden, pp. 225-236.

Chesbrough, H. (2003), *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press, Cambridge, MA.

Chesbrough, H. (2006), *Open Business Models*. Harvard Business School Press, Cambridge, MA.

Chesbrough, H. (2011), *Open Services Innovation: Rethinking Your Business to Grow and Compete in a New Era.* Jossey Bass, New York, NY.

Chesbrough, H. (2012), "Open innovation", *Research Technology Management*, Vol. 55, No. 4, pp. 20-27.

Ciesielska, M. (2010), *Hybrid organizations. A case of the open source-business setting*, Copenhagen Business School Press, Frederiksberg.

Ciesielska, M. (2012), "Nokia's involvement in two types of open source software communities", in Westenholz, A. (Ed.), *The Janus Face of Commercial Software Communities – An investigation into institutional (non)work by interacting institutional actors*, Copenhagen Business School Press, Frederiksberg, pp. 123-143.

Ciesielska, M. and Iskoujina, Z. (2012), "Trust as a success factor in open innovation. The case of Nokia and Gnome", in Jemielniak, D. and Marks, A. (Eds) *Managing Dynamic Technology-Oriented Business: High-Tech Organizations and Workplaces*. Information Science Reference, pp. 11-29.

Ciesielska, M. and Petersen, G. (2013), "Boundary object as a trust buffer. The study of an open source code repository." *Tamara Journal for Critical Organization Inquiry;* Vol. 11, No. 3; pp. 5-14.

Dittrich, K. (2007), "Nokia's strategic change by means of alliance networks. A case of adopting the open innovation paradigm?" in K. Sangeetha and Sivarajadhanavel, P. (Eds), *Nokia Inc. A global mobile leader.* Icfai's Professional Reference Book Series, Icfai University Press, Chennai, India.

Feldman, M.S. (2003), "A performative perspective on stability and change in organizational routines." *Industrial and Corporate Change,* Vol. 12, No. 4, pp. 727-752.

Feller, J. and B. Fitzgerald. (2002), *Understanding Open Source Software Development.* Addison-Wesley, London-Stockholm.

Fitzgerald, B. (2008), "The Transformation of Open Source Software. " *MIS Quarterly*, Vol. 30, No. 3, pp. 587-598.

Fitzgerald, B. and Kenny, T. (2004), "Developing an information systems infrastructure with open source software", *IEEE Software*, Vol. 21, No. 1, pp.50-55.

Fontana, A. and Frey, J.H. (1994), Interviewing: The art of science. In: Denzin, B.K. and Lincoln, Y. S. (Eds.) *Collecting and Interpreting Qualitative Materials*, Sage, London, pp. 47-78.

Fosfuri, A., Giarratana, M. S., and Luzzi, A. (2008), "The penguin has entered the building: The commercialization of open source software products", *Organization science*, Vol. 19, No. 2, pp. 292-305.

Galli, P. (2005), "Tool lowers migration costs", *Eweek*, Vol. 22, No. 17, pp. 23-25.

Gehring, R.A. (2006), The institutionalization of open source." *Poiesis Prax*, Vol. 4, pp. 54–73.

Ghosh, R.A. (Ed.) (2006), *Study on the economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU. Final report.* Available at: http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf (accessed 11 October 2008).

Gillies, V. and Edwards, R. (2005), "Secondary analysis in exploring family and social change: Addressing the issue of context", Forum: Qualitative Social Research, Vol. 6, Issue 1, Art. 44, Available at: http://www.qualitative-research.net/index.php/fqs/article/view/500/1077 (accessed 17 November 2013)

Grand, S. *et al.* (2004), "Resource allocation beyond firm boundaries: A Multi-Level Model for Open Source Innovation", *Long Range Planning*, Vol. 37, pp. 591-610.

Hamm, S. (2007), "Small player scores big open-source win", *Business Week Online*, p. 11.

Hammersley, M. (2010), “Can We Re-Use Qualitative Data Via Secondary Analysis? Notes on Some Terminological and Substantive Issues”, *Sociological Research Online*, Vol. 15, Issue 1, No. 5, Available at: http://www.socresonline.org.uk/15/1/5.html (accessed 16 November 2013).

Hemetsberger, A. and Reinhardt, C. (2006), “Learning and knowledge-building in open-source communities. A social-experiential approach”, *Management Learning*, Vol. 37, No. 2, pp. 187–214.

Hoppenbrouwers, J. (2007), “Community customers”, in: St Amant, K. and B. Still, *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*, Hershey: Information Science Reference, pp. 510-521.

Hunter, C., Jemielniak, D. and Postuła, A. (2010), ”Temporal and spatial shifts within playful work”, *Journal of Organizational Change Management*, Vol. 23, No. 1, pp. 87-102.

Idelchik, M, and Kogan, S. (2012), “GE's open collaboration model”, *Research Technology Management*, Vol. 55, No. 4, pp. 28-31.

Karjalainen, M. (2010), “Large-scale migration to an open source office suite: An innovation adoption study in Finland.’ *Electronic dissertation. Acta Electronica Universitatis Tamperensis 997,* available at: http://acta.uta.fi/pdf/978-951-44-8216-8.pdf (accessed 10 October 2012).

Kostera, M. (2007), *Organizational Ethnography: Methods and Inspirations*. Utbildningshuset Studentlitteratur.

Kozinetz, R. V. (1997), “I want to believe: ‘A netnography of the X-Files’ subculture of consumption.” *Advances in Consumer Research*, Vol.24, pp. 470-475.

Kozinets, R. V. (1998), “On netnography. Initial reflections on consumer investigations of cyberculture”, in Alba, J. and Hutchinson, W. (Eds.), *Advances in Consumer Research*,

Vol. 25, Association for Consumer Research, Provo, UT, pp. 366-371.

Kozinets, R. V. (2002), "The field behind the screen: Using netnography for marketing research in online communities", *Journal of Marketing Research*, Vol. XXXIX, pp.61-72.

Krishnamurty, S. (2005), "An analysis of open source business models" in: Feller, J. *et al. Perspectives on Free and Open Source Software,* The MIT Press, Cambridge, Mass.

Langdon, C.S. and Hars, A. (2007), "Open source software – Business models and customer involvement economics". in: St. Amant, K. and B. Still (Eds.) *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*, Hershey: Information Science Reference, pp. 522-531.

Langer, R. and S.C. Beckman. (2005), Sensitive research topics: netnography revisted. *Qualitative Market Research: An International Journal*, Vol. 8, No. 2, pp. 189-203.

Lerner, J. and Tirole, J. (2002), "Some simple economics of open source", *The Journal of Industrial Economics*, Vol. 50, No. 2, pp. 197-234.

Marsden, P.V. and Wright, J.D. (2010), *Handbook of Survey Research.* Second Edition. Bingley, UK: Emerald Group Publishing.

Moore, N. (2007), "(Re)using qualitative data", *Sociological Research Online*, Vol. 12, No. 3. Available at: http://www.socresonline.org.uk/12/3/1.html (accessed 17 November 2013).

Moran, G. (2012), "How I saved $8,000", *Entrepreneur*, Vol. 40, No. 4, p. 68.

Oliver, C. (1991), "Strategic responses to institutional processes." *Academy of Management Review*, 16, pp. 145-79.

Reay, T. and Hinings, C. R. (2009), "Managing the rivalry of competing institutional logics." *Organization Studies*, Vol. 30, No. 06, pp. 629-652.

Scott, W.R. (1994), "Conceptualizing organizational fields: linking organizations and societal systems", in: Derlien, H-U *et. al.*, *Systemrationalitat und Partialinteressen*, Nomos Verlagsgesellschaft, Baden-Baden.

Sheehan, K. (2001), "E-mail survey response rates: A review." *Journal of Computer-Mediated Communication,* Vol. 6, No. 2, p. 58.

Stafford, J. (Ed) (2004), *Migration in-depth, part 1: Why Health First is dumping MS Office*; Available at: http://searchenterpriselinux.techtarget.com/news/964811/Migration-in-depth-part-1-Why-Health-First-is-dumping-MS-Office (accessed 20 November 2012).

Stephens, N. (2007), "Collecting data from elites and ultra-elites: telephone and face-to-face interviews with macroeconomists." *Qualitative Research*, Vol. 7, No. 2, pp. 203-216.

Spradley, J. P. (1979), *The Ethnographic Interview.* Holt, Rinehart and Winston, New York.

Tilcsik, A. (2010), "From ritual to reality: demography, ideology, and decoupling in a post-communist government agency." *Academy of Management Journal*, Vol. 53, No. 6, pp. 1474-1498.

Ulhøi, J. P. (2004), "Open source development: a hybrid in innovation and management theory." *Management Decision*; Vol. 42, No. 9, pp. 1095-1114.

von Hippel, E. (1976), "The dominant role of users in the scientific instrument innovation process." *Research Policy,* Vol. 5, No. 3, pp. 212–239.

von Hippel, E. (2005), *Democratizing Innovation.* MIT Press, Cambridge, MA.

von Hippel, E, Ogawa, S. and de Jong, J.P.J. (2011), "The age of the consumer-innovator." *MIT Sloan Management Review,* Vol. 53, No. 1, pp. 27–35.

Watson, R.T., Boudreau, M-C., York, P.T., Greiner, M.E., and Wynn, D. Jr. (2008), "The Business of Open Source – Tracking the changing competitive conditions of the software industry". *Communications of the ACM*, April, Vol. 51, No. 4, pp. 41-46.

Weber, S. (2004), *The Success of Open source*, Harvard University Press, Cambridge Mass.

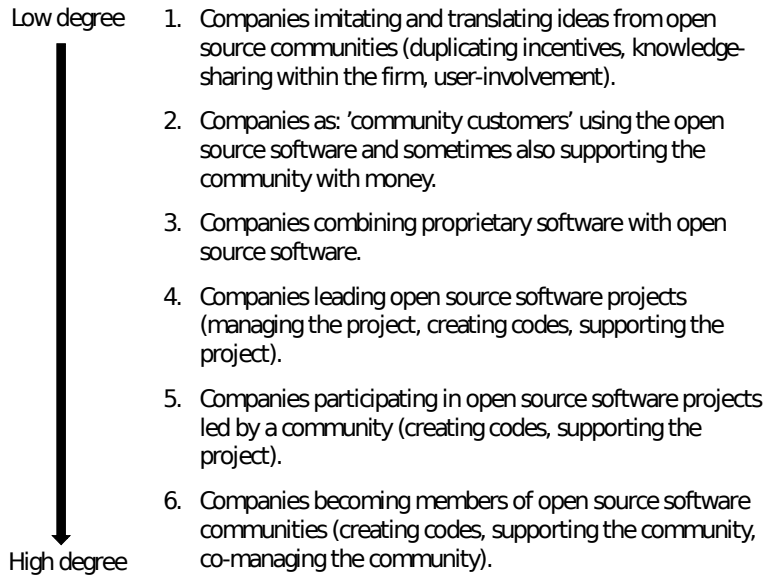
Westenholz, A. (Ed.) (2012), *The Janus Face of Commercial Software Communities – An investigation into institutional (non)work by interacting institutional actors*, Copenhagen Business School Press, Frederiksberg.

Westenholz, A *et. al.* (2012), "Methods", in Westenholz, A. (Ed.), *The Janus Face of Commercial Software Communities – An investigation into institutional (non)work by interacting institutional actors*, Copenhagen Business School Press, Frederiksberg, pp. 177-183.

Winston, A. (2011), "GE's Eco-innovation platform." HBR Blog Network October 26, 2011, Available at: http://blogs.hbr.org/winston/2011/10/ges-eco-innovation-platform.html (accessed 27 November 2012).

Figure 1: Firms' involvement in Open Source Software Communities

(Westenholz, 2012, p.28)



.

Table 1: Examples of Open Office Deployment

(Based on data from http://wiki.openoffice.org/wiki/Major_OpenOffice.org_Deployments as of October 30 2012)

Type Area	Governments, local authorities and agencies	Education institutions, from schools to Universities	Private sector
Africa	• South Africa • Morocco	• Namibia (300 schools) • South Africa (2205 machines deployed by Novell)	• N/A
Asia	• Japan • Malaysia • Vietnam • India • South Korea • Singapore • Macau • Pakistan	• Philippines (Tayabas Western Academy, Asia School of arts and Sciences and University of the Philippines) • India (Parshvanath College of Engineering, Thane)	• Japan (Sumitomo Electric Industries, Sumitomo Electric Industries Ltd., K.K. Ashisuto) • Thailand (Bangkok Airways, EGAT) • India (GB Engineering, LIC, ELCOT, ICICI Bank of India) • Singapore (Resolvo Systems Pte Ltd)
Europe	• Italy • Denmark • Latvia • Hungary • Belgium • UK • Germany • Poland	• Denmark (some schools, Copenhagen Institute of Technology, University of Southern Denmark) • UK (some schools) • France (some schools) • Poland (Polish Academy of Science-partly) • Germany (a school, University of Cologne) • Macedonia (to supply all	• Romania (ASIGEST, MOBEXPERT Group, PROFI Rom Food, Total Cont-Sibiu, Zendo Computer) • UK (Ford and Warren Solicitors, Future Publishing, Travel Republic) • France (Groupe Laurent, Peugeot Citroën) • Turkey (Kervan Gida) • Germany (LVM Insurance)

	• Netherlands • France • Austria • Macedonia • Finland • Romania • Spain • Norway • Portugal	schools) • Turkey (all primary schools) • Croatia (school teachers' training in OO)	
North America	• Some USA states • Canada	• USA (several schools and colleges)	• USA (Blackcountry.com, JH Larson Electrical Company, Ernie Ball Guitars, Health First Inc, Life Brokerage Equity Group, Miller Industries, Inc. NAFECO, Novell, Everex, Hustler Turf Equipment, Petrolink International Houston)
Oceania	• Australia	• Australia (Schools around New South Wales, University of Melbourne)	• Australia (De Bortoli Wines) • New Zealand (Egressive Ltd, MIP Holdings)
South America	• Argentina • Brazil • Paraguay	• Chile (high schools) • Brazil (University of Santa Cruz do Sul, Catholic University of Minas Gerais State)	• Argentina (Fen Hotels) • Paraguay (Fen Hotels) • Brazil (Casas Bahia)

Table 2: Six levels of firms' involvement in open source software communities

	Level 1: **Firm imitating or translating ideas from open source community**	**Level 2:** **Firm in the role of open source community 'customers'**	**Level 3:** **Firm combining proprietary and open source software**	**Level 4:** **Firm leading open source project**	**Level 5:** **Firm participating in open source project led by an open source community**	**Level 6:** **Firm becoming member of open source community**
Primary firm logic	Economic	Economic	Economic and technological	Primarily economic and technological. Secondary social	Primarily technological and economic. Secondary social	Technological, economic and social
Firms' type of relationship with open source communities	Borrowing ideas from open source development	Becoming a user of open source software	Developing a fit between proprietary and open source software	Financial and fully hosting an open source project	Co-developer in the project and financial supporting the community	Co-leader, co-developer, and financial supporting the community
Key actors	Company	Company	Company	Company as leader and open source community members as followers	Open source community as leader and company as follower	Voluntary developers and companies as members within the community
Firms' regulative license	Copyright	Copyright	Copyright and copyleft	Copyright and copyleft	Copyright and copyleft	Copyleft

Who is benefitting from the open source software development	The company is not benefitting from the open source software, but may capture innovation spill outs from other technology areas.	The company is benefitting by using open source software internally in the company	The company is enriching its own proprietary software by adding open source software	The company is the primary beneficiary. The followers in the open source community are secondary beneficiaries	The open source community is the primary beneficiary. The company as a follower is secondary beneficiary	Companies, customers and voluntary developers are all primary beneficiaries
Firm dilemma	Autonomy at the expense of potential innovation	Waiting for others to innovate versus buying innovation	Waiting for others to innovate versus internal innovation	Autonomy versus cooperation	Cooperation versus autonomy	Cooperation versus free-riding