

Innovating with Open-Sourcing: Governance Concerns for Managers

Maha Shaikh

London School of Economics and Political Science
m.i.shaikh@lse.ac.uk

Tony Cornford

London School of Economics and Political Science
t.cornford@lse.ac.uk

ABSTRACT

Among various forms of innovation in industry structures and business models an increasing number of companies have shown interest in aligning themselves to an open source software model as a means to capture intellectual energy, productive software processes and relevant technical skills. This is evident both within small and niche businesses, but also within the largest companies – a phenomenon known as open-sourcing. This paper presents findings from a field study of open-sourcing of software development within two large, global technology companies. It reports on the ways in which open-sourcing is accommodated within the corporate context, and assesses the innovative strategies managers use as they engage with this phenomenon and seek to work co-operatively with open source communities. The analysis focuses on three primary areas that emerge from the data and which are seen to require particular attention in such organizations; license and IPR regime; community approach; and a modified development process.

Keywords

Open source, open-sourcing, company adoption of open source, governance, control, and managerial coping strategies.

INTRODUCTION

The open source software model can serve as a means for firms to capture intellectual energy, learn productive software processes and access relevant technical skills. This is evident in both small and niche businesses, and within the largest corporations. This movement has been described as open-sourcing (Agerfalk and Fitzgerald, 2008; Shaikh and Cornford, 2008), but also under other conceptualizations such as corporate source (Dinkelacker *et al.*, 2002), insourcing (Hirschheim and Lacity, 2000), cosourcing (Kaiser and Hawk, 2004), and netsourcing (Kern *et al.*, 2002).

This area of research is relevant because what began as a small movement has since evolved, and permeated many walks of life, especially business. It is this latter area that we focus on in this paper because it has not seen too many in-depth studies and we have yet to understand the phenomenon of open-sourcing in any detail. Our paper is part of a larger study of this phenomenon. We present initial findings from a field study of the open-sourcing of software development within two large, global technology companies. These companies were chosen specifically because their experience with open source (more than ten years each) could provide us with a more nuanced understanding of what a move from proprietary to open source entails.

This work builds on Agerfalk and Fitzgerald (2008) and provides insight to the strategies that managers adopt to harness and control open source activity, and accommodate open-sourcing within the corporate context. This can be challenging for both parties – the corporation and the open source community - as one manager told us, from the perspective of the company, ‘it is about letting go of control’ in order to keep some control. The analysis focuses on three primary areas that emerge from the data and which are seen to require particular attention in such organizations; license and intellectual property rights (IPR) regime; community approach; and a modified development process.

LITERATURE REVIEW

There is a small, but growing literature on open source adoption by commercial enterprises (large, medium and small) (Butler *et al.*, 2008; Capra and Wasserman, 2008; Pulkkinen *et al.*, 2007; Martin and Hoffman, 2007; Holck *et al.*, 2005; Lindman *et al.*, 2008; Melian and Mahring, 2008; Ven and Mannaert, 2008). A number of issues are tackled in these studies such as governance, license issues, business innovation, and strategies of success identified by managers. The work reported here, which is ongoing, aims to create a more comprehensive framework within which to understand the various forms of open-sourcing, the nature of the decisions made by companies when they adopt open sourcing, and the various coping strategies that company managers use as they work with open source collectives.

Systems development and management, and software processes in particular, are fundamentally knowledge based activities. One key to success is sourcing talent and gaining access to appropriate knowledge communities. Doing this well will almost certainly require innovation and challenge traditional means (Chesbrough, 2006; Chesbrough *et al.*, 2007; Wood and Guliani, 2005). In other areas of business open innovation models, often linked to the internet, have attracted increasing attention (von Hippel, 2001; von Hippel, 2005; von Hippel and Krogh, 2003). For example, Proctor & Gamble (P&G) in part base their R&D strategy on an open model, named as the “Connect and Develop” innovation model. Huston and Sakkab (2006) describe it as a process to “leverage external assets and capabilities.... [in a] relationship of co-invention-based interaction with outside resources”. This model with its connections focus, seeks to tap into multiple knowledgeable communities across the globe (Huston and Sakkab, 2006; Huston and Sakkab, 2007; Sakkab, 2002). These authors are clear that this is not a conventional method of outsourcing R&D, but rather “in-sourcing creativity”, and as in OS processes, aims to tap into a large pool of people, ideas, developers and testers who can offer the vital diversity needed by a global company such as P&G.

Drawing from another strand in the OS literature we might understand open-sourcing as a new form of organizing, or the building of a novel type of virtual organization to serve new knowledge needs. Metiu and Kogut (2001) studied a number of software companies in four different countries and identified two distinct forms of organizing for innovation and creativity in globally distributed work. The established model they term the ‘global project model’, based on conventional ideas of specification and control. They also identify a new model emerging – termed the ‘open development model’. The ‘global project model’ at the most basic level implies that companies are able to take advantage of lower cost of labour by passing some work (routine tasks) to offshore low cost sites. This demands requirements specifications up front, and high degrees of control. In their analysis this model begins to transform over time into the ‘open development model’ as offshore firms and developers begin to not only follow requirements but also build their own skills, innovate and create their own requirements for client companies. This is similarly identified by Carmel and Agarwal (2002) in Stage 4 of their SITO model of offshore outsourcing.

Kogut and Metiu (2001) question how often offshore developers are able to move very far away from simple specification following. The ‘open development model’ may come to push at the boundaries of the ‘global project model’, but only when or if the motivation of contributors change, actively seeking new experience, knowledge and skills. This is an analysis that echoes strongly the general understanding of motivation in OS communities as building human capital and participation driven by the explicit purpose to learn and enhance skills (Lakhani and Wolf, 2005).

THEORETICAL INCLINATION

Borrowing concepts from von Hippel’s (2005) user-driven innovation we analyze our data through notions of principal agent theory (Alchian and Demsetz, 1972; Jensen and Meckling, 1976). Von Hippel’s main thesis of democratizing innovation is premised on the idea that collective action driven by users will lead to greater innovation and more successful adoption of it (von Hippel, 2005; Franke and von Hippel, 2003a; Franke and von Hippel, 2003b). Innovators include both individuals and firms, where lead users of innovations (or innovators) stand to profit the most. A lead user creates something that is useful to herself and thus perhaps to others too. If the latter is proved true then the commercialization of the idea can lead to economic gain.

In order to commercialize an idea an individual innovator needs the support of others or a firm. The basic ideas of principal agent theory are adapted by von Hippel (2005) to make sense of how, and why, both individuals and firms manage opportunistic behaviour in their relationship. In open-sourcing relations become a little more complicated as often there are no contracts signed between the firm and community members. A number of reasons, for example control over one’s work, need or sense of urgency to solve a problem, ability to customize a product, and experience and opportunity to learn, inspire innovators to persevere in spite of potential costs. So in effect it is not just the product and what you can do with it (can you share it, distribute it, etc) but the process of innovation (von Hippel, 2005) that leads to learning and knowledge creation.

From the other perspective, firms benefit from a larger pool of ideas and differing innovations that they can use to earn a profit. They need domain experts to innovate because knowledge is ‘sticky’ (von Hippel, 1994; 1998) and thus difficult to reproduce outside the context in which it was created. The idea of sticky knowledge can be linked to how companies that are actively encouraging and participating in open-sourcing consider resource capture (Shaikh and Cornford, 2008) to be one of the key reasons for turning to an open innovation process.

METHODOLOGY

Two large global technology companies were chosen as case studies. The larger study also encompasses data collection in small and medium enterprises but the focus in this paper is on managers in large companies and how they cope with the challenges open-sourcing brings. Our access to both companies was agreed upon on the condition of anonymity. Both

companies have moved towards greater use of open source software, ideas and development methods over the last ten years or more. These companies have a different focus on their use and adoption of open source and are thus at differing levels of adoption.

Data Collection

Our research method included semi-structured interviews carried out in person, but mostly via telephone. Each interview lasted an hour or more. The interviewees belonged to top and middle management, and also included software developers. In Company A access was negotiated in a traditional manner of approaching a contact and then asking the contact to suggest key personnel that could prove fruitful interviewees, keeping in mind the focus of our study. However, for Company B we did not have a similar contact, and were thus forced to search the Internet for names of personnel on open source related positions in large technology companies. We used project websites that we knew the large company was linked too and searched mailing lists for possible interviewee names. We found our first interviewees in this manner. Subsequently we successfully adopted the snowballing method whereby each interviewee was asked to offer a few more potential interviewees. Most unhesitatingly offered two or more names, people in their own company or colleagues in similar large technology based firms. This provided more than thirty interviews in less than two months of data collection.

Data Analysis

Interviews were transcribed and coded using Atlas.ti content analysis software. Using the tools of Grounded Theory (Glaser and Strauss, 1967; Strauss and Corbin, 1999), though not the full ontology, yielded a code book of forty-nine initial codes (reference withheld for anonymous reviewing purposes). Along with codes we added memos in the form of notes, concepts and broader emerging themes. Our theoretical inclination provided us with a lens through which to understand our data and offered the first 11 codes in the open coding step. Our theoretical ideas need to be understood more as a meta guidance to our analysis rather than offering concepts for micro analysis. The three main themes that we understood to be of relevance, and which our coding and memos gave rise too; include license and IPR regime; community approach; and a modified development process. The aim is in our larger study of open-sourcing to theorize this phenomenon using Grounded Theory tools and in this paper we begin this attempt by focusing on three themes.

FINDINGS AND DISCUSSION: BALANCE OF CONTROL AND INNOVATION

Of the various themes arising from an analysis of our data we chose to focus on only the three mentioned above because they related to managerial practices of coping, and strategizing innovation within a company. We have shown some of the data in Table 1 and indicated the code/memo/theme that each gave rise too.

Our analysis indicates a strong theme within companies to create an atmosphere that allows innovation to thrive but this is juxtaposed with a need to supervise through different control mechanisms imposed upon both the community and internally in the company. There is thus a dialectical need for control and innovation which needs to be balanced. From a company (principal) perspective we understand them as the controllers, and the community (agents) as the innovators, though the balance of power shifts back and forth. The premise for company engagement with open source communities is to capture resources and their innovations. Innovations are important but retaining the innovators is crucial for longevity. Focusing on the risk aspect of principal agent theory (Eisenhardt, 1989) we can understand how managers in both companies make substantial efforts to mitigate risk through control of the product, community, and process of development.

Product: License and IPR Regime

To work with open source is to accept the fundamental implications of an open license. But such licenses are a serious concern to commercial corporations whose natural instinct is to retain knowledge resources, or at least to retain the ability to mix code, proprietary and open, as needed. Not all open source licenses are as reciprocal as the GPL, and this allows companies to at times choose an appropriate one when initiating their own open source project. However, there is little or no choice for a company when they decide, for strategic reasons, to nurture or back an open source product which already has a community working on development and where the license has already been chosen. In practice being prepared to work with strong licenses is necessary.

A consequence of this is that, if companies want to seriously work with open source software, especially its distribution, there is a need for them to build in-house expertise in the various open source licenses and how they can be used. Interviewees from both companies indicated their legal department had become more skilled in open source licenses, and established some form of open source review board. The role of the latter group being to scrutinize code that is created for clients or is

distributed in any way to check for various license implications. This move has been accelerated by the increasing demand for open source products and software by customers encouraged by the prospect of reduced vendor lock-in.

Dual Licensing

There are possibilities of dual licensing (Valimaki, 2003; Olson, 2005; Comino and Manenti, 2008) and other complex 'work arounds'. Dual licensing made to release software under two different licenses, one which is usually open source and the other proprietary (Valimaki, 2003). However, according to our respondents, dual licensing and other work-arounds should usually be avoided because they cause complexity and critically, uncertainty as to the validity of potential IPR claims. If dual licenses are not the 'magic fix' envisaged then managers need to consider other issues when using, adopting or mixing company code with open source software.

Most Oft Adopted Licenses

Within the broad open source movement there are a large variety of different licenses in use ranging from the weak to the strong. Unsurprisingly, large companies appreciate and (when able to make a choice) implement weaker licenses such as Apache or BSD, rather than the viral/reciprocal GPL. Weaker licenses are a method to control the risks involved with what happens to 'their' product. However, often there is no choice and the license that will be used is dictated to by the existing licensing regime of the product and community that they choose to work with. More significantly, companies will try to affiliate with a product that is successful in terms of sustaining a strong community of developers around it, what ever the license they use. Thus IBM, SUN and many other large companies contribute to and participate in LINUX development. The technical credibility and developer base is ultimately the more relevant issue since resource capture – of experience and knowledge held by the community – is often the primary goal.

Community: Community Approach

All our interviewees, when asked why they feel that companies like theirs need to liaise with open source communities, cited their main reasons to be as access to a support network for expertise, testing, and ideas. These reasons form a key driver for pursuing an open source strategy. It is the community, perhaps more than the code that it provides, that interests large companies. The continued support, help, and importantly ideas (collective innovation) that attracts companies to open source collectives. Companies also understand that they need the goodwill of open source developers and must be seen to contribute back to the community.

But like all other decisions concerning open source, this brings with it a number of concerns for managers. Before aligning itself with any community a company must take into consideration how well the community matches with the company needs, governance structure, and the complementarity of goals.

Match between Company and Community

While conducting our interviews the respondents identified a number of aspects that they consider when making such decisions. They include (in order of relevance, again indicated by the interviewees); a healthy community; viable and needed technology; compatible license (if possible); and whether a community has members that are employees of other companies.

The most important factor suggested by all the respondents to the question of how they choose to work with a community was whether the community is healthy or not. If the community isn't healthy then it is unlikely that a company will want to get involved. The idea of a healthy community is interesting because, when probed, the interviewees explained that they meant a community that has sustained itself for at least some period of time, and that other companies should also be interested and working with the community. This latter point was well argued by an interviewee who explained that other companies' involvement is a good indicator that the product is of good quality. More companies taking an interest will cut the effort of each one and costs can be shared, *"to share the costs of polishing that piece of software, if you are the only company involved then you should ask yourself the question why is this. Are we so innovative that we were the first ones to think of this? It might happen that there is something about the software that makes it tricky to manage... so being able to share the development and maintenance costs is important"*.

Theme/Memo/sub-theme	Quotations	Respondent
License and IPR Regime	“We help our customers make better use of open source... we can negotiate any support agreement they want. In some cases we provide consulting and outsourcing services... customers are more comfortable with open source because they know that at the end of the day they know that nobody can take it away from them. It is a matter of getting (our company)to support them – they will trust that from an open source standpoint because in the worst case if one vendor doesn’t meet their support needs they can get another vendor to support them – they are not locked into any one vendor for support”.	Open source Consultant in Company B
Dual licensing	“We don’t do dual licensing, because we release some stuff for free, for free download and then you can buy a support contract. But our main way is by including components within bigger software product... It’s down to, there is a different sort of business model. The business model of Linux is essentially around the ecosystem”.	Open Source Manager in Company A
Most oft adopted licenses	“Where the project has been set up externally by another group then, we work obviously with that. So we contribute code to Linux, which is under the GPL. We contribute code to Apache, which is under the Apache license. Where we actually open source a project and create a new project ... we tend to do it around similar to our Apache license, but one we’ve defined. We prefer the Apache type license. Because it enables open source code to be mixed with closed source code and to create combined products, which is what we see as a lot of the way the industry is going and the usage is going and also, it enables a good sort of business model around. It works well from our point of view”.	Open source Consultant in Company A
Community Approach	“The goal in developing a community, you know, beyond the desire to have that community to support the open source project, what we wanted is a self sustaining venue in which people could get their questions answered .Find information. And in which we had Evangelists that were not from (our company). If you look at a traditional software company like, they may have a news group or a forum that people can ask questions and sometimes can answer ... you go in there and you interact with a support team. That tends to be a fairly heavyweight process. It takes time. What we wanted was this... we wanted a self sustaining community where you know, seven, twenty four, sixty five, you know. There is just people discussing, you know, chatting about one of their experiences and their problems, possible solutions. Once you get enough people together, they begin to answer one another’s questions. It’s not ... clearly, from a development standpoint, we still stay engaged and monitor ... we keep our finger on the pulse. It’s self sustaining. It’s out there and it’s a value”.	Consultant of Company B
Match between company and community	“To share the costs of polishing that piece of software, if you are the only company involved then you should ask yourself the question why is this. Are we so innovative that we were the first ones to think of this? It might happen that there is something about the software that makes it tricky to manage... so being able to share the development and maintenance costs is important”.	Open source Manager at Company B
The issue of requirements	“There is no formal requirement to process, right, in typical open source. Part of our job is, when we are getting ready to	Open source Manager at

	<p>think about an extra release, we'll ask the (Company A) community... What are your requirements? What do you guys need? It's our job to kind of on the (Company A) side to collect all that and to sift through it and represent what we believe are the best set of priority requirements. At the end of the day, (our company) is the one paying the salary for the open source programme. It's a set of requirements and because these are our people who are paying for it, gives us the right to say, we want to work on this stuff. It had to happen within a set of constraints. Our play here is the collaborative one with other community members including business partners. So, to be honest, it's a bit of a balancing act. We've got a set of (company) priorities and there is a set of things that community is interested in and we are interested in how the community is interested in our stuff, because it pays off. It's got to be something here that is also interesting to the community. I don't know that there is any magic there, other than it's just an experience that we've gathered over the years of having done this".</p>	Company A
Incentive schemes	<p>"A viral plank. We wanted something that would grow from the bottom up, you know. Not only would the chief technology, chief probation officer say, here's our strategy for solving in this part of our IT strategy...So, in other words, capturing the hearts and the minds of the developers out there".</p>	Company B employee
Modified Development Process	<p>"An internal open source bazaar, and it is run rather like a bazaar. Anyone can start a project. There are minimum controls around it to ensure that people don't put code in it that is inappropriate. But because it draws on people from different parts of the corporation with different levels of knowledge you can have some interesting mixtures of people who understand how to connect the internal systems together with some behaviour from external systems".</p>	Open source developer at Company A
Shift in governance and control	<p>"Letting go of control... (which is only possible) if you've got a shared vision. Obviously, there needs to be something that's sort of binds people together. If you've got the shared vision then you can sort of let go of control".</p>	Company A Manager
Managing innovation	<p>"(Our company) is a big company with a lot of access. We are really concerned about the pedigree, the providence of source code of a commercial product. That was easy when we all wrote it and was all behind (our company) firewall and we knew exactly where it came from and we knew who wrote it and we know when they did. But now, when you throw open source into that... number one is people are not (our company) employees. Sometimes we don't even know... the people who have written it have long since gone... The last thing we want is to see a law suit. There is a big part of what (our company) does to continue to assure the integrity both of what we contribute to open source and then we bring that back into (Company A) that everything is clean and it can go out in the commercial product. It's just making sure that what we've got is clean and it can go out.... The amount of risk that we are willing to take there is going to be very different from the amount of risks some start ups company will get".</p>	Open source Director at Company A

Table 1. Selection of analysis themes/memos stemming from data collected from two large, global technology companies

The second most important factor was stated to be the viability of the technology and the business value (direct or indirect) that the technology could bring to the company. This issue was closely linked to the question of license. However, the related issue to be considered is whether the application the open source software is being combined with will eventually be distributed to customers. If the answer to this question is no, then there is less reason to be concerned about the license. But if the answer is yes, then the potential for leaking their own IPR through viral licenses is a serious concern.

The Issue of Requirements

A significant question for companies adopting open-sourcing is the direction a software product will take in the future. More specifically, how, and by what means they can influence this direction, for example by feeding in new requirements for development effort, or encouraging appropriate interfaces to match their own in-house efforts. This poses the question of the degree to which companies can 'dictate' requirements for a product that is part produced by a community and part by company developers? As one manager pointed out, it is more a process of gauging each other (company and community) and balancing the needs of both. This is a delicate process and many companies, including the two studied here, have upset communities with a demanding approach to what is needed. Over time and experience both companies have learned to be relaxed and less directive.

One coping strategy described is to keep an in-house team of developers familiar with the community and the code and who can do the tasks and changes that the community is reluctant to do. Such a group may develop required functionality either for proprietary use, or to feed back to the open source project. This is an added expense but may have other value in fostering wider in-house knowledge and maintaining a resource as insurance against over-dependence on the open source community. Indeed, it is understood that the community will always be a little more risky.

Incentive Schemes

Companies use communities to get access to ideas, and innovation through collaboration. Various incentives are utilized to harness the sustained interest of a mass of developers around a product of interest. Managers from both companies studied provided a similar list of strategies used including short-term contracts, bounties, even full-time employment, but the most relevant form of enrolment was the technology itself, "*It was the technology. They were really excited about the technology*" (Company B employee). The interviewees were clear that the best way to enroll open source community members was to offer them an interesting technological challenge and access to leading edge ideas. Of course the other strategies mentioned are employed, and usefully, but technology and what it offers is the most reciprocally valuable approach for community-company collaboration.

Process: Modified Development Process

This final category covers the overall approach to managing the development process when open sourcing. Our respondents report a number of strategies such as creating a Steering Committee comprised of employees and community developers. This may have started with a committee where more members belong to the company. However, the cases we studied showed how this needed to be re-thought if the company wanted positive feedback from the community or to sustain their interest and commitment. Very quickly community interest in a project began to flag and the company was forced to reassess its attempt at exercising any direct control over the wider project. As we were told, it learnt that in order to keep any control it had to 'let go of control'. More community members were then asked to join the steering committee, voted on board depending on their experience and contribution to the project.

Managing Innovation

Part of the problem of attempting to encourage innovation is how to manage 'what happens next' - and it can suggest other open innovation opportunities that may be less easy to control. This can for example lead to the possibility of security leaks, divergent technology strategies or creation of competing products, not to mention contamination of work by viral licenses. Code itself can become vulnerable, especially if not all the employees are well-versed in the particularities of open source licenses and their implications for code hygiene. A positive aspect of easing control and distributing it more widely in and beyond the organization are changing work practices of employees, their motivation and their sense of collectivity through the need to share ideas and work.

Both companies studied informed us about various initiatives that had emerged from the adoption of not only open source software but the open source way of working and belief system (i.e. sharing, collaboration, and better coordination). One of the companies has "*an internal open source bazaar, and it is run rather like a bazaar*" (Company A developer). Innovation

encouraged internally in what used to be a traditional organization. Interviewees claimed that such changes were triggered by the adoption and awareness of open source.

CONCLUSION

This paper lays out some initial findings from two case studies of corporate engagement with open source communities. The paper focuses on how companies create innovative coping strategies to manage and control the risks involved with open source innovation and their relationship with an open source community - an attempt to control agent costs by managers.

The basic premise of company involvement stems from a worldview that sees collaborative and collective work as leading to the ability to harness more ideas, more eyeballs, which will in time lead to cutting edge innovation. Sometimes the goal is to foster competition in the market to disturb possible monopolistic practices, but on other occasions open sourcing collaboration can be a means to reduce competition and to better secure a place within a stable market. Open source, and all that it implies, - open process, greater flexibility, better scalability and faster time to market – suggest that it is no surprise that companies have attempted to find a way to work with this movement. It has nevertheless been a difficult process for companies who have different goals, different governance structures and software practices.

Our data reveals an evolution in practices as the large companies studied have worked through their understanding of what it means to work with open source communities. Such change occurs through use and engagement with the software or product itself, as much as with the community. Each company that has taken an open source route, be it in software or only process, has had a different business model structured around open innovation ideas. It is the open innovation business model based on collaboration and a slightly divergent manner of creating a profit, be it through the sale of complementary services or a strategic move of limiting competition that has been sought after by companies. Open source, according to our data, has in both companies lead to a 'culture of innovation' where innovative outcomes (products) have become the catalyst for a structure of innovation (process change). There is a degree of sharing and process change that indicates an evolution in practices in both companies which our respondents believe to be stimulated by a greater adoption of open source software and ideas in-house.

Our company focused study has facilitated our understanding of when, how, and why companies turn to open source adoption, and the various strategies they employ to ameliorate their concerns about such a strategy. The next stage of the study intends to provide a more balanced understanding of this relationship through an analysis of the implications of such collaboration for open source communities themselves. Some initial findings suggest that these communities do increasingly encourage company involvement, seen as providing sponsorship, and supporting the longevity of both the product and community, job possibilities and a chance to prove developer ability.

REFERENCES

1. Agerfalk, P. and B. Fitzgerald (2008) "Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy", *MIS Quarterly*, **32** (2), pp. 385-400.
2. Alchian, A. and H. Demsetz (1972) "Production, Information Costs, and Economic Organization," *American Economic Review*, **62** (5), pp. 777-795.
3. Butler, S., D. Adebajo and H. Ismail (2008) "Open Source Software and Leveraging of Business Effectiveness in Smes - a Case Study", *Ice-B 2008: Proceedings of the International Conference on E-Business*, pp. 93-100.
4. Capra, E. and A. I. Wasserman (2008) "A Framework for Evaluating Managerial Styles in Open Source Projects", *Open Source Development, Communities and Quality*, **275** pp. 1-14.
5. Carmel, E. and R. Agarwal (2002) "The Maturation of Offshore Sourcing of Information Technology Work", *MIS Quarterly Executive*, **1** (2), pp. 65-77.
6. Chesbrough, H. W. (2006) *Open Business Models* Harvard Business School Boston, MA.
7. Chesbrough, H. W., W. Vanhaverbeke and J. West (Eds.) (2007) *Open Innovation: Researching a New Paradigm* Oxford University Press London.
8. Comino, S. and F. M. Manenti (2008) *Dual Licensing in Open Source Software Markets* Address: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=985529.
9. Dinkelacker, J., P. Garg, R. Miller and D. Nelson (2002) "Progressive Open Source". in *Proceedings of the 2002 ACM International Conference on Software Engineering (ICSE'02)*, pp. 177-184.

10. Eisenhardt, K. M. (1989) "Agency Theory: An Assessment and Review", *Academy of Management Review*, **14** (1), pp. 57-74.
11. Franke, N. and E. v. Hippel (2003a) *Finding Commercially Attractive User Innovations* MIT Sloan School of Management
12. Franke, N. and E. Von Hippel (2003b) "Satisfying Heterogeneous User Needs Via Innovation Toolkits: The Case of Apache Security Software", *Research Policy*, **32** (7), pp. 1199-1215.
13. Glaser, B. G. and A. Strauss (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine, Chicago.
14. Hirschheim, R. and M. C. Lacity (2000) "The Myths and Realities of Information Technology Insourcing", *Communications of the ACM*, **43** (2), pp. 99-107.
15. Holck, J., M. H. Larsen and M. K. Pedersen (2005) "Managerial and Technical Barriers to the Adoption of Open Source Software", *Cots-Based Software Systems, Proceedings*, **3412** pp. 289-300.
16. Huston, L. and N. Sakkab (2006) "Connect and Develop: Inside Procter & Gamble's New Model for Innovation", *Harvard Business Review*, **84** (3), pp. 58-66.
17. Huston, L. and N. Sakkab (2007) "Implementing Open Innovation", *Research - Technology Management*, **50** (2), pp. 21-25.
18. Jensen, M. C. and W. H. Meckling (1976) "Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure", *Journal of Financial Economics*, **3** pp. 305-360.
19. Kaiser, K. M. and S. Hawk (2004) "Evolution of Offshore Software Development: From Outsourcing to Cosourcing", *MIS Quarterly Executive*, **3** (2), pp. 69-81.
20. Kern, T., L. Willcocks and M. C. Lacity (2002) *Netsourcing: Renting Your Business Applications and Services over a Network*, Financial Times/Prentice Hall,
21. Lakhani, K. R. and R. G. Wolf (2005) "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects" in *Perspectives on Free and Open Source Software*, (Feller, J., B. Fitzgerald, S. Hissam and K. R. Lakhani eds) MIT Press.
22. Lindman, J., M. Rossi and P. Marttiin (2008) "Applying Open Source Development Practices inside a Company", *Open Source Development, Communities and Quality*, **275** pp. 381-387.
23. Martin, K. and B. Hoffman (2007) "An Open Source Approach to Developing Software in a Small Organization", *Ieee Software*, **24** (1), pp. 46-+.
24. Melian, C. and M. Mahrng (2008) "Lost and Gained in Translation: Adoption of Open Source Software Development at Hewlett-Packard", *Open Source Development, Communities and Quality*, **275** pp. 93-104.
25. Metiu, A. and B. Kogut (2001) "Distributed Knowledge and the Global Organization of Software Development" *Wharton School, University of Pennsylvania*, Philadelphia.
26. Olson, M. (2005) "Dual Licensing" in *Open Sources 2.0: The Continuing Evolution*, (DiBona, C., M. Stone and D. Cooper eds) O'Reilly, Sebastopol, CA, pp. 71-90.
27. Pulkkinen, M., O. Mazhelis, P. Marttiin and J. Meriluoto (2007) "Support for Knowledge and Innovations in Software Development - Community within Company: Inner Source Environment", *WEBIST 2007: Proceedings of the Third International Conference on Web Information Systems and Technologies, Vol SeBeG/eL*, pp. 141-150.
28. Sakkab, N. (2002) "Connect & Develop Complements Research & Develop at P&G", *Research - Technology Management*, **45** (2), pp. 38-45.
29. Shaikh, M. and T. Cornford (2008) "Open-Sourcing: On the Road to the Ultimate Global Source?" in *2nd Global Sourcing, Services, Knowledge and Innovation Workshop* Val d'Isere, France, March 2008,
30. Strauss, A. and J. Corbin (1999) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, Sage Publications,
31. Välimäki, M. (2003) "Dual Licensing in Open Source Software Industry", *Systemes d'Information et Management*, **8** (1), p. 63+.
32. Ven, K. and H. Mannaert (2008) "Challenges and Strategies in the Use of Open Source Software by Independent Software Vendors", *Information and Software Technology*, **50** (9-10), pp. 991-1002.

33. von Hippel, E. (1998) "Economics of Product Development by Users: The Impact of Sticky Local Information", *Management Science*, **44** (5), pp. 629-644.
34. von Hippel, E. (1994) "Sticky Information' and the Locus of Problem Solving: Implications for Innovation", *Management Science*, **40** (4), pp. 429-439.
35. von Hippel, E. (2001) "Innovation by User Communities: Learning from Open-Source Software", *MIT Sloan Management Review*, **42** (4), pp. 82-86.
36. von Hippel, E. (2005) *Democratizing Innovation*, MIT Press, Cambridge, MA.
37. von Hippel, E. and G. v. Krogh (2003) "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science", *Organization Science*, **14** (2), pp. 209-223.