Lost and Gained in Translation: Adoption of Open Source Software Development at Hewlett-Packard

Catharina Melian[1] and Magnus Mähring[2]

1   Stockholm School of Economics, Center for Media and Economic Psychology, Box 6501, SE-11383 Stockholm, Sweden
catharina.melian@hhs.se
WWW home page: http://www.sse.edu

2 Stockholm School of Economics, Department of Marketing and Strategy, Box 6501, SE-11383 Stockholm, Sweden
magnus.mahring@sse.se
WWW home page: http://www.sse.edu

Abstract. What happens when an organization form that has emerged in one context is brought into a different context? In this paper, a longitudinal field study approach is used to explore how Hewlett-Packard (HP) molded open source software development (OSSD) into a proprietary software development approach called "Progressive Open Source" (POS). With the help of actor-network theory, we understand this as a process of translation and find that some central characteristics of OSSD where lost in the translation into POS while other characteristics were gained.

1   Introduction

The advent of open source software development (OSSD) has spurred considerable interest not only amongst individual systems developers participating in voluntary and unpaid software development over the Internet, but also increasingly amongst software development organizations seeking to improve the quality and productivity of their products and services [23, 34]. In this paper, we explore what happens when OSSD is brought into a traditional, high-performing, software development organization? With the help of actor-network theory, we explore this process of *translation* [6] in Hewlett-Packard, where key actors shaped and constructed the idea of Progressive Open Source (POS) through interactions and through the meanings that they assigned to their work, interactions and experiences [18]. The purpose of the study is thus to explore the translation of an extra-organizational form of organizing, open source software development, into a corporate context.

2   Background

Typically, open source software development takes place in Internet-based communities of software developers who voluntarily collaborate to develop free software that they or their organizations need [26, 34]. OSSD can be seen as a new way of organizing—different from software development conducted according to the traditional paradigm [24, 28]. While there is a lot of interest in adoption and adaption of OSSD within a corporate context, and development platforms for OSSD are now offered in corporate versions [3], little research has so far been devoted to the emergence and anatomy of "hybrid" organizational forms that combine OSSD practices with corporate requirements and needs [29].

Actor-network theory (ANT) addresses how technology influences and is influenced by social elements in a setting over time [6, 7, 20]. According to ANT, the ideas, values and intentions of social actors become inscribed in a technology through interaction. As a result, intentions become immutable [1, 2, 18]. The process through which this occurs involves creation of networks of aligned interests: actors form alliances, enroll other actors and use non-human actors (artifacts, technologies) to secure their interests. The resulting actor-networks are made up of human and non-human actors [7, 19].

The creation of an actor-network, referred to as translation or "the methods by which an actor enrolls others" [5, p. xvii], results—if successful—in a stable actor-network that encompasses non-actors. In our case, the focal non-human actor is the idea or technology of open source. ANT predicts that the translation of OSSD into a particular context includes some transformation of the idea/technology; the involved actors become aligned in interests, commitments and obligations and as part of this development the idea/technology is instantiated or situated [32].

In this paper, we employ two supplementing views of translation. Steiner [31] emphasizes the journey (or re-formation) of an idea, while Callon [6] emphasizes the formation of the network of actors that in an interactive process constructs itself and incorporates the idea/technology. With this combined view of translation, the process can be described as follows:

1. *Initial act of trust* [31] and *problematization* [6]: At the beginning of the process, a translator assumes that a particular model or idea (e.g., OSSD) makes sense and that it can be reproduced in a different context [31]. The translator opens up for new ways of thinking, and through social interaction the identities and interests of a set of relevant actors start to emerge [6].
2. *Incursion* and *extraction* [31] and *interessèment* [6]: The translator(s) raids the original idea for sense, and bring back that which makes particular sense for the organization. Relatedly, actors attempt to define "the problem" by manipulating and (re)negotiating relationships between concerned actors.
3. *Incorporation* [31] and *enrolment* [6]: In this stage, actors mold, or give particular form and body, to those aspects of an idea that fit into the organizational texture and the idea becomes incorporated. As this happens, actors come to accept

the designated roles attributed to them and the formation of an actor-network occurs.

4. *Appropriation* [31] and *mobilization* [6]: At this stage, the original idea is transformed and becomes intertwined with the organization. The idea is also appropriated—the new idea has taken over all relevant meaning from the old idea and the original idea has no separate identity anymore. Correspondingly, some actors come to represent others and mechanisms are put in place that stabilize the actor-network and keep actors behaving in accordance with interests. At this point, the actor-network and its ideas, technology, and artifacts have become institutionalized and are no longer seen as controversial.

5. *Reciprocation* [31]: While the actor-network is already at this point stable, the translated idea may at this point come to enhance the original idea. For example, the idea resulting from a translation may, if successfully institutionalized, contribute to enhancing the stature of the original idea.

The idea/technology in focus in this study is OSSD, and the translation studied is the re-formation of OSSD into POS within Hewlett-Packard. The challenge of adapting OSSD in a corporate context [29] can thus be seen as dependent on the molding of the idea of open source into a form that fits the context and the corresponding formation of a stable actor-network that performs the appropriation of the idea and sustains the idea and its use. On the basis of this argument and our brief exposé of ANT, we argue that ANT is well suited to advance our understanding of the challenge of corporate OSSD adoption. However, we have not found any studies of corporate OSSD adoption that employ ANT and only one study of OSSD using ANT outside of the corporate context [33]. This sets the stage for our study.

3   Methodology

In 2001, the first author established initial contact with Hewlett-Packard (HP). An agreement was reached to document HP´s efforts to introduce the new software development paradigm within the organization. Access to contacts, data and other resources was provided and the field researcher was assigned an office cubicle at HP labs in Palo Alto, California.

The longitudinal field study [4, 25] encompassed nine months in the field at HP between October 2001 and April 2002, in August 2002, and in May 2003. Time spent in the field included direct observation of work, participation in meetings, participation in workplace conversations and dialogues, and extensive field note writing [8, 9]. The study also encompassed 52 formal interviews (semi-structured interviews averaging two hours that were taped and transcribed), corporate and external documents and other forms of secondary data [27]. The data was subsequently coded using NVivo software for qualitative data analysis.[1] For the purpose of this paper, an

---

[1] For more information on research methodology and case details, see Melian [22].

actor-network lens was selected to further deepen the emerging understanding of the transformation (or translation) of OSSD in the HP context. In particular, the translation model presented in the previous section was used to structure, present and analyze the case.

4   The Translation of Open Source at Hewlett-Packard

In this section we briefly summarize the chains of events that resulted in the widespread use of progressive open source (POS) within Hewlett-Packard Company in the period from 2001 to 2003. Then as now, HP was an engineering company where product development and hi-tech engineering were high-status activities seen as central to the organization's survival and success.

4.1   Initial Act of Trust and Problematization

At the end of the 1990s, actors in several units within HP were involved in discussions on challenges related to product development. The 1990s had seen extremely rapid development in IT products and services as well as a rapid increase in demand for such products and services. There were recurrent discussions within different software development units in HP on the need to reform the software development process and a number of possible remedies and aspirations were studied. In order to increase cost efficiency and shorten time to market, it was perceived necessary to increase use of third parties (for collaborative and outsourced development) and to conduct around the clock ("24/7") development. In addition, software modularization and extensive, systematic reuse of software components was seen as an approach to avoid "reinventing the wheel" and to increase component compatibility.

> *A discussion came up about how we need to find a way we can work better with third parties, and there was some bantering about how there is Open Source, but we have security needs that we have to make sure that we don't expose our IP[2].... — HP Software developer.*

Many developers within HP were already tinkering with OSSD in their free time and examples of successful open source projects (Linux, Apache web server) were well-known within HP.

The idea of open source as an idea was thus met with an initial act of trust: actors within HP believed OSSD to offer an attractive approach and solution to perceived problems and challenges [31]. The descriptions and definitions of problems were, from an ANT perspective, not objective depictions of a "real" reality, but rather expressions of interests and interpretations by actors involved in the organizational discourse [7, 19]. The engineering community within HP was dominant in this initial identification of OSSD as a preferred idea for organizational betterment and object of trust.

[2] Intellectual property.

4.2   Incursion and Extraction, and Interessèment

The chief technology officer (CTO) of the Imaging and Printing business area, Lee Caldwell, initiated a project to address these issues, called CDP (for Collaborative Development Program). Robert (Rob) Miller, a senior scientist at HP, was appointed to lead the project.

Rob was just having the idea that this is just one of his jobs, and he kind of looking around and it was a combination of both Open Source and just the issue around collaboration and being able to tap into the expertise around the business and the company. — HP Software developer.

Miller started to search for successful examples of collaboration between development teams in HP and found the Sirius Firmware Cooperative, a group that had developed new firmware architecture for multifunction peripherals in the early 1990s. Four geographically dispersed teams collaborated using a common work process with interconnected core teams and a well-defined peer review process. In order to learn more about Sirius, Miller contacted Cathy Ammirati, head of the Sirius co-op. Ammirati was subsequently invited to join the CDP effort, while Sirius continued to function as a separate entity. In parallel, people involved in the CDP effort read the open source literature and tapped into the discourse on the potential of corporate adoption of open source present in Silicon Valley at this time.

I started learning about open source, and did the typical thing most people do, read 'The Cathedral and the Bazaar',3 started reading up as much as I could about it and said its very fascinating to see how we could bring this into HP, and how it could work inside of HP in such a structured environment of Hewlett-Packard… — Software developer involved in setting up CDP.

While the CDP effort was initiated by the CTO and had strong corporate backing, engineers also maintained and nurtured open source as "their thing". For example, the adoption of open source in the corporate setting was seen as a way to cope with the organizational "reality" shaped by the administrative/managerial community. In this way, different actors were able to selectively assign meaning to the new idea [31] and inscribe it with their intentions [1, 18].

4.3   Incorporation and Enrolment

By now the approach of CDP was to try to learn and evaluate what open source could mean to HP through experience with real software development tasks. By July 2000, the first tasks were set up in the CDP environment, which was still under development:

...we are driving the car down the road and we are improving it as we go. We're in there tinkering under the hood, somebody's hanging off the side fixing the door lock…. — Manager, CDP.

---

3 See Raymond [26].

Independent of the CDP effort, senior scientist Pankaj Garg at HP Labs had at this time initiated a software development environment, Corporate Source (CS), devised to function as an open source environment restricted to HP employees. Most of the development effort went into CDP as a development platform (including development, storage and communication tools) was purchased. CS was used separately and was continuously compared with CDP. In late 2000, the concept of progressive open source (POS) was put forward as a way to describe open source initiatives within HP. HP's increasing attention *to open source was also reflected in public speeches by HP's CEO:*

> *The open source movement is natural, inevitable and creates huge benefits. It's part of the next wave of computing, and that will involve participants and users within the industry in open source. — CEO Carly Fiorina, October 2000.*

Within CDP, a set of defined benefits were formulated that matched the previously identified challenges for HP software development. These included increased code reuse, improved code quality, faster development, faster debugging, and selected partner collaboration. In addition, rapid re-deployment of key developers between projects was seen as facilitated by the common platform and work process. Leading actors posted internal "tech reports" (later presented at public conference [10, 11]) that outlined the benefits of adopting open source within Hewlett-Packard.

The creation of the "POS" concept can be understood as creation of a vessel that facilitated the extraction of meaning from OSSD and the transfer of relevant meaning from the original to the new concept [31]. Also, this phase sees the introduction of an important non-human actor: the CDP development platform and the key attributes of the platform and related services (such as security, search and navigation, migration, and technical support).

4.4.   Appropriation and Mobilization

In late 2001 and early 2002, POS was established as a new and paradigmatically different approach to software development within HP. The different breeds and varieties of "open source" initiatives within HP are now explicitly and systematically described as being part of POS, with the notable exception of Sirius. Over a period of 18 months from the original launch of the CDP development platform, there was considerable growth in the number of developers on the CDP platform. The pilot project had encompassed 600 developers. In April 2001, CDP moved to a production environment and by August there were over 2000 users, ten percent of which outside HP. By January 2002 there were 3500 users and over the next few years, the CDP environment would grow to over 10,000 users.

The diffusion of the CDP platform was supported through increasingly well-defined roles: executive champions (the chief technology officer and the chief information officer), sponsors, R&D change leaders, and information technology staff. The executive champions stressed the need for collaboration between development groups in the organization and applied pressure on software managers to transition to the CDP platform. The sponsors committed development resources and the R&D

change leaders were appointed to enable and support a dialogue of culture change and POS training. The IT staff provided the critical collaboration infrastructure on a 24x7 supported basis.

The standardization of development work through the CDP development platform was a painful process for many development teams, but in most cases teams decided to adopt and adapt to the platform and related work methods. The CDP platform thus constituted a mechanism—inscribed with intentions—that stabilized the actor-network and restricted actors to behave in accordance with inscribed interests [6]. POS and CDP had by this time become dominant concepts to describe activities within HP aiming at open, collaborative development. Open source was no longer an important or frequently used concept. In accordance with Steiner's [31] view of translation, the new idea had taken over all relevant meaning from the old idea and the original idea had no separate identity in this particular context anymore.

As more developers transition into the CDP development environment, the development platform becomes an increasingly important non-human actor: Because of the standardization on CDP, third-party developers collaborating with HP had to transition into CDP. Acceptance of the CDP environment and its rules and roles became a necessary prerequisite for continuing to do business with HP.

However, as the use of CDP increased quite rapidly, scalability problems appeared: there were several breakdowns of the development platform during the growth period during 2002. In spite of the breakdowns, the actor-network had become strong and stable enough to impose itself on a gradually larger percentage of HP developers who are faced with a "take it or leave it" offer of joining CDP. For individual developers, there was a risk that the change in development environment would manifest itself in loss of productivity and motivation.

More profound changes in the software development process also started to appear. The previous common practice of working "under the radar" was no longer a preferred or even allowed approach to development work. Instead, all development work needed to be out in the open on the CDP platform. Software developers perceived increased pressure as a result of the open and transparent process: In the POS/CDP environment, every contribution, absence of contribution, mistake, misunderstanding, and spontaneous comment was directly attributable to an identified individual. This was in direct contrast to OSSD, where contributing developers are normally anonymous. While POS technology facilitated openness it also besieged privacy and exposed the individual:

> *...the fishbowl aspect of feeling like you're being watched with every little thing you do. ... It does affect the way that I post to the forum. — Software developer.*

In relation to this, HP managers started to comment to each other that a new "breed" of employee was needed—people that could cope with the "fish bowl" visibility of their actions and not let that hamper their openness or they way they interact with others.

4.5   Reciprocation

Through POS, HP managed to improve the efficiency, speed and quality of software development. Unlike the original idea of open source, the POS reincarnation also adhered to traditional criteria for software development in that development work was successfully conducted within time and budget. In this sense, POS can be seen as contributing to the status of OSSD [31] by demonstrating its superiority as development approach also in the (foreign) corporate context.

5   Discussion

We have followed the OSSD idea/technology as it traveled from its original, extra-organizational context and translated into POS; transformed, yet partly the same [17, 31]. Main differences between open source software development and progressive open source in Hewlett-Packard are found in **Error! Reference source not found.**.

POS became different from OSSD in order to survive and prosper; that is, in order to become an integral part of a stable and growing actor-network [6, 18]. In particular, our analysis of the case shows that key elements of the HP organization were perceived to be at odds with OSSD *as organizing*. The formal hierarchy and related reporting and control structure—including financial and other goals per unit and area, individual evaluations and rewards—as well as the HP culture were also seen as dominant and immutable features of the HP organization.

Consequently, the incursion and extraction of meaning [31] from OSSD by HP actors focused on those elements of OSSD that were compatible with key organizational design elements of the existing HP organization. The translation of OSSD into POS was mainly intended to change work practices related to software development work, to extend those work practices to incorporate selected corporate partners in controlled ways, and to allow tinkering with open communities for software development.

In this respect, POS can be considered a success. The goals of 24/7 development, more effective reuse of technology and increased number of parallel development processes were achieved. The spread of POS within HP has also been considerable and the practices developed during the case period are still used. In other words, a stable and growing actor-network that formed by the originators of POS, over 10,000 developers, the idea of POS and the CDP development platform. It can also be argued that the success of POS within HP has enhanced the original idea of OSSD (i.e., reciprocation) by making OSSD a more credible idea/technology for organizational adaption [31].

Table 1. A Comparison of Open Source and Progressive Open Source

Open Source Software Development (OSSD) [Based on 12, 15, 16, 26, 30, 35, 36]	Progressive Open Source (POS)
No formal organization, rather a social phenomenon.	Formal organizational entity, sometimes involving 3rd party developers.
Gift system [21].	Market system.
Underlying notion: abundance.	Underlying notion: scarcity.
Each gift is part of system of reciprocity. Every gift has to be returned in a perpetual cycle of exchanges.	Calculated gains and losses, utility maximization.
Very open and free flow of information.	Progressively more open – however controlled flow of information between trusted partners internal and sometimes external to the formal organization.
Voluntary teams working on projects.	Designated teams, i.e. employees and/or contractors working on defined projects loyal to a formal organization.
Meritocracy/benevolent dictatorship.	Traditional hierarchical organization.
Sociological phenomenon. Has the potential of bringing together software practitioners regardless of their prior record (employment, education, age, status, etc.). Only the validity of the contribution counts.	Organizational phenomenon. Technology in place that enables collaboration between software engineers employed and/or contracted by the organization and trusted partners.
No time and budget constraints.	Often strict time frames and budget restrictions.
Potentially limitless influx of knowledge and skills – infinite number of developers.	Controlled number of developers. Significantly smaller than Open Source community.
Public good: Non-rival (one person's consumption of the good does not reduce the amount available to another) and non-excludable (excluding others from consuming the public good is difficult or impossible)	Privately owned/controlled intellectual property rights.
Challenge: How to get people to contribute and avoid free riding. How to coordinate a large group of individuals that want to contribute.	Challenge: How to move from a control based organization to an organization based on empowerment and trust. How to reward collaborative behavior.
Motivation to contribute: Identity (honor, reputation, ego-boosting) Reciprocity Efficacy: individuals contribute because they have some impact on the system. Art and beauty of solving problems Work as vocation	Motivation to contribute: Monetary Patents Ranking system/reward-system

However, several interesting apparent paradoxes can be observed through comparison between OSSD and POS. POS has not only failed to fulfill some of the promises of the OSSD idea: in some aspects, POS produces the opposite of the original OSSD idea. While OSSD takes place in loosely coupled cyberspace communities [16] with weak ties [14] and without many of the restrictions and requirements of corporate contexts [16], the introduction of POS at HP increased the level of structure and standardization of development work. Standardization included forms of expression, language use, defined layers of openness, and the requirement to safeguard intellectual property also in controlled "open" source projects. In some ways, this was far from the corporate culture of "cowboy development", while in other ways it further enabled micro-managing by managers.

There were also considerable resemblances between OSSD and POS, such as transparency of the development process, constant up-to-date documentation, the large number of "eyeballs" involved in debugging, and 24/7 development. Together, these similarities suggest that central aspects of the open source development work process remained quite intact through the translation process. On the individual level, however, the differences were fundamental. Whereas OSSD enables flexible participation in development work with limited real life consequences, POS has created quite the opposite work environment. In the organizational context, with long-term social relationships and well-defined hierarchy, POS makes the quality and level of contribution of individual developers highly visible. This visibility shapes an environment where an underperforming employee can neither run, nor hide from scrutiny—a virtual panopticon in which every mistake is likely to get noticed [13]. In addition, the balance between openness and safeguarding of intellectual property places intellectual property decisions with individual developers and makes these decisions a part of daily practice.

6   Conclusions and Implications

Using actor-network theory, this research has studied the appropriation of open source practices in HP as a translation of an organization form from an extra-organizational context to a corporate context. The study shows that in the translation of open source into the corporate context, some fundamental aspects of OSSD are lost, while other fundamental aspects of the new hybrid form of software development—progressive open source—are shaped.

The study strongly suggests that a translation of OSSD into the corporate context may come to yield substantial and lasting benefits to efficiency, speed, quality, flexibility and reusability in software development. However, our study also reveals that while many of the characteristics of the work process may remain intact through the translation, the organizational characteristics of open source in the corporate context, particularly in terms of monitoring and control of the individual, may starkly contrast traditional open source environments. The implications of these findings for stake-

holders in organizations considering OSSD adoption are far-reaching and will hopefully inspire further research.

Acknowledgments

References

1. Akrich, M. The de-scription of technical objects, in W.E. Bijker and J. Law, ed., *Shaping technology/building society: Studies in sociotechnical change*, Cambridge, MA: MIT Press, 1992, 205-224.
2. Akrich, M., and Latour, B. A summary of a convenient vocabulary for the semiotics of human and nonhuman assemblies, in W.E. Bijker and J. Law, ed., *Shaping technology/building society: Studies in sociotechnical change*, Cambridge, MA: MIT press, 1992, 259-264.
3. Augustin, L.; Bressler, D.; and Smith, G. Accelerating software development through collaboration, in *Proceedings of The 24th International Conference on Software Engineering* Orlando, FL, 2002559-563.
4. Barley, N. *The innocent anthropologist: Notes from a mud hut*. Long Grove, IL: Waveland Press, 1983.
5. Callon, M. The sociology of an actor-network: The case of the electric vehicle, in M. Callon, J. Law, and A. Rip, ed., *Mapping the dynamics of science and technology*, London: Macmillan, 1986.
6. Callon, M. Some elements of a sociology of translation: Domestication of the scallops and the fishermen of st brieuc bay, in J. Law, ed., *Power, action and belief: A new sociology of knowledge?*, London: Routledge and Kegan Paul, 1986, 196-233.
7. Callon, M., and Latour, B. Unscrewing the big leviathan: How actors macro-structure reality and how sociologists help them to do so, in K. Knorr-Cetina and A.V. Cicourel, ed., *Advances in social theory and methodology: Toward an integration of micro-and macro-sociologies*, London: Routledge and Kegan Paul, 1981, 277-303.
8. Czarniawska, B. *Narrating the organization. Dramas of institutional identity*. Chicago and London: The University of Chicago Press, 1997.
9. Czarniawska, B. *Writing management*. Oxford University Press, 1999.
10. Dinkelacker, J., and Garg, P.K. Corporate Source: Applying open source concepts to a corporate environment, in *Proceedings of Proceedings of the First ICSE International Workshop on Open Source Software Engineering*, Toronto, Canada, 2001.
11. Dinkelacker, J.; Garg, P.K.; Miller, R.; and Nelson, D. Progressive open source, in *Proceedings of Proceedings of the 24th International Conference on Software Engineering (ICSE)*, Orlando, FL., 2002177-184.
12. Fitzgerald, B. The transformation of open source software. *MIS Quarterly*, 30, 3, 2006, 587-598.
13. Foucault, M. *Discipline and punish: The birth of the prison*. Harmondsworth: Penguin, 1977.

14. Granovetter, M.S. The strength of weak ties. *The American Journal of Sociology*, 78, 6, 1973, 1360-1380.

15. Kollock, P. The economies of online cooperation: Gifts and public goods in cyberspace, in M.A. Smith and P. Kollock, ed., *Communities in cyberspace*, London: Routledge, 1999, 220-242.

16. Kollock, P., and Smith, M.A. Communities in cyberspace, in M.A. Smith and P. Kollock, ed., *Communities in cyberspace*, London: Routledge, 1999, 3-28.

17. Latour, B. The powers of association, in John Law, ed., *Power, action and belief. A new sociology of knowledge?*, London: Routledge & Kegan Paul, 1986, 264-280.

18. Latour, B. Technology is society made durable, in J. Law, ed., *A sociology of monsters: Essays on power, technology and domination*, London: Routledge and Kegan Paul, 1991, 103-131.

19. Latour, B. *Aramis, or the love of technology*. Cambridge, MA: Harvard University Press, 1996.

20. Latour, B. *Pandora's hope: Essays on the reality of science studies*. Cambridge, MA: Harvard University Press, 1999.

21. Mauss, M. *The gift: The form and reason for exchange in archaic societies*. New York, NY: W.W. Norton, 1950.

22. Melian, C. *Progressive Open Source: The construction of a development project at Hewlett-Packard*, Dissertation, Stockholm School of Economics, 2007.

23. Nelson, M.L.; Sen, R.; and Subramaniam, C. Understanding open source software: A research classification framework. *Communications of the Association for Information Systems*, 2006, 17, 2006, 2-37.

24. O'Mahony, S., and Ferraro, F. The emergence of governance in an open source community. *Academy of Management Journal*, 50, 5, 2007, 1079-1106.

25. Pettigrew, A.M. Longitudinal field research on change: Theory and practice. *Organization Science*, 1, 3, 1990, 267-292.

26. Raymond, E.S. The cathedral and the bazaar. *FirstMonday*, 3, 3, 1997, Accessed: February 26, 2007, http://www.firstmonday.org/issues/issue3_3/raymond/.

27. Richards, L. *Handling qualitative data: A practical guide*. London: Sage Publications, 2005.

28. Roberts, J.A.; Il-Horn, H.; and Slaughter, S.A. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. *Management Science*, 52, 7, 2006, 984-999.

29. Shah, S.K. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, 52, 7, 2006, 1000-1014.

30. Smith, M.A. Invisible crowds in cyberspace: Mapping the social structure of the Usenet, in M.A. Smith and P. Kollock, ed., *Communities in cyberspace*, London: Routledge, 1999, 195-219.

31. Steiner, G. *After Babel: Aspects of language and translation*. Third ed., Oxford: Oxford University Press, 1998.

32. Suchman, L.A. *Plans and situated actions: The problem of human-machine communication*. Cambridge: Cambridge University Press, 1987.

33. Tuomi, I. Internet, innovation, and open source: Actors in the network. *FirstMonday*, 6, 1, 2001,   Accessed: December   8,   2007, http://www.firstmonday.org/ISSUES/issue6_1/tuomi/index.html.

34. von Hippel, E., and von Krogh, G. Open source software and the 'private-collective' innovation model: Issues for organization science. *Organization Science*, 14, 2, 2003, 209-223.

35. von Krogh, G., and von Hippel, E. The promise of research on open source software. *Management Science*, 52, 7, 2006, 975-983.

36. Weber, S. *The success of open source*. Cambridge, MA: Harvard University Press, 2004.