Handbook of Research on Open Source Software:
Technological, Economic, and Social Perspectives

Kirk St.Amant
Texas Tech University, USA

Brian Still
Texas Tech University, USA

INFORMATION SCIENCE REFERENCE

Hershey · New York

Acquisitions Editor: Kristin Klinger
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Assistant Managing Editor: Diane Huskinson
Copy Editor: Nicole Dean, Shanelle Ramelb, Ann Shaver, and Sue VanderHook
Typesetter: Diane Huskinson
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.info-sci-ref.com

and in the United Kingdom by
Information Science Reference (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Handbook of research on open source software : technological, economic and social perspectives / Kirk St.Amant and Brian Still, editors.
p. cm.
Summary: "This book examines how use of open source software (OSS) is affecting society, business, government, education, and law,including an overview of the culture from which OSS emerged and the process though which OSS is created and modified. Readers will gain an understanding of the complexities and the nuances related to OSS adoption and the range of its applications"--Provided by publisher.
Includes bibliographical references and index.
ISBN 978-1-59140-999-1 (hardcover) -- ISBN 978-1-59140-892-5 (ebook)
1. Shareware (Computer software)--Handbooks, manuals, etc. 2. Open source software--Handbooks, manuals, etc. I. St. Amant, Kirk, 1970- II. Still, Brian, 1968-
QA76.76.S46H35 2007
005.3--dc22
2006039844

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter XVIII
Curious Exceptions?
Open Source Software and "Open" Technology

Alessandro Nuvolari
Eindhoven University of Technology, The Netherlands

Francesco Rullani
Sant'Anna School of Advanced Studies, Italy

ABSTRACT

The aim of this chapter is to explore the differences and commonalities between open source software and other cases of open technology. The concept of open technology is used here to indicate various models of innovation based on the participation of a wide range of different actors who freely share the innovations they have produced. The chapter begins with a review of the problems connected to the production of public goods and explains why open source software seems to be a "curious exception" for traditional economic reasoning. Then it describes the successful operation of similar models of innovation (open technology) in other technological fields. The third section investigates the literature in relation to three fundamental issues in the current open source research agenda, namely, developers' motivations, performance, and sustainability of the model. Finally, the fourth section provides a final comparison between open source software and the other cases of open technology.

INTRODUCTION

Over the last 10 years, open source software development has increasingly attracted the attention of scholars in the fields of economics, management, and social sciences in general (for sociological contributions, see Himanen, Torvalds, & Castells, 2001; Weber, 2004; see Maurer & Scotchmer, 2006, for an account of the phenomenon from the economist's perspective). Although the significance of the software industry in modern economic systems can partially explain the increasing number of research contributions in this area, it is clear that the chief reason behind this growing interest is the fact that open source software development seems to represent a form of innovation process that challenges many facets of the current conventional wisdom concerning the generation of innovations in market economies (Lerner & Tirole, 2001).

Traditionally, economists have considered technological knowledge as a public good, that is, a good endowed with two fundamental features: (a) nonrivalry and (b) nonexcludability. Nonrivalry states that when one actor consumes or uses the good, this does not prevent other actors from consuming or using it. Obviously, this does not hold for standard economic goods: If Paul eats the apple, it is clear that Nathan cannot eat the same apple. On the other hand, both Paul and Nathan can breathe the fresh air of the park. Nonexcludability refers to the fact that when technological knowledge is in the public domain, it is no longer possible to prevent other actors from using it. Again, while Paul may force Nathan to pay for the apple, he cannot (legally) prevent Nathan from breathing the fresh air of the park. The traditional economist's viewpoint contends that market economies are characterized by a systematic underprovision of public goods as their production is, due to the two properties described above, not profitable for private firms. In these circumstances, the standard prescription is that governments should intervene, using tax revenues to supply directly the appropriate quantity of public goods. This reasoning is at the heart of the argument that is commonly used in making the case for the public support of scientific research (Nelson, 1959). It is worth noting that, historically, the allocation of public resources for the production of scientific knowledge has been organized around a rather particular institutional arrangement ("open science") capable of producing both incentives to create new knowledge and the public disclosure of scientific finding (Dasgupta & David, 1994).

Public funding, however, is not the only answer. Another solution put forward by the literature is based on the idea of inducing private firms to invest in the production of technological knowledge by means of an *artificial* system of property rights (Arrow, 1962). The most common example, in this respect, is the patent system. A patent assigns temporarily to its inventor the complete control of the new technological knowledge discovered. The rationale for this institutional device is straightforward: The prospect of the commercial exploitation of this temporary monopoly right will induce private firms to invest resources in inventive activities, that is, in the production of new technological knowledge.

In this context, open source software represents a case of the production of new technological knowledge (high-quality computer programs) carried out by individuals without any direct attempt of "appropriating" the related economic returns. Clearly, all this is at odds with the conventional wisdom summarized above.

Recent research has, however, shown that the innovation process characterizing open source software is not an isolated case. Instead, at least since the industrial revolution, similar types of innovation processes have been adopted in other industries in different periods. Following Foray (2004), we will refer to these episodes as cases of "open technology" in order to stress their similarity with open source software. It is worth warning the reader that in the literature, a variety of other terms and definitions such as "collective invention" or "community based innovation" are frequently used.[1] There is a growing awareness that these cases do not represent just "curious exceptions" to the traditional models of innovation based on public funding or on commercial exploitation by means of exclusive property rights. The aim of this chapter is to provide a compact overview of this literature and to compare these cases of open technology with open source software. Our belief is that this broader perspective can enrich our understanding of open source software.

BACKGROUND

Open Technology: A Neglected Model of Innovation

In a seminal paper, Robert C. Allen (1983) presented a detailed case study of technical change in

the iron industry of Cleveland (United Kingdom) during the period of 1850 to 1870. According to Allen, the Cleveland iron industry was characterized by a particular model of innovation, which he labeled collective invention. In the Cleveland district, iron producers freely disclosed to their competitors technical information concerning the construction details and performance of the blast furnaces they had installed. Information was normally shared both through formal (presentations at meetings of engineering societies, publication of design details in engineering journals, etc.) and informal channels (visits to plants, conversations, etc.). Additionally, new technical knowledge was not protected using patents so that competing firms could freely make use of the released information when they had to construct a new blast furnace. The consequence of this process of information sharing was that the blast furnaces of the district increased their performance very rapidly. Allen noted three essential conditions at the basis of the emergence of the collective-invention regime. The first condition refers to the nature of the technology. In the period considered, there was no consolidated understanding of the working of a blast furnace. The best engineers could do when designing a new blast furnace was to come up with some design guidelines on the basis of previous experiences. Obviously, the sharing of information related to the performance of a large number of furnaces allowed engineers to rely on a wider pool of information in their extrapolations, leading to a more rapid rate of technological progress. Second, blast furnaces were designed by independent consulting engineers who were normally employed on a one-off basis. In this context, the most talented engineers had a strong incentive to disseminate the successful design novelties they had introduced in order to enhance their professional reputation and improve their career prospects. Third, iron producers were often also owners of iron mines. As a consequence, improvements in the efficiency of blast furnaces would have led to an enhancement in the value of the iron deposits of the region. Thus, there was a keen interest in the improvement of the average performance of blast furnaces, as only improvements in the average performance would have influenced the value of iron deposits.

Following Allen's work, other scholars have pointed out the existence of a collective-invention regime in other industries. In a recent study, Nuvolari (2004) has shown that the three conditions of Allen's model of collective invention were also at work in the Cornish community of steam engineers during the first half of the 19th century. This case is particularly interesting because some evidence suggests that the emergence of the collective-invention regime was triggered by a widespread dissatisfaction toward the traditional model of innovation based on patents (in particular, James Watt's patent of 1769).

Other cases of collective invention have been noted in the historical literature, for example, the Western steamboat (Hunter, 1949) and the Lyon silk industry in the 19th century (Foray & Hilaire-Perez, 2000). Scholars have also noted similar knowledge-sharing episodes in several contemporary high-technology districts (most prominently in Silicon Valley; see Saxenian, 1994). However, it is worth noting that in these cases, the very dense knowledge flows between firms may be due to user-producer interactions (Lundvall, 1988) or episodes of know-how trade between engineers (von Hippel, 1987) rather than to the existence of a collective-invention regime in Allen's sense.[2]

A related stream of literature has highlighted the growing importance of user communities as sources of innovation (Franke & Shah, 2003; Shah, 2005). The starting point of these investigations is the observation that in many fields, a sizable share of inventions is due to the users of a specific product and not to its manufacturers (von Hippel, 1988, 2005). One interesting feature of the innovation processes centered on users is that they are often based on very intense knowledge exchanges in the context of specific communities.

Again, within these communities, inventions are normally released in the public domain, and there are no attempts of exploiting them by means of exclusive property rights. Research in this field (see Franke & Shah, 2003, for a detailed study of four user communities in sport equipment) has noted a variety of motivations for the emergence of this type of behavior. First, users belonging to these communities have a keen interest in the performance level of the product. Hence, as in the case of collective invention, the community seems to be characterized by a widespread belief that a mutual cooperative attitude toward inventive activities will enhance the rate of innovation. Second, the social structure of these communities seems to favor the emergence of an ethos prescribing reciprocity and mutual aid.

Apart from the field of sports equipment, in which this type of (user) community-based innovation seems to be prominent, research has identified the existence of this particular model in other industries, such as geographic information systems, astronomic instruments, and early computer and automobile users (see Maurer & Scotchmer, 2006; Shah, 2005).

MAIN FOCUS OF THE CHAPTER

Open Source Software: A Synthesis of Recent Research

One of the main issues to be explored in order to understand the existence and the success of open source software can be stated as follows: Why are developers willing to develop open source software if the typical licenses of this regime[3] prevent them to extract any direct monetary gain from the diffusion of their work? In other words, a study of the open source software phenomenon requires an understanding of developers' motivations.

In order to describe the structure of the landscape of developers' motivations, a first useful distinction has been put forward by Lakhani and Wolf (2005). In this chapter, the authors, following the work by Deci and Ryan (1985), Amabile (1996), and Lindenberg (2001), classify the motivations driving developers' participation into two main groups: intrinsic and extrinsic motivations. When the development activity is undertaken because it enhances developers' utility directly, providing a gain in terms of fun or creativity fulfillment, or a feeling of identity and belongingness to a group, the underlying incentives are said to be intrinsic because the actions they trigger have an intrinsic value for the agent. On the contrary, when the production of code is undertaken instrumentally to reach other goals, such as increasing wages, enhancing the agent's reputation on the job market, or fulfilling specific needs the existing software cannot satisfy, the motivations behind the action are defined as extrinsic because the increase in the individual utility is not due to action itself, but to its consequences.

Each one of the two regions of the developers' motivational landscape can be further structured to isolate the different mechanisms at work in each field. The FLOSS (free/libre open source software) surveys developed by Ghosh, Krieger, Glott, and Robles (2002; answered by 2,784 developers) and by David, Waterman, and Arora (2003; answered by 1,588 developers) offer a finer grain point of view on the motivational landscape. In both the surveys, the most popular answers to questions related to developers' incentives span from "I thought we should all be free to modify the software we use" to "As a user of free software, I wanted to give something back to the community," "I saw it as a way to become a better programmer," "to participate in a new form of cooperation," "to improve OS/FS [open source/free software] products of other developers," and "to improve my job opportunities." Thus, a series of different intrinsic and extrinsic motivations emerges.

Lakhani and Wolf (2005; see also Lakhani, Wolf, Bates, & DiBona, 2002), using survey data collected from 684 developers working on

287 open source projects, were able, by means of a cluster analysis exercise, to identify a number of archetypical cases of open source software developers. They find four clusters, each one approximately the same size as the others. For the members of the largest cluster, a personal sense of creativity and fun are crucial determinants of their contribution to the open source movement. Two other elements emerge as important in this group: the learning opportunities the community offers them, and the possibility to enhance their career through the diffusion of the code they produce. The population of the second cluster resembles the user communities described in the previous section: Skilled developers with specific needs the existing software cannot fulfill are pushed to create the program answering their needs (i.e., lead users). The third cluster is instead composed of paid developers who receive a wage connected to their production of open source products. Eventually, the fourth cluster gathers together individuals strongly committed to the community, moved by the willingness to reciprocate the received help and code, and having a strong ideological position in favor of the open source movement (e.g., believing that code should be open and participating in order to beat proprietary software).

From the empirical studies just described, some subsets of the two main motivation sets emerge. On the one hand, intrinsic motivation can have a psychological nature when it takes the form of fun or creativity fulfillment (Lakhani & Wolf, 2005; Torvalds & Diamond, 2001), or a social nature when it is a product of the interaction between community members and between them and the whole social body of the community, that is, its culture, its shared rules, its ideology, its debate, and so on. In such a thick social environment, developers are willing to participate because they identify with the community, they belong to the hacker culture and feel the need to follow its rules, they believe in the common enterprise they are undertaking, or simply because they care about their status or reputation in the community and are sensitive to peers' regard (Bagozzi & Dholakia, 2006; Dalle & David, 2005; Dalle, David, Ghosh, & Wolak, 2004; Hertel, Niedner, & Hermann, 2003; Himanen et al., 2001; Raymond, 1998; Weber, 2000, 2004; Zeitlyn, 2003). On the other hand, extrinsic motivations can be diversified into subcategories such as career concerns (Lerner & Tirole, 2002), when developers' production of code and diffusion is determined by the willingness to be recognized in the job market as valuable programmers; own use, when the open source community is conceived as a user community à la von Hippel (Hertel et al., 2003; Jeppesen & Frederiksen, 2006; Lakhani & von Hippel, 2003; von Hippel, 2001); and paid contributions (Roberts, Hann, & Slaughter, 2006), when developers are employees of firms active in the open source software environment.

A further element emerged from the cluster analysis by Lakhani and Wolf (2005): learning (von Hippel & von Krogh, 2003). Developers are often driven by the desire to improve their skills and perceive the community as a social environment where they can get help in solving problems by studying collectively new solutions and finding new challenges. Learning can be considered both an intrinsic and an extrinsic incentive, and it cannot be placed easily in one of the subsets defined above. It certainly has an individual and psychological nature, but since it develops alongside the agents' interaction, its nature is much broader. Once the open source community is conceived as a "community of practice" or an "epistemic community" (Cohendet, Creplet, & Dupouët, 2001; Lin, 2004), where the body of knowledge of the whole community interacts and coevolves with each individual's knowledge, learning can be clearly identified as a social process. The same blurred result can be found when conceiving learning as an extrinsic incentive: It can be an instrument for most of the

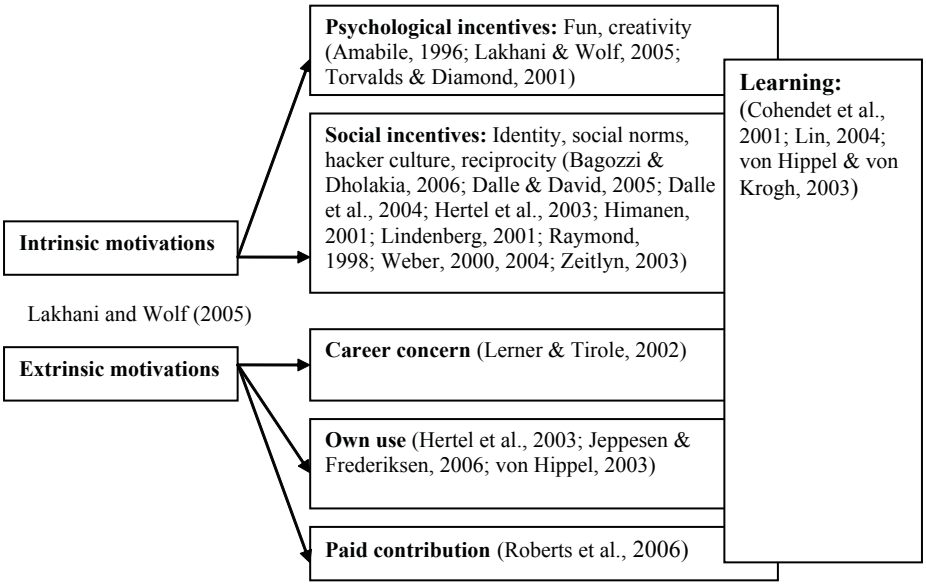
goals typical of the extrinsic motivations described above. Thus, it should be considered as a third group, intersecting all the previous sets. Figure 1 shows the structure of the motivations set as drawn from the quoted literature.

The description of the community proposed above is mainly focused on developers as individuals. However, other subjects are active in the open source environment: firms. Even in an open environment as open source, it is possible for firms to generate profits. The most famous example is given by firms assembling and distribution a ready-to-install version of the GNU/Linux operating system, like Red Hat or Novell. The literature has highlighted several ways by which firms can create value from their participation in the open source movement, but has also shown the instrumental use of their adherence to the community norms and ideology (Rossi & Bonaccorsi, 2005). In other words, as long as incentives are concerned, firms have a much narrower set of incentives, being motivated, as expected, by profit maximization. However, even if the participation of the firms in the open source community is only instrumental, they play an increasingly important role in the open source scene. As we will see in the following sections, they can be fundamental sources of code or related services the community is not willing to produce.

So far, we have given a brief account of the motivations sustaining developers' production of open source software. However, even if developers can decide to dedicate a high amount of effort and time to the production of code, this does not mean that open source represents a successful model of innovation. Thus, our next step is to focus on the performance of open source software as an innovation model.

The first thing to be noticed is that the distribution of open source projects in terms of the main performance indicators—the number of developers and forum or mailing-list discussions (Krishnamurthy, 2002), downloads (Healy & Schussman, 2003), and CVS (Concurrent Version System) commits and file releases (Giuri, Ploner, Rullani, & Torrisi, 2005)—is extremely skewed. Most of the projects remain small individual enterprises without a serious impact on the

Figure 1. Structure of developers' motivational landscape



landscape of the software industry. However, as argued by David and Rullani (2006), open source software should be regarded as a "dissipative" system, burning more resources than those used to produce the actual results. This characteristic is typical of self-organized social structures, where individuals choose on a voluntary basis how much effort to devote to what task and when. In order for the whole system to produce an outcome, several combinations of the available resources have to be worked out before the valuable ones can be selected by the environment.

Thus, on the one hand, the disproportion between the inactive projects and the successful ones characterizes the open source model as dissipative rather than an unsuccessful model of innovation. On the other hand, the same argument calls for a definition of the drivers of open source projects performance in order to be able to reduce the gap between the mobilized resources and those that are actually used.

A first result along this line of inquiry states that projects adopting a restrictive license like the GPL (General Public License) tend to have lower performance (Comino, Manenti, & Parisi, 2005; Fershtman & Gandal, 2004; see also Lerner & Tirole, 2005). This result could be due to a detrimental impact of the excessive openness of the GPL projects, which may be unable, for example, to attract firms and sponsored developers. A hybrid model of innovation, where the adopted license scheme is able to create a synergy between the community and other economic actors, should be then considered as a valuable configuration (Bonaccorsi, Giannangeli, & Rossi, 2006). A second result is that the division of labor has a significant positive impact on project performance. However, the variety and the level of members' skill sets (Giuri et al., 2005) and the costs connected to the coordination of many developers (Comino et al., 2005) have to be taken into account in order to avoid a net negative effect. Modularity at the level of the code structure has been analyzed by Baldwin and Clark (2006), who find that a modular architecture is able to attract more voluntary effort and reduce free riding. Applying an ecological perspective, Grewal, Lilien, and Mallapragada (2006) look at the developers' membership in different projects to draw a network of relationships between projects and developers. They show that projects with a central position in the network are more likely to exhibit high technical performance, but the network is not so crucial in determining the commercial success (i.e., number of downloads) of the produced software.

Having established what moves developers and what the drivers of the open source software innovative performance are, a last question regards the possibility to sustain such a structure over time. The contributions moving in this direction are scarce, and there is need for further research. A first contribution has been given by Gambardella and Hall (in press). The authors show that a coordination device is needed to assure the stability of collaboration. The adoption of the GPL can be thought of as such a mechanism, preventing any developer joining the project after the founder to adopt opportunistic behavior. This argument points out an interesting trade-off between performance and sustainability: Less restrictive licenses can induce higher performance, but can undermine the sustainability of the community. A second point has been made by David and Rullani (2006), showing that developers undertaking their activity on the SourceForge.net (http://sourceforge.net/) platform during the period of 2001 to 2002 exhibit a robust, nontransient tendency to participate in existing projects and also to create new projects. Sustainability, then, can be conceived at least as a valuable working hypothesis.

Open Technology and Open Source Software: A Comparison

Various researchers have noted the existence of important parallels between the model of open technology discussed previously and the findings emerging from ongoing studies of the open

source communities (see, among others, Foray, 2004; Nuvolari, 2005). In a nutshell, these are the main points of interest:

a. Both collective-invention regimes and the open source movement seem to emerge thanks to a perceived dissatisfaction toward the innovative performance delivered by traditional regimes based on exclusive property rights.
b. Case studies of collective invention and user communities seem generally characterized by remarkable performances in terms of rates of innovation. The same remarkable innovative performance has characterized some open source software projects, at least since the 1990s, when GNU/Linux was born.
c. However, only a restricted number of open source software projects are successful. Similarly, only few innovations coming from the users are really valuable, as well as only few contributions added to the common pool of collective inventions are really improving the performance of the sector. Thus, the models share the dissipative property described for the open source model of innovation.
d. Both collective invention and a number of open source software projects are characterized by high levels of complexity and uncertainty. In these conditions, a model of innovation based on knowledge sharing, cooperation, and continuous feedback permits the timely identification of the most promising lines of development.
e. Cases of collective invention, user-based innovation models, and open source software are forms of innovation processes involving heterogeneous sets of actors (in particular, engineers, lead users, and developers with different skills and talents, and firms) organized into communities.
f. Collective invention, open source software, and user communities rely on a complex set of motivational drivers, spanning from economic incentives, dissatisfaction toward tight intellectual property-rights regimes, psychological and social motives, and so on. Even if the open source software and the other examples of open innovation seem to rely on different compositions of the aforementioned motivational factors, it might well be that this plurality of motives represents one of the fundamental ingredients for sustaining both open source and open-technology regimes.

CONCLUSION

The Core of the Difference and the Challenges to Sustainability

The main difference between the three regimes of innovation can be found in the relationship between the communities of innovative agents and the involved firms. In a collective-invention regime, firms strategically suspend appropriation of the produced knowledge in order to deal with technological problems that an individual firm could not handle. In this sense, firms are the fundamental actors of collective-invention regimes. Accordingly, these regimes usually disappear when the collective effort to overcome the radical uncertainty in the technological space is not necessary anymore (i.e., when a specific technological trajectory or paradigm emerges; Dosi, 1982), and each firm is willing to return to the proprietary regime that will assure higher individual profits. On the contrary, the nexus between manufacturers and users is much tighter in user communities. Users innovate around the products of the firms, which in turn try to sustain users' involvement. Sometimes, these communities are originated directly by the firms, and other times they emerge

spontaneously through users' interaction. In the open source software case, the leading role is instead played by users and developers, and firms are mainly active in those spaces that the community does not or cannot reach. Firms have to adapt to the rules of the community and do not directly control on the product (Dahlander & Magnusson, 2005; Shah, 2006). Thus, the basic difference between the three models of innovation is in the balance between the roles of firms and of users and developers.

These considerations shed new light on the relative sustainability of these regimes. Collective inventions can exist only as long as firms do not profit enough from a traditional proprietary regime; this happens mostly in conditions of radical technological uncertainty (emerging phases of a novel technological paradigm). Instead, user communities and open source software seem to be characterized by different sustainability conditions (Osterloh & Rota, 2005). The sustainability of the former depends directly on the ability of communities and firms to involve individual users and keep their participation at a certain level; in the case of the latter, several factors, still to be fully identified, can induce a decay of the phenomenon or strengthen its sustainability. The foregoing discussion on the trade-offs between open source sustainability, performance, and level of openness (as defined by the license) clearly bears out this point.

REFERENCES

Allen, R. C. (1983). Collective invention. *Journal of Economic Behaviour and Organization*, *4*, 1-24.

Amabile, T. M. (1996). *Creativity in context.* Boulder, CO: Westview Press.

Arrow, K. (1962). Economic welfare and the allocation of resources for invention. In R. Nelson (Ed.), *The rate and direction of inventive activity: Economic and social factors.* Princeton, NJ: Princeton University Press.

Bagozzi, R. P., & Dholakia, U. M. (2006). Open source software user communities: A study of participation in Linux user groups. *Management Science*, *52*(7), 1099-1115.

Baldwin, C. Y., & Clark, K. B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science*, *52*(7), 1116-1127.

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

Chesbrough, H., Vanhaverbake, W., & West, J. (Eds.). (2006). *Open innovation: Researching a new paradigm*. Oxford, UK: Oxford University Press.

Cohendet, P., Creplet, F., & Dupouët, O. (2001, June). *Communities of practice and epistemic communities: A renewed approach of organizational learning within the firm*. Paper presented at the Workshop on Economics and Heterogeneous Interacting Agents, Marseille, France.

Comino, S., Manenti, F. M., & Parisi, M. L. (2005). *From planning to mature: On the determinants of open source take off* (Discussion Paper 2005-17). Trento, Italy: Trento University.

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

Dalle, J.-M., & David, P. A. (2005). The allocation of software development resources in "open source" production mode. In J. Feller et al. (Eds.), *Making sense of the bazaar: Perspectives on*

open source and free software. Cambridge, MA: MIT Press.

Dalle, J.-M., David, P. A., Ghosh, R. A., & Wolak, F. A. (2004, June). *Free & open source software developers and "the economy of regard": Participation and code-signing in the modules of the Linux kernel.* Paper presented at the Oxford Workshop on Libre Source, Oxford, UK.

Dasgupta, P., & David, P. A. (1994). Towards a new economics of science. *Research Policy*, *23*, 487-521.

David, P. A., & Rullani, F. (2006, June). *Open source software development dynamics: Project joining and new project generation on SourceForge*. Paper presented at the Meetings of the International Schumpeter Society (ISS), Sophia-Antipolis, France.

David, P. A., Waterman, A., & Arora, S. (2003). *The free/libre/open source software survey for 2003* (Preliminary draft). Unpublished manuscript.

Deci, E. L, & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior.* New York: Plenum Press.

Dosi, G. (1982). Technological paradigms and technological trajectories: A suggested interpretation of the determinants and directions of technical change. *Research Policy*, *11*(3), 147-162.

Fershtman, C., & Gandal, N. (2004). *The determinants of output per contributor in open source projects: An empirical examination* (Discussion Paper 4329). London: CEPR.

Foray, D. (2004). *The economics of knowledge.* Cambridge, MA: MIT Press.

Foray, D., & Hilaire-Perez, L. (2000, May). *The economics of open technology: Collective invention and individual claims in the "fabrique lyonnaise" during the old regime.* Paper presented at the conference in honour of Paul A. David, Turin, Italy.

Franke, N., & Shah, S. (2003). How communities support inventive activities: An exploration of assistance and sharing among end-users. *Research Policy*, *32*, 157-178.

Gambardella, A., & Hall, B. (in press). Proprietary vs public domain licensing of software and research products. *Research Policy.*

Ghosh, R. A. (1998). Cooking pot markets: An economic model for the trade in free goods and services on the Internet. *First Monday*, *3*(3). Retrieved March 22, 2007, from http://www.firstmonday.org/issues/issue3_3/ghosh

Ghosh, R. A., Krieger, B., Glott, R., & Robles, G. (2002). *Free/libre and open source software. Part IV: Survey of developers.* International Institute of Infonomics, Berlecom Research GmbH.

Giuri, P., Ploner, M., Rullani, F., & Torrisi, S. (2005, September). *Skills, division of labor and performance in collective inventions: Evidence from the open source software.* Paper presented at the EARIE Conference, Porto, Portugal.

Grewal, R., Lilien, G. L., & Mallapragada, G. (2006). Location, location, location: How network embeddedness affects project success in open source systems. *Management Science*, *52*(7), 1043-1056.

Healy, K., & Schussman, A. (2003). *The ecology of open source software development* (Working Paper). AZ: Department of Sociology, University of Arizona.

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

Himanen, P., Torvalds, L., & Castells, M. (2001). *The hacker ethic and the spirit of the information age.* London: Secker & Warburg.

Hunter, L. (1949). *Steamboats on the Western rivers: An economic and technological history.* Cambridge, MA: Harvard University Press.

Jeppesen, L. B., & Frederiksen, L. (2006). Why firm-established user communities work for innovation? The personal attributes of innovative users in the case of computer-controlled music instruments. *Organization Science*, *17*(1), 45-64.

Krishnamurthy, S. (2002). Cave or community? An empirical examination of 100 mature open source projects. *First Monday*, *7*(6). Retrieved March 22, 2007, from http://www.firstmonday.dk/issues/issue7_6/krishnamurthy

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: "Free" developer-to-developer assistance. *Research Policy*, *32*.

Lakhani, K. R., & Wolf. (2005). Why hackers do what they do: Understanding motivations and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software.* Cambridge, MA: MIT Press.

Lakhani, K. R., Wolf, R. G., Bates, J., & DiBona, C. (2002). *The Boston Consulting Group hacker survey* (Release 0.73). Author.

Lerner, J., & Tirole, J. (2001). The open source movement: Key-research question. *European Economic Review*, *45*, 819-826.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The Journal of Industrial Economics*, *L*(2), 197-234.

Lerner, J., & Tirole, J. (2005). The scope of open source licensing. *Journal of Law, Economics, and Organization*, *21*, 20-56.

Lin, Y. (2004). Contextualising knowledge-making in Linux user groups. *First Monday*, *9*(11). Retrieved March 22, 2007, from http://www.firstmonday.org/issues/issue9_11/lin/index.html

Lindenberg, S. (2001). Intrinsic motivation in a new light. *Kyklos*, *54*(2/3), 317-342.

Lundvall, B.-Å. (1988). Innovation as an interactive process: From user-producer interaction to the national system of innovation. In G. Dosi, C. Freeman, R. Nelson, G. Silverberg, & L. Soete (Eds.), *Technical change and economic theory.* London: Pinter.

Maurer, S. M., & Scotchmer, S. (2006). *Open source software: The new intellectual property paradigm* (NBER Working Paper 12148).

Nelson, R. (1959). The simple economics of basic scientific research. *Journal of Political Economy*, *67*, 297-306.

Nuvolari, A. (2004). Collective invention during the British industrial revolution: The case of the Cornish pumping engine. *Cambridge Journal of Economics*, *28*, 347-363.

Nuvolari, A. (2005). Open source software development: Some historical perspectives. *First Monday*, *10*(10). Retrieved March 22, 2007, from http://www.firstmonday.org/issues/issue10_10/nuvolari/index.html

Osterloh, M., & Rota, S. G. (2005). *Open source software development—Just another case of collective invention?* (CREMA Working Paper 2005-08).

Raymond, E. (1998). Homesteading the Noosphere. *First Monday*, *3*(10). Retrieved March 22, 2007, from http://www.firstmonday.org/issues/issue3_10/raymond/index.html

Roberts, J. A., Hann, I., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science*, *52*(7), 984-999.

Rossi, C., & Bonaccorsi, A. (2005). Intrinsic vs. extrinsic incentives in profit-oriented firms supplying open source products and services. *First Monday*, *10*(5). Retrieved March 22, 2007, from http://www.firstmonday.org/issues/issue10_5/rossi

Saxenian, A. (1994). *Regional advantage: Culture and competition in Silicon Valley and in Route 128*. Cambridge, MA: Harvard University Press.

Shah, S. (2005). Open beyond software. In C. DiBona, D. Cooper, & M. Stone (Eds.), *Open sources 2.0: The continuing evolution*. Sebastopol, CA: O'Reilly.

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

Torvalds, L., & Diamond, D. (2001). *Just for fun: The story of an accidental revolutionary.* New York: Texere.

Von Hippel, E. (1987). Cooperation between rivals: Informal "know how" trading. *Research Policy*, *16*, 291-302.

Von Hippel, E. (1988). *The sources of innovation*. Oxford, UK: Oxford University Press.

Von Hippel, E. (2001). Innovation by user communities: Learning from open source software. *MIT Sloan Management Review*, *42*(4), 82-86.

Von Hippel, E. (2005). *Democratizing innovation.* Cambridge, MA: MIT Press.

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

Weber, S. (2000). *The political economy of open source software* (BRIE Working Paper 140).

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

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

KEY TERMS

Collective Invention: An innovation model in which private firms engaged in the production or use of a specific good freely share one another's inventions and other pertinent technical information.

Dissipation: We call dissipation an innovation model that mobilizes (or "burns") more resources than those actually used to produce the outcome. Dissipation is typical of self-organizing and explorative organizations.

Intrinsic/Extrinsic Motivations: When an activity is undertaken because it enhances agents' utility directly, the underlying incentives are intrinsic because the actions they trigger have an intrinsic value for the agent. On the contrary, when an action is undertaken instrumentally to reach other goals, the motivations behind the action are defined as extrinsic because the increase of the individual's utility is not due to action itself, but to its consequences.

Sustainability: We call sustainable an innovation model that re-creates over time the premises for its own reproduction, that is, if it is endowed with a mechanism able to re-create incentives for the participants to continually invest in innovation. In this sense, the patent system as well as the public-funded research system can be conceived as sustainable.

User Community: An innovation model where a community of users of a particular product are the main source of innovation and where innovations are normally freely shared within the community.

ENDNOTES

1 Another term that is becoming increasingly popular in the management literature is "open innovation" (see Chesbrough, Vanhaverbeke, & West, 2006). The concept of open innovation refers to the fact that firms are increasingly making use of external sources of knowledge in their innovation processes. Clearly, this is somewhat related to the phenomenon of open technology sketched above as firms, in order to gain access to these external sources, are frequently required to adopt a more relaxed attitude toward the appropriation of their inventions. In this chapter, we will not deal with the literature on open innovation.

2 In know-how trading, information is typically exchanged by engineers belonging to competing firms on a bilateral basis. In collective-invention regimes, all the competing firms have free access to the potentially proprietary know-how.

3 The possibility for subsequent developers to change the open regime established by the initial choice of an open source license depends on the terms of each specific license. We refer the reader to Lerner and Tirole (2005) for a futher discussion on the different types of licenses.