
O

Open Source

Fabrizio Ferraro and Jordi Torrents
IESE Business School, Madrid, Spain

Abstract

The term open source (OS) comprises all the practices (technological, organizational and legal) that ensure to anyone the rights to execute, copy, modify and distribute a piece of computer software, thanks to the availability of its source code. These practices have had profound effects on the ► [software industry](#) and have been a fertile ground for theoretical and empirical work on the part of strategic management scholars. We review some of the insights of strategic management research on OS, focusing on three main streams of studies: (1) OS as ► [innovation](#), (2) OS communities' governance and organization, and (3) OS competitive dynamics in the software industry.

Definition The term open source (OS) software refers to the practices (technological, organizational and legal) that ensure to anyone the rights to execute, copy, modify and distribute a piece of computer software, thanks to the availability of its source code.

This entry was originally published on Palgrave Connect under ISBN 978-1-137-49190-9. The content has not been changed.

© The Author(s) 2016
M. Augier, D.J. Teece (eds.), *The Palgrave Encyclopedia of Strategic Management*,
DOI 10.1057/978-1-349-94848-2_360-1

Open Source in Strategic Management

The first explicit formulation of users' freedom to execute, copy, modify and distribute software is due to Richard Stallman in early 1980s (Stallman 1998). But the concept of open source (OS) diffused widely with the fast growth of the GNU/Linux operating system and the spread of Internet access, especially over the 2000–10 decade. By 2011, for instance, 62 % of servers providing web access were running the open source web server Apache (Netcraft 2011) and 82 % of top 500 supercomputers in the world were running GNU/Linux (Top500.org 2011).

The roots of the concept run deep in the history of computing and are tightly tied to the dynamics of scientific communities. Before the 1980s, software was openly shared among users, and access to the source code was the norm (Levy 2001; Weber 2004). By 1990s, the ► [software industry](#), to capture more value, had almost completely eradicated this practice and most software did not grant access to the source code, thus severely limiting its distribution and modification. In this context, OS re-emerged as an alternative model of software production and distribution. In the OS model, software is conceptualized as a public good: its use is non-rival, and its licence ensures that private intellectual property rights will not hinder developers' and users' ability to use, modify and redistribute it (O'Mahony 2003).

Strategic management scholars found the OS phenomenon fascinating for a number of reasons. First and foremost, from an ► **innovation** standpoint, the phenomena seemed to run counter to some well-established precept in the field on the role of strong intellectual property (IP) protection to foster innovation. Second, organizational researchers marvelled at the ability to attract highly technical volunteer work, coordinate their effort, and create sustainable communities without resorting to the corporate form. Finally, from a competitive dynamics perspective, OS represented an alternative business model within the software industry, and has been a fertile ground to explore the relationship between different models. We will address these areas in turn.

Innovation Strategy and OS

The success of OS has showed the strengths of its development process in terms of the quality of the products obtained and the time frames and resources needed to produce them (Lerner and Tirole 2002, 2005; Wheeler 2004). The fact that big firms – such as IBM, Intel and Google – have been involved in supporting and developing OS, which increasingly is a key part of their infrastructure or their sales strategy (Bradski et al. 2005; Capek et al. 2005; Samuelson 2006; Bryant 2007; Gawer and Henderson 2007; Alpern et al. 2010), has been a puzzle for management scholars. This contradicts the classic assumption that a weak regime of appropriability undermines the incentives of firms to invest in R&D in order to improve their products and to release new ones that contribute to innovation in the field. Thus, many researchers have wondered how OS projects are able to innovate, in order to solve complex technical problems, and freely reveal those innovations without appropriating private returns from selling the software.

One of the first approaches to this problem builds on the analytical distinction between two models of innovation. On the one hand, the ‘private investment’ model assumes returns to the innovator from private goods that are strongly protected with efficient regimes of intellectual

property. On the other hand, the ‘collective action’ model assumes that, under conditions of market failure, innovators collaborate in order to produce a public good (Von Hippel and Von Krogh 2003). OS development is an instance of a compound model of innovation that contains elements of both models and can offer society the ‘best of both worlds’ under many conditions. Thus, these two analytical models are not discrete states but two ends of a continuum.

But, how do firms position themselves in this continuum? Building on Teece (1986, 1992), Fosfuri et al. (2008) argue that firms can profit from their investment in OS by relying on the control of complementary resources, and that the heterogeneity in the distribution of such resources explains why some firms take more OS commercial actions than others: OS implies a reduction of the level of protection of the core product, which forces firms to secure the control of complementary resources to benefit from their commercialization efforts. They show that variations in firms’ endowments of intellectual property rights are associated with the likelihood of releasing OS software packages. More specifically, they studied the release of software packages under OS licences between 1980 and 2003 and found that firms were more likely to release OS products if they had (a) a larger stock of pre-existing software patents, and (b) a smaller stock of preexisting software trademarks. While patent protection of complementary assets might help firms capture some of the value created with innovative OS software releases, firms still need to cooperate with the emerging communities of OS developers in ways that fit their business models. Dahlander and Wallin (2006) showed that software firms sponsor individual developers in OS software communities in order to influence their development.

Governance and Organization of OS Communities

Beyond the issue of collaboration with corporations, OS communities provided researchers with a number of interesting organizational puzzles. At

the individual level, research on OS has focused on why talented software developers decide to work, in many cases, for free in OS projects (Lakhani and Wolf 2005). The literature has mainly stressed three compatible responses to the motivational puzzle. First, from the point of view of developers participating in OS projects, ‘scratching an itch’ (Raymond 2001: 23) is a commonplace response to this puzzle. To work in something that is considered interesting and useful is a powerful motivation. Second, because the work in OS projects is public and visible, developers can build a reputation based on their achievements. Reputation is an important currency in a highly technical field where the work of developers is hidden in the black box of closed source. Thus, reputation and professional development are also powerful motivations (Lerner and Tirole 2002; Lakhani and Wolf 2005; Bagozzi and Dholakia 2006; Roberts et al. 2006). And, finally, a stream of research has stressed that it is useful to conceptualize OS as a gift economy, where shared ethical and moral standards play a key role in boosting participation and commitment (Coleman 2004). Developers’ participation is only one of the many organizational and governance challenges of OS developers’ communities. The traditional scalability problems of software projects (Brooks 1995) should only be more critical when coupled with purely voluntary participation. Defying these gloomy predictions, OS communities kept growing, with some of them counting on the contribution of thousands of developers every year. Inspired by the software engineering approach, some scholars suggested that a modular structure of OS code is key to the scalability of these communities (Lerner and Tirole 2002; Mockus et al. 2002; MacCormack et al. 2006).

Another important stream of research has focused on the governance structures and leadership dynamics of OS communities. For ► [open innovation](#) communities, Fleming and Waguespack (2007) find that future leaders must first make strong technical contributions. They also must integrate their communities in order to mobilize volunteers and avoid the danger of forking and balkanization. This is enabled by

two correlated but distinct social positions: social brokerage and boundary spanning between technological areas. O’Mahony and Ferraro (2007) suggest that OS communities might need to blend bureaucratic and democratic mechanisms to achieve a functioning governance structure that will allow them to adapt to internal and external changes.

West and O’Mahony (2008) focus on the relationship between corporations and communities. They go beyond the code structure and development approach, and stress the role of IP rights and model of community governance in shaping what they call the participation architecture of different OS communities: the socio-technical framework that facilitates contribution from developers in the community and integrates their contributions in the software offering of the corporation.

Competitive Dynamics

Despite the impact of OS on the software industry in terms of competitive dynamics, very few studies have directly explored how proprietary and open source software interact in the marketplace. Casadesus-Masanell and Ghemawat (2006) model the Linux vs Microsoft Windows competition as a dynamic mixed duopoly in which a profit-maximizing competitor interacts with a competitor that prices at zero. In their game, the main advantage of OS is the ability of users to modify the source code directly, solving problems and improving it (i.e., faster demand-side learning). The main advantage of Microsoft Windows is its large initial installed base. In the model of Casadesus-Masanell and Ghemawat, as long as Windows has the advantage of a larger installed base at time zero, Linux never displaces Windows’ leadership position. Building on this basic result, they investigate the conditions that would allow Linux to succeed in the marketplace, and suggest that commitment by governments and large corporations could help Linux overtake Windows. Furthermore, and more counter-intuitively, they suggest that piracy might actually help Windows maintain its dominance, by artificially inflating its installed base. They confirm this

prediction with data on a cross-section of countries on Linux penetration and piracy rates and find out that in countries where piracy is high, Linux has a low penetration rate.

Future Research

Given the impact OS has had on our computing infrastructure and how it is revolutionizing the computer industry, it is not surprising that it has attracted the attention of academics beyond the confines of strategic management. Computer scientists and IS scholars have explored the technological factors enabling this growth and the software engineering practices that emerged in this space (Koch and Schneider 2002; Mockus et al. 2002; Michlmayr et al. 2007; Scacchi 2007). Legal scholars have focused on the role of different licensing schemes and their implications for intellectual property and culture (Lessig 1999; Benkler 2002; Moglen 2003; Carver 2005). Given the scope of this entry, we have not referred to this work in detail, but we do believe that future research should try to integrate findings from different disciplines. Leveraging what we are learning from OS communities to the broader problem of organizing innovative activity is another key direction for future studies, which should explore what is specific to software and what can be applied to other domains of activity.

See Also

- ▶ Collaborative Innovation
- ▶ Innovation
- ▶ Licensing Strategy
- ▶ Open Innovation
- ▶ Software Industry

References

- Alpern, B., S. Augart, S.M. Blackburn, M. Butrico, A. Cocchi, P. Cheng, J. Dolby, S. Fink, D. Grove, and M. Hind. 2010. The Jikes research virtual machine project: Building an open-source research community. *IBM Systems Journal* 44: 399–417.
- Bagozzi, R.P., and U.M. Dholakia. 2006. Open source software user communities: A study of participation in Linux user groups. *Management Science* 52: 1099–1115.
- Benkler, Y. 2002. Coase's penguin, or, Linux and the nature of the firm. *Yale Law Journal* 112: 369–446.
- Bradski, G., A. Kaehler, and V. Pisarevski. 2005. Learning-based computer vision with intel's open source computer vision library. *Intel Technology Journal* 9: 119–130.
- Brooks Jr., F.P. 1995. *The mythical man-month: Essays on software engineering*. Massachusetts: Addison-Wesley.
- Bryant, R. E. 2007. Data-intensive supercomputing: The case for DISC. Technical report No. CMU-CS-07-128, Computer Science Department, Carnegie Mellon University.
- Capek, P.G., S.P. Frank, S. Gerdt, and D. Shields. 2005. A history of IBM's open-source involvement and strategy. *IBM Systems Journal* 44: 249–257.
- Carver, B.W. 2005. Share and share alike: Understanding and enforcing open source and free software licenses. *Berkeley Technology Law Journal* 20: 443.
- Casadesus-Masanell, R., and P. Ghemawat. 2006. Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Science* 52: 1072–1084.
- Coleman, G. 2004. The political agnosticism of free and open source software and the inadvertent politics of contrast. *Anthropological Quarterly* 77: 507–519.
- Dahlander, L., and M.W. Wallin. 2006. A man on the inside: Unlocking communities as complementary assets. *Research Policy* 35: 1243–1259.
- Fleming, L., and D.M. Waguespack. 2007. Brokerage, boundary spanning, and leadership in open innovation communities. *Organization Science* 18: 165–180.
- Fosfuri, A., M.S. Giarratana, and A. Luzzi. 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science* 19: 292–305.
- Gaver, A., and R. Henderson. 2007. Platform owner entry and innovation in complementary markets: Evidence from Intel. *Journal of Economics and Management Strategy* 16: 1–34.
- Koch, S., and G. Schneider. 2002. Effort, co-operation and coordination in an open source software project: GNOME. *Information Systems Journal* 12: 27–42.
- 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*, ed. Feller Joseph, Fitzgerald Brian, Scott A. Hissam, and Karim R. Lakhani. Cambridge, MA: The MIT Press.
- Lerner, J., and J. Tirole. 2002. Some simple economics of open source. *Journal of Industrial Economics* 50: 197–234.
- Lerner, J., and J. Tirole. 2005. The economics of technology sharing: Open source and beyond. *Journal of Economic Perspectives* 19: 99–120.

- Lessig, L. 1999. *Code and other laws of cyberspace*. New York: Basic Books.
- Levy, S. 2001. *Hackers: Heroes of the computer revolution*. New York: Penguin.
- MacCormack, A., J. Rusnak, and C.Y. Baldwin. 2006. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science* 52: 1015–1030.
- Michlmayr, M., G. Robles, and J.M. Gonzalez-Barahona. 2007. Volunteers in large libre software projects: A quantitative analysis over time. In *Emerging free and open source software practices*, ed. S.K. Sowe, I.G. Stamelos, and I. Samoladas. Hershey: IGI Publishing.
- Mockus, A., R.T. Fielding, and J.D. Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology* 11: 309–346.
- Moglen, E. 2003. The dotCommunist manifesto. Available at http://emoglen.law.columbia.edu/my_pubs/dcm.html. Accessed 23 Nov 2012.
- Netcraft. 2011. May 2011 web server survey. *netcraft.com*. Available at <http://news.netcraft.com/archives/2011/05/02/may-2011-web-server-survey.html>. Accessed 4 May 2011.
- O'Mahony, S. 2003. Guarding the commons: How community managed software projects protect their work. *Research Policy* 32: 1179–1198.
- O'Mahony, S., and F. Ferraro. 2007. The emergence of governance in an open source community. *Academy of Management Journal* 50: 1079–1106.
- Raymond, E.S. 2001. *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol: O'Reilly & Associates.
- Roberts, J.A., I.H. Hann, and S.A. Slaughter. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science* 52: 984.
- Samuelson, P. 2006. IBM's pragmatic embrace of open source. *Communications of the ACM* 49: 21–25.
- Scacchi, W. 2007. Free/open source software development: Recent research results and methods. *Advances in Computers* 69: 243–295.
- Stallman, R. M. 1998. The GNU Project. Available at <http://www.gnu.org/gnu/thegnuproject.html>. Accessed 23 Nov 2012.
- Teece, D.J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15: 285–305.
- Teece, D.J. 1992. Competition, cooperation, and innovation: Organizational arrangements for regimes of rapid technological progress. *Journal of Economic Behavior & Organization* 18: 1–25.
- Top500.org. 2011. Operating system share for 11/2010 | TOP500 supercomputing sites. Available at <http://www.top500.org/stats/list/36/os>. Accessed 4 May 2011.
- Von Hippel, E., and G. Von Krogh. 2003. Open source software and the 'private-collective' innovation model: Issues for organization science. *Organization Science* 14: 209–223.
- Weber, S. 2004. *The success of open source*. Cambridge, MA: Harvard University Press.
- West, J., and S. O'Mahony. 2008. The role of participation architecture in growing sponsored open source communities. *Industry & Innovation* 15: 145–168.
- Wheeler, D. A. 2004. Linux kernel 2.6: It's worth more! Available at <http://www.dwheeler.com/essays/linux-kernel-cost.html>. Accessed 23 Nov 2012.