

The governance of open source initiatives: what does it mean to be community managed?

Siobhán O'Mahony

Published online: 14 June 2007
© Springer Science+Business Media B.V. 2007

Abstract The concept of 'open source' software initially referred to software projects managed by grassroots communities in public forums. Since 1998, the concept has been adapted and diffused to new settings that extend beyond software. While the open source community has maintained control over which software licenses can be considered 'open source', little attention has been paid to the elements that constitute community management. More private parties now contribute to OSS communities and more hybrid governance models have emerged. Before we can understand how hybrid models differ from a community managed model, a more precise definition is needed. This essay takes a step in this direction by identifying five core principles critical to community-managed governance.

Keywords Community management · Governance · Open source software · Software development

Generally, you want to do things by consensus of the community. But if the community can't decide, there has to be someone who's appointed to make the decision one way or another. (Linux Standards Base Volunteer Contributor, interviewed on March 12, 2001)

1 Introduction

Scholars researching open source software (OSS) projects have become increasingly aware of the need to specify that most of the research to date has examined projects

S. O'Mahony (✉)
Graduate School of Management, University of California at Davis, AOB IV One Shields Avenue,
Davis, CA 95616-8609, USA
e-mail: somahony@ucdavis.edu

that are community managed. At an earlier time it was not necessary to specify that an open source project was community managed—for a community-managed development model was considered part of the definition of an open source project. Because of the enhanced elasticity of the open source term, the conception of community management has become decoupled from the conception of an open source project. OSS projects that do not depend upon a community-managed governance model now abound. This may be partly due to the success of the open source frame, but I argue that it calls for renewed attention to explicate how a community-managed mode of governance differs from other forms.

While early observers of OSS projects referred to them as 'self managed' or 'self governed', as the quote on the prior page indicates, neither term accurately describes how open source governance models actually work. Most successful, mature OSS projects producing commercial grade code have well developed approaches to not only software development, but to their overall governance. Often this includes a formal leadership role, a representative body of decision-makers and a non-profit foundation to protect the community's interests (O'Mahony 2003, 2005). However, the core elements that constitute community-managed governance remain under explored.

This essay accomplishes two objectives. First, I trace the ways in which the open source frame has been adapted and diffused to make a compelling case for renewed attention to the principles of community-managed governance. Second, drawing upon prior research, I explore some of the underlying principles that may constitute a community-managed mode of governance. In so doing, the goal is not to assume that the community-managed approach is the only governance model consistent with open source ideals. There are many successful variations of open source governance that do not rely upon community management (Markus, this issue; Shah 2006; O'Mahony and West 2006). Rather, the goal is to articulate the parameters of community management so that scholars are better positioned to discuss its variants with more precision.

2 Diffusion and adaptation of the open source development model

2.1 Frame elasticity

Since the term open source was created in 1998, it has come to mean many things: a type of software license, an approach to software development, a type of community, and a type of business model. More recently, 'open source' approaches to innovation have been ported to areas beyond software to include medical devices, sporting goods, consumer products and even baked goods (Chesbrough 2003; von Hippel 2005; Gladwell 2005; Shah 2005; Chesbrough et al. 2006). When a term that is used to frame a social phenomenon (or 'frame') takes root and is applied to new contexts, its meaning often becomes more 'elastic' (Benford and Snow 2000). Frames guide individuals' interpretations of events to align with those of a collective, be that a community or a social movement (Goffman 1974; Snow et al. 1986). To increase the popularity of their ideas, organizers often try to make their framing more elastic to include greater variance in interpretation and thus mobilize

a larger group of participants (Benford and Snow 2000; Snow and Benford 1992). And when collectives acquire a new reference group or target audience they are more likely to modify their frame (McAdam 1986).

The success of Linux, Raymond's articulation of the open source development model (1999) and Netscape's source code release of the Mozilla browser in 1998 helped attract a commercial audience receptive to open source code and the enhanced efficiency and reliability that an open approach to developing software could provide. This inspired a faction within the free software community to create a new frame to broaden their appeal to commercial interests. As one attendee at the 1998 'reframing' meeting explained to me: "Not all of the people who started this thing, really believed in free software by the (original) definition....I felt like—I wanted to dilute the definition and—very definitely did." (*OSS movement founder, interviewed on July 19, 2000*).

The new frame, 'open source', highlighted the pragmatic benefits associated with source code and a transparent, public software development process. It also signaled the OSS community's stance toward the business world: "This terminological debate is understood by all parties to be a proxy for wider issues about the community's relationship to the business world" (*A non-profit founder, interviewed in 2000*). The 'open source' frame had a dramatic affect on public discourse, expanding the audience for open source software in ways its creators could not have imagined. "The whole transition from majority use of free software as a label to majority use of open source happened in the early summer of 1998 in *six weeks [with hand emphasis] flat*. What that showed me was that there was a huge, pent up demand for a more effective, less confrontational way of pitching the message" (*OSS movement founder, interviewed on March 14, 2001*).

Since that time, the open source frame has been adapted and diffused in many ways. However, as the open source frame acquires more elasticity, the governance model that was once implicitly assumed to be associated with that term now requires articulation if it is to be more precisely understood. While some elements of the open source model have been adapted, other elements have been left behind. To illustrate, I identify a few ways in which selected elements of the open source model have been adapted and diffused to motivate an exploration of the community-managed governance model. In doing so, I argue that the notion of community management has become decoupled from the notion of open source and that it is deserving of further inquiry in and of itself - particularly in light of the hybrid open source communities that are emerging.

2.2 Diffusion: new sponsors of open source projects

Individuals are no longer the only parties founding open source projects. The open source development model has been adopted by new sponsors that include public and private corporations, transnational organizations such as the European Union and the United Nations; as well as nation states like Brazil, Denmark, China, Japan, South Korea and South Africa; and foundations such as the Mellon Foundation. However, little research explores the degree to which open source projects founded by public or private sector organizations differ from those founded by grassroots

means. Such sponsors may not only face new challenges, but also vary in their use of community-management principles.

First, foundation, corporate, nation-state and transnational sponsors of open source projects are likely to face different challenges than grassroots founders, particularly if they are trying to build a community around code that has already been developed (O'Mahony and West 2006). When code and community do not develop in parallel, the learning curve can be steep, which can affect external developers' ability and motivation to contribute. For example, in a study of the Eclipse developer community (once an internal IBM project, but now independently managed), it took over 4.5 years to attract a pluralistic developer community (O'Mahony 2005). Sponsored open source projects may also develop different norms and rules which affect a volunteer's motivation to contribute (Shah 2006) and thus the ability of an independent community to form.

Second, much of our knowledge of the open source development model rests on infrastructure projects such as operating systems and webservers. New types of sponsors have expanded the use of the open source development model to end user applications. Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), Inventory Management, healthcare management applications all have open source equivalents. While it is not clear whether such projects are community-managed, it is easy to imagine that sponsors of software projects with more commercial applications may pursue greater control over their development process in order to be accountable to investors and customers (O'Mahony and West 2006).

Third, firms building open source business models have recently experienced an infusion of venture capital which can affect not only the number but the types of open source communities founded. Between 2004 and 2005, three times as many venture capital dollars (\$300 M) went to twice as many firms (33)—the largest spike in venture investment in OSS firms since 2000 (O'Mahony and Raj 2006). Since most firms funded to develop an open source business are likely to sponsor an open source project, this type of investment suggests that the trend toward corporate sponsorship of open source projects is not likely to dissipate soon. However, external investment brings additional stakeholders that can influence the design of open source communities and thus their governance.

Projects founded by new types of sponsors are often referred to as open source projects because they operate with open source licenses. However, the three points explored here: the challenge of building a community when code and community are not developed in parallel, potential differences in the nature of the technology itself, and an expanded base of stakeholders, may affect the degree to which the community-managed model can be applied by new types of sponsors.

2.3 Adaptation: importing the open source model to new contexts

Many organizations have borrowed elements of the open source model and applied them to new contexts. Some adapt legal and community elements of the open source model without necessarily importing the community-managed development model. For example, Science Commons and Creative Commons have introduced new legal

mechanisms influenced by OSS licenses to the fields of biotechnology and the arts. Creative Commons offers a modular copyright license that allows creators of cultural content to recombine use, attribution and redistribution rights to their work depending on their desire to share their work. Science Commons helps craft biological material transfer agreement protocols to facilitate the sharing of biological materials for research. Neither community integrates contributions from disparate parties as an open source community might. And as private foundations, they are not necessarily community managed. But by crafting a new set of legal mechanisms, they provide tools to enable a distributed group of creators to legally share and build upon each others' work (much like an open source community would), which has enabled a large community to grow around those tools.

Other communities have embraced the participatory and public development approach of integrating micro-contributions from a broader community without necessarily importing open source legal innovations. Communities like Wikipedia, Flickr, and del.icio.us all depend upon direct contributions of content from their members, but only Wikipedia remains a nonprofit enterprise. In corporate managed communities, access rights to the community are typically not distributed and decisions are centralized. In Wikipedia, rights are distributed to users: the ability to access and change content is open, subject to the norms of the community. Time, tenure, and a well-established set of normative rules determine which regular contributors acquire more rights within the community. Experts may argue over the quality of content produced (Giles 2005), but Wikipedia is arguably the first application of the open source community-managed model to a new context. As one of the world's top ten most visited websites (Lakhani and McAfee 2007), Wikipedia lends some credence to the community-managed model's relevance to new contexts.

Selective elements of the open source model have also been adapted by firms embracing open, participatory and distributed innovation processes. However, this can be achieved without incorporating either legal or community elements from the open source model. For example, in the 'open innovation' model, firms are encouraged to collaborate with hobbyists and user/innovators to obtain ideas to improve products and services (Chesbrough 2004; von Hippel 2005; Chesbrough et al. 2006). If commercial success requires iteration among innovators, then firms should open their boundaries to contributions from external parties. However, using external resources to broaden the innovation pipeline may not necessarily entail community. Organizations can broker problem solvers and seekers in a hub and spoke model without allowing a community to form (Lakhani 2006). In this context, openness usually refers to the degree to which a firm is willing to engage with external parties in their innovation process, but does not entail elements of community management or of legal innovation.

This brief overview of ways in which the open source development model has been adapted and diffused suggests that the community-managed aspect of the model is no longer tied to the idea of OSS development. The open source term is protected via a certification mark managed by the Open Source Initiative (OSI). However, as any contributor to an open source project will confirm, a project is not an open source project by its license alone—a project's ability to operate as a

community is critical. But there is no arbiter to determine whether a project is community-managed, independent, sponsored by a corporate or non-profit entity, or some hybrid. The information provided by either a license or a domain name (e.g. .org or .com) only conveys so much. More precise distinctions are needed to communicate the boundaries between firm, nonprofit organization and community. With more constituents, more capital, and a more diverse group of open source project founders, the idea of what it means to be an 'open source initiative' is no longer clear. With the new frame's success, comes the concern that the meaning of community management has become de-coupled from the term itself.

3 The community-managed governance model

Since Raymond first articulated the open source community development model, much academic research about OSS has followed. However, as Markus notes in this issue, the concept of community governance has received little attention. Drawing upon Lynn et al. (2000), Markus defines open source governance as "the means of achieving the direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of the OSS development projects to which they jointly contribute" (Markus, this issue).

The community-managed model of OSS development is one such mode of governance. To provide some theoretical grounding to the principles that constitute community-managed governance, I drew upon my prior research of four large, mature OSS communities (the Apache, Debian, GNOME, and Linux Standards Base communities) from 2000 to 2002. With data from over 80 informants, in-person observations, and project documents, I identified five features that informants from these communities associated with community-managed governance: (1) Independence, (2) Pluralism, (3) Representation, (4) Decentralized decision-making, and (5) Autonomous participation (Table 1).

3.1 Independence

Online and open source communities abound but not all of them are independent. A community that is independent does not rely upon the resources from any one organization, but is supported by a diverse body of participants. Control over the community is independent of any one sponsor but rests with the members of the community itself. While a formal governance body can ensure that the collective's interests are represented, it is not required. To be independent, decision-making at the lowest levels is unencumbered from any single external controlling influence.

A community-managed governance system operates outside the reach of authority embedded in employment relations. Contributors to an open source project may be volunteers or may be paid by their employers to work on the project, but decision-making on the project takes place independently from the employment structure that guides the workplace. While some form of hierarchy may be present in both community and firm models, the hierarchy that emerges in a community exists independent from the compensation and reward structure that is associated

Table 1 Principles of the community managed governance model

Principle	Definition and application
Independence	<p>A community managed governance model is one that is independent of any one sponsor. Independence can be examined by evaluating the:</p> <ul style="list-style-type: none"> Diversification of material support Decision-making structure Independence from employment relations
Pluralism	<p>A community managed governance model is one that preserves multiple and perhaps competing approaches, methods, theories or points of view. Pluralism can be examined by evaluating the:</p> <ul style="list-style-type: none"> Diversity of contributors Management of conflict Determination of leadership—processes and turnover
Representation	<p>A community managed governance model is one where contributing members can be represented in community-wide decisions. Representation can be examined by evaluating the:</p> <ul style="list-style-type: none"> Membership structure and rights Ways in which those in authority are accountable to contributing members Degree to which members can exercise voice on community members
Decentralized decision-making	<p>A community managed governance model is one where some degree of decision-making is decentralized. It can be examined by evaluating:</p> <ul style="list-style-type: none"> How contributors gain code commit rights How contributors gain access to the project's decision-making structure The degree to which project communications and activities are publicly available
Autonomous participation	<p>A community managed governance model welcomes participation and allows members to contribute on their own terms. Opportunities for autonomous participation can be examined by evaluating:</p> <ul style="list-style-type: none"> Newcomer joining rate Newcomer socialization opportunities Rate of idea generation

with a firm. For example, on the projects that I studied, even if a paid contributing developer changed employers, his committer status on the project remained unchanged.

A community's independence can be ascertained by understanding its: (1) basis of material support; (2) decision-making structure; and (3) independence from authority structures rooted in employment relations. For example, in communities like GNOME, members adopted formal means to ensure that their member-elected board of directors remains independent from any single external controlling influence. If a majority of board of directors is elected from one firm, provisions are made to have someone step down. Other communities like Debian and Apache have taken steps to ensure that monetary and in-kind support from corporate donors is diversified.

3.2 Pluralism

A closely related principle is the principle of pluralism. A pluralistic community allows many approaches, methods, theories or points of view to be legitimate or plausible in pursuing a course of action. It is up to each holder of a particular perspective to advance their own approach. By ensuring pluralism, community-managed projects not only prevent a dominant control group from forming, they also encourage the sustenance of a multilateral participant base.

The preservation of pluralism is critical to achieving the benefits associated with the open source development approach. For example, consider the well known phrase: 'given enough eyeballs, all bugs are shallow' (Raymond 1999: 27). The common interpretation is that different people 'see' different things. An alternative view is that diversity in location, application, usage needs, and computing environments allows many more permutations of testing than could be feasibly supported by a single firm. A reformulation of this lay theory might state that "greater variance in participation will lead to more resilient code as long as the number of possible combinations of code and computing environments exceeds the number of people participating". From this perspective, pluralism can foster quality.

Community-managed projects can be pluralistic but still operate with some degree of differentiation within their membership—pluralism does not require the absence of leaders. What it does require is equal access to the community's code and process—or equal opportunity to become a valued member of the community. A transparent development model that offers not only source code, but a publicly available model of trouble shooting and some insight as to the project's technical direction can enable pluralism. To determine to what extent a community is pluralistic, one might examine: (1) the geographic, demographic and functional diversity of contributing members; (2) how community members manage conflict; and (3) how leaders emerge and their rate of turnover. For example, Debian has a Technical Committee to foster resolution when technical conflicts occur. They are also one of the few open source projects to have their leaders elected annually. The Apache community typically resolves project level conflict within Project Management Committees (PMCs). Leadership turnover within these groups is ratified by members.

3.3 Representation

Because it is difficult to rest formal control in the hands of many (Harrison 1960; Coleman 1974), communities often find themselves needing to establish some means of representing the interests of their members. Communities may start as direct democracies and move towards representative democracies as they mature to help them manage their scale (Howison 2007). For example, the Apache, Gnome, and Debian communities did not start with a representative democracy but moved to one over time. When representatives are used in a community, their authority is often limited to make decisions on behalf of the project as opposed to earning authority over other members or over code level decisions.

Creating member representation requires a community to determine a membership base. This is done formally on projects such as GNOME, Debian, Apache and the Linux Standards Base and more informally on projects such as the Linux kernel project. The Linux Standards Base project has created three membership classes for all of the parties essential to their community: namely, firms, individuals and non-profit foundations. Each has different rights and responsibilities and representation in the community's governance.

To determine the extent to which a community provides representation, one might examine (1) the community's membership structure and rights; (2) ways in which those in authority are accountable to contributing members; and (3) the degree to which members can exercise voice. For example, in addition to the right to elect a leader, every Debian member also has the right to propose a General Resolution to recall its leader, which may or may not be seconded and voted on by the rest of the community.

3.4 Decentralized decision-making

A community-managed governance model distributes some decision rights to community members. If one organization holds all decision rights, then the project is not community-managed. In general, decision-rights operate at 3 levels: (1) code level decisions, (2) sub-project level decisions, and (3) community wide decisions. A code level decision involves any determination of new features, bug fixes, or modifications that should be included in the code base. Depending on the architecture of the project, this type of decision may happen within sub-projects or it may happen at the top level project. Some code level decisions may affect only one module while other types, like security decisions, may apply system wide.

A community-wide decision is a decision on the process, organization or assets of the community. This could include decisions on whether to allow a commercial organization the use of the community's image or trademarks or decisions that affect how the community as a whole may be represented to outsiders. In the Apache, GNOME, Debian, and the Linux Standards Base communities, community-wide decisions are determined by community representatives, while code level decisions remain the meritocratic purview of local code owners.

Of the community management principles articulated here, the distribution of decision making rights has been the most challenging for sponsor founded open source projects. In a study of 12 open source projects founded by corporate sponsors, O'Mahony and West (2006) found that most of these operated with a transparent development model but that few provided direct access to the code base or distributed decision rights to those outside the founding firm. Only 2 of 12 projects studied delegated release authority to community members at large (O'Mahony and West 2006).

A way to examine decision-making in a community is to identify (1) how members gain code level access rights (e.g. Von Krogh et al. 2003); (2) how members gain decision-making rights (e.g. Heckman et al 2006); and (3) the degree to which project communications and activities are publicly available. A public or transparent development process is necessary to support decentralized decision-

making so that a large body of people can learn enough to participate in decisions. This is a necessary, but not sufficient condition for community management.

3.5 Autonomous participation

Of course, a community will not survive long enough to develop a governance model if potential contributors do not have the freedom to contribute on their own terms. Part of what attracts open source developers to a community is the opportunity to learn, solve technical problems or improve their skills (Lakhani and Wolf 2005; Dalle and Jullien 2003; Hertel et al. 2003; Hars and Ou 2002).

The four communities I studied valued this element explicitly and often stated that it was critical to the ability to sustain a community. One informant from the Apache project felt strongly that the community should not limit opportunities for new contributors: “Serendipity still is an important component. You still need to plan for somebody you have not anticipated even playing a role—you need to create the opportunity for that to happen...[otherwise] you are turning away the possibility that somebody who is motivated to contribute actually would” (*Apache Contributor, interviewed on September 28, 2000*).

Thus, it is important that any governance model, whether informal or formal, does not impinge on members’ freedom to contribute on the basis of their own interests, motivations, and abilities. Communities often go further than this—recruiting and mentoring new members and highlighting opportunities for contribution through communication outreach. To determine whether a community provides enough access to foster autonomous participation, one could examine: (1) newcomer joining rate; (2) newcomer socialization opportunities; and (3) rate of idea generation.

4 Conclusion

This essay began by arguing that the community-managed governance model has become decoupled from the notion of an open source project. This may be due to the growing elasticity of the open source frame and its subsequent adaptation and diffusion. With growth in the diversity of sponsors and stakeholders of open source projects, new types of governance models may emerge. These models may borrow elements of the community-managed model or import it wholesale. Preliminary research suggests that hybridization approaches are likely. If we do not have a good understanding of how the community-managed governance model works in practice, our ability to research such hybrids will also be limited. As new types of sponsors launch open source projects and as other types of distributed content creation communities are formed, we need a framework to understand and compare community forms.

Thus, greater attention needs to be paid to the principles and processes that constitute community management. What does it mean to be community-managed? This essay has taken a step in answering this question by drawing upon field research of 4 large, mature OSS communities. I identified 5 common principles that

provide some grounded theory on the elements of community-managed governance: (1) Independence, (2) Pluralism, (3) Representation, (4) Decentralized decision-making, and (5) Autonomous participation. I have also provided some measures to operationalize these principles in the hope of fostering comparative research to determine their generalizability. Future research would also do well to examine how these principles are enacted throughout a community's lifecycle, for less mature projects may operate differently.

With better explication of the elements that underlie community-managed governance, we may be more equipped to understand not just new variants within the open source community, but other types of online communities. Future research needs to examine the ways in which new technological capabilities (such as wikis) and governance models (in particular the distribution of rights) create new affordances or constraints for community members. This should be of concern to both communities and firms, for the affordances and rights offered to a community are likely to affect the type of contributors attracted to a community (O'Mahony and West 2006; Shah 2006).

Acknowledgements Thanks to Joel West, Sonali Shah, Mitch Kapor, Mitchell Baker, Brian Behlendorf, James Howison, and Kevin Crowston for their comments on this paper and these ideas. Thanks also to the reviewers and editors for their helpful comments.

References

- Benford, R. D., & Snow, D. A. (2000). Framing processes and social movements: An overview and assessment. *Annual Review of Sociology*, 26, 611–639.
- Chesbrough, H. (2003). *Open innovation*. Boston, MA: Harvard Business School Press.
- Chesbrough, H. (2004). Managing open innovation. *Research-Technology Management*, 47(1), 23–26.
- Chesbrough, H., Vanhaverbeke, W., & West, J. (Eds.) (2006). *Open innovation: Researching a new paradigm*. Oxford: Oxford University Press.
- Coleman, J. S. (1974). *Power and the structure of society*. New York: W. W. Norton & Company Inc.
- Dalle, J. M., & Jullien, N. (2003). 'Libre' software: Turning fads into institutions? *Research Policy*, 32, 1–11.
- Giles, J. (2005). Internet encyclopedias go head to head. *Nature*, 438(7070), 900–901.
- Gladwell, M. (2005). The bake-off. *The New Yorker*, September 25. Retrieved March 7, 2007, from http://www.gladwell.com/2005/2005_09_05_a_bakeoff.html.
- Goffman, E. (1974). *Frame analysis*. Cambridge, MA: Harvard University Press.
- Harrison, P. M. (1960). Weber's categories of authority and voluntary associations. *American Sociological Review*, 25(2), 232–237.
- Hars, A., & Ou, S. (2002). Working for free?—Motivations of participating in open source projects. Proceedings of the 34th Hawaii International Conference on System Sciences. *International Journal of Electronic Commerce*, 6(3), 25–39.
- Heckman, R., Crowston, K., Li, Q., Allen, E., Eseryel, U. Y., Howison, J., & Wei, K. (2006). *Emergent decision-making practices in technology-supported self-organizing distributed teams*. Paper presented at the International Conference on Information Systems (ICIS 2006), Milwaukee, WI.
- Hertel, B., Niedner, S., & Herrmann, 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.
- Howison, J. (2007). *Effective organization for uncertain collaborations: Lessons from free (libre) and open source software development teams*. Presentation at the 2nd conference on the Institutional Foundations of Industry Self Regulation, Harvard Business School, Cambridge, MA. Retrieved March 2007, from http://www.hbs.edu/units/tom/conferences/docs/Howison_poster.pdf.

- Lakhani, K. (2006). *Broadcast search in scientific and technical problem solving: Finding solutions from the periphery*. MIT Sloan Working Paper. Retrieved March 7, 2007, from <http://www.stern.nyu.edu/ciio/WorkOnline/IS20052006/021506.pdf>.
- Lakhani, K., & McAfee, A. (2007). *Wikipedia (A)*. Harvard Business School Case 9607712. Boston, MA: Harvard Business School.
- Lakhani, K., & Wolf, B. (2005). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In H. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 3–21). Cambridge, MA: MIT Press.
- Lynn, L. E. Jr., Heinrich, C. J., & Hill, C. J. (2000). Studying governance and public management: Why? How? In C. J. Heinrich, & L. E. Lynn Jr. (Eds.), *Governance and performance: New perspectives* (pp. 1–33). Washington, DC: Georgetown University Press.
- McAdam, D. (1986). Recruitment to high risk activism: The case of freedom summer. *American Journal of Sociology*, 92, 64–90.
- O'Mahony, S. (2003). Guarding the commons: How community managed projects protect their work. *Research Policy*, 32, 1179–1198.
- O'Mahony, S. (2005). Non-profit foundations and their role in community-firm software collaboration. In J. Feller, B. Fitzgerald, S. Hissam, & K. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 393–413). Cambridge, MA: MIT Press.
- O'Mahony, S., & Raj, N. (2006). *Mozilla: The rebirth of Firefox*. Harvard Business School Case 907015, Harvard Business School Publishing, Boston, MA.
- O'Mahony, S., & West, J. (2006). *The participation architecture of online production communities*. Paper presented at the Academy of Management, Atlanta, GA.
- Raymond, E. S. (1999). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly.
- Shah, S. (2005). Open beyond software. In C. DiBona, D. Cooper, & M. Stone (Eds.), *Open sources 2.0: The continuing evolution* (pp. 339–360). Sebastopol, CA: O'Reilly.
- Shah, S. (2006). Motivation, governance & the viability of hybrid forms in open source software development. *Management Science*, 52(7), 1000–1014.
- Snow, D. A., & Benford, R. D. (1992). Master frames and cycles of protest. In A. D. Morris, & C. M. Mueller (Eds.), *Frontiers of social movement theory* (pp. 133–155). New Haven, CT: Yale University Press.
- Snow, D. A., Rochford, E. B. Jr., Worden, S. K., & Benford, R. D. (1986). Frame alignment processes, micro-mobilization, and movement. *American Sociological Review*, 51(4), 464–481.
- von Hippel, E. (2005). *Democratizing innovation*. Boston, MA: MIT Press.
- von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32, 1217–1241.

Author Biography

Siobhán O'Mahony is an Assistant Professor at the Graduate School of Management, University of California at Davis. Her research examines organizing processes in community forms, the management of common information resources, and factors affecting cumulative innovation.