

## THE EMERGENCE OF GOVERNANCE IN AN OPEN SOURCE COMMUNITY

SIOBHÁN O'MAHONY  
University of California, Davis

FABRIZIO FERRARO  
University of Navarra

**Little is known about how communities producing collective goods govern themselves. In a multimethod study of one open source software community, we found that members developed a shared basis of formal authority but limited it with democratic mechanisms that enabled experimentation with shifting conceptions of authority over time. When members settled on a shared conception of authority, it was more expansive than their original design. A statistical test of the predictors of leadership reinforced this finding. By blending bureaucratic and democratic mechanisms, the governance system evolved with the community's changing conceptions of authority.**

One of the most significant problems in organizational scholarship concerns how social collectives govern, organize, and coordinate the actions of individuals to achieve collective outcomes. Many classic works of organizational scholarship have grappled with this problem, leading to a variety of proposed optimal organizing forms and governance systems (Blau, 1955; Gouldner, 1954; March & Simon, 1958; Ouchi, 1979; Stinchcombe, 1959). Most attention has been devoted to bureaucratic forms, with very little attention given to community forms of organizing.

However, organizational forms that do not use a bureaucratic basis of authority have a long history (Coleman, 1970, 1974, 1993). Such forms have not

received the attention they deserve (Marsden, 2005; Stern & Barley, 1996), perhaps because the institutional persistence of the dominant corporate bureaucratic form (DiMaggio & Powell, 1983; Zucker, 1977) inhibits other sources of variety (Stinchcombe, 1965). In particular, relatively little is known about the process of organizing in communities—that is, how social groups accomplish the critical task of coordinating the actions of multiple individuals to achieve important outcomes (Heath & Sitkin, 2001; Weick, 1979). Furthermore, understanding community forms of organizing is important to organizational theory because greater variety in organizational forms increases the range of tools or solutions that society can bring to social problems (Hannan & Freeman, 1989; Rao, 1998; Romanelli, 1991; Stinchcombe, 1965).

This research takes a step toward filling this gap by examining how a social group designed a shared basis of authority and thus, a governance system. In conducting this research, we heeded Heath and Sitkin's (2001) advice to devote more attention to the general processes of organizing that help reveal how groups of people carry out their goals. Historically, the inability to develop a shared basis of authority has led many collectivist groups to fail (Coleman, 1980; Etzioni, 1959; Harrison, 1960; Swidler, 1979). Thus, governance in community forms not only offers a critical lens into conceptions of organizational control (Fligstein, 1990) but also provides an indicator of how communities can be sustained (Rothchild & Russell, 1986).

Our examination of the emergence of a governance system in an open source software community shows how a community uses a formal bu-

---

This research has been supported by the Stanford Center for Work, Technology and Organization, the Stanford Technology Ventures Program, the Social Science Research Council, the Harvard Business School Division of Research, a Spanish Government Research Grant (Ministerio de Educación y Ciencia - SEJ2006-11833) and the Public-Private Sector Research Center at IESE Business School. The data collection and processing efforts of John Sheridan, Vikram Vijayaraghavan, Daniel Berch, Mariano Belinky, and Jordi Colomer and editorial support provided by Clare Flaherty and Alyssa Razook were much appreciated. We thank Steve Barley, Beth Bechky, Bruno Cassiman, Michael Cohen, John King, Mark Mortensen, Woody Powell, Matteo Prato, Marc Ventresca, JoAnn Yates, and participants in Harvard Qualitative Inductive Research and the Strategy Research Forum for their helpful comments on previous versions. We also thank Sara Rynes and our reviewers for thoughtful and detailed comments that improved the article. All errors or omissions are our own. Both authors contributed equally to this work.

reaucratic basis of authority to reinforce its meritocratic norms. However, this approach depends upon democratic mechanisms that not only limit that basis of authority, but also allow the system to adapt with members' changing interpretations of leadership. After showing how the community introduced formal authority, we analyze conceptions of authority to explore how community members interpreted leadership over time. We find that although technical proficiency is an important criterion for leadership in such a group, it is not sufficient. Despite community members' espoused preferences for "hands-off leaders," skill in building the organization became increasingly important over time. We show this not only through our analysis of espoused conceptions of authority, but also by modeling the behaviors most likely to be associated with the community's leadership. We conclude by furthering a grounded theoretical perspective on governance in communities and explore the relevance of these findings for traditional organizations.

#### COMMUNITY FORMS OF KNOWLEDGE SHARING AND PRODUCTION

To further the research agenda described above, we focused on communities involved in knowledge sharing and production. Community forms appear to be increasingly important to solving problems and sharing knowledge (Brown & Duguid, 1991, 2000, 2001; Hargadon & Bechky, 2006; van Maanen & Barley, 1984) and may be well suited to an economy that relies upon the production and diffusion of knowledge (Adler, 2001; Powell & Snellman, 2004). More broadly, theorists now recognize that community forms can provide an alternative to market and hierarchical forms of organization and production (Adler, 2001; Bradach & Eccles, 1989; Ouchi, 1980; Powell, 1990). Yet little is known about how communities organized around production govern themselves.

Production communities typically shun bureaucratic elements such as an authoritative division of labor (Rothschild-Whitt, 1979). The goals of collective forms of production vary, yet they tend to share a common means: namely, one that embraces democratic participation in both production and management (Rothschild & Russell, 1986). Traditional forms include kibbutzim and cooperatives (e.g., Ingram & Simons, 2000; Kanter, 1968; Kieser, 1989; Rothschild & Russell, 1986; Rothschild & Whitt, 1986; Rothschild-Witt, 1979; Simons & Ingram, 1997; Swidler, 1979), but

their study has remained largely on the periphery of organizational theory's terrain (Coleman, 1970, 1974, 1993). More recently, research on communities engaged in learning and knowledge sharing has flourished.

Scholars have shown how members of occupational communities and communities of practice inside firms cooperate to further problem solving, learning, and skill development (Brown & Duguid, 2001, 2000, 1991; Lave & Wenger, 1991; Pickering & King, 1995; Wenger, 1998, 2000). Despite the fact that occupational communities extend outside the boundaries of an organization (Van Maanen & Barley, 1984), this construct has typically been used as a means to understand how individuals exert autonomy and control over their work inside organizations (Bechky, 2003; Orr, 1996). Although communities of practice help individuals achieve their learning goals, they typically do so within the context of a firm's objectives (Lave & Wenger, 1991; Wenger, 1998, 2000). In both cases, community members are engaged in managing production not for its own sake, but for the benefit of their employers. Thus, they are limited in the degree to which they can govern production. Because occupational communities and communities of practice operate within an authority structure that already exists, how such forms learn to govern themselves has not been a focus of this research.

Scholars have also shown how online communities outside firms share information and social support (Cummings, Sproull, & Kiesler, 2002; Fayard & DeSanctis, 2007; Smith & Kollock, 1999; Parks & Floyd, 1996; Parks & Roberts, 1997; Rheingold, 2000). However, little empirical work has examined how such communities manage production. In modern production communities, contributors, independently of their employment context, voluntarily collaborate to create goods or services for either public or private benefit (von Hippel & von Krogh, 2003). Members are typically geographically dispersed and depend upon the Internet as a means of communication and coordination (Kollock, 1998). Within the last decade, online production communities have begun producing information goods such as scientific knowledge,<sup>1</sup> art,<sup>2</sup> general knowledge,<sup>3</sup> and software.

Open source software communities are the example of online production most often recognized by organizational theorists (Kogut & Metiu, 2001; Lee

<sup>1</sup> See [www.sciencecommons.org](http://www.sciencecommons.org).

<sup>2</sup> See [www.creativecommons.org](http://www.creativecommons.org), [ccmixtr.org](http://ccmixtr.org), [flickr.org](http://flickr.org).

<sup>3</sup> See [www.wikipedia.org](http://www.wikipedia.org).

& Cole, 2003; Moon & Sproull, 2002; von Hippel, 2005; von Hippel & von Krogh, 2003). Scholarly analysis of open source software communities has addressed status dynamics (Stewart, 2005), member contribution patterns (Mockus, Fielding, & Herbsleb, 2002; Shah, 2005), and why individuals are likely to contribute to such efforts (Dalle & Jullien, 2003; Hars & Ou, 2002; Hertel, Niedner, & Herrman, 2003; Lakhani & Wolf, 2005; Lerner & Tirole, 2002). Researchers have learned that open source communities create informal and formal social structures to manage membership and joining processes (O'Mahony & Ferraro, forthcoming; von Krogh, Spaeth, & Lakhani, 2003), but little has been done to understand how these projects are governed (see Shah [2006] for a recent exception).

Production communities may be a fledgling form of organization, yet scholars and practitioners alike see them as increasingly important to an information- and knowledge-based economy (Armstrong & Hagell, 1996; Powell & Snellman, 2004; Sawhney & Prandelli, 2000; Seidel & Stewart, 2003; Williams & Cothrel, 2000). Production communities differ from the community forms that have previously been researched in three ways that make them theoretically distinct and ripe for study. First, unlike communities of practice or occupational communities, production communities are not associated with a single employer or workplace. Second, unlike online communities, production communities must integrate individual contributions into a common pool, which can heighten interdependencies and the need for coordination mechanisms (e.g., Thompson, 1967). Third, production communities often "own" the output of their efforts, and members work toward collective goals outside of the scope of their employment (O'Mahony, 2003).

Taken together, these three distinctions suggest that production communities need a way to manage their interdependence to achieve common goals. Yet, without the benefit of markets or hierarchies, they may have few resources with which to do so. Scholarly conceptions of community forms as ideal types tend to overweight the roles that norms, trust, mutuality, and reciprocity play in organizing community activities without addressing how people actually coordinate their work to achieve collective ends (Adler, 2001). To move beyond ideal types, we take seriously the complicated nature of directing individual efforts toward a common goal without the benefit of contractual or hierarchical reinforcement. In any organization, there is a constant tension between fulfilling individual goals and integrating them with a common objective (March & Simon, 1958; Ouchi, 1979). A focus on governance allows us to

explore how community forms resolve this tension.

## GOVERNANCE IN COMMUNITY FORMS

Before an organization can develop a form of governance, it must establish a shared basis of authority (Etzioni, 1959). Organizations without a consensual basis of authority lack an important condition necessary for their survival (Coleman, 1980; Etzioni, 1959; Harrison, 1960). Organizations with directly democratic forms of participation do not manage to scale well and are noted for having difficulty managing complexity and decision making—all of which can hasten their demise (Johnson & Whyte, 1977; Rothschild & Russell, 1986; Rothschild & Whitt, 1986; Whyte & Whyte, 1988). The need to coordinate interdependent member activities and integrate member contributions in a production context is likely to exacerbate the need for a shared basis of authority.

However, modern collective forms of production do not tend to rely on any single one of the three bases of authority (tradition, law, and charisma) theorized by Weber (1978). These forms are created with few traditions to guide them and so do not inherit any traditional basis of authority. Since there is no authoritative division of labor, collective production communities do not rely upon what Weber would call a legally rational basis of authority rooted in position. And finally, since many contributors may not have met in person, given that such communities are usually geographically distributed, developing a basis of authority that rests on charisma, although possible, is less likely (Wellman & Gulia, 1999; Wellman, Salaff, Dimitrova, Garton, Gulia, & Haythorntwaite, 1996).

Weber recognized that "anti-authoritarian" systems that placed a high value on individual autonomy existed (Weber et al., 1978), but "he did not systematically analyze the problems of authority and power peculiar to these groups" (Harrison, 1960: 233; Satow, 1975; Willer, 1967). However, developing some form of authority is a particular problem for voluntary social groups that can lead to the compromise of their goals:

The ideology of voluntary social groups in America tends to be anti-authoritarian. The constituency of these groups is distrustful of centralization and further rationalization of their organizations. However, to achieve the imperative goals of these voluntary associations, bureaucracy is necessary, social tension increases, and the problems of authority and power become increasingly acute. (Harrison, 1960: 232)

For a production community to retain the interest and commitment of voluntary members, any form of authority introduced must simultaneously preserve democracy and accountability to its members. To achieve the efficiencies for which bureaucracy is known, some form of rationalization is necessary (e.g., Chen & O'Mahony, 2007). How communities resolve this conflict is underexplored, largely because of unresolved theoretical issues in the conceptualization of bureaucracy.

In a bureaucratic system, a legal and rationally based positional authority decouples the authority of a person from that of his or her position to prevent incumbent patrimony or favoritism (Weber, 1978). In Weber's eyes, meritocracy was an inevitable outcome of bureaucratic rule. Those with more technical competence were rewarded with positional authority; that is the essence of the "exercise of control on the basis of knowledge" (Weber, 1978). Parsons (1947) and Gouldner (1954) were among the first to note that Weber's notion of bureaucracy could lead to contradictory outcomes, arguing that positional authority and technical competence could be decoupled and lead to bureaucracy without meritocracy.

Adler and Borys (1996: 62) traced organization scholars' conflicted approach to bureaucracy back to this source of ambiguity and proposed a way to help reconcile conflicting evidence on the impact of bureaucratic rule. They argued that bureaucratic rules can be applied to either help people in their jobs (enabling), or to control them (coercive). They suggested that enabling bureaucracies are better suited for creating knowledge and that autocratic bureaucracies are better suited for predictable environments. Adler extended this line of thinking by exploring the contours of community forms, tentatively concluding that community forms of knowledge creation are likely to grow only if community norms are balanced by "hierarchical rules to ensure stability and equity" (2001: 228; see also Adler & Heckscher, 2006). This view suggests that community forms must blend some type of positional authority with more participatory and democratic means. However, little empirical work has extended this theoretical framing.

Recent scholarship on open source communities suggests that any governance system introduced to them must be meritocratic to attract high-quality contributions from voluntary members (Lee & Cole, 2003; Kogut & Metiu, 2001; Moon & Sproull, 2002). By rewarding merit with greater status, responsibility, or opportunities to enhance development (Stewart, 2005; von Krogh

et al., 2003), production communities can satisfy contributors' needs for recognition and reward in ways that their work lives may not (e.g., DiMaggio & Anheier, 1990; Drucker, 2001). Although it is widely recognized that successful open source projects often have strong leaders (Mockus et al., 2005; Moon & Sproull, 2002), few have examined the roots of such governance systems.

To further a grounded theoretical understanding of how communal forms organize, we collected both qualitative and quantitative data to examine how one open source community designed and implemented a governance system over 13 years. We identified four distinct phases of this process: de facto governance, designing governance, implementing governance, and stabilizing governance. We found that the failure of autocratic rule in the de facto governance phase spurred the design of a formal governance structure that included a positional basis of authority. To reinforce the community's meritocratic norms; this bureaucratic basis of authority was simultaneously limited with a directly democratic mode of governance. However, the role of a leader in the community remained open to interpretation. Thus, we evaluated espoused conceptions of authority in the implementing governance and stabilizing governance phases to learn how members interpreted the leadership roles they created. Finally, to move beyond analysis of the rhetoric of the community, we investigated what leadership *behaviors* became most valued by members of the community in their final phase of governance.

This research provides two distinct theoretical contributions. First, we clarify earlier speculation about how production communities organize (e.g., Adler, 2001; Adler & Borys, 1996) and find evidence of a limited form of bureaucracy that is more enabling than it is coercive. The simultaneous introduction of democratic and bureaucratic practices not only limits the reach of bureaucracy, but also allows members to experiment with varying interpretations of authority. These findings provide insight as to how community forms develop both a shared basis of authority and a governance model, thus speaking to a fundamental, long underaddressed question for organizational theory (Coleman, 1980; Harrison, 1960; Satow, 1975). Second, we show that even in a community of open source programmers that espouses the value of technical contributions above all else, members' conceptions of leadership change over time to increasingly value organization-building contributions. Democratic mechanisms enable the community's governance

system to adapt as members learn how to interpret leadership and authority in a community context. This observation suggests an evolving and context-dependent notion of meritocracy and that democratic mechanisms serve an important adaptive function in emerging organizational forms.

## PART I: INDUCTIVE APPROACH

In Part I, we use ethnographic and archival data to explore how the governance system designed by this community operated over a 13-year period, and how different conceptions of leadership emerged that invoked varying degrees of authority over time. An inductive approach is particularly apt for examining phenomena that are emergent or poorly understood (Strauss & Corbin, 1990), allows room for the unanticipated, and is most suitable for grounded theory building (Edmondson & McManus, 2006)—the goal of this research. In Part II, with quantitative data gathered from the same setting, we pursue a deductive approach to examine the behaviors that were most important to community members in filling the leadership positions they created.

### Research Setting

We used theoretical sampling (Glaser & Strauss, 1999; Strauss & Corbin, 1990) to select an open source community that had two features that would allow us to explore governance in depth: long duration and leadership turnover. Successful leadership turnover implies that leadership positions have become institutionalized or decoupled from the founder of a project. Of several open source software communities studied in the first author's prior research, the Debian community was selected for further analysis because its longevity (it was formed in 1993) exposed it to governance issues that occur only over time. Preliminary fieldwork also revealed that the Debian community had, unlike other open source projects previously examined (e.g., Lee & Cole, 2003; Moon & Sproull, 2002; Raymond, 1999), experienced frequent leadership turnover since the founder left the project in 1996. As one informant explained, this was perceived to be a strength of the project and a source of longevity:

I have complete confidence in the Debian project to find its own path to the future. Four different project leaders over the past five years have done nothing to shake that confidence. We have shown ourselves quite capable of thriving despite changes in leader-

**TABLE 1**  
**Data Sources**

Data Source	Interviews	Project Records
Debian project leaders	4	
Debian contributors	14	
User contributors	30	
Leader candidate platforms (1999–2006)		34
Election debates (2000, 2003, 2005–06)		4
Meetings (1998–2005)		32
Project general resolutions (1999–2006)		5
Mailing list postings (1998–2006)		17,317
Project documents (constitution, policy manual, social contract)		100+ pages

ship, and without leaders who felt a strong desire to change Debian's vision.

As of this writing, nine different leaders have led Debian over 13 years, suggesting that the community has created a mode of governance independent of its founder.

Debian is a free operating system that uses the Linux kernel developed by Linus Torvalds; it is a Linux *distribution*. An operating system is the set of basic programs and utilities that make computers run. The kernel is the most fundamental program on a computer that manages demands on system resources: It allows you to run multiple programs simultaneously. According to industry analysts, Debian has 25 percent of the market for Linux distributions, second only to the Red Hat distribution, which is produced by Red Hat, a publicly traded company (Netcraft, 2005). Over 1,000 Debian community members in over 40 countries contribute to the development of the Debian Linux distribution (which comprises over 9,000 packages of code). Developers who contribute to the community consider themselves to be part of “an association or a club, much like your local LUG [Linux user group] or Rotary, with the principle exception being that we hardly ever meet face to face.”<sup>4</sup> Although the Debian community does not sell the code they produce, over 150 vendors commercially distribute the project's software. In return, Debian receives corporate support and equipment donations from at least ten firms, two of which are *Fortune* 500 firms commercializing Linux (e.g., Casadesus-Masanell &

<sup>4</sup> From step 2, Debian New Maintainer process, located at <http://uk.debian.mirror.anlx.net/devel/join/nm-step2>.

Ghemawat, 2006; Fosfuri, Giarratana, & Luzzi, forthcoming).

## Methods

Data on the Debian Linux community were collected as part of a larger ethnographic study of the open source community. Table 1 lists the sources of data used for the current study. Forty-eight semi-structured interviews were conducted with Debian contributors, focusing on membership, sponsorship, decision making, and governance. These data were supplemented by observations and analysis of 34 leader candidate platforms, 4 election debates, notes from 32 meetings, 5 general resolutions, and 17,317 postings to the project mailing list. We searched mailing lists and collected project data from online archives to identify topics related to leadership and governance. These archives included the project's constitution, policy manual, reference manual, social contract, charter, and by-laws. Triangulation among multiple sources of evidence provided greater depth and accuracy by allowing us to draw upon different perspectives (Campbell & Fiske, 1959; Yin, 1994).

We analyzed these ethnographic data using Atlas TI software, an open coding application for qualitative data. Our analysis proceeded through four rounds of coding. In the first round, we coded governance decisions and events in chronological order; the recall of Debian's second leader and the subsequent creation of a constitution that outlines the parameters of a leader's authority are examples of such events.

In the second round of coding, each author independently identified phases in the evolution of Debian's governance system: (1) *de facto* governance (1993–97), (2) designing governance (1997–99), (3) implementing governance (1999–2003), and (4) stabilizing governance (2003–06). We identified these phases by noting the presence of defining governance events from the first round of coding. Table 2 lists the events, examples of evidence for them, and information on the type and strength of evidence. We resolved any discrepancies between our independent codings by revisiting and discussing the data and coding.

In the third round of coding, we wanted to determine how conceptions of authority evolved. After 1999, the Debian community formally elected leaders. Thus, we independently coded data on leader candidates' electoral platforms and election debates from 1999 to 2006. In these leadership platforms and debates, leader candidates articulated their visions for the project, their ideas to improve it, and their claims as to what they would do as

leaders. They also made prescriptive statements about what they thought a leader should and should not be able to do in the community. We assessed leader candidates' degree of authority by evaluating the decision-making power that candidates associated with the leadership position. After resolving minor disagreements in classification, we identified and named five conceptions of leadership that varied in their degree of authority and the extent of their focus on organizational versus technical concerns: (1) hands-off leader, (2) technical manager, (3) visionary leader, (4) organization builder, and (5) organization leader. Table 3 depicts these sequential conceptions and provides exemplary evidence for them.

We replicated this coding with available data from *de facto* leaders from the period 1993–98. We asked an external researcher to classify the platforms independently and compared this classification with ours. The independent researcher rated the leadership platforms on the basis of the descriptions articulated here. Comparing this researcher's codes with ours, we calculated our interrater reliability using Cohen's kappa and obtained very satisfactory results ( $\kappa = .80$ ; 84 percent agreement rate). After considering the relevance of the disagreements encountered, we deemed it safe to proceed with our original coding. In identifying these five conceptions of leadership, we noticed a shift from technical to organization-building concerns. To further understand the contours of this shift, we studied the content of leadership platforms at another level.

In the fourth round of coding, each author independently coded the 34 leadership platforms and classified parts of the texts as focused on addressing organizational issues, engaging in technical discussions and debates, or providing biographical information. One platform statement from 2003 provides an example of a text item coded as "[related to] organizational issues": "I think one of the main tasks of the Project Leader is to coordinate and motivate people—to lead. This is why I said that while the external functions of the DPL are important, the internal functions are even more important. While it is quite hard to lead a project consisting of so many people with so diverse expectations and personalities, I think that it can actually be done. Thus, my main aim as Debian Project Leader is to lead, motivate and coordinate." An example, from another 2003 platform, of text coded as "[related to] technical issues" is, "Similarly, while everyone complains about the release cycles of Debian, it is not clear at all how frequently we should release. For example, SuSE and Red Hat

**TABLE 2**  
**Phases of Governance in an Open Source Community**

Defining Event <sup>a</sup>	Exemplary Evidence	Evidence Strength <sup>b</sup>
<p><i>Phase I: De facto governance, 1993–1997</i></p> <p>Autocratic leadership emerges and is challenged</p>	<p>But the big thing, which changed with [the 2nd DPL], like he was really managing everything and kept tight control, which, was plausible because Debian was much smaller. Then actually [with the third leader] he never wanted to have someone like [that] again, and that is why he started to write the constitution, which splits the power. (Debian developer member)</p>	<p>Medium</p>
<p><i>Phase II: Designing governance, 1997–1999</i></p> <p>Formal authority is developed</p>	<p>The Project Leader may . . . define an area of ongoing responsibility or a specific decision and hand it over to another Developer or to the Technical Committee; . . . lend authority to other Developers; . . . make any decision which requires urgent action; [or] for whom no one else has responsibility. (Debian constitution, article 5.1)</p>	<p>High</p>
<p>Formal authority is limited through democratic means</p>	<p>Together, the Developers may: 1) Appoint or recall the Project Leader. 2) Amend this constitution, provided they agree with a 3:1 majority. 3) Make or override any decision authorised by the powers of the Project Leader. (Debian constitution, article 4.1)</p>	<p>High</p>
<p><i>Phase III: Implementing governance, 1999–2003</i></p> <p>Varying conceptions of formal authority are debated</p>	<p>It's not clear exactly what the DPL is supposed to do. What do people expect? Because Debian was so big, everyone expects something else. So, some people say, "Yeah the DPLs should only go to conferences, and present Debian to the outside world, but he shouldn't do anything internal because that is working anyway." (Debian developer member)</p>	<p>High</p>
<p>Community members elect leaders through democratic means</p>	<p>I'll cast my DPL vote towards the end of the cycle, as usual. Here's why: At the start of the cycle, I hadn't made up my mind. . . . I haven't finished reading the candidate platforms and debate material yet. . . . This year, since once again I couldn't attend them live, I have to read them afterward, which takes time. . . . Voting in Debian is just like voting anywhere else, you often have to do a lot of reading to understand the issues. (Debian developer member)</p>	<p>High</p>
<p><i>Phase IV: Stabilizing governance, 2003–2006</i></p> <p>A shared conception of formal authority emerges</p>	<p>When community members became disgruntled with a leader's actions, they worked within the system by proposing a General Resolution to recall the leader. However, members voted to discuss the matter further and, in a second vote, the leader's authority was re-affirmed. (general resolutions, 2006)</p>	<p>Medium</p>

<sup>a</sup> All three types of data—interviews, secondary sources, and project documentation—provided evidence for for all four phases.

<sup>b</sup> "High" indicates that a finding was apparent in most of the data. "Medium" indicates a finding was consistently repeated. "Low" indicates that a finding was suggestive.

have introduced offerings of their distributions that have similar release cycles to that of Debian. There are many reasons why you can upgrade your system only every two years. Yet, there are also users

who want the newest software and like to upgrade twice a year."

Finally, an example of text, also from 2003, coded as "[related to] biographical information" is

**TABLE 3**  
**Debian Project Leader Conceptions of Authority, 1993–2006**

Conceptions of Authority	Degree of Positional Authority	Exemplary Evidence <sup>b</sup>	De Facto Governance		Designing Governance				Stabilizing Governance					
			1993–96 <sup>b</sup>	1996–97 <sup>b</sup>	1998 <sup>b</sup>	1999	2000	2001	2002	2003	2004	2005	2006	
Hands-off leader	Limited	The DPL should have a fairly hands-off approach . . . the DPL will need to delegate some of his responsibilities to others.				2	1	1					1	
Technical manager	Medium	The DPL is the public figure head of Debian. (2000 leader election) A chronic problem in Debian has been our overly long release cycle, if elected this shall be my major focus. (2000 leader election)	1			1	1							1
Visionary leader	Medium	It has been a long time since Debian had a clearly articulated vision. It's time to fix that . . . there is no reason that we can't eventually be a truly universal operating system! I can't imagine a more compelling vision for our future. (2002 leader election)				1				1	1			
Organization builder	High	We have learned a harsh lesson about the organizational structure in Debian. Debian is continuously growing and changing and we need to know how to handle that. Old structures and ideas will need to be revised, and new ones will arrive. (2000 leader election)				1	2	2	1	1	1	1	1	1

Continued on next page





the following: "I hold a Master degree in Philosophy and have recently completed a Master of Science in Psychology. I'm currently doing a Master of Software Systems Engineering at the University of Melbourne and am looking forward to pursuing a PhD in Software Engineering about Debian and Free Software afterwards." We counted the number of words dedicated to each of these three themes and compared their evolution. To ensure the robustness of our results, we asked another researcher to code the platforms independently and compared his results to ours. There was a .85 correlation in the results for organizational issues, .95 for technical issues, and .90 for biographical content. Given these results, we deemed it safe to proceed with our original coding.

### Analysis and Results

In this section, we present results of our analysis of our qualitative data to show how the community transitioned from a de facto governance model to one that integrated positional authority with directly democratic means. We then trace how community members subsequently interpreted a leader's authority. In Part II of the article, we present a model based on quantitative data from the first year (2003) of the final phase (stabilizing governance) of our study. This model predicts which developers were the most likely to become members of the project's leadership team.

**Phase I: De facto governance (1993–97).** We found that for its first five years, the Debian project operated reasonably well without a formal means of representing its contributors in project governance. The founder of the project considered contributors' opinions as they were expressed but had the final say on decisions. When the founder left the project to pursue personal interests, he informally passed the leadership to one of his trusted lieutenants, who maintained a very active role on the project. The decision to transfer authority was made unilaterally, and the second leader continued in the absence of any formal governance process to provide contributing members with formal representation or to resolve disputes. However, he did hold a more expansive conception of the leadership role than did the founder. We found that subsequent misinterpretation of the appropriate role of a leader in the community led to the failure of auto-cratc leadership and the construction and delimitation of positional authority.

Without agreement on the leader's authority, members resisted what some characterized as auto-cratc leadership. Aware of the problem, the second

leader recalled, "There was no formal structure whatsoever at that time. You had to lead where people would follow otherwise they would just walk off." When the second leader began referring to his role as that of "president" and initiated a series of organization-building projects to manage Debian's growing scale, community members balked at his actions. In assuming the title of president, the second leader had essentially assigned himself positional authority without developing a shared basis for it.

The project leader felt this strain and, to both enhance his legitimacy with project members and develop a shared basis of authority, proposed that the community elect a board of directors. In a message to the community, he explained that "this election relieves me of the (often quite awkward) position of being the only person with any real authority over Debian." His intent was to create "a leadership team for Debian that can survive the departure of any team member." In the months that followed, members lost faith in his leadership and asked him to withdraw from the project's first election. "A number of developers convinced me that it was time for a leadership change, and I thus withdrew from the project leader election." Although the second leader had initiated the design of a more formal means by which project members could achieve representation in governance, he might have introduced too many changes too quickly for project members concerned about a potential concentration of power. The "recall" of the second de facto leader led directly to a phase in which members designed a governance system that could represent their interests.

**Phase II: Designing governance (1997–99).** We found that community members' shared experience in recalling the second leader inspired the design of a governance system that could operate independently of any one person's conception of leadership. The third leader to assume the title did so only by vowing to be less directive than the second and leading the collective drafting of a constitution to formalize leadership roles, rights, and responsibilities. As one community member recalled, "[The third project leader] said we should formalize things and come up with a structure that could address the problems faced by a Debian Project Leader. . . . The project leader was looking at the way the project was going . . . and saying, no way is this going to work, we have to do something." The third leader led a debate on the mailing list in which a constitution was drafted, revised, and finally ratified by 357 developers. As the third leader explained, "We basically ratified the constitution using itself. That was sort of a test case as well,

which went pretty well, so we use it for project leader elections." The governance system designed at this time embraced two important and potentially contradictory elements: (1) formal positional authority and (2) limitation of that authority through democratic means.

Positional authority is rooted in a defined role, decoupled from a person's individual characteristics; it separates the personal from the official and provides a legal-rational basis of authority (Coleman, 1980; Merton, 1940; Weber et al., 1978). The Debian constitution outlines positional authority by specifying the rights and limitations of contributing members and positions of power, such as that of Debian project leader (DPL). As one member explained, "We had this DPL project leader but we never really said what he could or could not do. He had been doing things, so it [the constitution] kind of codified what he had been doing or what we wanted to be doing and what we did not want him to be doing." With the constitution in place, the project held its first election in January 1998, introducing the role of project leader (as opposed to "president"). However, we found that Debian members were only interested in supporting a positional basis of authority if this role was also limited in ways that facilitated democratic control by the rest of the community.

We found that positional authority, once created, was limited in four ways that preserved democratic rule. First, the Debian constitution requires those with positional power to defer to the wishes of the collective by making "decisions which are consistent with the consensus of the opinions of the developers." In practice, this tenet was emphasized often, and the project leader's ability to make unilateral decisions explicitly questioned. For example, years later, a participant in the 2006 leadership debate emphasized that "technical decisions shouldn't be made by the DPL, period." In response, another participant wrote, "The DPL's role is to lead discussion within Debian, and technical discussions are what makes Debian great. That's different from actually making the final decision though, which generally isn't the DPL's place." The concept of leader in this community is thus based on consensus building as opposed to autocratic rule, and the power of a leader only comes from his or her deference to democratic values.

Second, a Debian project leader is subject to the same rules as any member; he or she is not entitled to special privileges. In practice, this means that the project leader has no more power over what goes into Debian's code base than any other member. The authority that comes with the role is just enough to encourage consensus building. As an-

other community member explained, "The DPL in this case is just a normal developer, with a higher soap-box. People are more likely to listen to him and will perhaps decide to follow his attempts at moderation." Just like a developer, a project leader can only enact a policy change through a general resolution, which must be approved by a majority of members. Developers vying for the leadership role realized that they thus needed a rationale as to why they wanted the position. For if a leader did not have more authority than any other member, why would one want to pursue the position? A winning candidate in 2006 admitted as much when exploring this question: "I don't believe this [the DPL position] would dramatically alter my work for Debian, rather it would allow me a bit more flexibility in achieving the goals I just mentioned and thus let me give them a higher priority."

A third limit on positional power is the failsafe measure of recall by the collective via general resolution. Any member has the right to propose a general resolution that can counter a leader's actions. As one project leader candidate noted, "The DPL is not particularly empowered to stand against the will of the developers, since they can overrule anything he or she does with a General Resolution [GR]. However, a GR is a weighty process and a DPL who consistently acts in discord with the will of the developers can do damage simply by wasting the Project's time." Our analysis of the history of use of general resolutions noted that they were used very rarely (five times in eight years).

The fourth way that democratic rule constrains positional authority is through a countervailing source of authority. For example, projectwide decision-making power is split between the project leader and a technical committee empowered to "decide any technical matter where developers' jurisdictions overlap." The technical committee cannot introduce new proposals but has the authority to resolve disputes. In order to overrule a developer, a supermajority (three-fourths) of the committee must agree; in effect, consensus is mandated. As one long-time member explained, the technical committee "actually has more power than the DPL in some regards" and further limits positional authority.

The technical committee is able to say, we have looked at both sides of the discussion, if we do it your way, every single maintainer in Debian has to recompile his packages, we just don't think that is a good idea. So they will say, you "know they will vote against you." And that is what it really is for, is when the DPL can't really say it, when the rest of Debian can't decide or won't decide or whatever, the

DPL says fine, we will decide with the technical committee.

The Debian project leader can call upon the technical committee to help reach a decision but cannot force a decision, so the leader's ability to unilaterally settle disputes is limited.

By approving the constitution, project members developed a shared basis of positional authority. The four mechanisms described above limited the reach of bureaucratic rule by reinforcing democratic rule by community members as a whole. In the next two phases, the integration of democratic mechanisms also ensured that the governance system, once implemented, could evolve with the wishes of the collective.

**Phase III: Implementing governance (1999–2003).** In 1999, with the constitution ratified, Debian developers began electing project leaders for one-year terms, effectively initiating a phase of experimentation with the leader role. Candidates nominated themselves and posted leadership platforms on the Web site outlining their goals for the project nine weeks before the end of the previous leader's term. Platforms were debated in a three-week polling period that provided ample time for members from all time zones to participate. A long polling period was important so that volunteers who did not work on the project every day, or even every week, had a chance to vote. Participation in early leader elections ranged from 51 to 60 percent and somewhat declined (to 43 percent) in later years.

Although most developers happily went about their work without too much thought to governance, a leader could now be singled out when the community ran into difficult or ambiguous situations, and our data show that members valued this. The year following the passage of the constitution, one project leader candidate noted that "it introduces a bit of official rules and politics, but I think it will allow us to work as the sort of organized anarchy that we have always used while adding some much needed safety nets." Still, a governance system designed on paper had to be put into practice—that is, Debian members had to interpret exactly what this new role would entail. Our ethnographic data indicated that our informants varied a great deal on their interpretation of the project leader role. To provide better traction on how the leader role was interpreted, we turn to our analysis of the leader candidate platforms.

Given the importance the community placed on delimiting positional authority in phase II, designing governance, we expected community members to prefer "hands-off" leadership. However, our data

suggested otherwise, as a hands-off conception of leadership never received a majority vote from community members. Table 3 indicates how often each conception of authority was represented in each election year and which one won the majority vote.<sup>5</sup> We found that a *hands-off* conception of authority was characterized by a belief in the self-organizing ability of the community and the danger of expanding the authority of the project leader. For example, a leader candidate in 1999 stated that he "didn't see the project as something that can be steered or directed," and submitted no "specific plan" in his leadership platform. A candidate in 2001 stated that the leader role was "important but limited, with the majority of the power to act rightfully left in the hands of developers." Despite the community's interest in limiting positional authority, no leader won an election with such a platform.

The *technical manager* conception of authority focused on the coordination of technical work. These leader candidates were concerned with shortening and enhancing the predictability of release cycles. Technical managers focused on engineering management skills and tools to remove neglected packages, reduce bugs, and improve the software development process. Accomplishing these goals would require more authority than hands-off leadership entailed. For example, a candidate in the 1999 elections wanted to "create a stable and bug-free distribution for our users" and devoted most of his election platform to specific technical issues, arguing that "once there is a deep freeze, a group should be assigned to assess the bugs affecting the current frozen distribution."<sup>6</sup> Technical managers also never won an election.

Candidates with a *visionary leader* conception of authority focused on the need for a cohesive project vision, without advocating specific technical or organization-building activities. These leader candidates argued that the community needed to define a clear vision for the future. However, they suggested that it was their prerogative to define such visions, implicitly questioning the community's

---

<sup>5</sup> Although the first three leaders did not create platforms nor run for office, we coded their conceptions of authority based on our interview and mailing list data and included that data in Table 3. Inclusion of these data allow for a comparison of the evolution of interpretation of authority over the project's lifetime (1993–2006).

<sup>6</sup> Code in "deep freeze" cannot have new features added to it. This leader wanted to stop the addition of new features (which was more interesting to volunteer contributors) and assign people to work on fixing new bugs (which was less interesting and more mundane work) so that the release could be produced.

ability to collectively do so. Visionary leaders felt comfortable taking unilateral initiatives consistent with their visions, without necessarily making them collective efforts. For example, a winning candidate in 2002 stated, "Debian has achieved some wonderful results, but whether we will continue to improve is largely a function of how motivated we are. My experience leading volunteer efforts suggests to me that the best motivators [have] a strong shared vision which each participant can feel connected to, and processes that allow contributors to see results from their efforts." Leaders with visionary platforms won elections in 1999 and 2002 but not after 2002.

Some candidates espoused a conception of authority that emphasized skill as an *organization builder* and the ability to develop organizational structures and processes to improve the community as a whole. Organization builders proposed projects that were more organizational than technical. For example, an unsuccessful candidate in the 2001 election proposed culling developers from the project who were "no longer really contributing." "I propose that we experiment with and ultimately apply, automated tools for tracking package and developer activity, and act accordingly," this candidate stated. This proposal, which would have required explicit monitoring of members, was never implemented. Such a change would obviously have expanded the Debian project leader's authority to new domains. Candidates with organization-building platforms won elections in 2000 and 2001.

#### **Phase IV: Stabilizing governance (2003–06).**

Variation in conceptions of leadership that prevailed throughout the third phase began to reach settlement with the emergence of a new conception of authority, the *organization leader*, in 2003 (see Table 3). Organization leader platforms connected both organization-building and technical activities to the project's vision and goals. Unlike organization builder candidates, who often presented extensive lists of process improvement projects, candidates with an organization leader concept focused on the challenges of motivating volunteers and aligning their interests with those of the community. These leader candidates were more likely to recognize the importance of communication, culture, and relational skills relative to procedural improvements. One unsuccessful candidate in 2005 argued that "the overall goal is to make Debian a fun and rewarding context to work and spend time in, so much so that one misses and longs for it when one can't be or work on in it." To achieve this goal, this candidate recommended "small teams . . . , a more friendly and helpful en-

vironment . . . , making people aware of their leadership role . . . and having more frequent real-life meetings."

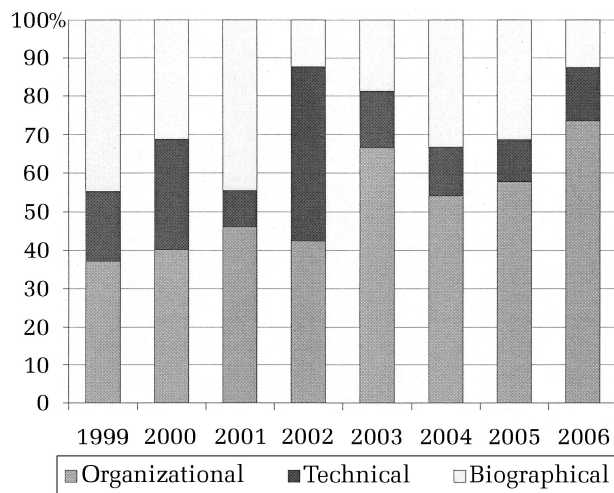
Organization leaders openly acknowledged that although there were many things they would like to change, their ability to implement them was limited. They thus recognized that powers of persuasion would be critical, given the limited authority associated with the project leader position. For example, an unsuccessful 2006 candidate said:

I must also acknowledge the fact that the DPL does not have the power to simply impose changes as he/she sees fit. The best that the DPL can do here is to encourage us to improve, sometimes by discussion and debate and sometimes by leading by example. . . . My own priority would be to fix some of the social issues first, as these are most pressing and in my opinion the most likely to cause long-term damage to the project.

Like visionary leaders, organization leaders were aware of the need to develop a compelling vision for the community. However, they did not think it was their job to do so alone. Rather, they saw the job of the leader as corralling and organizing members around a common vision. Organization leaders were also more cognizant of the limitations of a project leader's authority to actually implement organization-building projects than were visionary leaders. This awareness may reflect the community's learning what a leader could actually accomplish in this environment as well as a growing recognition of the importance of other members' motivations to contribute to the project. However, the amount of authority needed to carry out such a platform was still high. The last four elections in our study period (2003–06) were won by organization leaders and represent a limited consensus on the degree of authority appropriate for a project leader.

To verify whether a general trend from technical to organizational concerns was present in these conceptions of leadership, we turn to our textual analysis of leader platforms conducted in the fourth round of coding. This analysis confirmed that, over time, project leader candidates focused their electoral platforms increasingly on organizational issues, as opposed to technical issues. Figure 1 graphically depicts these thematic patterns over time. For example, in 2006, 73 percent of the text of a leader candidate's platform, on average, was devoted to organizational issues, compared to only 37 percent in 1999. To test whether these changes were statistically significant, we regressed the proportions of text dedicated to organizational or technical issues and to biographical information on a

**FIGURE 1**  
**Word Use in Leader Candidate Platforms**



trend line, measured with a scale ranging from 1 to 8. The regression models testing the changes in the proportion of technical issues and biographical information were not significant, but when we regressed the proportion of text dedicated to organizational issues on a trend line, the resulting model was statistically significant ( $F = 19.3$ ,  $df = 6$ ,  $p < .01$ ;  $R^2 = .76$ ), and the coefficient for the trend line was 0.05 ( $t = 4.39$ ,  $p < .01$ ), meaning that, on average, the proportion of text dedicated to organizational issues increased every year by 4.7 percent.

Debian's achieving some consensus on a preferred conception of leadership does not imply that the community was free from strife or conflict (e.g., Coleman, 2005). Our claim is merely that in phase IV, when conflict over a leader's authority occurred, community members resolved the issues within the new governance framework that had been created and debated in phases II and III. An event in 2006 offers evidence of such settlement; a group of developers used the general resolution process to recall the project leader when they became unsatisfied with his management of the boundaries of the project. This was the first leader recall in ten years, but now a formal process existed to handle challenges to a leader's authority. The recall did not pass the general resolution process, but it provides evidence of members' ability and willingness to work within the governance system they designed.

**Conclusion.** Under a *de facto* governance system (1993–97), Debian members had little shared basis of authority, and autocratic leadership failed. When Debian members entered a phase of designing governance (1997–99), they focused on both developing and limiting a positional basis of au-

thority that could be checked by democratic means. However, translating governance design into practice (1999–2003) introduced much interpretation and variation as to the level of authority appropriate for a project leader, since democracy encouraged variety. During this time, we identified five different active conceptions of leadership. Our multimethod approach revealed, however, that our informants' espoused preferences, in the design phase, for hands-off leadership was not supported by later systematic evaluation of leadership data.

After four years of implementation, a new conception of authority emerged in 2003: organization leadership. This year marked a transition to a shared conception of authority and a period of stabilized governance that survived a crucial test: the attempted recall of a leader. As Selznick explained, "Giving life to a constitution is partly a matter of achieving general consensus regarding proper ways of winning power and making laws" (1957: 6). By the end of 2006, the Debian community had achieved such a consensus. With statistical techniques, we now explore the actual behaviors associated with becoming a leader in this community in 2003, the start of the final phase of our study (stabilizing governance).

## PART II: DEDUCTIVE APPROACH

From our qualitative data, it would appear that developers engaged in organization building would be more likely to assume leadership than those who were more hands-off or more concerned with technical issues. To test this prediction, we modeled the results of one leader election four years after the system was designed (in 2003) with statistical analysis. This approach allowed us to move beyond analysis of discursive and rhetorical strategies used by leader candidates to an evaluation of the *behaviors* that were important to community member's conception of leadership.

Implicit in the literature on open source communities is the assumption that these communities operate in a meritocratic manner (e.g., Kogut & Metiu, 2001; Lee & Cole, 2003; Raymond, 1999). Thus, positions of authority should be allocated according to merit. However, it is not clear what merit really means in this context—technical contributions or organization building? Our interview data suggested that informants were aware of a status hierarchy based on technical contribution and observed the movement of people through it. One developer, commenting on the 2001 election, noted, "I have seen a lot of developers go from nobodies to being absolutely huge on the project.

You know for instance, [Dan Evans], one of the guys who is running for the DPL right now.”

The same informant who commented on Dan's evolution from a “nobody” to “huge” also noted that “there is no hierarchy to really say you know I am bigger than you so I will win. And it is a good thing and a bad thing. A lot of Debian is I say a meritocracy. You know if my code is better than your code, they will use it. Right?” Our informants' interpretation of meritocracy implied that the quality of one's code would be predictive of leadership. Thus, we hypothesized that contributors with greater technical contributions were more likely to become leaders. To operationalize this hypothesis, we explicitly considered both the amount of technical contributions as well as the impact of a developer's contributions on the community.

*Hypothesis 1a. The greater the amount of a community member's technical contribution, the higher the probability that the community member becomes a leader.*

*Hypothesis 1b. The greater the impact of a community member's technical contribution, the higher the probability that the community member becomes a leader.*

However, our qualitative data suggested that developers valued different strengths in a leader. Although some leadership candidates thought it was a “social position,” others thought it was a “merit badge rather than a position of trust.” In the 2002 election, one candidate noted that neither technical expertise nor social capital was sufficient: “The best candidate will not necessarily be the most popular or technically accomplished candidate, but the candidate who has the strongest grasp of what it will require to lead the Debian Project effectively.” This statement suggested that organization building might be as important as technical contribution but did not specify what activities would enable people to grasp “what it takes.” Our analysis of the organization building and organization leadership conceptions of authority in Part I of this research showed that such leaders tended to focus on enhancing communication and face-to-face interactions to better coordinate individual efforts within the community.

One former leader explained the importance of knowing people and their needs to foster internal coordination on the project. “Internally it's like coordination, and checking that everything is working well. So, for example, those people who were playing important roles, I've asked them, “What can I do for you?” For example, the security team was quite overloaded, so I was trying to find

some people for them . . . asking what is up? What can I do?” Although organization building has many facets, we realized that in a distributed global community, it would be impossible to do any organization building without posting messages online or meeting people face-to-face; these are necessary conditions for organization building in such a setting.

Three recent studies of the emergence of informal leaders in online communities showed that leaders posted more messages than nonleaders (Cassell, Huffaker, & Tversky, 2005; Misiolek & Heckman, 2005; Yoo & Alavi, 2004). Furthermore, the extent of “offline” interaction is likely to influence the amount of effort people put into building online communities (Butler, Sproull, Kiesler, & Kraut, forthcoming). In their study of Internet engineering task force (IETF) committees, Fleming and Waguespack (2007) discovered that, after an individual reached a certain level of technical contribution, collaborating with high-status contributors became a more important predictor of leadership. Thus, we propose that community members who communicate more with each other in both offline and online forums are more likely to be on the leadership team.

*Hypothesis 2a. The more a community member participates in online discussions, the higher the probability that a community member will become a leader.*

*Hypothesis 2b. The more people in the community one meets face to face, the higher the probability that a community member will become a leader.*

If we found that Hypotheses 1a and 1b were supported but Hypotheses 2a and 2b were not, then we could assume that community members favored a purely technical management approach to leadership. This result would contradict our findings on espoused conceptions of leadership in Part I of the study. If we found that Hypotheses 2a and 2b were supported, then we would have more systematic confirmatory evidence of members' evolving sense that organization building was a legitimate expansion of a project leader's authority.

## Methods

Our field research helped us identify data sources that could be used to predict the leadership behaviors that were important to the community. These data sources included the project's directory, developer database, bug-tracking database, package

popularity database, and identity authentication database. We collected data from these different sources and integrated them into one database, merging the observations on the basis of developers' names and e-mail addresses. With these data, we estimated a logit regression model to predict who became a member of the community's leadership team in 2003. Since the dependent variable was dichotomous, logistic regression was appropriate (Long, 1997).<sup>7</sup> Next, we explain how we operationalized each variable.

**Leadership team.** In measuring our dependent variable, we considered not only the position of project leader, but the community's full leadership team, which also includes the project secretary, software release coordinator, developer accounts manager, and members of the technical committee. Release coordinators and developer accounts manager are volunteers for indefinite appointments. Release coordinators shepherd the project to closure and developer account managers are responsible for allocating account rights to the project's code repository. As explained earlier, the technical committee (selected by the project leader) has the constitutional right to resolve technical disputes. We viewed these positions as providing critical leadership for the community, and our informants concurred. We coded these positions for the year 2003, the latest year for which a full set of data was available. In 2003, 831 developers were recognized as full-fledged members of the community, but we could only find complete data on 815 developers. Rather than imputing values for the missing data, we tested our hypotheses on these 815 observations. We also compared the missing observations with the ones used in the analysis and found that only one of the developers on the leadership team had missing data. The developers with missing data were primarily not leaders and, on average, maintained fewer packages, maintained packages with less impact, posted fewer messages online, and had met fewer developers than developers with complete data. Given the direction of our hypotheses, the missing data do not introduce meaningful bias in the analysis since, if anything, they make our estimates more conservative.

**Independent variables.** We operationalized technical contribution and organization-building activities with two different measures for each construct. Technical contribution was measured in terms of the number of software packages maintained by each developer and by the number of people that used these packages. Together, these measures capture both the quantity of a developer's contribution and his or her impact on the community. We operationalized organization-building activities as the amount of communication a developer engaged in on the project's central mailing list and as degree centrality, measured by the number of other developers the focal developer had met face-to-face (Wasserman & Faust, 1994). We describe these variables in more detail in the following paragraphs.

To measure each developer's *amount of technical contribution*, we collected data from the project's bug-tracking database on the number of software packages a developer maintained in 2002. A package is a discrete unit of software code that can be maintained independently of the rest of an operating system but has a standardized interface that allows integration with other packages. To maintain a package is to manage the receipt and review of code contributions from other contributors and to "package" these smaller contributions into a discrete module.

To measure *the impact of technical contribution* for each developer, we measured the popularity of the packages he or she maintained. Since early 2001, Debian users have been able to install a "popularity-contest package" that automatically calculates the number of people that download, install, and use a particular package. The popularity count is akin to a citation count in academic circles: it signifies how many people in the community use or rely upon your work. We computed the sum of the votes for all the packages maintained by developers as a measure of the criticality of their contributions for others. Although these data might be biased by the fact that not all developers had installed the "pop-con package" as of 2003, the sample of data we obtained provides a good proxy for package use. Since 2004, the results of the popularity contest have been used to determine which packages to include in Debian distribution CDs as well as to determine which packages to eliminate from the archives. We tested to see whether there was a substantial difference between the results of the popularity contest in 2003 and in more recent years (2004 and 2005) and found that the results were highly correlated, supporting our claim that even for 2003, this was a robust measure of a package's use by others.

---

<sup>7</sup> We also estimated the full model with the rare event logit procedure developed by King and Zeng (1999) to deal with cases in which the dependent variable assumed the value 1 in only a small number of cases (fewer than 1 percent). The model is significant and there is no substantial difference from the results we report with a traditional logistic regression in Table 5.



Since members are globally distributed, online mailing lists are the key infrastructure that enables communication and coordination in the Debian project. On the mailing lists, developers discuss changes to the code, coordination issues, and the future direction of the project. Debian has over 100 different mailing lists, focused on different sub-groups of developers and specific technical issues. To measure projectwide *online communication*, we focused on one list, "debian-devel," to which all developers subscribe. We downloaded all the postings to this list and measured each developer's number of e-mail postings as a first proxy for organization-building activity. Although discussion on debian-devel can also address technical issues, this mailing list is the one where most of the critical organizational discussions in the community are initiated, mainly because it contains the most members. Therefore, we believe that the number of e-mails a developer posted on this mailing list

provided a reasonable proxy for his or her propensity to communicate publicly with the rest of the community. Our qualitative data suggested that leaders interested in organization building would be more likely to make proposals on the most central list.

The other variable we used to measure a developer's involvement in the organizational life of the community was his or her *degree centrality*, a count of the number of other community members the developer had met as of January 1, 2003. Our analysis in Part I indicated that meeting other developers face-to-face required some effort and was associated with interest in building the organization. To measure who met whom, we took advantage of an interesting practice that the project adopted in 1994 to authenticate member identities. In 2000, this practice, which uses "public key cryptography," became a condition for becoming a Debian project member.

## EXHIBIT A

### Keysigning Party Invitation

Subject: Reminder for Signers, Signees

From: [xxxxxx@yyy](mailto:xxxxxx@yyy)

Date: 16 Feb 2001 14:53:56 -0800

So, people who need their key signed, remember to:

1. Send me your key by tonight, so I can assemble them into a nice keyring + printout for the signers.
2. Bring the following tomorrow:
  - \*Positive form of ID, like a driver's license or passport
  - \*Your own copy of your key fingerprint
3. Tomorrow I'll have:
  - \*Nice name + key fingerprint printouts for signers to use.
4. Signers should bring:
  - \*A pen or pencil to mark off the fingerprint printouts once they've checked IDs.

If you've never done a keysigning party before, here's how it works: the keysigner and keysignee meet. Keysigner checks the signee's ID against his/her printout, to make sure that he/she is talking to the right person. Then, signer makes a check next to the signee's name, to indicate "identified". Keysignee then reads out key fingerprint to keysigner. If the fingerprints match, and the ID is good, they make another check mark next to the name on their printout, indicating "key fingerprints match." ONLY IF BOTH CHECKMARKS EXISTS should the signer sign the key.

After the party, the next steps are:

- ✓ Every signer downloads the unsigned keyring from the Web URL I'll publish.
- ✓ Every signer signs the keys they verified (i.e., checked their print out).
- ✓ Every signer sends me their edited keyring with now-signed keys.
- ✓ I combine the signatures and make a new keyring, with all the signed stuff, and make that available on the Web.

Note that to be a Debian New Maintainer you only need one signature from one Debian maintainer. However, it's good for the Web of Trust for you to give and get lots of (valid) signatures.

Public key cryptography works with asymmetric “key” pairs (one public and one private). One party has a private key that is not shared and uploads a public key to a central “keyring file.” Other developers can then retrieve this public key and use it to confirm that messages sent from the first party are indeed from the sender. This procedure confirms that the content of any electronic message is from the avowed sender but does not verify the real-world identity of the sender. Thus, to make public key cryptography a useful means of authenticating identity, a real-world identity must be linked to a public key.

The Debian project does this through the practice of “key signing.” Developers sign each others’ keys to verify that they have met a person face-to-face. Members consider this to be an expression of trust: the signer meets the person, reviews government-issued identification, and then indicates belief that a particular public key belongs to the person who claims it via a digital signature. These digital signatures are uploaded to a central file where all developers can access them. Developers can get their keys signed in various ways. For example, the Debian Web site lists developers who want to have their keys signed and willing key signers around the world. This way, Debian developers who are traveling for other reasons can arrange to meet each other. Community members also attend “key-signing parties” where people meet to expressly sign keys, but these parties are often combined with other project activities. Exhibit A is a sample invitation to a Debian key-signing party.

Since each key signing is dated and requires a face-to-face meeting, these data indicate when project members met each other. We collected data from the Debian keyring, consisting of keys signed by dyads from 1994 (when the practice began) until January 1, 2003. We used these data to measure the degree centrality of each developer, or the number of community members each developer had met face-to-face. Since this measure only counts each single face-to-face meeting, we consider it a very conservative measure.

We controlled for a member’s *tenure* on the project (in months), taking the square root of the values because tenure did not follow a normal distribution. We also controlled for a member’s geographical *location*, which could only be established at the continent level.

## Results

Table 4 shows means, standard deviations, and correlations among the variables we considered. By comparing the means of community members on

the leadership team with those of the average developers (columns 1 and 3 in Table 4), we found that in 2003, the leadership team members had been in the community longer (on average, 72.4 versus 38.8 months)<sup>8</sup> but maintained approximately the same number of packages (10.9 vs. 9.3) as the average developers.<sup>9</sup> However, the packages that leaders maintained were used more widely than those of the average developers (411.6 vs. 34.8 users). Members of the leadership team were also much more active on the mailing list (52.5 vs. 3.9 messages posted) and had met many more other Debian developers (43.1 vs. 9.2). The descriptive statistics are thus in line with our hypotheses that developers who were making greater technical contributions (in terms of impact but not effort) and who were more engaged in organization building were more likely to become members of the leadership team.

Table 5 reports the logit coefficients and the odds ratios for the logistic regression models predicting membership in the leadership team in 2003.<sup>10</sup> In the final model, the independent variables were tenure, continent (with “Europe” as the default category), the number and popularity of code packages maintained (to measure technical contribution), and the amount of participation in online discussions and degree centrality (to measure organization building).

**Technical contribution.** The results reported in Table 5 suggest that the sheer amount of technical contribution, as measured by the number of software packages maintained, has a negative association with the likelihood of becoming a member of

<sup>8</sup> Many of the independent variables were transformed to address nonnormal distribution. In this section, in order to facilitate the comparisons between leaders and the rest of the developers, we report the descriptive statistics in the original scale. For the case of tenure, we took the power of the value in Table 5, since  $(\sqrt{x})^2 = x$ . Therefore:  $(8.52)^2 = 72.4$  months.

<sup>9</sup> For packages, usage and mailing list postings, we also report the descriptive statistics in the original scale, by taking the exponential of the value in the Table 5 since  $e^{\ln(x)} = x$ . Therefore:  $e^{2.39} = 10.9$ .

<sup>10</sup> To help the interpretation of the logit coefficients, we report the odds ratios in parentheses in the complete model (but only for coefficients that are statistically significant). Odds ratios were computed by taking the antilogarithm of the logit coefficient; thus, for the effect of degree centrality on leadership in 2003, we took the coefficient from model 5 in Table 6 and simply computed  $e^{0.055} = 1.056$ . Values exceeding 1 indicated an increased likelihood of becoming a member of the leadership team, and values less than 1 indicated decreased odds.

TABLE 4  
Descriptive Statistics and Correlations<sup>a</sup>

Variable	Mean for Leadership Team Members		Mean for All Observations		s.d.	Minimum	Maximum	1	2	3	4	5	6	7	8	
	Members	s.d.	Observations	s.d.												
1. Leadership team, 2003	1.00	0.00	0.01	0.10	0.00	0.00	1.00									
2. Tenure <sup>b</sup>	8.52	1.29	6.23	1.53	2.15	10.66	.15**									
3. North America	0.50	0.54	0.32	0.47	0.00	1.00	.04	.07*								
4. Europe	0.38	0.52	0.52	0.50	0.00	1.00	-.03	-.03	-.71**							
5. Other continent	0.13	0.35	0.16	0.37	0.00	1.00	.00	-.05	-.29**	-.46**						
6. Technical contribution <sup>c</sup>	2.39	1.55	2.23	1.47	0.00	6.93	.01	-.01	-.06	.04	.03					
7. Impact of technical contribution <sup>c</sup>	6.02	2.54	3.55	2.68	0.00	9.51	.09*	.01	-.00	.02	-.03	.73**				
8. Online communication <sup>c</sup>	3.96	2.12	1.35	1.68	0.00	6.43	.16**	-.05	.02	.02	-.05	.38**	.47**			
9. Degree centrality <sup>b</sup>	43.13	40.45	9.55	13.43	0.00	114.00	.25**	.14**	-.08*	.15**	-.10**	.24**	.21**	.33**		

<sup>a</sup>  $n = 815$ .

<sup>b</sup> Square root of the number of months. Measured for 2002.

<sup>c</sup> Log-transformed variable.

\*  $p < .05$

\*\*  $p < .01$

**TABLE 5**  
**Results of Logistic Regression Analysis for 2003 Leadership Team Membership<sup>a</sup>**

Variable	Model 1		Model 2		Model 3		Model 4		Model 5		Odds Ratio
	<i>b</i>	s.e.	<i>b</i>	s.e.	<i>b</i>	s.e.	<i>b</i>	s.e.	<i>b</i>	s.e.	
Tenure <sup>b</sup>	1.25**	(0.35)	1.25**	(0.33)	1.32**	(0.40)	1.55**	(0.56)	1.65**	(0.51)	5.22**
North America <sup>c</sup>	1.02	(0.84)	1.01	(0.83)	0.78	(0.85)	0.81	(0.89)	1.57	(0.71)	4.89*
Other	0.66	(1.26)	0.65	(1.23)	0.64	(1.24)	0.24	(1.26)	1.45	(1.20)	4.26
Technical contribution <sup>d</sup>			0.01	(0.20)	-0.72*	(0.34)	-0.98**	(0.35)	-1.34**	(0.45)	0.26**
Impact of technical contribution <sup>d</sup>					0.55**	(0.17)	0.25 <sup>†</sup>	(0.14)	0.41*	(0.17)	1.50*
Online communication <sup>d</sup>							0.83**	(0.32)	0.57	(0.41)	1.76
Degree centrality									0.06	(0.02)	1.06**
Intercept	-14.49**	(3.18)	-14.50**	(3.23)	-15.87**	(3.78)	-17.89**	(5.26)	-19.85**	(4.56)	
Wald $\chi^2$ ( <i>df</i> )	13.32 (3)		16.61 (4)		31.25 (5)		27.78 (6)		58.62 (7)		
Pseudo $R^2$	0.23		0.23		0.34		0.47		0.58		

<sup>a</sup>  $n = 815$ . Robust standard errors are reported for all models; odds ratios are reported for the complete model only.

<sup>b</sup> Square root of the number of months.

<sup>c</sup> Compared to developers located in Europe (default category).

<sup>d</sup> Log-transformed variable.

<sup>†</sup>  $p < .10$

\*  $p < .05$

\*\*  $p < .01$

the leadership team. Hypothesis 1a therefore finds no support. However, the impact of a developer's technical contribution, as measured by the number of users who installed the packages a developer maintained, is positively associated with the propensity to be on the leadership team. For every 100 users who installed a package, the package owner (developer) was 2.3 times more likely to become a member of the leadership team.<sup>11</sup> Hypothesis 1b is therefore supported.

**Organization building and leadership.** The results reported in models 4 and 5 in Table 5, do not provide strong supporting evidence for Hypothesis 2a. Although the variable online communication is positively associated with the likelihood of joining the leadership team in model 4, this effect loses statistical significance when degree centrality is included, in model 5. Therefore, we have to conclude that the sheer amount of communication does not lead to the attainment of leadership positions. We attribute our lack of results for online communication to the quality of our measure. Indeed, our ethnographic evidence leads us to conclude that communication with the community through the mailing list is an important antecedent

of the attainment of leadership positions, but only if other community members recognize this impact as significant. Our communication measure may conflate critical conversations with casual talk and, in future research, selected content-coding of the messages could help tease apart these effects.

To test Hypothesis 2b, we used the conservative measure of face-to-face meetings obtained from the cryptography keyring. In Table 5 (model 5), we found that meeting ten more developers in the community increased the likelihood of becoming a member of the leadership team by 56 percent (odds ratio = 1.06). Three mechanisms may be responsible for these results. One, Debian developers may be more likely to vote for someone they have met face-to-face, because they have developed a more trusting relationship with that person (e.g., Jarvenpaa, Knoll, & Leidner, 1998; Rocco, 1998; Weisband & Atwater, 1999). Two, candidates who have met more people face-to-face may be more proactive in creating coalitions or in seeking votes from other members (e.g., Thompson, 2005). Three, reciprocal concerns may be at play; members may be more likely to vote for developers they've met because they believe that, once in leadership roles, these developers will be able to help them in the future. Given the data with which we are working, we are not in a position to adjudicate among these mechanisms. However, all three mechanisms are consistent with our theoretical argument that both technical contribution and organization building are

<sup>11</sup> The natural logarithm of 100 is 4.6, and a one-unit increase in the log of users increases the odds by 46 percent; therefore, to interpret the meaning of the odds value we obtained for this measure, we computed  $4.6 \times 0.50 = 2.3$ .

required for a developer to attain a leadership position.

Most informants stated that face-to-face meetings were helpful in facilitating coordination on the project and that, according to one interviewee, they gave "a less impersonal feel to the names we see on mailing lists." As one leader who promised to raise funds to support travel said, "Meetings in real life . . . are very important. Future interactions are just different once you have gotten drunk with someone." However, others expressed reservations, fearing that "nonelectronic social relationships" could create or exacerbate inequalities in status. As one 2004 Debian leadership candidate's electoral platform said:

Debian should be a meritocracy, and *merit should be measured by one's contributions to the Project, not by a developer's nationality or non-electronic social relationships*. Debian was born on the Internet and could not have existed without it. The ease of electronic communication is our greatest asset, and the most effective leveler of inequities that we possess. We must not abandon this essential attribute in favor of provincialism.

Despite our informants' espoused preference for a "pure," meritocratic system, which was echoed in the literature, the attainment of leadership positions cannot be explained solely by an individual's technical contribution to the community. Members of the leadership team were also more likely to engage in organization-building activities. This evidence reinforces our earlier finding on the evolution of the conception of authority described in the inductive section of the paper.

## DISCUSSION

A large body of scholarship has examined the introduction of democratic or participatory mechanisms into bureaucratic organizations (Adler, Goldoftas, & Levine, 1997, 1999; Barker, 1993; Cappelli & Neumark, 2001). We have studied the reverse process: the introduction of bureaucratic mechanisms into community forms. This inquiry is important both because scholars increasingly recognize the viability of community forms of production as alternatives to markets and hierarchy (Adler, 2001; Bradach & Eccles, 1989; Ouchi, 1980; Powell, 1990) and because academic knowledge of how community forms manage production has rested upon conceptions of ideal types.

Although there has been a surge of interest in communities devoted to sharing knowledge, improving their work practice, sustaining their occupational identity, and exchanging social support,

scholars know very little about how these production communities govern themselves. For many years, scholars have acknowledged that even social systems that disregard authoritarian rule need some type of governance system to coordinate, manage scale, and sustain themselves (Harrison, 1960; March & Simon, 1958). However, little research has shed light on how communities engaged in production manage this tension. Our research bridges this gap by showing how one open source community designed and implemented a governance system.

### Blending Democratic and Bureaucratic Organizing Mechanisms

One reason community forms of production have not received as much attention as traditional capitalist forms (e.g., Perrow, 1991; Stern & Barley, 1996) is the former's inability to resolve problems of power, authority, and governance (Rothschild & Russell, 1986; Rothschild-Whitt, 1979, 1986). With ethnographic research, we showed how a production community designed a governance system that combines a constitutional basis of authority with democratic mechanisms to ensure control by the majority. We found that early in Debian's history, there was little shared basis of authority, which led to conflict and the recall of the second leader. Barnard (1938) argued that authority was most effective when it was unquestioned, reaching a taken-for-granted state. Only after the community created a constitution did the role of Debian project leader acquire Barnard's theorized taken-for-granted status—albeit in a limited fashion.

Debian's governance system incorporates mechanisms that reinforce both bureaucratic and democratic values. A rational basis of authority has been created by defining leadership positions and their associated rights, fulfilling the community's need to have someone with the authority to represent the project. However, democratic rule is reinforced by (1) requiring leaders to defer to the community, (2) granting leaders limited authority over technical matters, (3) allowing members to recall the leader's authority, and (4) creating a source of countervailing power. Positional authority is restricted to facilitating coordination and projectwide decisions, aiding in the resolution of conflict among individuals, and representing the project to outsiders. Thus, most authority remains laterally distributed. By combining elements of democratic and bureaucratic control, this system allows both regimes to coexist (e.g., Bradach & Eccles, 1989; Stark, 1999). Thus, we propose that to successfully introduce a bureaucratic basis of authority into a community

form, members must also design democratic mechanisms to delimit that basis of authority. However, our research suggests that these democratic mechanisms play a dual role; they not only ensure that the governance system represents the community's interests, but also provide the system with an adaptive mechanism. By blending democratic with bureaucratic mechanisms, this community was able to adapt its conceptualization of authority to the changing conceptions of its members.

### Reconceptualizing Meritocracy

We did not want to just understand how a production community develops a governance system, but also how such a system works in practice. We identified six different conceptions of authority and found that leaders embraced more organization-building behaviors over time. The tension between technical and organization-building contributions to the project was both apparent in the discourse of leader candidates and confirmed by analysis of which developers became leaders in 2003. A developer was more likely to become a leader when his or her technical contributions were widely used by other members. A developer's effort alone did not have the same effect. Contradicting a simplistic meritocratic explanation, developers who engaged in organization-building behaviors were more likely to become members of the leadership team. Thus, Debian may be a meritocracy, but merit is not measured solely by technical contribution. A prevalent assumption in research on open source software communities is that the contributions community members value the most are purely technical. Our study shows otherwise. In this community, the informal work of coordinating individual efforts and linking them to community goals became vital to leadership, particularly as the project matured.

This exploration of how meritocracy works in practice contributes directly to the debate, started by Parsons (1947) and Gouldner (1954) and extended most recently by Adler and Borys (1996), as to the precise relationship between meritocracy and bureaucracy. Are positions of authority the reward for technical competence? The answer turns on the definition of technical competence as the cornerstone of merit. If organizational competence becomes more important to community members than technical competence, we would argue that meritocracy still reigns. This research suggests that more nuanced thinking about meritocracy is needed. Any examination of meritocracy must develop a context-specific understanding of how

merit is conceptualized. Rather than questioning whether the outcomes of bureaucratic rule are in fact consistent with meritocracy, a more fruitful pursuit would be to examine how conceptions of merit evolve.

What is counterintuitive is that a community so wary of the effects of positional authority that its members actively limited it would, over time, prefer leaders who expanded their reach of authority. Members of this community were technical experts with many of the latest software development and coordination tools at their disposal. They also used many different types of media to manage their work. Yet coordinating their efforts and managing interdependence grew problematic without the help of people managing the organizational structure. This observation suggests that although technology may have changed the ability of groups such as Debian to coordinate efforts over space and time, even the most savvy online communities are not immune to well-known general principles of organizing (March & Simon, 1958; Michels, 1911; Ouchi, 1979).

### Limitations

This research is based on one in-depth case and so can only provide an existence proof of the design and implementation of a community governance system. The Debian community could be viewed as an extreme case, yet we would argue (with Starbuck [1993, 1998] and Siggelkow [2007]) that the study of extreme cases can help to resolve research deficiencies present in other methods, such as overgeneralization or neglect of individuality, complexity, and variety. Many organizations strive to differ from others and fill critical niches that would not be identified without extreme case research (Starbuck, 1993, 1998). The Debian community fits these criteria well. We deliberately chose a community that had evidence of leadership turnover. Many production communities initiated by strong founders (e.g., Linux and Wikipedia) have not experienced leadership turnover and thus have not institutionalized a basis for leadership. However, this does not preclude the relevance of our findings to founder-led communities, for these design choices may be in their future. We would expect our findings to apply to other types of distributed communities engaged in knowledge creation or production, but future comparative studies can confirm the extent to which these findings are generalizable.

## Implications for Traditional Organizations

Our findings on the importance of face-to-face interaction in predicting community leadership confirm earlier predictions that few communities exist purely in an online form and suggests the need to understand how hybrid forms function (Fiol & Connor, 2005; Griffith & Neale, 2001). Our findings are consistent with the results of experimental studies showing that group members like each other more when communicating face-to-face versus electronically (Weisband & Atwater, 1998) and that even a single face-to-face meeting can trigger more trustworthy behavior online (Rocco, 1998). For a leader to be elected by a community that restricts leadership authority, he or she must undoubtedly earn the trust of project members.

Perhaps members are more confident that their leader will not abuse the little authority entrusted to leaders if they have met face-to-face. These results may not be surprising to those who have proposed that face-to-face communication is the "gold standard" of communication (e.g., Nardi & Whittaker, 2002; Nohria & Eccles, 1992), but thus far the evidence supporting this argument has been primarily experimental. Our in-depth case study gives these findings ecological validity. Although external validity speaks to a study's ability to generalize, ecological validity speaks to a study's ability to approximate a real-life situation under study (Brewer, 2000; Shadish, Cook, & Campbell, 2002). These are independent constructs, but when ecological validity is improved, external validity often improves as well.

Our findings also have implications for autonomous work groups and their leaders in traditional organizations (Hackman, 1978). Autonomous work groups have responsibility for a whole task and the autonomy to make decisions on how they carry out their work (Alper, Tjosvold, & Law, 1998; Hackman, 1978). Recent research has identified leader behaviors that are critical to the success of such groups, such as building relationships, scouting for information, persuading constituents to offer their support (Druskat & Wheeler, 2003), and fostering team members' self-observation and evaluation (Manz & Sims, 1987). Our findings suggest that the leadership needs of autonomous work groups may evolve over time and that more longitudinal research is necessary to understand how and why.

Our findings also have implications for scholars studying the difference between task- and relationship-oriented leadership styles (Bons & Fiedler, 1976; Judge, Piccolo, & Ilies, 2004). Researchers have shown that individuals who emerge as leaders are likely to be extroverted, emotionally stable,

open to experience, and conscientious (Judge, Bono, Ilies, & Gerhardt, 2002), as well as high in cognitive ability (Taggar, Hackett, & Saha, 1999) and self-monitoring (Day, Schleicher, Unckless, & Hiller, 2006). With few exceptions, this research has taken an individual differences approach (Ilies, Gerhardt, & Le, 2004) without providing insight as to how a leader's focus on either task (e.g., technical) or relational (e.g., organizational) concerns may evolve and adapt with a team's or project's growing experience. Our study suggests that there is a contingent relationship between the nature of leadership and the scale and maturity of a community. Can individual leaders adapt their leadership styles? Although considerable research has examined what predicts leadership effectiveness (Judge et al., 2004), little research has examined what predicts leadership ascension.

This research explains how governance and leadership emerged in a distributed community having no obvious basis of authority—a context that is relevant to the management of professional expertise in many contexts. As Scott pointed out, hierarchical systems have been increasingly "giving way to more decentralized and horizontal systems, particularly among organizations in the newer industries" (2004: 12). We would predict that the more information and knowledge are distributed, the more likely it is that democratic approaches will be appropriate. Future research should examine how democratic decision mechanisms can be integrated with more traditional forms of bureaucratic and professional or occupational control (Van Maanen & Barley, 1984). As porous and dynamic organizational boundaries (Santos & Eisenhardt, 2005) enable a growing body of constituents and specialists relevant to decision making, governance systems that support a community of distributed experts are likely to increase in importance.

## REFERENCES

- Adler, P. S. 2001. Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, 12: 215–234.
- Adler, P. S., & Borys, B. 1996. Two types of bureaucracies: Enabling and coercive. *Administrative Science Quarterly*, 41: 61–89.
- Adler, P. S., Goldoftas, B., & Levine, D. 1997. Ergonomics, employee involvement, and the Toyota production system: A case study of NUMMI's 1993 model introduction. *Industrial and Labor Relations Review*, 50: 416–437.
- Adler, P. S., Goldoftas, B., & Levine, D. 1999. Flexibility

- versus efficiency? A case study of model changeovers in the Toyota production system. *Organization Science*, 10: 43–68.
- Adler, P. S., & Heckscher, C. 2006. Towards collaborative community. In C. Heckscher & P. S. Adler (Eds.), *The firm as a collaborative community: Reconstructing trust in the knowledge economy*: 11–106. New York: Oxford University Press.
- Alper, S., Tjosvold, D., & Law, K. S. 1998. Interdependence and controversy in group decision making: Antecedents to effective self-managing teams. *Organizational Behavior and Human Decision Processes*, 74: 33–52.
- Armstrong, A., & Hagell, J. 1996. The real value of on-line communities. *Harvard Business Review*, 74(3): 134–141.
- Barker, J. R. 1993. Tightening the iron cage: Concertive control in self-managing teams. *Administrative Science Quarterly*, 38: 408–437.
- Barnard, C. I. 1938. *The functions of the executive*. Cambridge, MA: Harvard University Press.
- Bechky, B. 2003. Object lessons: Workplace artifacts as representations of occupational jurisdiction. *American Journal of Sociology*, 109: 720–52.
- Blau, P. M. 1955. *The dynamics of bureaucracy*. Chicago: University of Chicago Press.
- Bons, P. M., & Fiedler, F. E. 1976. Changes in organizational leadership and the behavior of relationship- and task-motivated leaders. *Administrative Science Quarterly*, 21: 453–473.
- Bradach, J. R., & Eccles, R. 1989. Markets versus hierarchies: From ideal types to plural forms. W. R. Scott (Ed.), *Annual review of sociology*, vol. 15: 97–118.
- Brewer, M. 2000. Research design and issues of validity. In H. Reis & C. Judd (Eds.), *Handbook of research methods in social and personality psychology*. Cambridge, U.K.: Cambridge University Press.
- Brown, J. S., & Duguid, P. 1991. Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization Science*, 2: 40–57.
- Brown, J. S., & Duguid, P. 2000. *The social life of information*. Boston: Harvard Business School Press.
- Brown, J. S., & Duguid, P. 2001. Knowledge and organization: A social-practice perspective. *Organization Science*, 12: 198–213.
- Butler, B. S., Sproull, L., Kiesler, S., & Kraut, R. Forthcoming. Community effort in online groups: Who does the work and why? In S. A. L. Weisband (Ed.), *Leadership at a distance*. Erlbaum.
- Campbell, D. T., & Fiske, D. W. 1959. Convergent and discriminant validation by the multi-trait-multi-method matrix. *Psychological Bulletin*, 56: 81–105.
- Cappelli, P., & Neumark, D. 2001. Do “high-performance” work practices improve establishment-level outcomes? *Industrial and Labor Relations Review*, 54: 737–775.
- Casadesus-Masanell, R., & Ghemawat, P. 2006. Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Science*, 52: 1072–1084.
- Cassell, J., Huffaker, D., & Tversky, D. 2005. *How to win a world election: Emergent leadership in an international online community*. Paper presented at the Second Communities and Technologies Conference, Milan.
- Chen, K., & Mahoney, S. 2007. *The selective synthesis of competing logics*. Working paper, Harvard Business School, Boston.
- Coleman, E. G. 2005. *Three ethical moments in Debian*. Available at SSRN: <http://ssrn.com/abstract=805287>
- Coleman, J. S. 1970. Social inventions. *Social Forces*, 49: 163–173.
- Coleman, J. S. 1974. *Power and the structure of society*. New York: Norton.
- Coleman, J. S. 1980. Authority systems. *Public Opinion Quarterly*, 44: 143–163.
- Coleman, J. S. 1993. 1992 presidential address: The rational reconstruction of society: *American Sociological Review*, 58: 1–15.
- Cummings, J., Sproull, L., & Kiesler, S. 2002. Beyond hearing: where real-world and online support meet. *Group Dynamics*, 6(1): 78–88.
- Dalle, J. M., & Jullien, N. 2003. “Libre” software: Turning fads into institutions? *Research Policy*, 32: 1–11.
- Day, D. V., Schleicher, D. J., Unckless, A. L., & Hiller, N. J. 2002. Self-monitoring personality at work: A meta-analytic investigation of construct validity. *Journal of Applied Psychology*, 87: 390–401.
- DiMaggio, P. J., & Anheier, H. 1990. The sociology of nonprofit organizations and sectors. In W. R. Scott & J. Blake (Eds.), *Annual review of sociology*, vol. 16: 137–159. Palo Alto, CA: Annual Reviews.
- DiMaggio, P. J., & Powell, W. 1983. The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, 48: 147–160.
- Drucker, P. F. 2001. *The essential Drucker*. New York: HarperCollins
- Druskat, V. U., & Wheeler, J. V. 2003. Managing from the boundary: The effective leadership of self-managing work teams. *Academy of Management Journal*, 46: 435–457.
- Edmondson, A., & McManus, S. E. 2007. Methodological fit in management field research. *Academy of Management Review*, 32: In press.
- Etzioni, A. 1959. Authority structure and organizational effectiveness. *Administrative Science Quarterly*, 4: 43–67.
- Fayard, A. L., DeSanctis, G. 2007. *Enacting language*



- games: The emergence of collective identity in on-line forums.** Working paper, NY Polytechnic University, New York.
- Fiol, C. M., & O'Connor, E. J. 2005. Identification in face-to-face, hybrid, and pure virtual teams: Untangling the contradictions. *Organization Science*, 16: 19–32.
- Fleming, L., & Waguespack, D. 2007. Brokerage, boundary spanning and leadership in open innovation communities. *Organization Science*, 18: 165–180.
- Fligstein, N., 1990. *The transformation of corporate control*. Cambridge, MA: Harvard University Press.
- Fosfuri, A., Giarratana, M. S., & Luzzi, A. In press. The penguin has entered the building: The commercialization of open source software products. *Organization Science*.
- Glaser, B. G., & Strauss, A. L. 1999. *The discovery of grounded theory: Strategies for qualitative research*. Hawthorne, NY: Aldine de Gruyter.
- Gouldner, A. 1954. *Patterns of industrial bureaucracy*. New York: Free Press.
- Griffith, T. L., & Neale, M. A. 2001. Information processing in traditional, hybrid, and virtual teams: From nascent knowledge to transitive memory. In B. M. Staw & R. I. Sutton (Eds.), *Research in organizational behavior*, vol. 23: 379–421. Stamford, CT: JAI.
- Hackman, J. R. 1978. The design of self-managing work groups. In B. King, S. S. Streufert, & F. E. Fiedler (Eds.), *Managerial control and organizational democracy*. Washington DC: Winston & Son.
- Hannan, M., & Freeman, J. 1989. *Organizational ecology*. Cambridge, MA: Harvard University Press.
- Hargadon, A., & Bechky, B. 2006. When collections of creatives become creative collectives: A field study of problem solving at work. *Organization Science* 17: 484–500.
- Harrison, P. M. 1960. Weber's categories of authority and voluntary associations. *American Sociological Review*, 25: 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.
- Heath, C., & Sitkin, S. 2001. Big-B versus big-O: What is organizational about organizational behavior? *Journal of Organizational Behavior*, 22: 43–58.
- 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: 1159–1177.
- Ilies, R., Gerhardt, M. W., & Le, H. 2004. Individual differences in leadership emergence: Integrating meta-analytic findings and behavioral genetics estimates. *International Journal of Selection and Assessment*, 12: 207–219.
- Ingram, P., & Simons, T. 2000. State formation, ideological competition, and the ecology of Israeli workers' cooperatives. *Administrative Science Quarterly*, 45: 25–53.
- Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. 1998. Is there anybody out there? The implications of trust in global virtual teams. *Journal of Management Information Systems*, 14(4): 29–64.
- Johnson, A. G., & Whyte, W. F. 1977. The Mondragon system of worker production cooperatives. *Industrial and Labor Relations Review*, 31: 18–30.
- Judge, T. A., Bono, J. E., Ilies, R., & Gerhardt, M. W. 2002. Personality and leadership: A qualitative and quantitative review. *Journal of Applied Psychology*, 87: 765–780.
- Judge, T. A., Piccolo, R. F., & Ilies, R. 2004. The forgotten ones? The validity of consideration and initiating structure in leadership research. *Journal of Applied Psychology*, 89: 36–51.
- Kanter, R. M. 1968. Commitment and social organization: A study of commitment mechanisms in utopian communities. *American Sociological Review*, 33: 499–517.
- Kieser, A. 1989. Organizational, institutional, and societal evolution: Medieval craft guilds and the genesis of formal organizations. *Administrative Science Quarterly*, 34: 540–564.
- King, G., & Zeng, L. 1999. *Logistic regression in rare events data*. Working paper, Department of Government, Harvard University. Available from <http://gking.harvard.edu>.
- Kogut, B., & Metiu, A. 2001. Open-source software development and distributed innovation. *Oxford Review of Economic Policy*, 17: 248–264.
- Kollock, P. 1998. Social dilemmas: The anatomy of cooperation. In J. Hagan & K. S. Cook (Eds.), *Annual review of sociology*, vol. 24: 183–214. Palo Alto, CA: Annual Reviews.
- 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*: 3–22. Cambridge, MA: MIT Press.
- Lave, J., & Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge, U.K.: Cambridge University Press.
- Lee, G., & Cole, R. E. 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science*, 14: 633–649.
- Lerner, J., & Tirole, J. 2002. The simple economics of open source. *Journal of Industrial Economics*, 52: 197–234.

- Long, J. S. 1997. *Regression models for categorical and limited dependent variables*. Thousand Oaks, CA: Sage.
- Manz, C. C., & Sims, H. P. 1987. Leading workers to lead themselves: The external leadership of self-managing work teams. *Administrative Science Quarterly*, 32: 106–129.
- March, J. G., & Simon, H. A. 1958. *Organizations*. New York: Wiley.
- Marsden, P. 2005. The sociology of James S. Coleman. In D. S. Massey & K. S. Cook (Eds.), *Annual review of sociology*, vol. 31: 1–24. Palo Alto, CA: Annual Reviews.
- Merton, R. K. 1940. Bureaucratic structure and personality. *Social Forces*, 18: 560–568.
- Michels, R. 1911. *Political parties: A sociological study of the oligarchical tendencies of modern democracy*. New York: Free Press.
- Misiolek, N., & Heckman, R. 2005. *Patterns of emergent leadership in virtual teams*. Paper presented at the 38th Hawaii International Conference on System Sciences.
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11: 309–346.
- Moon, J. Y., & Sproull, L. 2002. Essence of distributed work: The case of the Linux kernel. In P. Hinds & S. Kiesler (Eds.), *Distributed work: New research on working across distance using technology*: 381–404. Cambridge, MA: MIT Press.
- Nardi, B., & Whittaker, S. 2002. The place of face to face communication in distributed work. In P. Hinds & S. Kiesler (Eds.), *Distributed work: New research on working across distance using technology*: 83–110. Cambridge, MA: MIT Press.
- Netcraft. 2005. *Strong growth for Debian*. Retrieved from [http://news.netcraft.com/archives/2005/12/05/strong\\_growth\\_for\\_debian.html](http://news.netcraft.com/archives/2005/12/05/strong_growth_for_debian.html), September 14, 1906.
- Nohria, N., & Eccles, R. G. 1992. Face-to-face: Making network organizations work. In N. Nohria & R. G. Eccles (Eds.), *Networks and organizations: Structure, form, and action*: 288–308. Boston: Harvard Business School Press.
- O'Mahony, S. 2003. Guarding the Commons: How community managed projects protect their work. *Research Policy*, 32: 1179–1198.
- O'Mahony, S., & Ferraro, F. Forthcoming. Managing the boundary of an open project. In W. Powell & J. Padgett (Eds.), *Market emergence and transformation*.
- Orr, J. 1996. *Talking about machines: An ethnography of a modern job*. Ithaca, NY: ILR Press.
- Ouchi, W. 1979. A conceptual framework for the design of organizational control mechanisms. *Management Science*, 25: 833–848.
- Ouchi, W. 1980. Markets, bureaucracies, and clans. *Administrative Science Quarterly*, 25: 125–141.
- Parks, M. R., & Floyd, K. 1996. Making friends in cyberspace. *Journal of Communication*, 46(1): 80–97.
- Parks, M. R., & Roberts, L. D. 1997. Making MOOic: The development of personal relationships on-line and a comparison to their offline counterparts. *Journal of Social and Personal Relationships*, 15: 517–537.
- Parsons, T. 1947. *Theory of social and economic organization*. New York: Oxford University Press.
- Perrow, C. 1991. A society of organizations. *Theory and Society*, 10: 725–762.
- Pickering, J. M., & King, J. L. 1995. Hardwiring weak ties: Interorganizational computer-mediated communication, occupational communities, and organizational change. *Organization Science*, 6: 479–486.
- Powell, W. W. 1990. Neither market nor hierarchy: Network forms of organization. In B. M. Staw & L. L. Cummings (Eds.), *Research in organizational behavior*, vol. 12: 295–336. Greenwich, CT: JAI Press.
- Powell, W. W., & Snellman, K. 2004. The knowledge economy. In K. S. Cook & J. Hagan (Eds.), *Annual review of sociology*, vol. 30: 199–220. Palo Alto, CA: Annual Reviews.
- Rao, H. 1998. Caveat emptor: The construction of non-profit consumer watchdog organizations. *American Journal of Sociology*, 103: 912–961.
- Raymond, E. S. 1999. *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly & Associates.
- Rheingold, H. 2000. *The virtual community: Homesteading on the electronic frontier*. Cambridge, MA: MIT Press.
- Rocco, E. 1998. *Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact* (Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Los Angeles). Retrieved September 14, 2006 from <http://delivery.acm.org/10.1145/280000/274711/p496-rocco.pdf?key1=274711&key2=3679328511&coll=&dl=acm&CFID=15151515&CFTOKEN=6184618>.
- Romanelli, E. 1991. The evolution of new organizational forms. In W. R. Scott & J. Blake (Eds.), *Annual review of sociology*, vol. 17: 79–103. Palo Alto, CA: Annual Reviews.
- Rothschild, J., & Russell, R. 1986. Alternatives to bureaucracy: Democratic participation in the economy. In J. F. Short (Ed.), *Annual review of sociology*, vol. 12: 307–328. Palo Alto, CA: Annual Reviews.
- Rothschild, J., & Whitt, J. A. 1986. *The cooperative workplace: Potentials and dilemmas of organiza-*

- tional democracy and participation.** New York: Cambridge University Press.
- Rothschild-Whitt, J. 1979. The collectivist organization: An alternative to rational-bureaucratic models. *American Sociological Review*, 44: 509–527.
- Santos, F. M., & Eisenhardt, K. M. 2005. Organizational boundaries and theories of organizing *Organization Science*, 16: 491–508.
- Satow, R. L. 1975. Value-rational authority and professional organizations: Weber's missing type. *Administrative Science Quarterly*, 20: 526–531.
- Sawhney, M., & Prandelli, E. 2000. Communities of creation: Managing distributed innovation in turbulent markets. *California Management Review*, 42(4): 24–54.
- Scott, W. R. 2004. Reflections on a half-century of organizational sociology. In K. S. Cook & J. Hagan (Eds.), *Annual review of sociology*, 30: 1–21. Palo Alto, CA: Annual Reviews.
- Seidel, M. L., & Stewart, K. J. 2001. *The C-form: Emergence of a post-industrial organizational form.* Paper presented at the annual meeting of the Academy of Management, Washington, DC.
- Selznick, P. 1957. *Leadership in administration: A sociological interpretation.* Berkeley: University of California Press.
- Shadish, W., Cook, T., & Campbell, D. 2002. *Experimental and quasi-experimental designs for generalized causal inference.* Boston: Houghton-Mifflin.
- Shah, S. K. 2005. Open beyond software. In C. Dibona, D. Cooper, & M. Stone (Eds.), *Open sources 2*: 339–360. Sebastopol, CA: O'Reilly Media.
- Shah, S. K. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, 52: 1000–1014.
- Siggelkow, N. 2007. Persuasion with case studies. *Academy of Management Journal*, 50: 20–24.
- Simons, T., & Ingram, P. 1997. Organization and ideology: Kibbutzim and hired labor, 1951–1965. *Administrative Science Quarterly*, 42: 784–814.
- Smith, M., & Kollock, P. 1999. *Communities in cyberspace.* London: Routledge.
- Starbuck, W. H. 1993. Keeping a butterfly and an elephant in a house of cards: The elements of exceptional success. *Journal of Management Studies*, 30: 885–921.
- Starbuck, W. H. 1998. *Learning from extreme cases.* Retrieved December 16, 2005, from <http://pages.stern.nyu.edu/~wstarbuc/extreweb/Extreweb.htm>.
- Stark, D. 1999. Hierarchy: Distributing intelligence and organizing diversity. In J. Clippinger (Ed.), *The biology of business: Decoding the natural laws of enterprise*: 153–179. San Francisco: Jossey-Bass.
- Stern, R. N., & Barley, S. R. 1996. Organizations and social systems: Organization theory's neglected mandate. *Administrative Science Quarterly*, 41: 146–162.
- Stewart, D. 2005. Social status in an open source software community. *American Sociological Review*, 70: 823–842.
- Stinchcombe, A. L. 1959. Bureaucratic and craft administration of production: A comparative study. *Administrative Science Quarterly*, 4: 168–187.
- Stinchcombe, A. L. 1965. Social structure and organizations. In J. G. March (Ed.), *Handbook of organizations*: 142–193. Chicago: Rand McNally.
- Strauss, A., & Corbin, J. 1990. *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Thousand Oaks, CA: Sage.
- Swidler, A. 1979. *Organization without authority: Dilemmas of social control in free schools.* Cambridge, MA: Harvard University Press.
- Taggar, S., Hackett, R., & Saha, S. 1999. Leadership emergence in autonomous work teams: Antecedents and outcomes. *Personnel Psychology*, 52: 899–926.
- Thompson, J. D. 1967. *Organizations in action.* New York: McGraw-Hill.
- Van Maanen, J., & Barley, S. R. 1984. Occupational communities: Culture and control in organizations. In B. M. Staw & L. L. Cummings (Eds.), *Research in organizational behavior*, vol. 6: 287–365. Greenwich, CT: JAI Press.
- 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: 209–223.
- 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.
- Wasserman, S., & Faust, K. 1994. *Social network analysis: Methods and applications.* New York: Cambridge University Press.
- Weber, M. 1978. *Economy and society.* Berkeley: University of California Press.
- Weick, K. E. 1979. *The social psychology of organizing* (2d ed.). Reading, MA: Addison-Wesley.
- Weisband, S., & Atwater, L. 1999. Evaluating self and others in electronic and face-to-face groups. *Journal of Applied Psychology*, 84: 632–639.
- Wellman, B., & Gulia, M. 1999. Net surfers don't ride alone: Virtual communities as communities. In B. Wellman (Ed.), *Networks in the global village*: 167–194. Boulder, CO: Westview.
- Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., & Haythornthwaite, C. 1996. Computer networks

- as social networks: Collaborative work, telework, and virtual community. In J. Hagan & K. S. Cook (Eds.), *Annual review of sociology*, vol. 22: 213–238. Palo Alto, CA: Annual Reviews.
- Wenger, E. 1998. *Communities of practice: Learning, meaning, and identity*. Cambridge, U.K.: Cambridge University Press.
- Wenger, E. 2000. Communities of practice and social learning systems. *Organization*, 7: 225–246.
- Whyte, W. F., & Whyte, K. K. 1988. *Making Mondragon: The growth and dynamics of the worker cooperative complex*. Ithaca, NY: ILR Press.
- Willer, D. E. 1967. Max Weber's missing authority type. *Sociological Inquiry*, 37: 231–239.
- Williams, R. L., & Cothrel, J. 2000. Four smart ways to run online communities. *Sloan Management Review*, 41(4): 81–91.
- Yin, R. K. 1994. *Case study research: Design and methods* (2nd ed.). Beverly Hills, CA: Sage.
- Yoo, Y., & Alavi, M. 2004. Emergent leadership in virtual teams: What do emergent leaders do? *Information and Organization*, 14(1): 27–28.
- Zucker, L. G. 1977. The role of institutionalization in cultural persistence. *American Sociological Review*, 42: 726–743.



**Siobhán O'Mahony** ([somahony@ucdavis.edu](mailto:somahony@ucdavis.edu)) is an assistant professor at the Graduate School of Management at the University of California, Davis. She received her Ph.D. in management science and engineering from Stanford University. Her research examines organizing processes in community forms and how common information resources are used to further innovation.

**Fabrizio Ferraro** ([fferraro@iese.edu](mailto:fferraro@iese.edu)) is an assistant professor of general management at IESE Business School—University of Navarra. He received his Ph.D. in management science and engineering from Stanford University. His current research interests include the role of economics, language, and ideology in shaping management practices, the formation of social networks in entrepreneurship, and the role of networks in governance processes.



Copyright of *Academy of Management Journal* is the property of *Academy of Management* and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.