20 Nonprofit Foundations and Their Role in Community-Firm Software Collaboration

Siobhán O’Mahony

Contributors to community-managed projects have created nonprofit foundations, despite the fact that such formal structures are an anathema to the hacker ethos of technical autonomy and meritocratic decision making. The technical organizations that emerged since the federal government privatized the Internet may have partially influenced the design of these foundations, but some features are the unique product of managing community software in a commodity world. One thing that stands out from either the early Internet working groups or the typical corporate standard-setting bodies of the past is the role of nonprofit software foundations in enabling collaboration between a community of individuals and corporate actors.

Many people may be surprised by the emergence of open source software foundations, because incorporation requires a degree of formality that may seem inconsistent with the portrayal of open source contributors as guided only by their own desires (e.g., Raymond 2001). As many who contribute to such projects know, most open source projects manage their efforts through normative control on discussion mailing lists and instant message channels. Minimal constraints define programming formats and protocols, but not the technical direction of the project. Despite collectively determined code format and check-in and -out procedures, the technical direction of most community projects is typically the product of negotiation among a small group of core developers. How these forms of control shape the architecture and evolution of software is still not well understood. What we do know is that the combination of peer-based normative control and collectively determined procedural standards have been effective enough to permit commercial-grade software to be developed without the efficiency benefits typically equated with bureaucratic controls.

Many programmers who contribute to community-managed projects identify with the hacker community. As readers of this book do not need to be told, the hacker community is not one that embraces centralized modes of governance. The hacker ethos, as articulated by those who know it best (Levy 1994; Raymond 2001; Pavlicek 2000; Himanen 2001), values the intrinsic satisfaction from solving technical challenges, as well as truth, independence, and individual autonomy. "A happy programmer is one who is neither under-utilized nor weighed down with ill-formulated goals and stressful process friction" (Raymond 2001, 74). This is particularly true for projects that rely on volunteer contributors, for volunteers are more likely to be motivated by intrinsic reasons and thus less likely to welcome formal organizing mechanisms (Lakhani and Wolf 2003; Chap 1, this volume; Butler et al. 2002). Indeed, these studies show that many volunteer contributors to software projects do so in order to learn, hone their skills, solve a technical challenge, "make their mark," or improve their careers (Lakhani and Wolf 2003; Chap 1, this volume; Lerner and Tirole 2002; Hann et al. 2002).

The literature on the hacker ethos is not inconsistent with research on the motivations, preferences, and occupational identities shared by engineers and other technical workers. Organizations have long struggled with how to manage people who are more likely to be motivated by the work itself, less likely to want to leave their work for management positions, and less likely to respect authority that is not rooted in technical competence (Ritti 1971, 1998; Whalley 1986; Whalley and Barley 1997). The concept of a dual career ladder was essentially an attempt to integrate the ethos of the engineer within an organizational framework (Allen and Katz 1986). While there is variance in the motivations of contributors to free and open source software, underlying this divergence is a shared belief in the value of challenging work, technical autonomy, self-management, and freedom from a positional basis of power. Thus, even programmers that do not explicitly identify with the hacker community or the ideals of the open source and free software movements may hold beliefs about the forms of organization they prefer. This is important, because as the open source and free software community becomes more diverse in attitude and affiliation, fewer elements of the hacker ethos may be as widely shared. The occupational identity that is common to programmers who prefer the community development model may provide a source of organizational resiliency that extends beyond individual motivations or political affiliations (Van Mannen and Barley 1984).

The Organizational and Legal Dilemma

Given these preferences, why would community-managed projects create nonprofit foundations? What role, if any, do these foundations have in fostering collaboration between communities and firms? The commoditization of open source and free software created new opportunities for many projects, but also created new dilemmas. Managing community software in a commodity world brought new challenges such as how to treat corporate contributions of code, how to communicate to the press the difference between a project and a company, and how to enforce a community's terms for software modification and distribution within a user and developer population that was growing not only larger but also more diverse in its attitudes toward commercial software. With growth in market share and enhanced media and industry attention came a degree of exposure that even the most mature projects had not heretofore experienced. Greater public exposure elicited new areas of vulnerability. With more users of the software, there was greater probability that liability issues could arise and, as an unincorporated entity, fewer protections to prevent volunteers from individual liability.

This is because communities are not legal actors. Community-managed software projects are open source or free software projects initiated and managed by a distributed group of individuals who do not share a common employer.[1] Contributors may be associated with the free software or open source social movements, unaffiliated or sponsored by a firm. Most importantly, contributors are not employees of the project and project relations are independent of employment relations.[2] Community mailing lists are well-bounded: membership is clear and members share distinct norms that guide list behaviors and programming protocols. Yet they have few legal rights.

The lack of legal rights granted to online communities became a real problem when several leaders within the community realized that they might have difficulty protecting the "Linux" and "open source" terms and concepts. After the open source term was created in early 1998, firms and members of the press sometimes used the term "open source" in ways that extended beyond what creators of the term had intended. Companies were not just downloading free software for their own use, they were bundling it with other software and selling it in combination with hardware and services. While long-term contributors to free and open source software were delighted to see their work proliferate, firms developing Linux and other open source products and services sometimes created confusion as to what

these terms represented, and as to where community work stopped and corporate work began.

In 1999, the small group of community leaders who created the open source term found the concept too common to earn trademark rights. The leaders announced, "We have discovered that there is virtually no chance that the U.S. Patent and Trademark Office would register the mark 'open source'; the mark is too descriptive. Ironically, we were partly a victim of our own success in bringing the 'open source' concept into the mainstream. So 'Open Source' is not and cannot become a trademark" (OSI Announcement, June 16, 1999).

These leaders created a nonprofit organization, the Open Source Initiative (OSI), to ensure that the open source concept would not be misrepresented as the concept grew in commercial popularity.[3] Without legal rights, community-managed projects not only had trouble defending their concepts and code, but also were unable to form contracts and legal agreements as a single entity. One Fortune 100 executive faced with structuring a formal relationship with the Apache Project in the late 1990s, noted this unusual state by asking: "How do I make a deal with a Web page?" Collaboration between a firm and a community-managed project was a relatively foreign idea and there was little precedent to help make it happen.

Organizing Options and Models

What organizational options are available for open source and free software programmers who want to move beyond the status of "a Web page" and at the same time avoid forming a firm? Cooperatives are one legal form with communal norms and values. Producer cooperatives pay their members a set price for their contributions and apportion dividends pro rata to their members yearly. Consumer cooperatives pay earnings to members based on the amounts that members spend, as opposed to the amounts they sell (Hansmann 1996: 13–14). Both of these forms redistribute profits to their members, which is incompatible with the goals of community-managed software projects. What unites software communities is the goal of producing open source and free software (Williams 2002; Pavlicek 2000; Raymond 2001) and perhaps a shared culture and occupational identity (van Mannen and Barley 1984). What does not bind the community is the desire to earn a profit as a direct product of their collective work.[4]

Other organizing possibilities include forming a consortium, alliance, or task force, as technical communities critical to the development of the

Internet have done. Indeed, the open source and free software communities are not the first technical communities to wrestle with the problem of creating a form that can exist independent of any one person. The U.S. government's privatization of the Internet led to the creation of professional working groups and technical societies that were familiar to leaders of community-managed software projects. Internet standards work that was once the responsibility of Defense Advanced Research Projects Agency (DARPA) has, since 1986, been delegated to the Internet Engineering Task Force (IETF). The IETF calls itself a "loosely self-organized group of people [volunteers] who contribute to the engineering and evolution of Internet technologies" ("The Tao of IETF" 2001). The IETF differs from corporate-led standard-setting bodies in that it maintains no specific membership or dues requirements. Any interested individual can attend a meeting, join a working group mailing list, or contribute to a project. Members do not represent their affiliated organizations, but act on their own capacity.

On the other hand, the World Wide Web Consortium (W3C) is a consortium of organizations: individuals cannot become members.[5] Three universities on different continents host the W3C.[6] This design was explicitly intended to preserve pluralism and prevent the emergence of a United States–centric World Wide Web (WWW) (Berners-Lee, Fishetti, and Dertouzous 2000). A third organization responsible for ensuring that all Internet domain names will be universally resolvable, the Internet Corporation for Assigned Names and Numbers (ICANN), has not been able to successfully integrate individual and organizational representation.[7] ICANN has declared that its structure is incomplete, its funds inadequate, and is currently pursuing major reform efforts. It is an unlikely organizational model.[8] Neither the W3C nor the IETF are incorporated, but both have incorporated hosts. The Internet Society (ISOC), a nonprofit professional membership association that allows both individuals and organizations to be members, hosts the IETF. Many of the technical working groups under the ISOC charter have well-established processes for receiving, reviewing, and integrating comments on technical standards that stem from early government-sponsored efforts. These processes, as well as IETF's focus on individuals as members, may have influenced the type of form leaders in the open source software that community leaders wanted to create.

The first foundation for community-managed software, the Free Software Foundation (FSF), was created even before the IETF, ISOC, W3C, or ICANN organizations existed. Table 20.1 shows when organizations representing

Table 20.1
Institutions founded to represent technical communities

Date	Organization founded	Mission
1979	*ICCB-DARPA*	Develop TCP/IP Protocal Suite
1983	*Internet Architecture Board*	Provide oversight of archictecture of Internet, integrate working group activities
1985	Free Software Foundation (FSF)	Dedicated to promoting computer users' right to use, study, copy, modify, and redistribute computer programs (instiutional host for GNU project and steward of the GNU GPL)
1986	*Internet Engineering Task Force (IETF)*	Concerned with the evolution of the Internet architecture and the smooth operation of the Internet
1991	*IANA*	Dedicated to preserving the central coordinating functions of the global Internet for the public good
1992	*Internet Professional Society (ISOC)*	Provides leadership in addressing issues that confront the future of the Internet and is the organization home for the groups responsible for Internet infrastructure standards
1994	*W3C (World Wide Web Consortium)*	Develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential
1997	**Software in the Public Interest**	**Helps organizations develop and distribute open hardware and software (institutional host for Debian)**
1998	Open Source Initiative	Dedicated to managing the promoting the Open Source Definition
1998	*ICANN*	Responsible for global DNS management
1999	**Apache Software Foundation**	**Provides support for the Apache community of open source software projects**
1999	Linux Professional Institute	To design and deliver a standardized, multinational, and respected program to certify levels of individual expertise in Linux

Table 20.1
(continued)

Date	Organization founded	Mission
2000	Perl Foundation	Dedicated to the advancement of the Perl programming language through open discussion, collaboration, design, and code
2000	FreeBSD Foundation	Dedicated to supporting the FreeBSD operating system
2000	Free Standards Group	Dedicated to accelerating the use and acceptance of open source technologies through the development, application, and promotion of standards
2000	**GNOME Foundation**	**Provide a user-friendly suite of applications and an easy-to-use desktop; to create an entirely free desktop environment for free systems**
2000	KDE League	Promote the use of the advanced Open Source desktop alternative by enterprises and individuals and to promote the development of KDE software by third-party developers
2000	Linux International	To work with corporations and others to promote the growth of the Linux operating system and the Linux community
2001	Python Foundation	Advancing open source technology related to Python programming language
2001	Jabber Foundation	Provides organizational and technical assistance to projects and organizations within the Jabber community
2002	Open Source Application Foundation	To create and gain wide adoption for software applications of uncompromising quality using open-source methods

Key:

Internet Governance Organizations

Free Software/Open Source Organizations

Organizations in bold studied in greater detail.

technical communities were founded. However, until 2002, the FSF was not a membership organization. FSF leadership viewed a democratic governance structure as potentially detrimental to its mission, stating, "We don't invite all the people who work on GNU to vote on what our goals should be, because a lot of people contribute to GNU programs without sharing our ultimate ideals about why are we working on this" (FSF founder, April 26, 2001).

The trade-off that the FSF made to ensure commitment to its political goals was to sacrifice democratic goals.[9] Thus, while its influence technically, legally, and conceptually is immeasurable, its influence as an organizational model for community-managed projects was limited. Without members, the FSF functioned as a corporate shell for the GNU project and as a political organization devoted to changing the social, economic, and legal arrangements that guide software development.

Since Fortune 500 firms were first challenged with the idea of "collaborating with a Web page," Apache, Debian, Gnu, Gnome, FreeBSD, Jabber, Perl, Python, KDE, BIND, Samba, the Linux kernel, Linux Standards Base, Mozilla, and Chandler have designed private nonprofit foundations to "host" their projects. The institutional hosting concept may be borrowed from the IETF and W3C models, but these projects have adopted it in different ways. This chapter compares the foundations created by the Debian, Apache, and GNOME projects and concludes by examining the role of nonprofit foundations in community-firm software collaboration.

Research Methods

Between April 2000 and April 2001, I interviewed 70 contributors to community-managed projects[10] to find out how the commercialization of Linux was affecting the free software community and the peer-managed development style that had emerged over the late 1980s and 1990s. I was curious as to how commercial attention and participation on open source and free software projects would affect the hacker culture and loose decision-making structure. Two-thirds of my informants were corporate-sponsored and the rest were volunteers. Most of the sponsored contributors had initially been volunteers and now worked in firms supporting the development of open source software. To assess how specific projects were affected, I focused on the structuring activities of three of them: Debian, GNOME, and Apache. Observations at project meetings, conferences, "hackathons," and other events, coupled with online project documentation such as project discussions, charters, bylaws, and meeting minutes,

helped provide triangulation of the data. This data was coded and analyzed, with a focus on the emergence of common themes that held across variance in perspectives.

Comparing the Emergence of Three Foundations

After the FSF was established in 1985, few foundations emerged until the Debian project created one (Software in the Public Interest) in 1997. The Apache httpd server group founded the first membership-based foundation in 1999. During the course of this research, the GNOME project began crafting their foundation. Each of these projects varied in their stance toward commercial relations, but they all shared a large, mature user and developer base and had attracted commercial attention. Comparison of their approaches shows how different project ecologies approached the task of building a foundation at different points in time.

Debian

Debian, the most popular noncommercial distribution of Linux, has been operating for almost 10 years under the guidance of six different leaders. Over 1,000 members of Debian contribute to the 7,000 packages that constitute the Debian operating system.[11] Debian is viewed, even by long-time members in the community, as a serious hacker's distribution. Thus, it is of no surprise that although Debian was one of the earliest projects to create a nonprofit foundation, Software in the Public Interest (SPI), it did so with some ambivalence. Members of Debian were more resistant to the idea of incorporation and had greater fear of losing control over the technical direction of their project than members on the other projects. However, some leaders were concerned enough about individual liability to want to pursue incorporation and encouraged resisters to adapt the idea in its most minimal form.

Of the three foundations studied, SPI is the least active: it does little more than hold Debian's assets. Members of Debian revised their bylaws to stipulate that SPI's role is to merely serve the project as a legal steward. Debian, like the FSF, has struggled with how to become a membership organization. All potential contributors must pass a very formalized five-step process to become an official Debian member. However, membership in Debian does not trigger membership in SPI. Project members preferred an "opt-in" approach, as opposed to equating Debian membership with SPI membership. Membership in SPI has thus been slow to activate, which has led to some concern about how an appointed board can represent the

project. As a board member commented, "SPI without a membership is just a legal framework for Debian, but with a membership it becomes an organization that can attempt to move on issues that are key to the development of the Net. This is also why the membership is important: SPI without a membership (and just a board of directors) may not always reflect the concerns of the community. With a membership, SPI becomes a representation of the community, and can involve itself in issues that decide the future of that very community" (SPI Board Member Posting, October 26, 2001).

SPI is not structured to ensure representation of project members. However, the Debian Constitution outlines a sophisticated process whereby project members elect leaders for a one-year term. Member representation thus rests within the project as opposed to the foundation. While the other two foundations created a role for firms to provide a voice into their organization, SPI did not.[12] Debian was also the only project of the three to have an internal project leader initiate incorporation. The other two projects all received legal assistance in drafting their charters and thinking through governance issues from two different Fortune 500 companies.

Apache

The primary reason for incorporation proffered by informants on all three projects was to obtain protection from individual liability. The ability to accept donations, hold assets, host additional projects, and represent the project as one legal entity to the public were also concerns. Apache was the only project to explicitly mention the welfare of its customers as an additional reason to incorporate.[13] As one founding member explained, "It is a valuable project. Most of us will stay involved in it forever. Also our customers need stability. Apache had to be seen as an ongoing group and I think that making a legal entity was the way to do it" (Founding Member #1, Apache Project, September 28, 2000).

This is evidence of the distinctness of the Apache group culture. First, the founding eight members of this project licensed their httpd server software under a license that allows proprietary extensions of their work (the Apache License, a variant of the BSD License). Second, many of the early contributors worked in enterprises that were building Web sites and using the software for commercial purposes. The Apache group was also one of the earliest projects to be approached by a Fortune 500 firm to collaborate and the first project to create a membership-based foundation that integrated project governance. This is most likely why informants from pro-

jects that incorporated later often cited the Apache Software Foundation (ASF) as an influential model.

The code contributing population of Apache is smaller than Debian and more centralized. Over 400 individuals contributed code to the httpd server project between February 1995 and May 1999, but the top 15 developers contributed over 80 percent of code changes (Mockus, Fielding, and Herbsleb 2000). Code contributions are a necessary but insufficient condition for membership. ASF membership additionally requires nomination by a current member, a written application and a majority vote. There are currently 109 members of the ASF, of which 34 percent are independent or have no organizational affiliation. The ASF has maintained that only individuals can become members, but that companies may be represented by individuals. The ASF has not implemented any formal approaches to ensure pluralistic representation as of yet, although it has been discussed. Sponsored contributors are from organizations that are diverse enough that no majority or controlling interest from a single organization has yet to emerge.

The ASF's governance structure is akin to a federalist model. Since its founding in 1999, the ASF has grown to host 20 projects in addition to the httpd server. Each project has a Project Management Committee (PMC) with a chairman who also serves as an officer of the corporation. PMC chairs report project status to ASF members and the board, but technical direction of the projects remains the purview of those on the project. ASF members meet annually to elect their board of directors. Directors do not typically interfere with the discretion of PMC leaders, but can decide whether to charter, consolidate, or terminate PMCs. The ASF also organizes conferences and annual face-to-face meetings. Neither the ASF nor SPI employ people to manage administration, largely because members on both projects did not want to engage in the business of "managing people." Members on both projects worried that engaging in employment relations might distract them from what they best enjoyed about participating in their respective projects.

According to volunteer and industry informants, ways to formalize the "core" Apache group had been the subject of discussion for some time prior to incorporation, but corporate interest in collaborating was a catalyst to begin drafting the ASF bylaws. "With [a Fortune 500 company] getting involved and wanting to figure out what the structure was, we realized that we needed to kind of solidify our processes a bit and put some formalism to it" (Founding Member #2, Sponsored Contributor, Apache, September 28, 2000).

The ASF did not create an explicit role for firms other than through sponsored individual contributors. However, the ASF has engaged in several formal transactions to accept intellectual property contributions from Fortune 500 companies, most recently brokering an intellectual property agreement on an open source implementation of Java with Sun Microsystems.[14] Like SPI, the ASF holds assets in trust for the Apache project and the other projects it hosts. This includes the Apache trademark, donated hardware and equipment, and intellectual property donated by firms as well as by members. The ASF asks volunteer contributors to sign an agreement that ensures the software they donate rightfully belongs to them and assigns a nonexclusive copyright license to the ASF. The ASF was more vigilant in seeking copyright assignment than the other two projects.[15]

GNOME

More than 500 developers contribute to the GNU Object Model Environment (GNOME) project, 20 percent of whom are reportedly full-time paid developers. GNOME is a complete, graphical user interface (GUI) desktop application designed to run on Linux-based operating systems, BSD, and a variety of other Unix and Unix-like operating systems.[16] The GNOME Foundation membership is larger than the other projects with over 300 members, and new members do not require a vote by the majority. Candidates who feel that they have made nontrivial contributions are welcomed to apply for membership, but the exact criteria are not well articulated at this stage. Foundation members have the right to elect a Board of Directors and have held three successful elections thus far. GNOME has hired an executive director to oversee fundraising and the development and growth of the foundation.

More corporations directly participated in the creation of the GNOME Foundation than on the other projects. Similar to Apache, a different Fortune 500 firm donated their legal expertise to help a steering committee draft the GNOME charter and file the necessary paperwork. While firms that wanted to collaborate with the Apache project were primarily interested in seeing the group formalize to make transactions more viable and secure, firms working with the GNOME project wanted to influence the foundation to gain a greater voice in decision making.[17] As one contributor working on the bylaws put it, "with [Fortune 500 firm #2] coming to the front, all these issues of control and governance became so much more urgent, because look at [firm #2]—it's a very competitive, very aggressive culture there. And the way they started their conversations with GNOME reflected that" (Sponsored Contributor, GNOME, February 8, 2001).

The GNOME foundation resisted this type of direct pressure by granting firms a role on an Advisory Board that provides a venue for firms to articulate their concerns and ideas, but does not offer technical decision-making rights.

The GNOME project was the only project of the three that allowed their foundation to assume control over release coordination. If there was one role assumed by foundations that was most controversial in the eyes of informants, it was release coordination. Release coordination includes setting a schedule, choosing the modules that will define a release, and marketing.[18] One informant felt that granting the foundation release coordination authority could effectively blur the boundaries between organizational and technical decision making and threaten members' control over the technical domain. "The reality is that, in my opinion, the foundation is going to end up running GNOME. And people don't want to say that because it just runs counter to the democratic values of the thing, but [. . .] if you look at release coordination alone, it gives you so much control, that you're effectively running the thing. Because what you end up saying when you do a release is deciding what is a part of it and what is not a part of it, right?" (Sponsored Contributor, GNOME, February 8, 2001).

How this authority is enacted with the developers directly responsible for modules within GNOME is still evolving. The GNOME Foundation has greater project representation within its foundation, but also has centralized more power than the other two projects.

Evidence from informants and project documentation indicates that GNOME faced greater pressures from commercial sources to coordinate in ways that were atypical to the hacker ethos than did the other two projects. These pressures were manifested in project members' resistance to expressed commercial preferences for a more predictable and stable development environment. Centralized release coordination authority enhances a firm's ability to more reliably predict components and deadlines associated with a release and thus better manage its own product development activities.

GNOME's experience may have differed because its foundation worked with more firms in more formalized and explicit roles than did the other two projects or because application development by its nature demands more commercial collaboration than software development at the operating system and Web server level. Pressure to coordinate may also be a function of commercial interest in the advancement of open source desktop applications or a function of the later stage at which the GNOME foundation was developed. (The GNOME foundation was created much later

than either the ASF (1999) or SPI (1997), at a time when commercial entities had become more aware of open source software.) Regardless of the weight attributed to these reasons, the GNOME project experienced more commercial pressure when creating their foundation than either Apache or Debian. Their resulting foundation exhibits greater centralized authority over software development.

Other Foundations

Nonprofit foundations help programmers retain the normative order they prefer while creating a legal entity that can protect their work in commercial markets. In addition to these three foundations, there are now at least a dozen foundations that support the development of free and open source software, five of them founded in 2000 alone. All but two of the foundations listed in Table 20.1 are 501(c)(3) nonprofit organizations.[19] The precise structure of each foundation reflects challenges specific to each project's ecology, but there are also patterns developing that will likely change with challenges from commercial markets. This is an evolving model that has yet to reach settlement. One of the most well-known open source projects, the Linux kernel project, initially resisted the need to create a foundation:

> For a long time, there has been some talk about having more structure associated with the kernel. The arguments have not been that strong. People just expect the structure to be there. So they want to build structure because they think it is wrong to not do it. That seems to be the strongest argument, even though it is never said that way. But there have been for example, commercial companies who wanted to transfer intellectual property rights and there is nothing to transfer to, which makes their legal people scratch their heads, right? (Project leader, Linux kernel, March 12, 2001)

The community's and industry's faith in the leadership of this project and the leader's disinterest in institution building enabled the Linux kernel project to manage legal ambiguity for a long time without undue pressure to incorporate. With the creation of http://kernel.org, the Linux kernel project now has a shell foundation in place but trademark rights remain individually held.

Foundation Efficacy

It is too early to determine how successful project foundations have been at fulfilling their mission. Project leaders recognized that any structure that was too formal or burdensome would conflict with the hacker ethos and lead to potential mutiny. A successful organizational design was, in the

eyes of one informant, one that "members could live with"; an organization that infringed minimally upon the hacker ethos of technical autonomy and self-determination: "[A]s far as I can tell, we have created an organization that can live with the community and the community can live with it and work together towards maintaining our software over a long period" (Volunteer contributor, Apache Project, July 19, 2000).

Informants indicated that there were early signs that their foundations helped facilitate communication between communities and firms and helped to avoid, or at least diffuse, potential problems. If this were true, these effects would be more difficult to detect.

Another test of the efficacy of a foundation is its ability to maintain mutually beneficial relations between firms and communities. Informant explanations of mutualism often focused on the different types of competencies and resources that communities and firms could contribute to technical problems:

> I think our main contribution is that we are using Debian and we are looking at Debian from a commercial point of view. And making improvements to make it more attractive to companies as an alternative to the commercial systems. So we are doing work that a nonprofit group is not necessarily interested in doing and looking at Debian from a different point of view. So our hope is that by doing that, we are going to be able to help Debian improve and expand its audience beyond where it is now. (Former leader, Debian, open source firm founder, February 16, 2001)

As this informant explains, the customer-oriented commercial lens that firms brought to development work could provide a different, but complementary, focus to the more foundational concerns of hackers. Complementary as opposed to competing foci fostered symbiotic working relations between community-managed projects and firms. To the degree that firms and community projects share the same goals and interests (for example, to expand their market share) despite divergent motivations, and to the degree that each type of actor maintains different foci, informants felt that symbiotic relations were possible. Maintaining this balance was understood however to require social structures that reinforced pluralism and the balancing of community and firm interests.

Facilitating Community-Corporate Collaboration: A New Actor in the Supply Chain

The foundations that emerged in this study are incorporated and organized by and for individual members. They produce benefits for the public, but do not redistribute profits to their members. What is unique about these

foundations, in relation to technical communities of the past, is that these foundations also own assets that are sold by third parties in commercial markets and may in fact compete with other commercial offerings. Firms that use free and open source software have, in effect, allowed community-managed projects that grew out of a politically motivated social movement to become a part of their supply chain. This interdependence has fostered a new set of working relations among community projects, their foundations, and firms. Figure 20.1 outlines the role of nonprofit foundations in this new collaboration model. Foundations hold the assets and property rights of technical communities that produce software, but do not pay their developers or redistribute profits to their members. Community members retain the ability to set their own technical direction and manage the culture, norms, and governance of their own projects. In return for assigning their intellectual property to a foundation, they are granted protection from individual liability and a means to legally represent the project.

Firms can sell and distribute the community's work at a profit by creating complementary software, hardware, and services that reflect their conception of market needs. They can modify the work of the community as long as they respect the terms of community licenses and contribute improvements back to the code base where required. In return, firms offer sponsorship and support to both individuals and foundations. Individual volunteers that are working on components of critical interest to firms may be hired to continue their efforts as sponsored contributors. Proprietary code, financial resources, hardware, and equipment that firms wish to donate to the project are entrusted to the foundation. In return, some foundations offer firms advisory or sponsor roles: mechanisms that can provide them with a voice on the project. On a day-to-day basis, commercial support of community-managed projects is enacted through the sponsored contributors that work on those projects. On a legal basis, the foundations play an important mediating role. In figure 20.1, release coordination is depicted with a question mark sitting between the authority of projects and their foundation. The strength and role that foundations play when collaborating with firms may well depend on the degree to which the authority of the foundation touches the technical core of the project.

In this model, the ownership and maintenance of code is decoupled from its sale and distribution. Without some means to retain their rights, it is unlikely that community-managed projects would have had the base of power necessary to engage with firms and create this model (O'Mahony

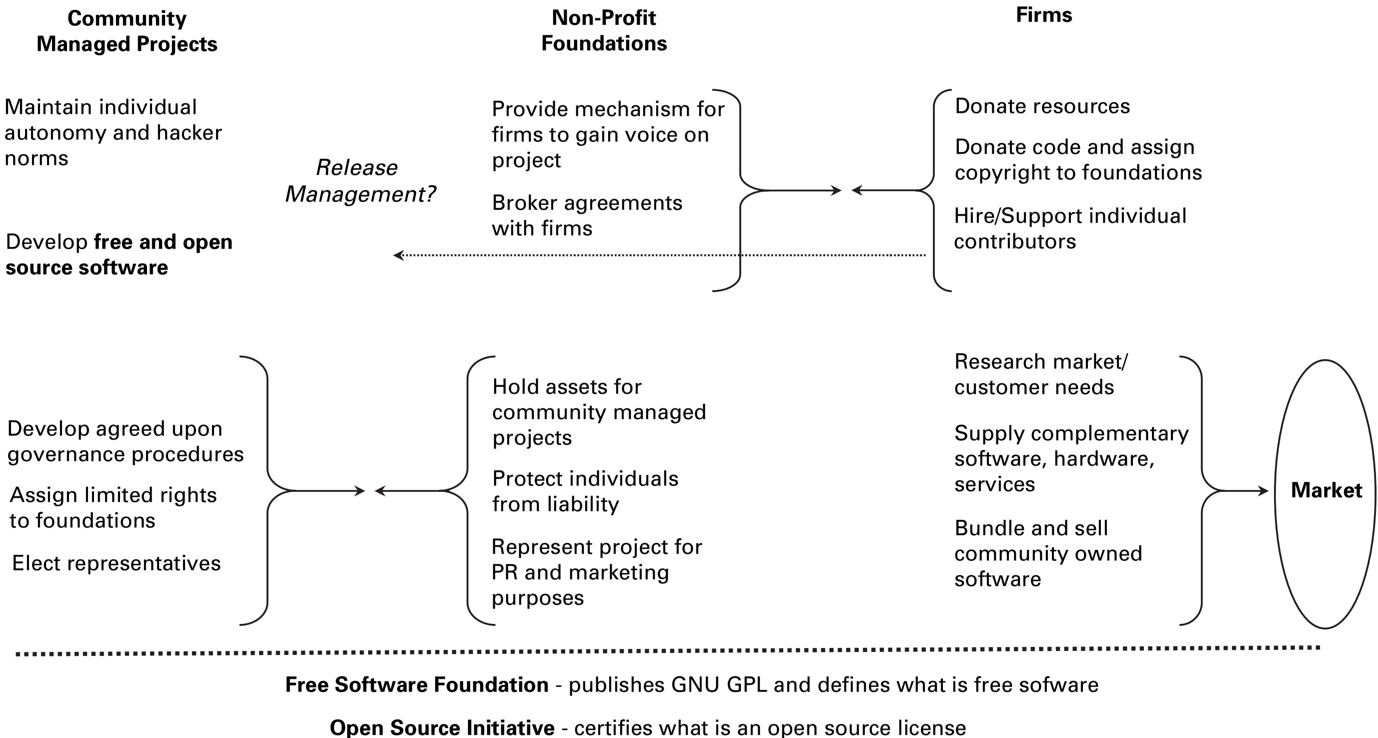

Figure 20.1
The role of nonprofit foundations

2003). Firms, for example, could have legally used community-developed software without necessarily collaborating with them. Community-managed projects held two bases of power that helped firms consider them a credible partner for collaboration: the market share and user base that derived from a project's technical excellence and the legal and normative controls that encouraged users to "give back" to the project. These two bases of power offset technical communities' lack of economic and political power and helped establish them as a viable commercial actor with which firms could partner.[20]

Granted, this analysis provides a rather static view of the legal and organizational structures that underlie a larger and more complex network of social relationships that flow in and out of these different forms. For example, a volunteer contributor could become sponsored by a firm and then be elected to a board position in a nonprofit foundation. Individuals who were once volunteers and have since founded firms may be active in shaping the nonprofit foundations that represent their project. Informants often stressed that they wished to perceive each other as individual contributors without regard to organizational affiliation. And yet, many informants that occupied two or more roles acknowledged that they often experienced role conflict when their activities touched multiple interests.

An implicit but unarticulated tenet of the hacker ethos is the desire to maintain pluralism. This belief takes two forms. First, there is pluralism in voice and process. Raymond has argued that with "more eyes, more bugs are shallow" (2001). An unstated condition is that diverse eyes are necessary for this lay maxim to hold. The more programmers from diverse cultures and backgrounds run various applications in different computing environments, the more likely it is that each user will detect problems unique to them. This allows code to be tested and contributions designed at a level that would require more permutations than are possible at most software firms. Diversity matters as much as volume. The second form of pluralism is required to make multilateral contributions possible: pluralism in the computing infrastructure itself. Software that is created independent of any one vendor's terms, is portable to different types of operating systems, and is interoperable with other applications allows pluralistic contributions to continue. The principle of pluralism depends upon shared standards and protocols, but I would argue that it also depends upon a form of organization that prevents dominant interests from forming.

Herein lies a source of conflict. Individuals contributing to community projects want to recognize each other as individuals, retain their individid-

ual autonomy, and remain as free from their employment affiliations as possible. On the other hand, without recognition of organizational affiliation, preserving pluralism will be more difficult. Project responses to potential conflict of interest problems have varied, but one feature that works in their favor is public disclosure. The organizational affiliation of project leaders is typically publicly available. When the relationship of one's activities to one's organizational affiliation becomes suspect, other community members are likely to be vocal about their concerns. For contributors who adopt project-based e-mail addresses, affiliation is less public. Over time, this could lead to further blurring of these different roles. The governance structure foundations provide may become one way to help preserve pluralism.

The evolution of a symbiotic relationship between community-managed projects and firms required adaptation from both actors, and some of these changes are manifested in the roles that nonprofit foundations fulfill, but not all. An understanding of how community-managed projects and firms maintain this relationship at the level of code contribution requires much more explication than has been discussed here. This structural examination of the community-firm collaboration model distributes a very different set of power, ownership, and rights than has been fully appreciated. From an economic perspective, one might ask whether community-managed projects outsourced their distribution costs, or whether firms outsourced their development costs. Arguments could be made to support both lines of thought, which is in itself perhaps a test of mutualism. A more sociological perspective might question whether community-managed projects that are both politically and pragmatically motivated have successfully resisted cooptation by powerful market dominants. Legally, nonprofit foundations play a critical role in preventing this from happening, but this role reinforces mutual relations that are normatively maintained. Equally significant implications are likely to stem from the intellectual and innovative contributions that can result from collaboration with a new type of actor in the software industry.

Notes

The research in this chapter was supported by a grant from the Social Science Research Council, with funds provided by the Alfred P. Sloan Foundation, as well as by funds from Stanford University's Center for Work, Technology, and Organization and the Stanford Technology Ventures Program. This research benefited from the helpful comments of the editors as well as Steve Barley, Bob Sutton, Mark

Granovetter, Jason Owen-Smith, Woody Powell, Neil Fligstein, Doug Guthrie, Rachel Campagna, Fabrizio Ferraro, and Victor Seidel. All errors are mine. I also thank my informants, who generously contributed their time.

1. A few community-managed projects allow organizations to participate as contributors; most only allow individuals to participate as contributing members.

2. I use *community-managed software project* to distinguish from *open source* and *free software* projects that can be sponsored and managed by firms, because firms can also start and manage open source projects.

3. The definition of *open source software*, which is based on the Debian Free Software Guidelines, is located at http://www.opensource.org. Without a trademark, the OSI, in consultation with their attorneys, designed an "open source" certification program that helps ensure that corporate software licenses that claim to be open source do indeed meet the criteria for open source as defined by the community.

4. This does not exclude the possibility of earning a profit from modifications, extensions of products, hardware, and services to collectively produced efforts.

5. There are more than 450 members who pay dues to the consortia and nearly 70 full-time staff around the world who contribute to W3C specifications.

6. The Massachusetts Institute of Technology (MIT) in the United States, the European Research Consortium for Informatics and Mathematics (ERCIM) in Europe, and Keio University in Japan host the W3C (http://www.w3c.org).

7. ICANN was created in 1998 after a Department of Commerce white paper recommended that this federal function mission be privatized. For more information, see http://www.icann.org/general/white-paper-05jun98.htm.

8. For more information see "President's Report: ICANN—The Case for Reform," February 24, 2002, located at http://www.icann.org/general/lynn-reform-proposal-24feb02.htm. The privatization of ICANN may have been a more challenging task than that of the IETF, because global DNS management requires the active participation of governments and because it had less time to grow a community to support it before Internet access became ubiquitous.

9. In 2002, the FSF developed an associate membership plan, but it offers members limited decision-making rights.

10. I interviewed seven more contributors to community-managed projects in 2002–2003, for a total of 77.

11. The FSF supported Debian in its early years (1994–1995).

12. However, Debian does acknowledge the 143 vendors in 39 countries that sell Debian distribution and its other corporate supporters with a Partners Program.

13. http://www.apache.org/foundation/press/pr_1999_06_30.html

14. "Apache Software Foundation Reaches Agreement with Sun Microsystems To Allow Open Source Java Implementation," March 25, 2002, located at: http// jakarta.apache.org/site/jspa_agreement.html

15. The FSF is also vigilant in asking software contributors to assign their copyright. For more information on a comparison of copyright assignment practices across different projects, see O'Mahony 2003.

16. http://foundation.gnome.org/press/pr-gnome20.html.

17. Although a Fortune 500 firm helped catalyze the creation of the ASF, I did not find primary or secondary evidence of direct pressure from firms in the design of their foundation.

18. Gnome Project Charter, October 23, 2000.

19. The Free Standards Group and Linux International are incorporated as a 501(c)(6) organization. This class of nonprofits is reserved for business leagues and groups such as chambers of commerce. One distinction is that 501(c)(3) organizations provide public benefits, while 501(c)(6) organizations provide mutual benefits to a designated group. In order to earn a 501(c)(3) exemption from taxation from the IRS, an organization must be primarily devoted to charitable, religious, educational, scientific, literary, or public safety endeavors. The IRS has interpreted the development of free and open source software as furthering education or scientific goals.

20. Two other factors may have been important in enabling this collaborative model to unfold: the presence of a monopoly in the software market and digital technology. If cooperatives are partial, as opposed to identical suppliers of the same good, incumbent nonmonopoly firms are more likely to cooperate with a community form. Thus open source software's weakness in some areas of the consumer market coupled with the presence of a monopoly might have provided an opportunity structure favorable to cooperation with nonmonopoly firms. A second enabling factor is the material attributes of digital intellectual property itself. The ability to decouple development, modification, ownership, and distribution of rights helped grant organizing flexibility.