An Initial Description of the C-Form

Article *in* Research in the Sociology of Organizations · November 2011

DOI: 10.1108/S0733-558X(2011)0000033005

2 authors:

Marc-David L. Seidel
University of British Columbia - Vancouver
33 PUBLICATIONS **1,065** CITATIONS
SEE PROFILE

Katherine J. Stewart
University of Maryland, College Park
26 PUBLICATIONS **1,254** CITATIONS
SEE PROFILE

AN INITIAL DESCRIPTION OF THE C-FORM

Marc-David L. Seidel and Katherine J. Stewart

ABSTRACT

This chapter seeks to enhance organizational theory's current typology of organizational architectures to explain a flourishing modern architecture that has developed utilizing the inexpensive communication paths created by technology such as the Internet and wireless networks. As communication and coordination costs have dropped, new organizing methods have grown that are difficult to understand using the traditional organizational architectures. In this chapter, we introduce a new community architecture, the "C-form," which is categorized by (1) fluid, informal peripheral boundaries of membership; (2) significant incorporation of voluntary labor; (3) information-based product output; and (4) significantly open sharing of knowledge. Although the domain of open source software (OSS) is frequently cited as an example of such communities, we argue that the form expands well beyond the domain of software to a wide variety of information-based products. Drawing on a culture frame, we develop an initial set of principles of C-forms and finally explore the implications of the C-form for the modern organizational world.

In 1857, the Archbishop Richard Chenevix Trench spoke at a meeting of the London Philological Society and suggested the need to address deficiencies

Communities and Organizations
Research in the Sociology of Organizations, Volume 33, 37–72

ISSN: 0733-558X/doi:10.1108/S0733-558X(2011)0000033005

in then-current dictionaries. He argued that a truly comprehensive dictionary was needed, one that included *all* English words, present and past. In addition, the dictionary would include the history of each word, documenting how its use had developed and changed over time. To create such a dictionary would be a monumental undertaking. To accomplish it,

> Trench presented an idea-an idea that … was potentially dangerous, and even revolutionary. But it was the idea that in the end made the whole venture possible.
>
> The undertaking of the scheme, he said, was beyond the ability of any one man … [it] must be instead the 'combined action of many.' It would be necessary to recruit a team-moreover, a huge team, one probably comprising hundreds and hundreds of unpaid amateurs, all of them working *as volunteers*. (Winchester, 1998, p. 93; italics in original)

The Society accepted the idea in 1858, and Herbert Coleridge became the first editor of what was to become the Oxford English Dictionary (OED). Coordination of the thousands of volunteers needed was time consuming and labor intensive. In 1861, the volunteers contributed words and usages on slips of paper that reached a weight of two tons. By 1882, volunteers had provided a total of 3,500,000 slips. The first complete publication of the OED did not arrive until 1928.

The idea of using globally distributed volunteers to produce a large complex product was shocking and revolutionary in 1857. In 1991, the idea of using globally distributed volunteers to produce a reliable, enterprise-ready computer operating system was similarly shocking and revolutionary. Yet it was done: Linux, an operating system produced in this fashion, is currently used by millions of individuals and organizations worldwide. Whereas it took 70 years to cumulate and combine the efforts of thousands of worldwide volunteers to produce the first edition of the OED in 1928, which included over 400,000 words, it took less than a decade for globally distributed contributors to produce over 2,000,000 lines of code for Linux (Godfrey & Tu, 2000). The advent of advanced information technologies has made communication across large distances virtually instantaneous and nearly free, allowing communities like those that produced the OED and Linux to become commonplace. These types of communities expand well beyond the domains of dictionaries and software. The proliferation of advanced information and communication technologies (ICT) is an environmental phenomenon that has set the stage for the development of the new form that is now employed in numerous industries and domains as diverse as software, reference book publishing, geography, product design, scientific data, nanotechnology, and consumer information guides.

As Schoonhoven (2003) notes, while discussing the impact of the open source software (OSS) phenomenon, it is difficult to conceptualize this organization of work without defining it. Dijksterhuis, Van den Bosch, and Volberda (1999) argue that the emergence of new organizational forms is a coevolutionary process at the intersection of managerial actions, institutional influences, and environmental phenomena. Our goal in this chapter is to answer Schoonhoven's call to define this phenomenon by reviewing the substantial body of work that has started to study the structure and introducing an architecture that captures the key characteristics of this kind of organization of work. This is a type of community-based form of organization, which we refer to as a "C-form."

There has been a long history of research aimed at understanding the organization of work. Davis and Marquis (2005) argue that organizational theories are bounded by the institutional environment at the time of the theory creation. Building on the scientific management approach of Taylor (1911), Barnard (1938) introduced problems of cooperation, whereas Simon (1945) introduced bounded rationality. Both enhanced organizational theory by considering behavioral aspects of management. The behavioral theory of Cyert and March (1963) added a discussion of the political nature of work. Contingency theory (Galbraith, 1973; Lawrence & Lorsch, 1967) advanced the field by considering the various environmental conditions that true organizations face. Bartlett and Ghoshal (1993) proposed that a radical change in both internal and external firm environments led large global corporations to create a new organizational form. They argued that the new form was fundamentally different from the multidivisional form (M-form) and called for a "managerial theory of the firm." As Davis and Marquis (2005) suggest, perhaps organizational theories are once again outdated and need to be updated to reflect and account for modern developments. We seek to add to this historical stream of advances in organization theory by describing the emerging architecture enabled by the wide diffusion of recent advances in ICT. Understanding the organization of work at the community C-form level gives a richer understanding of relations among all members of a field than studying a single organization ever could. Examples of the C-form have emerged as a result of the OSS phenomenon that has garnered attention in the information systems and organizational literature (Gallivan, 2001; Lee & Cole, 2003; O'Mahony & Ferraro, 2007; Schoonhoven, 2003; Stewart, 2005; Stewart & Gosain, 2006a; von Hippel & von Krogh, 2003), but it is not limited to that domain (nor are all OSS projects created using C-forms). The architecture has also appeared in numerous other information-based domains, including some that result in physical products such as

cars and beer (cf. Raasch, Herstatt, & Balka, 2009). The new means of organizing that these communities use to produce these products are changing the way that organizations are built and interact with their environments by blurring traditional organization boundaries for knowledge workers in all information-based industries.

Carroll and Hannan (2000) define organizational architectures as a method of organization that can be found in many different organizational populations, as opposed to organizational forms, which have an industrial boundary-based definition. Modern examples of organizational architectures include the M-form (Williamson, 1975) and the "network form" (Podolny & Page, 1998). The Carroll and Hannan (2000) definition of architecture and form differs slightly from the Bartlett and Ghoshal (1993) usage of the term organizational form. Carroll and Hannan (2000) suggest that an organizational form has an industrial boundary, while architectures are more universal design characteristics that can exist in numerous industries. For the purposes of this chapter, we use the Carroll and Hannan definition. While current architecture theories describe many organizations, we are now faced with a theoretically unaddressed organizational reality. The current organizational architecture typology fails to adequately capture the essence of the new organizing method that has emerged, which relies on a community of contributors without clear organizational boundaries. Our goal in this chapter is to describe and define this architecture. We hope to expand the tools of organizational theorists for understanding these new kinds of organization that have been emerging at an increasing pace with the advent of inexpensive ICT. Although not all OSS are produced using C-forms (cf. O'Mahony, 2007), one of the most studied domains of C-forms to date is in OSS development; thus, we choose one of the most studied examples from this domain, Linux, for in-depth discussion.

The remainder of the chapter is structured as follows. We first provide a brief overview of the development of OSS and summarize existing perspectives for studying it. We next examine one illustrative example, Linux, and attempt to understand the organization of work using multiple existing organizational architectures. We conclude that these existing architectures do not adequately capture the organizational characteristics of the phenomenon, and we then draw out the defining characteristics of the C-form based on the points of divergence from existing architectures. Following this discussion, we review the development of several non-software organizations that we believe are best characterized by the C-form. To exemplify the architecture, we draw from communities providing educational and consumer information products on the Internet. In the latter part of the chapter, we use a culture

frame to synthesize past work and develop principles regarding the C-form architecture. Finally, we discuss the issues and opportunities the new architecture presents to organizational researchers.

OVERVIEW OF OPEN SOURCE SOFTWARE

In the early days of computing (i.e., 1950s–1960s), software was generally shared in a readable format that allowed users to edit it. As the software industry evolved through the 1970s, it became common practice to distribute software in a closed format that prevented user modification. In reaction to this trend, the Free Software Foundation (FSF) was created in 1985 with the purpose of promoting individuals' rights to use, copy, modify, and redistribute software.[1] Another organization, the open source initiative (OSI), was formed in 1999 with a goal of helping to bridge the ideological gap between the business world and the world of open software development. Defining characteristics of both free software and OSS include licensing provisions that allow free redistribution, ensure the availability of the source code, and allow distribution of derived works under the same terms as the original.[2] While those involved in the creation of free software and OSS highlight important distinctions in their philosophies, for the purposes of this chapter, we group the two together and refer to software licensed under these kinds of provisions as OSS. Many of the software programs that comprise the critical infrastructure of modern society are OSS programs. Examples include Apache, which is the most widely used web server, and Sendmail, which handles most Internet e-mail.

Research has identified common organizational characteristics of OSS development. OSS projects are typically begun by individuals or small groups working outside the formal boundaries of their job responsibilities (i.e., working voluntarily) (Bretthauer, 2002; Lee & Cole, 2003) or by organizations that open their software to a broader development community. Individuals begin OSS projects when they encounter an unmet software need or a problem with some existing software. The software does something they do not like, or it does not do something they want it to do (DiBona, Ockman, & Stone, 1999). They start working on a solution to the problem and post the code on the Internet with a call for other interested parties to participate. Many OSS start-up projects do not attract contributors beyond the initial founders (Krishnamurthy, 2002) and show little or no activity after their public announcement (Chengalur-Smith & Sidorova, 2003). These projects

are typically considered to be unsuccessful (Crowston, Howison, & Annabi, 2006; Stewart & Gosain, 2006a).

Studies of more successful OSS development efforts have found that they tend to evolve such that there are distinct categories of participants. Basic distinctions among these categories of participants include core developers, peripheral developers, and nondevelopers (Mockus, Fielding, & Herbsleb, 2002; von Krogh, Spaeth, & Lakhani, 2003). Core developers are those that make the largest number of contributions to and exercise the greatest level of control over the software. There is a larger group of peripheral developers, who contribute software code but in a more intermittent fashion. A still larger group of nondevelopers participate in many ways ranging from making suggestions to the developers (e.g., by reporting bugs or making requests) to providing technical assistance to other users (Lakhani & von Hippel, 2003). Coordination and communication in OSS efforts is generally achieved electronically, using Internet web sites such as sourceforge.net, which provides free services to OSS projects. Each voluntary member is responsible for his or her own Internet connection, hardware, and physical work space. Hence, project coordination is handled with virtually no monetary cost to the community.

Some OSS projects have spurred the creation of other communities and organizations, in particular spin-off projects that develop complementary software and for-profit companies that provide complementary services. Examples of these are OSS projects building software to be run on the Linux operating system and Red Hat, an organization that sells support for Linux. Likewise, some existing traditional organizations have begun OSS projects, perhaps the best known example among these is Netscape. After facing extensive competitive pressure from Microsoft, Netscape turned the coding of its web browser into an open source project in 1998. The creation of this project eventually paved the way for the creation of the popular Firefox web browser, which is created using the open source model, and stole significant market share back from Microsoft's Internet Explorer. Firefox was introduced in November 2004, and it was downloaded more than 100 million times in its first year (Mozilla, 2005). By 2008, it had captured 20% market share.

The increasing prominence of OSS has come with increasing research attention across many disciplines. This research has focused on many issues of interest to the current discussion including the motivation of individuals to become involved in OSS development (Hars & Ou, 2002; Lakhani & Wolf, 2005; Lerner & Tirole, 2002), the means by which they do so (Fang & Neufeld, 2009; von Krogh et al., 2003), how individuals' efforts are

coordinated and controlled (Gallivan, 2001), the influence of OSS culture on projects' effectiveness (Stewart & Gosain, 2006a, 2006b), and the design and evolution of governance mechanisms in OSS (O'Mahony, 2007; Shah, 2006).

MARKET, HIERARCHY, NETWORK, OR SOCIAL MOVEMENT?

One of the earliest and most widely cited discussions of the organization of OSS work was an essay published by Eric Raymond, an active longtime participant in OSS projects and the first president of OSI. Raymond (2000a, 2000c) refers to open source development as a bazaar. In organization theory terms, this bazaar might be referred to as a market. To examine the efficacy of viewing OSS development projects in these terms, we turn to Powell's (1990) review of the key distinctions among markets, hierarchies, and networks. The first three columns of Table 1 are borrowed from Powell (1990). We have added column four to show the ways in which we believe OSS projects created in a C-form are similar to and distinct from each of these modes of organization.

As can be seen in Table 1, OSS C-forms are distinct from markets and hierarchies in their normative basis (in open source communities, e.g., code is given away and property rights are often purposely foregone), their means of communication (communication is many-to-many, and the actions of contributing to a project or using its output are a strong signal as to the value of the project), their methods of conflict resolution (C-forms tend to have fiat without supervision), their tone (open competition of ideas in which the best solutions win), and actor preferences (they are both independent in the sense that they are largely uncoordinated and interdependent in that the independent actions must be assembled into a final product. Perhaps, the largest difference from the other three methods of organizing is the lack of formal or legal contracts. They do not rely on legal employment relationships, market, or alliance contracts.

An alternative perspective could view the open source phenomenon as a social movement (Lounsbury, Kelty, Yavuz, & Colvin, 2009). The guiding belief is that software should be provided in a form that can be modified by the user. The movement is enabled by inexpensive communication, and many members share the "hacker culture" (Levy, 1984). Unlike the typical social movement, opposition is not posed by the state or by other social movements. Instead, opposition comes from profit-seeking companies such

Table 1. The C-Form versus Other Methods of Organizing.

	Market	Hierarchy	Network	C-Form
Normative basis	Contract-property rights	Employment relationship	Complementary strengths and long term contract relationship	Common goals, open sharing and volunteer philosophy
Means of communication	Prices	Routines	Relational contracting	Many-to-many, action, common culture
Methods of conflict resolution	Haggling-resort to courts/ enforcement	Fiat/supervision	Norm of reciprocity/ reputational concerns	Inner circle act as gatekeepers, some fiat, not much supervision
Degree of flexibility	High	Low	Medium	High
Amount of commitment among parties	Low	Medium to high	Medium to high	High in inner circle, low to medium otherwise
Tone or climate	Precision and/or suspicion	Formal/ bureaucratic	Open-ended, mutual benefits	Meritocracy, common values, higher purpose
Actor preferences or choices	Independent	Dependent	Interdependent	Independent (choices to contribute/use), Interdependent (outcomes depend on coordination of independent choices)

as Microsoft that stand to lose revenues if these beliefs are widely accepted (O'Mahony & Bechky, 2008). Evidence of conflict is widespread. One example of this conflict can be seen in the Halloween documents (Valloppillil & Cohen, 2000). The Halloween documents are an internal Microsoft strategy memorandum detailing possible responses to the threat of the Linux OSS community. The Microsoft authors conclude that

> Open Source Software poses a direct, short-term revenue and platform threat to Microsoft, particularly in server space. Additionally, the intrinsic parallelism and free idea exchange in Open Source Software has benefits that are not replicable with our current licensing model and therefore present a long term developer mindshare threat … commercial quality can be achieved/exceeded by Open Source Software projects … More importantly, OSS evangelization scales with the size of the Internet much faster than our own evangelization efforts appear to scale.

Eric Raymond published a response (2000b) that concluded as follows:

> Linus Torvalds jokes about world domination, but Bill Gates means it. What Microsoft has done in the past is very bad, but I would not have gone to war with them over the

> past. The real issue is that they won't leave me and my friends any safe place. They want to hijack the Internet we built with brains and sweat and blood; they want top-to-bottom control of computing everywhere; they're determined to have it all, forever and ever, amen. That is the deepest subtext of the Halloween memoranda. And that, ultimately, is why I must be Bill Gates' enemy.

If open source is a social movement, the OSS communities that form around specific projects such as those related to Linux may be seen as social movement organizations (SMOs) associated with it. The resources are essentially the brainpower of the contributors, and the goal is to create software that serves as a free alternative to commercial software, in particular, an operating system to beat out Microsoft. Although parallels may be drawn, there are several factors that distinguish a typical SMO from a typical OSS community. For example, SMOs generally focus on changing something, whereas OSS organizations focus on creating something, a specific output, which may bring them into direct competition with profit-seeking organizations. Schoonhoven (2003) points out that the creation of useable products of high complexity requiring coordination of geographical dispersed experts differentiates open source communities from classic SMOs. Another difference can be seen in the organization of members. McCarthy and Zald (1977) note a unique type of constituent in an SMO, the "isolated constituent" who does not come into face-to-face contact with others. Because the primary means of communication in OSS is distributed, the majority of members may be isolated rather than the minority. Finally, as the OSS "movement" has evolved, participants have become more diverse and a wider array of motivations and cultural elements have emerged to play significant roles (O'Mahony, 2007). The position of OSS as "opposed" to closed for-profit software production has been tempered, as evidenced, for example, by the annual Open Source Business Conference (www.osbc.com), and boundary organizations have been developed to facilitate collaboration between for-profit companies and community-run OSS projects (O'Mahony & Bechky, 2008).

EXAMINING LINUX FROM EXISTING ARCHITECTURAL PERSPECTIVES

As discussed above, one of the most widely publicized examples of OSS is the Linux operating system, which has been the subject of many studies and essays (Fink, 2003; Hertel, Niedner, & Herrmann, 2003; Lee & Cole, 2003; Ljungberg, 2000; Raymond, 2000a, 2000b, 2000c; von Hippel & von Krogh, 2003). Linux development was started in 1991 by a Finnish student named

Linus Torvalds. Motivated by his own need for an operating system and a desire to learn from the project, Torvalds began building Linux using software that was already freely available. He posted an early version of his operating system on the Internet with an invitation for others who might be interested to help him develop it. Others quickly began contributing and Linux grew rapidly. By 1994, companies (e.g., Red Hat) and publications (e.g., Linux Journal) had started focusing on Linux distribution and use. By 2000, more than 14,000 people had participated in the development of Linux (Lee & Cole, 2003), and many well-known companies were building products and services using Linux including Google, Oracle, and Hewlett-Packard. As of 2010, Linux was well entrenched in organizations' enterprise applications as well as serving as a critical component in everything from cell phones to Mars rovers (Gallaugher, 2010, p. 201). In this section, we consider how the creation and maintenance of Linux might be viewed using the organizational architectures in organization theory's current typology: U-form, M-form, matrix, and network form. Our goal in this section is to highlight the ways in which Linux, as an exemplar of the new C-form architecture, stretches the boundaries of existing architectures. Our approach to describe the new architecture follows Bartlett and Ghoshal (1993) in that we use an example of a community that employs the new form, the community that produces Linux, as a basis for discussing the shortcomings of existing architectures' ability to capture the nature of the C-form. Communities that utilize the new C-form may share some of the characteristics of existing architectures and in that sense might be seen as evolutions or extensions of those architectures. However, these communities have several unique aspects that differentiate them, and we argue the accumulation of those differences are of a large enough scale that viewing these new methods of organization as special cases of existing architectures would be detrimental to understanding how this architecture may be effectively leveraged.

Linux As a U-Form Organization

U-forms have functional divisions such as R&D, sales, and engineering staffed with employees. There are fairly clear boundaries between functional divisions, and membership in multiple divisions is uncommon. Employees understand exactly which box on the organization chart they fit into. Employees in each division report to division heads, and there is a clear chain of command straight to the top of the organization. Dahlander and

O'Mahony (forthcoming) describe an alternate, more "horizontal" work structure that exists in many open source communities, and this is more appropriate in viewing Linux. In Linux, there are groups that focus on various functions including kernel development, development of related components, and documentation. However, membership in these subgroups is fluid, with no strict chain of command, and many contributors participate in multiple functions. Boundaries to peripheral contribution are fairly low and only increase as one gets closer to the core center of the community. Additionally, contributors are not employees of any single organization. Contributors may in fact be employees of multiple competing organizations such as Red Hat and other support vendors. Similarly, there may actually be multiple subgroups working on functional tasks that overlap. Clearly, software production could easily be organized following a U-form, but in the case of Linux, the U-form architecture does not capture key components of the organization of work.

Linux As an M-Form Organization

M-forms are organized around geographical or product categories, which contain functional subdivisions staffed by employees. Again, employees understand exactly which box of the organization chart they fit into, and there is a clear chain of command. Transferring from one division to another requires significant bureaucratic activity. As noted above, Linux contributors are not employees of a single organization. There is no formal employment contract with a single organization that oversees production and no strict chain of command. Additionally, Linux does not have geographical boundaries. It does have subgroups that operate in different languages, but they are not geographically constrained. Contributors may jump from project to project with fewer bureaucratic hurdles. They are free to come and go as they wish. The distinct lack of a contractual employment relationship, lack of strong hierarchy, and freedom of internal transfers are atypical for an M-form.

Linux As a Matrix Organization

In matrix organizations, employees report to both vertical and horizontal chains of command. For instance, a developer may report to both an engineering manager and a sales manager. In the production of Linux, there are no employees, and chains of command are informal in the sense that

there is no strong overarching organizational hierarchy to enforce them. For example, while Linus Torvalds has wielded significant control over the content of the Linux kernel, anyone who disagrees with his decisions has been free to branch off and create their own version of Linux. In discussing the first several years of Linux development, Alan Cox, a leader in the Linux community, said, "There's no 'official' anything … There's this continuous army of people looking and cross-checking for stuff … the willingness to give up control is a basic principle of open source development … the development community is in charge" (McMillan, 1999). The ability for Linux contributors to participate in dozens of Linux subgroups, when viewed from a matrix perspective, would create a multidimensional matrix impossible to interpret. Thus, while responsibilities and roles can be identified in OSS projects generally (von Krogh et al., 2003), the formalized view of chains of command does not capture the true nature of development work organization.

Linux As a Network Organization

In the network architecture, numerous organizations enter into contractual relationships to provide various components and services. For instance, there may be one company that handles R&D, another to handle sales, another to handle manufacturing, and another to handle publicity. Many of these organizations also provide these services to other organizations. Podolny and Page (1998) define the network form as "any collection of actors ($N \geq 2$) that pursue repeated, enduring exchange relations with one another and, at the same time, lack a legitimate organizational authority to arbitrate and resolve disputes that may arise during exchange." The network form is created through repeated economic exchange – frequently characterized by contractual relationships. The network form relies on the assumption of economically motivated self-interested network actors. At first glance, this is perhaps the closest of the four major architectures to capturing the Linux structure. There is a network of individuals and organizations that together produce Linux. However, in the first several years of development, Linux was produced by a loose network of individuals that did not include the organizations that are now involved. The network form literature has primarily focused on networks of organizations, not individuals (Ahmadjian & Lincoln, 2001; Baum, Calabrese, & Silverman, 2000; Beckman & Haunschild, 2002). Additionally, the network relationships found within the Linux community are not contractual or formalized, and economic exchange among the actors may be hard to identify because the licensing terms of Linux

require free sharing of knowledge produced by actors in the form of source code. The basis for early collaboration in Linux moved beyond the classic capitalistic aspects of society to a postcapitalistic system (Adler, 2001) based on trust, intrinsic motivation, and altruism. This is perhaps the largest difference from the network form that relies on self-interested organizational actors entering into mutually beneficial contractual relationships.

Although we can to some degree fit Linux into any of the architectures discussed, doing so necessitates a limited view of the structure of the community that has created it. The fact that we can see aspects of each of these architectures in the creation of Linux is itself an indicator that it does not fit well into any one architecture. Each of the types of organizational architectures discussed is incapable of capturing some of the unique and defining aspects of how Linux has been created and maintained. They are inadequate to fully describe the method of organizing the work that has yielded Linux. They would be similarly inadequate to describe the method of organizing the work of many other OSS projects and more broadly the entire realm of C-forms. We now turn to the defining characteristics of the C-form architecture.

DEFINING CHARACTERISTICS OF THE C-FORM

Based on the differences from existing architectures discussed in the previous section, we propose that the C-form architecture has four characteristics that define it: (1) fluid, informal peripheral boundaries of membership; (2) significant incorporation of voluntary labor; (3) information-based product output; and (4) significantly open sharing of knowledge. The first two characteristics, and to some extent the fourth, set the C-form apart from U-forms, M-forms, matrix organizations, and network organizations. The third characteristic distinguishes C-form organizations from most social movements and other voluntary organizations. Continuing to use Linux as an example, we now turn to a more detailed discussion of each of the defining characteristics of the C-form.

Fluid, Informal Peripheral Membership Boundaries

We define the community of workers in a C-form as consisting of all who contribute to its output. This differs dramatically from traditional organizational membership, which is normally defined through a

contractual employment relationship. As observed in prior studies, C-form members may include members of inner and outer circles, where the inner circle is composed of members who are most heavily involved and have decision authority over the product. For example, Ye and Kishida (2003) identify eight distinct types of community members in OSS projects and arrange them in an onion-like structure with the project leader at the center, surrounded by core members, active developers, peripheral developers, bug fixers, bug reporters, readers, and at the outermost ring, passive users. It is generally the case that anyone may easily participate at the periphery of a project, but to reach the inner levels, where decision authority resides, contributors must gain experience participating at the periphery (Fang & Neufeld, 2009; von Krogh et al., 2003). Thus, boundaries are very permeable at the outer edge and gradually become less permeable as one moves toward the center. In the Linux example, the founder developed an inner circle of "lieutenants" who helped him to screen and incorporate contributions into new releases of the software. Fig. 1 depicts a simple hypothetical C-form with only two kinds of participants, those in an inner circle and those in an outer circle.

The C-form is a permeable community lacking formal membership boundaries near the periphery. Members may come and go at will. On the basis of our definition of the C-form and empirical observations in the OSS context (Dahlander & O'Mahony, forthcoming; Fang & Neufeld, 2009; von Krogh et al., 2003), we make the following assertion about membership: to be a member, a person must contribute, and the more a person makes high-quality contributions, the more likely he or she will progress toward the inner circle. Peripheral entry has very low barriers, and entry into the inner circle may be more regulated than the periphery. Quite possibly, entry into the inner circle can parallel selection and entry into a more traditional organization. But the vast majority of contributors remain near the periphery.

Given that participation determines membership, it may be difficult to determine at any given time exactly who is a member of the community because evidence of participation (i.e., contributions) happens at discrete moments spread across time. The inner circle may be defined more clearly, but the overall membership is difficult to bound. Members come from a community spread across other traditional organizations and across geography.

The community nature of the C-form leads related support organizations to interact with the C-form members differently than traditional organizations. Advertising campaigns for major Linux-related companies have frequently included statements regarding community membership such as

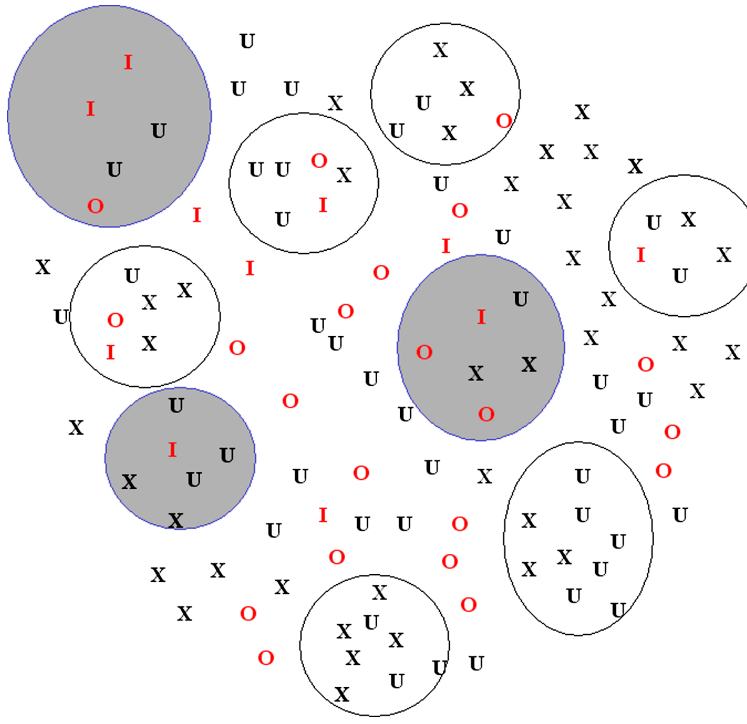

Fig. 1. A Hypothetical C-Form Organization. I's represent members of the inner circle. O's represent members of the periphery, or outer circle. U's represent individual consumers of the C-form product, who do not participate in its production. X's represent individuals who are not consumers. Shaded circles demarcate membership in an organization that seeks rents from the product of the C-form. Unshaded circles represent membership in organizations that utilize the C-form product but are not designed specifically around the C-form. All I's and O's signify members of the C-form.

"An active member of the Linux community" (VA Linux Systems), "At the center of Linux" (Linuxcare), "There is a bonafide internet conspiracy" (InformIT), and "Taps the collective consciousness of the entire Linux/Open Source development community" (Slashdot). The organizations attempt to claim membership in the C-form to gain social acceptance and subsequent usage of their products and services by C-form members. But it should be noted that the products of these support organizations is not the product of

the C-form. They rely on the information-based product output of the C-form and provide additional products and services that complement the base product output. Leveraging the permeable nature of the C-form, these related support organizations attempt to position themselves as members in the contributors' eyes.

Significant Incorporation of Voluntary Labor

The C-form does not provide its members with financial remuneration and does not enter into legal employment contracts with contributors. This is in stark contrast to traditional organizations that normally provide financial reward for employees through a contractual relationship. This fact has been the subject of much discussion regarding the motivation of C-form workers. Simply put, what do they get out of it? Several prominent figures in the open source arena have discussed this question, and the answers that have been put forth range from reputation (e.g., "Open source's best programmers … are involved in a reputation game" (DiBona et al., 1999)) to intrinsic motivation (e.g., "free software made available to him an exhilarating increase in his own creativity, of a kind not achievable in his day job …" (Moglen, 1999)). Research has supported the importance of motivations related to developer utility (i.e., people contribute to OSS projects to satisfy their own needs for the product of the project (Hars & Ou, 2002; Shah, 2006; von Hippel, 2001)), intrinsic motivation such as deriving feelings of competence or altruism (Crowston & Scozzi, 2002; Hars & Ou, 2002), and motivations related to building skills and reputation that may lead to future economic returns (Lerner & Tirole, 2002; Roberts, Hann, & Slaughter, 2006). Such future returns may come from job opportunities at for-profit companies that develop products using the output of a C-form. As is the case with Red Hat, these companies may pay some of their employees to contribute work to the C-form (O'Mahony & Bechky, 2008; West & O'Mahony, 2008). In fact, in some projects, there may be a significant percentage of contributors who are paid by their employers (Lakhani & Wolf, 2005); however, none of them are paid by the C-form community itself.[3] The C-form may incorporate the work of individuals paid by several different employers, but it is dependent on voluntary contributions from individuals or the organizations that pay some contributors. Finding a way to coordinate and balance the interests of many voluntary contributors, including individuals acting of their own volition and employees of companies with an interest in the C-form product, is a unique challenge of the form.

Information-Based Product Output

The C-form is characterized by the production of an information-based product. It shares this aspect with many organizations in the knowledge economy. In the case of Linux, the information-based product is an operating system. It would be impractical to produce hard goods using a C-form as the costs of coordination and transportation would be excessive. Information-based products are well suited to C-form production because the cost of duplication of the final product is trivial. This enables a free sharing of the product without damage or cost to those that create it. That is not to say that hard goods cannot result from a C-form. In fact, a C-form dedicated to product design can facilitate the creation and fine-tuning of the information necessary to create a physical product. The C-form, by its very nature, leaves room for peripheral services such as customer support and manufacturing to be filled by support organizations, and profit-seeking companies may emerge to provide physical goods and services associated with the information product of the C-form. As an example, hundreds of for-profit organizations have sprung up dealing with various aspects of supporting or enabling Linux. These include companies that provide 24-hour user support services through call centers, organizations that do corporate installations of Linux for those without the technical skills to do so, and hardware manufacturers such as Dell and IBM that sell hardware with Linux preinstalled. Similarly for-profit organizations can develop to produce physical goods based on C-form information output such as the Free Beer project that fine-tunes a recipe for beer that is now produced by various for-profit brewers.

Significantly Open Sharing of Knowledge

The C-form is built around an open sharing of ideas and knowledge both within the organization and outside of it. This aspect is radically different from traditional organizations that guard trade secrets, patents, and intellectual property with armies of lawyers. For C-forms, there is little attempt to contain organization-specific knowledge or hide it from competitors. This represents a significant departure from most other types of organizations. Imagine the Coca-Cola Company taking this attitude and opening the formula for Coke up to suggested improvements by interested parties. As an example of open knowledge sharing, Linux's biggest competitor, Microsoft, has complete access to the raw source code of the

entire Linux operating system. Strategic plans for future product development are openly shared as well.

Work on knowledge sharing has focused on the issue of how to encourage employees to share their knowledge effectively with others (Dyer & Nobeoka, 2000). C-forms do not have that problem; sharing is a prerequisite to membership. By definition, all members share. Are there lessons to be learned here for non-C-form organizations interested in leveraging their employees' knowledge? Moglen (1999) suggests part of the motivation for sharing in open source projects is that participants are assured under the licensing arrangements that their contributions will always remain free and no one else can turn them into proprietary products. In this sense, no one else can take credit for them.

EXAMPLES OF THE C-FORM

Our discussion above focused on the OSS domain, in particular, on Linux as a single example of a C-form, to demonstrate each of the characteristics of the architecture and help differentiate it from other architectures. In this section, we discuss some of the numerous examples of communities that we believe are best characterized by the C-form. C-forms appear in domains other than OSS. Although C-forms have been gaining popularity and wide notice recently, the architecture actually first began to appear at least 140 years ago. The earliest example we have been able to find is the community that developed around the creation of the OED, described at the start of this chapter. This community is described in more detail by Winchester (1998). OED volunteers selected the period of history that interested them and in large part determined themselves what they would contribute. At first, they sent their contributions directly to the editor for review. However, within a few years, Frederick Furnivall created a team of assistants to act as intermediaries, screening the volunteers' contributions. Members bore many of their own costs of communication, thereby relieving the OED C-form of those costs. The dictionary was published and sold in volumes. This early example of a C-form demonstrates all of the features listed above. It had an inner circle of editors and a much larger group of voluntary peripheral contributors. It lacked formal boundaries of membership, heavily incorporated voluntary labor, produced an information-based product, and had open sharing of knowledge. As the OED was being published in the early 1900s, the groundwork was being laid for the next generation of C-forms.

ICT advances during the 20th century enabled C-forms to flourish and evolve in many domains beyond software.

Non-Software C-Forms

The advent of the Internet led to the flourishing of OSS quite naturally as the original users of the Internet were computer savvy programmers. However, the C-form can be seen in other domains as well. In their review of the usage of such principles in the biology and nanotechnology domains, Lounsbury et al. (2009) highlight several C-form style communities designed to spread and develop scientific knowledge. Similarly, von Hippel (2001) points out the similarities between open source communities and communities of sports enthusiasts who share their efforts in developing innovations in sporting equipment. In these cases, individuals develop new designs (for nanotechnology or sporting equipment) and share those designs with others who help to improve upon them. In terms of the four defining characteristics: (1) fluid, informal peripheral boundaries of membership allow anyone with the relevant knowledge to contribute their thoughts and designs; (2) the communities heavily incorporate voluntary labor; (3) information-based product output is utilized by for-profit companies utilizing the designs of the community members; and (4) designs and thoughts are openly shared.

Similarly on the consumer watch front, A Bell Tolls (ABT) was established as a hobby by three friends in 1997. It was a clearinghouse of long-distance rate information for the domestic U.S. market (Steele, 2001). Over the years, ABT grew into a large force in the long-distance industry providing data to many government regulatory authorities and consumer protection agencies (HB2914, 2000). The membership of the community was composed of consumer advocates, state and federal regulators, telecommunications pricing specialists, telecommunications consultants, bill auditors, accountants, sales representatives, reporters, and consumers. However, the three founders served as an inner circle that verified and maintained quality checks on the information provided. ABT focused on keeping a watch on long-distance rate changes through the creation of a comprehensive database. Corrections and updates were provided through e-mail by over 12,000 members and distributed through e-mail and the web free of charge to interested parties. Top media outlets such as the New York Times, LA Times, Washington Post, Wall Street Journal, ABC, NBC, and CBS also disseminated warnings from ABT when a major consumer issue was discovered (ABellTolls, 2011) further intensifying the impact of the C-form's

information product on the long-distance industry. ABT spawned a for-profit entity named A Bell Tolls, Inc., which provided customer service to consumers wishing to switch long-distance providers based on the ABT database.

As the economy has shifted toward a greater prevalence of information goods (Shapiro & Varian, 1999), C-forms have begun to play a role in the design of physical goods ranging from beer to cars. Raasch et al. (2009) discuss several examples, including OSCar and Free Beer. The OSCar project was begun by Markus Merz in 1999 with the goal of designing a simple and economical car using open source principles. By 2009, the estimated size of the community contributing to the project was approximately 3,000. It has been followed by the Riversimple Urban Car project, which has already produced a working demonstration model and plans to start production in 2013 of its hydrogen fuel cell vehicle.

The Free Beer project was started in 2005 by a group of students at the University of Copenhagen and a local artist collective known as Superflex with the express purpose of demonstrating how concepts from the OSS movement could be used outside of the digital world. Applying the principles of development they had observed to work in OSS settings, they developed a recipe for beer, "Free Beer." The recipe has been used and modified by many groups and commercial brewers to produce, and in some cases sell, versions of Free Beer (http://freebeer.org/blog/). The realm of possibilities is clear. Voluntary labor pools can be organized in a C-form to produce non-software information-based products. These products can include a wide range of products such as blueprints, designs, architectural drawings, books, and information databases. In fact, many C-forms go so far as to produce commercially viable products evidenced from the success of companies such as Red Hat, which packages and sells Linux in a user-friendly form with customer support, sport equipment manufacturers who mass-produce products based on the innovations freely shared by communities of sports enthusiasts, and A Bell Tolls, Inc., which serviced the needs of long-distance users based on the database created by the ABT C-form.

CULTURE-BASED C-FORM PRINCIPLES

The C-form is rooted in community. Communities are defined by the shared culture of their members (Bechky, 2003), and culture has been highlighted as an important factor in the functioning of OSS projects (Scacchi, 2010; Stewart & Gosain, 2006a, 2006b). We therefore look to the literature on

culture in organizations to synthesize some of the current work on C-forms, mostly done in the domain of OSS, into established principles regarding unique implications of the C-form architecture. This set of principles, using a culture frame, is a first step. Clearly, other theoretical frames can be explored using the C-form architecture, and we hope that future work will expand upon this initial beginning. Similarly, most of the findings synthesized in this section have been generated in the context of software development, and hence, testing is needed in other contexts where the C-form is beginning to thrive.

The managerial view of culture (Ouchi, 1980) suggests that employees can be controlled through intensive socialization processes that internalize the values of an organization. The sociological view suggests that culture emerges based on informal social connections in the workplace (Trice & Beyer, 1993). Both these views of culture are directly relevant to understanding the C-form. The sociological view suggests that C-form cultures would emerge and be reinforced based on the interaction of the founders of a C-form and all the contributors – an evolutionary perspective. The next section relies on this perspective to develop principles regarding the interactions of C-form members, in particular, how their conformance with the culture leads to advancement in a C-form. Due to the difficulty of identifying and monitoring C-form members, the managerial view of culture suggests that cultural control would be critical for the maintenance and growth of a C-form.

Culture and C-Form Membership

C-forms may develop many differing and unique cultural aspects. However, based on the definition of the architecture, some aspects of culture are expected to extend across C-form organizations. The fourth defining characteristic of the C-form is significantly open sharing of knowledge, which operates as a shared cultural belief for contributors (Stewart & Gosain, 2006a). This differs from many other kinds of organizations in which knowledge is a basis of power such that maintaining an information imbalance between individuals by having structural holes strengthens an individual's position. The extensive literature on information brokering through the strategic use of structural holes (Burt, 2004) suggests individuals are motivated to refrain from openly sharing information if they wish to maintain power within a corporate culture that gives power to information brokers. This suggests that to reach the top of a traditional organization, one needs to build structural holes by maintaining information imbalances. However, Fleming and

Waguespack (2007) showed that brokers were less likely than boundary spanners to reach leadership positions in a C-form. Furthermore, C-forms have more a "middle" than a "top" (Dahlander & O'Mahony, forthcoming), and a basic component of the culture of the C-form is a belief in the open sharing of knowledge within and outside of the organization (Stewart & Gosain, 2006a). This is more in line with findings by Collins (1995) about workplace democracy, which showed management gained by sharing power, similar to Kanter's (1979) proposal. This type of power frame seems more appropriate for the C-form. It is by making contributions to the information-based product of the C-form that individuals move toward the inner circle (Dahlander & O'Mahony, forthcoming; Fleming & Waguespack, 2007; von Krogh et al., 2003). This is in line with the fourth defining characteristic of a C-form regarding open sharing of knowledge. Those individuals who practice this community belief progress to the center of the community. Thus,

Principle 1: Open sharing of knowledge increases an individual's power within a C-form organization, resulting in movement towards the core.

Sharing knowledge is only one aspect of C-form culture. The first defining characteristic of the C-form, that members can easily participate near the periphery, combines with the second defining characteristic regarding voluntary labor to create a culture that is open to peripheral contributors with many potentially different motivational bases, but heavily enculturates and indoctrinates them. Pressure to conform with the unique values of the C-form naturally intensifies as members engage more deeply and move closer to the center. Movement toward the center of the C-form involves exposure to many facets of the community's culture and as such is expected to be associated with members internalizing the values of the community (Weaver & Agle, 2002). Strong culture organizations tend to be settings where members experience intrinsic motivation through identification with the culture and values of the organization. C-forms may initially attract contributors to the periphery for various reasons including identification with the purpose of the C-form, social network connections to current members, desire to participate in an innovative unique project, the need for a particular solution, status seeking, the desire to learn, or a classic employment relationship with a traditional organization that values the output of the C-form (Grewal, Lilien, & Mallapragada, 2006; Hahn, Moon, & Zhang, 2008; Lakhani & Wolf, 2005; O'Mahony & Bechky, 2008; Shah, 2006). Case studies of some OSS projects have found that initial participation is mostly driven by extrinsic reasons such as the need for a particular solution, but individuals who continue to participate and move toward the center of the

project develop more intrinsic motivations for participating, doing so for fun and out of a sense of identification (Roberts et al., 2006; Shah, 2006). Regardless of the initial draw to contributing, over time as a contributor is socialized through enculturation and starts to identify more strongly with the culture of the C-form, motivation for contributing tends to shift:

Principle 2: As individuals move closer to the center of the C-form (i.e., the inner circle) their motivations shift to being more intrinsic.

Culture and C-Form Success

Success is a vague concept that may have many different meanings across organizations and communities. To develop principles regarding the impact of culture on success in C-forms, we focus on aspects we believe are particularly important based on the unique characteristics of the form. In terms of inputs, because C-forms incorporate voluntary labor and have permeable peripheral boundaries that allow individuals to come and go freely, attracting and retaining contributors is a key component of success. Whereas traditional views of team work have viewed team size as an input (Ilgen, Hollenbeck, Johnson, & Jundt, 2005; Martins, Gilson, & Maynard, 2004), in OSS projects, participation rates have been viewed as an important outcome (Krishnamurthy, 2002; Mockus et al., 2002; Stewart, Ammeter, & Maruping, 2006). In terms of outputs, because a defining characteristic of C-forms is their provision of an information-based product, assessment of that output is another important aspect of success. Such assessment might be made directly by measuring quality as compared to similar products or less directly by measuring level of usage or market share. Both these approaches have been discussed as appropriate in the OSS domain (Crowston et al., 2006; Stewart et al., 2006) and could be expanded more generally to the C-form. We focus first on the inputs (attraction and retention of members) and then on the outputs (success of the C-form in its marketplace).

Because membership is voluntary and peripheral community boundaries are informal, the coordination and control of members may depend heavily on cultural mechanisms (Scacchi, 2010; Stewart & Gosain, 2006a). However, because potential members of C-forms are distributed and communicate using ICT, it may be difficult for them to be drawn into initial participation due to their inability to understand the culture of the organization. When new members attempt to participate without adequate knowledge of the

organizational culture, their participation may not be recognized or may even be discouraged (von Krogh et al., 2003). Members of virtual teams are less likely to acquire tacit knowledge than members of non-virtual teams (Griffith, Sawyer, & Neale, 2003). Because cultural knowledge is often tacit, and C-form members operate in a virtual team environment, it may be particularly difficult to communicate important aspects of the culture. Thus, C-forms that make their cultural expectations explicit may be more successful in attracting additional participation. Scacchi (2010) highlights ways that successful OSS projects may make cultural expectations explicit through various artifacts referred to as software informalisms. Examples of making the culture explicit from existing C-forms include posting a FAQ or etiquette statement with questions and answers about norms of participation,[4] welcome letters that focus on norms that are sent to new contributors, and mentoring programs with more experienced contributors that focus on instilling cultural norms. These types of explicit communication of the organizational culture may help potential new members feel more confident in their ability to contribute and the likelihood that their contributions will be valued.

Principle 3: C-forms that have explicit documentation of cultural expectations will be more successful in attracting contributors than those that do not.

In addition to communicating important aspects of the culture to potential new members, it is critical for organizations to have mechanisms for propagating and reinforcing cultural expectations among current members. One means by which cultures are maintained is through practices such as rites (Beyer & Trice, 1987; Trice & Beyer, 1984). Rites are planned events carried out for an audience, and they help organization members cope with uncertainty, reinforce status distinctions, and maintain group welfare (Trice & Beyer, 1993). Logically, in a community that relies on strong culture to keep members active, certain kinds of rites may take on an even greater importance than in other kinds of organizations. To maintain voluntary input, it may be particularly important to have rites of passage, which facilitate status transitions (e.g., from the outer to inner circles), and rites of enhancement, which recognize an individual's accomplishments (e.g., valuable contributions to the C-form product). Both of these kinds of rites focus on recognizing individual members (Trice & Beyer, 1993) and thus help satisfy important motivations for voluntary C-form contribution such as the desire for reputation (Hars & Ou, 2002). Examples of these kinds of rites in a C-form may include community-run voting for top contributors,

community-run graduation ceremonies (to move from periphery to inner circle), key signing events where project members meet one another and confirm their "real-world" identities (O'Mahony & Ferraro, 2007), and being selected by the community to be included in the credits of the information-based product output. Thus,

Principle 4: C-forms that develop community-based rites of passage will have higher retention rates than those that do not.

Principle 5: C-forms that develop community-based rites of enhancement will have higher retention rates than those that do not.

While rites tend to happen at discrete times to mark specific events, ongoing social exchanges are another important means of forging and maintaining the bonds among contributors. Jarvenpaa and Leidner (1999) demonstrated that social exchanges early in the life of a global virtual team were associated with better team outcomes. Building trust in a distributed environment relying on ICT has unique challenges. Computer-mediated interaction operates with lower initial levels of trust than face-to-face communication (Naquin & Paulson, 2003). However, Alge, Wiethoff, and Klein (2003) found that computer-mediated teams can overcome the initial trust barriers faced compared to face-to-face teams through social interaction. Social communication helps build trust among members. Building trust among members of a C-form is a key in its long-term success (Stewart & Gosain, 2006a). In the ICT-enabled world of the C-form, online social outlets such as chat rooms, forums, instant messaging groups, and e-mail can enhance social interaction and build trust. There are also face-to-face opportunities such as geographically based meet-ups, user groups, development groups, social events, and in-person community-based rites such as key signing (Bagozzi & Dholakia, 2006; Fleming & Waguespack, 2007; O'Mahony & Ferraro, 2007).

Principle 6: C-forms that have formalized social outlets will have higher retention rates than those that do not.

All methods of organizing can be subject to misconduct of members that is based in either instrumental self-interested motivations or other normative factors (Vardi & Wiener, 1996). Although the first defining characteristic of informal peripheral boundaries of membership is a source of growth, C-forms must also find ways to filter disruptive contributions to maintain the culture and the quality of the C-from product. Unlike classic organizational forms, which can fire employees for misconduct, the informal boundaries

make such sanctions more difficult. In addition to selecting and retaining productive contributors, C-forms also need to find methods of stopping negative contributions (i.e., encouraging attrition among those who do not match the organization's culture). A contributor that continually violates the cultural norms of the C-form can damage the culture, spread negativity, cause confusion, cause distraction, sabotage, turn the product to their own benefit, and generally wreak havoc. One example of this type of problem was the defacing of Wikipedia, a C-form encyclopedia, by malicious contributors (Nuttall, 2005). Whereas organizations that rely on employment contracts might simply dismiss such individuals, C-forms, which incorporate voluntary labor, require other methods to remove disruptive members. These methods include formalized mentoring programs where contributions from a potentially negative source are screened carefully by senior members before inclusion, restriction of privileges, and outright banning from participation, which may take the form of a rite of degradation similar to firing in a classic organization (Trice & Beyer, 1993). The Wikipedia example highlights the importance of filtering disruptive contributors. In response to malicious posting of fictitious content, Wikipedia responded by instituting a set of review procedures controlled by the inner circle (Nuttall, 2005). These methods help ensure the quality of the end product of the C-form. These quality control mechanisms most directly affect the third defining characteristic of an information-based output. Low-quality outputs reduce acceptance, and as such,

Principle 7: The product of C-forms that have monitoring systems to check for abusive behavior by contributors will be more successful in the marketplace.

Principle 8: The product of C-forms that have sanctioning processes to ban contributions from certain individuals will be more successful in the marketplace.

FUTURE DIRECTIONS

The C-form architecture that we have described is characterized by informal peripheral boundaries, incorporation of voluntary labor, a focus on information-based output, and significantly open sharing of knowledge. We have discussed the ways in which we see this architecture as similar to and distinct from those already present in the lexicon of organizational

theory and used a culture frame to synthesize research findings regarding the structure and internal workings of communities that employ the C-form to suggest an initial set of principles. Although there has been substantial research on C-forms in recent years, being such a new development, there are many additional research avenues to pursue. In this section, we highlight some of the more promising areas.

Demography and the C-Form

One of the more intriguing areas for future research is the role of demography. Whereas the inner circle is expected to be homogenous with regard to relevant values, norms, and other aspects of the community culture, members of both the inner and outer circles may be demographically diverse. There is the potential for demographic diversity to be especially high in C-forms because they lack geographical or other boundaries for participation. Some members of the Python open source project web site have noted the important role of diversity with the statement, "The most important skill Python can teach is the delicate skill of working in a diverse group" (quoted in Ducheneaut, 2005, p. 339). At the same time, given the voluntary nature of contribution and the fact that individuals are drawn to similar others through homophily, diversity is by no means a given. In fact, a dramatic gender imbalance has been observed in the OSS domain, which is quite male dominated (Ghosh, Glott, Krieger, & Robles, 2002; Weiss, 2005).

Although the reliance on ICT and the relatively fluid boundaries it creates has the potential to increase the level of demographic diversity among members, this same fact may ease some of the difficulties typically associated with demographic heterogeneity. Unlike face-to-face communication, which highlights surface-level differences, electronic-mediated communication tends to reduce the salience of demographic differences (Barsness, Diekmann, & Seidel, 2005; Bhappu, Griffith, & Northcraft, 1997) and encourages individuals to move toward deeper-level communication focusing on content. Such deeper levels of understanding can lead to the disappearance of classic discriminatory outcomes (Petersen, Saporta, & Seidel, 2000; Seidel, Polzer, & Stewart, 2000). On the contrary, while demographic diversity could be reduced in salience, other kinds of diversity could become even more powerful. For example, one of the three kinds of diversity identified by Harrison and Klein (2007) is disparity, reflecting differences in power and resources. Because of the meritocratic nature of advancement on the C-form and the involvement of both volunteer and company-sponsored contributors, disparity could play a significant role in some C-forms.

While traditional organizations have been moving toward norms of inclusiveness and diversity to reduce negative outcomes of demographic diversity, C-forms should be able to do so even more effectively due to the reduced salience of demographic differences. Some OSS C-forms have purposely taken steps directed at reducing the gender imbalance (Qiu, Stewart, & Bartol, 2010). This area requires additional research both at the individual contributor level to determine awareness of other demographic profiles of co-contributors and at the community level to determine the effects of potential discriminatory outcomes. It is likely that both of these will be lower than in traditional organizations.

Early Stage Growth

As argued above, providing status-reinforcing rites and a forum for social interaction and trust building may help satisfy C-form members' motivations related to reputation and a sense of belonging and identification. Another important means of building a strong culture and enhancing member identification is to create a common enemy through a process of disidentification (Elsbach & Bhattacharya, 2001). This enemy can be a competing group, a social problem, an institution, or a political cause. Having an identifiable enemy may help delineate in-group and out-group boundaries thereby enhancing members' identification with the C-form (Turner, Hogg, Oakes, Reicher, & Wetherell, 1987). Uniting around the common purpose of defeating the enemy may also strengthen member's commitment to and identification with the overall culture. Having an identifiable enemy in the early stages of development can help in defining the C-form culture and identity. Given that the output of a C-form is, by definition, an information-based product, enemies may be found in traditional organizations that produce competing products, as with the Linux versus Microsoft example. There is substantial room for research understanding the role that common enemies play in early stage C-form growth. This research should focus on C-form inputs as it is most likely that C-forms with an identifiable "opponent" during the initial start-up will have more success in attracting and retaining contributors than those that do not.

Industry Level of Analysis

What will the world look like 10 years from now? We believe the appearance of the C-form has the potential to alter the organizational landscape in drastic ways, as it already has done in the software industry. What are the

characteristics of industries well suited to the C-form and how will those industries be altered if the architecture spreads? More specifically, what are the economic and social factors that foster the creation of C-form communities in an industry? Under what conditions are C-form communities likely to be more successful than other organizations in the same industry?

Given the global dispersion of members, the C-form would not be possible on such a wide basis without the drastic reduction in communication costs of the past two decades. The existence of a C-form depends on inexpensive and efficient communication. Traditional organizational forms came about to reduce coordination costs that open markets presented. Inexpensive and efficient communication is critical for allowing the participation and coordination of many widely dispersed individuals through a many-to-many communication structure. Such communication has become available through the Internet and wireless networks. These communication networks provide efficiency. Each member is responsible for his or her own communication costs, making communication essentially free for the organization. Before the drastic reduction in coordination costs, C-forms such as the OED took decades to get products completed – making them impractical for most applications.

Additional pressure to develop this architecture may have come from the constantly reducing cycle times involved in information products. The C-form allows product changes to be made virtually as the demand for them arises rather than having to go through a time-consuming bureaucratic process. It also allows for a virtually unlimited amount of highly skilled labor to focus on the most pressing problems. With new technologies and new demands constantly emerging, the bureaucracy involved in the response of more traditional organizational architectures does not allow enough speed to react. In the C-form, the consumers who need new products or features help create them, whereas in more traditional models, a company that has sold a product is less able and less motivated to respond quickly. This contrast parallels the contrasts between a free market (the C-form) and a centrally controlled economy (traditional hierarchical organization). Using the Linux-Microsoft comparison, demand is determined in the Linux C-form based on the individual needs of the broader market. On the contrary, Microsoft needs to create a long-term strategic plan, allocate production capacity to specific projects, and predict future needs.

With regard to diffusion outside the software industry, other information products are candidates for production by C-form. Advice and consulting services may lend themselves well to the C-form. In fact, we are now seeing numerous Internet communities, such as ABT, sprout up, which are utilizing the C-form. These range from counseling services to industry-specific

consumer watch groups. These C-form communities all have permeable peripheral boundaries, inner and outer circles, operate using public information disclosure, do not focus on economic exchange as currency, and take on new projects based on the interests of the members. Thus, we suggest that C-forms are likely to arise when the key resources in production are human resources (e.g., domain-specific ability and knowledge) and the primary output is information-based.

Traditional Architecture Competition

Clearly, the C-form is not ideally suited for all contingencies. A set of environmental conditions can still exist, which favor other architectures. Therefore, under what conditions might we expect C-form communities to outperform rivals? The potentially large number of people working on problems that are the most relevant to the market, combined with the fluid communication between all of the C-form contributors, leads us to expect that C-forms are more efficient than other organizations in getting market feedback. Given the speed with which they get feedback and the large number of minds that may be immediately applied to a problem, C-forms are able to respond more quickly than other architectures to changing conditions and environmental shocks. For these reasons, C-forms should be more successful than other architectures in very high-speed environments. However, "pure" C-forms may not necessarily be needed. Just as we explained above how Linux might be viewed as a "special case" of other architectures, it may be that organizations best characterized by other architectures can "borrow" aspects of the C-form architecture to create viable hybrids. In fact, we see this occurring already: many innovative for-profit companies are finding ways to leverage voluntary community contributions without taking on the other defining characteristics of the C-form. Examples include the Netflix Prize – Netflix created a contest with a $1M prize going to the first team that could produce a specified level of enhancement in one of Netflix's key algorithms (see www.Netflixprize.com). Innocentive (www.innocentive.com) provides a platform used by many traditional companies to run contests similar to the Netflix Prize.

CONCLUSION

This chapter has answered Schoonhoven's (2003) call by introducing a definition of a new architecture – the C-form – and discussing the issues this

architecture raises for researchers. This is a method of organizing based on the voluntary work of persons interested in creating an information-based product to be made available for public consumption. Its recent growth relies on the noneconomic motivations of some members and the drastic reduction in coordination and communication costs made possible by the Internet and wireless networks.

The structure of the economy is rapidly changing, and new methods of organization are emerging as the initial assumptions that led to the creation of traditional organizational architectures are violated. The C-form, as characterized in this chapter, expands the tools of organizational theorists to understand this new community architecture and promises to be an important subject for future research. We hope that the research questions presented here stimulate researchers to grapple with these issues that current organizational theories of traditional architectures are unable to fully address.

NOTES

1. See http://www.fsf.org/.
2. See the Open Source Initiative web site at www.opensource.org for a detailed definition of OSS.
3. Not all OSS is built by a C-form. In cases where open source software is built exclusively or almost exclusively by the employees of a single organization, such as MySQL (Capra, Francalanci, & Merlo 2008), we would not consider these to be examples of the C-form.
4. See http://us3.samba.org/samba/ml-etiquette.html for an example.

ACKNOWLEDGMENTS

We thank Zoe Barsness, Fabiola Bertolotti, Lisa Cohen, Janet Dukerich, Sanjay Gosain, Sirkka Jarvenpaa, Poonam Khanna, Karim R. Lakhani, Hank Lucas, Sally Maitlis, Siobhan O'Mahony, Ithai Stern, Jim Westphal, and seminar participants at the University of British Columbia for helpful comments and discussions on earlier drafts of this manuscript.

REFERENCES

ABellTolls. (2011). Archived press mentions. Retrieved from http://abtolls.com/information/archivedpress.html. Accessed on 11 May 2011.

Adler, P. S. (2001). Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, *12*(2), 215–234.

Ahmadjian, C. L., & Lincoln, J. R. (2001). Keiretsu, governance, and learning: Case studies in change from the Japanese automotive industry. *Organization Science*, *12*(6), 683–701.

Alge, B. J., Wiethoff, C., & Klein, H. J. (2003). When does the medium matter? Knowledge-building experiences and opportunities in decision-making teams. *Organizational Behavior and Human Decision Processes*, *91*(1), 26–37.

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

Barnard, C. I. (1938). *Functions of the executive*. Cambridge, MA: Harvard University Press.

Barsness, Z. I., Diekmann, K. A., & Seidel, M. D. L. (2005). Motivation and opportunity: The role of remote work, demographic dissimilarity, and social network centrality in impression management. *Academy of Management Journal*, *48*(3), 401–419.

Bartlett, C. A., & Ghoshal, S. (1993). Beyond the M-Form: Toward a managerial theory of the firm. *Strategic Management Journal*, *14*, 23–46.

Baum, J. A. C., Calabrese, T., & Silverman, B. S. (2000). Don't go it alone: Alliance network composition and startups' performance in Canadian biotechnology. *Strategic Management Journal*, *21*(3), 267–294.

Bechky, B. A. (2003). Sharing meaning across occupational communities: The transformation of understanding on a production floor. *Organization Science*, *14*(3), 312–330.

Beckman, C. M., & Haunschild, P. R. (2002). Network learning: The effects of partners' heterogeneity of experience on corporate acquisitions. *Administrative Science Quarterly*, *47*(1), 92–124.

Beyer, J. M., & Trice, H. M. (1987). How an organizations rites reveal its culture. *Organizational Dynamics*, *15*(4), 5–24.

Bhappu, A. D., Griffith, T. L., & Northcraft, G. B. (1997). Media effects and communication bias in diverse groups. *Organizational Behavior and Human Decision Processes*, *70*(3), 199–205.

Bretthauer, D. (2002). Open source software: A history. *Information Technology and Libraries*, *21*(1), 3–11.

Burt, R. S. (2004). Structural holes and good ideas. *American Journal of Sociology*, *110*(2), 349–399.

Capra, E., Francalanci, C., & Merlo, F. (2008). An empirical study on the relationship among software design quality, development effort, and governance in open source projects. *IEEE Transactions on Software Engineering*, *34*(6), 765–782.

Carroll, G. R., & Hannan, M. T. (2000). *The demography of corporations and industries*. Princeton, NJ: Princeton University Press.

Chengalur-Smith, S., & Sidorova, A. (2003). Survival of open-source projects: A population ecology perspective. In: *24th International Conference on Information Systems*, Seattle, WA (pp. 782–787).

Collins, D. (1995). A socio-political theory of workplace democracy: Class conflict, constituent reactions and organizational outcomes at a gainsharing facility. *Organization Science*, *6*(6), 628–644.

Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, *11*(2), 12231148.

Crowston, K., & Scozzi, B. (2002). Open source software projects as virtual organizations: Competency rallying for software development. *IEE Proceedings Software*, *149*(1), 3–17.

Cyert, R. M., & March, J. G. (1963). *A behavioral theory of the firm*. Englewood Cliffs, NJ: Prentice Hall.

Dahlander, L., & O'Mahony, S. (forthcoming). Progressing to the center: Coordinating knowledge work organization science.

Davis, G. F., & Marquis, C. (2005). Prospects for organizational theory in the early twenty-first century: Institutional fields and mechanisms. *Organization Science*, 332–343.

DiBona, C., Ockman, S., & Stone, M. (1999). *Open sources: Voices from the open source revolution*. Sebastopol, CA: O'Reilly.

Dijksterhuis, M. S., Van den Bosch, F. A. J., & Volberda, H. W. (1999). Where do new organizational forms come from? Management logics as a source of coevolution. *Organization Science*, *10*(5), 569–582.

Ducheneaut, N. (2005). Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work*, *14*, 323–368.

Dyer, J. H., & Nobeoka, K. (2000). Creating and managing a high-performance knowledge-sharing network: The Toyota case. *Strategic Management Journal*, *21*(3), 345–367.

Elsbach, K. D., & Bhattacharya, C. B. (2001). Defining who you are by what you're not: Organizational disidentification and the national rifle association. *Organization Science*, *12*(4), 393–413.

Fang, Y., & Neufeld, D. (2009). Understanding sustained participation in open source software projects. *Journal of Management Information Systems*, *25*(4), 9–50.

Fink, M. (2003). *The business and economics of Linux and open source*. Upper Saddle River, NJ: Prentice Hall.

Fleming, L., & Waguespack, D. M. (2007). Brokerage, boundary spanning, and leadership in open innovation communities. *Organization Science*, *18*(2), 165.

Galbraith, J. R. (1973). *Designing complex organizations*. Reading, MA: Addison-Wesley.

Gallaugher, J. (2010). *Information systems: A manager's guide to harnessing technology (version 1.0)*. Irvington, NY: FlatWorld Knowledge.

Gallivan, M. J. (2001). Striking a balance between trust and control in a virtual organization: A content analysis of open source software case studies. *Information Systems Journal*, *11*(4), 277–304.

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). *Free/libre open source software: Survey and study, workshop on advancing the research agenda on free/open source software*. Brussels, Netherlands: International Institute of Infonomics, University of Maastricht.

Godfrey, M. W., & Tu, Q. (2000). Evolution in open source software: A case study. In: *International Conference Software Maintenance*, San Jose, CA (pp. 131–142). Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber = 883030

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

Griffith, T. L., Sawyer, J. E., & Neale, M. A. (2003). Virtualness and knowledge in teams: Managing the love triangle of organizations, individuals, and information technology. *MIS Quarterly*, *27*(2), 265–287.

Hahn, J., Moon, J. Y., & Zhang, C. (2008). Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, *19*(3), 369–391.

Harrison, D., & Klein, K. (2007). What's the difference? Diversity constructs as separation, variety, or disparity in organizations. *Academy of Management Review*, *32*(4), 1199–1228.

Hars, A., & Ou, S. (2002). Working for free? Motivations for participating in open source projects. *International Journal of Electronic Commerce*, *6*(3), 25–39.

HB2914. (2000). Washington State House of Representatives-An Act Relating to Telecommunications Customers: House Committee On Technology, Telecommunications, & Energy. Retrieved from http://www.leg.wa.gov/pub/billinfo/1999-00/Htm/Bills/House%20Bills/2914.htm. Accessed on 11 May 2011.

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

Ilgen, D. R., Hollenbeck, J. R., Johnson, M., & Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology*, *56*(1), 517.

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and trust in global virtual teams. *Organization Science*, *10*(6), 791–815.

Kanter, R. M. (1979). Power failure in management circuits. *Harvard Business Review*, *57*(July–August), 65–75.

Krishnamurthy, S. (2002). Cave or community? An empirical examination of 100 mature open source projects. *First Monday*, *7*(6). Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id = 667402

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: ''Free'' user-to-user assistance. *Research Policy*, *32*(6), 923.

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

Lawrence, P. R., & Lorsch, J. W. (1967). *Organization and environment*. Boston, MA: Harvard University Graduate School of Business Administration.

Lee, G. K., & 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*(6), 633–649.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The Journal of Industrial Economics*, *50*(2), 197–234.

Levy, S. (1984). *Hackers: Heroes of the computer revolution*. New York, NY: Dell.

Ljungberg, J. (2000). Open source movements as a model for organizing. *European Journal of Information Systems*, *9*(4), 208–216.

Lounsbury, M., Kelty, C., Yavuz, C., & Colvin, V. (2009). Towards open source nano: Arsenic removal and alternative models of technology transfer. *Advances in the Study of Entrepreneurship, Innovation, and Economic Growth*, *19*, 51–78.

Martins, L. L., Gilson, L. L., & Maynard, M. T. (2004). Virtual teams: What do we know and where do we go from here? *Journal of Management*, *30*(6), 805.

McCarthy, J. D., & Zald, M. N. (1977). Resource mobilization and social-movements: Partial theory. *American Journal of Sociology*, *82*(6), 1212–1241.

McMillan, R. (1999). Kernal driver. *Linux Magazine*, *1*(5), 39.

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*(3), 309–346.

Moglen, E. (1999). Anarchism triumphant: Free software and the death of copyright. *First Monday*. Retrieved from http://www.blagblagblag.org/anarchism/index.html

Mozilla. (2005). Press release, Retrieved from http://www.mozilla.org/press/mozilla-2005-10-19.html. Accessed on 11 May 2011.
Naquin, C. E., & Paulson, G. D. (2003). Online bargaining and interpersonal trust. *Journal of Applied Psychology*, *88*(1), 113–120.
Nuttall, C. (2005). Wikipedia to restrict access to encyclopedia. Retrieved from http://news.ft.com/cms/s/98413d60-6ff1-11da-a1f7-0000779e2340.html. Accessed on 11 May 2011.
O'Mahony, S. (2007). The governance of open source initiatives: What does it mean to be community managed? *Journal of Management & Governance*, *11*, 139–150.
O'Mahony, S., & Bechky, B. A. (2008). Boundary organizations: Enabling collaboration between unexpected allies. *Administrative Science Quarterly*, *53*, 422–459.
O'Mahony, S., & Ferraro, F. (2007). The emergence of governance on an open source project. *Academy of Management Journal*, *50*(5), 1079–1106.
Ouchi, W. G. (1980). Markets, bureaucracies, and clans. *Administrative Science Quarterly*, *25*(1), 129–141.
Petersen, T., Saporta, I., & Seidel, M. D. L. (2000). Offering a job: Meritocracy and social networks. *American Journal of Sociology*, *106*(3), 763–816.
Podolny, J. M., & Page, K. L. (1998). Network forms of organization. *Annual Review of Sociology*, *24*, 57–76.
Powell, W. W. (1990). Neither market nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, *12*, 295–336.
Qiu, Y., Stewart, K., & Bartol, K. (2010). Joining and socialization in open source women's groups: An exploratory study of KDE-women. In: *The proceedings of the 6th international conference on open source systems*, Notre Dame, IN, May 30, 2010. Retrieved from http://www.springerlink.com/content/1868-4238/
Raasch, C., Herstatt, C., & Balka, K. (2009). On the open design of tangible goods. *R&D Management*, *39*(4), 382–393.
Raymond, E. S. (2000a). The cathedral and the bazaar. Retrieved from http://www.catb.org/~esr/writings/cathedral-bazaar. Accessed on 11 May 2011.
Raymond, E. S. (2000b). The halloween documents: An appreciation. Retrieved from http://www.catb.org/~esr/not-the-osi/halloween-rant.html. Accessed on 11 May 2011.
Raymond, E. S. (2000c). The magic cauldron. Retrieved from http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/. Accessed on 11 May 2011.
Roberts, J., Hann, I. H., & Slaughter, S. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science*, *52*(7), 984–1000.
Scacchi, W. (2010). Collaboration practices and affordances in free/open source software development. In: I. Mistrík, J. Grundy, A. van der Hoek & J. Whitehead (Eds.), *Collaborative software engineering* (pp. 307–328). New York, NY: Springer.
Schoonhoven, C. B. (2003). Open source software and the 'private-collective' innovation model: Issues for organization science. *Organization Science*, *14*(2), 208208.
Seidel, M. D. L., Polzer, J. T., & Stewart, K. J. (2000). Friends in high places: The effects of social networks on discrimination in salary negotiations. *Administrative Science Quarterly*, *45*(1), 1–24.
Shah, S. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, *52*(7), 1000–1014.
Shapiro, C., & Varian, H. (1999). *Information rules: A strategic guide to the network economy*. Boston, MA: Harvard Business School Press.

Simon, H. A. (1945). *Administrative behavior*. New York, NY: Macmillan.
Steele, J. (2001). Are you well-connected? Retrieved from http://web.archive.org/web/20010823151345/http://chicagotribune.com/business/chi-0108210195aug21.story. Accessed on 21 August 2001.
Stewart, D. (2005). Social status in an open-source community. *American Sociological Review*, *70*(5), 823–842.
Stewart, K. J., Ammeter, A., & Maruping, L. (2006). Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research*, *17*(2), 126–144.
Stewart, K. J., & Gosain, S. (2006a). The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly*, *30*(2), 291–314.
Stewart, K. J., & Gosain, S. (2006b). The moderating role of development stage in affecting free/open source software project performance. *Software Process Improvement and Practice Journal*, *11*(2), 177–191.
Taylor, F. W. (1911). *The principles of scientific management*. New York, NY: Harper.
Trice, H. M., & Beyer, J. M. (1984). Studying organizational cultures through rites and ceremonials. *Academy of Management Review*, *9*(4), 653–669.
Trice, H. M., & Beyer, J. M. (1993). *The cultures of work organizations*. Englewood Cliffs, NJ: Prentice Hall.
Turner, J. C., Hogg, M. A., Oakes, P. J., Reicher, S. D., & Wetherell, M. S. (1987). *Rediscovering the social group: A self-categorizing theory*. Oxford, UK: Blackwell.
Valloppillil, V., & Cohen, J. (2000). The Halloween documents. Retrieved from http://www.catb.org/~esr/halloween. Accessed on 11 May 2011.
Vardi, Y., & Wiener, Y. (1996). Misbehavior in organizations: A motivational framework. *Organization Science*, *7*(2), 151–165.
von Hippel, E. (2001). Learning from open source software. *Sloan Management Review* (Summer), 82–86.
von Hippel, E., & von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, *14*(2), 209–223.
von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, *32*(7), 1217.
Weaver, G. R., & Agle, B. R. (2002). Religiosity and ethical behavior in organizations: A symbolic interactionist perspective. *Academy of Management Review*, *27*(1), 77–97.
Weiss, T. R. (2005). *Panel: Open-source needs more women developers*. Portland, OR: Computerworld.
West, J., & O'Mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and Innovation*, *15*(2), 145–168.
Williamson, O. E. (1975). *Markets and hierarchies: Analysis and antitrust implications*. Stanford, CT: Sineaur.
Winchester, S. (1998). *The surgeon of crowthorne: A tale of murder, madness, and the love of words*. London, England: Viking.
Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. *Proceedings of the 2003 international conference on software engineering*, Portland, Oregon (pp. 419–429). Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1607228