The Dissertation Committee for Jonathan Paul Sims certifies that this is the approved version of the following dissertation:

INTERACTIVE ENGAGEMENT WITH AN OPEN SOURCE COMMUNITY: A STUDY OF THE RELATIONSHIPS BETWEEN ORGANIZATIONS AND AN OPEN SOURCE SOFTWARE COMMUNITY

Committee:

Craig Crossland, Co-Supervisor

Andrew D. Henderson, Co-Supervisor

Ethan Burris

James Fredrickson

Melissa Graebner

Joel West

INTERACTIVE ENGAGEMENT WITH AN OPEN SOURCE COMMUNITY: A STUDY OF THE RELATIONSHIPS BETWEEN ORGANIZATIONS AND AN OPEN SOURCE SOFTWARE COMMUNITY

by

Jonathan Paul Sims, B.A.; M.B.A.; M.S.Man.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2013

Acknowledgements

This dissertation was made possible thanks to the help and support of a number of people, both inside and outside of academia.

First, I would like to thank my committee Co-chairs, Craig Crossland and Andy Henderson. Their guidance throughout this process has been invaluable.

Each member of my dissertation committee provided valuable assistance with various aspects of this research. Ethan Burris helped with survey methodology and wording of hypotheses. Department Chair James Fredrickson provided funding support and helpful encouragement. Melissa Graebner helped frame the inductive pilot study and provided constant encouragement and support. Finally, Joel West shared his vast knowledge on the areas of innovation, open innovation, and software development. In sum, each committee member's unique contributions have made this a stronger document, and I learned a great deal from each one.

I would also like to thank the Herb Kelleher Center for Entrepreneurship at the McCombs School of Business for their generous financial support of this research, as well as both Craig Crossland and Andy Henderson for their individual financial support.

Finally, I would like to thank my parents, Dr. Henry Paul Sims, Jr. and Dr. Laura Smail Sims, for providing patient, helpful guidance and encouragement. Without their love and support, this document would not exist. It is an honor for me to follow them into academia.

Interactive Engagement With An Open Source Community: A Study Of The Relationships Between Organizations And An Open Source Software Community

Jonathan Paul Sims, Ph.D.

The University of Texas at Austin, 2013

Supervisors: Craig Crossland and Andrew D. Henderson

This dissertation theoretically develops and empirically tests a model of interactive firm engagement with an open source software community. An inductive pilot study and subsequent interview analysis suggest that the nature of the relationship between a firm and an open source community varies in the degree by which a firm both "takes from" and "gives to" the community. I propose that a firm will experience direct effects from both giving to and taking from the community, and further propose that the interaction of these two behaviors, which I call interactive engagement, will lead to three firm-level consequences: an increase in the number of new products, higher levels of incremental (as opposed to radical) innovation, and shortened development and debug time. I test these hypotheses using regression analysis of questionnaire responses collected from 250 organizations that work with a popular open source software community.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Scholars of strategic management have long debated whether the origins of sustainable competitive advantage lie within the firm (e.g. Barney et al., 2001; Peteraf, 1993; Wernerfelt, 1984), or are predominantly shaped by external forces (Demsetz, 1973; Porter, 1980). Innovation, defined here as *the use, combination, or accumulation of resources and knowledge used in the development and introduction of new products and services* (Dodgson et al., 2002; Murray & O'Mahony, 2007) has become a significant part of this discussion. Specifically, researchers debate whether innovations are best developed within a firm's boundaries, or are best acquired through interaction with outside environments. Early innovation scholars, observing that many firms of their time relied on their own research and development efforts, focused their study on internal sources. More recent research has shifted toward an exploration of the influence of external sources (e.g. Powell et al., 1996; Stuart, 2000). Most research examining external sources of innovation examined the ways firms interact with each other. Examples include acquisitions (e.g. Benson & Ziedonis, 2009; Graebner & Eisenhardt, 2004), joint ventures (e.g. Kogut, 1988), and alliances (e.g. Shah & Swaminathan, 2008; Stuart, 2000).

A smaller but growing body of work has investigated the influence of external communities, which are fundamentally different organizational forms than firms, marked by egalitarian membership and norms of reciprocity. O'Mahony and Lakhani argue that "communities play an underappreciated role in organizational theory—critical not only to occupational identity, knowledge transfer, sense-making, social support, innovation, problem-solving and collective action but, enabled by information technology,

increasingly providing socio-economic value—in areas once inhabited by organizations alone" (2011: 2). Defined here as, "voluntary collections of actors whose interests overlap and whose actions are partially influenced by this perception" (O Mahony & Lakhani, 2011: 4), communities have been shown to facilitate the flow of knowledge (Brown & Duguid, 1991, 2001) and alliance formation (Rosenkopf et al., 2001). The most frequently studied communities include user communities, brand communities, trade and industry groups (e.g. Barnett et al., 2000; Lenox, 2006), and the focus of this dissertation, open source software communities.

Open source software communities are associations of volunteers who work collaboratively, often despite geographical separation, to develop and advance a software platform that anyone, regardless of community membership, may acquire and use free of charge. In exchange, programmers are asked to improve upon it and make their improvements available to others (Coar, 2006; 2008). In recent years, open source software has enjoyed surging popularity. As evidence, consider that the open source distributor SourceForge listed a mere 10,000 open source projects in 2001 (Murray & O'Mahony, 2007). As of December 2012, that number had grown to over 324,000, with an average rate of over 4 million downloads per day (Sourceforge.net).

Many early open source pioneers argued that software should be free for all to use. The most notable was former MIT programmer Richard Stallman, who publicly argued, "Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way" (Stallman, 1983). Yet despite such anticorporate origins, open source software has become a medium that firms of all sizes may use to develop business models and innovative

products and services. Consider two companies that have both leveraged the Drupal open source software community, the setting for my research. In the first example, website development firm AcademicWeb only embraced open source after spending nine successful years building websites for universities using (and paying for) proprietary software. Not only did the firm benefit by replacing expensive software with a free open source alternative, but it also embraced community norms by contributing back both software and assistance. CoachX provides a second and very different example. Launched with the help of the original founder of the open source community, CoachX was able to raise over $12 million in venture capital funding to help advance the use of Drupal among large corporations, a market once served almost exclusively by providers of expensive "enterprise-grade" solutions. To do this, CoachX not only devoted itself to helping improve the Drupal software, but also actively supported numerous firms within the community. The experiences of AcademicWeb and CoachX, as well as several other firms (identified in Table 4), are further explored as part of a pilot study described in the third chapter of this dissertation.

The influence of open source stretches beyond the entrepreneurial firms studied in this dissertation. Even industry giants have ignored open source at their peril. Consider an anecdote from Microsoft. In 2007, CEO Steve Balmer stated, "We've worked very hard on making the value of a commercial company surpass what the open source community can deliver, because frankly, it's not a business model we can embrace. It's inconsistent with shareholder value" (Wilcox, 2007). Two years later, Microsoft funded and launched the CodePlex Foundation to promote open source software, and appointed their former Senior Director of Platform Strategy as its interim president (Edge, 2009).

The study of firm engagement with an open source community also has theoretical merit, and through this research, I seek to contribute to strategic management theory in three ways. First, by exploring exchange relationships between firms and an external community, this dissertation answers Chesbrough and Appleyard's (2007) call for the development a theory explaining how firms can create value through open models without abandoning traditional views of how value can be captured and maintained within the firm. In other words, I hope to clarify the mechanisms by which firms participate in these communities and the consequences of that participation. In the fourth chapter, I describe three processes central to open innovation: taking from the external environment, giving to an external environment, and processes that consider the interaction of the two, which I call interactive engagement.

Second, I hope to advance management scholars' understanding of community norms. Known for collective authority, voluntary participation, and egalitarian or meritocratic norms among members (Rothschild-Whitt, 1979), communities represent a fundamentally different organizational form from firms. As O'Mahony and Lahkani (2011) point out, these organizations have an increasingly important role in management science.

Third, by using open source communities as my empirical context, I hope to deepen understanding of these communities and their impact on firm strategy. Most research of communities has examined the motives of individual contributors (e.g. Federman, 2006; Fleming & Waguespack, 2007; Lakhani & von Hippel, 2003; Roberts et al., 2006; West & O'Mahony, 2008). With a few notable exceptions (e.g. Bonaccorsi et al., 2006; Fosfuri et al., 2008; Henkel, 2006; Shah, 2006), the means by which firms

interact with open source communities has received relatively little attention. This research adds to that body of work in two ways; first, by validating a measure of interactive engagement with an open source community, and second, by testing the influence of interactive engagement on several dependent variables.

This study seeks to contribute to these discussions by answering the research question: *How does variation in the type and intensity of participation in an external open source software community affect outcomes related to firm-level innovation?* To answer this question, I theoretically explore and empirically study firm participation in open source communities by classifying two firm-level behaviors: *giving to* and *taking from* the community. I label the interaction of these behaviors as "interactive engagement," and propose testable hypotheses to examine the consequences of this phenomenon.

This dissertation is organized into seven chapters. Following this introduction, the second chapter reviews relevant literature, providing a historical retrospective of research on communities, open source, innovation, and the open innovation perspective in management theory. The third chapter presents an inductive pilot study of the phenomena under investigation using analysis of original interviews with firms actively engaged in an open source community. The fourth chapter provides a theoretical view of such engagement between a firm and an open source community, and presents several testable hypotheses. The fifth chapter describes methods that are used to empirically test my hypotheses. The sixth chapter presents results of my hypothesis tests, and the seventh chapter provides discussion and concluding remarks.

In summary, the overall purpose of this dissertation is to contribute to strategy and innovation research by investigating the nature of exchange relationships between firms and communities, and the consequences of these relationships on individual firms.

Chapter 2: Literature Review

This literature review is divided into three sections. I begin by reviewing the research on external communities, with a particular emphasis on open source communities and the interactions between open source communities and traditional firms. I then provide a broad historical overview of innovation research, concentrating on work examining internal versus external sources of innovation. Finally, I review related work from the open innovation perspective and identify research opportunities.

External Communities in Management Research

The study of firm interactions with external communities has both theoretical and phenomenological rationale. By "external," I mean communities that are external to the boundaries of the firm. External communities are theoretically interesting because their presence is not easily explained by leading theories of the firm, which predominantly suggest that firms provide vertical and hierarchical control of production when the market-based mechanisms prove insufficient (Williamson, 1991). Communities are an example of what Williamson would call a "hybrid mode," such as "long-term contracts, franchising, joint ventures, and the like—that are located between markets and hierarchies" (1991: 80). Communities also differ from firms by constituting a "collectivist-democratic organization," which Rothschild-Whitt (1979) argued differed from bureaucratic organizations (i.e., firms) along several dimensions. Communities, like her ideal type of collectivist-democratic organizations, are based on collective authority and participation, informal relationships and egalitarian norms. Finally, communities also embody a "network form of organization," which Powell (1990) argued were separate from either markets or hierarchies.

The theoretical perspectives above treat "communities" as a one-dimensional type of organizational form, but communities vary in both form and function. Management researchers have divergent definitions of the word "community," and that definition has shifted over time. In early definitions, geographic proximity among members was the point of common agreement (Hillery, 1955), but the communities common in current management research are typically bound by technology, not geography. Recently, West and Lakhani (2008) were explicit about the need for researchers to clarify the construct of community, pointing out that literatures overlap in their descriptions of community types. I define a community in accordance with O'Mahony and Lakhani, as, "voluntary collections of actors whose interests overlap and whose actions are partially influenced by this perception" (2011: 4).

Four different types of external communities appear most frequently in management literature: user communities, brand communities, trade and industry groups, and open source communities. I define the first three types here, and cover open source communities in depth in the next section. First, *user communities* are typically informal collections of individuals with a desire to share enthusiasm or knowledge about a specific topic or product (Hienerth & Lettl, 2011). In his study of a nascent sporting goods industry, Hienerth (2006) documented how firms participated in a user community of product enthusiasts to identify and test ideas for new kayaking products. Many user communities, however, share information between members in a way that does not involve firms, and most studies of user communities focus on user-to-user interactions rather than community–firm interactions. One example is Lakhani & von Hippel (2003), who studied user-to-user assistance within the Apache open source project that was

crucial to IBM's e-commerce strategy (cf. West, 2003). User communities can incubate the creation of spin-off industries and entrepreneurial ventures (Hienerth, 2006). Brand communities are a "specialized, non-geographically bound community, based on a structured set of social relationships among admirers of a brand" (Muñiz & O'Guinn, 2001: 412). Trade and industry associations are voluntary communities of businesses working to achieve some common economic or technological objective. This category includes standard-setting organizations, voluntary nonmarket institutions that enable coordination to achieve technical operability (Fleming & Waguespack, 2007; Rysman & Simcoe, 2008; Waguespack & Fleming, 2009).

Regardless of their type, external communities share certain traits. Communities typically have meritocratic governance processes that allow more active and motivated participants to seek and obtain leadership roles. Communities have voluntary membership structures. Typically, communities control what they produce, be it software (in the case of open source communities) or policies (in the case of trade associations) (Demil and Lecocq, 2006; O'Mahony, 2003; O'Mahony and Ferraro, 2007; West and O'Mahony, 2008). Finally, community members characteristically share a common identity and share a sense of belonging with each other (von Hippel 2007).

Open Source Communities in Management Research

Open source communities, the setting for this dissertation research, consist of geographically distributed volunteers working together on a software product. There are important fundamental differences between open source communities and traditional firms. Scott's (2002) typology classifies organizations as rational, natural, or open systems. Using his typology, traditional firms are best described as rational systems: they

have specific goals (e.g., shareholder value) and formalized structures (e.g., hierarchy and defined job responsibilities). Open source communities, in contrast, are best described as open systems—structures of "interdependent activities linking shifting coalitions of participants" (Baum & Rowley, 2002: 2–3). In autonomous open source communities—those not sponsored or controlled by a firm (cf. West & O'Mahony, 2008)—there is often ambiguity surrounding shared goals and an informal governance structure compared to what is common in traditional firms.

Table 1 lists five features that differentiate traditional firms from open source communities. First, traditional firms are highly selective in terms of accepting and retaining members (i.e., their employees), while membership in open source communities is typically open to anyone. Traditional firms also use legally binding and formal agreements that define relationships between the organization and their employees. In contrast, open source communities have no formal agreements. Members are connected by far more egalitarian norms that those common in private software firms, leading some to suggest that open source has more in common with scientific communities than with private firms (Kogut & Metiu, 2001).

In traditional firms, the owners are the shareholders, represented by an elected Board of Directors, whose mission involves maintaining the firm's existence and, typically, providing some form of value to the shareholders. Open source communities are not characterized by such formality of role or objective. While there may be a leadership structure at the very top of the community, the key stakeholders are the members themselves.

Traditional firms most often seek to create intellectual capital internally, codify it through intellectual property (IP) protection, and defend it legally within legal systems. Conversely, autonomous open source communities use governance policies and licensing rules that promote continued sharing and development. By design, these communities keep their processes of innovation public. Work is distributed to anyone who sees value in its use. Where private companies seek to protect their intellectual property, open source communities avoid private ownership. By doing so, however, open source communities create what may be a risk to their continued existence. With no means to continually appropriate value from their creations, open source communities are at a constant risk of fading into obscurity and "forking," a process where members apply their knowledge to new ventures rather than the existing concerns of the larger community (Von Krogh & Von Hippel, 2006). Firms avoid such forking by both incentivizing organizational members to stay within the firm and legally prohibiting them from employing organizational property for employees' own ventures.

Bonaccorsi and colleagues (2006) called open source a "radical innovation" because of its demonstrated ability to disrupt the software industry. Large-scale open source software projects have indeed had a significant impact on technology companies. The four most prominent include the Linux open source operating system, Apache Web-server software, the MySQL database, and the PHP programming language. Together, these four open source software systems, known collectively as "LAMP," provide an ecosystem that enables the creation and hosting of dynamic Internet websites with little involvement from traditional "closed" software suppliers (West, 2007).

The implication of open source to large firms was investigated by West (2003), who examined how three technology leaders—Apple, IBM, and Sun—each used a mix of proprietary and open software to advance their strategic objectives. He concluded that a three-stage evolutionary process led these software leaders toward integrating open source: firms that began as proprietary software leaders eventually adopted open standards, which in turn led to the acceptance and inclusion of open source code. Recently, West and O'Mahony (2008) compared twelve open source projects that were either sponsored or founded by traditional firms. When firms designed clear production processes, implemented community governance, and clarified how intellectual property would be allocated, these sponsored communities were more successful.

To consider the future of open source, Bonaccorsi and Rossi (2003) used a simulation model to investigate adoption of both open and proprietary software standards. They concluded that both would continue to coexist, except when there existed considerable initial opposition to open source. Their work suggested that traditional vendors of proprietary software should take the threat of open source software seriously: "Given a distribution of beliefs biased towards open source and in the absence of incumbent advantages the only way for commercial software to control the market is to engage in a fierce quality competition supported by massive R&D efforts" (2003: 1254).

Early research on open source software focused primarily on the motivations of individuals who volunteer for these communities (Kogut & Metiu, 2001). While previous literature had assumed that intrinsic motivations underlie participation (Mockus et al., 2002; Roberts et al., 2006; Von Hippel, 2001), Roberts and colleagues (2006) did not find a significant relationship between intrinsic motivation and participation in the community.

Instead, multiple motivations were in play, but those that were extrinsic in nature (e.g., financial reward) did not crowd out those that were intrinsic.

Most open source communities are managed by a collective and codified governance structure (O'Mahony, 2007). A second and related stream of research explores how these governance structures of open source communities influence individual participation. For example, Roberts et al. (2006) concluded that Apache's meritocratic governance structure was effective at promoting individuals based on their contribution to the community.

FIRM INTERACTIONS WITH OPEN SOURCE AND OTHER COMMUNITY FORMS

While most, but not all, open source software communities remain independent entities separate from firms, they are fertile grounds for firm participation, and numerous studies have shown that firms can create business models using the software provided by open source communities (e.g. Federman, 2006; Fleming & Waguespack, 2007; Hars & Qu, 2002; Lakhani & von Hippel, 2003; Lakhani & Wolf, 2005; Roberts et al., 2006; von Krogh et al., 2003; Zeitlyn, 2003). Most of these studies suggest that firms develop a "hybrid" approach combining open source software with either proprietary software or complementary products such as hardware (Fosfuri et al., 2008). Henkel (2006) found that firms gave about half of their code to the open source community, but that there was a high degree of variation. Smaller firms were found to give back more, a result that the author attributed to resource scarcity.

Dahlander and Magnusson (2006) conducted case analyses of four companies whose business models depended upon the success of products developed in part by open

source software communities. They concluded that such firms must excel in three areas: accessing the open source community as a means to extend their resource base, aligning the firm's strategy with the interests of the community members, and assimilating the contributions of community members into firm products. Dahlander and Wallin (2006) found that community members who were employed by firms acted more strategically in these communities, communicating more frequently with high-status members of the community.

A review of empirical management literature on external communities suggests that by engaging these communities, firms may acquire resources, introduce innovative ideas, build influence, and develop social capital. Below, I consider each of these outcomes in turn.

Resources

Access to resources appears to be one of the principal drivers of firm participation in open source communities. Here, I define resources in line with Noda and Bower (1996), as a valuable asset that may be acquired and used by the firm. When working with an open source community, firms are able to easily and freely download the community's core software. So long as their use is consistent with the community's licensing structure, these firms may then use the software as they see fit. (Bonaccorsi et al., 2006; Dahlander, 2007; Dahlander & Wallin, 2006). Doing so has been shown to lower a firm's own development (Bonaccorsi et al., 2006; Samuelson, 2006) and production costs (Harhoff et al., 2003). Communities can also leverage the membership of open source communities for both support (Lerner & Tirole, 2002), and expertise (Gruber & Henkel, 2006). In several papers, community members have located and fixed

flaws in a company's product (Hienerth & Lettl, 2011; Raymond, 2001), such as software bugs (Henkel, 2006). In another study, community members helped to create product prototypes (Hienerth & Lettl, 2011). While the availability of these resources may appear to benefit smaller and resource-constrained firms, larger firms stand to gain as well. For example, West (2003) explained how industry leaders Apple and IBM both incorporated open source software into their own products.

Innovative Ideas

External environments can be a valuable source of knowledge (Rigby & Zook, 2002) that contributes to firm innovation (e.g. Pittaway et al., 2004). I define innovation in line with prior work as *the use, combination, or accumulation of resources and knowledge used in the development and introduction of new products, and/or services to market* (cf. Dodgson et al., 2002). In strategic management literature, innovation is considered an increasingly important determinant of how firms attain sustainable competitive advantage (Franko, 1989).

Powell et al. (1996) showed that in complex industries like information technology, collaborative networks provide firms with knowledge essential to innovation. In their words, "networks of collaboration provide entry to a field in which the relevant knowledge is widely distributed and not easily produced inside the boundaries of a firm or obtained through market transactions" (139). Open source communities can act as such a network of collaboration, providing firms with a rich source of novel ideas. Bonaccorsi et al. (2006) argue that active participation in the community "is likely to be an important asset for firms" because it provides information about "products, services, customers, and the eventual opening of new product niches" (1088). Research that examines firm–

community interaction offers several examples. For example, community members in user or open source communities are often hobbyists or enthusiasts who are willing to share relevant knowledge (Jeppesen & Frederiksen, 2006). Research on brand communities shows that firm interaction with these communities can provide firms with valuable insights on customer behavior and ideas useful for product improvements (Füller et al., 2008).

Influence

Through participation in external communities, firms gain influence. Firm participation in standard-setting communities is often driven by a desire to influence adoption of a chosen technology or standard (Garud et al., 2002). This medium can be particularly important for firms seeking to commercialize new technology or address technical challenges inherent in newer markets (Snow et al., 2011). If successful, such community endorsement can enable a firm to sell complementary assets or services that depend on the standard (Bonaccorsi et al., 2006; Henkel, 2006; von Hippel & von Krogh, 2003). But again, large firms can leverage these communities for their own purposes. West (2003) shows how established firms Apple, Sun, and IBM each embraced different open source communities as a hedge against the common threat posed by industry leader Microsoft.

Social Capital

By participating in external communities, firms can also acquire social capital (Dahlander & Magnusson, 2005). Research suggests that this may happen in three ways. For younger firms, community participation allows firms access to others and increases the odds of alliance formation (Rosenkopf et al., 2001). Second, firms may join

communities to demonstrate positive citizenship behavior (Stam, 2009). Finally, external communities provide all firms with visibility, enabling them to showcase skills (Henkel, 2006, 2009) and increase both status (Rosenkopf et al., 2001) and perceived legitimacy among peers (Dahlander & Wallin, 2006; Fleming & Waguespack, 2007; Lerner & Tirole, 2002; Stam, 2009).

INNOVATION IN MANAGEMENT LITERATURE

Through innovation, firms create new products and services without direct competition (Kim & Mauborgne, 1999; Schumpeter, 1934). This allows firms to differentiate themselves, and in doing so, escape cost-based competition. Research in economics emphasizes other benefits: innovation drives down internal costs of development and increases the price that may be charged in the market. In strategic management literature, innovation is considered an increasingly important determinant of how firms attain sustainable competitive advantage (Franko, 1989). This connection has been strengthened by current trends, such as the growth of information technology (Hitt et al., 1998) and increased market speed (Hidalgo & Albors, 2008).

Broad use of the term "innovation" sets it apart from much of the vernacular in strategic management. Innovation is a popular topic not only in scholarly journals, but also in practitioner publications and the popular business press. Perhaps as a consequence of such widespread use, scholars have not arrived at a single definition. In fact, innovation literature has been deemed a "fragmented corpus [where] scholars from a diversity of disciplinary backgrounds adopt a variety of ontological and epistemological positions to investigate, analyze and report on a phenomenon that is complex and multidimensional" (Adams et al., 2006: 22).

Definitions of innovation typically fall into one of two groups: those describing either a process or a product. The process perspective considers how a firm recombines inputs, such as existing knowledge and technology (Enkel et al., 2009; Murray & O'Mahony, 2007). A process may focus on problem solving (Dosi, 1982) or learning (Cohen & Levinthal, 1990). Scholars of the process tradition typically describe a phenomenon in which one actor innovates, resulting in some effect on the external environment (Gupta et al., 2007) that may include competitors, an entire industry, or even society at large. The product perspective focuses on outcomes, and typically new products. According to Tushman and Moore (1982), this perspective implies an important role for the firm. A recent survey by Linder et al. (2003) suggests that practitioners' views fall in line with the product perspective, and they consider innovation in terms of new products rather than business process or technological developments.

The definition I use in this dissertation draws from both the process and product perspectives. I apply two boundary conditions. First, I consider the results of innovation to be products and/or services released to market. This excludes innovation outcomes such as internal administrative changes, as well as any unreleased products. Secondly, and consistent with work on open innovation, I emphasize firm actions, as I see the purposeful use of innovation as a means for managers to influence the fate of their firm.

Chronology of Innovation Research

The most prominent early scholar of innovation was Austrian economist Joseph Schumpeter, whose contributions address the influence that both young and established firms have on the state of innovation. In *The Theory of Economic Development* (1934),

Schumpeter emphasized the role that newer firms have in influencing innovation paradigms. Through a process of "creative destruction," he proposed that younger, newer firms succeed when they develop innovations that disrupt the status quo of the market and weaken the positions of established players. Here, Schumpeter emphasized the importance of entrepreneurial vision and its ability to create shocks to the market that disrupt competitors and provide the innovator with extraordinary returns (Conner, 1991; Malerba & Orsenigo, 1995; Nelson & Winter, 1982). Since Schumpeter, scholars have associated innovation with a change to the status quo. For example, Tushman and Anderson (1986) found that breakthrough innovations, which they called "technological discontinuities," significantly influenced the larger environment, creating new opportunities for some firms and heightened ambiguity for others.

In *Capitalism, Socialism and Democracy* (1942), Schumpeter explored how established firms, by creating institutionalized processes of research and development, created barriers to entry for smaller firms (Malerba & Orsenigo, 1995). In this perspective, firms created value through large-scale actions, not incremental changes (Conner, 1991). Schumpeter asserted that innovation increased under two primary circumstances: market concentration and with the growth in firm size. Despite the fact that theories relating market concentration to innovation have spawned decades of research, empirical findings are mixed (Cohen & Levin, 1989). Schumpeter argued that innovation would increase with firm size because larger firms are able to devote more resources to R&D efforts (i.e., firm scope) and complement those efforts with other activities. However, as Ahuja et al. (2008) point out, size may also stifle innovation through bureaucracy and limiting incentives to individual inventors. Empirical findings

relating simple definitions of size to innovation have also been inconclusive (2008). Regardless, the significance of Schumpeter's work cannot be overstated. The number of empirical articles testing Schumpeter's work has only been exceeded by those testing the relationship between industry concentration and profitability (Cohen & Levin, 1989).

The years following Schumpeter's work were marked by the growth of large organizations that embraced bureaucratic control structures. "During the 1950s," writes Hidalgo and Albors (2008: 114) firms were advised to innovate when it was possible to protect their intellectual property (Teece, 1986), and avoid innovation where there was a strong possibility of spillover into the external environment.

From the 1960s through the 1980s, firms increasingly began pursuing their goals by engaging other organizations. Strategic alliances grew in popularity in the 1970s, and the second half of the 1980s saw a rapid rise in the number of alliances, especially in the technology space (de Man & Duysters, 2005; Tzeng, 2009). For innovation researchers, the pendulum swung toward a more sociological perspective emphasizing the grassroots origins of innovation. In this view, innovation is a process that cannot easily be controlled. Instead, innovative firms should foster an entrepreneurial culture and provide autonomy to those responsible for innovation. Empirical examples include "skunk works" teams where innovators could work in isolation from organizational routines that might stifle creativity. As examples, Tzeng (2009) cites Kanter's study of innovative U.S. firms (1983), Quinn's study of companies from the U.S., Europe, and Japan (1985), and Hamel's study of IBM (2000). In these examples, the authors discovered that innovators within firms were most successful when they were allowed to act with some degree of autonomy.

As the pace of technology increased in the 1990s, some innovation researchers began to see innovation as a form of "craftsmanship" that takes time to develop (Arthur, 2001; Tzeng, 2009). In this perspective, firms innovate by envisioning changes that defy simple path-dependencies, current market conditions, and even the desires of consumers. Innovators are viewed as inspired, devoted, and even revolutionary individuals Ford Motor Company stands out as a prominent example (Tzeng, 2009). This perspective maintains the belief that bureaucratic environments stifle innovation, in part because innovation in viewed as an emergent process (Mintzberg, 1978; Tzeng, 2009). Ford Motor Company stands out as a prominent example; Founder Henry Ford once quipped, "If I asked my customers what they wanted, they would have simply said a faster horse" (Hornby, 2009: 51). A more recent example is Steve Jobs and Apple. During Jobs' tenure, the firm repeatedly released products to market acclaim despite rarely engaging in market research to determine customer interests (Isaacson, 2011).

Contemporary innovation research is an amalgamation of these earlier perspectives. Elements of the early firm-centric view are visible in today's research on dynamic capabilities (Eisenhardt & Martin, 2000), which investigates how internal mechanisms influence innovation. As I will describe in greater detail later in this chapter, scholars of open innovation (Chesbrough, 2003; Wang & Ahmed, 2007), follow earlier researchers in suggesting that innovation can be increased through interaction with the external environment.

Internal and External Sources of Innovation

Innovation research, broadly, has studied best practices for both internal and external innovation, and the open innovation perspective (e.g. Chesbrough et al., 2006a)

has worked to reconcile whether firms are better off when they innovate internally or when they draw on ideas from the external environment. Because this question is central to my dissertation, I review how researchers have examined this question in the past.

Internal innovation research typically examines a firm's research and development efforts. This approach was popular in the post-Schumpeterian era of big business that emphasized corporate control (Tzeng, 2009). At first, it may seem logical that internal innovation efforts are likely to be the most fruitful, as they are the most proximate to the firm and enable a firm to draw upon its own resources and capabilities to create competitive advantage (Dierickx & Cool, 1989). Moreover, intellectual property laws may more easily protect innovations developed in-house than those created in partnership with outsiders. However, as a firm grows in size, internal coordination mechanisms grow more complex. Studies have empirically demonstrated that large firms attempting to transfer knowledge between divisions face challenges similar to firms trying to transfer knowledge from external sources. Causal ambiguity, difficult relationships, and a lack of absorptive capacity can be significant barriers to knowledge transfer within a firm (Szulanski, 1996).

External innovation sources include any innovative process where ideas either originate outside of the firm or are developed in collaboration between a firm and external entities. Acquisitions and alliances are two often-researched vehicles by which firms work with the external environment. Acquisitions can bolster innovation by improving the transmission of knowledge between two firms (Bresman et al., 2010) or by fostering economies of scale (de Man & Duysters, 2005). One might argue that firms with more acquisition experience are likely to be the most successful, but research has

produced inconsistent findings. Anand and Khanna (2000) declared acquisition experience to be an important predictor of success, Hitt et al. (1991) found that it negatively affected a firm's own R&D intensity, and Zollo and Singh (2004) concluded that it had no influence on post-acquisition performance. The most successful acquisitions tend to be between firms with a moderate overlap between knowledge bases (Ahuja & Katila, 2001; Sampson, 2007).

Compared to acquisitions, alliances appear more likely to foster innovation. An alliance is defined as "an agreement between two or more firms to jointly manage assets and achieve strategic objectives" (Yin & Shanley, 2008: 473). An innovation-seeking firm may choose to ally with others in the hopes of reducing contractual differences (Anand & Khanna, 2000), finding production efficiencies, or improving its ability to monitor the competitive landscape by sharing insights (Williamson, 1975). In a meta-analysis, de Man and Duysters were blunt: "the studies reviewed here point to a very clear overall conclusion: alliances are outperforming mergers and acquisitions in terms of their effect on innovation" (2005). They point out that the failure rate of alliances, while high at 50 to 60%, is below that for acquisitions, at 70%. Consistent with work on acquisitions, alliance research has attributed success to cooperation between firms with a moderate degree of technological diversity (de Man & Duysters, 2005: 1384).

Collectively, research shows that while external sources of innovation can be the source of new and valuable ideas (e.g. Utterback, 1971), information is often difficult to transfer. While researchers do not agree upon a normative blueprint, they do provide two conclusions. First, research on acquisitions (Ahuja & Katila, 2001) and alliances (de Man & Duysters, 2005; Sampson, 2007) suggests that a moderate degree of knowledge

overlap between the two firms is most likely to contribute to innovation performance. Secondly, research points to the importance of governance mechanisms that provide structure for what could otherwise be a haphazard process (Kale et al., 2002).

Any discussion of internal versus external innovation is incomplete without mention of absorptive capacity, which has been a mainstay of management research and the subject of over 900 peer-reviewed articles in management journals (Lane et al., 2006). Absorptive capacity, defined as a firm's "ability to recognize the value of new information, assimilate it, and apply it to commercial ends" (Cohen & Levinthal, 1990) is facilitated by individuals who act as conduits for information transfer across firm boundaries (e.g. Yin & Shanley, 2008). In his seminal paper on the topic, Tushman (Cohen & Levinthal, 1990) found that the effective transfer of external ideas was facilitated when those occupying boundary-spanning roles shared information with individuals inside the firm. These boundary spanners are necessary but not sufficient for the development of absorptive capacity. A firm must also have its own unique capabilities to benefit from external knowledge (1977) dismiss the possibility that absorptive capacity could be acquired on the market. They argue that that it is both highly path-dependent and cumulative}. These capabilities may include individual behavior, learning, or organizational processes. . Furthermore, Cohen and Levinthal (1990) dismiss the possibility that absorptive capacity could be acquired on the market. They argue that it is both highly path-dependent and cumulative; firms with absorptive capacity in the past will be more likely to maintain it in the future. In their literature review of work on absorptive capacity, Lane et al. (2006) observed that there were few empirical articles on the subject, and "fewer still that have examined absorptive capacity directly" (851). The

empirical evidence that does exist suggests that inter-organizational relationships and community membership both foster absorptive capacity, and one outcome of absorptive capacity is increased incremental innovation (Lane et al., 2006). Recent attempts to extend knowledge about absorptive capacity have focused more on how firms relate to the external environment. For example, Dyer and Singh (1998) argued that interaction and collaboration was important to developing absorptive capacity, and that the benefits would be shared by those who worked together. In their review of over 80 papers on the subject, West and Bogers (2013) found consistent evidence that absorptive capacity increases the likelihood that external innovation sources will improve both innovation and financial performance.

THE OPEN INNOVATION PERSPECTIVE ON FIRM ENGAGEMENT IN COMMUNITIES

In recent years, studies of firm interactions with external communities have been popular among scholars of open innovation. Open innovation is a normative perspective within management theory that advocates "the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively" (Chesbrough, 2006). This work, led by Henry Chesbrough (2003) and other scholars of open innovation, (Chesbrough & Appleyard, 2007; Chesbrough et al., 2006b) argue that in a modern world marked by technological change and worker mobility, firms prosper not only by taking innovative ideas from the external environment, but by purposely giving internally developed ideas to the external environment.

I pause here to quickly clarify the difference between "open innovation" and "open source." All open source is not necessarily open innovation; individuals often confuse the two (West & Gallagher, 2006b). Open innovation, defined above, is a perspective in management literature. Open source is different; it describes the specific phenomenon in the technology industry where communities of volunteers create and freely distribute software. Open-innovation focuses on value capture and the creation of rents; proponents of open-source software often eschew such motives. It is understandable that some may become confused, as the study of open source provides a rich context to better understand ideas of open innovation. Quoting West and Gallagher, "Open-source software is a great exemplar of open innovation because of the shared rights to use the resulting technology as well as the collaborative development of the technology" (2006a: 322). Their work examined how firms used open-source communities to advance a variety of profitable business models, including selling or donating compliments, spinning out noncommercial technology to promote other objectives, or pooling R&D with a community (Almirall & Casadesus-Masanell, 2010).

Open innovation is often considered as three related but distinct processes (Enkel et al., 2009; Vanhaverbeke et al., 2008; West & Gallagher, 2006a; West & Lakhani, 2008). In the first "outside-in" process, firms incorporate external ideas (from suppliers, customers, or outside communities) into internal innovative practices. This is the most commonly researched process, building on related bodies of work such as absorptive capacity (Cohen & Levinthal, 1990). The second, less research process is "inside-out," where firms license or otherwise transfer internally developed innovations outside the firm (Chesbrough, 2003). The third process is a "coupled" process, defined by Gassmann

and Enkel as "coupling the outside-in and inside-out processes by working in alliances with complementary partners" (2004: 6). While some have theorized that this combination of inflows and outflows can be beneficial to a firm (Chesbrough et al., 2006a; e.g. Enkel et al., 2009), to date there has been little empirical research. This notion of "combination" on inflows and outflows is an important theme in this dissertation because it most closely describes the continual process of "engagement" between a firm and an external community.

These issues present an opportunity for management researchers who are able to assess the strategic processes that take place when a firm decides to work closely with an external community.

SUMMARY AND RESEARCH OPPORTUNITIES

As the LAMP ecosystem has demonstrated and scholarly work has argued (e.g. Marquis, 2003), communities can stand the test of time and weather significant change. In short, communities aren't going away, and firm engagement of these organizational forms is, by all indications, on the rise. The growth of open source software communities has provided a rich empirical setting for those researching innovation. While much of this has focused on the unique aspects of these new organizational forms, a small but growing body of literature is learning how firms interact with these communities. Yet scholars are just now beginning to examine how a firm's interaction with communities may affect strategy process and performance (West & Lakhani, 2008). This dissertation continues this practice.

Specifically, I suggest there are two significant research opportunities that remain for scholars studying firm interaction with open source communities. First, a majority of

the research I reviewed on firm–community interactions has considered outcomes without considering when and how they should be sought. In other words, only a few studies have examined the phenomenon from a strategic process perspective, or considered potential benefits and liabilities of engaging with external communities. Consider that engaging an open source community requires firms to spend time and effort in ways that may not immediately benefit the firm. Community participation necessitates time and attention, often by top managers. These managers are charged with a myriad of responsibilities, from the mundane (e.g., paying membership fees), to the important, such as assessing the motives of other firms in the community, deciding whether to participate in community governance (e.g., O'Mahony and Ferraro, 2007), and choosing the most appropriate employees to represent the firm (Rosenkopf & Tushman, 1998). Managers may then need to monitor their employees' involvement. Dahlander and Wallin (2006) found that firm-sponsored community members are likely to spend more time interacting with the community than those who are not sponsored. Collectively, these tasks may distract managers from the firm's core purpose (Stam, 2009). Moreover, several studies have shown that individuals are often driven to participate primarily due to social motives (Dahlander and Gann, 2010; Shah, 2006; West and Gallagher, 2006), which may conflict with the interests of the firm.

Second, scholars have not codified the exchange relationships between communities and firm. Studies typically examine these relationships by considering the artifacts of exchange (e.g., software, assistance, time) in isolation from each other. This creates the opportunity to investigate the phenomenon using a wider lens, considering the myriad forms that exchange might take.

Third, when considering the growing corpus of open innovation work, research exploring the "outside-in" process is more common than work investigating either the "inside-out" or the "coupled process" (West & Bogers, 2013). Specifically, I used a multiple-case design based on interview data. More specifically, I conducted several interviews, using later interviews to confirm, reject, or refine themes that emerged from the earlier interviews. Interviews were conducted in person at technology conferences in both Austin and San Francisco in 2010. My initial motivation was to attend the conferences to meet entrepreneurs and business executives who had worked with open source communities. I met my first interview participant following a panel discussion that explored how firms develop business models by using and working with open source software. Other interview participants were identified using snowball sampling. I approached the first interviewee following a conference presentation}. Interaction between firms and an open source community is an ideal setting for such studies because the deep connections and frequent interactions between firm and community make it difficult to discretely identify practices that are either "outside-in" and "inside-out." Exchange becomes a function of deeper and more frequent interactions.

The next chapter uses an inductive study of firms that have chosen to work with an open source community. My goal is to develop a better understanding of this phenomenon by identifying important constructs and possible causal relationships between those constructs.

Chapter 3: Inductive Pilot Study

INTRODUCTION

This chapter provides a summary of my inductive pilot study of the phenomenon of firms working closely with an open source software community. I conclude with three propositions that led to the creation of independent variables used in the subsequent quantitative inquiry described in the remainder of this dissertation.

Chapter 2 explains that while open source communities have received attention in the literature, much of the research has centered on the motivations of individual contributors or communities' governance mechanisms. The phenomenon of organizations embracing open source communities—and the processes and consequences of such behavior—have received significantly less attention, and the consequences of such behavior remain relatively unexplored.

Due to a relative lack of prior theory, I conducted an inductive pilot study of this phenomenon. This chapter provides an overview of those efforts, which included conference attendance at both open source and general technology conferences and interviews with executives at several companies.

Three constructs emerged from this research process, each explored in detail below. First, *taking from the community* describes processes where firms take community resources and assets for use in their own business models and processes. The second, *giving to the community*, describes the inverse—business practices where firms donate assets and resources developed by the firm back to the community. The third construct, *interactive engagement*, describes the interaction of both giving and taking.

METHODS

My pilot study used a limited but similar research approach by Graebner (2004). Specifically, I used a multiple-case design based on interview data. Specifically, I used a multiple-case design based on interview data. More specifically, I conducted several interviews, using later interviews to confirm, reject, or refine themes that emerged from the earlier interviews. Interviews were conducted in person at technology conferences in both Austin and San Francisco in 2010. My initial motivation was to attend the conferences to meet entrepreneurs and business executives who had worked with open source communities. I met my first interview participant following a panel discussion that explored how firms develop business models by using and working with open source software. Other interview participants were identified using snowball sampling. I approached the first interviewee following a conference presentation; this initial interviewee then recommended other interview targets.

Following each interview, I asked participants to identify others who would be willing to participate in the research process. This process led to nine interviews held during the spring and summer of 2010. Of those nine interviews, eight were digitally recorded with fidelity allowing for accurate transcription. Two of the eight interviews were held with executives at companies that did not work closely with an open source community and were subsequently excluded from analysis. Two of these remaining six interviews focused on a single firm, leaving a corpus of six interviews describing five firms. The final resulting five firms are described in Table 4.

Individuals interviewed included the founders and C-level executives at technology firms that employed between 14 and 85 employees each. Two firms

specialized in high-end website design for clients in either the higher education or nonprofit sector, one firm developed a unique social media platform for news organizations, one provided training and technical support to other technology firms, and one provided large-scale technical support enabling smaller technology firms to provide solutions to large clients. At the time the interviews were held, all five firms had previously established strong ties with a specific open source community, Drupal. For four of those five firms, such ties had developed and grown over time.

Interviews were conducted in a free-form manner and ranged from 30 minutes to close to two hours. Prior to each interview, I reviewed publicly available information about each company. I began each interview by explaining that the purpose of my research was to learn more about the processes by which the participant's firm worked with one or more open source communities. Initial questions invited participants to share the history of his or her firm; later questions became more specific as I sought to learn more about firm–community interactions.

Interview analysis began after all interviews were collected and transcribed. Consistent with suggestions by Eisenhardt (1989) and Eisenhardt and Graebner (2007), I searched transcripts for evidence of constructs common among different settings and firms. I created charts and tables to compare these constructs. Finally, I developed propositions based on the refinement of these constructs that elucidate and refine the process and consequences of three patterns of firm behavior when interacting with an open source community. To preserve anonymity, the names of participating firms have been changed in this dissertation.

In addition to these formal interviews, I conducted more than 30 informal conversations with individuals who were connected with the Drupal organization, or its nonprofit arm, the Drupal Association. These informal conversations were mainly used to provide a natural validation and insight to support the more formal interviews.

FINDINGS

Following are the main findings of the inductive phase of the research. First, I present the three concepts that eventually coalesced to form the main independent variables of the subsequent quantitative study. Within each, I identify examples of how these behaviors transpired.

Taking from the Community

Firms in my sample all took resources, to some degree, from the Drupal community. By definition, open source software is freely available for anyone to download, its use governed only by the community's open source license. While some licenses may restrict what a firm can do with the code it modifies, licenses do not limit who can download the software. This is the case with the Drupal community. Table 5 lists behaviors, motives, and anticipated outcomes associated with interview participants' descriptions of *taking* from the community.

Scholars investigating open source have uncovered several benefits from taking, including, as described previously, lower development costs (Samuelson, 2006), as well as access to both support (Bonaccorsi et al., 2006; Lerner & Tirole, 2002), and expertise (Gruber & Henkel, 2006).

Interview participants described three variations on the theme of "taking": taking software, ideas, and help. Each is described below.

Taking Software

For software companies, taking code from the Drupal community was a common if not universal practice (The phrase "source code" refers to the human readable text created by a programmer used to generate software.) Any firm can freely download and use the work of others in the community, saving time by not having to develop software that performs a similar function. The description of this process by the CEO of NewsShare supports the assertions of researchers who argue that taking code allows firms to save time and money (e.g. Samuelson, 2006):

> The first thing we did we built a prototype. And, I credit Drupal with a lot of our success in that public community because we spent about 3.5 or 4 months with a really small team of about two to three people. We spent less than $50,000.

Taking code from the community is simplified by the modular nature of the source code. In other words, firms are able to easily download different "pieces" of source code that work well together. Modularity provides several benefits in an open source community. Contributors can select from among different modules to solve a problem, choosing some and rejecting others. Different groups can work on complementary modules separately, confident that they will work with the underlying system, without having to understand the intricacies of that underlying system. Finally, modularity allows community members to replace older modules with newer ones,

enabling them to improve the community's product without the involvement of a large group of community members.

The value of modularity is twofold. First, modules provide different functionality. Second, they fit well into an overall software framework, connecting to other "pieces" of software without altering the functionality of the overall system. Both HelpingWeb and NewsShare compared this approach to Lego toys. In the end, this modular approach provides convenience. HelpingWeb's CEO described a process where such convenience enticed them to download code, "Sometimes it's unintentional, just because there's a high level of transparency and you'll say, 'Oh, so-and-so just checked a new plugin in that does that. Awesome. I'll go use it.' "

By combining several of these software "modules," firms that take code from the community are more easily able to quickly build functional products of their own. AcademicWeb's CEO described this process: "I think this is a cool thing that Drupal can do…. You can basically, on the fly, go grab some information, display it, and then use all the stuff you get from Drupal out of the box, commenting, ratings, all of that."

Taking Ideas

Firms also engaged in *taking* ideas from others in the community, which they were then able to incorporate into product offerings. While ideas are more ethereal than software, the modularity of the software also facilitated taking ideas, as knowledge is embedded in code. MediaNet's CEO described this process best:

> When I say to my clients . . . you've got all of these parts. And they're sitting here and you can use them. And [my clients] just get so excited as they reach in the

> box and start pulling those parts out, and we piece something together on paper what we're going to build for them. And then they go, wow, that's already built and that's built, and I could use this and this and this and this, that's fantastic.

Some firms made a habit of sharing best practices and ideas publicly during community-sponsored conferences, a process I will describe in detail in the next section. Other firms are then able to use adopt these ideas for their own learning and development. As the MediaNet CEO describes, knowledge is embedded in software:

> you'd be sitting in this room with hundreds of other people who were all being equipped and trained by these top notch companies, and they're giving away company secrets constantly. They're showing you, "OK here we just did *Spin Magazine* segment, here's how we themed it, and here's how you guys can do better on your theming." And you're sitting there going, wow—who does this? This is awesome. I love this.

Taking Help

Finally, firms routinely accepted help and assistance from the community. One example identified by participants was using the community to identify and fix errors in a company's software. As an example, AcademicWeb's CEO described these benefits by saying, "there is value to peer code review, especially when you're a small shop. Because the community can give you resources that you don't have . . . in terms of development." This quote is consistent with "Linus' Law" that "with enough eyes, all bugs are shallow," articulated by the cofounder of the open source Linux operating system, Linus Tovalds

(Bonaccorsi et al., 2006; Harhoff et al., 2003). This axiom suggests that by taking help from the community, firms can reduce the time it takes to debug a software product.

Variation in Taking

It became clear that not all firms take code, ideas, and help at the same rate or at the same time. Firms such as AcademicWeb, whose founding predates the community and whose products did not initially use the community's software, had less reason to take anything from the community. Once they made the decision to use Drupal exclusively, taking increased.

Collectively, these observations lead to the first Proposition:

> *Proposition 1. Firms that work with an open source software community rarely limit themselves to taking tangible resources, such as software. These firms also take intangible resources, often in the form of product ideas and development assistance.*

Giving to the Community

The second major theme that emerged from my interviews was giving to the community, which I define as the voluntary donation of knowledge or resources from a firm to a community and its members. Table 6 lists behaviors, motives, and anticipated outcomes associated with interview participants' descriptions of *giving*.

To date, there has been limited research exploring why firms would give or "freely reveal" proprietary inventions. At first blush, this might seem to be in violation of typical norms of intellectual property protection found in most firms. Scholars have used

both game theoretic models (Raymond, 2001) to investigate this occurrence, concluding that despite management resistance, giving may, in fact, be to the interest of the contributor.

This idea that firms stand to benefit by giving to the community was acknowledged by NewsShare: "You contribute in ways that are somewhat self-serving. Because I mean it's difficult as a commercial entity to be truly altruistic." The most visible self-serving motive among interview participants was a shared interest in improving the community's software product, a topic I describe in greater detail below.

Other motives, however, were more social in nature, and included friendships and a sense of fair play and reciprocity. HelpingWeb's CEO used the word "karma" (obviously an extrapolation of the original meaning of the word) to describe this process:

> The . . . idea that it tends to just get referred to as, "Oh, you get karma. You're nice to people, you get karma," but a lot of what happens in open source, because it's been noncommercial for so long, the fact that you solved a problem and then contributed it back isn't a monetary thing. It's—you've given a gift to everyone and now everyone, there's an implied, "Oh," and they share these things with you.

Analysis of interview transcripts revealed four different types of "giving": giving software, giving help, and supporting the community. I describe each in detail below.

Giving Software

Like most open source communities, Drupal depends on contributions of software by its members. The community itself does not employ individuals to write or test code;

all of the necessary work to create, develop, and maintain the community's software must come from its membership and volunteers who make up the leadership of the community.

Across interviews, the overriding rationale for giving code appears to be improving the community's software product. Contributors can give code in one of two ways. First, they can make contributions to the "core," an important foundational layer of software that is updated every couple of years. Interview participants described contributions to the core as highly significant, in part because giving this type of software is no guarantee that it will be incorporated into the community's product. The final arbiter remains the individual who started Drupal almost ten years ago; only he can authorize permanent changes to the "core" of the software. As NewsShare's founder describes:

> Drupal 7 will have . . . 732 people . . . [that] will have actually contributed . . . code that actually got into Drupal Core. That's 732 people. There are probably 4 or 5 times that many people who contributed code, but it didn't get past the threshold, it didn't get the combination of quality — conceptual quality, meaning the concept of what they were doing wasn't sound.

Secondly, contributors may also give software in the form of new modules. As described earlier, these modules of code can be selectively downloaded and used by anyone. Modules extend the functionality of the core, allowing users to serve unique needs that may not apply to everyone. The CEO of MediaNet, a frequent contributor of modules, provides an example:

> The module [enabled] . . . somebody to come on to Drupal and . . . load photographs and be able to edit them direct within the Drupal site. So that was a project that we sanctioned and said "yeah, so we'll pay for that, that's a fantastic idea. It's innovative. People will love this . . . build that module and commit it back to the community.

The act of giving code to the community was universally praised among interview participants. Lulabot's CEO criticized the practice of keeping code private: "if you keep [lines of code] to yourself they die, they stagnate and they become your code to maintain and your code to—and it just becomes your stuff. And ultimately it ends up sort of dragging you down more."

The CEO of NewsShare was quick to recognize those who gave software as the primary source of the community's strength: "It's the core hackers, it's people who volunteer their time that have gotten Drupal to where it is, and who really are the ones that have made Drupal successful."

Giving Help

After giving code, participants described several ways firms provided assistance to others. Examples spanned the gamut. At one extreme, firm employees remained connected throughout the day, so they could be ready to help each other tackle problems. "It's very common in the Drupal community," describes the CEO of HelpingWeb, "for, say, a bunch of developers [to be] sitting around in a[n online] discussion channel, you're just shooting the breeze a lot, talking about bugs you're finding you're trying to plow through."

At the other extreme, firms had explicit and well-organized efforts to provide help by sharing otherwise proprietary knowledge. HelpingWeb's CEO provides one example:

> We also started doing a podcast. And we did a weekly podcast where we would get on and basically talk about everything that we learned about Drupal that week. "Oh, I didn't know Drupal did this." "Oh, look at this, this is fascinating." We'd do interviews with people doing things and stuff like that, but we basically just would like give away our knowledge for free.

AcademicWeb's CEO describes a similar approach:

> We write a blog post, we write case study for our website, we write a case study for proposals. . . . We have all different kinds of audiences so that . . . solidifies our relationship within the community. . . . It gives back. Tells them how we build things. What modules we use."

Supporting The Community by Contributing Knowledge and Labor

Several acts of giving were directed toward supporting the growth, development, and infrastructure of the Drupal community itself. Rather than simply giving software or help, interviewees descried behaviors that were consistent with going "above and beyond." According to the CEO of HelpingWeb, "We don't just give back code, we give back so much knowledge, so much information. We have so many videos and blog posts and articles that we just give, give, give, give, give."

Giving in ways that support the community spanned a variety of behaviors and initiatives. For example, MediaNet was an early leader in organizing events that brought programmers together. According to its CEO:

> [We're] expected to go above and beyond. So I began attending the local Drupal Camp on a monthly basis. When the reins were passed on from the Drupal Camp in Vancouver, our team got involved in starting to organize that, so the next one coming up is the June summit.

In another example, CoachX supported the community through marketing efforts. In the words of its CEO, "We're doing a lot of benevolent marketing of Drupal and the enterprise… spending a lot of dollars around the marketing and sales, just for the benefit of Drupal, not necessarily always for CoachX."

Two of the five firms, NewsShare and CoachX, supported the community by helping to develop software infrastructure. Infrastructure is a broad term that describes the technical underpinnings, or support structures, that allow a software platform to be both robust and stable. A strong system of infrastructure support allows the community's open source software to scale to meet the needs of a more diverse pool of clients, notably larger companies. While large software firms can invest in building their own infrastructure, doing so is more challenging for an open source software platform that relies on the contributions of smaller firms with smaller budgets. In 2007, NewsShare was one of the first firms to recognize that the community needed a platform to handle large data migrations. To see that it was built, they partnered with CoachX. In the words of NewsShare's CEO:

> I [was] spending a tremendous amount of money creating all of these data migration tools, working . . . to get them to where we need them to be. I shouldn't be shouldering this investment on my own. I simply don't have the money to spread around . . . and so we struck a deal with CoachX. They ended up matching our investment.

Providing this kind of support for the community has been integral to CoachX. As its CEO said:

> If we can be a provider of services to the ecosystem, that's kind of where I see that now. Some of those services are SAAS [Software As A Service], some of those are bolt-on services to sites we might build, [and] some of them are services that are related to specific expertise. If you're a five-person shop and you've got a client and they say, "We need an expert on search to come in," where do you find that? Hopefully you can go to CoachX and say, "Hey, I need a search guy for this particular engagement, you know?" Boom, specialist.

Providing this kind of support to the community has not gone unrecognized. In the words of HelpingWeb's CEO, "I don't think a lot of enterprises would take Drupal seriously without a company like [ours] that they could turn to."

Summary

The examples above highlight that there are different forms of giving, including giving code, help, and supporting the community. Giving also varies in volume. Firms appeared more likely to give when they had been using the community's software since their founding, or whose business models were dependent on the success of the

community's software. While HelpingWeb claims, "I would say [we give back] 90%, 95%... of the stuff we build," and each of the five companies interviewed described how they contributed software, it became clear in both these interviews and subsequent conference participation that not every firm gives at such a high level.

AcademicWeb and HelpingWeb are both examples of firms that struggled with the issue of what to give and what to keep to themselves. According to the CEO of HelpingWeb, "I spend time looking at what that balance is between giving and keeping proprietary, and I spend time looking at that balance between being the kind of company that a giant corporation could write a six figure check to and being the kind of company that people in the Drupal community that work for an NGO want to work with."

Proposition 2. Firms that work with an open source software will go beyond donations of software to also give support, typically by sharing knowledge with both the community and with other member firms.

"Interactive Engagement:" The Interaction of Taking and Giving

The third construct, interactive engagement, describes the interaction of giving and taking. I coin the term interactive engagement to mean the process of both giving to and taking from the external community. I have named this construct interactive engagement for two reasons. First, this name describes an ongoing process whereby firms are continually interacting with the community and engaging the community's membership in this process; here, it is often difficult to clearly delineate between firm acts of "giving" and "taking." Second, it also helps to think of interactive engagement as akin to the statistical notion of an interaction between two variables. That is, the effect of

interactive engagement is more than merely the linear sum of the two firm behaviors of taking and giving. In this section, I describe the norms of interactive engagement and outcomes that were described by the firms I met with that engaged in both high levels of giving and taking. Table 7 uses illustrative quotes to describe these outcomes.

Norms of Interactive Engagement

When asked how she managed her time, an executive at HelpingWeb replied, "a lot of the work that we do for clients is interfacing with the community . . . of the time I spend working with clients, 50% of that time consists of engaging with the community." Her response was typical of interview respondents. Employees at interactively engaged firms spend much of their workday interacting with others in the community who are not employees of their own firm. Interaction is so frequent that it can be difficult to discern where "taking" stops and "giving" begins. Three norms emerged across my interviews—enthusiasm for the community among firm employees, a sense of responsibility for the community's development and well-being, and an aversion to perceiving other firms within the community as competitors. Each is described below.

1. Enthusiasm for the Community

Across all five firms, executives described their involvement in the Drupal community with enthusiasm and pride. Firms had made critical business decisions to engage and grow the community. For example, after eleven years, AcademicWeb decided to drop their use of all other software systems and use Drupal exclusively. HelpingWeb and CoachX were both making money providing training and infrastructure services to

other firms in the Drupal community. All five of the firms proudly advertised the fact that they were working with the community. Perhaps the CEO of MediaNet put it best:

> We are constantly . . . branding around the concept that we have "the power of Drupal." And I think that's really the key of what this is. As you look at each of these different distributions that are starting to come together, and how much excitement is generated in communities that have formed around those . . . people start to see what it can do for them in that realm, it's exciting.

Interactively engaged firms saw the success of the community as integral to their own success. According to the CEO of CoachX:

> We fail without the community . . . without an ecosystem successfully implementing what we were doing we wouldn't get any customers because we wouldn't have success stories to talk about. We wouldn't have people out evangelizing what we're doing."

2. Concern for Community Development

For interactively engaged firms, whose future is tied to the community, enthusiasm can become concern for the community's well-being. HelpingWeb used a metaphor of "growing the wave," to describe high levels of community involvement: their goals were to both help the community grow the wave and to successfully surf atop it. In their words, "[participating in the community] grows the wave, and that's something that's—we make really conscious, thoughtful decisions. We have to really think about how can we grow the wave that we're riding on and still stay on the top of it."

Part of growing that wave was a feeling that firms should help the community identify and address areas where it was weak. Doing so would benefit all firms that worked with the community. Because the community is made primarily of entrepreneurial firms with limited resources, a single firm could not easily address challenges that applied to the entire community. In other words, community challenges were the responsibility of the community. The founder of NewsShare described the challenge thus:

> No single company, no matter how much money you have, no matter what resources you have, can justify building out things like quality assurance frameworks and load testing frameworks, and dealing with performance and scalability problems. These are the perfect kinds of things to work on collaboratively. They're not . . . your business. And, you can never do them justice on your own.

To address community challenges, the entrepreneurs I spoke with took on informal leadership roles. At times, these roles were official; the Co-CEO of AcademicWeb serves on the board of the Drupal Association. Other times, these executives were concerned with addressing one specific community challenge. The CEO of NewsShare describes the leadership role it assumed to address its technical concerns:

> Three years ago in Barcelona I sat down with Kieran [assumed spelling] and Dries and I was like, "No one is going to take Drupal seriously if we can't scale this thing without a quality assurance, automated testing framework on this thing."

> Kieran took it to heart, put a tremendous amount of effort coordinating efforts across multiple companies and building it to where it is today.

3. Co-opetition

Perhaps the strongest and most clearly articulated norm was that of co-opetition, or cooperating with firms that are ostensibly competitors. Interactively engaged firms did not describe each other as competitors, even when their markets overlapped. Consider CoachX, a relatively young firm that was able to enter an already mature Drupal community with venture funding and the involvement of the Drupal founder. According to its CEO, "We work really hard to help the community understand that competitors are not within the community."

The leaders of smaller firms shared the same sentiment, even welcoming the involvement and help of larger firms that could otherwise pose a significant threat. The CEO of MediaNet described the norm of cooperation among firms of different sizes:

> In this community it seems like a lot people are motivated to help the other guy to get up. And if you look at it, to me it was kind of like Three Musketeers, all for one and one for all OK. To me it was like this. This is what I want to be involved with, because there's an element of the bigger guy looks over the little guy and he never ever gets anything back from the little guy, but the little guy is so helped by the big guy that one day a little guy helps out the next little guy.

One company shared a story of two Web development firms who were located in the same city and competing in the same market. Rather than competing head to head, these two firms collaborated, setting up redundant hardware so that, if one company's

technology were to fail, the other would immediately take responsibility for supporting the company's websites. In the words of the HelpingWeb CEO, "Other shops that we work with rely on us and vice versa. There's just implicitly a lot of deep networking that goes on there because we're all relying on things that each of us is working on." One interviewee extended this perspective beyond his firm and the community. According to MediaNet's founder, "That's how people should be operating in all senses of business. There shouldn't be that sense of it's a dog-eat-dog, cut-throat world where we're really trying to put the other guy down. . . . This is what business should be like."

Outcomes of Interactive Engagement

Interview participants described how their high levels of involvement in the community led to two clear outcomes: employee recruitment and client referrals.

1. Employee Recruitment

Interactively engaged firms were better able to identify and vet prospective employees, thanks to their close ties in the community. Consider this example, provided by HelpingWeb's CEO:

> I found another guy in the Drupal community, Scott Smith, and said, "Listen, you have helped me out on message boards. You seem like a very knowledge guy. Can I just like get you on the phone and pay you to explain this to me?" And he [responded], "You would do that?" I [responded], "Yeah. I did the math and like I would save a lot of money by having someone like [you]." And he turned me into a Drupal expert and I finished that project and we did another project together.

The employees who were referred by others in the community were often seen as higher quality, with greater levels of relevant experience. According to the founder of NewsShare, "From a talent recruitment standpoint, or a features and functionality standpoint—you want to be on the cutting edge, driving the platform where you want to go."

2. Referrals

Interactively engaged firms were also a rich source of client referrals. These referrals often came from firms that could be considered competitors. Consider this statement from the NewsShare CEO: "One of the guys on the panel . . . said that 15% of his business last year came from sponsoring conferences, referrals, 50% came from competitors. People he knew in the community, other shops that said, 'We're too busy.'" AcademicWeb's CEO had a similar story: "Drupal has been fantastic for us, in terms of a tool that we use to implement, and the work that we've done in the community certainly does help with the generation. We get unsolicited RFPs literally every day."

Summary

In conclusion, the norms of interactive engagement and outcomes that were described by interview participants provide evidence that the interaction of giving and taking presents outcomes that are different from the sum of the behaviors of giving and taking.

Proposition 3. Firms that demonstrate both high levels of "taking from" and "giving to" the open source software will experience interactive effects of this

combined behavior that supersede the direct effects of either "taking from" or "giving to" the community.

Chapter 4: Theory Development and Hypotheses

This research examines the participation of firms in an open-source community. I hope to answer the question of why such firms would give away limited resources to these communities, and what they get in exchange. In doing so, I hope to address two calls for research articulated in extant literatures. First, by examining the firm-level consequences of giving, taking, and its interaction, I answer West and Lakhani's (2008) call for more research on the outcomes of firm interaction with communities. Second, this study will explore how firms balance giving to and taking from a community, answering Chesbrough and Appleyard's (2007) call for research that examines how firms can embrace open models while maintaining traditional models of value capture.

This chapter begins by exploring two dimensions of participation—giving to and taking from an open source community—and presenting a 2x2 matrix as a heuristic to describe various degrees of *interactive engagement*, the interaction of both giving to and taking from an open-source community. The remainder of the chapter presents hypotheses examining the direct effects of both giving and taking, as well as the consequences of interactive engagement.

A Heuristic of Engagement: Giving, Taking, and Interactive Engagement

In this section, I combine the constructs of "giving" and "taking," to create a heuristic of firm engagement with an open source community. I define taking as *the absorption of resources from an open source community into a single firm*. I define giving as *the voluntary donation of resources from a single firm to an open source community*. For the illustrative purposes of this heuristic, I use the word "resources" as a broad term that includes the core artifact of the community (software), knowledge

(including help using code or business advice), as well as intangible resources such as brand legitimacy and visibility.

I do not use the word "typology," because I do not rule out the possibility that a firm may move from one classification to another (Doty & Glick, 1994). Figure 1 illustrates how "giving" and "taking" in a 2x2 modality forms a heuristic that illustrates different archetypes of firm engagement. According to their level of giving and taking, firms can be classified using this heuristic as unengaged, free-riding, open-handed, or interactively engaged. Figure 1 provides a conceptual illustration of these four archetypes. Later in this chapter, I present formal hypotheses that examine the consequences of giving, taking, and their interaction.

Unengaged Firms

Unengaged firms are those that participate with the community, but score lower on both giving to and taking from the community. One would expect that the consequences of giving and taking, formally articulated as hypotheses later in this chapter, would not apply to these firms. Since I operationalize giving and taking as continuous variables, scoring lower does not mean that unengaged firms do not have any kind of relationships with the community. Many firms may take software from the community once, perhaps to assess its function or value. Similarly, firms may give a few lines of code simply out of benevolence or because of preexisting social relationships. That said, I expect these infrequent acts to be instigated by individuals acting on their own, rather than by multiple employees acting with the explicit support of upper management.

Free-Riding Firms

Firms classified as free riders take from, but do not give to the community. The topic of free riding has received some attention in the management literature, and I adopt the definition offered by Dyer and Nobeoka, "members who enjoy the benefits of the collective good without contributing to its establishment and/or maintenance" (2000: 348). Granovetter (2005) argued that free riding would be less likely in smaller groups, where norms that discourage such behavior could be stronger. In open source communities, Baldwin and Clark (2006) found that the modularity of the codebase (i.e., the ease with which components could be incorporated into a new product) would reduce free-riding.

There may be several reasons for a firm to take and not give. Later in this chapter, I hypothesize that one direct effect of *taking* is an increase in firm productivity. Scholars investigating open source have uncovered several benefits from taking, including, as described previously, lower development costs (Samuelson, 2006), as well as support (Bonaccorsi et al., 2006; Lerner & Tirole, 2002), and expertise (Gruber & Henkel, 2006). By giving and not taking, free-riding firms may enjoy many of these benefits without engaging in acts of reciprocity.

Open-handed Firms

Open-handed firms are those that give, but do not take from the open source community. One might be tempted to personify the behavior of these firms as "benevolent." Just as the term "free rider" is often characterized by self-interest at the expense of others, open-handed firms may be characterized by caring for the needs of the community at the expense of the firm's own interests.

To date, there has been some limited research exploring why firms would give or "freely reveal" proprietary inventions. Scholars have used both game theoretic models (Harhoff et al., 2003) and questionnaires (Henkel, 2006) to investigate this occurrence, concluding that despite management resistance, giving may be in the interest of the contributor. While benevolence may be a motivating factor, there could be tangible benefits to be had from giving and not taking. Later in this chapter, I formally hypothesize that giving directly leads to the formation of social ties with employees of other firms. These ties may be valuable for reasons unrelated to the community. A second possibility is that giving allows a firm to divest projects that began internally but are no longer core to the company's own mission, as with the "spin-out" projects studied by West & Gallagher (2006a). Finally, firms may perceive growth of the community to be in their own long-term interest, and give to promote the community. In interviews, software executives working with the Drupal open source community used an illustrative metaphor: in the technology industry, every company was fighting for a larger slice of the pie, while those working closely with the open source community were trying to bake a larger pie.

Interactively Engaged Firms

Finally, I define interactively engaged firms as those that score highly on both giving to and taking from the community. Later in this chapter, I formally hypothesize three effects from the interaction of giving and taking. For now, I provide a conceptual overview of the concept.

It is logical to describe the relationships between the boundary spanners of interactively engaged firms (often midlevel programmers) as reciprocal: these are the

individuals who are involved in the daily acts of giving and taking with each other via the community. At the micro level, reciprocal relationships have been shown to increase trust (e.g. Cropanzano & Mitchell, 2005; Molm, 2003), and there is reason to believe the same holds true for relationships between firms reciprocally engaged with an open source community. At the firm level, Dyer and Chu (2003) found that information sharing fostered trust among Japanese automakers.

Interactive engagement enables firms to exhibit trust-enhancing behaviors: in this case, benevolence (Barney & Hansen, 1994) and competence (Shah & Swaminathan, 2008). Research on giving to a community shows that it allows firms to demonstrate benevolence when they donate code that helps to advance the community's product (Harhoff et al., 2003; Henkel, 2006). Such giving demonstrates competence when donations are of sufficient quality to be incorporated into the community's core products. Acts of taking allow this firm to demonstrate competence through its ability to integrate and extend the utility of community products in its own products.

THE DIRECT EFFECTS OF TAKING FROM THE COMMUNITY

Extant literature has shown that external environments can be a valuable source of knowledge (Henkel, 2006; Rigby & Zook, 2002) that contributes to firm innovation (e.g. Pittaway et al., 2004). An open source community is one such environment. I define "taking from the open source community" (taking) as *the absorption of resources from an open source community into a single firm*. This definition of taking applies the outside-in process, in which firms innovate by incorporating external ideas into internal practices (Chesbrough et al., 2006a; Enkel et al., 2009), to a dyadic relationship between a firm and an open source community.

The construct of "taking" overlaps the concept of "absorptive capacity" (e.g. Cohen & Levinthal, 1990; Zahra & George, 2002). Cohen and Levinthal define absorptive capacity as a firm's "ability to recognize the value of new information, assimilate it, and apply it to commercial ends" (1990: 128)}. Taking, like absorptive capacity, is facilitated by *boundary spanners*, individuals who act as conduits for information transfer across firm boundaries (1990: 128); these are typically the midlevel programmers and managers employed by a software firm. Taking is also facilitated when organizations structure communications in ways that facilitate the efficient assimilation of knowledge.

I argue that taking from an open source community enables a firm to increase its rate of software creation productivity. I define software creation productivity in accordance with Krishnan et al. (2000) considering the total lifecycle of individual projects. Taking from a community enhances software creation productivity through two mechanisms. First, using resources acquired externally allows a firm to save money and allocate internal resources to activities in line with increased productivity. Samuelson (2006) demonstrated that the costs of software development can be significantly reduced through the use of open source. By using free software provided by the community, firms can reduce the costs of their own internal development (Samuelson, 2006), as well as avoid costly licensing fees required by private software providers (Bonaccorsi et al., 2006). Gruber and Henkel (2006) even showed that the use of open source code could reduce the liability of smallness in entrepreneurial startups by freeing these firms from the necessity to engage in costly internal R&D. Open source code is not only free, but often reliable (Lerner & Tirole, 2002). The difference between fixed and total costs can

be lower, allowing the firm to increase profit margins (Bonaccorsi et al., 2006). In sum, taking allows a firm to create products and services with less expenditure of resources.

Secondly, taking allows a firm to acquire valuable procedural knowledge and expertise. Taking from an open source community can provide a firm with access to knowledge (Goode, 2005; Von Krogh & Von Hippel, 2006) and expertise (Gruber & Henkel, 2006). Community members help a firm learn how to incorporate and extend the usefulness of its own code, allowing the firm to avoid further allocation of internal resources.

In summary, I suggest that taking from an open source community enables a firm to increase its productivity through two mechanisms: cost reductions resulting from the use of outside resources and the application of procedural knowledge acquired from the community. I caution that I do not expect the resulting increase in productivity to be linear; benefits to productivity will not continue at the same pace indefinitely. Rather, I expect that the more a firm takes from a community, the more software creation productivity will increase at a decreasing rate.

Therefore, I propose:

H1. Taking from an open-source community will be positively related to the productivity of software development.

THE DIRECT EFFECTS OF GIVING TO THE COMMUNITY

As noted previously, I define "giving to the open source community" (giving) as *the voluntary donation of resources from a single firm to an open source community*. At

first, the concept of voluntarily donating company resources may seem at odds with rational rent-seeking behavior. Researchers, however, have concluded that firm behavior such as revealing code to a community rarely harms the firm (Henkel, 2008), and may even benefit the firm by providing benefits related to knowledge and learning (Lakhani & von Hippel, 2003). Giving is also discretionary; and firms have been shown to vary the degree to which they give code to an open source community. According to Henkel (2006), this behavior allows firms to protect their own interests while supporting the needs of the community.

Giving code also raises potential agency issues between the individual contributor and the firm he or she works for. Put differently, the motives of an individual contributor to give may differ from the motives of the firm. These motives may be further affected by the fact that the individual contributor likely has allegiances to both his or her firm and the open-source community. Henkel (2006) directly investigated this principal-agent problem, concluding that one's identification with the community was a strong driver of revealing code, but that both general identification with a "free software ideology" and personal pride were not. He concludes that the potential agency problems are not severe, and instead, individual programmers act as "champions of revealing," that support and encourage the firm to give (i.e. reveal) code to the community.

I suggest that one consequence of giving to an open source community is the development of social ties between the employees of the different firms that give to the community. I define a social tie as *a relationship between two or more individuals in different firms, characterized by friendship and trust*. Consistent with Ingram and Roberts (2000), as well as work on embeddedness (Uzzi, 1996, 1997; Uzzi & Lancaster, 2004), I

support the belief that social ties provide more than just sentimental benefits between individuals; they can have instrumental importance to organizations. Research on communities supports this assertion. In their open innovation study of startups participating in the IEEE standards-setting group, Waguespack and Fleming (2009) found that the only significant effect leading to a liquidity event was physical participation in community activities. This social interaction was even more influential than authoring technical standards or employing a leader of the community within the firm. Adding additional evidence, Rosenkopf et al. (2001) found that participation by mid-level managers in technical communities facilitated alliance formation for less-established firms.

Research has also shown social ties to improve performance by fostering collaboration (e.g. Ingram & Roberts, 2000). Uzzi (Uzzi, 1996), for example, demonstrated that organizations sharing social ties were more likely to engage in joint problem solving and were better able to quickly address challenges. Others have found that social ties between executives lead to similar patterns in corporate behavior (Galaskiewicz & Wasserman, 1989).

Social ties also facilitate the exchange of information between organizations (Geletkanycz & Hambrick, 1997; Ingram & Roberts, 2000). Uzzi (1997), argued that embedded ties are particularly useful for facilitating the transfer of proprietary and tacit information. Other work has shown social relationships to facilitate the accumulation of reputational benefits for entrepreneurial firms. In a study of how new ventures accumulate reputation, Petkova et al. (2008), using grounded theory, linked the

development of reputation in entrepreneurial firms to the formation of social relationships with stakeholders.

I argue that two mechanisms underlie the process by which giving to an open source community fosters the development of social relationships between employees of different firms. First, by *giving*, a firm and its employees demonstrate support for this development. This may happen several ways. By allowing employees to volunteer time and assistance, a firm recognizes the value of giving and signals its support for community governance. Giving also demonstrates a firm's commitment to the community's technology. By giving to an open source community, a firm helps promote the development and growth of a technology standard it has adopted as its own (Bonaccorsi et al., 2006; Fosfuri et al., 2008; Gruber & Henkel, 2006; Henkel, 2006).

Second, giving provides the impetus for community members to interact with the employees of other firms. Gerber and Henkel (2006) argued that by donating code, firms demonstrate their technical proficiency to the community. The benevolent act of giving code may signal that a firm may be willing to provide the assistance required to use or extend such code, encouraging others in the community to seek advice and guidance.

For the reasons discussed above, I propose:

H2. Giving to a community will be positively related to the strength of social ties between a firm and an open source community.

While a cross-sectional research design prevents this study from investigating the issue directly, I wish to address the possibility of reverse causality: that pre-existing social ties between employees may lead a firm to give to a community. While this likely

does occur at times—especially when individuals with pre-existing social ties are hired—I believe it is less common than an act of giving resulting in the formation of social ties. First, the geographic distribution of open source community members makes the formation of social ties difficult. Consider two ways that geographically distributed open source community members may form a social tie: conference attendance or participation in online community forums. In the first case, I suggest that firms are more likely to sponsor employees' conference attendance when they have a business rationale to do so; giving provides a compelling reason to pay for conference attendance. In the second case, participation in online forums is unlikely to lead to social ties unless one participant gives something—be it code or assistance. In sum, geographic distribution and the cost of community participation make it more difficult for social ties to form absent some form of giving to the community.

INTERACTIVE ENGAGEMENT AND NUMBER OF NEW PRODUCTS

The launch of new products and services is important, even critical, for organizational performance (Damanpour, 1991). Products constitute the means by which firms compete, and as such, successful product development can be an important source of competitive advantage (Brown & Eisenhardt, 1995). Taking from an open source community will have an effect on the number of new products depending on the level of giving by the firm. This is because the social ties that result from giving will have two effects on the firm related to product development. First, social ties provide access to new product ideas that can be acquired from the community. Secondly, social ties encourage firms to embrace a more modular approach to software development, which can both

increase the diversity of a firm's product portfolio and speed product development. Each of these mechanisms is described below.

Taking, Higher Giving, and New Product Development

Access to Product Ideas

Scholars of product development have long focused on the importance of communications with external groups (Brown & Eisenhardt, 1995). Much of this research has emphasized the importance of boundary spanners, employees who communicate often with people external to the firm (Allen, 1971; Tushman, 1977). Boundary spanners facilitate product development by bringing ideas from the external environment and coordinating outbound communication (Brown & Eisenhardt, 1995). Gatekeepers are relevant to the current context because they form the connection between a firm and an open source community. Through their social ties, these individuals provide the firm with access to new product ideas. While this is important, recent research has suggested that access to ideas is not enough. For one, access alone does not speak to ideas' relevance or quality. For new ideas to translate into new products, it is important to facilitate both the fluid exchange and novel combination of ideas (Smith et al., 2005). The social ties that result when firms experience higher levels of giving facilitate both.

A community provides participating firms with visibility and the opportunity to showcase themselves (Henkel, 2006, 2009). When a firm gives to a community, the social ties that result enable the firm to learn about other firms' products, as well as their strategies and target markets. They learn how other firms are using what they take from the community for the creation of their own products and services, potentially reducing the liability of newness (Gruber & Henkel, 2006). The benefits of giving were also

illustrated by Harhoff et al., who used a game theoretic model to compare the benefits of secrecy and "free revealing," concluding that "in a world of self-interested agents with complementary capabilities, free revealing can be profitable" (2003: 1767).

Social ties facilitate this exchange because knowledge flows more efficiently when relationships between parties are established (Tushman, 1977) and positive (Szulanski, 1996). Specifically, social ties increase the number of ways that firms communicate with the community. When Ancona and Caldwell (1992) may have limited generalizability, however, because their sample focused on product development teams within larger organizations} studied the communications practices of new product development teams, they found that teams that engaged in multiple communication practices were more likely to acquire information and resources from outside groups. They identified four communication practices: task coordination, ambassadorial (e.g., impression management), scouting (e.g., scanning for useful information) and guard activities (e.g., preventing release of proprietary information). The social ties that form between a firm as a result of giving increase the likelihood that a firm will engage in the first three behaviors. For example, firms that give need to coordinate how their contribution will be incorporated into the community's product, manage their reputation in the community, and scout for information on the existing needs of the community.

It is possible that Ancona and Caldwell (1992) may have limited generalizability, however, because their sample focused on product development teams within larger organizations; their interactions with "external" groups may have included groups within the same firm. However, others who have focused on interorganizational relationships have found strong performance effects of knowledge sharing. Dyer and Hatch (2006)

found that higher rates of knowledge sharing between Toyota and its suppliers led to sustained improved supplier performance, concluding that "firms with useful knowledge can achieve competitive advantages by exploiting those knowledge assets with a supplier network" (716). McEvily and Marcus found that when firms go beyond sharing information to engage in joint problem solving, they are better able to acquire both complex and tacit information. When a firm gives to a community, the social ties facilitate a similar kind of collaborative problem-solving process; firms actively work to see that their contribution can be incorporated into the community's product. In sum, consistent with this research on teams both between (Dyer & Hatch, 2006; McEvily & Marcus, 2005) and within organizations (e.g. Ancona & Caldwell, 1992), I expect that a diversity of communication practices between a firm and community will facilitate increased acquisition of useful knowledge.

Second, in addition to the exchange of information on existing products, social ties also facilitate the creation of new and novel product ideas. Several studies have demonstrated that through the process of exchanging and combining information, new knowledge is created (Nahapiet & Ghoshal, 1998). In their study of the product design firm IDEO, Hargadon and Sutton (1997) argued that ideas are unevenly distributed between groups, and that their combination can lead to the development of new products, as ideas from one group can address the needs of another. In the current context, this happens when firms that give to a community contribute to the evolution of the community's software product. In doing so, they are developing new knowledge about the ways it can be extended, refined, and adapted to new markets. The generation of such

new knowledge is an essential antecedent to the creation of new products (Kogut & Zander, 1992).

Norms of Modularity

In addition to acting as a source of product ideas, social ties also encourage firms to adopt a more modular approach to software development, which can increase the diversity of a firm's product portfolio and speed product development. Software development meets Herbert Simon's definition of a complex system, "one made up of a large number of parts that interact in a nonsimple way" (1962). Software manufacturers cope with this complexity by adopting a "modular" approach to development. I define modularity in line with Langlois (2002) and Ethiraj and Levinthal (2004). The modern automobile provides a useful analogy}, as "a general set of design principles for managing the complexity of such large-scale interdependent systems. It involves breaking up the system into discrete chunks that communicate with each other through standardized interfaces or rules and specifications" (Ethiraj & Levinthal, 2004: 161). The modern automobile provides a useful analogy; modules in software development are akin to the "parts" of a car. When a muffler breaks or a tire goes flat, drivers need only replace those individual parts. Software developers take a similar approach by building modules that extend the functionality of the core products, while maintaining some level of separation from other parts of the software.

Extant research has described how open source communities are more likely than private firms to adopt a modular approach to software development (MacCormack et al., 2006; Raymond, 2001). Open source communities have been found to encourage such

modular development in part because it allows for an efficient division of labor. I argue that if a firm decides to collaborate with an external community in developing software, it is more likely to use a modular design. Through modularity, disparate contributions can be more easily managed and integrated into the whole. Linus Torvalds, the lead developer behind the popular open-source Linux operating system, explained how modularity was critical to the success of the community:

> With the Linux kernel it became clear very quickly that we want to have a system which is as modular as possible. The open-source development model really requires this, because otherwise you can't easily have people working in parallel. It's too painful when you have people working on the same part of the kernel and they clash (Torvalds, 1999: 108).

Baldwin and Clark found that "modularity . . . create[s] opportunities for the exchange of valuable work among developers" (2000: 1116). I suggest that as firms increase the amount that they give to the open source community, the resulting social ties facilitate increased exposure to the kind of valuable work described by Baldwin and Clark; work made possible by the community's modular approach to software design. By developing and contributing modules to the community, firms that give are more likely to adopt a modular approach themselves. First, and most simply, social ties expose the firm to specific software modules created by the community that it may not have learned of absent its social ties; they can take and use these software "parts" for their own purposes. Put differently, giving enables a firm to think of the community's product not as single object, but as a collection of interoperable parts, many of which can be taken and used by the firm to address its own needs. Moreover, exposure to the benefits of modularity at the

community level can inspire employees to embrace the approach for their own projects. As firms that give see how a modular approach makes it possible for an open source community to combine the contributions of multiple, disparate, geographically distributed volunteers into a coherent and dependable product, they are more likely to consider how the approach may help advance their own projects.

A firm's embrace of modular design for community contributions affects its own product development efforts by allowing a firm to diversify its product portfolio. Modules can be recombined to make multiple products, targeting a specific need or market niche. Consider as an example how Microsoft packages its Office software suite, which includes several individual programs (or "modules") including Microsoft Word, Excel, and PowerPoint. Microsoft also makes other Office programs (or "modules") including its Access database, OneNote note taking program, and InfoPath communications platform. By recombining these programs in different ways, Microsoft can more easily target different markets and different price-points. A version of Office that includes only Word, Excel and PowerPoint may be offered as a low cost version for students, while one that adds InfoPath and Access can be sold at higher prices to Enterprise IT managers. Using this modular approach, these two products were developed with less independent effort. By enabling a firm to create new products by combining different modules (Baldwin & Clark, 2000) a modular approach also increases the speed of product development (Ethiraj & Levinthal, 2004; Ulrich & Eppinger, 2000).

Taking, Lower Giving, and New Product Development

When a firm takes from a community but has lower levels of giving, the relationship between taking and product development should be weaker. Under these

circumstances, taking should not necessarily lead to a higher number of products because the firm does not have the same access to product ideas.

Absent the social ties that come from giving to the community, a firm lacks access to a rich source of ideas for new products. It will be more difficult for it to learn of other products provided by firms engaged in the community. Knowledge of competitors' products will come later in a product cycle, typically when a firm announces its product to the broader market. The ability to generate original and new ideas will also suffer because a firm that does not give will develop fewer social ties and thus be exposed to fewer ideas. Without contributing to the community's product, the firm is not exposed to—and is unable to address—the challenges and problems of others in the community, depriving it of a source of new product ideas (e.g. Hargadon & Sutton, 1997; Nahapiet & Ghoshal, 1998) tacit knowledge (McEvily & Marcus, 2005) and even performance benefits (Dyer & Hatch, 2006).

The strength of social ties and community participation has been empirically tied to both liquidity events (Waguespack & Fleming, 2009) and alliance formation (Rosenkopf et al., 2001). Consistent with this work, I argue that a dearth of social ties will reduce the ways in which a firm communicates with an open source community. The various communications practices identified by Ancona and Caldwell (1992) will occur with less frequency, if at all. Firms that do not give to a community have no need to engage in task coordination behavior because they are not concerned with the integration of their contribution into the community's product. A lack of friendship ties makes scouting more difficult and ambassadorial behavior no longer necessary. Consistent with

their research, I suggest this scarcity of communication practices will adversely affect the firm's ability to acquire new product ideas from the community.

Moreover, when a firm does not give, the absence of social ties leaves it less exposed to the norms of modularity so important to an open source community. Removed from these norms, the firm is more likely to follow internal, insular development practices that primarily build the company's idiosyncratic knowledge. Internal software developers no longer need to embrace modularity in order to ensure that their code works well with disparate others. Under these circumstances, it becomes easier for internal teams to work in an inter-reliant (i.e. nonmodular) fashion. Using such an approach, new products are likely to take longer to develop because modification takes longer. The net effect is that, absent norms of modularity, the firm will produce fewer new products.

Taken together, access to product ideas and norms of modularity suggest that when a firm that takes from a community also gives back, it is likely to develop a higher number of new products.

> *Hypothesis 3. The relationship between taking from an open-source community and the number of new products will be positively moderated by giving to the open source community, such that the number of new products will be higher when giving is high than when giving is low.*

INTERACTIVE ENGAGEMENT AND INCREMENTAL INNOVATION

Innovation is an increasingly important means by which firms seek to achieve sustainable competitive advantage (Franko, 1989). To better understand the ways that

interactive engagement between a firm and a community influences organizational innovation, this section explores relationships between taking, giving, and one of the most researched characteristics of innovation, the degree to which an innovation is incremental in nature.

Radical vs. Incremental Innovations

Innovations have often been considered by the degree to which they are incremental in nature. Typically, scholars have argued that innovations can be classified along a continuum that stretches from radical to incremental (Dewar & Dutton, 1986; Henderson & Clark, 1990). At one end, radical innovations represent significant departures from past practices. These innovations are sometimes called "revolutionary," "breakthrough," or "disruptive" (Crossan & Apaydin, 2010). They disrupt an existing technological trajectory. Radical innovations occur when firms "ask a new set of questions, to draw on new technical and commercial skills, and to employ new problem solving approaches" (Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Ettlie et al., 1984)(Henderson & Clark, 1990: 9). In other words, they are more often discontinuous, and inspired by ideas that are separate from the firm's past. For firms, radical innovations may offer significant reward, but because they do not address the needs of current customers, can also be quite risky. Prior research on the antecedents of innovation show that radical innovations are more common when specialists are concentrated and a firm has an aggressive or ambitions technology policy

At the other end of the continuum, incremental innovations are characterized by modest improvements or extensions of existing products. These innovations meet the needs of current customers by relying on established designs (Dewar & Dutton, 1986; Ettlie et al., 1984; Henderson & Clark, 1990). Incremental innovations build upon and reinforce a firm's core competencies, as well as current technical capabilities. They are more common in decentralized and complex organizations (Benner & Tushman, 2003). In the short term, the outcomes of incremental innovations are more predictable because firms may rely on past experience to gauge their likelihood of success .

Incremental innovations are common in software development. Basic software development is incremental by necessity; developers must rely on established codebases and programming practices so that their improvements are compatible and do not break existing functionality. Incremental software innovation is also driven by market incentives; by focusing on incremental changes, software developers ensure that newer releases continue to meet the needs of existing customers, provide compatibility with other software, and minimize the need for customers to learn how to use new features. Most commercially available software follows such an incremental approach. Consider the popular computer operating system Microsoft Windows. Each new version offers some changes to encourage users to pay for an upgrade, but maintains consistency of both the design and functionality of previous versions. Doing so ensure that users familiar with Windows Vista or Windows 7 can use the latest version, Windows 8, with minimal training. Similarly, by focusing on incremental technical changes, Microsoft ensures that third-party software products that were compatible with previous versions will continue to work with the newer product.

Radical software innovations, on the other hand, typically target a new market. Apple's iOS operating system, arguably a radical innovation in that it ported the firm's desktop operating system to lightweight mobile devices, was created so that customers could use their fingers directly on a small phone screen. Before this innovation, input for mobile phones was limited to the use of a small keyboard (e.g., the Blackberry) or a pointing device called a stylus (e.g., Windows CE devices). Apple's radical innovation broke with these design traditions and introduced an operating system for mobile phones that used human touch. In doing so, it required users to learn a new way of interacting with their phones. It created a new software ecosystem, since programs written previously for other devices would not work with the new technology. Following the introduction of iOS in 2007, Apple has relied on incremental improvements to extend the mobile operating system to newer devices such as the Apple iPad. As Microsoft has done with Windows, this focus on incremental innovation has allowed Apple to leverage consumers' existing knowledge of how to use the iPhone with the new device, and provided compatibility with third-party software programs written for the iPhone.

I suggest that open source software is also relatively incremental, both for the benefit of users, and also for the geographically distributed engineers who contribute to the community's software. Contributing firms must consider numerous technical interdependencies to ensure that their contribution maintains compatibility with the existing product without breaking or altering functionality that has been adopted in previous versions. Incremental contributions that maintain compatibility and continuity with existing users are more likely to be accepted. Research on new product development

supports this assertion; Bonner and Walker (2004) has shown that highly embedded customer feedback was more likely to lead to the success of incremental products.

Conversely, radical innovations cause disruptions for software firms and members of the open source community. Radical software innovation may be perceived as threatening by others in the community, because software that introduces dramatically new functionality is by definition inconsistent with the needs of current customers. Should an open source community readily adopt contributions that are radical, the resulting product would be less likely to satisfy the incremental needs of current customers and more likely appeal to those using the more radical elements of the community's product. When users have such divergent needs, the community's software product can be at increased risk of "forking," a process in which the code is split into multiple versions incompatible with each other (Lerner & Tirole, 2002). In sum, firms that submit such radical innovations to an open source community are more likely to be out of step with community interests.

Taking, Higher Giving, and Incremental Innovation

The effect of firms taking code from open source communities on innovation incrementalism is likely dependent on the level of giving by the firm. Social ties created through giving influence the types of knowledge a firm takes from an open source community and impose social constraints on how that knowledge is subsequently used by the firm. First, consider when a firm taking from the community also engages in a higher level of giving. The type of knowledge a firm takes when it also gives is likely to be different than when a firm doesn't give. Giving requires a firm to learn specific technical details of the open source community's product, especially if a firm hopes to have its

contributions adopted. The social ties that form as a result of giving have been shown to facilitate the exchange of such detailed information between organizations (Ingram & Roberts, 2000; Uzzi, 1997). When social ties become a readily available source of knowledge, firms will rely upon them to solve various problems. Henderson and Clark (1990) argued that in technical environments, established communication channels are an antecedent of incremental innovation because they connect those trying to solve complex problems with those who have solved similar problems in the past. In their words:

> As a product evolves . . . communication channels develop and help engineers to work efficiently, but the evolution of the product also means that engineers face recurring kinds of problems. Over time, engineers acquire a store of knowledge about solutions to the specific kinds of problems that have arisen in previous projects. When confronted with such a problem, the engineer does not reexamine all possible alternatives but, rather, focuses first on those that he or she has found to be helpful in solving previous problems. (Henderson & Clark, 1990: 16)

Applied to the current setting, firms can rely on social ties to the community when trying to solve two types of technical problems: those related to either improving the community's software, or learning from others how to apply the community's work to the firm's own needs (Gruber & Henkel, 2006). Both of these technical problems lend themselves to the transfer of detailed knowledge that solves incremental problems, not radical ones.

In addition to the type of knowledge taken, firms that give may also be constrained in how they apply what they take. Giving fosters the development of multiple

social ties that collectively constitute what Ingram and Roberts call a "friendship network" where "the multiplex character of friendships . . . creates increased levels of trust, empathy, and reciprocity" (Henkel, 2006; 2000: 389). Social ties improve communications between organizations, and those organizations which share them are more likely to approach problem solving together (Uzzi, 1996). Turning to these communication channels for help can become automatic and natural (Henderson & Clark, 1990). When ties between a firm and community are especially strong, communication norms between the two can approach what Lubatkin et al. (2001) describe as a *reciprocal learning alliance*, where firms' "primary intent is to co-experiment and leverage each other's unique, but complementary, knowledge structures" (1362).

But a friendship network also "enables mechanisms of social control" (Ingram & Roberts, 2000: 389) that enforce norms that support the group's common interests (Ingram & Roberts, 2000). The common interest among firms that give to a community includes the development of the community's product, the survival and governance of the community, and the effective application of community solutions to the needs of member firms.

As discussed earlier, the norms of open source communities are decidedly incremental, because the common interest shared by participants is the continued development of the community's product. Incremental innovations are more consistent with these common interests because they are in line with the incremental norms and technical needs of the community. Independent firm actions that are incremental signal to the community that the firm is not developing products that may one day compete directly with the community's offerings. Both behaviors help ensure that social ties

remain intact. Pursuing incremental innovations such as improvements to the community's products represents lower risk behavior than pursuing radical innovations that may constitute a break with the interests of the community and could jeopardize these social ties. Conversely, using what they take from the community for radical innovations represents a departure from the common interests of the community and the friendship networks that form.

Firms that seek community adoption of their contributions are better off when they do not pursue developing radical innovations, because they are less likely to meet the community's more incremental technical needs. Moreover, open source communities often attract individual contributors and smaller firms that face resource constraints, and are less able to develop radical innovations.

Finally, radical innovations that remain in the firm's hands can signal to the community that the firm is positioning itself to pursue products that don't target the needs of the community. These forms of firm behavior can encourage other group members to respond "in the form of less friendly treatment" (Ingram & Roberts, 2000: 392), and jeopardize social ties.

Taking, Lower Giving, and Incremental Innovation

When a firm takes from a community but has lower levels of giving, the relationship between taking and incrementalism should be different. Under these circumstances, taking should not automatically favor incremental innovations. Absent the social ties that come from giving to the community, the complementary knowledge structures described by Lubatkin et al. (2001) are less likely to form. The firm has had no

input into the development of what they are taking, nor influence over its direction. As a consequence, the community does not act as a source of detailed technical knowledge, and a firm must alone determine just how to use and apply what they have taken from the community. Separated from the influence of open source communities, the knowledge these firms take from the community is less likely to be applied to incremental innovations, because they are removed from the relatively incremental norms of the community. Any changes they make to the community's product will reflect their own knowledge alone. This idea is also supported by Bonner and Walker (2004), who found that highly innovative products are more likely to have an advantage when input is received from a heterogeneous (as opposed to embedded) group.

Moreover, when firms don't give, they aren't subsequently constrained by resulting social ties, and the formation of friendship network between a firm and a community is not likely. Without a friendship network, a firm is more likely to view an open source community from the perspective of a "free rider," that is, as a freely available pool of software that requires no payment or acts of reciprocity. Absent a friendship network, a firm can use what it takes from the community for radical innovations without damaging friendships. There are no friends to discourage the pursuit of radical innovations by explaining how such innovations could damage the community's reputation with users accustomed to steady, incremental improvements. Furthermore, absent social ties, a firm is less likely to care about jeopardizing the community's reputation in the first pace. In sum, a firm that does not give is more likely to use what it takes for radical innovations, because absent a friendship network, there is no risk of reprisal from friends in the community.

Taken together, these arguments suggest that when a firm that takes from a community also gives back, innovations are likely to be more incremental in nature.

Hypothesis 4. The relationship between taking from an open source community and innovation incrementalism will be positively moderated by giving to the open source community, such that innovation incrementalism will be higher when giving is high than when giving is low.

INTERACTIVE ENGAGEMENT, DEVELOPMENT, AND DEBUG TIME

In fast-paced industries such as high technology, the speed of software development can significantly impact firm profitability (Eisenhardt & Tabrizi, 1995). In one study, Vesey (Gupta & Wilemon, 1990; 1991) showed that holding budgets constant, a six-month delay lowered 5-year earnings by 33%. One reason is that customers will often decide to purchase a competing product rather than wait for one that has been delayed (Blackburn et al., 1996). Despite the benefits of timely development, software products are frequently delivered late (Van Genuchten, 1991). Some have argued that part of the reason is because software development is more of an art than a science, and that managing software development is exceedingly difficult (Blackburn et al., 1996), in part because adding labor can actually lengthen development time (Brooks, 1995).

To understand the ways that interactive engagement between a firm and an open source community influences the software development process, this section explores relationships between taking, giving, and resulting development and debug time. I define *development time* as the duration of time spent writing software for a particular product,

and *debug time* as the portion of that time spent identifying and removing errors from a product's codebase.

Below, I argue that the effect of taking from an open source community on both development and debug time will depend on the level of giving by the firm. Specifically, social ties that result from giving shorten development and debug time because the firm can turn to a network of friends in the community for assistance at multiple stages in the product development process.

Taking, Higher Giving, Product Development, and Debug Time

The social ties that result from giving to an open source community provide a firm with both technical and normative information useful for speeding development and debug processes. As mentioned earlier, researchers have found open source communities to provide a firm with access to knowledge (Goode, 2005) and expertise (Gruber & Henkel, 2006; Von Krogh & Von Hippel, 2006). Below, I explore how giving facilitates access to this knowledge and expertise and speeds each of the three stages of software development.

First, research has shown that firms that devote more time to the early stages of product design (e.g., performing customer research and collaborating with suppliers) enjoy faster development processes. In their surveys of software developers, both Blackburn and Scudder (1996) and Gupta and Wilemon (1990). The same is true with an open source community} found that firms which allocated more time up front to determining customer needs had faster development processes. In the current setting, social ties to the open source community provide a valuable sounding board for product

ideas and plans. Past management research has demonstrated that more frequent communications with suppliers and customers provides new information (Brown & Eisenhardt, 1995; Katz & Tushman, 1981). The same is true with an open source community; in the early stages of a development process, social ties provide the pathways through which ideas for products are exchanged and vetted.

Second, social ties promote the transfer of fine-grained and tacit information that can be important as development moves beyond early planning. Much of the real work of software development involves transforming ideas into workable code. For this stage, in the words of Blackburn and Scudder, "virtually no amount of management technique and team coordination can overcome a lack of talented developers and coders" (1996: 891).

I argue that the social ties that result from giving to a community provide an important source of talent that can augment the work of employees within the firm. Several studies have shown that effective use of "gatekeepers" contributes to successful product development (Brown & Eisenhardt, 1995; Katz & Tushman, 1983), because these groups can import useful new information from the outside environment. As argued by Brown and Eisenhardt, "when [external] communication is frequent, project teams are likely to develop an absorptive capacity such that they become more efficient in gaining and using the information being conveyed. Both of these factors should improve the productivity *and pace* of the development process" (1995: 368, emphasis added; Katz & Tushman, 1983). Defined as a firm's "ability to recognize the value of new information, assimilate it, and apply it to commercial ends" (Cohen & Levinthal, 1990: 128), absorptive capacity in this context means that firms are able to extract ideas via their

social ties to the community and apply them to reducing both development and debug time.

Furthermore, research has shown how social ties enable firms to address challenges more quickly (Uzzi, 1996) and improve performance by fostering collaboration (e.g. Ingram & Roberts, 2000). One way this happens is by reducing uncertainty. Gupta and Wilemon (1990; Uzzi, 1996) found that a majority of respondents cited "technological uncertainty" (including the need for ensuring compatibility) as a reason for development delays. When social ties are made to a community of experts with tacit technical knowledge, the firm can turn to those experts for help addressing uncertainty and finding ways to solve problems. In other words, the more a firm gives to a community, the more resulting social ties will provide tacit and detailed feedback, speeding development.

For example, Milson et al. (1992) argued that firms wishing to accelerate new product development should adopt five actions: simplify, eliminate delay, eliminate steps, speed operations, and perform processes in parallel. Social ties to an open source community have the potential to encourage a firm to follow such practices: development becomes simpler when it conforms to the modular norms of the community (Baldwin & Clark, 2006), and delay and additional steps can be avoided, because rather than relying on support from one individual, a firm can turn to an entire community of technically proficient assistants.

Third, friendships in the community can become an "extra set of eyeballs," used to analyze and debug company code in the later stages of development. This works in two

ways. First, feedback from community members will help the firm ensure that its products and contributions are technically compatible with the work of the community. Second, anecdotal evidence suggests that developers pay more attention to documenting and detailing their code when they know that others will see it. Doing so helps promote a more cautious development process that may make bugs less likely to occur in the first place.

Taking, Lower Giving, Product Development, and Debug Time

When a firm takes from a community but has lower levels of giving, the relationship between taking and both development and debug time should be weaker. Without social ties, a firm that takes code is not able to count on the assistance of friends in the community to help speed development and address bugs.

Consider the following example. A firm that commonly takes from a community downloads open source code, hoping to integrate the software into their own products and services. Through the development process, inevitable challenges arise—parts of the open source software cannot be easily understood, and development slows. Without friends in the community that the firm can quickly turn to, they must seek assistance by posting questions on public discussion boards, hoping that somebody from the community will help a firm that has never contributed to the community's product. Waiting for a reply further stalls development. Moreover, the fact that the firm does not have strong social ties with the community may influence the nature of feedback the firm receives. This example illustrates a case where the absence of social ties prevents a firm from accessing additional assistance with both development and debug processes.

Moreover, absent social ties, there are fewer consequences for firms that launch products that are incompatible with the work of the community, or at the extreme, develop a "forked" version of the community's product that is intentionally designed to be incompatible. In other words, when a firm doesn't give to a community, social ties are not present to provide assistance or new ideas that help to focus and simplify a company's products.

Taken together, these arguments suggest that when a firm takes from a community and also gives back, both development and debug time should be shorter.

Hypothesis 5a. The relationship between taking from an open source community and product development time will be negatively moderated by giving to the open source community, such that product development speed time will be lower when giving is high than when giving is low.

Hypothesis 5b. The relationship between taking from an open source community and product debug time will be negatively moderated by giving to the open source community, such that product debug time will be lower when giving is high than when giving is low.

SUMMARY OF HYPOTHESES

Figure 2 and Table 2 summarize the hypotheses of this research. In Figure 2, I graphically show the various hypothesized relationships between the independent variables, taking, giving, and interactive engagement (taking X giving) with the dependent variables. Table 2 provides a textual summary of the hypotheses.

Chapter 5: Methods

The purpose of this study is to investigate the consequences of firm engagement with an open source community. In this chapter, I describe the methods I used to conduct the study.

THE CONTEXT: OPEN SOURCE COMMUNITIES

Open source communities develop and freely distribute software to anyone, from individual amateur programmers to large corporations. Over the last quarter century, these voluntary organizations have emerged to become an important part of the way high-technology firms do business. The growth of open source software projects is impressive. The open source software distributor SourceForge listed a mere 10,000 open source projects in 2001 (Murray & O'Mahony, 2007). As of June 2011, that number had grown to over 260,000 (Sourceforge.net).

Large-scale open source software projects have a significant impact on technology companies. Linux, now 20 years old, ran 91% of all supercomputers in November of 2010 (Top500.org, 2010), and is a popular embedded operating system in consumer devices (Henkel, 2006). Linux, together with three complementary open source platforms including the Apache Web-server software, MySQL database, and PHP programming language, are known collectively as "LAMP," and provide an ecosystem that enables the creation and hosting of dynamic Internet websites with little involvement from traditional "closed" software suppliers (West, 2007).

Bonaccorsi and colleagues (2006) called open source a "radical innovation" for its potential to disrupt the software industry. In part because of LAMP, the significance of open source software is growing for large firms. Consider one anecdote from industry

giant Microsoft. In 2007, CEO Steve Balmer stated, "We've worked very hard on making the value of a commercial company surpass what the open source community can deliver, because frankly, it's not a business model we can embrace. It's inconsistent with shareholder value" *(Wilcox, 2007)*. Two years later, Microsoft funded and launched the CodePlex Foundation to promote open source software, and appointed Sam Ramji, the firm's former Senior Director of Platform Strategy, as its interim President *(Edge, 2009)*. This anecdote illustrates how open source is a movement that even industry giants have difficulty ignoring. The implication of open source for large firms was investigated by West (O'Mahony, 2007), who examined how three technology leaders—Apple, IBM, and Sun—each used a mix of proprietary and open software to advance their strategic objectives. He concluded that a three-stage evolutionary process led these software leaders toward integrating open source: firms that began as proprietary software leaders eventually adopted open standards, which in turn led to the acceptance and inclusion of open source code.

Open source software communities consist of independent programmers, typically not paid by the community, who encourage each other to take, improve, and then redistribute improved software to others (2003). Because they are bound by far more egalitarian norms that those common in private software firms, some have suggested that open source has more in common with scientific communities than with private firms (Kogut & Metiu, 2001). Because their egalitarian motives and norms are far different than other sources of innovative ideas (Podsakoff et al., 2003), the potential merits of engaging these new organizational forms can be difficult for firms to evaluate. This

dissertation directly examines the consequences of such firm engagement with an open source community.

SUBJECTS AND PROCEDURES

Subsequent to the inductive phase of my research, I conducted a questionnaire-based quantitative study to investigate the hypotheses developed in Chapter 4. I contacted leaders of the Drupal Association, the nonprofit arm of the Drupal open source community, to explore the possibility of conducting a survey-based research study. They provided permission to proceed, and later reviewed and made suggestions regarding my questionnaire.

Drupal provides *content management system* (CMS) software that allows users to create Internet websites that automatically pull in content from a variety of sources. Drupal creates software by coordinating the contributions of geographically distributed volunteers. Private firms are actively engaged in each community, allowing their employees to develop and enhance the community's product as they undertake their own projects.

Drupal is both the name of an open source software platform and the name of the community responsible for its development. Drupal is currently enjoying a surge of popularity, and is now used by approximately 2.3% of all websites worldwide and 71% of all websites that use a content management system (W3Techs, 2013), respondents were assured that their answers would remain anonymous and that the study has no expected "right" or "wrong" answers.

The typical respondent was expected to be either an executive or developer in a software firm. Based on previous interactions with members of Drupal, I anticipated that

most of these companies would be small to medium in size (i.e., fewer than 50 employees), and be privately owned. I successfully collected questionnaire responses from 250 representatives of Drupal organizations, each of which reported that they work with the Drupal open source software community. I conducted a preliminary statistical power analysis. In order to achieve a statistical power of 0.8 with an effect size of 0.15 with three independent variables (giving to the community, taking from the community, and their interaction) with a p-value of 0.05, according to this analysis, I needed a sample of at least 76 firms. Thus, the questionnaire response of 250 well exceeded this target.

Surveys were made available online in the spring and summer of 2012, and promoted at a Drupal Conference in Denver, held from March 19 to March 23, 2012. Survey responses from a total of 250 organizations were collected between March 19 and April 22 of 2012. This response rate represents approximately 41% of a total estimated target population of approximately 600 companies that were members of the Drupal Association in March of 2012 (J. Redding, personal correspondence, January 9, 2013).

Of these 250 responses, 167 were collected during the Drupal Conference in Denver in March of 2012, and 83 were collected after the conference. Post-hoc independent samples t-tests, conducted using a Bonferroni correction, indicated that only one control variable, firm size, differed between responses collected at the Drupal conference and afterwards}. Developer conferences, held twice a year in both the United States and Europe, routinely sell out. Drupal is used to produce prominent websites including WhiteHouse.gov, Economist.com, and the main University of Texas at Austin website, utexas.edu.

Drupal is an ideal setting for this study for two reasons. First, Drupal has been widely embraced by firms. While individuals also participate in the community, a growing number of companies use Drupal to create websites, making this community an ideal setting to find managers who can speak from the perspective of making firm-level decisions. Second, like most open source communities, Drupal's governance processes are relatively meritocratic, offering contributors (in this case, firms) the opportunity to play a role in community governance and providing a fertile ground for the study of interactive engagement.

The leadership of Drupal agreed to cosponsor the research effort. I first became familiar with the Drupal leadership in 2010, when I attended conferences on open-source management Austin, as well as the Drupal Association's own conference in San Francisco. The transcripts of recorded interviews, most with CEOs of software firms working with Drupal, provide the basis for my inductive pilot study described in Chapter 3. Since 2010, I have maintained close contact with members of the Board of Directors of the Drupal Association, and was confident that their sponsorship of my study would be helpful in boosting questionnaire response rates.

SAMPLING PROCESS

Data were collected using an online questionnaire approved by the Institutional Review Board at the University of Texas at Austin. The full questionnaire is included in Appendix A. The questionnaire was pretested with 16 doctoral students who provided comments. The use of questionnaires to collect data on both independent and dependent variables may introduce common-method bias. I made efforts to minimize any perceived connection between the independent and dependent variables in the questionnaire. Also,

consistent with recommendations in Podsakoff et al. (2003) firms that responded at the conference were significantly larger than those collected later. This may reflect the possibility that larger firms, with more resources, were better able to send employees to participate in the conference. There were no other significant differences for any other control and dependent variables.

Respondents were also split into two categories based on different promotion programs. A t-shirt was offered to 119 respondents in exchange for completing the full survey; the remaining 131 were entered into a raffle drawing to win an Apple iPad tablet computer. Post-hoc independent samples t-tests, conducted using a Bonferroni correction, indicated that there were no significant differences between these groups for any dependent or control variables.

MEASURES

Whenever possible, I adapted existing validated questionnaire items from other management studies to measure variables of interest. However, validated questionnaires do not yet exist for my independent variables, giving and taking, because these measures were derived through the inductive means described in Chapter 3. Specific questionnaire items for all measures are included in Appendix A. All measures were reviewed by Drupal administrators, and some slight revisions were made according to their constructive comments. Drupal officials thus provided an informal validation of the constructs and the specific questions. As described in the following section on analysis, interviews and exploratory factor analyses were also used to validate these constructs and the related questionnaire items.

Independent Variables

To create my independent variables for giving and taking, I used eighteen questionnaire items that measured various aspects of exchange between a firm and an open source community. These initial items were developed following interviews conducted in the first phase of the research (see Chapter 3, "Inductive Pilot Study"), and further refined through discussions with the Drupal Association and through the pretesting processes. I thus created 18 questionnaire items to measure my two independent variables: taking from and giving to a community. Measures were designed to collect information on the type, intensity, and duration of both constructs. For example, I collected information on such firm behaviors as soliciting assistance, incorporating community code into the firm's own software products, volunteering, and offering assistance to other members of the community.

Following data collection, I used exploratory factor analysis (EFA) to reduce these 18 items into factors that I named "giving" and "taking,", in accordance with the propositions presented in Chapter 3. Typically, EFA has three major steps, starting with the extraction phase, where typically a "principal axis" extraction method is used. One then selects whether or not to rotate the solution and decides how many factors to rotate. Third, items that cross-load on multiple factors or provide nebulous theoretical merit to an underlying construct are eliminated from subsequent analyses.

In my first factor analysis, I examined all 18 items using principal axis extraction. I elected to use a promax rotation because initial interviews suggested that my theoretical constructs (i.e, the factors) might be correlated; a promax rotation is an oblique rotation

that allows such correlation between factors. This analysis identified four factors with eigenvalues over 1. Of the 18 survey items, six items had high cross-loadings with other items, and were subsequently removed through a process of multiple exploratory analyses. The final 12 items were then analyzed again using the same extraction (principal axis) and rotation (promax), this time specifying a four-factor solution (each of these factors had an eigenvalue greater than 1.0). The pattern matrix of these four factors is shown in Table 8.

To test the robustness of these results, I conducted a variety of analyses using alternate extraction and rotation methods, including principal factors extraction and varimax and oblique rotation. I also made several runs using various numbers of factors to rotate. These analyses consistently produced similar results, and were also consistent with the more common criterion of eigenvalues > 1 for rotation. My description of each of the four factors is described below.

Factor 1, which I have named "duration of engagement," includes high loadings on three items that all ask respondents about the number of years their firm was working with the Drupal community. However, this factor contained items originally intended for both taking and giving, which was not consistent with my prior theoretical development. I concluded that this variable does not cleanly identify either giving or taking, but instead, simply measures how long a firm was engaged in the community. Moreover, this construct does not represent any hypothetical development from the literature or my theoretical analysis. Thus, this construct was dropped from subsequent analysis.

Factor 2, "taking" includes three items that measure the intensity by which a firm is reliant on Drupal. Specifically, this item includes high loadings for using Drupal since

the firm was founded, the percentage of employees using the community's software, and the percentage of products and services based on the community's software. Accordingly, consistent with my original intent, I labeled this item "taking."

Factor 3, "giving code" contains items with high loadings on three questionnaire items asking about firms' contributions to the Drupal community. This construct related to the heart of my prior theoretical development. Two items—contributions to "core" and "themes"—are both software related. These items cleanly measure the extent to which a firm gives software and related documentation to the community, thus the name "giving code."

Factor 4, "giving help" contains high loadings for items measuring the degree to which the firm encouraged employees to volunteer in the community, paid employees to volunteer, and assisted other firms in the community for free. Clearly, this factor measures the extent to which a firm is willing to assist others in the community, hence the label "giving help."

These findings are interesting, as I originally surmised that I would identify single factors for both giving and taking. Yet, here, in the exploratory factor analysis, I identified a single factor for taking (as expected) but also identified two distinct factors that measure different aspects of giving—either giving code or help. Rather than choose one of these constructs as the single measure of "giving," or collapsing the components into a single giving construct (with a Chronbach alpha of 0.65), I concluded that the factor analysis identified two distinct forms of giving, and decided to continue my analysis using both giving factors as separate independent variables. I named these two giving factors "giving code" and "giving help."

Finally, to create the final independent variables giving code, giving help, and taking, I summed the standardized values for each respective component variable. Cronbach alpha results were all satisfactory, at .7, .8, and .8 for the items combined for taking, giving code, and giving help, respectively – all exceeded the 0.70 recommended minimum for multi-item scales (Santos, 1999).

Dependent Variables

Software Creation Productivity. Most definitions of productivity refer to a ratio of output divided by input. For example, in their review of productivity in software development, Sudhakara et al. write, "Traditionally, productivity can be defined as a ratio of output units produced to the input units of effort" (Fleming & Waguespack, 2007). In my case, I am interested in a definition of productivity specific to this industry, yet still remaining true to the notion of output/input. Subjects from my pilot study referred to the notion of software productivity; that is, how productive a firm is in terms of producing a unit of software. Subjects had an intuitive understanding that the higher the cost to produce a unit of software, the lower the productivity.

In my survey, I approached this issue by asking each firm, "In the last year, what was the average total cost (including development and support) of a typical product or client job?" I call this "average cost of client job." This question is a measure of input. To convert to productivity, I take the inverse of this measure; that is, one unit of output (a client job) divided by the average cost of a client job. Note that as the average cost of a client job goes up, productivity goes down. Conversely, as the average cost of a client job goes up, productivity goes down. This is consistent with the subject's intuitive understanding of software productivity. I examined whether the distribution of the

productivity measure was linear. It was found to be nonlinear, so I converted the measure to a linear distribution by using a logarithmic transformation of this variable for subsequent analysis.

Social Ties. I adapted a questionnaire item used by Westphal (1999) to measure social ties between a firm and the community. Specifically, I asked respondents, "How many individuals outside your company, but involved in the Drupal community, do you consider to be acquaintances or friends?"

Number of New Products. Consistent with Sivadas and Dwyer (2000), I collected information on new products using questionnaire items. Specifically, respondents self-reported an estimate of the number of new products or client jobs for each of the past three years.

Incremental Innovation. As discussed in Chapter 3, incremental and radical innovations lie on opposite sides of a continuum (Henderson & Clark, 1990). Incremental innovations include improvements to existing products and the use of established designs (Dewar & Dutton, 1986; Henderson & Clark, 1990), while radical innovations are typically new products or represent significant departures from past designs and practices. I adapted items from Green and Gavin's (Dewar & Dutton, 1986; Ettlie et al., 1984; 1995), validated measure of this scale.

Product Development Time. Similar to the methods used by Lee and Xia (2010), I used a questionnaire to collect estimates of the development time for the typical company product or client job.

Product Debug Time. Consistent with others (e.g. Jorgensen, 2008), I have designed a questionnaire item to assess debugging practices. Specifically, a self-report item asks respondents to estimate the typical percentage of product development time spent in debugging processes.

Control Variables

Strategic Posture. To capture the extent to which a firm is either conservative or risk-taking, I adapted questionnaire items used by Covin and Slevin (1989) and Lumpkin and Dess (1990; 1996). The questionnaire items assess technological innovativeness, risk taking, competitive intensity, and independence.

Firm Size. Firm size has shown significance in other studies of firms adopting open source business models (Bonaccorsi et al., 2006), including some that have shown smaller companies to be more likely to donate code to the Linux open source community (Gruber & Henkel, 2006). I measured firm size as the number of employees.

Firm Performance. Since my sample consisted predominantly of private firms, market- and accounting-based measures of performance are not readily available. Consistent with others who have studied performance among private firms, I adapted Likert-scale questionnaire items from Dess (1987; Henkel, 2006). These items ask respondents to compare their firm's performance to industry peers in both total sales growth and a personal assessment of overall performance.

Growth Rate. I control for the annual growth rate by calculating the annual change in company size, as measured by the change in employee count from 2010 to 2012.

Founder as CEO. A single questionnaire item asked whether or not the firm founder is the current CEO.

Age of Firm. Respondents were asked to provide the year the company was founded.

Summary of Methods

To summarize, I developed a questionnaire to measure various aspects of engagement between firms and the Drupal open source community. The most important variables were giving and taking from the Drupal community. I developed original questionnaire items for these measures. Other measures were in most cases taken from existing management literature.

With the cooperation of the Drupal Association, I collected data from 250 organizations through the use of an online questionnaire. For the items intended to measure giving and taking, I performed an exploratory factor analysis, and found strong support for three constructs of interest: taking, giving code, and giving help. Cronbach alpha analysis showed high reliability for each of the constructs.

Chapter 6: Results

I tested my hypotheses using a sample of 250 responses to the 2012 Drupal Business Survey. I originally collected 281 responses, but 31 represented second or third responses from the same organization. To identify the most representative response from each organization, I selected the highest-level employee who had completed the most responses (i.e., the fewest items left unanswered).

I used ordinary least squares (OLS) regression to test each hypothesis. Table 16 summarizes the results of all hypothesis tests. Table 9 provides descriptive statistics and correlations among my key variables. Tables 10 through 15 present the regression results for each of my hypotheses. Each regression begins with a model containing only my control variables. For Hypotheses 1 and 2, which tested the direct effects of taking and giving, respectively, the independent variables were included in a second model. For the remaining hypotheses, which examined the interaction effects of my single taking and two giving variables, I constructed a third model that added the interaction terms.

OLS regression is appropriate because it allows for testing the effect of continuous independent variables, and for the inclusion of various controls (i.e., firm size, growth). This approach assumes independence, and that the dependent variables are normally distributed. For variables that were not normally distributed in the raw survey data (i.e. productivity, firm size, and firm age) I used a log transformation in my regression.

OLS results can be, however, sensitive to both multicollinearity and the influence of outliers. To assess multicollinearity, I calculated tolerance statistics and Variance

Inflation Factor (VIF) values for each regression test. VIF values were all well below the recommended cutoff of 10 (Cohen et al., 2003), indicating that multicollinearity is not a major concern. To address the potential influence of outlier variables, I conducted a robustness check using Cook's Distance, which is useful for identifying outlier cases that may affect regression accuracy. Following Bollen and Jackman (1990), for each regression, I excluded cases where Cook's Distance was greater than 4/n, where n is the number of cases included the original regression.

Table 9 provides descriptive statistics and correlations of my key variables. Several of the significant correlations are interesting in their own right. Considering the independent variables, taking is correlated with both forms of giving, suggesting that firms tend to engage the firm comprehensively. Considering the heuristic developed in Chapter 3, this correlation suggests that firms are more likely to be either "unengaged" or "interactively engaged," and less likely to fit in the "open handed" or "free riding" categories. Much of the literature on open source suggests that the free nature of open source code would entice firms to free ride, but this correlation would suggest otherwise. Interestingly, the correlation between taking and giving help is higher (.51), that it is with giving code (.27). I believe this may suggest that firms that need to take but are not yet able to give code seek to give back by helping others. While the factor analysis clearly indicated that giving code and giving help are discrete factors, we see here they are also highly correlated (.48), suggesting that when firms engage in one form of giving, the other is not left aside.

Considering controls, firm age and size are behind some of the most significant and sizable correlations. Age is highly and negatively correlated with taking (−.72) and

giving help (−.31), which I attribute to the fact that older firms' business models are more likely to predate the Drupal Community. Firm size is also highly and negatively correlated with taking (−.73) and Giving Help (−.32), which I believe signals that larger firms have a more diverse portfolio of activities, many of which do not rely on the use of Drupal code or its community.

Hypothesis 1 proposed that taking from the community would be associated with higher software creation productivity. Hypothesis 1 was supported (Table 10, beta = .20, $p < 0.05$).

Hypothesis 2 proposed that giving to the community would be associated with social ties between the firm and the community. I tested this hypothesis with each of the two variables identified for "giving," in my exploratory factor analysis. The hypothesis was supported for giving code (Table 11, beta = .12, $p < .001$), but not for giving help (Table 11, beta = .02, n.s.).

Hypothesis 3 proposed that the interaction of taking and giving (which I called "interactive engagement") would be associated with an increase in the number of new products. This hypothesis was not supported for either the interaction of taking and giving code (Table 12, beta = −.01, n.s.) or for the interaction of taking and giving help (Table 12, beta = .02, n.s.).

In Hypothesis 4, I proposed that the interaction of taking and giving would be associated with an increase in incremental innovations. This hypothesis was supported for both taking and giving code (Table 13, beta = .04, $p < .05$). For taking x giving help, I found a significant result in the opposite direction than hypothesized (Table 13, beta = −.04, $p < .05$). In other words, the interaction of taking and giving help led to innovations

that were more radical in nature. I discuss this unexpected finding in the next chapter. These significant interactions are illustrated in Figure 3.

Finally, Hypotheses 5a and 5b proposed that the interaction of taking and giving would lead to reduced development and debug time, respectively. Hypothesis 5a was not supported for either taking and giving code (Table 14, beta = −.02, n.s.) or for the interaction of taking and giving help (Table 14, beta = .001, n.s.). For debug time, Hypothesis 5b, the interaction of taking and giving code, was not significant (Table 15, beta = −.01, n.s.), but the interaction of taking and giving help was significant in the opposite direction as hypothesized (Table 15, beta = .03, $p < 0.05$), suggesting the interaction of taking and giving help is associated with longer debug times. This significant interaction is illustrated in Figure 4.

SUMMARY OF HYPOTHESES TESTING

Table 16 shows the results of my hypothesis tests. Overall, some hypotheses were confirmed, and some were not. I found support for the direct effect of taking on productivity and of giving code on social ties. My results were more mixed when considering the interaction of taking and giving either code or help. The interaction of taking and giving code was associated with increasing incremental innovation, but not with an increase in the number of new products, lower development time, or lower debug time. The interaction of taking and giving help was not associated with an increase in new products or lower development time, but surprisingly, it was associated with more radical innovations and a longer debug time.

Chapter 7: Discussion and Conclusions

This dissertation was inspired by the observations of entrepreneurs building their firms while cooperating with open-source communities. I was further motivated by research that explores the influence of communities on innovation. My goal was to answer the research question, "How does variation in the type and intensity of participation in an external open source software community affect outcomes related to firm-level innovation?"

Since this question was inspired primarily by phenomenological observation, I began with an inductive pilot study, recording in-person interviews with executives at software firms that worked with the Drupal open source software community. This process provided narratives that clarified how firms interact with an open source community. In Chapter 3, I explained how these firms varied in the degree that they took software, ideas, and help from the community, and also in the degree they give code, help, and support back to the community. Finally, I explained how firms that combined both giving and taking showed enthusiasm for the community and concern for its continued development, in addition to displaying comfort in cooperating with competing firms. These observations inspired the subsequent development of hypotheses.

Following this pilot, I conducted a quantitative study with two goals: (1) develop an instrument to measure the exchange relationship between firms and an open source community, and, (2) test several hypotheses related to the consequences of these exchanges. To collect data, I formed a partnership with the Drupal Association, the nonprofit arm of the Drupal open source software community, and collected survey data from 250 organizations.

I presumed that my analyses would identify "giving to" and "taking from" the community as discrete behaviors. Exploratory factor analyses found that this was true for taking from the community, but the concept of giving was found to split into two discrete concepts of "giving code" and "giving help." This finding is interesting in its own right, because it demonstrates that these are two discrete behaviors, and allowed me to examine their relationships with my dependent variables independently of each other.

Giving code and giving help were significantly correlated (.481**), and both were significantly correlated with the formation of social ties (.556** for giving code and .418** for giving help).Younger firms were more likely to give both code and help. However, firms that gave help tended to be younger and more likely to be led by their founder than firms that gave code. This may occur because giving help is ostensibly easier that giving code}. This result is not surprising, and was further supported by interviews conducted as part of my inductive pilot study; younger firms, who are typically more resource-constrained, can more easily give help as they develop the capabilities to give code. In other words, giving help may be a vehicle for firms to acquire the skills they need to make more substantive code contributions later.

DISCUSSION OF HYPOTHESIS TEST RESULTS

I continued hypothesis testing using the single "taking" variable and both "giving" variables. Table 16 reports the findings of each of these hypotheses. Hypotheses 1 and 2 explored the direct effects of taking and giving, respectively. I found support for my first hypothesis, which built on prior work suggesting that taking from the community would lead to an increase in software-creation productivity (e.g. Bonaccorsi et al., 2006; Gruber & Henkel, 2006; Samuelson, 2006). This result is not surprising, and was further

supported by interviews conducted as part of my inductive pilot study; most executives confirmed that taking from the community allowed their firms to reduce the cost of internal software development (e.g. Samuelson, 2006), and their own research and development efforts (e.g. Gruber & Henkel, 2006). Finding support for this hypothesis also lends further credibility to statements from interview participants that they were able to apply ideas and help taken from the community (Table 5) to bolster their firm's productivity.

Consistent with my finding that giving code and giving help were two discrete activities, I conducted two tests for any hypotheses that included giving to the community, testing the effects of both giving code and giving help. Hypothesis 2, which argued that giving would lead to an increase in a firm's social ties with the community, was supported for firms that gave code, but not for those that gave help. At first, this seems counterintuitive; we are accustomed to thinking that the benevolent act of giving help would naturally generate goodwill and promote the development of social ties. However, giving code may be more indicative that that a firm had the capability to provide something that a larger portion of the community found valuable.

Giving help can be comparatively short lived and less frequently observed. A second possibility is that these two variables have differential effects on the number of interactions between firm and community boundary spanners. While giving code is an act that benefits the community directly, giving help may be more ethereal, directed toward a smaller audience, and as such, less frequently recognized as beneficial in the long run. In other words, firms may not interact with as many individuals in the community by helping as much as they do by giving code, leading to different effects on the

development of social ties. Giving code also requires more frequent use of the three communication practices identified by Ancona and Caldwell (1992) that I suggested would lead to the acquisition of social ties—task coordination, impression management, and scanning for useful information.

The remaining hypotheses proposed that the interaction of giving and taking, or what I have called "interactive engagement" would lead to a higher number of new products, innovations that are more incremental than radical, and both lower development and debug times. I tested the moderating effects of both giving code and giving help on taking. Here again, outcomes differed depending on whether the firm gave code or gave help.

Hypothesis 3 argued that giving would positively moderate taking, increasing the number of new products. The moderating effect was not supported for either interaction. In fact, firm size was the only significant indicator in all three models, with larger firms statistically realizing a larger number of new products. This finding may in part be a consequence of a low sample}. Giving help does not have this limitation; only 109 firms completed enough of the questionnaire to accurately test this hypothesis.

The fourth hypothesis argued that innovations resulting from interactive engagement would be more incremental in nature. My results were mixed, finding support in that giving code positively moderated taking (.04, $p < .05$), leading to more incremental innovations. However, I found that giving help significantly and *negatively* moderated taking (-.04, $p < .05$), leading to innovations that were more radical in nature. Both of these significant interactions are illustrated in Figure 3. Why might this be the case? Again, it is important to differentiate giving code and giving help. As argued in

Chapter 4, giving code requires a firm to be attuned to the relatively incremental needs of other community members. Giving code also requires a firm to focus on its own core competencies and current capabilities, behaviors that several researchers found to be associated with incremental innovations (1992); conversations in the guise of "giving help" between firms and other community members are free to explore more radical ideas that depart from the more incremental norms of giving code.

Finally, Hypotheses 5a and 5b argued that giving would negatively moderate taking, resulting in both lower development and debug time, respectively. Neither giving code nor giving help was shown to significantly moderate taking in a way that led to a statistical decrease in development time. However, if we look at Model 2 (Table 14), we see that there was a marginally significant ($p < .1$) negative direct effect of taking on development time, which is consistent with the finding in H1 that taking led to an increase in software creation productivity.

Considering debug time, there was no significant effect of the interaction of taking and giving code, but the interaction of taking and giving may help lead to longer debug times. Here, I believe it is again important to contemplate the difference between giving help and giving code. When code is exchanged, it can readily be used to replace older code and reduce or fix bugs. However, giving help does not necessarily have the same effect. Giving help through providing advice and assistance, albeit well intentioned, still requires the recipient to perform the work of modifying software. This takes time, which explains how giving help could lead to longer debug times.

RESEARCH IMPLICATIONS

This dissertation contributes to the debate within strategic management concerning how firms can leverage external sources of innovation, specifically the growing body of work exploring the influence of external communities. At the outset, it is important to acknowledge that firm interactions with an open source community are not necessarily positive or negative. My inductive exploration suggested that these behaviors can be inspired by the idiosyncratic perspectives of founders who are drawn to the egalitarian and meritocratic norms of community participation. It is also important, however, to acknowledge the growing role of communities in both management research and practice (O Mahony & Lakhani, 2011). This dissertation was undertaken to uncover the consequences of interactions between firms and one increasingly popular type of community, those developing open source software. I found that different combinations of taking, giving code, and giving help have real consequences for firms.

Theoretically, I sought to contribute to work within the open innovation perspective (Chesbrough, 2003; Chesbrough & Appleyard, 2007; Chesbrough et al., 2006b). As described in the first chapter, the open innovation perspective typically considers three types of behaviors: the most-researched "inbound" process whereby firms take innovative ideas from the external environment, the "outbound" process in which firms divest innovations to the external environment, and the least-researched "coupled" process in which firms simultaneously import innovations from the external community and divest internally developed innovations into that environment. This work contributes most to research on the "outbound" and "coupled" processes. Specifically, the finding

that firms experience different effects from giving code and giving help contributes an important nuance to the outbound process.

The concept of interactive engagement, in which firms both give and take, is analogous to the "coupled" process in open innovation. Results suggested that in some cases, the interaction of these two behaviors have an effect that is above and beyond the direct effects of either giving or taking in isolation, specifically for innovation incrementalism and debug time. This dissertation also provides a step in developing a scale useful for measuring exchange relationships between firms and a community. While this dissertation used firms that participate in an open source software community to develop this scale, there is no reason it may not be adapted for measuring relationships with other types of communities.

MANAGERIAL IMPLICATIONS

This work also provides managerial implications, particularly for executives considering how to best manage a relationship with an open source community. Specifically, this work demonstrates that giving code and giving help lead to different results. This research suggests that managers should consider what mix of taking, giving code, and giving help is best for their firm. Premeditated and purposeful community engagement may become a vehicle that firms may use to achieve their own strategic objectives.

Overall, this research adds evidence to the argument, popular in the open innovation perspective, that both the "outside in" and "inside out" approaches to engaging the external environment have consequences. The burgeoning use of external communities is ample evidence that innovation can indeed come from external sources.

LIMITATIONS

This dissertation has several limitations. First, both the inductive pilot and quantitative studies were conducted on firms that work with a single open source community. As a result, one cannot readily assume that these results would apply to firms working with a vastly different type of community, such as a trade association or brand community. Moreover, Drupal stands out from other open source communities in at least two ways: the community has been surging in popularity, and the leadership is demonstrably in favor of business engagement. Both facts may confound efforts to generalize these findings to communities that are declining in popularity or have different political dynamics. By examining multiple communities, future research may be able to identify community attributes (e.g., popularity, growth, leadership structure) that affect results among member firms.

A second limitation is the exclusive use of survey methods to collect data for statistical analysis, which could lead to common method bias. This limitation was difficult to avoid because the overwhelming majority of participating firms were private, making secondary data sources difficult to find or verify. As described in Chapter 4, efforts were made to construct the survey instrument in ways that minimized common method bias (Podsakoff et al., 2003). Future research may better address this problem by identifying creative ways to obtain additional data from the community itself or from other secondary sources.

Finally, I used a cross-sectional research design, which limits my findings to associations and prevents one from assuming causality. Future studies can address this limitation by using a longitudinal design.

OPPORTUNITIES FOR FUTURE RESEARCH

The work presented in this dissertation represents the start of a larger research program that will both address many of the limitations described above and extend various components of this dissertation. Below I describe three research projects this work has inspired.

First, I plan to extend the pilot study described in Chapter 3 into a comprehensive qualitative study of firms that work with the Drupal community. Respondents from 102 organizations indicated they would be willing to participate in subsequent interviews, which provides a strong base of firms for such a study. Through such an inductive study, I hope to both develop a greater understanding of the processes behind taking, giving code, and giving help, as well as learn more about minor themes that emerged in the pilot study, including the importance of community health to individual firms, the widespread embrace of collaboration with competitors, and the important leadership role played by the community's founder.

Second, plans are currently underway to extend this research into a multiyear longitudinal study of firms that work with Drupal. This dissertation has spawned what I hope will be a long-term partnership with the Drupal Association. Such a partnership makes the collection of longitudinal data possible, and current plans call for the 2013 Drupal Business Survey to be launched in May of 2013 at the community's conference in Portland, Oregon. The collection of additional data will allow me to perform additional analysis on a larger sample that includes the data in the survey used for this dissertation. Most importantly, it will allow me to investigate causal relationships between my independent variables (taking, giving help, and giving code) and the dependent variables.

I also plan to use this expanded data set to perform other statistical analyses, including but not limited to structural equation modeling.

The 2013 and future surveys provide an opportunity to improve measures used in this dissertation. For example, in 2012, in an effort to encouraging participants to complete the survey, I allowed respondents to skip some survey items. While this decision may have resulted in a larger overall number of responses, several survey items had relatively lower response rates than others. The 2013 survey will be designed to encourage respondents to complete more items. Additionally, I will introduce measures to identify the geographic location of responding firms, and refine my measure of software creation productivity.

Third, to extend the generalizability of these findings, I plan to compare firms that work with Drupal to those that work with other open source communities. Doing so would allow me to identify features of the open source community that may promote or hinder firm involvement. As described previously, Drupal is a popular community that has enjoyed tremendous growth in recent years. Comparing the experience of firms that work with Drupal to those working with less popular or slower growing communities will shed light on the significance of community attributes.

CONCLUSION

The research question investigated in this dissertation, "how does variation in the type and intensity of participation in an external open source software community affect outcomes related to firm-level innovation?" is related to a number of important questions within strategic management, including whether innovations are best developed internally

or in concert with external environments, and the growing role of communities. This dissertation contributes both inductive and deductive findings to these debates. Improved understanding of community–firm interactions holds the promise of helping managers and scholars alike understand how firms may best work with these organizational forms.

Appendix A: Tables and Figures

Table 1: Comparison of Traditional Firms and Open Source Communities

	Traditional Firms	**Open Source Communities**
Membership	Selective throughout	Open to anyone
Membership agreements	Formal and contractual	Informal and egalitarian
Critical stakeholders	Owners (Board of Directors, Stockholders)	Members
Legal Protection of Intellectual Property	Common, often imperative	Mutual support of licensing agreements; often difficult to enforce
Dissolution mechanisms	Bankruptcy or sale	Fading into obscurity; Forking

Table 2: Hypotheses

#	**IV**	**DV**	**Hypotheses**
H1	Taking	Firm Productivity	Taking from an open-source community will be positively related to the productivity of software development.
H2	Giving	Social Ties	Giving to a community will be positively related to the strength of social ties between a firm and an open source community.
H3	Interactive Engagement (Giving code x Taking code)	Number of New Products	The relationship between taking from an open-source community and the number of new products will be positively moderated by giving to the open source community, such the number of new products will be higher when giving is high than when giving is low.
H4	Interactive Engagement (Giving code x Taking code)	Innovation Incrementalism	The relationship between taking from an open-source community and innovation incrementalism will be positively moderated by giving to the open source community, such that innovation incrementalism will be higher when giving is high than when giving is low.
H5	Interactive Engagement (Giving code x Taking code)	Development and Debug Time	5a. The relationship between taking from an open-source community and product development time will be negatively moderated by giving to the open source community, such that product development speed time will be lower when giving is high than when giving is low. Hypothesis 5b. The relationship between taking from an open-source community and product development time will be negatively moderated by giving to the open source community, such that product debug time will be lower when giving is high than when giving is low.

Table 3: Summary of Reviewed Empirical Literature on Communities in Management

Authors (year)	**Type of Community**	**Conclusion / Main Finding**
Algesheimer et al. (2005)	Brand Community	Identification with a community leads to greater community engagement, but also normative community pressure.
Bertels et al. (2011)	Communities of practice	Communities moderate the relationship between dispersed collaboration and innovation performance
Bonaccorsi et al. (2006)	Open source	Communities make possible the use of hybrid business models, enabling firms to save money and risk
Dahlander (2007)	Open Source	De novo, or newer firms, developed new business models to work with pre-existing open source communities
Dahlander and Wallin (2006)	Open source	Communities can serve as a "complementary assets," to develop products and services, but requires that that firms participate, interact and learn
Dahlander and Magnusson (2005)	Open source	Communities force firms to consider their relationship with the community
Dahlander and Magnusson (2008)	Open source	Communities force firms to consider their relationship with the community
Dokko and Rosenkopf (2010)	Standard-setting	Communities provide access to individuals with social capital; a means to coordinate standards
Faraj and Johnson (2011)	Standard-setting	Communities facilitate discussion and dialogue between the geographically dispersed
Fjeldstad et al. (2012)	Varies	Communities provide opportunities for participants, but require them to change longstanding attitudes.
Fleming and Waguespack (2007)	Trade Association	Communities provide a forum for firms to learn about (and influence) standards
Fosfuri et al. (2008)	Open source	Communities provide firms with complementary resources
Franke et al. (2008)	User Community	Providing user communities with innovation tools results in improved idea development and evaluation.
Franke and Shah (2003)	User Community	Innovation-related ideas are freely shared among participants in a user community.
Fuller et al. (2008)	Brand Community	Communities and their members can be a source of firm innovation; firms must consider how best to nurture brand community members
Garg et al (2011)	Online	Communities facilitate diffusion of information among members

	Communities	
Garud et al. (2002)	Trade Association	Communities promote adoption of a new technological innovation.
Grewal et al. (2006)	Open source	Communities force firms to consider the relationships of their employees (embeddedness) with the community
Haefliger et al. (2008)	Open source	Communities provide firms with software that can be used to lower search costs
Heere et al .(2011)	Brand Community	Community identification affects identification with other communities, and consumer behavior
Henkel (2006)	Open source	Communities help set standards, increase demand for complementary products and services, provide external development support (955)
Henkel (2008)	Open source	Communities provide development support, standard-setting, legitimacy; force firms to consider downsides
Hienerth and Lettl, 2011	User community	Communities support entrepreneurship by providing feedback, development contributions, testing, and promotion
Hienerth (2006)	Open source	Communities help set standards, increase demand for complementary products and services, provide external development support (955)
Jeppesen and Frederiksen (2006)	User community	Communities provide firms access to lead users and their ideas
Jeppesen and Lakhani (2010)	User community	Communities enable firms to find innovative ideas to complex problems
Lee and Cole (2003)	Open source	Communities may allow firms to extend product and knowledge creation beyond the limits of their own resources.
MacCormack et al. (2006)	Open source	Communities influence the design architecture (i.e. modularity) of products; highlight the importance of managerial choice (and co-location) in such decisions.
Mahr and Lievens (2012)	Virtual communities	Communities allow firms to affordably reach a large number of lead users
McAlexander et al. (2002)	Brand Community	Communities provide ways for the firm to understand and better serve the needs of customers
Muniz and O'Guinn (2001)	Brand Community	Communities provide ways for the firm to understand and better serve the needs of customers
O'Mahony (2003)	Open source	Community participation suggests firms should consider the legal and normative ways these groups protect their work
O'Mahony and Bechky (2008)	Open source	Communities provide a forum for firms to work with each other
O'Mahony and Ferraro	Open source	Implied: Communities have governance processes that firms should understand

(2007)		
Piva et al. (2012)	Open source	Communities boost firms' innovation performance
Porter and Donthu (2008)	Virtual communities	Communities provide a forum for engaging customers
Rosenkopf et al. (2001)	Trade Association	Communities strengthen relationships between member firms; helps less-established firms to form alliances
Schau et al. (2009)	Brand Community	Communities can be a source of value for firms by providing product understanding, skills, and emotional commitments
Shah (2006)	Open source	Communities offer assistance and means to increase adoptions (cites West '03); requires firms to consider property rights ("code control")
Snow et al. (2011)	Brand Community	Communities provide a means for "collective development and sharing of knowledge" and allowing firms to overcome technological and coordination problems.
Stam (2009)	Open source	Communities boost firms' performance in a curvilinear fashion related to participation; require firms to address new strategic issues
von Krogh et al. (2003)	Open source	Communities have norms by which individuals become contributors; firms should understand these processes
Wade (1995)	Trade Association	Communities provide access to other firms who may provide support and complements
Wade (1996)	Trade Association	Communities support agreement on technological design; social ties among members are important
Waguespack and Fleming (2009)	Standards-setting	Firms that attended community functions had a higher likelihood of an equity event.
Wang and Ramiller (2009)	Trade Association	Communities provide learning benefits
Wasko and Faraj (2005)	Trade Association	Communities encourage individuals to contribute when there are reputational benefits, have something to share, or are embedded.
West (2003)	Open source	Communities provide a vehicle for firms to develop complementary assets

Table 4:
Description of Firms Studied

Firm Pseudonym	**Title of Interviewee(s)**	**Founding Year**	**Size (Employees)**	**Market**
AcademicWeb	Co-Founders, Co-CEOs	1996	22	Website creation for higher education
MediaNet	Founder & CEO	2001	14	Multi-media website creation
HelpingWeb*	Founder & CEO	2006	17	Website development and community support
NewsShare	Founder & CEO	2004		Social media platform
CoachX	CEO	2007	85	Infrastructure software and community support

*2 Interviews

Table 5:
Taking from the Community

Behavior	**Motive**	**Anticipated Outcomes**	**Illustrative Quotes**
Taking Code	Convenience	Shortened development time Simplified development	"We were able to do so many different things with Drupal in such a short period of time simply by pulling together a bunch of contributed modules, customizing the interface. " "We've got these big building blocks, these modules… what do you need? Do you need friend lists? Do you need Facebook integration? Do you need rating systems?"
Taking Ideas	Learning	Shortened development time Creative problem solving	"…as we started to go to local events… you'd be sitting in this room with hundreds of other people who were all being equipped and trained by these top notch companies, and they're giving away company secrets constantly. They're showing you, 'OK here we just did Spin Magazine segment, here's how we themed it, and here's how you guys can do better on your theming.' And you're sitting there going, wow who does this? This is awesome. I love this."
Taking Help	Support	Shortened debug time	"… there is value to peer code review, especially when you're a small shop. Because the community can give you resources that you don't have… in terms of development."

Table 6:
Giving to the Community

Behavior	**Key Motive**	**Anticipated Outcomes**	**Illustrative Quotes**
Giving Code	Improving the Community's Product	Product Improvement Reputation Benefits	"…we're the largest entity in the Drupal community probably. We're an active, really active part of it. I don't even know that -- we have probably 12 or 13 committers." "I would say [we give back] 90%, 95%... of the stuff we build." "… there's thousands of people who develop on Drupal, but I would say there's maybe a couple of hundred people that have a good reputation and are looked at as strong contributors." "Drupal 7 will … 732 people [that] have code that actually got into Drupal Core. That's 732 people. There are probably 4 or 5 times that many people who contributed code…"
Giving Help	Friendship	Reputation Benefits	"It's very common in the Drupal community for like, say, a bunch of developers sitting around in a discussion channel, you're just shooting the breeze a lot, talking about bugs you're finding you're trying to plow through…" "…we build a site, and we would write the technical write up and share it with everybody on Drupal.org. Everybody saw it." "….we write a blog post, we write case study for our website, we write a case study for proposals,… we have all different kinds of audiences so that... solidifies our relationship within the community… it gives back. Tells them how we build things. What modules we use."
Supporting the Community	Strengthening Relationships among Community Members	Friendship Product Improvements	"We've done dozens, if not like hundreds presentations at DrupalCon over the years as a company" "…people expected to do the above and beyond. So I began attending the local Drupal Camp on a monthly basis. When the reins were passed on from the Drupal Camp in Vancouver, our team got involved in starting to organize that, so the next one coming up is the June summit. So helping organize the events, doing training at the events, presentations… Those are all above and beyond types of things."

Table 7:
"Interactive Engagement:" The interaction of high levels of *Taking from* and *Giving to* the Community

Outcome	**Illustrative Quotes**
Reciprocity	"Other shops that we work with rely on us and vice versa. There's just implicitly a lot of deep networking that goes on there because we're all relying on things that each of us is working on."
Referrals	"… we fail without the community...without an ecosystem successfully implementing what we were doing we wouldn't get any customers because we wouldn't have success stories to talk about. We would have people out evangelizing what were doing because they made a living off of it as well." "Drupal has been fantastic for us, in terms of a tool that we use to implement and the work that we've done in the community certainly does help with the generation. We get unsolicited RFPs literally every day."
Recruitment	"From a talent recruitment standpoint… you want to be on the cutting edge, from a driving the platform where you want to go."
Community growth	"No single company, no matter how much money you have, no matter what resources you have, can justify building out things like quality assurance frameworks and load testing frameworks, and dealing with performance and scalability problems. These are the perfect kinds of things to work on collaboratively. They're not... your business. And, you can never do them justice on your own." "Our strength is in numbers, and you want more people using the platform. " "…there are parts of the community [inaudible; 0:10:37.8], that they feel -- and you've met some of the guys. So, I don't know what they said to you, but they're nervous are we going to squash them. You know? And, I think in most cases it's naiveté about how a market grows, and the bigger pie is good for everybody."

Table 8:
Final Factor Analysis Loadings (Pattern Matrix)

Questionnaire Item	Factor			
	1 "Duration of Engagement"*	2 "Taking"	3 "Giving Code"	4 "Giving Help"
Years using Drupal	**.94**	-.04	-.06	-.02
Years Seeking Assistance	**.95**	.02	-.04	-.04
Years Contribution	**.75**	-.02	.12	.12
Use Drupal Since Founding	-.17	**.63**	-.06	.04
Use of Drupal in Products (%)	.18	**.75**	.02	-.05
Employees using Drupal (%)	-.01	**.94**	.01	-.06
Contributions to Drupal Core	.02	-.07	**.81**	.04
Contributions to Drupal Docs	-.01	.12	**.64**	.10
Contributions to Drupal Themes	-.04	-.04	**.99**	-.10
Encouragement of Volunteering	-.03	.16	-.07	**.83**
Payment of Volunteers	.04	-.17	.00	**.83**
Assisting others for Free	.00	.19	.12	**.50**

Extraction Method: Principal Axis Factoring.

Rotation Method: Promax with Kaiser Normalization.

* "Duration of Engagement" was dropped from subsequent analysis because it did not represent a theoretically relevant aspect of interaction between firms and the open source community.

Table 9:
Descriptive Statistics and Correlations of Key Variables

	Mean	S. D.	1	2	3	4	5	6	7
1. Taking	.02	2.51	1						
2. Giving Code	-.04	2.58	.267**	1					
3. Giving Help	-.01	2.55	.504**	.481**	1				
4. Interactive Engagement (Taking x Giving Code)	1.72	6.47	.042	.539**	.176*	1			
5. Interactive Engagement (Taking x Giving Help)	3.20	5.90	-.072	.194**	.029	.493**	1		
6. Productivity (log)	-9.51	2.04	.288**	-.152	.019	-.043	-.093	1	
7. Social Ties	0.00	1.00	.329**	.556**	.418**	.545**	.240**	-.080	1
8. Innovation Incrementalism	0.00	1.00	.093	.101	.032	.070	-.126	-.082	.032
9. Number of New Products	.02	3.03	-.204*	.028	-.161*	-.084	.114	-.170	.009
10. Development Time	0.00	1.00	-.167	.303**	.122	.122	.134	-.711**	.218**
11. Debug Time	3.00	.71	-.125	-.077	-.113	-.134	.071	.005	-.121
12. Strategic Posture	-.00	1.63	-.024	.102	.166*	.139	-.073	-.241**	.150
13. Firm Size	2.93	2.70	-.728**	-.025	-.319**	.125	.140*	-.333**	-.071
14. Performance	0.00	1.00	-.304**	.143	.008	.268**	.148	-.425**	.177*
15. Firm Growth	22.64	49.00	.108	.179*	.131	.070	.123	-.127	.184*
16. Founder is CEO	1.71	.45	.547**	.154*	.323**	-.031	-.048	.124	.171*
17. Firm Age (log)	2.26	1.35	-.724**	-.142*	-.307**	.076	.035	-.178	-.116

	8	9	10	11	12	13	14	15	16
9. Number of New Products	-.033								
10. Development Time	.006	.222*	1						
11. Debug Time	-.025	.122	.030	1					
12. Strategic Posture	.058	.033	.241**	.106	1				
13. Firm Size (log)	-.020	.328**	.294**	.236**	.033	1			
14. Performance	-.071	.115	.319**	-.122	.426**	.399**	1		
15. Firm Growth	.042	-.050	.071	-.008	.110	-.023	.186*	1	
16. Founder is CEO	.058	-.243**	-.057	-.169*	.043	-.671**	-.204**	.103	1
17. Firm Age (log)	-.061	.204*	.181*	.124	-.045	.691**	.204*	-.231**	-.618**

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).

Table 10:
Hypothesis 1 Regression Results
Dependent Variable: Software Productivity

	Model 1 Controls	**Model 2** Direct Effects
Strategic Posture	-.239** (.077)	-.192* (.077)
Firm Size	-.522** (.096)	-.410** (.100)
Performance	-.348 * (.157)	-.332* (.151)
Growth	.001 (.003)	.002 (.003)
Founder is CEO	-.590 (.402)	-.391 (.404)
Firm Age	-.074 (.147)	.111 (.169)
Taking		.204* (.094)
Giving Code		-.114* (.045)
Giving Help		-.039 (.061)
Constant	-7.364** (.866)	-8.436** (.921)
ΔR^2		.04*

** p<0.01, * p<0.05, + p<0.1 [N = 91; Standard errors are shown in parentheses.]

Table 11:
Hypothesis 2 Regression Results
Dependent Variable: Social Ties

	Model 1 Controls	**Model 2** Direct Effects
Strategic Posture	-.022 (.042)	-.010 (.077)
Firm Size	-.022 (.043)	.032 (.041)
Performance	-.026 (.043)	.152* (.067)
Growth	.001 (.001)	.000 (.001)
Founder is CEO	-.344+ (.194)	.178 (.173)
Firm Age	-.036 (.077)	.094 (.073)
Taking		.128* (.042)
Giving Code		.115** (.045)
Giving Help		.023 (.027)
Constant	-.743+ (.423)	-.708+ (.398)
ΔR^2		.259**

** p<0.01, * p<0.05, + p<0.1 [N = 113; Standard errors are shown in parentheses.]

Table 12:
Hypothesis 3 Regression Results
Dependent Variable: Number of New Products

	Model 1 Controls	**Model 2** Direct Effects	**Model 3** Interactions
Strategic Posture	.012 (.064)	.034 (.065)	.049 (.067)
Firm Size	.188 * (.066)	.224 * (.065)	.219 * (.074)
Performance	.015 (.122)	.025 (.122)	.008 (.126)
Growth	-.002 (.003)	-.001 (.003)	-.001 (.003)
Founder is CEO	-.080 (.291)	.019 (.298)	.009 (.303)
Firm Age	-.123 (.113)	-.026 (.133)	-.017 (.135)
Taking		.104 (.077)	.105 (.078)
Giving Code		-.043 (.039)	-.041 (.052)
Giving Help		-.047 (.047)	-.054 (.048)
Taking x Giving Code			-.005 (.024)
Taking x Giving Help			.019 (.019)
Constant	-.294 (.641)	-.792 (.707)	-.815 (.719)
ΔR^2		.028	.008

** p<0.01, * p<0.05, + p<0.1 [N = 109; Standard errors are shown in parentheses.]

Table 13:
Hypothesis 4 Regression Results
Dependent Variable: Innovation Incrementalism

	Model 1 Controls	**Model 2** Direct Effects	**Model 3** Interactions
Strategic Posture	.051 (.061)	.056 (.060)	.020 (.060)
Firm Size	.156* (.066)	.205* (.072)	.210** (.070)
Performance	-.128 (.116)	-.123 (.114)	-.129 (.113)
Growth	.002 (.002)	.001 (.002)	.002 (.002)
Founder is CEO	.652* (.282)	.640* (.288)	.669* (.281)
Firm Age	-.119 (.112)	-.040 (.121)	-.069 (.119)
Taking		.132+ (.069)	.107 (.068)
Giving Code		.068+ (.037)	.035 (.039)
Giving Help		-.050 (.046)	-.032 (.045)
Taking x Giving Code			.043* (.018)
Taking x Giving Help			-.041* (.018)
Constant	-1.273* (.624)	-1.56* (.677)	-1.50* (.660)
ΔR^2		.066*	.059*

** p<0.01, * p<0.05, + p<0.1 [N = 120; Standard errors are shown in parentheses.]

Table 14:
Hypothesis 5a Regression Results
Dependent Variable: Development Time

	Model 1 Controls	**Model 2** Direct Effects	**Model 3** Interactions
Strategic Posture	.092 (.061)	.037 (.058)	.035 (.061)
Firm Size	.190** (.066)	.117+ (.067)	.123_ (.068)
Performance	.183 (.114)	.166 (.107)	.191+ (.112)
Growth	-.001 (.002)	-.003 (.002)	-.004 (.002)
Founder is CEO	.406 (.304)	.080 (.299)	.109 (.306)
Firm Age	.042 (.109)	-.024 (.114)	.003 (.118)
Taking		-.114+ (.063)	-.097 (.066)
Giving Code		.093** (.034)	.111** (.039)
Giving Help		.099* (.046)	.095+ (.048)
Taking x Giving Code			-.017 (.018)
Taking x Giving Help			.001 (.018)
Constant	-1.191+ (.654)	-.272 (.661)	-.367 (.683)
ΔR^2		.115**	.007

** p<0.01, * p<0.05, + p<0.1 [N = 94; Standard errors are shown in parentheses.]

Table 15:
Hypothesis 5b Regression Results
Dependent Variable: Debug Time

	Model 1 Controls	**Model 2** Direct Effects	**Model 3** Interactions
Strategic Posture	.141**	.139**	.160**
	(.040)	(.041)	(.042)
Firm Size	.125**	.119*	.109*
	(.044)	(.047)	(.047)
Performance	-.344**	-.342**	-.351**
	(.072)	(.073)	(.074)
Growth	.003*	.003*	.003+
	(.002)	(.002)	(.002)
Founder is CEO	-.115	-.115	-.138
	(.200)	(.213)	(.213)
Firm Age	-.118	-.130	-.106
	(.072)	(.087)	(.086)
Taking		-.022	-.014
		(.050)	(.049)
Giving Code		-.018	-.006
		(.025)	(.031)
Giving Help		.020	.012
		(.031)	(.031)
Taking x Giving Code			-.012
			(.015)
Taking x Giving Help			.029*
			(.012)
Constant	3.086**	3.129**	3.099**
	(.425)	(.482)	(.478)
ΔR^2		.009	.046+

** p<0.01, * p<0.05, + p<0.1 [N = 99; Standard errors are shown in parentheses.]

Table 16:
Hypotheses Test Results

	Taking x GivingCode	***Taking x GivingHelp***
H1 (Taking → Productivity)	Supported	
H2 (Giving → Social Ties)	Supported (Giving Code)	Not Supported (Giving Help)
H3 (Engagement → # New Products)	Not Supported	Not Supported
H4 (Engagement →Incrementalism)	Supported	Not Supported^
H5a (Engagement → Lower Development Time)	Not Supported	Not Supported
H5b (Engagement → Lower Debug Time)	Not Supported	Not Supported^

^ Significant in the opposite direction

Figure 1: Heuristic of Firm Engagement with an Open source Community

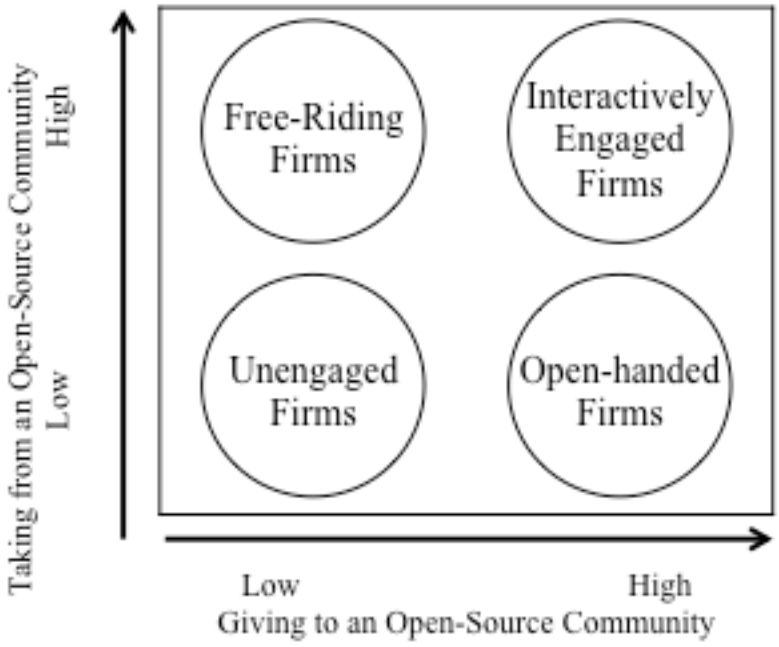


Figure 2: Theoretical Model

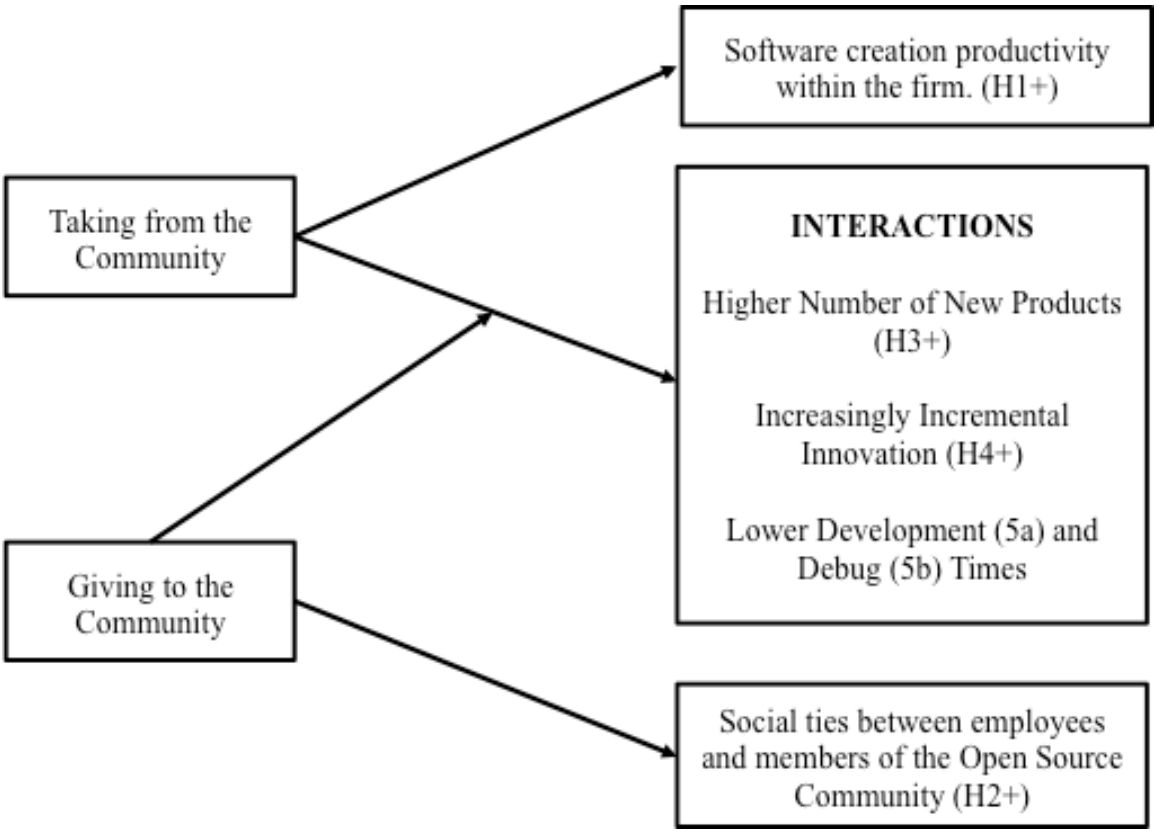
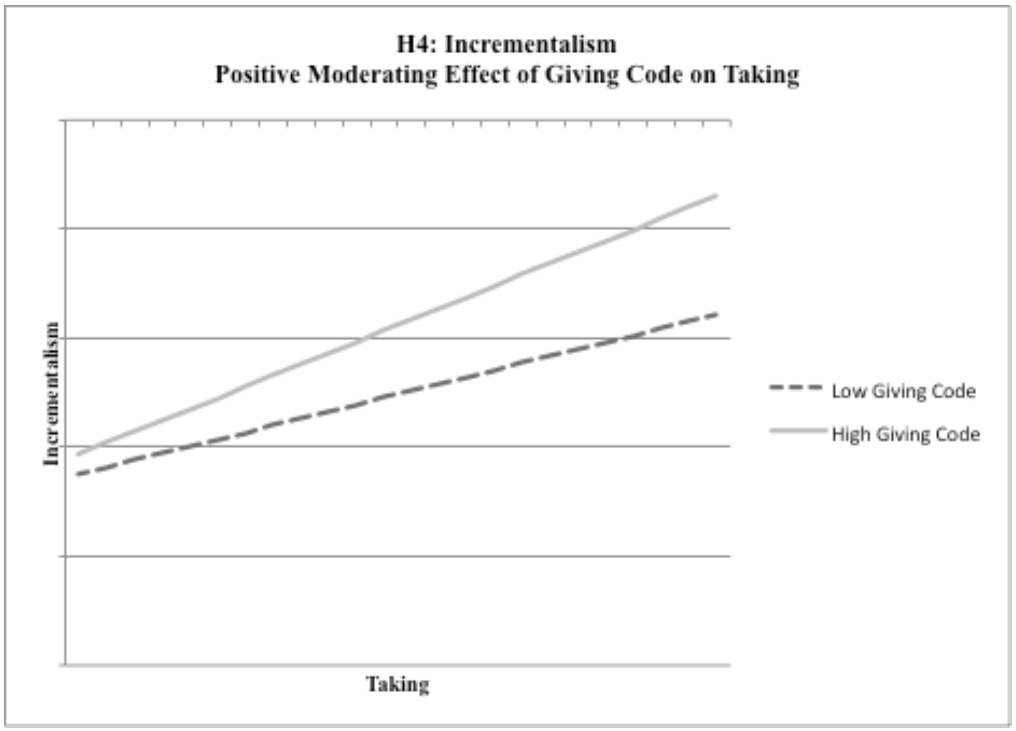


Figure 3: Graphs of Statistically Significant Interaction Effects in Hypothesis 4



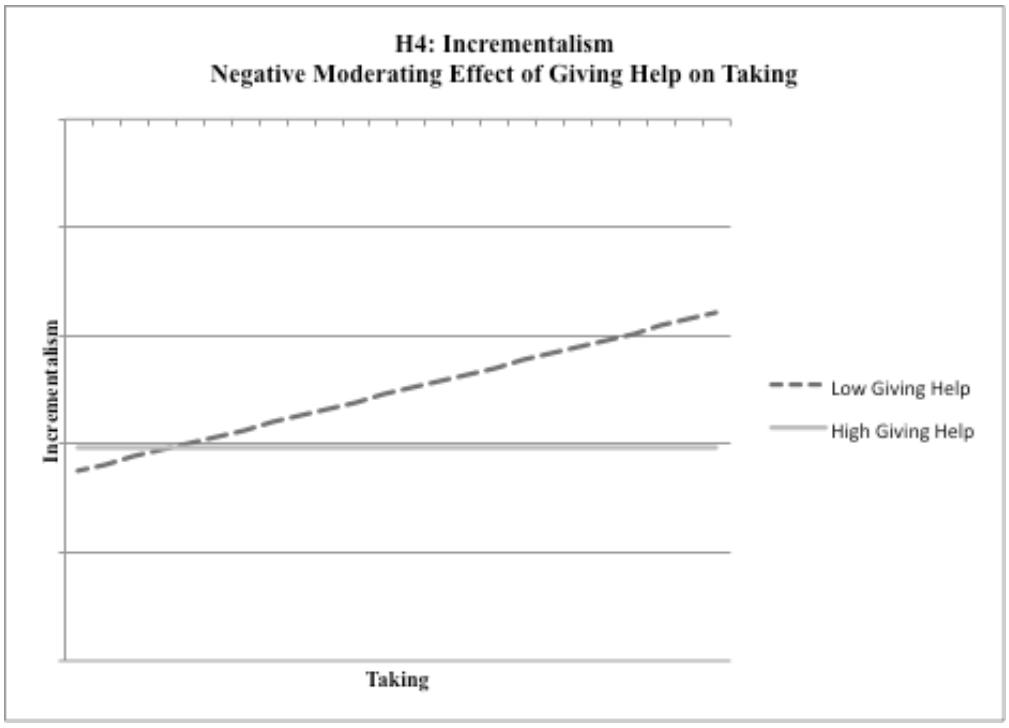
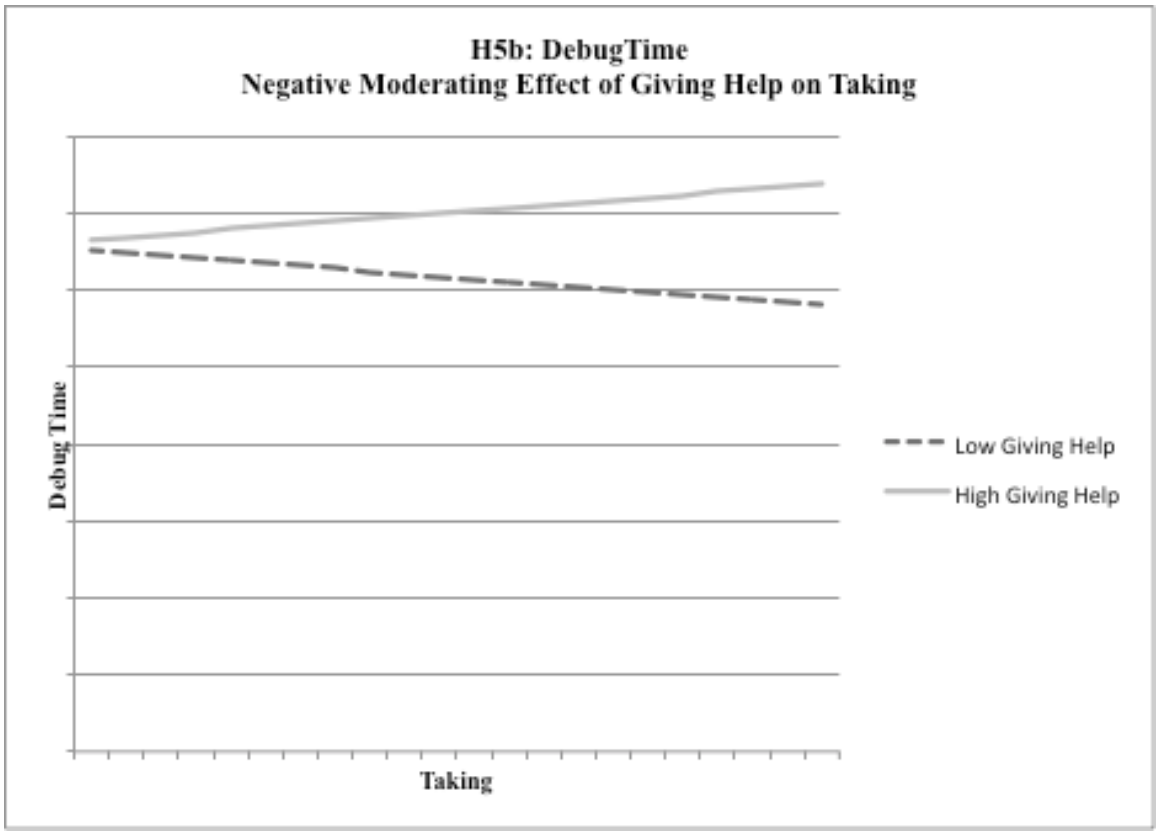


Figure 4: Graph of Statistically Significant Interaction Effect in Hypothesis 5b



Appendix B: 2012 Drupal Business Survey

(This page intentionally left blank. The survey begins on the next page.)

McCOMBS SCHOOL OF BUSINESS

Drupal™ Association
2012 BUSINESS SURVEY

I invite you to participate in a questionnaire, entitled "**Private Firm Engagement in the Drupal Open Source Software Community**." The study is being conducted in cooperation with the Drupal Association by:

Jonathan Sims
PhD Candidate, Management Department
McCombs School of Business
The University of Texas at Austin
1 University Station Austin, TX 78712
jonathan.sims@phd.mccombs.utexas.edu

The questionnaire has been designed to help researchers as well as the Drupal Association better understand how private companies such as yours interact with the Drupal open source community.

A few introductory points:

- **We know you are busy, so this questionnaire is short.** It includes 37 items and should take approximately 10-15 minutes to complete.
- **Your responses are valuable to Drupal.** The Drupal Association will use these responses to better server our corporate community. This research is official sponsored by the Drupal Association.
- **Your specific answers will be confidential.** Only analysis performed on aggregated results will be shared with the Drupal Association and may be used in subsequent research. After you complete the questions, you may request a copy of our findings.
- **Your identifying information (name, email, job title) will be kept secure**. Identifying records will be kept by Mr. Sims alone and will not be shared with any outside organization. Answers and individual Information shared with the Drupal Association are anonymous.
- **Your participation is voluntary**. You have the right to withdraw from participation at any time without penalty. If you wish to withdraw from the study or have any questions, contact Mr. Sims, listed above.

Questions about your rights as a research participant. *If you have questions about your rights or are dissatisfied at any time with any part of this study, you can contact, anonymously if you wish, the Institutional Review Board by phone at (512) 471-8871 or email at orsc@uts.cc.utexas.edu.*

By completing this survey you acknowledge that you are at least 18 years of age and have given consent to participate in this survey. If you agree to participate please select "Yes' below.

I have read the above information and have sufficient information to make a decision about participating in this study. I consent to participate in the study:

□ Yes □ No

First, a few short demographic questions.

(Again, your information will be kept confidential and made anonymous.)

Your Personal Information

First Name: _____ Last Name

_____

Email address:

_____

Job Title:

_____

Your Company Information

Company name: _____

Current number of employees: _____

Approximate number of employees last year: _____

What year was your company founded? _____

What does your company do? Please check all that apply.

□ Distribution of Open Source Core Software
□ Sales of Software Products
□ Consulting Services
□ Systems Implementation and Integration Services
□ Support Services
□ Maintenance Services
□ Remote Administration Services
□ Training Services
□ Web Development
□ Other, Please Specify:

_____

Is the founder of your company also the current CEO?

□ Yes □ No

Now, we'd like to learn about how your company receives support from the Drupal community.

How often does your company use software code from the Drupal open source community?

Never	Rarely	Sometimes	Quite Often	Very Often
□	□	□	□	□

For how many years has your company used code from the Drupal open source community? (Select one)

❍ Less than 1 Year
❍ 1 Year
❍ 2 Years
❍ 3 Years
❍ 4 Years
❍ 5 Years
❍ 6 Years
❍ 7 Years
❍ 8 Years
❍ 9 Years
❍ 10+ Years
❍ Not applicable

Has your company used Drupal since your company was founded?

❍ Yes ❍ No

How often does your company utilize each of the following Drupal support channels?

	Never	Rarely	Sometimes	Quite Often	Very Often
IRC	❍	❍	❍	❍	❍
Forums	❍	❍	❍	❍	❍
Meetups	❍	❍	❍	❍	❍
Other, please specify	❍	❍	❍	❍	❍

How many years has your company asked for assistance from the Drupal open source community?

- ❍ Less than 1 Year
- ❍ 1 Year
- ❍ 2 Years
- ❍ 3 Years
- ❍ 4 Years
- ❍ 5 Years
- ❍ 6 Years
- ❍ 7 Years
- ❍ 8 Years
- ❍ 9 Years
- ❍ 10+ Years
- ❍ Not applicable

What percentage of your current products or services are based on the Drupal open source community?

None	1-25%	26-50%	51-75%	76-100%
❍	❍	❍	❍	❍

Approximately how many employees of your company regularly interact with individuals in the Drupal open source community? _____

In addition to Drupal, approximately how many other open-source communities does your company work with?

- ❍ No others
- ❍ 1 additional community
- ❍ 2-5 additional communities
- ❍ 6 or more additional communities

Below, we'd like to learn about how your company contributes (or gives to) the Drupal community.

How often does your company contribute code to the Drupal open source community?

Never	Rarely	Sometimes	Quite Often	Very Often
❍	❍	❍	❍	❍

For how many years has your company contributed code to the Drupal open source community?

- ❍ Less than 1 Year
- ❍ 1 Year
- ❍ 2 Years
- ❍ 3 Years
- ❍ 4 Years
- ❍ 5 Years
- ❍ 6 Years
- ❍ 7 Years
- ❍ 8 Years
- ❍ 9 Years
- ❍ 10+ Years
- ❍ Not applicable

What percentage of your company's contributions to Drupal have been accepted and incorporated into each of the following Drupal products?

	None	1-25%	26-50%	51-75%	76-100%
Drupal Core	❍	❍	❍	❍	❍
Modules	❍	❍	❍	❍	❍
Documentation	❍	❍	❍	❍	❍
Themes	❍	❍	❍	❍	❍

How often does you company encourage employees to volunteer time to the Drupal open source communities?

Never	Rarely	Sometimes	Quite Often	Very Often
❍	❍	❍	❍	❍

How often does your company pay employees for the time spent working on contributions to the Drupal community (e.g. core, modules, documentation, themes)?

Never	Rarely	Sometimes	Quite Often	Very Often
❍	❍	❍	❍	❍

How often does your company assist other companies in the Drupal open source community free of charge?

Never	Rarely	Sometimes	Quite Often	Very Often
❍	❍	❍	❍	❍

Below, we ask products and your relationship with Drupal.

Below, please provide an estimate of how many NEW products (or client jobs) your company began in each of the past three years.

2009	
2010	
2011	

For the five questions below, please rely on your own knowledge or readily available information about projects your company has completed (or is currently completing) in the last year.

In the last year, how many products (or client jobs) did your company complete?	
In the last year, what was the average total cost (including development and support) of a typical product or client job?	
How many hours (development time) does the typical project (or client job) take?	
What percentage of the time for each project or client job is spent engaged in debugging (i.e. not creating new code)?	
What percentage of the time for each project or client job is spent developing custom Drupal modules?	

For the two questions below, please reflect on your personal relationship with members of the Drupal community.

How many individuals OUTSIDE YOUR COMPANY but involved in the Drupal Community do you consider to be acquaintances or friends?	
How many of these people do you consider acquaintances but not close friends?	

Below, we ask about your company's technical field.

(A* technical field *refers to software development related to the creation and sales of new products or services.)

How rapid is the progress of technical knowledge within your field?

Very Slow						Very Rapid
❍	❍	❍	❍	❍	❍	❍

Broadly, how easy (or difficult) was it for your company to gain the necessary technical knowledge for the development of current products/services?

Very Easy						Very Difficult
❍	❍	❍	❍	❍	❍	❍

How easy (or difficult) was it for your company to get up to speed on Drupal? In other words, how easy or difficult was it to adjust to the learning curve for Drupal?

Very Easy						Very Difficult
❍	❍	❍	❍	❍	❍	❍

The questions below ask for subjective input about your company's performance compared to companies of similar sales volume in your industry over the past year. Please rely on your own knowledge or readily available information to answer.

Compared to other companies in my industry, my company's total sales growth is in the...

Bottom 20%	Bottom 40%	Middle 20%	Top 40%	Top 20%
❍	❍	❍	❍	❍

Compared to other companies in my industry, I would rate my company's overall performance (or success) in the...

Bottom 20%	Bottom 40%	Middle 20%	Top 40%	Top 20%
❍	❍	❍	❍	❍

In dealing with its competitors, my company…

...is very seldom the first company to introduce new products or services...			...or...			...is very often the first business to introduce new products or services.
❍	❍	❍	❍	❍	❍	❍

In general, the top managers of my company have…

... strong preference for low-risk projects (where success is more certain)...			...or...			...a strong preference for high-risk projects (where success is less certain)
❍	❍	❍	❍	❍	❍	❍

The questions below ask for your input about the Drupal Association and Drupal.org.

How does your company use drupal.org for development and site building?

What does it mean to your business if drupal.org is improved for developers and site builders?

Would you contribute funds to a web development team to make drupal.org better for developers and site builders?

How much would you donate each year towards this effort?

- ❑ Less than $100
- ❑ $100 - $1,000
- ❑ $1,001 - $2,500
- ❑ $2,501 - $5,000
- ❑ Greater than $5,000

Thank you for your time!

Q65 Would like to receive a copy of the published survey results?

- ❍ Yes
- ❍ No

To thank you for your participation, we will email you a coupon that may be redeemed online for a FREE Limited-edition Drupal Association t-shirt (Normally sold for $50). Where should we email the t-shirt coupon?

- ❍ Email the t-shirt coupon to me at the email address I provided earlier.
- ❍ Email the coupon to the following email address:
  _____

References

Adams, R., Bessant, J., & Phelps, R. 2006. Innovation management measurement: A review. ***International Journal of Management Reviews***, 8: 1, 21-47.

Ahuja, G., & Katila, R. 2001. Technological acquisitions and the innovation performance of acquiring firms: A longitudinal study. ***Strategic Management Journal***, 22: 3, 197-220.

Ahuja, G., Lampert, C. M., & Tandon, V. 2008. Chapter 1: Moving beyond schumpeter: Management research on the determinants of technological innovation. ***Academy of Management Annals***, 2: 1, 1-98.

Allen, T. J. 1971. Communications, technology transfer, and the role of the technical gatekeeper. ***R&D Management***, 114-21.

Almirall, E., & Casadesus-Masanell, R. 2010. Open versus closed innovation: A model of discovery and divergence. ***Academy of Management Review***, 35: 1, 27-47.

Anand, B. N., & Khanna, T. 2000. Do firms learn to create value: The case of alliances. ***Strategic Management Journal***, 21: 3, 295.

Ancona, D. G., & Caldwell, D. F. 1992. Bridging the boundary: External activity and performance in organizational teams. ***Administrative Science Quarterly***, 37: 4, 634-665.

Arthur, W. B. 2001. ***How growth builds upon growth in high-technology***. Paper presented at the Annual Sir Charles Carter Lecture, Northern Ireland Economic Council, Belfast (United Kingdom).

Baldwin, C. Y., & Clark, K. B. 2000. ***Design rules***. Cambridge, Mass.: MIT Press.

Baldwin, C. Y., & Clark, K. B. 2006. The architecture of participation: Does code architecture mitigate free riding in the open source development model? ***Management Science***, 52: 7, 1116-1127.

Barnett, W. P., Mischke, G. A., & Ocasio, W. 2000. The evolution of collective strategies among organizations. ***Organization Studies***, 21: 2, 325-354.

Barney, J., Wright, M., & Ketchen Jr, D. J. 2001. The resource-based view of the firm: Ten years after 1991. ***Journal of Management***, 27: 6, 625.

Barney, J. B., & Hansen, M. H. 1994. Trustworthiness as a source of competitive advantage. ***Strategic Management Journal***, 15175-190.

Benner, M. J., & Tushman, M. L. 2003. Exploitation, exploration, and process management: The productivity dilemma revisited. ***Academy of Management Review***, 28: 2, 238-256.

Benson, D., & Ziedonis, R. H. 2009. Corporate venture capital as a window on new technologies: Implications for the performance of corporate investors when acquiring startups. ***Organization Science***, 20: 2, 329-351.

Blackburn, J. D., Scudder, G. D., & van Wassenhove, L. N. 1996. Improving speed and productivity of software development: A global survey of software developers. ***IEEE Transactions on Software Engineering***, 22: 12, 875-885.

Bollen, K., & Jackman, R. 1990. Regression diagnostics: An expository treatment of outliers and influential cases. In J. Fox, & J. S. Long (Eds.), ***Modern methods of data analysis***: 257-291. Newbury Park, CA: Sage.

Bonaccorsi, A., Giannangeli, S., & Rossi, C. 2006. Entry strategies under competing standards: Hybrid business models in the open source software industry. ***Management Science***, 52: 7, 1085-1098.

Bonaccorsi, A., & Rossi, C. 2003. Why open source software can succeed. ***Research Policy***, 32: 7, 1243.

Bonner, J. M., & Walker, O. C. 2004. Selecting influential business to business customers in new product development: Relational embeddedness and knowledge heterogeneity considerations. ***Journal of Product Innovation Management***, 21: 3, 155-169.

Bresman, H., Birkinshaw, J., & Nobel, R. 2010. Knowledge transfer in international acquisitions. ***Journal of International Business Studies***, 41: 1, 5-20.

Brooks, F. P. 1995. ***The mythical man-month: Essays on software engineering***. Reading, MA. : Addison-Sesley.

Brown, J. S., & Duguid, P. 1991. Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovating. ***Organization Science***, 2: 1, 40-57.

Brown, J. S., & Duguid, P. 2001. Knowledge and organization: A social-practice perspective. ***Organization Science***, 12: 2, 198-213.

Brown, S. L., & Eisenhardt, K. M. 1995. Product development: Past research, present findings, and future-directions. ***Academy of Management Review***, 20: 2, 343-378.

Chesbrough, H. W. 2003. ***Open innovation: The new imperative for creating and profiting from technology***. Boston, Mass.: Harvard Business School Press.

Chesbrough, H. W. 2006. ***Open business models : How to thrive in the new innovation landscape***. Boston, Mass.: Harvard Business School Press.

Chesbrough, H. W., & Appleyard, M. M. 2007. Open innovation and strategy. ***California Management Review***, 50: 1, 57-76.

Chesbrough, H. W., Vanhaverbeke, W., & West, J. 2006a. ***Open innovation: Researching a new paradigm***. Oxford: Oxford University Press.

Chesbrough, H. W., Vanhaverbeke, W., West, J., & ebrary Inc. 2006b. ***Open innovation researching a new paradigm***. http://ezproxy.lib.utexas.edu/login?url=http://site.ebrary.com/lib/utexas/Doc?id=10194775.

Coar, K. 2006. ***The open source definition (annotated)***. http://www.opensource.org/docs/definition.php.

Cohen, J., Cohen, P., & Stephen, G. 2003. ***Applied multiple regression/correlation analysis for the behavioral sciences***. Hillsdale, NJ: Erlbaum.

Cohen, W. M., & Levin, R. C. 1989. Moving beyond schumpeter: Management research on the determinants of technological innovation. In R. Schmalensee, & R. D. Willig (Eds.), ***Handbook of industrial organization***, Vol. 2: 1059-1107. Amsterdam ; New York.

Cohen, W. M., & Levinthal, D. A. 1990. Absorptive-capacity: A new perspective on learning and innovation. ***Administrative Science Quarterly***, 35: 1, 128-152.

Conner, K. R. 1991. A historical comparison of resource-based theory and 5 schools of thought within industrial-organization economics: Do we have a new theory of the firm? ***Journal of Management***, 17: 1, 121-154.

Covin, J. G., & Slevin, D. P. 1989. Strategic management of small firms in hostile and benign environments. ***Strategic Management Journal***, 10: 1, 75-87.

Covin, J. G., & Slevin, D. P. 1990. New venture strategic posture, structure, and performance: An industry life cycle analysis. ***Journal of Business Venturing***, 5: 2, 123.

Cropanzano, R., & Mitchell, M. S. 2005. Social exchange theory: An interdisciplinary review. ***Journal of Management***, 31: 6, 874-900.

Crossan, M. M., & Apaydin, M. 2010. A multi-dimensional framework of organizational innovation: A systematic review of the literature. ***Journal of Management Studies***, 47: 6, 1154-1191.

Dahlander, L. 2007. Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source software communities. ***Industrial & Corporate Change***, 16: 5, 913-943.

Dahlander, L., & Magnusson, M. G. 2005. Relationships between open source software companies and communities: Observations from nordic firms. ***Research Policy***, 34: 4, 481-493.

Dahlander, L., & Wallin, M. W. 2006. A man on the inside: Unlocking communities as complementary assets. ***Research Policy***, 35: 8, 1243-1259.

Damanpour, F. 1991. Organizational innovation: A meta-analysis of effects of determinants and moderators. ***Academy of Management Journal***, 34: 3, 555-590.

de Man, A.-P., & Duysters, G. 2005. Collaboration and innovation: A review of the effects of mergers, acquisitions and alliances on innovation. ***Technovation***, 25: 12, 1377-1387.

Demsetz, H. 1973. Industry structure, market rivalry, and public policy. ***Journal of Law and Economics***, 16: 1, 1-9.

Dess, G. G. 1987. Consensus on strategy formulation and organizational performance: Competitors in a fragmented industry. ***Strategic Management Journal***, 8: 3, 259-277.

Dewar, R. D., & Dutton, J. E. 1986. The adoption of radical and incremental innovations: An empirical analysis. ***Management Science***, 32: 11, 1422-1433.

Dierickx, I., & Cool, K. 1989. Asset stock accumulation and sustainability of competitive advantage. ***Management Science***, 35: 12, 1504-1511.

Dodgson, M., Gann, D. M., & Salter, A. J. 2002. The intensification of innovation. ***International Journal of Innovation Management***, 6: 1, 53.

Dosi, G. 1982. Technological paradigms and technological trajectories: A suggested interpretation of the determinants and directions of technical change. ***Research Policy***, 11: 3, 147-162.

Doty, D. H., & Glick, W. H. 1994. Typologies as a unique form of theory building: Toward improved understanding and modeling. ***Academy of Management Review***, 19: 2, 230-251.

Dyer, J. H., & Chu, W. J. 2003. The role of trustworthiness in reducing transaction costs and improving performance: Empirical evidence from the united states, japan, and korea. ***Organization Science***, 14: 1, 57-68.

Dyer, J. H., & Hatch, N. W. 2006. Relation-specific capabilities and barriers to knowledge transfers: Creating advantage through network relationships. ***Strategic Management Journal***, 27: 8, 701-719.

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.

Dyer, J. H., & Singh, H. 1998. The relational view: Cooperative strategy and sources of interorganizational competitive advantage. ***Academy of Management Review***, 23: 4, 660-679.

Edge, J. 2009. ***Sam ramji: On the codeplex foundation and more***. http://lwn.net/Articles/357339/.

Eisenhardt, K. M. 1989. Building theories from case-study research. ***Academy of Management Review***, 14: 4, 532-550.

Eisenhardt, K. M., & Graebner, M. E. 2007. Theory building from cases: Opportunities and challenges. ***Academy of Management Journal***, 50: 1, 25-32.

Eisenhardt, K. M., & Martin, J. A. 2000. Dynamic capabilities: What are they? ***Strategic Management Journal***, 21: 10-11, 1105-1121.

Eisenhardt, K. M., & Tabrizi, B. N. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. ***Administrative Science Quarterly***, 40: 1, 84-110.

Enkel, E., Gassmann, O., & Chesbrough, H. 2009. Open r&d and open innovation: Exploring the phenomenon. ***R&D Management***, 39: 4, 311-316.

Ethiraj, S. K., & Levinthal, D. A. 2004. Modularity and innovation in complex systems. ***Management Science***, 50: 2, 159-173.

Ettlie, J. E., Bridges, W. P., & O'Keefe, R. D. 1984. Organization strategy and structural differences for radical versus incremental innovation. ***Management Science***, 30: 6, 682-695.

Federman, M. 2006. The penguinist discourse: A critical application of open source software project management to organization development. ***Organization Development Journal***, 24: 2, 89-100.

Fleming, L., & Waguespack, D. M. 2007. Brokerage, boundary spanning, and leadership in open innovation communities. ***Organization Science***, 18: 2, 165-180.

Fosfuri, A., Giarratana, M. S., & Luzzi, A. 2008. The penguin has entered the building: The commercialization of open source software products. ***Organization Science***, 19: 2, 292-305.

Franko, L. G. 1989. Global corporate competition: Who's winning, who's losing, and the r&d factor as one reason why. ***Strategic Management Journal***, 10: 5, 449-474.

Füller, J., Matzler, K., & Hoppe, M. 2008. Brand community members as a source of innovation. ***Journal of Product Innovation Management***, 25: 6, 608-619.

Galaskiewicz, J., & Wasserman, S. 1989. Mimetic processes within an interorganizational field: An empirical test. ***Administrative Science Quarterly***, 34: 3, 454-479.

Garud, R., Jain, S., & Kumaraswamy, A. 2002. Institutional entrepreneurship in the sponsorship of common technological standards: The case of sun microsystems and java. ***Academy of Management Journal***, 45: 1, 196-214.

Geletkanycz, M. A., & Hambrick, D. C. 1997. The external ties of top executives: Implications for strategic choice and performance. ***Administrative Science Quarterly***, 42: 4, 654-681.

Goode, S. 2005. Something for nothing: Management rejection of open source software in australia's top firms. ***Information & Management***, 42: 5, 669-681.

Graebner, M. E. 2004. Momentum and serendipity: How acquired leaders create value in the integration of technology firms. ***Strategic Management Journal***, 25: 8,Äê9, 751-777.

Graebner, M. E., & Eisenhardt, K. M. 2004. The seller's side of the story: Acquisition as courtship and governance as syndicate in entrepreneurial firms. ***Administrative Science Quarterly***, 49: 3, 366-403.

Granovetter, M. 2005. The impact of social structure on economic outcomes. ***Journal of Economic Perspectives***, 19: 1, 33-50.

Green, S. G., & Gavin, M. B. 1995. Assessing a multidimensional measure of radical technological innovation. ***IEEE Transactions on Engineering Management***, 42: 3, 203.

Gruber, M., & Henkel, J. 2006. New ventures based on open innovation ,äî an empirical analysis of start-up firms in embedded linux. ***International Journal of Technology Management***, 33: 4, 356-372.

Gupta, A. K., Tesluk, P. E., & Taylor, M. S. 2007. Innovation at and across multiple levels of analysis. ***Organization Science***, 18: 6, 885-897.

Gupta, A. K., & Wilemon, D. L. 1990. Accelerating the development of technology-based new products. ***California Management Review***, 32: 2, 24-44.

Hamel, G. 2000. Waking up ibm. ***Harvard Business Review***, 78: 4, 137-146.

Hargadon, A. B., & Sutton, R. I. 1997. Technology brokering and innovation in a product development firm. ***Administrative Science Quarterly***, 42: 4, 716-749.

Harhoff, D., Henkel, J., & von Hippel, E. 2003. Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations. ***Research Policy***, 32: 10, 1753.

Hars, A., & Qu, S. 2002. Working for free? Motivations for participating in open-source projects. ***International Journal of Electronic Commerce***, 6: 3, 25-39.

Henderson, R. M., & Clark, K. B. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. ***Administrative Science Quarterly***, 35: 1, 9-30.

Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded linux. ***Research Policy***, 35: 7, 953-969.

Henkel, J. 2008. Champions of revealing--the role of open source developers in commercial firms. ***Industrial & Corporate Change***, 18: 3, 435-471.

Henkel, J. 2009. Champions of revealing--the role of open source developers in commercial firms. ***Industrial & Corporate Change***, 18: 3, 435-471.

Hidalgo, A., & Albors, J. 2008. Innovation management techniques and tools: A review from theory and practice. ***R&D Management***, 38: 2, 113-127.

Hienerth, C., & Lettl, C. 2011. Exploring how peer communities enable lead user innovations to become standard equipment in the industry: Community pull effects. ***Journal of Product Innovation Management***: s1, 175-195.

Hillery, G. 1955. Definitions of community. ***Rural Sociology***, 20: 2, 111-123.

Hitt, M. A., Hoskisson, R. E., Ireland, R. D., & Harrison, J. S. 1991. Effects of acquisitions on research & development inputs and outputs. ***Academy of Management Journal***, 34: 3, 693-706.

Hitt, M. A., Keats, B. W., & DeMarie, S. M. 1998. Navigating in the new competitive landscape: Building strategic flexibility and competitive advantage in the 21st century. ***Academy of Management Executive***, 12: 4, 22-42.

Hornby, J. 2009. ***Radical action for radical times: Expert advice for creating business opportunity in good or bad economic times***: SAS Publishing.

Ingram, P., & Roberts, P. W. 2000. Friendships among competitors in the sydney hotel industry. ***American Journal of Sociology***, 106: 2, 387.

Initiative, O. S. 2008. ***Open source initiative membership project***. http://www.opensource.org/membership.

Isaacson, W. 2011. ***Steve jobs***: Simon & Schuster.

Jeppesen, L. B., & Frederiksen, L. 2006. Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. ***Organization Science***, 17: 1, 45-63.

Jorgensen, N. 2008. Putting it all in the trunk: Incremental software development in the freebsd open source project. ***Information Systems Journal***, 11: 4, 321-336.

Kale, P., Dyer, J. H., & Singh, H. 2002. Alliance capability, stock market response, and long-term alliance success: The role of the alliance function. ***Strategic Management Journal***, 23: 8, 747-767.

Kanter, R. M. 1983. ***The change masters : Innovation for productivity in the american corporation***. New York: Simon and Schuster.

Katz, R., & Tushman, M. L. 1981. An investigation into the managerial roles and career paths of gatekeepers and project supervisors in a major r&d facility. ***R&D Management***, 11103-110.

Katz, R., & Tushman, M. L. 1983. A longitudinal-study of the effects of boundary spanning supervision on turnover and promotion in research and development. ***Academy of Management Journal***, 26: 3, 437-456.

Kim, W. C., & Mauborgne, R. e. 1999. Creating new market space. ***Harvard Business Review***, 77: 1, 83-93.

Kogut, B. 1988. Joint ventures: Theoretical and empirical perspectives. ***Strategic Management Journal***, 9: 4, 319-332.

Kogut, B., & Metiu, A. 2001. Open source software development and distributed innovation. ***Oxford Review of Economic Policy***, 17: 2, 248-264.

Kogut, B., & Zander, U. 1992. Knowledge of the firm, combinative capabilities, and the replication of technology. ***Organization Science***, 3: 3, 383-397.

Krishnan, M. S., Kriebel, C. H., Kekre, S., & Mukhopadhyay, T. 2000. An empirical analysis of productivity and quality in software products. ***Management Science***, 46: 6, 745.

Lakhani, K. R., & von Hippel, E. 2003. How open source software works: "Free" user-to-user assistance. ***Research Policy***, 32: 6, 923-943.

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

Lane, P. J., Koka, B. R., & Pathak, S. 2006. The reification of absorptive capacity: A critical review and rejuvenation of the construct. ***Academy of Management Review***, 31: 4, 833-863.

Langlois, R. N. 2002. Modularity in technology and organization. ***Journal of Economic Behavior & Organization***, 49: 1, 19.

Lee, G., & Xia, W. 2010. Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. ***MIS Quarterly***, 34: 1, 87.

Lenox, M. J. 2006. The role of private decentralized institutions in sustaining industry self-regulation. ***Organization Science***, 17: 6, 677-690.

Lerner, J., & Tirole, J. 2002. Some simple economics of open source. ***Journal of Industrial Economics***, 50: 2, 197.

Linder, J. C., Jarvenpaa, S., & Davenport, T. H. 2003. Toward an innovation sourcing strategy. ***MIT Sloan Management Review***, 44: 4, 43-49.

Lubatkin, M., Florin, J., & Lane, P. 2001. Learning together and apart: A model of reciprocal interfirm learning. ***Human Relations***, 54: 10, 1353.

Lumpkin, G. T., & Dess, G. G. 1996. Clarifying the entrepreneurial orientation construct and linking it to performance. ***Academy of Management Review***, 21: 1, 135-172.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. 2006. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. ***Management Science***, 52: 7, 1015-1030.

Malerba, F., & Orsenigo, L. 1995. Schumpeterian patterns of innovation. ***Cambridge Journal of Economics***, 19: 1, 47-65.

Marquis, C. 2003. The pressure of the past: Network imprinting in intercorporate communities. ***Administrative Science Quarterly***, 48: 4, 655-689.

McEvily, B., & Marcus, A. 2005. Embedded ties and the acquisition of competitive capabilities. ***Strategic Management Journal***, 26: 11, 1033-1055.

Milison, M. R., Raj, S. P., & Wilemon, D. 1992. A survey of major approaches for accelerating new product development. ***Journal of Product Innovation Management***, 9: 1, 53-69.

Mintzberg, H. 1978. Patterns in strategy formation. ***Management Science***, 24: 9, 934-948.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. 2002. Two case studies of open source software development: Apache and mozilla. ***ACM Trans. Softw. Eng. Methodol.***, 11: 3, 309-346.

Molm, L. D. 2003. Theoretical comparisons of forms of exchange. ***Sociological Theory***, 21: 1, 1-17.

Muñiz, A. M., & O'Guinn, T. C. 2001. Brand community. ***Journal of consumer research***, 27: 4, 412-432.

Murray, F., & O'Mahony, S. 2007. Exploring the foundations of cumulative innovation: Implications for organization science. ***Organization Science***, 18: 6, 1006-1021.

Nahapiet, J., & Ghoshal, S. 1998. Social capital, intellectual capital, and the organizational advantage. ***Academy of Management Review***, 23: 2, 242-266.

Nelson, R. R., & Winter, S. G. 1982. ***An evolutionary theory of economic change***. Cambridge, MA: Belknap Press of Harvard University Press.

Noda, T., & Bower, J. L. 1996. Strategy making as iterated processes of resource allocation. ***Strategic Management Journal***, 17159-192.

O Mahony, S., & Lakhani, K. R. 2011. Organizations in the shadow of communities. ***SSRN eLibrary***.

O'Mahony, S. 2007. The governance of open source initiatives: What does it mean to be community managed? ***Journal of Management & Governance***, 11: 2, 139-150.

Peteraf, M. A. 1993. The cornerstones of competitive advantage: A resource-based view. ***Strategic Management Journal***, 14: 3, 179-191.

Petkova, A. P., Rindova, V. P., & Gupta, A. K. 2008. How can new ventures build reputation? An exploratory study. ***Corporate Reputation Review***, 11: 4, 320-334.

Pittaway, L., Robertson, M., Munir, K., Denyer, D., & Neely, A. 2004. Networking and innovation: A systematic review of the evidence. ***International Journal of Management Reviews***, 5/6: 3/4, 137-168.

Podsakoff, P. M., MacKenzie, S. B., Jeong-Yeon, L., & Podsakoff, N. P. 2003. Common method biases in behavioral research: A critical review of the literature and recommended remedies. ***Journal of Applied Psychology***, 88: 5, 879.

Porter, M. E. 1980. ***Competitive strategy: Techniques for analyzing industries and competitors***. New York, NY: Free Press.

Powell, W. W. 1990. Neither market nor hierarchy: Network forms of organization. ***Research in Organizational Behavior***, 12295-336.

Powell, W. W., Koput, K. W., & Smith-Doerr, L. 1996. Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology. ***Administrative Science Quarterly***, 41: 1, 116-145.

Quinn, J. B. 1985. Managing innovation: Controlled chaos. ***Harvard Business Review***, 63: 3, 73-84.

Raymond, E. S. 2001. ***The cathedral and the bazaar***. Sabastopol, CA: O'Reilly and Associates.

Rigby, D., & Zook, C. 2002. Open-market innovation. ***Harvard Business Review***, 80: 10, 80-89.

Roberts, J. A., Il-Horn, H., & Slaughter, S. A. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. ***Management Science***, 52: 7, 984-999.

Rosenkopf, L., Metiu, A., & George, V. P. 2001. From the bottom up? Technical committee activity and alliance formation. ***Administrative Science Quarterly***, 46: 4, 748-772.

Rosenkopf, L., & Tushman, M. L. 1998. The coevolution of community networks and technology: Lessons from the flight simulation industry. ***Industrial & Corporate Change***, 7: 2, 311-346.

Rothschild-Whitt, J. 1979. The collectivist organization: An alternative to rational-bureaucratic models. ***American Sociological Review***, 44: 4, 509-527.

Rysman, M., & Simcoe, T. 2008. Patents and the performance of voluntary standard-setting organizations. ***Management Science***, 54: 11, 1920-1934.

Sampson, R. C. 2007. R&d alliances and firm performance: The impact of technological diversity and alliance organization on innovation. ***Academy of Management Journal***, 50: 2, 364-386.

Samuelson, P. 2006. Ibm's pragmatic embrace of open source.(international business machines corp.)(linux global partners). ***Communications of the ACM***, 49: 10, 21(25).

Santos, J. R. A. 1999. Cronbach's alpha: A tool for assessing the reliability of scales. ***Journal of Extension***, 37: 2, 1-5.

Schumpeter, J. A. 1934. ***The theory of economic development: An inquiry into profits, capital, credit, interest, and the business cycle***. Cambridge, MA: Harvard University Press.

Schumpeter, J. A. 1942. ***Socialism, capitalism and democracy***: Harper and Brothers.

Scott, W. R. 2002. ***Organizations: Rational, natural, and open systems*** (5th ed.). Englewood Cliffs, NJ: Prentice Hall.

Shah, R. H., & Swaminathan, V. 2008. Factors influencing partner selection in strategic alliances: The moderating role of alliance context. ***Strategic Management Journal***, 29: 5, 471-494.

Shah, S. K. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. ***Management Science***, 52: 7, 1000-1014.

Simon, H. A. 1962. The architecture of complexity. ***Proceedings of the American Philosophical Society***, 106: 6, 467-482.

Sivadas, E., & Dwyer, F. R. 2000. An examination of organizational factors influencing new product success in internal and alliance-based processes. ***Journal of Marketing***, 64: 1, 31-49.

Smith, K. G., Collins, C. J., & Clark, K. D. 2005. Existing knowledge, knowledge creation capability, and the rate of new product introduction in high-technology firms. ***Academy of Management Journal***, 48: 2, 346-357.

Snow, C. C., Fjeldstad, ò. D., Lettl, C., & Miles, R. E. 2011. Organizing continuous product development and commercialization: The collaborative community of firms model. ***Journal of Product Innovation Management***, 28: 1, 3-16.

Sourceforge.net. 2011. ***About sourceforge***. http://sourceforge.net/about.

Stallman, R. 1983. ***The gnu manifesto***. http://www.gnu.org/gnu/manifesto.html.

Stam, W. 2009. When does community participation enhance the performance of open source software companies? ***Research Policy***, 38: 8, 1288-1299.

Stuart, T. E. 2000. Interorganizational alliances and the performance of firms: A study of growth and innovation. ***Strategic Management Journal***, 21: 8, 791.

Szulanski, G. 1996. Exploring internal stickiness: Impediments to the transfer of best practice within the firm. ***Strategic Management Journal***, 1727-43.

Teece, D. J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. ***Research Policy***, 15: 6, 285-305.

Top500.org. 2010. ***Operating system family share for 11/2010***. http://www.top500.org/stats/list/36/osfam.

Torvalds, L. 1999. The linux edge. In S. DiBona, S. Ockman, & M. Stone (Eds.), ***Open-sources: Voices from the open-source revolution***: 101-111: O'Reilly Media.

Tushman, M., & Moore, W. L. 1982. ***Readings in the management of innovation***. Boston: Pitman.

Tushman, M. L. 1977. Special boundary roles in innovation process. ***Administrative Science Quarterly***, 22: 4, 587-605.

Tushman, M. L., & Anderson, P. 1986. Technological discontinuities and organizational environments. ***Administrative Science Quarterly***, 31: 3, 439-465.

Tzeng, C.-H. 2009. A review of contemporary innovation literature: A schumpeterian perspective. ***Innovation: Management, Policy & Practice***, 11: 3, 373-394.

Ulrich, K. T., & Eppinger, S. D. 2000. ***Product design and development*** (2nd ed.). Boston: Irwin/McGraw-Hill.

Utterback, J. M. 1971. The process of technological innovation within the firm. ***Academy of Management Journal***, 14: 1, 75-88.

Uzzi, B. 1996. The sources and consequences of embeddedness for the economic performance of organizations: The network effect. ***American Sociological Review***, 61: 4, 674-698.

Uzzi, B. 1997. Social structure and competition in interfirm networks: The paradox of embeddedness. ***Administrative Science Quarterly***, 42: 1, 35-67.

Uzzi, B., & Lancaster, R. 2004. Embeddedness and price formation in the corporate law market. ***American Sociological Review***, 69: 3, 319-344.

Van Genuchten, M. 1991. Why is software late? An empirical study of reasons for delay in software development. ***IEEE Transactions on Software Engineering***, 17: 6, 582-590.

Vanhaverbeke, W., Van de Vrande, V., & Chesbrough, H. 2008. Understanding the advantages of open innovation practices in corporate venturing in terms of real options. ***Creativity & Innovation Management***, 17: 4, 251-258.

Vesey, J. T. 1991. The new competitors: They think in terms of speed to market. ***Academy of Management Executive***, 5: 2, 23-33.

Von Hippel, E. 2001. Innovation by user communities: Learning from open-source software. (cover story). ***MIT Sloan Management Review***, 42: 4, 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.

Von Krogh, G., & Von Hippel, E. 2006. The promise of research on open source software. ***Management Science***, 52: 7, 975-983.

W3Techs. 2013. ***Usage statistics and market share of drupal for websites***. http://w3techs.com/technologies/details/cm-drupal/all/all.

Waguespack, D. M., & Fleming, L. 2009. Scanning the commons? Evidence on the benefits to startups participating in open standards development. ***Management Science***, 55: 2, 210-223.

Wang, C. L., & Ahmed, P. K. 2007. Dynamic capabilities: A review and research agenda. ***International Journal of Management Reviews***, 9: 1, 31-51.

Wernerfelt, B. 1984. A resource-based view of the firm. ***Strategic Management Journal***, 5: 2, 171-180.

West, J. 2003. How open is open enough?: Melding proprietary and open source platform strategies. ***Research Policy***, 32: 7, 1259.

West, J. 2007. ***Value capture and value networks in open source vendor strategies***. Paper presented at the 40th Annual Hawai'i International Conference on System Sciences, Waikoloa, Hawai'i.

West, J., & Bogers, M. 2013. Profiting from external innovation: A review of research on open innovation. ***Journal of Product Innovation Management***, Forthcoming.

West, J., & Gallagher, S. 2006a. Challenges of open innovation: The paradox of firm investment in open-source software. ***R&D Management***, 36: 3, 319-331.

West, J., & Gallagher, S. 2006b. Patterns of open innovation in open source software. In H. Chesbrough, W. Vanhaverbeke, & J. West (Eds.), ***Open innovation: Researching a new paradigm***: 82-106. Oxford: Oxford University Press.

West, J., & Lakhani, K. R. 2008. Getting clear about communities in open innovation. ***Industry & Innovation***, 15: 2, 223-231.

West, J., & O'Mahony, S. 2008. The role of participation architecture in growing sponsored open source communities. ***Industry & Innovation***, 15: 2, 145-168.

Westphal, J. D. 1999. Collaboration in the boardroom: Behavioral and performance consequences of ceo - board social ties. ***Academy of Management Journal***, 42: 1, 7-24.

Wilcox, J. 2007. ***Microsoft's open-source contradiction***. http://www.microsoft-watch.com/content/corporate/microsofts_open_source_contradiction.html.

Williamson, O. E. 1975. ***Markets and hierarchies, analysis and antitrust implications: A study in the economies of internal organization***. New York, NY: Free Press.

Williamson, O. E. 1991. Strategizing, economizing, and economic-organization. ***Strategic Management Journal***, 1275-94.

Yin, X., & Shanley, M. 2008. Industry determinants of the "merger versus alliance" decision. ***Academy of Management Review***, 33: 2, 473-491.

Zahra, S. A., & George, G. 2002. Absorptive capacity: A review, reconceptualization, and extension. ***Academy of Management Review***, 27: 2, 185-203.

Zeitlyn, D. 2003. Gift economies in the development of open source software: Anthropological reflections. ***Research Policy***, 32: 7, 1287.

Zollo, M., & Singh, H. 2004. Deliberate learning in corporate acquisitions: Post-acquisition strategies and integration capability in us bank mergers. ***Strategic Management Journal***, 25: 13, 1233-1256.