

THE EMERGENCE OF A NEW COMMERCIAL ACTOR:
COMMUNITY MANAGED SOFTWARE PROJECTS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
MANAGEMENT SCIENCE AND ENGINEERING MANAGEMENT
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Siobhan Clare O'Mahony
June 2002

UMI Number:

Copyright 2002 by
O'Mahony, Siobhan Clare
All rights reserved.

UMI Microform
Copyright 2002, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
Copying under Title 17, United States Code.

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

© Copyright by Siobhan C. O'Mahony 2002
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen R. Barley (Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert I. Sutton

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Mark Granovetter

Approved for the University Committee on Graduate Studies:

Abstract

Institutional theory has matured to the point where we know a great deal about how institutions, once formed, are reproduced. We know less about how new institutions are constructed. This may be because institutional theory does not have a well-developed notion of political process, which makes explaining institutional change problematic. Institutional theory also has difficulty explaining how individual actors contribute to the creation of social structures. Without such an understanding, we risk reifying structures that are our own creation. For this reason, some organizational theorists (McAdam and Scott, 2002; Davis and McAdam, 2000; Rao, Morrill & Zald, 2000; Swaminathan and Wade, 1999; Rao, 1998; Fligstein & Mara-Drita, 1996; Davis and Thompson, 1994) have used social movement concepts to better explain dynamics of change between economic actors. A few (Rao, Morrill & Zald, 2000; Rao, 1998) have postulated that social movements may provide a source of innovation for new organizational forms. This study extends recent research in this area by examining how social movements might inspire the creation of new organizing mechanisms.

Social movement theory explains the conditions under which collectives coalesce and mobilize support. It is grounded on studies of social and political action, where the targets of change are state, regulatory or policy actors. Instead of neglecting political processes in institutional life, social movement theory assumes conflictual relations between challengers and defenders¹ of established social orders. The social movement concepts of framing, resource mobilization, and political opportunity help explain how challenging groups achieve

¹ McAdam and Scott (2002) define challengers as those individuals, groups or organizations seeking to challenge the advantaged position of dominants or fundamental structural-procedural features of an institutional field. Dominants, or what I call defenders, are those individuals, groups or organizations around whose actions and interests the institutional field tends to revolve.

or do not achieve their goals. The possibility of mutual gains between challengers and defenders is, however, unexplored, largely because of the sharp dichotomies drawn between challenging and defending groups in prior research. My research uses a more integrative perspective than past work, examining how a new form of organizing and logics for interaction emerged as a product of the interactions between challengers and defenders of the established order. It unpacks the processes by which challengers and defenders adapted their practices to accommodate each other and shows how challengers and defenders, in negotiating their interests and rights, shape the design of a new organizational form.

Research Design. My research asks: if new organizing forms result from recombination of existing elements, how might challengers and defending groups affect their construction? To answer this question, an inductive, ethnographic approach was used to examine the interactions between community managed software projects from the open source social movement (challengers) and established firms in the software industry (defenders). With interviews of 70 contributors to community managed projects and close examination of the practices used by 4 projects to manage their interactions with firms, I find that both communities and firms changed some of their practices to better accommodate each other. They, however, also resisted each other's influence on dimensions critical to their identity and purpose. These two competing processes of accommodation and preservation resulted in the emergence of a new organizational form and a new set of logics to support community and firm collaboration.

Accommodation. Firms changed commonly accepted industry practices to conform to the community's normative and legal practices. Firms allowed employees contributing to community managed projects to retain the rights to their work and to represent themselves as individuals as opposed to representing the firm. Community

projects changed attributes of their informal and ideologically inspired form of organizing to facilitate their survival in a commercial context. Community projects incorporated as non-profits and defined a role for firms, albeit one without technical decision-making rights.

Preservation. Each actor was willing to make some concessions in order to better interact with the other, but they also resisted becoming more like each other in important ways. Firms used tactics to preserve their market power by influencing the technical direction and management of community projects. Community managed projects resisted such efforts with tactics to preserve their technical autonomy and their communal norms and values. Community members also adopted tactics to prevent the infiltration of rationalistic controls.

A New Form of Organizing. In negotiating their rights and interests, community managed projects and firms created a set of social arrangements that were new to both of them. Competing processes of mutual accommodation and preservation helped define the attributes of a new form: private non-profit software foundations. The design of the foundation's governance structure helps adjudicate the differing agendas and interests of the two actors and, yet, serves the interest of both parties. The foundation's restriction from redistributing profits satisfies the ideological values held by community members. Its formalized structure and legal status satisfies the needs of firms who wish to work with community projects. In the emergent negotiated order, firms are free to distribute community developed software although they do not have ownership rights to it. Community managed projects retain control and management of their work. Non-profit foundations hold their assets and help mediate between competing interests.

Discussion. If new organizational forms result from new combinations of old institutional elements, than a theoretical explanation of the emergence of new forms must be

able to explain processes of synthesis. I argue that processes of conflict and compromise are necessary to explain the synthesis of old and new institutional elements, and thus the emergence of new forms. Two traditional institutions: copyright and incorporation were adapted by community managed projects for new ends to better protect their values in a commercial setting. This research also provides further evidence that a social movement can be a source of institutional innovation. The real innovation, however, emerges from the interactions between the movement and incumbents, not just the movement itself.

Implications. Theoretical implications from this research suggest that social movement theorists need to move beyond assessing social movement outcomes to examine the broader effects that movement activity can established social structures. Furthermore, if social movement theory is to be useful to institutional theory, then a middle ground between political and apolitical assumptions needs to be found. In order to explain institutional change, we need to introduce processes of conflict to institutional theory, but it is a more nuanced version than the one historically embraced by social movement research. Practical implications from this research suggest a new loci for the development of software, innovation and standard setting, new types of actors that firms may engage with in the future, and new tactics for interacting with them.

Future research would do well to examine: how the network structure of contributors affects leadership of community managed software projects; how the shift from a free software to an open source software frame affected the formation of a market for open source software; the design of decision-making mechanisms in value rational groups; and how non-proprietary software affects the development of technical skills. In addition to contributing to organizational theory, this research contributes to studies of technology and innovation management as well as studies of technical and occupational communities.

Acknowledgements

Steve Barley is most responsible for piquing my interest in organizational behavior and technical work. Having known each other and worked together for the past 12 years, Steve has never treated me as anything less than his colleague and I am indebted to him for his guidance, intellectual stewardship and friendship. Bob Sutton has been a faithful and ever amusing co-advisor who has always helped me find resources and get to the point, while provoking new ideas. I thank Mark Granovetter for his time and insight, as he helped me enormously in directing my thoughts, often with a quick dry wit. I have also enjoyed the exchange and rapport of fellow department faculty such as Kathy Eisenhardt and Diane Bailey who have always been supportive and interested in my work.

I thank my informants, who were generous with their time and their experiences. I will refrain from naming them in order to maintain their confidence but want to thank Karsten Self, in particular, as he was ever patient in explaining the intricacies of the free software world to me and introduced me to many people. I also enjoyed the company of Scott McNeil who devoted his time and helped connect me to people. The interest and enthusiasm of Jan Chong, Lilly Irani, Matt Spiegelman and Vikam Vijayaraghavan made data collection more fun.

Stanford is a great place to study organizational behavior and I have enjoyed the interdepartmental exchange that has flourished over the past five years. At various talks and seminars I have benefited from the company and thoughtful insights of Jim March, Dick Scott, Woody Powell, Jason Owen-Smith, Jeffrey Pfeffer, JoAnne Martin, Tom Byers, Dan McFarland, Roberto Fernandez, Michael Morris and Neil Fligstein. I am grateful for the

experience of working on books with Tom Kelley, Robert Sutton, and Jeff Pfeffer. Michael Schrage's probing questions, interest and encouragement also spurred me on.

I thank the Social Science Research Council, Center for Work, Technology and Organization and the Stanford Technology Ventures Program for their support of my studies and research. Doug Guthrie and the SSRC Program on the Corporation made a difference in my transition from student to candidate and now to professor by providing feedback and introducing me to new opportunities and areas of study. Doug's skillful guidance of this program has turned it into an enormous asset for graduate students. I know that I benefited from the exchange it provided as well as from Doug's interest and care and hope that it will continue so others may also benefit.

The job search process was a great experience for me and the feedback from people who invited me to present at the University of California at Irvine, INSEAD, Georgetown, Boston College, MIT, Harvard, and Columbia was helpful in finishing my dissertation. I feel very lucky to have met so many talented and genuinely nice people in the field and thank Elaine Romanelli, Phil Anderson, Mike Tushman, Ray Horton and Heather Haveman for their interest and guidance.

Jim Pauli, my mentor and friend for the past ten years, helped me to make the transition from consulting to graduate school. I appreciated the support of EDS during my first few years and the encouragement of my former colleagues at Price Waterhouse and EDS, especially Julie C. Quinn, Elizabeth Livingston, Mike Cappellutti, and Robyn Dahlgren. Lori Cottle and Paula Wright helped by guiding me through the administrative haze that can daunt graduate students and doing all they could to help with recommendations and degree progress.

The person who has perhaps borne the worst of the frustrations that can accompany graduate studies is Fabrizio Ferraro, who has been an unfailingly loyal and encouraging officemate. I feel truly lucky to have earned the friendship of Victor Seidel who has shown me how to raise the bar and sets a new standard for how to treat people. Department alumni Beth Bechky, Jeannie Kahwajy, Robin Dillon and Jolin Marie Salazar-Kish were also on the team that kept me sane. My graduate school experience would not have been as amusing without Mark Mortensen, Victor Seidel, Fabrizio Ferraro, Stine Grodal and Kelley Porter's hallway conversations. I will miss the WTO 4th floor more than I can say, but I look forward to continuing to collaborate with my Stanford colleagues. I know they will miss having someone to tease - but Stine is more than qualified.

I am grateful for the fun and valued opinions garnered on long 'Dish' walks with Sarah Reybuck, Antje Kann and Sheira Ariel, whose good cheer always inspired me, and for 'happy lunches' with Christina Fong, Agnish Chakravarti, Greta Hsu, and Ozgecan Kocak. Susan Kaplan made a difference in my life by providing a ready ear and good counsel. Rachel Gomez has been a wonderful listener, friend and helped make the last five months easier for me. Joelle Maher, Will Eagle, Parnell Pearson, Joe Gilmartin and the rest of the gang at the Black Horse always welcomed me and never failed to provide laughs. I thank my friend of 17 years, Leigh Ferreira, who has always reminded me when to take a break, have some fun, and focus on the bright side. I will miss our nights out in the city and the good times with Jodi and Marcus Bannon and Marie Ferreira. Most importantly, I thank my grandfather for his continual support; my brother and sister who always support me; and my parents for never holding me back from pursuing my interests and for always telling me I could.

Table of Contents

Abstract

I.	Theoretical Motivation.....	1
	A. First Wave: A Dynamic Institutionalism.....	1
	B. The Second Wave: Explaining Institutional Persistence.....	3
	C. The Third Wave: Explaining Institutional Change.....	7
	D. The Relevance of Social Movement Theory.....	11
	E. Research Question.....	17
	F. Developing a Research Strategy.....	18
II.	Context of Study.....	23
	A. Origins of Free Software Movement.....	23
	B. GNU/Linux and the Community Development Model.....	28
	C. Frame Transformation: Creating the Open Source Frame.....	34
	D. Market Response to Frame Transformation.....	43
	E. Explaining Mobilization and Growth.....	46
III.	Methods.....	54
IV.	Different Actors and Logics for Interaction.....	61
	A. Community Logics for Commercial Participation.....	61
	B. Commercial Logics for Community Participation.....	66
	C. Differences and Interaction Difficulties.....	74
V.	Community Managed Projects.....	78
	A. Accommodation.....	79
	1. Incorporation.....	79
	a. Threats from Commercial Interests.....	85
	b. Threat of Rationalistic Controls.....	86
	2. Representation.....	90
	3. Defining a Role for Firms.....	94
	B. Preservation.....	96
	1. Limit the Foundation's Authority and Role.....	97
	a. Guarding the Technical Domain.....	98
	b. Limiting Employment.....	106
	2. Delimit the Roles Available for Firms.....	114
	a. Restricting Board Composition.....	114
	b. Creating Mechanisms for Voice.....	120
	3. Protecting the Community's Intellectual Property.....	121
	a. Licensing Terms.....	122
	b. Normative Control through Public Forums.....	125
	c. Protection of Brand.....	128
	d. Reassignment of Individual Rights to the Collective.....	131

VI.	Firms.....	135
	A. Accommodation.....	136
	1. Individual Representation.....	136
	2. Work Assignment Rights.....	139
	3. Release of Proprietary Code.....	142
	4. Adapt Work Style.....	144
	B. Preservation.....	147
	1. Sponsor Contributors in Key Roles.....	152
	2. Donate Resources.....	158
	3. Shape Foundation.....	163
VII.	Discussion.....	167
	A. Political Opportunity and Context for Collaboration.....	168
	B. The Negotiated Interaction Order.....	174
	C. Comparison of Foundations.....	178
	D. Recombination.....	182
	E. Comparison with Traditional Forms.....	184
	F. Efficacy of Foundations.....	186
	G. Sponsored Contributors and Role Conflict.....	190
	H. Founding Movement: Frame Elasticity and Dilution.....	193
VIII.	Implications.....	200
	A. Contributions.....	201
	B. Sustainability.....	203
	C. Limitations.....	208
	D. Generalizability.....	209
	E. Alternative Explanations.....	210

References

Tables and Figures

List of Tables and Figures

Table II-1	Comparison of Traditional and Community Software Development Models
Table II-2	Frame Comparison
Figure II-1	Frame Transformation and Media Mentions
Figure II-2	Open Source and Linux Firms Receiving Round One Venture Capital Funding
Table III-1	Theoretical Rationale for Project Selection
Table III-2	Project Attributes
Table III-3	Informant Demographics
Table V-1	Changes in Community Practices
Table V-2	Community Preservation Tactics
Table VI-1	Changes in Firm Practices
Table VI-2	Firm Preservation Tactics
Table VII-1	Comparison of Foundation Attributes
Table VII-2	Comparison of Organizational Forms

Chapter I. Literature Review

This chapter examines three waves of thought that have characterized the empirical and theoretical treatment of institutional change. It is argued that a stronger theory of action, political processes and exposition of the relationship between individual action and the creation or dissemblance of social structures is needed in order to explain institutional change. Social movement concepts are recommended as one way to help further this agenda. However, a conflict driven basis for action and strong dichotomies between types of actors may inhibit the application of social movement to organizational theory. The chapter concludes with a revised agenda for institutional theory that borrows from social movement theory, poses two research questions and explains the strategy pursued to operationalize this study.

A. The First Wave: A Dynamic Institutionalism. Old institutional theory (Hughes 1936; Hughes 1937; Selznick, 1949) assumed that conflict, negotiation and power were integral to institution building (Hirsch, 1997: 1706). In contrast to the staid and often pejorative sense of the word institution today, “old” institutionalists (e.g. Hughes, Selznick) analyzed factions and constituents that contested the development of institutional frameworks and agendas, chronicling these political processes with passion, vigor, and detail. Change was viewed as vital to institutional construction and adaptation. Any snapshot of institutional life would provide the illusion of a disquieting equilibrium that hid the amalgamation of processes lying beneath. Institutions were conceived of as the product of mediation, political negotiation and power contests on the organizational battlefield (Abbott,

1992: 755) and not solely as instrumental solutions to the problem of collective action (Ostrom, 1990).

Selznick's study of the Tennessee Valley Authority (Selznick, 1949) shows how an institutional mandate can change when united with its local context. His analysis of the role of constituencies and special interest groups and their ability to influence or co-opt institutional mandates testifies to the fact that politics matter, particularly in the construction of new institutions. This is a good example of empirical work in the tradition of the first wave of institutional theorists. Similarly, Hirsch's 1975 study found that the pharmaceutical industry was able to establish powerful trade associations and favorably influence patent and copyright laws whereas the recording industry was not as proactive in establishing such institutional mechanisms, and did not receive such favorable gains (Hirsch, 1975). This demonstrates that although the ability of an industry to affect its institutional environment could vary, it was feasible.

Institutional change was an important focus of old institutional theory. Selznick, when reflecting on his own influential body of work (1996), discussed his vision of what a theory of institutions should offer. "Institutional theory *traces the emergence* of distinctive forms, processes, strategies, outlooks, and competences as they emerge from patterns of organizational interaction and adaptation²" (Selznick, 1996: 271). He defined institutionalization as "the emergence of orderly, stable, socially integrating patterns out of unstable, loosely organized, or narrowly technical activities" (Broom and Selznick, 1955: 238). Selznick's emphasis on the word

² Emphasis added.

“emergence” indicates that the process of institutional construction should be a central tenant.

Old institutional theorists recognized that institutions constrain conduct in two ways: by bringing individual behavior within a normative order, and by making individual actions hostage to the history of prior action (Selznick, 1992: 232). Individual or collective action is never free from the shadows of established social norms and structures. However, their focus was not whether this normative order was constraining, but on whether those that strove to change social structures could garner the organization, power, and resources necessary to do that. Thus, the processes and tactics used to achieve those ends were also of interest.

B. The Second Wave: Explaining Institutional Persistence. The second wave of institutional theory, also called ‘new institutional theory’ (e.g. DiMaggio and Powell, 1991; Meyer and Rowan, 1983) argues that the old institutional studies did not appreciate how larger institutional dynamics like organizational fields, industry, law, and the state affect the construction and maintenance of institutions. Greater explanatory weight is placed on the existing social order as an inhibitor of change. Shared cultural understandings with normative, cognitive, and regulative underpinnings help organizational actors understand what behavior is appropriate in larger institutional fields (Scott, 1995). Compliance with taken for granted institutional blueprints or norms is rewarded with the benefits of legitimacy, such as status and access to resources which can further aid in organizational survival. Organizations thus mimic each other: conforming with commonly accepted ways of organizing in order to gain legitimacy, eradicate uncertainty or comply with professional norms (DiMaggio and Powell, 1991; Meyer

and Rowan, 1991) and may do so even when conformity conflicts with efficiency (Meyer and Rowan, 1991). This helps explain the stability of institutions and their resistance to change (DiMaggio and Powell, 1991; Meyer and Rowan, 1991).

Empirical and theoretical work in this tradition helped explain how homogeneity in organizational form and practice is sustained (Jepperson, 1991) even when it may be de-coupled from operational requirements. Researchers have well documented how organizations in heavily institutionalized environments conform to similar cultural understandings (Zucker, 1977) and imitate each other by adopting similar organizational structures and practices (DiMaggio and Powell, 1994; Palmer, et al, 1993; Fligstein, 1985). Specific examples include: acquisition practices (Haunschild, 1993); product cycles (Zander and Kogut, 1995); corporate giving (Galaskiewicz, 1985); organizational leadership choices (Fligstein, 1987) and civil service reform (Tolbert and Zucker, 1983). Greenwood and Hinings (1993) found that organizations tend to operate with a set of coherent structures and systems or organizing blueprints. Knowing that organizations imitate each other, are more likely to do so in more institutionalized contexts (Goodrick and Salancik, 1996), and that this is related to their chances of survival (Baum and Oliver, 1991) is now well established.

Critics of the new institutionalism questioned the conditions under which mimetic forces will operate as well as the extent to which firms would mimic each other. Two studies (Tolbert and Zucker, 1983 and Westphal, Gulati, and Shortell, 1997) found that late adopters of organizational practices were more likely to be affected by institutional forcers than were early adopters. Early adopters of civil service reform (Tolbert and Zucker, 1983) and Total Quality Management (TQM)

practices (Westphal, 1997) did so for internal organizational requirements while later adopters did so for normative reasons to enhance their legitimacy. This temporal distinction, suggests that a more dynamic approach that considers political processes may be needed to pinpoint just how and when organizations absorb and react to the understandings of their institutional environments as well as a greater understanding of the mechanisms by which understandings become shared.

In their review of institutional research, Scheiberg and Clemens (2001) characterize much of this literature as borrowing from the imagery of epidemiology, where an infectious agent is released, there is exposure among agents, and the presence or absence of trend lines are identified. The micro-interactions that shape institutional creation, legitimation and destruction receive little attention. The danger with this approach is that the further away the level of analysis moves, the further we move from identifying the mechanisms that can act as the levers of change. Those theorists who have examined mechanisms that transmit institutional understandings and practices have found interesting and often counter-intuitive findings.

For example, Fligstein's study (1985) of the spread of the multidivisional form from 1919-1979 found that the adoption of the multidivisional form was related to the background of organizational leaders and their perceptions of what their competitors were doing. Fligstein argues that key actors' construal of their situation may depended upon their structural position within the organization and their links to other organizations in the environment (1985: 389). Similarly, Galaskiewicz and Burt (1991) found evidence of inter-organizational imitation in philanthropy practices, but concluded that imitation operated not through behavioral cohesion but

through structural equivalence. Thus, similar role structures within and outside of firms may encourage the adoption of similar norms as opposed to social affiliation.

These two studies point to the danger in assuming that institutional understandings come to be widely shared through an unknown viral process. If shared understandings about the way organizations are designed are communicated through structural equivalence as opposed to social affiliation, it suggests very different sources for institutional change. It is thus not enough to know that a relationship exists, we need to know why it exists (Hedstrom, 1998). Without that knowledge, we have black boxed one of the most important social processes critical to social change – the emergence of institutions. Furthermore, identification of mechanisms may help us understand why consequences differ (Elster, 1998) and lead to very different policy and behavioral implications.

More recently, institutional theorists have tried to reconcile these divergent perspectives on institutional change by identifying conditions in which organizations adapt to or affect their institutional environments (Hoffman, 1999; Kraatz and Zajac, 1996; Greenwood and Hinings, 1996; Holm, 1995; Elsbach, 1994; DiMaggio, 1991; Oliver, 1991; Brint & Karabel, 1991). There is empirical evidence that institutions' position on critical issues can change over time (Hoffman, 1999); that organizations can influence their environments (Oliver, 1991; Elsbach, 1994); and that institutions can change in ways that are inconsistent with the predominant form in their environment (Kraatz and Zajac, 1996).

In defining conditions for different types of change, Greenwood & Hinings (1996) postulate that the more embedded an organization is in a normative institutional structure, the greater the likelihood of evolutionary as opposed to

revolutionary change. Others have returned to Hirsch's 1975 approach to study how an organization's position or attitude on critical issues changes over time (Hoffman, 1999) or have identified the types of strategies organizations use to influence their institutional environments (Oliver, 1991; Elsbach, 1994) and their effectiveness in acquiring the desired changes (Elsbach, 1994). Kraatz and Zajac's study of over 500 liberal arts colleges (1996) found that many colleges changed in ways that differed from their institutional environments and did so without the costs previously associated with illegitimacy. Instead, they found that institutional changes were closely related to customer and technical preferences, which suggests that organizations may be more responsive to environmental conditions than institutional theory would predict.

Institutional theorists have amply demonstrated that established institutions have a powerful influence on organizational life. This helps explain why organizations may be slow to adapt to their environments. What remains unexplained is how new institutions emerge, disappear or change. In attempting to redress the old institutional perspective's neglect of larger structural forces, the pendulum swung too far in the other direction. The gains made in understanding stability and social reproduction in the second wave came at the expense of the gains made in the first wave: understanding how competing actors, motives, and interests shape the construction of institutions (Hirsch, 1975; Hirsch and Lounsbury, 1997; Hirsch, 1997).

C. The Third Wave: Explaining Institutional Change. Had this imbalance gone unaddressed, institutional theory would be in danger of becoming a theory of institutional stability. Luckily, its success in explaining the durability of

institutions and its difficulty in explaining their mutability (Clemens and Cook, 1999) has helped ignite new thinking on the subject, what I call institutional theory's third wave. This wave of research, which began at the turn of this century, moves beyond explaining inter-organizational influence, adoption and dissemination of similar practices and forms, to examine the creation, destruction, and replacement of institutions.

One way to correct for the second wave's preoccupation with structure is to reintroduce a stronger notion of agency and to explicate the link between individual action and the creation of institutional structures at a more refined level of analysis (Greenwood and Hinings, 1996; Hirsch and Lounsbury, 1997; Selznick, 1996). Hirsch and Lounsbury (1997) argue that institutional theory's central weakness is an ambiguous theory of action that does not explicate how micro-level action contributes to macro-level change.

A theory of action is critical to any theory hoping to explain the constitution and influence of institutions for three reasons. First, it provides a link between the micro dynamics of individuals and social structure. A theory of purposive action is necessary to understand how and why the actions of individuals combine to produce social outcomes and connect people to macro-social consequences (Coleman 1986: 1321). Schneiberg and Clemens (2001) point out that the current model of action in institutional theory is built upon evidence of higher order effects combined with the absence of associations predicted by economic or technological determinism or the presence of mimetic organizational behavior. The tactics and strategies that link individual motivations and goals to collective action are murky.

Second, if we accept the duality of structure (Giddens 1984; Sewell, 1992) that is embodied by the notion of institution, we accept that although institutions constrain actions on some levels, they are nothing more than the endogenous productions of our own actions. We cannot learn of their making without a dynamic perspective that includes the construction, maintenance, and deconstruction of institutional life. In order to adjudicate between “institutions as constraints on action” and “institutions as culturally constitutive of actors”, levels of analysis between individuals and social structure must be linked (Schneiberg and Clemens, 2001). Failing to adjudicate between these two tensions results in a reified notion of social structure.

Third, if political processes are necessary to overturn and change existing institutions (Burns, 1961) than we cannot access these processes without a stronger theory of action. I agree with Hirsch and Lounsbury’s analysis (1997) that a greater theory of action is necessary but also draw upon Clemens and colleagues arguments (1993; Clemens and Cook; 1999; Schneiberg and Clemens, 2001) that an integrated institutional perspective on change requires a more nuanced understanding of political processes. Institutional theorists such as Holm (1995) and Clemens and colleagues (1993; Clemens and Cook; 1999; Schneiberg and Clemens, 2001) have persuasively argued that a deeper understanding of political processes is needed to explain how different modes of organizing gain and lose power and popularity over time. As Holm observed “new institutions are not built from scratch, but are built upon older institutions and must replace or push back preexisting institutional forms” (Holm, 1995: 400). The ‘push back’ of preexisting institutional forms requires an understanding of the dynamics of both the ‘push’ (a greater action

orientation) and the 'back' (the political context in which action occurs). Thus, we need to understand political action, not just action. That is, how patterns of action map on to competing actors, interests and motives.

These three elements are needed to explain how individual actions contribute to the construction of new social structures. Without these essential links, institutions are mysteriously constituted and portrayed as a force which is 'out there' and which acts on organizations. Concurrently, a well-developed theory of action must maintain respect for the power of institutional influence. The problem then becomes one of understanding how and why the actions of individuals combine to produce new types of organizing mechanisms within the context of their embedded institutional landscape.

A renewed attention to the sources, mechanisms and processes of institutional change was crystallized in a recent special issue of the *Academy of Management Journal* (Dacin, Goodstein, and Scott, 2002). Most notably, several qualitative studies in this issue demonstrate the micro-interactional and contested nature of institutional change (Greenwood, Suddaby, Hinings, 2002; Zilber, 2002; Garud, Jain and Kumaraswamy, 2002; Lawrence, Hardy, and Phillips, 2002; Townley 2002). Competing rationalities (Townley, 2002); a change from a regulatory to a market logic (Lounsbury, 2002) and intense collaborations (Lawrence, Hardy, Phillips, 2002) provide sources of institutional innovation. The introduction of leaders with different backgrounds and philosophies (Kraatz and Moore, 2002; Zilber, 2002); the endorsement of professional associations (Greenwood, Suddaby, Hinings (2002) and the emergence of proto-institutions (Lawrence, Hardy, Phillips, 2002) help enact institutional change. This renewed attention to different types of

actors, motives, rationalities and political processes is promising in that these studies are better able to pinpoint the processes by which institutional change happens. Theorists in this stream have also drawn upon social movement theory to do the same thing. Social movement concepts, which embrace a strong theory of action and respect for political process, may help provide the engine that institutional theory lacks.

D. The Relevance of Social Movement Theory. Social movement theory can explain how collectives coordinate their efforts to challenge the existing order and effect change. A social movement is defined as “a set of opinions and beliefs in a population representing preferences for changing some elements of the social structure or reward distribution, or both, of a society” (McCarthy and Zald, 1977: 20). Social movement theorists typically ask very different questions than do organizational theorists, which may explain why there have been so few links between these two areas (McAdam & Scott, 2000; Scott, 1999). While organizational theorists are traditionally more interested in explaining dynamics of the dominant social and economic order, social movement theorists are more interested in understanding challenges to the established order and sources of instability within the existing infrastructure (McAdam & Scott; Scott, 1999).

Social movement theory has grown from its initial reliance on discontent as a motive for organizing and now devotes more emphasis to the ability of social movements to mobilize participants, supporters, and resources and make use of their existing societal infrastructure for their survival and goal attainment (McCarthy and Zald, 1977). The framing of cultural understandings (Snow, et al, 1986), mobilizing structures (McAdam et al, 1996) tactical innovations (McAdam, 1983), and strategic

interactions between social movements and the existing order (McAdam, 1983) are argued to be critical to explaining the success of collective action. Grievances or a sense of injustice may help mobilize resources, but may not be sufficient. Social movements are viewed as being strategic in their interactions with the established institutional order and their ability to use resources such as media and the law to advance their causes, while acknowledging that these same resources can constrain their advance.

Empirical work has shown (McAdam 1983; McAdam and Paulsen 1993) that social movements do not emerge spontaneously but that supporters are recruited and resources acquired through preexisting social networks and institutional structures. McAdam and Paulsen (1993) found that individual ties with other volunteers were not as important as a multiplicity of ties embedded in many organizations that supported the movement in explaining the recruitment of civil rights volunteers. Individual connections with 'like-minded' organizations made them more susceptible to join and support a 'like-minded' social movement.

Social movements mobilize by creating or taking advantage of political opportunities and frame issues to develop shared meanings that both characterize the movement and its place in the larger institutional order. The concept of frame derives from Goffman's (1974) analysis of how schematas link individual interpretations of events to broader social systems. Frames guide the process of meaning construction, helping individuals to interpret and organize their experiences in a way that enables future collective action. Social movements use frames to guide individual interpretations and align them with those of the movement in order to garner further support and action (Snow et al, 1986).

A strength of the social movement perspective is its exposition of the mechanisms collectives use to advance their causes despite significant power and resource imbalances. Frame construction and alignment (Snow et al, 1986; Snow and Benford, 1992) and tactical innovations (McAdam, 1983; Clemens, 1993; Morris 1993) as well as the timing of political opportunities have been found to be central to a social movement's ability to affect change. What can organizational theory learn from social movement theory? By understanding the tactics, timing, mobilizing structures, and challenger's patterns of interaction between challengers and defenders of the established order, we might be able to unpack the word 'politics' to identify patterns that contribute to institutional change.

Social movement theory also offers a more fluid conception of boundaries, membership, and actors whose relations are negotiated in relation to existing social structures through processes of identity formation and framing. Davis and McAdam (2000) question whether taking organizations as units of analysis is helpful to understanding an economy that is increasingly episodic and network based. "We have argued that the imagery of organizations-as-units has finally become more misleading than enlightening, leading one to ask the wrong kinds of questions and use the wrong kinds of mechanisms to make sense of the social structure of the economy" (Davis and McAdam, 2000: 214). In their call for new vocabularies to explain new social structures, Davis and McAdam urge organizational theorists to reconsider the power of social movement theory as its theoretical vocabulary may be more appropriate for an economy that no longer depends on organizations as well bounded units of economic activity. Indeed, Fligstein and Mara-Drita's examination of how institutional entrepreneurs used framing and mobilizing tactics to help create

the Single Market Program for the European Union demonstrates that new world orders may require new units of analysis (1996).

A weakness of social movement theory is its assumption of dichotomous challengers and defenders that do not share hope for mutual gains. Furthermore, social movement theorists have construed social movement outcomes narrowly, which may not capture how their actions impinge on established structures. Analysis of how interactions between social movements and defenders of the established order might change each other is not typical of social movement research. Social movement outcomes are typically viewed as the ability of the movement to realize its goals (Giugni, 2000). For example, in Gamson's survey of 53 challenging groups (1990), he identifies four possible outcomes for social movements: full response, co-optation, preemption, and collapse³. All of these outcomes focus on changes to the movement, and interestingly enough, not to the target of their discontent.

Social movement theorists agree that examination of the consequences of social movement activity is underdeveloped (Cress and Snow, 2000 citing McAdam, McCarthy, and Zald, 1988, Giugni, 1998; Giugni, 2000). Most work focuses on the political and policy outcomes desired by the movement as opposed to broader cultural and institutional effects (Giugni, 2000). This may be because social movement theorists have been too movement centric and neglectful of the

³ Gamson assesses how challenging groups may or may not reach their goals along two dimensions: the degree to which they are acknowledged and accepted as a challenging group by the defending order and the degree to which they obtain new advantages. Acceptance results when challenging groups are viewed by their targets as representative of a legitimate set of interests (1975/1990). He defines full response as the achievement of acceptance and new advantages and collapse as the achievement of neither. Co-optation refers to social movements that have gained acceptance without new advantages and preemption to refer to movements that haven't received recognition or acceptance, but have made new advantages that help pursuit of their goals. (Gamson, 1975/1990: 29).

institutional fields in which they operate (McAdam and Scott, 2002). In Giugni's (2000) review of social movement outcomes, he argues for greater attention to be paid to "the processes and dynamics that allow movements to make an impact on different aspects of society" (Giugni, 2000). In order to identify those effects, one must step outside the social movement as a unit of analysis and assess interactions with established institutions that might produce institutional change. This is also necessary if social movement theory is to inform organizational theory. Through understanding the dynamics between challengers and defenders and not just the actions of challengers that social movement theory will be able to be useful to organizational theory.

Critics might argue that social movement theorists focus on exceptions or outliers to the established social order, and may have little to contribute to studies of routine economic activities. It is true that social movement theory has been grounded in political and social action as opposed to commercial action. However, Davis and McAdam (2000) respond that much of what is considered "routine economic activity" is actually nothing of the sort, but characterized by turbulence, as any student of the diffusion of innovations would agree. The Davis and McAdam model (2000) applies social movement concepts to explain innovation and change within an industry. In this model, the emergence of new industries is characterized as a "highly contingent process in which destabilizing changes set in motion a sequence of linked mobilization dynamics" (Davis and McAdam, 2000: 218). Collectives challenge industry leaders with different attributions of a threat or opportunity. For example, they may develop an innovative form of collective action

that escalates uncertainty and leads to sustained industry turbulence (Davis and McAdam, 2000).

Some organizational theorists have theorized how social movement concepts might apply in a commercial context (McAdam and Scott, 2002; Davis and McAdam, 2000; Rao, Morrill and Zald, 2000; Swaminathan and Wade, 1999; Rao, 1998; Davis and Thompson, 1994). Social movement theory has been used to explain the emergence of new industries and new markets (Swaminathan and Wade, 1999), changes in the power and structure of shareholders (Davis and Thompson, 1994) as well as new forms and institutions (Rao, 1998; Fligstein and Mara-Drita, 1996; Rao, Morrill and Zald, 2000). In revisiting earlier studies of the microbrew and gourmet coffee industry, Swaminathan and Wade (1999) found that entrepreneurs used resource mobilizing tactics usually associated with social movements to secure legitimacy and market share in the face of opposition from more established institutions. They conclude that new institutions are more likely to survive if they construct an identity that is distinct from the dominant form and difficult for incumbents to assume (Swaminathan and Wade, 1999).

Rao, Morrill and Zald (2000) use social movement theory to explain not just the mobilizing and tactical behavior of collectives, but also the emergence of new organizational forms. The construction of new forms is viewed as a political process in which social movements work to de-institutionalize existing beliefs about extant forms and to establish new forms that are consistent with the movement's beliefs and values. They may use tactics such as boundary setting and maintenance in relation to the incumbent order in order to acquire legitimacy. Incumbent institutions are challenged and may be restructured via political contestation through

the infusion of new norms, values and ideologies. In order for challenging forms to become institutionalized, “activists construct boundaries around activities and validate these boundaries such that a new category of organizations emerges” (Rao, Morrill, and Zald, 2000).

E. Research Questions. Without a theory of action and adequate incorporation of political processes and conflict, it is difficult to explain institutional change. Social movement concepts are useful in explaining processes of emergence and contestation. However, if social movement theory is to be useful to organizational theory, it must move beyond examination of cycles of contention and look at the dynamics of interaction between challengers and defenders of the established order. Challengers to incumbent institutions can provide fodder for institutional innovation (Rao, Morrill and Zald, 2000) and might have more incentive to do so (Clemens, 1993). Clemens study of women’s groups (1993) showed that groups that were marginalized by existing institutions had greater incentives and freedoms to develop alternative forms for organizing to offset their lack of power. .

If social movements can provide a source of organizational innovation and new organizational forms are the product of recombining existing organizational elements (Rao, 1998), how does this recombination occur? If new organizational forms result from recombination of existing elements, how might challenging and defending groups affect their construction? How might interactions between challengers and defenders of established institutions lead to the creation of new social structures? If it could be shown that a challenging group altered the behavior of incumbents, we might gain a fuller understanding how patterns of contestation affect institutional change. A theoretical explanation of the emergence of new forms must be able to

explain how competing actors and philosophies become integrated. Such a theory requires more compromise than social movement theory offers and more conflict than institutional theory currently offers. Neither theory alone can adequately explain how new forms of organizing emerge and become an accepted part of institutional life.

F. Developing a Research Strategy. Developing a research strategy that links social movement and institutional theory requires careful consideration of both context and method. Traditional institutional research, of the epidemiological sort, often operationalizes various “institutions” as independent variables and seeks to explain patterns of diffusion across organizations and the consequences of adoption for organizational survival (Schneiberg and Clemens, 2001). Schneiberg and Clemens (2001) have argued that “institutionalist research designs must confront two core challenges: capturing the influence of higher-order factors on lower-level entities and identifying specific institutional mechanisms”. If our standard research designs always consider institutions as independent variables, then we run the risk of assigning social structures both causal and temporal primacy over individual actors. Instead of measuring exposure and infection, a study of institutional emergence requires tracking the changing cast of competing actors and identifying the shifting repertoire of alternative models or practices (Schneiberg and Clemens, 2001). This suggests the need for a grounded theoretical approach where institutions might emerge as the outcome of study.

Despite the fact that many sociologists lament the variablization of sociology associated with survey research⁴ (Homans, 1986; Coleman, 1986; Sorensen, 1998) and advocate a return to studies that are richer in social context, they are still a rarity. Thus, because concepts central to institution building such as power, interests and leadership are difficult to measure and highly sensitive to context in their operation (Coleman, 1986), a situated or in vivo study is better suited for detecting the micro-level interactions by which structure is both replicated and challenged.

In fact, it is difficult to detect ways in which social structures are challenged outside of their natural settings. The advantage of an in vivo study is that the context of social relations is not abstracted from social and cultural data (Arensberg, 1954, 1968). Sociologists of every day life (Adler, et al, 1987) argue that we cannot learn of patterns among social relations by detaching them from their naturally occurring context. If interaction between challenging and defending groups is of interest, the integrity of social actors and action must be respected as they are found. Thus a qualitative approach that appreciates the situational is better suited for detecting how individuals affect macro level structural changes.

Context of study is also of greater importance for this type of exploration as much organizational research occurs in highly institutionalized settings that may not be conducive to the detection of mechanisms that foster change. Weick (1974) has advocated that organizational researchers should pay close attention to not only what

⁴ Coleman feared that by isolating specific attributes to characterize collective action, individual concerns and motivations would become atomized and the role of community as a context for decision-making obscured. "The end result (of survey studies) is extraordinarily elaborated methods for analysis of the behavior of a set of independent entities (most often individuals), with little development of methods for characterizing systemic action resulting from the interdependent actions of members of the system" (Coleman, 1986: 1316).

they theorize about, but where they do it. He encourages researchers to avoid studies of big business and the typical sponsors of organizational research and to instead focus on every day events, places, absurd organizations or what he terms 'elsewheres' - places that are typically below the radar of organizational researchers. Criteria for research settings that meet Weick's definition of 'elsewheres' would have low levels of legitimacy, lack formal recognition or be what Clemens (1993) and Schneiberg and Clemens (2001) would call on the periphery of institutionalized organizational fields. "(M)argins matter. We need to pay close attention to processes that unfold in the peripheries, interstices and overlaps of institutional fields. Such social locations are found where authority structures are weak, where participants are exposed to multiple models or logics, and where, consequently, actors have the opportunity and the resources to experiment with new or hybrid forms." (Schneiberg and Clemens, 2001). In such non-traditional settings, phenomena may be more visible and available for hypothesis generation and lead to more theoretically divergent outcomes.

Several other theorists have advocated a similar, but more modest approach by noting that groups in low institutionalized fields or groups facing some type of economic, political, or technological transformation may be fruitful for developing grounded theory about how the actions of individuals affect structural outcomes (Fligstein and Dauber, 1989; Greenwood and Hinings, 1996). Fligstein and Dauber speculate that "organizations facing problems that seem to demand solutions outside the repertoire of normal practice" are good settings for detecting institutional change (1989) as are research settings where organizational, occupational or institutional boundaries are challenged (Barley and Tolbert, 1997: 104; Park, 1921).

The research setting selected for this study is a social movement that has become a commercial competitor: the free software movement. Supporters of the movement are globally distributed and develop non-proprietary software over the Internet. This movement does not just protest or seek change, but manages the production of commercially valued software in a manner that is consistent with the movement's beliefs. This community can also be considered to be on the periphery of the software industry, since at the time of this study, established firms did not recognize or understand the community's organizing mechanisms. This movement and its spin-off movement, the open source movement, will be discussed in greater detail in Chapter II. Both the context and the charter of this community is consistent with Weick's definition of an 'elsewhere' and fits well with both Fligstein and Dauber's (1989) and Barley and Tolbert's (1997) criteria for likely sources of socially innovative behavior. This is not only a natural setting for the study of the microdynamics of institutional change, but an opportunity to study the interactions of a social movement with commercial entities. In the last five years, software developers that are part of this movement have begun to cooperate with established firms on software development projects.

This research unpacks the process by which challengers (community managed software projects from the free software and open source movements) and defenders (firms in the computer industry that approached these projects) adapted their practices to accommodate each other. It shows how challengers and defenders, in negotiating their interests and rights, shape a new organizational form: a non-profit software foundation. The next chapter discusses this research context in more

detail and Chapter III discusses the research methods used to operationalize this strategy.

Chapter II. Context of Study

The ideal context of study for this set of questions is a challenging group in an industry context with social movement roots. The free software and open source social movements provide such a context as they are grounded in social and political action and can now be considered an industry competitor.⁵ This chapter draws upon preliminary interviews that helped frame the design of this study to explain the history of both movements. A full exposition of methods will be discussed in Chapter III.

A. Origins of the Free Software Movement. Richard Stallman, a prominent member of the software sharing community at MIT in the 1970s, founded the free software movement in 1985 (Stallman, 1999). A social movement is “a set of opinions and beliefs in a population representing preferences for changing some elements of the social structure or reward distribution, or both, of a society” (McCarthy and Zald, 1977: 20). A desire for change has also been interpreted as discontent with the current system. The free software movement desires change in two socioeconomic and legal systems: the software industry and intellectual property laws that restrict the sharing and use of software.

⁵ It is safe to say that the open source movement is a competitor within the software industry as the dominant firm in that industry, Microsoft, has performed competitive analysis on the open source software movement that is now publicly available (www.opensource.org/halloween). This document, released in October of 1998, stated that “to understand how to compete against OSS we must target a process rather than a company”. Furthermore, Microsoft, named the open source movement as a competitor as evidence of competition in their market in their defense of monopoly charges. However, as this form or ‘process’ this group was not well understood, this group can still be considered as one that was on the periphery.

In the early 1980's, when software vendors began restricting the sharing of their code, Stallman protested and refused to comply. He felt that the exchange of information and cooperation was a strong tradition in the hacker community (Levy, 1994) and that the norms in his community would be threatened (Stallman, 1999). He feared that programmers would become dependent upon software vendors. This fear extended not just to the legal restrictions associated with commercial code, but the very nature of the code itself. Vendors could enforce proprietary restrictions by compiling their code into machine-readable language, or binary code, as opposed to providing the source code for their programs. Programmers cannot read binary code, as it is merely a string of 1s and 0s. Source code can be understood by a skilled programmer and fixed or modified. "The rule made by the owners of proprietary software was, if you share with your neighbor, you are a pirate. If you want any changes, beg us to make them." (Stallman, 1999: 54). To avoid this threat of dependence, he decided to build a freely available operating system, the GNU system, a recursive acronym, "GNU is Not Unix", in 1984 and posted a call to action for others to join his project (Stallman, 1999).

Stallman chose the words 'free software' to reflect what he considered basic freedoms: the right to run programs for any purpose, the right to modify programs to suit your needs, the right to redistribute copies either gratis or for a fee, and the right to distribute changes back to the community (1999: 56). The definition of free software does not preclude one from selling it, but it does require that the freedoms outlined by Stallman remain a part of future distributions of that software. Stallman viewed free code as an inalienable right and his decision to develop a free software system a moral choice (Stallman, 1999: 55). He considered this a moral choice as he

did not want to work in a profession with proprietary restrictions. “Another choice, straightforward but unpleasant, was to leave the computer field. That way my skills would not be misused.” (Stallman, 1999: 55). Instead, he created the term, free software, and a set of beliefs associated with it. This can be considered a collective action frame and the precursor to the founding of the free software movement.

Snow et al (1986) theorize that social movements use frames to stimulate engagement, participation, and resources. A collective action frame guides individual interpretation of events (Goffman, 1974) and beliefs into alignment with those of the movement. This facilitates the development of congruent goals between supporters and the movement (Snow et al, 1986). Collective action frames: 1) offer a diagnosis of a problem or sense of injustice (Gamson, 1992); 2) assign blame or causality (Benford and Snow, 2000); 3) offer a prognosis of how the group can effect change, and 4) create an identity or offer a vision of the collective (Gamson, 1992).

The free software frame identified a perceived injustice and offered a solution to combat it, with a call for others to join. New legal terms to guide the use and distribution of free software were an additional component of the collective action frame. Stallman feared that if his free software was in the public domain, it might not remain freely available.

But then I thought about the question of whether people would change the software and make proprietary versions of it. And I realized that if they did that, they could defeat the whole point, they could negate the effort. Someone could make an improved proprietary version and it could displace the free version. And as a result, people might be using my code but they would not have the freedom that I hoped they would have and I would not have it either unless I kept using the inferior free version. *But if nobody joined me, it would not do much good.* And so I decided to look for some way I could stop that from happening. And in discussions with a lawyer I worked out the

idea of copyleft. (*Interview with Richard Stallman, March 20, 2001, emphasis added*)⁶

Stallman had the foresight to realize that if he put his free and open source code in the public domain, it could be appropriated and become closed or made proprietary. This would hamper his ability to create free software that would remain free and might inhibit others from working with him. In order to allow his software to be freely circulated, and yet protected from appropriation, Stallman drafted the GNU General Public License (GNU GPL), using principles he termed “copyleft”. The creation of this license was thus a mobilizing tactic to encourage others to join his project.

This software license is based on copyright law and is written in the form of traditional software licenses. “Copyleft uses copyright law, but flips it over to serve the opposite of its usual purpose: instead of a means of privatizing software, it becomes a means of keeping software free” (Stallman, 1999: 59). The copyleft principle, as codified in the GNU GPL, allows for source code to be freely modified and distributed. However, it requires all of those who use the software to license the source code for derivative works under the same license. The goal of copyright is to restrict unauthorized use. The goal of copyleft is to restrict unauthorized appropriation.

Software in the public domain does not have this protection – it can be modified and made proprietary. Software that is released under the GNU GPL cannot be modified and made proprietary because this self-perpetuating clause

⁶ The names of all informants have been disguised to preserve anonymity except in cases where informants permitted their names to be used.

permanently guides the use of software. Software licenses differ from simple contract law that only binds parties to the agreement and operates more like real estate law, where successive parties are bound to its terms of use (Rosenberg, 2000: 90). Thus, the license remains binding to the software, as opposed to depending on an agreement with users. With this self-perpetuating clause, this license protects the software for the life of the works.

This license is a critical foundation of the free software movement, and the later, open source movement for three reasons. First, it creates a way for communities to collectively pool their efforts and create software without the risk of it being adopted by firms for proprietary purposes⁷.

I wanted to give the community a feeling that it was not a doormat, a feeling that it was not prey to any parasite who would wander along. If you don't use copyleft,..... anybody can come along and say: [speaking very firmly] "Ah, I want to make a non-free version of this. I'll just take it." And, then, of course, they probably make some improvements, those non-free versions might appeal to users, and replace the free versions. And then, what have you accomplished? You've only made a donation to some proprietary software project. And when people see that that's happening, when people see, other people take what I do, and they don't ever give back, it can be demoralizing. (*Interview with Richard Stallman, March 20, 2001*)

The license created a protective legal space in which a community could form and be protected from 'free riders' that did not share their beliefs. Second, it constitutes a legal form of protest against proprietary software that is compatible within the existing legal system. Rather than violate the licenses of commercial vendors by sharing software illegally, Stallman created a new set of rules within an established institution: software licensing. Third, the GNU GPL also provides a normative

mandate for those who use free software to contribute their modifications and improvements back to the communities that manage that code. This helps to codify the hacker community's norm of reciprocity so that others could be made more formally aware of this norm. The GNU GPL is a legal and social innovation inspired by a healthy discontent with the ownership privileges associated with proprietary software designed to mobilize contributors to the GNU operating system project.

The GNU GPL has not been tested in a court of law although legal analysts have drafted legal arguments that might help defend it (Lee, 1999). To further protect the GNU project, Stallman created the Free Software Foundation in 1985, a tax-exempt charity that would support free software development (Stallman, 1999: 60). This foundation is primarily responsible for enforcing the terms of the GNU GPL. The founding of the FSF was a mobilizing tactic to help garner resources and establish credibility for the GNU project.

B. GNU/Linux and the Community Development Model⁸. By 1990, Stallman had made significant progress building the GNU system but was missing the kernel⁹ (Stallman, 1999; Moody, 2001). In 1991, Linus Torvalds, a student at the University of Helsinki, developed a Unix compatible kernel called Linux based on the Minix operating system, a commercial system developed by Andy

⁷ The ability of software projects to protect their work from appropriation will be discussed in more detail in Chapter V.

⁸ I use the word community development model to refer to what is more commonly known as the open source development model due to the fact that the model is used by members of both the free software and open source movements.

⁹ The kernel starts and stops other programs and is responsible for allocating resources to them - a central component to any operating system.

Tanenbaum for educational purposes (Torvalds, 1999; DiBona et al, 2000).¹⁰

Torvalds decided to create a Unix compatible operating system, posted his code on the Internet, and formed a Linux newsgroup (MacCormack and Herman, 2000).

After identifying compatibility between Linux and the GNU system, Torvalds integrated his kernel with components from the GNU project in 1992 and licensed it under the GNU GPL.

Combining the Linux kernel with the GNU system developed by Stallman resulted in a complete and free operating system¹¹ that re-energized the free software movement.¹² From 1991 to 1998, Linux grew from 10,000 lines of code used by one person to 1.5 million lines of code used by 7.5 million people (MacCormack and Herman, 2000). By 1994, it was estimated that over 10,000 CD Roms containing Linux had been distributed (Rosenthal, 1994) with many more copies downloaded directly from the Internet. By 1997, Linux was in use in more than 120 countries (Moody, 1997). In five years, Linux grew from zero market share of the server operating systems market in 1994 to 25% of the market in 1999 (Rosenberg, 2000). While the exact number of contributors to Linux is unknown, it is estimated that over 10,000 people, many of them voluntary, contributed to the code and that Torvalds contributed less than 5% of the total code (MacCormack and Herman, 2000). Understanding how one person coordinated the efforts of so many requires a more thorough understanding of the development model that Torvalds is best

¹⁰ Torvalds and others in the Minix community were frustrated by Tanenbaum's reluctance to let the system expand beyond educational purposes (Tanenbaum, 1999; DiBona et al, 1999; Rosenberg, 1999).

¹¹ It should be noted that Linux was not the first freely available operating system as FreeBSD, OpenBSD, and NetBSD existed before Linux.

¹² The resulting system is called GNU/Linux by members of the free software movement and Linux by the open source movement, most firms and the general public at large. From this point forward I use whichever term informants used.

known for initiating but that other community software projects have since emulated and modified.

Community managed software projects are usually initiated by an individual or group of individuals who start a project on a private or non-profit Internet domain, set up a mailing list, invite others to participate, and grow the project as it attracts more interest.¹³ New contributors (most likely users of the software) may contribute bug reports and improvements by submitting patches and responding to problems and inquiries on project mailing lists. As experience with the project builds, a core team emerges, most often, based on responsiveness and talent. Membership in the core team usually is distinguished by 'commit access' or the ability to commit changes directly to the code base (Behlendorf, 1999; 2000). The code base or repository is hosted on privately donated servers and core members of the group manage access to it. Maintainers of specific packets or modules of code review code contributions submitted by a wider body of contributors and, based on their usefulness and their effect on other components, (Behlendorf, 1999; 2000) may accept or reject them (Raymond, 1999; Behlendorf, 1999; 2000). While periphery contributors may be anyone around the world who is interested in the project, the group selects and manages the rights of core contributors more carefully. A comparison of how this development model compares with traditional software development models is provided in Table II-1.

Traditional software development is sponsored and owned by a firm. Software development projects are often team based with a project manager. While programmers may be internal and external to the firm, the development environment

is protected from public access. Technical requirements are determined based on the firm's analysis of customer and market needs. Applications may be sold directly to consumers in 'shrinkwrapped' boxes or sold bundled with hardware and other professional services to firms.

The open source or community development model is not sponsored by any one firm but typically starts with an individual or small group's idea, the creation of a mailing list and call for participation. A code repository is developed and stored on the Internet. Programmers, many of who may be volunteers, who begin to use and like the software may offer suggestions and start to contribute through project mailing lists. The founding group provides write access to the code base to volunteers they have deemed technically astute and reliable. Technical requirements are determined through the interactions and preferences of those who are developing the code and are not targeted to any one firm's customer or market strategy. When community managed code is released, users are notified through mailing lists and the software is available for download. The group that develops the software is not the same group that sells it, but firms may sell and profit from the community's work as long as they abide by the community's licensing terms and development norms.

This development model was first consciously articulated by Raymond in 1997 and later in his book, "The Cathedral and the Bazaar" (1999). Raymond (1999¹⁴) argues that a fundamental strength of the community development model is its approach to software debugging. He explains, in pragmatic terms, how software

¹³ Corporate initiated open source and free software projects are a more recent phenomenon and may follow a different model. The term community managed project refer to projects that are founded

¹⁴ The essay that was the genesis of the book is located at:
<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>.

developed in the community context can become robust enough to rival commercial grade software without the resources associated with proprietary development. He argues that the number and diversity of people reviewing open source code is its strength: ‘given enough eyeballs, all bugs are shallow’ (Raymond, 1999: 27).

Bugs are argued to be shallow because, when programmers from all over the world have access to the code and are running different applications on different machines, the code can be simultaneously tested at a level that would require many permutations of testing by a single firm. The argument is not that the more people who use the code, the more people will report and fix bugs. Rather, the assumption is that the more people from a variety of different computing environments are involved in running code that they can inspect and modify, the more likely it is that each user will have a unique perspective and detect bugs or problems that others might not see or experience. Greater participation from a large and diverse group of users will enhance the quality of the code. A reformulation of this lay theory might state that; “greater variance in participation will lead to stronger and more resilient code as long as the number of possible permutations of code and computing environments exceeds the number of people participating”.

How does the community development model compare with traditional approaches to software development? The model and Raymond’s theory (1999) is contrary to traditional approaches to software development that find the costs of scaling participation outweigh the benefits (Brooks, 1995). Brooks argues that adding more people to a project, late in its development cycle, may only introduce more bugs and threaten the project. As more bugs are found, more bugs have to be fixed by a limited pool of developers, and the more likely it is that scheduled

deadlines are missed. While this maxim has been tested under varying conditions since it was first formulated in 1975, after surveying the evidence in his latest version of the book (1995), Brooks argues that his law still has merit. An expanded body of contributors associated with the community managed development model would seem to violate Brooks' law.

Brooks argues that although few companies allocate one-half of their projected software project schedule for bug testing and fixing, most actually spend half of their schedule for that purpose (Brooks, 1995: 20). And, in fact, recent evidence (Mockus, Fielding, and Herbsleb, 2000) indicates that the majority of contributors to community projects are more likely to detect and fix bugs than they are to contribute directly to the code's design and functionality. In their study, Mockus, Fielding, & Herbsleb (2000: 6) found that almost all new functionality was implemented by a small core group of programmers. However, almost 400 volunteers contributed to the project over four years. These developers appear to be most critical to problem detection and correction as they generated almost half of the problem reports (Mockus, Fielding, & Herbsleb, 2000). Thus, a large body of non-core developers may be more important in identifying problems than in contributing directly to functionality. The resulting model is partially consistent with Brooks' theory: a small core group of programmers are responsible for the bulk of development with an expanded body of people testing and contributing fixes to the code.

Other computer scientists have advocated approaches to software development that are not necessarily inconsistent with the community development model. For example, Mills, a mathematician noted for bringing more rigor into

software development, noted that “(t)he idea is to make programming a more public practice, under common scrutiny of many team members, rather than a private art” (cited in Brooks, 1995: 299). Mills saw that when work was exposed to more people’s gaze, programmers felt more peer pressure to do things properly the first time and were better able to detect bugs, which helped improve the quality of the code (Mills, 1971; Brooks, 1995). Thus, the community development model is not as antithetical to traditional software practices as might be presumed. However, Mills’ conception of the public practice of programming was much more formalized than the community development model that emerged with the development of Linux. Both Brooks and Mills assumed a common organizational backdrop to software development, whereas the community development model takes place without a common organizational or institutional sponsor. The ability of the community development model to scale; the emergence of a shared understanding of that model; coupled with the ability of the Linux operating system to attract users and developers helped spur what social movement theorists (Snow et al, 1986; Benford and Snow, 2000) would call a frame transformation.

C. Frame Transformation: Creating the Open Source Frame.

Through frame transformation, a social movement can settle disputes about misinterpretations and offer greater clarity (Goffman, 1974; Snow et al, 1986). New values may be instilled, old meanings jettisoned, and erroneous beliefs or “misframings” reframed (Goffman, 1974: 308) in order to garner support and secure participants. As GNU/Linux attracted more contributors and users, Stallman sensed that a faction with friendlier views toward commercial software was growing, but

was unsure of what to name this group or how to handle what he sensed was a growing divide.

And in fact for a couple of years I had been noticing the two different political and philosophical camps in the community and I had been thinking that I should write about this, and thinking about names to give these two camps. I was thinking about calling one the Freedom party and the other, sometimes I called them the Bandwagon party, Success party or Popularity party. So I realized that the term open source created a name for that other camp. And I decided, and it took me about two days to decide this, that now that that camp had made up another name for themselves, what I should do is talk about my disagreement with that. (*Founder, Free Software Movement*)

Collective action frames are not mere aggregations of individual beliefs, but the product of negotiated shared meaning (Gamson, 1992). In the quote above, the founder of the Free Software Movement identified an alternative frame that held a different diagnostic analysis of the social problem they wanted to address. He identifies this group by referring to their emphasis on market share; thus the terms bandwagon, success, and popularity. In his eyes, this growing faction did not see commercial interests as a problem, and in fact, wanted to pursue commercial adoption of their software. Their diagnosis of the problem was spreading the use of non-proprietary software as opposed to eliminating proprietary software. Others also identified this faction as early as 1996.

“That conference [Freely Redistributable Software Conference] was when I first became consciously aware that the leadership role in the culture was passing from RMS [the free software movement founder] and the old MIT AI [artificial intelligence] crowd to the Linux people.” (Open Source Founder in Moody, 2001: 149)

The commercial attention that Linux attracted was having an effect on who hackers looked up to and admired and was thus shifting the status order within the community (Moody, 2001).

This new faction within the free software community, who enjoyed the growth of interest in Linux, was increasingly dissatisfied with the free software frame, which led to what Benford (1993) and Benford and Snow (2000) would call a 'frame dispute'. A frame dispute is a contestation of the diagnoses and prognosis of a frame internal to a social movement (Benford, 1993; Benford and Snow, 2000). This newly formed faction felt that the free software frame might prevent them from seizing new opportunities in the commercial market. One concern was that firms might be confused by the word 'free' as this word could give the impression that free software could not be used to earn a profit¹⁵.

People could not identify with the word 'free software', it was too scary. People needed less confrontation and a less political term. It was dragging us down. A good example is the term free software that sounds really good to an idealistic, shaggy haired hacker in Birkenstocks but scares the crap out of Jay Randa in Techio. We were making basic mistakes like not adopting our language to our target audience. (*A Founder, Open Source Movement and participant in the February and April, 1998 reframing meetings*)

A second concern was that firms would be reluctant to engage in the business of "free software" because of the political agenda and moral philosophy associated with it (Perens, 1999). The dissenting faction within the free software movement wanted

¹⁵ Stallman's conception of free software ensured liberties with respect to accessing, changing, and distributing source code. His intention was not to prohibit people from selling software for a price (Stallman, 1999). In fact, the Free Software Foundation, sells copies of GNU software.

to widen their collective action frame so that more actors, and more varied types of actors would be willing and interested in participating with them.

Not all of the people who started this thing, really believed in free software by the (original) definition....I felt like --- I wanted to dilute the open source definition and ---- very definitely did.” (*An open source founder and former project leader on free software project*)

McAdam (1996) has theorized that new reference publics and target audiences may affect why social movements modify their collective action frames. A new reference public emerged after from the spread of the adoption of Linux and this created a fissure in attitudes toward commercial software that created conditions for a frame dispute. As Benford and Snow (2000) and McAdam’s (1986) research would predict, these two dynamics contributed to a frame transformation: the creation of the open source frame. Social movements use frame transformation in order to gain greater legitimacy and acceptance from conventional audiences when their values and beliefs are antithetical to conventional lifestyles or beliefs (Snow et al, 1986). This frame transformation was motivated by an internal dispute about the resonance of the current frame with a new referent: a commercial audience. It was directed externally; to correct misinterpretations and erroneous beliefs that were held, not by the movement, but by commercial actors.

A group of leaders within the dissenting faction, many of whom had credible reputations in the free software movement, met in February of 1998 to brainstorm how to better market free software to make it more attractive to firms and commercial interests in general (Raymond, 1999; Moody, 2001). They wanted to remove the potential for confusion that the words ‘free software’ were incompatible

with creating a market for it. The new faction wanted to detach the new frame from a frame that held that proprietary software was morally wrong, otherwise how could they court commercial interest? “Since social movements seek to remedy or alter some problematic situation or issues, it follows that directed action is contingent on identification of the source(s) of causality, blame, and/or culpable agents.” (Benford and Snow, 2000). Thus, continuing to attract a new audience was contingent upon recasting culpability. After some debate and consideration of alternate terms, the group chose the term ‘open source’ to emphasize the pragmatic benefits of open and available source code. Attendants at the meeting, some of whom were entrepreneurs and business leaders interested in building commercial businesses using free software, reported that some of the alternative phrases included: freeware, sourceware, and freed software (Moody, 2001).

Informants were very deliberate in transitioning to more commercially minded frame of action and used the word frame to describe what they were trying to do.

I have not been changing the facts on the ground. What I have been doing is creating a new discourse, creating a new narrative, a new frame of discourse about what this all means. (*A Founder, Open Source Movement*)

The new frame absorbed Raymond’s (1999) more pragmatic arguments about the benefits of open source code: namely the accessibility, efficiency, and reliability of open source code development and distribution processes (Stallman, 1999; Perens, 1999; Raymond, 1999) discussed earlier. The moral position of developing free software for the sake of preserving freedoms was abandoned in favor of a logic that extolled open source development techniques as leading to higher quality code.

The new frame makes discontent a less explicit and central motivator for collective action, and in fact, almost disappears all together. The founding principles of freedom are recast into a language more familiar to firms and more conducive to market activity. Table II-2 compares the free software and open source frames according to characteristics of collective action frames identified by Benford and Snow (2000).

Because the new frame's diagnosis of the problem and locus of attribution does not focus on disagreement with the proprietary practices common to the commercial world, it is more elastic or inclusive than the old frame. Proprietary software is no longer held to be immoral, just not as maintainable and correctable as open source software. In the language of informants, 'diluting' the free software frame weakened its narrative fidelity, but enabled it to become more inclusive of diverse attitudes toward commercial interests. Frames that are more inclusive can tolerate greater variance in their interpretive scope and may be better able to mobilize a broader and more diverse group of participants than frames that are more rigid and inelastic (Benford and Snow, 2000; Snow and Benford, 1992).

However, frames that are more rigid and inelastic may maintain greater narrative fidelity, which can affect the degree to which a frame resonates with its founding audience. The free software frame had not varied since its initial construction in 1985. Community members debated the narrative fidelity of the new frame in online forums for several months in early 1998. The central tension was whether the open source software frame, in trying to resonate with commercial audiences, was sacrificing its narrative fidelity. Movement leaders understood that

this debate over the acceptance of the new frame would signal the free software community's stance toward the business world.

This terminological debate is understood by all parties to be a proxy for wider issues about the community's relationship to the business world. (*A Founder , Open Source Movement, www.opensource.org*)

In April of 1998, 18 unofficial community leaders met at a summit and endorsed “the term ‘open source’ and [its] accompanying economics and self interest based arguments” with a vote (www.opensource.org).

The creation of the open source frame led to the founding of the open source movement, or what McAdam (1995) might call a “spin-off” movement. Social movements are not discrete entities; they are embedded in larger families of fields (McAdam and Scott, 2002) that include reference publics (McAdam, 1995). McAdam argues that spin-off movements draw inspiration from initiator movements and modify the framing of the original movement to capitalize on the “ideational, tactical and organizational lessons of earlier movements” (McAdam, 1995: 229). The frame dispute, consequent transformation and creation of the open source movement can be viewed as a spin-off movement that not only had a different diagnosis and more elastic reach, but that strove to avoid what they saw as ‘mistakes’ made by the founding movement that inhibited commercial growth.

We had been trying for years to get our way of doing things more widely adopted in the computer industry and the Fortune 500 and the problem was we weren't speaking the language that those guys could hear. (*A Founder, open source movement*)

The problem of gaining acceptance from a commercial audience was not construed as problematic among those who identified with the free software movement, for they believed that all software should be free, and that combining or integrating free software with proprietary software was harmful to their goal.

With a different diagnosis of the problem and a stronger resonance with commercial audiences, the new frame helped attract and mobilize a larger body of participants than had participated in the free software movement. Between 1998 and 2000, thousands of open source projects were initiated. The number of known open source projects has just about doubled every year. Sourceforge¹⁶ a non-profit site that claims to be the world's largest open source development site, grew from 1 project and 4 users in November 1999 to 11,502 projects a year later and 37,398 projects in April of 2002. As of this writing, over 395,346 developers¹⁷ from over 80 countries were registered on the site (<http://sourceforge.org>). Of the 25,364 projects that reported their license (68%), 95% meet the definition of open source¹⁸, and of those, 73% are licensed under the GNU GPL (<http://sourceforge.org>).

As a result of the frame transformation, the free software community harbors two social movements, the free software and the open source movement.¹⁹ The Free Software Movement maintains its distinct identity and goals, but does not consider itself to rival the Open Source Movement.

¹⁶ Sourceforge is located at <http://www.sourceforge.org>.

¹⁷ Not all of the registered users on Sourceforge.org can assumed to be contributors, so this number also includes supporters and people who are generally interested in open source development.

¹⁸ According to the Open Source Initiative's definition.

¹⁹ (www.gnu.org/philosophy/free-software-for-freedom.html).

The Free Software movement and the Open Source movement are today effectively separate movements, although we can and do work together on practical projects (*Founder, Free Software Movement*²⁰)

What may have prevented deeper balkanization between the two movements is the prognosis they share. The two frames differ in their diagnostic framing of the social problem they are trying to change. However, they share the same prognosis: both frames advocate the creation and dissemination of open source or free software.

That other group of people -- which is called the open source movement -- they only cite the practical benefits. They deny that this is an issue of principle. They deny that people are entitled to the freedom to share with their neighbor and to see what the program's doing and change it if they don't like it. They say, however, that it's a useful thing to let people do that. So they go to companies and say to them, "You know, you might make more money if you let people do this." So, what you can see is that to some extent, they lead people in a similar direction, but for totally different, for fundamentally different, philosophical reasons. (*Founder Free Software Movement, May 29, 2001*)

This shared prognosis provides a broad enough umbrella under which members who identify with either movement can cooperate on software development projects. However, because project members may harbor differing diagnostic frames, they still harbor very different attitudes toward commercial interests and how they affect community interests. This tension is manifested in technical decisions about the boundaries between proprietary and open source code as well as in the design of organizational mechanisms for the project. The tension between these two frames will be discussed as it relates to the findings from this study.

²⁰ (www.gnu.org/philosophy/free-software-for-freedom.html)

D. Market Response to Frame Transformation. As evidence of the resonance that the open source frame had within the software industry, informants reported that within six weeks of the April 1998 vote, the term was in use by industry media.

As an index of how quickly it happened, the whole transition from majority use of free software as a label to majority use of open source happened in the early summer of 1998 in *six weeks (with hand emphasis) flat*. What that showed me is that there was a huge, pent up demand for a more effective, less confrontational way of pitching the message. (*A Founder, Open Source Software Movement, Interview, July 19, 2000, Monterey, CA*)

Six months after the creation of the new frame, the dominant player in the software industry, Microsoft, referred to open source software as a “direct, short-term revenue and platform threat...and a larger developer mindshare threat” (www.opensource.org/halloween). Within a year, it was a commonly accepted term by the mainstream national press.

And I hold the dubious honor of being the first person that used [the word] open source in public with its current meaning. And like a year later it was all over the papers and everything. I never saw anything grow so fast. (*A Founder, Open Source Movement, Former Project Manager, Linux Distribution Project*)

Members in the community were aware that simplifying the concept of open source and its relationship to business potential could help to enhance their publicity.

People had been talking about free software for like 15 years and within 2 or 3 months, everyone was talking about open source. It was much easier for journalists to understand. (*Sponsored Contributor, Founder, Linux*)

Data on the prevalence of articles using the words, 'open source', 'free software', and 'Linux' show that the new frame did indeed receive greater attention from the media. As, Figure II-1 shows, the 'free software' phrase has been mentioned steadily²¹ at a rate of about 30 articles a month over the past eleven years with very little change. The 'open source' phrase receives almost no mentions until 1998, when the graph shows a steep rate of growth from the spring of 1998 until the spring of 1999. 'Open source' dips slightly in 2000, and in 2001, is back up to its 1999 levels. Linux, on the other hand, begins receiving more media attention than free software before the open source term is created, surpassing free software's citation level in 1995 and rising steadily to 278 in 1998. There is a slight inflexion point after 1998, where it rises to a high of 785 article mentions in 1999 and falls thereafter. This demonstrates that, prior to the creation of the open source frame, media attention devoted to Linux escalated several times over the level of attention devoted to free software; lending support for the argument that a new reference public was a catalyst for the frame transformation.

However, market awareness did not equate to market adoption. After a year of media recognition, industry analysts noted in April, 1999 that "(w)e think that we are still at a point where Linux installation is proceeding covertly" (Hambrech & Quist, 1999). Several corporate informants confirmed that they became aware of Linux through programmers who unofficially brought Linux into the workplace. Community based informants also reported installing and using Linux on corporate resources without authorization. In addition to citations in the business press, two

²¹ ABI Inform Proquest database was used to measure the citation count for these key words as this database includes the Wall Street Journal and many of the computer industry periodicals with the widest circulation.

other indicators of the market's response to the frame transformation and the technical success of Linux are the attractiveness of open source software to investors and the rate of adoption of open source software by existing firms.

Only a handful of firms that relied upon open source software as part of their business model were funded between 1992 and 1998. The first firm to base their services on open source software, Cygnus, was founded in 1989 (Tiemann, 1999). The first firm to receive round one venture capital funding that included open source or Linux as a part of their business plan or model appeared in 1992 (Venture Economics, 20001).

There were not many companies having anything to do with free software say in 1995. There were a few. There was Cygnus Support and a couple of other companies like that in other countries. (*Free Software Movement Founder*)

Figure II-2 shows that between 1992 and 1998, another 6 firms received round one funding and that between 1999 and 2001, 43 firms received round one financing (Venture Economics, 2001). This rise in the funding of open source firms cannot be separated from the investment community's wider interests of investing in computing and Internet related companies during this time. Of all firms funded by venture capitalists, the percent of firms that used open source as a part of their business strategy never reaches 0.25% until the boom year of 2000, when the proportion climbs to 0.5%. However, it does illustrate that open source firms were viewed as credible enough to receive funding. There was no indication that firms with free software listed as a part of their business model received funding prior to 1998.

The rate at which existing firms adopted Linux and other open source software applications also indicates commercial acceptance of the open source movement. The installed base of the current Linux operating system is estimated to be about 20 million and projected to double by 2004 (International Data Corporation, 2001). The Linux share of the web server market is about a third and projected to grow to 41% by 2005 (Web Host Industry Review, 2001). Almost two-thirds of public web servers run an open source webserver application, Apache (Netcraft, 2002). However, estimates of the use of open source software are difficult even by the most astute industry analysts as purchasing is not required to indicate adoption of the software.

E. Explaining Mobilization and Growth. The frame transformation may have had greater resonance with commercial actors and helped to attract a broader audience than the free software movement did, but it does not fully explain the voluntary participation, contribution and funding of software without requisite ownership rights. Several types of accounts can help explain the growth in support for the open source movement: 1) Technical; 2) Economic and 3) Social Movement. A cursory review of the arguments offered by these perspectives are examined, not to explain the commercial adoption of Linux and open source software, but to explain participation in the development of open source software itself.

The Internet is, no doubt, an important enabling factor for the growth and sustenance of open source projects as, without it, distributed development would be more difficult to manage. Linux and the open source development model have been used as an example of how electronic networks can reduce the costs of coordination and facilitate more collaborative social and work structures (Malone and Laubacher,

1998; Malone, 1997; Malone, Yates, and Benjamin, 1987). Myers (1994) speculated that the speed and ease of information transfer associated with computer networks might help social movements spread information. Reduced costs of information sharing, dissemination and coordination may have reduced barriers to participating on open source projects, but it does not fully explain why people would devote their time and energy to contribute code to community projects.

From an economic perspective, voluntary contributions to open source software might be subject to what economists call free rider problems (Olson, 1965). Free rider problems occur when the benefits associated with the provision of the good are non-exclusive: they do not flow back solely to the contributors who provide them. It is argued that when the benefits from collective contributions are non-exclusive, rational actors will have inadequate incentives to contribute to such ventures (Olson, 1965). This argument is an important one as it provides a rationale for the public support of public goods. Since the private provision of public goods is subject to free rider problems, private providers may be less likely to invest in public goods. Thus, the good will be under produced and may be better suited for public provision. Yet, in the case of open source software, thousands of volunteers donate their private resources to produce open source software with the knowledge, and even hope, that non-contributors will also benefit from their efforts.

This is a puzzle to economic theorists. One approach to this problem is to identify selective incentives or returns to voluntary contributors of public goods that are not readily apparent. Lerner and Tirole's approach (2000) to explaining the rationale behind contributors to open source projects follows in this tradition. Lerner and Tirole (2000) argue that the primary reason open source contributors

donate their efforts to open source projects is to gain a reputation for their knowledge and skills to improve their career prospects. They identify two delayed rewards that open source programmers may receive: the career concern incentive (job offers, shares in commercial open source based companies or future access to the venture capital markets) and the ego gratification incentive (a desire for peer recognition) (Lerner and Tirole, 2000: 14). This argument draws upon Raymond's analysis (1999) that ego gratification, the status and recognition of having one's code accepted and becoming included on the list of contributors, was a driving incentive for participants on open source projects.

Lerner and Tirole consider these two rewards to constitute a "signaling incentive" (2000:15). Signaling incentives are argued to provide a way for technically skilled programmers to overcome theorized rigidities in the market for technical skill. For example, labor market rigidities could arise from employers who have difficulty evaluating the technical skills of sophisticated programmers due to lack of knowledge of programmers' technical domains. This could be a barrier in connecting otherwise appropriate matches between buyers and sellers. Contributing to open source projects could thus be a way for programmers to overcome such barriers and signal their capabilities to potential employers in the labor market. This account may offer a partial explanation of some of the benefits that can accrue from participation in open source projects, but it neglects important distinctions between the nature of open source and proprietary software.

First, public credits for contributions to open source projects is not sufficient evidence to base the claim that reputation is a primary incentive to work on open source projects. Public recognition may be a benefit of participation, but it cannot

be the primary motivators because many of the core contributors who devoted the most effort worked on these projects long before public awareness of the project emerged. Second, while a few open source leaders have benefited financially from their participation several years after their involvement in open source projects, this is also not sufficient evidence to support a financial based incentive as a primary incentive. This fact, taken alone, does not account for the natural progression of careers. Talented and well-known experts in their field are likely to experience financial and status progression, over time, independent of their work on open source software. Third, in order to argue that career benefits are a primary incentive to participate on open source projects, one would have to demonstrate prior knowledge of the future stream of benefits, which Lerner and Tirole (2000) fail to do.

The only other explanation of voluntary contributions to open source projects considered is altruism – which is quickly dismissed. “The media like to portray the open source community as wanting to help mankind, as it makes a good story” (Lerner and Tirole, 2000: 2). More disturbing, evidence that is contrary to their position is brushed aside, without further analysis.

“An argument often heard in the open source community is that people participate in open source projects because programming is fun and because they want to be “part of a team.” While this argument may contain a grain of truth, it is puzzling as it stands for, it is not clear why programmers who are part of a commercial team could not enjoy the same intellectual challenges and the same team interaction as those engaged in open source development.” (Lerner and Tirole, 2000: 18)

This quote suggests that there may be an important distinction between open source and proprietary software that is unexplored. How do the ends to which proprietary

and open source efforts are directed differ? Answering this question requires a closer examination of how contributors to open source projects conceive of their work and protect their efforts. Fundamental distinctions in ownership and rights between the nature of proprietary and open source software might affect one's inclination to participate in open source projects.

Traditional social movement perspectives suggest that when a group's anticipated way of life is disrupted or there is evidence of structural strain, collective behavior is more likely to appear (Marx and Wood, 1975). A fear of dependence on proprietary vendors and a threat to the hacker way of life was a motivating factor in Stallman's creation of the GNU GPL and the founding of the free software movement. The continued use of this license could also be interpreted to indicate a healthy discontent with the ownership privileges associated with proprietary software. Discontent over the domination of the market by a few proprietary vendors may also have played an important role in the mobilizing the open source community. The founder of Linux speculated that the increasing dominance of a few players in the operating system environment helped motivate people to participate on the Linux software project (Littman, 1999).

One reason Linux has been so successful is [that it's] the underdog...People developed Linux because their interest wasn't getting market share....I think the [market is] in a sick position where people don't have a choice.. You have a huge homogenous market, and you have one entity that controls a large portion of that market. [But] eventually monopolies do not work (Torvalds quoted in Littman, 1999).

During the early stages of both open source and free software movements, discontent with current market conditions may have helped motivate volunteer

efforts to pool their efforts on software development. However, as the community has grown, this explanation is more difficult to defend.

Lakhani et al (2002) have found wide variance in the degree to which community members share discontent with proprietary software as well as variance in the degree to which they feel that discontent is a primary motivating force for participation in the movement. Lakhani et al's survey of Sourceforge project participants indicates that contributors may participate for a variety of reasons: the opportunity to learn from others and enhance their skills; to take part in a community and to enhance their careers (2002).

Analysis of the purposive reframing of free software and the national attention that it received demonstrates the ability of a frame transformation to attract and recruit new supporters. Enhancing the free software frame's elasticity and reducing its narrative fidelity helped attract a greater number of supporters, a greater proportion of which were financially well established and could, in turn, help garner resources. In creating a new language and a new discourse, the new collective action frame helped community managed software projects to gain legitimacy from a new reference public and mobilize participation. Tactical innovations can help challenging groups overcome 'the powerlessness that has confined them to a position of institutionalized political impotence.' (McAdam, 1983:736). McAdam and Clemens (1993) have found that challenging groups have more incentives to develop new techniques to offset their lack of power. In addition to the frame transformation, the open source community used tactics such as a world-wide system

of Linux User Groups, organizing protests and installfests²², establishing self-appointed and paid evangelists and advocates, educating industry analysts and members of the media, creating online publications and their own media outlets, participating in dialogues and discussions on community websites and newsgroups, as well as developing professional marketing tools to attract supporters.

These three perspectives are all likely to be relevant in helping explain participation on open source projects, as neither one alone can fully explain the phenomena. To date, most of the research on open source projects has focused on explaining the rationale behind growing participation in this movement (Lee and Cole, 2000; Lerner and Tirole, 2000; Johnson, 2001; Bessen, 2001; Von Hippel, 2001). Others have examined the historical development of the Linux kernel (Moon and Sproull, 2000), contribution patterns (Lee and Cole, 2000); and how user communities contribute to innovation (Von Hippel, 2001; Tuomi, 2000; Lakhani & Von Hippel, 2000). How distributed volunteers working on community software projects of commercial value manage themselves, their intellectual property, and their interactions with firms has been unexamined.

The focus of this study is not to explain incentives to participate in community managed projects nor the growth in participation and adoption, but to explain the practices and tactics community based projects use to manage their work and their relations with firms. This research examines how community managed

²² An installfest is where members of the Linux community come together to install Linux and other open source programs on people's computers. People new to open source attend installfests and members in the local community voluntarily meet at donated or temporary spaces to load the software on their machines and teach them about open source software.

projects and firms came to collaborate on software development projects. It also unpacks the organizational mechanisms that emerged from their interactions.

This chapter provided some back ground on the free software and open source movements in order to show how one movement spun off from the other; how community software development practices emerged; and how a social movement came to be considered an industry competitor. This environment is crucial to understand the design of this study, particularly because the unit of analysis differs from that of a traditional organizational or industry setting. Chapter III will discuss the methods used to examine emerging relations between firms and community managed projects and Chapter IV will discuss how these two different types of actors offered to explain their desire to collaborate.

Chapter III. Methods

This research was guided by an inductive, qualitative approach using ethnographic methods. Glaser and Strauss argue that the crucial elements of a theory are often best found best with “data on structural conditions, consequences, deviances, norms, processes, patterns and systems” (Glaser and Strauss 1967/99: 18). This method is particularly apt for phenomena that are new, emergent, or poorly understood (Strauss and Corbin, 1990). As both this research question and setting are poorly understood, this method is appropriate. Furthermore, qualitative methods allow room for the unanticipated and are most suitable for grounded theory building – the goal of this research. While there is no specific methodology for grounded theory building, it has three distinct features: theoretical sampling, the making of constant comparisons, and the use of a coding paradigm to ensure conceptual development (Strauss, 1987).

Theoretical sampling guided the selection of four community managed software projects to study. I define community managed software projects as self managed bounded groups that join together over the Internet to produce software under non-proprietary licenses, but do not operate as a firm or produce software for profit. The term community is used in a value neutral manner to indicate unit of analysis without requiring complete agreement on the logics behind collective efforts. It refers to a collective that works toward a common goal with diversity in belief and approach as members may belong to either the free software or the open source movements. Community managed software projects offer what Daft and Lewin would call a midrange theory slice (1993). They are collectively managed, highly distributed, focused on technical problem solving, manage their work through

information technology, do not have easily discernible boundaries, and have in the last two years begun to cooperate with firms. Merton believed that theories at the 'middle range' focused on specific social problems rather than high abstractions derived from a master conceptual scheme (Merton, 1949/61) were better able to advance grounded theory.

Data from an initial set of exploratory interviews helped identify a relevant theoretical consideration for sampling: the degree to which a community project collaborated with or desired association with firms. A range in the degree to which projects were aligned with commercial interests was desired in order to ascertain distinctions that might affect these differences. Based on preliminary interview data and information on the projects available over the Internet, projects were selected along a continuum between close or cordial community/corporate relations and distant community/corporate relations²³. Figure III-1 represents this continuum and Table III – 1 provides a more detailed theoretical rationale for each project. The Webserver project was one of the first to establish formal relations with a Fortune 500 firm while the Linux Distribution Project has opted not to become involved in commercial distributions and has struggled to retain that distinction. Detailed information about the attributes of each project is provided in Table III-2.

Inductive, interpretative approaches benefit from variance in data sources as potential biases of people and methods are more apt to be reconciled when underlying causes of variation are discovered and examined (Glaser and Strauss

²³ No attempt was made to survey the internal consensus of members beliefs toward commercial activity. A project's stance toward commercial activity was based on 1) the type of license they used; 2) the degree and number of firms that the project was working with and 3) attitudes toward commercial firms drawn from a sample of members on the projects.

1967/99: 68). Multiple sources of information enable triangulation and validation of theoretical constructs that withstand analysis from varying perspectives. Data was collected from three primary sources: 1) observation at user group meetings, technical presentations and conferences; 2) informant interviews, and 3) project data archived on the Internet that detailed the project's interactions and structural developments. Over 90 hours were spent observing and meeting informants at 27 different events (project meetings, user group meetings and conferences) between April, 2000 and April, 2001. Seventy semi-structured interviews were conducted with members within the open source and free software communities. Most (84%) were conducted in person, although some (16%) were conducted over the phone.

Interviews were conducted using ethnographic methods (Spradley, 1979). During an ethnographic interview, the interviewer maintains native strangeness and attempts to understand the terms and meanings that are most salient to informants. Informants were assured of anonymity and interviews were taped when permission was granted²⁴. In some cases, informants allowed their names to be used, but anonymity was maintained in the writing of the study. The structure of interviews varied according to the role of the informant. Gaining an understanding of the informant's introduction to the project and subsequent role was a starting point for all interviews. Individual employment relationships and project experiences were followed by discussions of "how things were done on projects" and how project (or firm) practices had changed or been affected by the rise in commercial interests in their work. Membership, sponsorship, decision-making, and the governance of projects were important foci of the interviews.

²⁴ Permission was granted to tape interviews in all cases.

Fifty-six percent of informants were identified through face-to-face meetings and events, while others were identified directly through online documentation of their project or from referrals from other informants. About two-thirds of the respondents could be identified as having a corporate sponsor. In the sample of 70 informants, 8 of the 53 open source companies (15%) that received funds from venture capitalists were represented. In addition, informants from three Fortune 500 firms that were dominant players in their industries took part in the study. Establishing a representative sample of the community cannot be determined as the demographics of the community are largely unknown, nor is it a requirement of this research.

However, the centrality of these actors and the esteem that is held for them by others in their community can be ascertained through their evaluations of each other. A few leaders in the free software community created a 'trust network' that allows members to evaluate each other's reputation and technical skill²⁵. Members (not observers) can rank each other as an Apprentice, Journeyer, or Master. It should be noted that these ratings are highly subjective and may not be related to project performance. Informants indicated that rankings could be based on demonstrated technical skill, affinity, and/or the perceived amount of contributions to the community. A minimum of 40% of the informants participated²⁶ in this network and of those that participated, more than half of them were at the Master

²⁵ This community is publicly available (www.advogato.org), but you must be a registered user to evaluate someone. The metric is adjusted so that the opinions of those at a higher level are weighted more than those at lower levels. The network also has some unique properties that prohibit attacks on a person's ratings.

²⁶ Only 41% of informants could be identified by their names in the trust network database. It is possible that there are more participants in this network as people will occasionally use nicknames or well known user logon names. Thus, those using aliases would not have been identified and the proportion participating could be potentially larger.

level and another third were at the Journeyer level. This indicates that a large portion of informants was well regarded by their peers. Detailed information about the attributes of the informants is provided in Table III-3. The selection of informants was not guided by this network, but by capturing the perspectives of as many different types of actors as possible. The perspectives of six different types of actors involved in community projects were captured:

1. **Volunteer contributors to community managed projects** - contributors that work on community managed software projects on their own free time.
2. **Sponsored contributors to community managed software projects** - contributors that work for firms but are authorized to work on open source or free software projects as part of their employment.
3. **Representatives from open source and free software firms founded by community members** - founders and employees of open source and free software firms that are distinct from firms founded on proprietary principles.
4. **Representatives from established firms that sponsor participation in community managed projects** - representatives from Fortune 500 firms that sponsor participation in community software projects and are important contributors of talent and resources.
5. **Representatives from mediating open source and free software institutions** – representatives from organizations that protect and promote the concept, words, and rights associated with open source and free software.
6. **Supporters and organizers of the free software and open source movements** – who may be active in institution building but not active code contributors to projects

Obtaining the perspective of the six different actors was critical to understanding how different motivations, interests, contributions, and roles affected interactions between communities and firms. Community managed projects provide a point for the convergence of all of these actors within the community.

Project data was also collected from online archives and included documents such as: mission statements, charters, bylaws, meeting minutes, and mailing list archives as well as a database of developer/members where available²⁷. Project data examined included: 1) project origins, 2) demographics of contributing programmers (where available), 3) development contribution patterns, 4) technical progression of the project, 5) organization of the project, 6) institutional status and financial backing, and 7) licensing and distribution terms. Ethnographic methods guided the examination of project documents with a special effort to capture the perspectives, actions, and interactions of all actors involved.

Interviews were transcribed and then, with other project data, coded and analyzed using Atlas TI software, a qualitative coding application. Over 54,000 lines of text were coded to identify common practices and tactics used on community projects. Where identified, variance between projects was compared with other project attributes to develop an understanding of sources of differentiation as well as similarity. The variance inherent in different types of actors and their roles facilitated the process of constant comparison and the development of codes and constructs. I created tables to indicate where specific practices and tactics replicated across projects and did the same in comparing attributes of the organizational structures that each project created. I generated conceptual categories based on the data and then incorporated evidence into the text from which the category emerged to illustrate the concepts (Glaser and Strauss, 1967/99). The roles and relationships of

²⁷ This community is technically very astute and aware of the ease at which data, once collected, can be misappropriated. Most are thus very sensitive about collecting data that is not necessary. Minimal information about participants in community projects is maintained and data mining in general is frowned upon. As a researcher in the community, I did not collect any data that was not available to the participants themselves.

informants on specific projects are provided with their quotes except in cases where it might reveal an informant's identity.

The validity test for this type of study is quite different from that of statistical studies where the phenomena of study have been operationalized as specific variables. As opposed to evaluating statistical validity, one way to evaluate grounded theory is by its internal validity and by its ability to fit the results into a model that survives scrutiny and provides a closer fit between the model and the social context on which it was based (Glaser and Strauss, 1967/99). Replication within and across cases is one way to verify the extent to which findings apply (Yin, 1981). I have indicated where findings do not fit with the model and offered possible explanations. In assessing replication within and across cases, the focus was not on counting the frequency of the phenomena, but in interpreting and decoding how specific phenomena contributed to the relations under study. Strauss and Corbin suggest that a qualitative study should fit the substantive area without forcing (Glaser and Strauss 1967/99); be comprehensible to readers and those studied; and be broad enough to be applicable to a variety of contexts related to the phenomena (1990). These criteria guided the development of text from coded constructs.

Chapter IV. Different Actors and Logics for Interaction

Chapter II detailed how members within the free software movement consciously developed a new frame to de-emphasize their political and moral foundations and replace them with more pragmatic arguments about the benefits of their software. Transforming the free software frame was considered the first intentioned change the free software community made to be more welcoming to commercial participation. It may have provided a context more conducive to cooperation, but it does not explain why community managed projects and firms, two very different types of actors with very different values, would want to work with each other. This chapter will discuss why members in the movement courted commercial interests and why firms were interested in working with community managed projects. Both commercial and community informants offered logics to explain their interests in collaborating, however differences between these two types of actors could make interactions difficult.

A. Community Logics for Commercial Participation. Why did community projects want to work with firms? Informants who had participated in the creation of the open source frame spoke of their respect for the free software movement and their agreement with its mission, but questioned the degree to which they agreed with its political and philosophical agenda.

And I actually believe a lot of the same things as [the Free Software Movement Founder]. The only place where he and I disagree is that I feel that free software and proprietary software should get along. And he wants all software to be free. (*Former Project Leader, Linux Distribution Project*)

The new frame signaled a new receptivity toward commercial participation in community software projects; that they could 'get along'. Some did not feel as though this receptivity was necessarily new, but that it had been a latent sentiment within the movement all along, but only recently articulated.

Well actually I would contend that, [the project] from the beginning had people that were working on it because they had some website they were building that they were responsible for and that they probably pulled down a living being responsible for – so you know it has made a lot of sense. People like to think about open source as just this hobbyist kind of thing but actually since the beginning, there have been a significant number of people who have done it because it was a better way to accomplish some commercial task. (*A Founder, Webservice Project*)

As Lakhani et al's study (2002) recently confirms, programmers who had contributed to and worked on community projects for many years often benefited from this participation in a multitude of ways: improving their skills and learning new techniques that helped them solve technical problems at home as well as on the job. The informant in the quote above is striving to make a distinction between his project and the perception that it was merely a hobby. He uses the word hobby in a derogatory fashion, primarily because to him, the word hobby reflects a lack of purpose or problem solving. He emphasizes that the project's work is purposeful and directed toward solving real, and in some cases, commercial, problems. What he and other informants actually described was the way in which their work on community managed projects overlapped between hobby and work domains.

Working together and creating source code in a visible and public manner helped people solve technical problems that crossed personal and professional lines. Programmers learned from each other and practiced honing skills that were also used in the workplace. Informants reported that both the use of source code and the

openness of the development process enhanced learning and the development of technical skills. Closed code and proprietary restrictions made the job of programmers more difficult by making bug detection and fixing more problematical. Programmers shared Stallman's frustration with closed source code. Many felt that the free exchange of open code helped them to learn, build their skills, and solve technical problems that applied to both personal and professional realms. Open source code helped programmers overcome a potential threat to their occupation.

However, approval of the free software movement's mission did not always extend to agreement on the beliefs that proprietary software was immoral or that it should not exist at all.

I don't think the commercial side of Linux has ever been viewed as a bad thing. (*Founder, Linux Distribution Project*)

Informants who identified with the open source frame recognized that community projects could benefit from commercial participation. The benefit that informants talked about most often was the ability to increase the relevance of their work to wider audiences and to improve its corresponding market share. The free software movement founder who had first noticed this value and interest in growth or market share, used the words 'popular' or 'success' to label this new group. In reality this value was more complex. A goal of the open source movement was not just to enhance growth and market share, but also to expand to a different type of audience, one that was less technical than they. Informants explained that the coding they did, they did for themselves. Software produced by community managed projects is designed by a technical community for a technical audience. It may thus require greater skill to install and use than commercially produced software that is designed

to appeal to a wider, and thus more varied skill base. Because community managed software is more difficult to use, it may not appeal directly to a large population of non-technical consumers.

And that is why today, the most successful open source projects have been those for developers. Operating systems, webservers, rather than end user applications. That is not to say, open source won't work [for applications], it is just we need feedback loops. We need ways for the people writing the code to be motivated because they are solving someone else's problem – and that is done by those developers being hired by a company. Or it is done through a real efficient marketplace – where you know it is money in exchange for going in a specific direction. (*A Project Founder, Webservice Project*)

Developing software that was easier for non-technical audiences to use could help community managed projects gain greater adoption, but some informants also recognized that most volunteer participants would not be motivated to do that. Firms could provide feedback and incentives for them to design and program code that was more accommodating to a broader base of users.

Collaboration with firms could thus help in two ways: providing financial incentives to work on areas that were not of direct interest to volunteers and to provide assistance in developing better interfaces that were designed for the consumer market. Informants who identified with the open source frame saw firms as offering skills, attention and resources that could help bridge the gap between technical and commercial software users.

Informant: The thing is the commercial side tends to pick up different things than the community side.

Interviewer: Technical features?

Informant: Well, they tend to concentrate on different technical features too but most of the commercial stuff tends to be centered on supporting

customers or doing the kind of stuff that non-commercial people really aren't interested in...Which I think is wonderful and I think one of the reasons we have worked really well together. (*Project Founder, Linux*)

Commercial attention to areas of software development that the community neglected or was less interested in could benefit the project's technical development and market reach. Thus, for those who were most aligned with the open source frame, collaboration with firms did not necessarily threaten project goals, but was viewed as a way to help further the progress and adoption of community managed software.

A second benefit that stemmed from collaboration with firms was the opportunity to work on new problems in new computing environments. Firms that brought their problems to project members widened the range of technical challenge that community members were exposed to, and in the process, allowed individuals to test and strengthen the range of their skills and knowledge.

Informant: The only kind of overlap there is, which is the good kind of overlap is that commercial interests brought in a lot of problems that did not use to be there.

Interviewer: Brought in problems?

Informant: Oh yeah, they had a lot of new interesting technical problems like what do you do with terabyte disks and large scale clustering? Things that much technical people are kind of interested in but they never get to actually play with. For example, there are a lot of people who are interested in doing performance work on extreme loads and the only place where that actually happens is the commercial setting. (*Project Founder, Linux*)

Lastly, the most obvious benefit to be gained from commercial participation on community projects was financial and in-kind support. While informants emphasized that their projects were not resource intensive, as projects grew, their technical

environments necessarily needed to become more sophisticated. Firms could offer servers, hosting and other services needed to support global development at a larger scale.

It's [working with firms] a necessity. No matter what anybody says, it was never really the idea, it was never going to be like, we're going to be the rebels, we don't like big business. (*Volunteer Contributor, GUI Desktop Project*)

Informants were pragmatic and sanguine about the gains projects could make from working with firms. At the same time, they did not forget that most firms in their industry depended on proprietary software to earn profits and that this would affect their motivations within the community. They were welcoming, but circumspect.

We write software as an invitation for other people to write free software. And we want them to join the team, not just sort of suck it. I had this scale that I use at talks that goes: Benefactor, symbiote, parasite. NASA is a benefactor. They sponsored [---] who developed most of the Ethernet card drivers in Linux. That was the US government. That is what Microsoft now wants to lobby about ok? Red Hat is a symbiote, they make a contribution, a very important contribution and they also take money out. Ok? Parasites, without naming any names, a parasite would take the software and make as much as they can and return nothing. (*Former Project Leader, Linux Distribution Project*)

B. Commercial Logics for Community Participation. Despite a more market minded frame, there are several reasons why participating in community projects²⁸ could still be unattractive to firms. Three of the four projects examined in this research licensed their software under terms that restricted

²⁸ Participation in a community project differs from adoption of open source software, as participation requires contributing code, ideas or resources to the project.

proprietary use of their work²⁹. The economic arguments used to explain individual participation on community projects discussed in Chapter II also apply to firms. Any investment of a firm's time and resources in a community project would have to be shared with the community. Thus, firm resources would be devoted to collective efforts where the gains would not flow solely back to them, but be shared with other potential free riders.

What incentives did firms have for participating in community projects?

One senior executive reported that his firm's initial reaction to the popularity of open source software was consistent with traditional economic logic.

[Our] corporate strategy types recognized that open source could be, or I think it is probably fair to say that they primarily recognized it as a major threat to (us). A few of the more visionary ones thought it might also be an opportunity, but it was largely feared in that we make money selling software and the executives believed then, and I think their mind has been changed a fair amount since, that software source code was inherently incredibly valuable and the way you made money was to take this incredibly valuable stuff and keep it secret and make people pay to get to use it. Which is the old fashioned approach to software. And these executives are old fashioned.
(*Senior Executive, Fortune 500 company, Contributor to several open source projects*)

Advocates of open source and free software inside firms tried to gain support for working with community projects by reducing the perceived threat they posed to the firm. Informants from firms also recognized the same distinction that volunteers made: that the strengths and weaknesses of community and firm developed software were complementary. This helped corporate decision makers to reconsider how open source projects related to their business models.

²⁹ One of the projects uses a license that is similar to the Berkeley Distribution License and allows proprietary extensions of their work, provided that credit to project members is included.

The closer to the operating system the less money we make, and the more willing people are to use it [open source/free software]. The closer you are to applications, where we make enormous amounts of money, the less people are willing to use it [open source/free software]. So we could tell them honestly that while we believed open source is going to be a big transition, and in fact a paradigm shift – that is going to change all sorts of things that it was not going to threaten the places where we make a lot of money in the near term. (*Senior Executive, Fortune 500 company that now contributes to several community projects*)

The weakness of community managed software in consumer markets helped to alleviate the concerns of firms whose profits derived from consumer applications. This helped firms understand where commercial interests could make a contribution to community development that would not significantly wrench their existing business models. Some informants from software firms considered their work on open source projects to be an important platform to their proprietary work. Three of the four projects in this study were very close to the infrastructure as opposed to the application layer. The GUI Desktop Project was the only project at the application layer. This perceived reduction in the competitive threat posed by community managed projects helps explain how collaboration might have become more feasible, but does not fully explain why proprietary alternatives were not more attractive.

From an economic perspective, participating in community projects would be irrational if a firm depended upon the ownership of software for profits. Some of the firms that approached these four projects were firms with revenue streams that did not solely depend on the sale of software, but also on hardware and services. In these cases, community managed software could reduce their expenses and greatly enhance their project margins. They could sell hardware and services bundled with community managed software without paying the licensing fees associated with

proprietary software. However, these firms had business units that did depend on the sale of software for profit. Thus, this account does not fully explain firm interest in community projects.

Another explanation is that firm participation was not fully authorized or sanctioned and that its introduction into corporate business models was emergent and rationalized ex ante. Some long standing community members worked in firms and reported unofficially using or contributing to open source and free software without the firm's knowledge. Several executive informants reported of 'skunkworks' projects that they learned of after the fact. One executive at a Fortune 500 company sponsored a Linux Users Group meeting in his hometown to assess the degree of local interest. He was surprised to find several of his own top information technology people in attendance and to learn that Linux and other free software and open source variants were in use within his firm. He decided that if his staff were passionate enough to attend a meeting about Linux outside of work, then he would have to learn more about it and take it more seriously.

If the motivation for firms to participate in community projects is not immediately obvious to an outsider, it was also not immediately clear to firms. Aware that employees were participating in community projects without an official policy, another Fortune 500 firm called for a moratorium on open source software until the rationale for participation was fully developed.

Informant: [--] said you are not doing any open source until you can tell me why.

Interviewer: So sort of ah freeze? For the group?

Informant: For the whole corporation, yeah. So the motivations, were maybe not, well were completely understandable. (*Senior Executive, Fortune 500 company*)

Informants in senior positions at firms that contributed to the projects examined here offered several logics to explain their interest. The most popular explanation offered was the attraction of the market share associated with these projects.

Informant: And so they [Fortune 500 decision makers] said this is a complete sales and marketing decision.

Interviewer: Based on the number of users?

Informant: Right, completely a sales and marketing decision. This has got the market behind it and if we don't go with it? And [if we do] we look good in the open source eyes, which at that time, was kind of a visionary thing to be able to say and then it was you know, not only do we look good, but gee take a look at these hundreds of thousands of you know millions of customers that we immediately get and you know it was a no brainer for [Fortune 500 firm] (*Former Fortune 500 employee, Sponsored Contributor, Webserver Project*)

Three out of the four projects examined in this study were mature³⁰. One in particular held the majority portion of a major industry market. Two others held significant but not threatening portions within the Linux operating system market. A fourth was a standard setting project that did not have a product that would necessarily claim market share, but that was vital to other distributions in the Linux operating system market.

Firms also viewed the user base associated with these projects as a readily available source of ideas and innovation that they could tap for their own purposes. Participation in community projects provided proximity to the project's global user

³⁰ At the time of this study, the age of the projects ranged from four to eight years.

base and to their feedback cycle of improvements (Von Hippel, 2001). Benefits from such affiliation could be significant (Von Hippel, 2001; Constant, Sproull, & Kiesler, 1997). A sophisticated, global user base was also more likely to have diverse skills and computing environments. This range of feedback could help firms improve the portability and interoperability of their products (Raymond, 1999) and enhance their market share.

This is all, open source is just a methodology for doing great product management. Our product is an open source product. I now have the email addresses of 1,600 of my customers, and I know exactly what they say. (*Fortune 500 Product Manager*)

Informants from firms that approached the projects in this study also reported that an important incentive for doing so was to gain a competitive advantage in an industry that was dominated by a monopoly. The desire to create more choice and to reduce the market share of a federally declared monopoly in the software industry was an important motivator for firms as well as individuals to contribute to community projects. This motivation was spoken of in terms that were both philosophical and strategic.

Informant: The fact is that nobody has figured out how they are going to make money off of this stuff yet.

Interviewer: Including [your company]?

Informant: Including [my company], including [Fortune 500 company] all these guys but they are jumping in like crazy

Interviewer: How do you explain that?

Informant: Again I think it is because they are looking at the alternative, the alternative is funding all of their money to Redmond, that is really the only alternative..... the ultimate competitor would always be Microsoft. I think

what I am seeing is that their only chance to make money is to break the control or the lock that Microsoft has on the economics of personal computing. And so they are doing it any way they can. They have chosen Linux and they are running it. And it is almost like they are running with it and saying we will figure out how to make money on this later, we just have to get to the point where it is possible. (*Sponsored Contributor, Standard Setting Project*)

Firms also perceived that they could gain a temporal advantage from the transparent and public development process. Firms that joined community managed projects gained access to a fast paced public development process supported by highly skilled programmers. The visibility of software development on these projects offered firms access to critical code developments earlier than software developed in a closed, proprietary environment. Proprietary alternatives often have longer release cycles with fewer updates in between releases and are less transparent in their development process as well as in their code. Informants did not agree on whether technical superiority of community projects played a role in the decision for firms to participate on open source projects. However, the transparency offered by both the source code and the open source development process, may have made community managed software easier to access, evaluate, modify, and maintain than proprietary alternatives (Raymond, 1999).

Commercial informants also reported that the talent of those who were managing community development work was a motivator for them. Some contributing members of the four community managed projects in this study are also internationally well regarded in their field for their technical skill and are important figures in the media whose opinions are sought by industry analysts. A few sit on national technical standard setting boards and committees. Firm decision makers viewed participation on community projects as a way to not only access technical

experts to help solve their problems, but also to raise their own status in the technical realm.

Ninety-five percent of the code that gets written [by our firm] gets committed to a public repository you know. Not just that we publish stuff as open source once a quarter or something like that, but [the fact] that it is instantly out there in the open is incredibly powerful as an employee happiness kind of issue, a morale kind of issue. They [employees] know that even that if they were to leave the company, they could still play with this fun code. (*Open Source Firm Founder*)

Association with talented community software maintainers offered firms a benefit that they, in turn, could use to attract new talent and retain existing talent. Working on code that was not proprietary allowed programmers to develop skills in an area that would not be specific to a firm and that would always be available to them.

Firms who wanted to participate in community managed projects rationalized that they were trading the ownership of software as a source of competitive advantage for other benefits: the ability to tap and recruit top technical talent, to access a large, pre-established market and user base, to gain a temporal advantage in software development, and to reduce the market share of a monopoly in the software industry. Commercial informants reported that these logics took time to develop as the rationale for participating was initially unclear even for firms who wanted to support such actions.

It is possible that without the frame transformation, these logics may not have emerged in quite the same way. Bansal and Roth found that issue salience was a strong determinant of a firm's responsiveness to environmental issues (2000). The creation of the open source frame and the media attention it generated enhanced the salience of community managed software for firms. However, when firms started to

actively work with community managed projects, their ability to incorporate community managed software into their existing business models and make a profit was still undetermined. In Bansal and Roth's (2000) study of why companies respond to environmental concerns, they identified several reasons why firms to respond to environmental concerns, but note that the motivations of firms are often mixed. Industry analysts found the dual strategies for managing open source and proprietary software pursued by some of the top competitors in the computer industry to appear contradictory (Economist, 10/11/01). However, the logics for participating in community managed software projects were compelling enough to motivate several Fortune 500 and several smaller firms to engage with the four projects examined as a part of this study.³¹

C. Differences and Interaction Difficulties. Firms and community projects held different logics for wanting to work with each other. Divergent logics for participation were but one reason why interactions between community projects and firms could be difficult. The most theoretically interesting challenge for firms was learning to deal with a new type of actor. Community projects were collections of individuals that were not legally bound in any way. These two actors were not only not accustomed to cooperating with each other on the development of software, but had no way of legally recognizing each other. For example, a firm that wanted to enter into formal agreements was confounded by how to interact with the Webserver Project. This firm wanted to establish common understandings: ground

³¹ All four projects have some involvement from one or more Fortune 500 firms as well as from several smaller firms. Involvement could include: the support of hardware or technical services, direct donations, donations of code or other intellectual property, or the sponsorship of contributing programmers.

rules for participation and for the transference of intellectual property. This was, after all, how they were used to working with other firms.

Informant #1: We all set around and Jim goes “how am I going to do a deal with a webpage? And I said well we will figure something out. (*Fortune 500 firm representative*)

Interviewer: How do I do a deal with a webpage?

Informant #2: An amorphous group of individuals who don't have any legal status or any way to make commitments, which is implicit in the idea of a deal. (*A Project Founder, Webservice Project*)

Our legal and economic institutions do not recognize unincorporated collectives as legitimate actors in our commercial system. Thus, community managed projects could not enter into agreements as collectives, only as individuals.

Communities are also not eligible for a host of other privileges that corporations enjoy, such as protection from individual liability. Coleman considered a corporation to be a juristic person, a person that does not exist naturally, but can be found only in law (1974). Coleman argued that in order to gain the benefits of a corporate actor, collectives must give up rights, resources and power to form the corporate body. In turning over these rights to a corporate entity, individuals lose some degree of effective and immediate control over them. In return for giving up their rights, individuals receive the implicit agreement that greater benefits will arise from association with the corporation (1974: 42-43). Coleman's concern was how individuals could benefit from the legal privileges extended to corporations while maintaining control of their rights.

Charters and bylaws that offer voting rights are one way that members' interests can be realigned with the new corporate body. Corporate interests however

become larger than the sum of its members as a new set of interests emerge – corporate survival. Thus, Coleman argues, juristic persons gain power at the expense of natural persons. His greatest concern was that the power held by natural persons would “continually decrease with a corresponding increase in that held by corporate actors” (Coleman, 1974: 37). Coleman predicted that individual-corporate relations could be problematic because corporations’ size, legal privileges, and access to information would lead to asymmetries of power. Thus, community projects and firms might have trouble cooperating simply because of the vast differences in their respective power and rights, not to mention differences in economic power and motive.

Community project-firm relations might also be hampered by the simple fact that both natural persons and juristic persons prefer to interact with their own kind. Legal entities prefer to engage with other legal entities. A corporate actor’s frustration with how to interact with non-legal entities is expressed in his frustration with how to “make a deal with a web page?” For example, the media did not know how to treat this new form, although they did recognize that ‘the community’ was not only a recognizable entity but also, a viable force in commercial markets.

Infoworld, a popular computing magazine, announced in 1998, that they were awarding the Best Technical Support Award to the Linux user community (Foster, 1998). Readers in the December issue of the magazine reported that they got more answers to their questions and more support overall from mailing lists, postings, Linux webpages, and emailing developers directly than they did from commercial vendors who sold support (Foster, 1998). Perplexed as to how to make such an award to such an undefinable entity, *Infoworld* stated that “this year marks the first

time we're going to have to figure out just what our winner's name is" (Foster, 1998). They settled on the Linux user community.

Gaining legal recognition and protection from individual liability was an important incentive for community managed projects to incorporate. In the example below, a firm's transference of intellectual property rights to a project is left "murky" because this project was, as of yet, unincorporated.

(R)egarding [Fortune 500 firm Theta]'s statement about assigning various code to "[the project]." Because this was announced before the official announcement of the [Foundation], the actual recipient of the code was left "murky." (*Websaver Board Meeting, June 22, 1999*)

Community managed projects and firms faced two different types of challenges in cooperating with each other. First, they were motivated by different logics and could hold very different values with regard to proprietary software. While the open source frame was more cordial to commercial interests, it still embraced principles that were antithetical to the industry's accepted institution of proprietary work as the basis of competitive advantage. Second, community projects and corporations have asymmetric power and rights due to their very different legal status.

The next two chapters, Chapter V and VI, detail how community projects and firms adapted their practices to facilitate working with each other and explains how they also resisted becoming more like each other on critical dimensions.

Chapter V. Community Managed Projects

The differences in logic and form discussed in Chapter IV could have made collaboration between firms and community managed projects prohibitively difficult, if it were not for the fact that both firms and community projects changed some of their work practices, organizing structures and policies to accommodate these differences. Changes of accommodation were, however, met by a competing tension. Firms and community managed projects, although willing to make some concessions in order to better cooperate, also used tactics of preservation to ensure that accommodations did not threaten their core identities or objectives. Firms used tactics to preserve their market power by influencing the technical direction and management of community projects. Community managed projects resisted such efforts with tactics to preserve their technical autonomy and prevent the infiltration of rationalistic controls.

The word practice is used to refer to the social doing that is customary within a community of practice (Wenger, 1998). Practices may be implicit or explicit, but they are an “unmistakable sign of membership in communities of practice” and are crucial to the success of their enterprise (Wenger, 1998: 47). This study focuses on the community’s organizing and governance practices, as opposed to software development practices. Unfortunately, diachronic data is not available on these changes, but informants relayed changes in their practices that happened between the summer of 1998 and April of 2001. I have included date information when it was possible and when it would enhance theoretical development.

The firms discussed here are those that approached or worked with the four community managed projects examined as a part of this research. Of the 70 informants, about 1/3 were volunteers and 2/3 were sponsored by either a Fortune 500 or open source firm. Sponsored contributors represented three different Fortune 500 firms and eighteen different open source firms. This chapter discusses ways in which community managed projects changed to accommodate a commercial interest and pursued tactics to preserve their values and interests when they diverged. Chapter VI examines firms in the same manner.

A. Accommodation

In addition to creating a frame that more explicitly encouraged commercial interests, community managed projects incorporated, allowed firms to begin to work with them, and developed formal mechanisms to represent their projects to the outside world.

1. Incorporation. In Coleman's analysis of the rise of corporate power, he acknowledges that one way for individuals to reassert power over corporate actors is to create "countervailing corporate actors to offset the power of existing ones." (1974: 73). Challenging groups are usually not in power, and a difference in form can exacerbate that imbalance. However, as Clemens found (1993), challenging groups may be more inclined to innovate in their tactics and forms of organizing in order to circumvent imbalances in power. All four community projects incorporated as non-profit organizations³² within three to five years of their founding.³³

³² The Webservice and Linux Distribution Projects have received their 501c(3) nonprofit status, the GUI Desktop Project is awaiting approval, and the Standard Setting Project is an approved 501(6) organization.

In their evaluation of the unique properties of the non-profit form, DiMaggio and Anheier (1990) suggest that non-profit organizations signal trust worthiness, primarily because their non-distribution constraint proscribes distributing net income (Hansmann, 1980) and because they typically pursue goals that differ from firms. The special tax exemptions that the United States government grants to non-profit organizations rest upon the assumption that non-profits behave differently from firms (Powell and Clemens, 1998). Non-profit organizations may be more appealing when certain goods are seen as inappropriate for market exchange or requiring special protection from corruption by the profit motive (DiMaggio and Anheier 1990: 144). DiMaggio and Anheier also suggest that non profit organizations may also be more conducive to professional autonomy as their charters may mirror professional ideologies more closely than firms (1990: 142).

Informants reported several reasons for incorporation that, at first glance appear to be more pragmatic, but not inconsistent with these theoretical explanations for non-profit corporations. Contributors to community projects wanted to establish a permanent institution so that projects could live beyond the efforts of their founders. The project with the largest market share, the Webserver Project also wanted to offer greater stability for their customers.

It is a valuable project. Most of us will stay involved in it forever. Also our customers need stability. [The Webserver Project] had to be seen as an ongoing group and I think that making it a legal entity was the way to do it.
(*A Project Founder, Webserver Project*)

The permanence of a legal entity would also enhance the legitimacy of projects in the marketplace. Informants wanted the same protections and privileges afforded to corporations: protection from individual liability if there were flaws found in their software. They wanted to be able to enter into agreements collectively and to protect their work, their trademarks, licenses, and their brand.

I thought that it [the corporation] would own trademarks including the open source trademark. A very big concern at that time was liability protection. And one of the ways you protect individual developers from liability is you shield them behind a corporation.” (*Volunteer Contributor, Linux Distribution Project*)

We’re developing software and writing documents. Either we hold copyrights as individuals or we have a company holding copyrights. There’s liability protection that comes from being a corporation. (*Sponsored Contributor, Standard Setting Project*)

Community managed software is freely available, but that does not mean that it cannot be used without restriction. Project members were intent on protecting their intellectual property and the liability of their members. Incorporation provided a vehicle that could offer these protections. It also improved their ability to attract donations, which was difficult to do without non-profit status. Incorporating offered the additional benefit of providing an infrastructure that would enable the group to “host” or sponsor other projects that might spin off from the founding project.

Contributors to projects were unified in agreeing that the rights associated with incorporation would be beneficial to ensuring the longevity of projects, but also feared that incorporation would taint the spirit of people coming together to collectively engage in an activity that they enjoyed doing for the sake of doing it.

At that same time, we were very leary of creating a Foundation, creating a corporate entity around something that was a community based project.
(*Founding Contributor, Webserver Project*)

It could be a non-legal entity and [we could] just have this thing be a mailing list basically, and not be incorporated or anything. That didn't work.
(*Sponsored Contributor, GUI Desktop Project*)

Although incorporation was viewed as a worthwhile endeavor, groups were hesitant to pursue it. The benefits of incorporation became more appealing as the community's work became more widely disseminated in commercial markets. Protection from liability and the ability to enforce the group's trademarks and intellectual property became more critical as more people in the commercial world used their work. The ability to raise funds also became more critical to the community's ability to protect their intellectual property. Legal counsel was needed in order to file copyrights, trademarks and incorporation.

However, in three of the four projects (The Webserver, GUI Desktop, and Linux Distribution Projects), the actual catalysts for incorporation were firms that wanted to work with these projects. Two different Fortune 500 firms went so far as to lend their legal expertise to help the GUI Desktop and Webserver projects draft charters and file the necessary paperwork.

It acted as a catalyst, forced us to make the decision to go and do it. (The firm) assisted on the legal side. (*Volunteer Contributor, Webserver Project*)

So with [Fortune 500 Firm Theta] coming to the front, all these issues of control and governance became so much more urgent, because look at [Firm Theta], it's a very competitive, very aggressive culture there. And they way they started their conversations with [the project] reflected that. (*Sponsored Contributor, GUI Desktop Project*)

Firms were motivated to assist communities to incorporate as it would provide a legally recognized entity with which they were more familiar. Firms that were helping the Webserver and GUI Desktop Projects wanted to be able to form agreements with an actual entity and have a point of contact: a person or place to 'go to'. Incorporating and creating a non-profit foundation also required a board, officers, regular meetings, and a host of other trappings, that communities were for the most part keen on avoiding.

All projects held open solicitations on community project mailing lists for comment on the construction of bylaws. Projects varied as to the specific ways in which members eventually agreed to support incorporation, but they followed a similar pattern: a few prominent leaders would draft preliminary ideas about incorporation and post them to the project website. Other contributors would comment via the mailing list.

Interviewer: So, the structure for the Foundation, how was that decided?

Informant: Well, it was all done in the public, we discussed everything in the public – everything – long debate -- there's like a thousand messages in the archive for about three months of discussion. Intense. Very intense. (*Founder, GUI Desktop Project*)

One person would volunteer to assimilate the comments and present a condensed range of options or edit the website. This way, members could craft their arguments and their concerns about incorporation in full view of each other. The process of creating a foundation and drafting corporate bylaws was not easy for any of the projects. It required articulating a minimum set of formal roles to guide the incorporation, namely the Board and Officers of the Corporation as well as the

decision-making process by which the Boards would be created and the role the foundation would play in relation to the project.

[T]hrough the process of incorporation, we also had to articulate our motivations to be this group. (*A Project Founder, Former Volunteer, Sponsored Contributor, Webserver Project*)

In these discussions, the arguments those supporting incorporation used to gain the support of dissenting members emphasized the protective attributes as opposed to the more formalized and rational aspects of incorporation. Several project leaders argued that incorporation was critical to enabling contributors to preserve their work in an increasingly commercial context.

When it [the foundation] came out a lot of people actually thought that it was some sort of like, you're selling out to companies? But, you know, we've been thinking of it very much in the reverse because what we're doing here is we're setting up a formal way that decisions get made, in a process that it goes through which is separate from any particular company. (*Sponsored Contributor, GUI Desktop Project*)

This informant, who has directly involved in institution building, viewed the creation of a formal structure for the foundation as a way to prevent cooptation or the emergence of a controlling influence. He indicates however that others did not initially fully agree. Informants did agree that incorporation could help projects assert their legal rights and shield individuals from liability concerns. However, the idea of creating such a formal structure also elicited two types of perceived threats to community projects.

One set of concerns stemmed from the adoption of more formal and rational practices that administering a foundation would entail. Members recognized

that incorporation might affect the informal and communal ways in which development work was currently structured. The other set of concerns stemmed from firms' interest in participating in the project's development and organizational efforts. Contributors worried that greater commercial involvement might lead to cooptation or the emergence of a single controlling interest on the project.

a. *Threats from Commercial Interests.* From early encounters with firms that were either selling their work or interested in cooperating with them, community members worried about firms' ability to respect their development and distribution terms. Within projects, there was often disagreement about the intentions of firms and the degree of risk affiliated with their association.

There were fears that [Fortune 500 firm] would buy us up, or that we had some sort of contract with them at [another firm] and we were selling [GUI Desktop Project] down the river. All sorts of problems, But all of these problems were aired.....People were scared in the community and get jumpy and then they realize it's only going to have good effects and they get on with it." (*Volunteer Contributor, GUI Desktop Project*)

Most, at a minimum, recognized that the motivations of firms differed from theirs and that this might have an effect.

These companies, they are contributing to it and it's really wonderful and the hackers are contributing to it and it's like, well actually, these companies are contributing to it because its in their business interests, so they bring an agenda and you need to be aware of that. (*Sponsored Contributor, Former Board Member, GUI Desktop Project*)

Project contributors that embraced the free software frame as opposed to the open source frame were more apt to recognize the potential for firms to influence their work or its distribution terms, the potential for cooptation and to propose tactics to

prevent it. Individuals on every project recognized that firms not only had different incentives, but that their resources could advantage them relative to the efforts of volunteers. For example, if firms hired masses of volunteers in order to gain access to projects, the distribution of firm affiliated versus independent volunteers might be affected and the pluralistic source of project contributions jeopardized.

Informants viewed the ability of firms to influence development work as potentially positive and negative in its consequences. Chapter IV detailed some of the benefits that informants felt could emerge from working with firms. Many viewed firm participation as vital to the on-going success and vitality of the project. Firm contributions were welcomed, provided that they were offered on the community's terms. Contributors recognized that while firms might make helpful contributions to the project, too heavy a concentration of contributions from one firm could violate the project's core values and threaten the independence of its development path. If communities allowed undue influence over their technical or organizational development, three aspects of the project could be at risk: contributors' individual autonomy over the technical direction of the project, the project's informal development practices and norms, and compliance with their licensing terms.

b. *Threat of Rationalistic Controls.* The other threat that worried project members stemmed not from commercial interests, but from the types of formalized changes that members envisioned as associated with incorporation. Incorporating for pragmatic reasons was acceptable to contributors, as long as its effects did not infringe on their technical autonomy. While only the Linux Distribution Project formally defined the rights of contributors prior to incorporation, the rights of

contributing members were more generally acknowledged as the right to contribute on one's own terms and to freely submit code modifications to guide the technical direction of the project. These rights were kept in check by each other.

Most of the group will acknowledge that I have done a good portion of that work. That has been the last two years of my life full time plus some [work] that I have done. But I still have to, I mean I still have to defend what I want to do. If I want to do it and everybody goes, no you are just wrong, then it doesn't go into the code. (*Sponsored Contributor, Webservice Project*)

As this informant explains, regardless of his tenure with the project and the amount of contributions he makes, his code contributions are still evaluated on the same terms as any other member. Informants reported that key criteria for reviewing and accepting code contributions were: technical merit, its ability to help the project's progress, its impact on other sub-units of the project, and the perceived experience and expertise of the contributor, in relation to the project. Thus, technical autonomy in this sense is not pure, but is negotiated through interactions with one's peers.

If the role of the new corporation was extended to include technical decisions, the very values that united the project could be threatened. Members in three of the four projects were concerned that their work practices might become subject to more rationalistic controls if the foundation was granted too much power. Contributors felt that an unfettered corporation could diminish not only the experience of working on community projects, but also, their own motivations for participating on such projects. While these concerns manifested in different ways, the fear that community norms would be affected by a more formalized structure was widely shared.

We try to keep it simple, because none of us really want to be in the mode of being a bureaucratic organization. We don't want to run an organization. We don't consider that to be fun. We are just doing that because we need the Foundation for the background for our fun projects. (*Contributor and Board Member, Websaver Project*)

The articulation of these two types of threats reflected contributors' appreciation of how the demands associated with commercial markets might influence their project's operation and direction as well as their own motivations for working together as a group. Arguments that emphasized the protective benefits of incorporation and deemphasized the formal and legal requirements associated with incorporation helped contributors to surmount these threats and fostered the internal consensus needed to support non-profit foundations.

Knowing the type of entity they did not want to become helped guide the creation of the foundation's role and identity. As Swaminathan and Wade would predict, maintaining a distinct identity from incumbents was important for challenging groups (2000). Community projects did not want to become firms. This was avoided by incorporating and creating a foundation. But incorporation itself was a deviation from the community's preferred form of organizing via mailing lists over the Internet. The community's organizational form signaled the group's identity to its members and others (Clemens, 1993; Simons and Ingram, 1997). As Clemens has argued, "models of organizations are part of the cultural tool kit of any society and serve expressive or communicative as well as instrumental functions" (1993: 771).

Designing a governance structure for the foundation can thus be viewed as a way for community projects to balance their adjustment in form from an identity based on shared values to a legal identity. The need to establish a formal system of governance also provided community managed projects with the opportunity to

define the roles firms could play in the community. Three of the projects (the Webservers, GUI Desktop, and Standard Setting Projects) decided to elect board members³⁴ which required defining a pool of potential voters³⁵. Two projects (the GUI Desktop and Standard Setting Project) created boards that were slated to represent different types of members: firms, non-profits and individuals.

Folks felt like, you know, if I'm a hacker, I should have a voice in this. I should be able to decide who gets to run this thing [the foundation]. [That is] when I got to say, let's have slates. Right? Because, its like, how do you get a board elected that balances out all these different needs nicely, right?
(Sponsored Contributor, GUI Desktop Project)

Balancing representation of community and firm members on foundation boards was a critical dimension of the design of the foundations and will be discussed further in Section B.

Incorporation provided a channel by which firms could voice their interests to community projects, make formal agreements, and introduce a more rationalist structure to community projects. Thus, firms had an incentive for donating their services to helping communities incorporate: helping them to become 'firm-like'.³⁶ There is variance as to whether incorporation could be interpreted as an act to preserve the project's livelihood or an accommodation to better work with firms. Arguments could be made for both. The three projects that received corporate assistance to incorporate, all did so after the development of the open source

³⁴ The Linux Distribution Project chose to have an appointed board with an elected project leader.

³⁵ Although, all four projects began with relatively open and porous boundaries in terms of who could join these groups, in the last two years, groups have been developing a more formally defined membership processes.

³⁶ The Linux Distribution Project incorporated prior to the development of the open source frame. For this project, incorporation was an independent activity pursued primarily for fund raising and liability protection without the assistance or influence of firms.

frame³⁷. The one project that incorporated without firm assistance or input, the Linux Distribution Project, did so prior to the creation of the open source frame and did so for reasons that were more clearly linked to its own survival. For this project, incorporation was less of an accommodation and more of a project initiated act.

On the other three projects, incorporation can be interpreted as accommodating, but also preserving in its consequences. Informants from the other three projects indicated that firms played an important and catalyzing role in initiating and helping with the incorporation process. It is thus impossible to tell when they would have incorporated on their own. The four projects incorporated between three to five years after their founding. The average time across the four projects from founding to incorporation was 4 years. The one project (GUI Desktop Project) that was founded closest to the creation of the open source frame incorporated faster than any of the other three. This may indicate that commercial interests may have helped accelerate a path that projects might have pursued on their own.

2. Representation. A second area that community projects became concerned with and adopted their practices to address was ways to represent the project. Establishing a point of contact within communities was problematic for firms as roles and responsibilities within the community were not easily discernible to those who did not participate in the mailing lists and IRC on a day to day basis. Some firms were reluctant to express their interest in these public forums as they feared public attacks or worse, that members of the press would identify their

³⁷ The one project, the Linux Distribution Project, that incorporated prior to the open source frame also did not adopt the open source frame. The reasons for incorporation were however not substantially different from the other projects.

overtures. In three of the four cases, firm representatives initially contacted de facto project leaders privately. These private conversations then opened up to communication with a larger select body of core team members.

But we didn't do it [approach the project] the open, we did it in private conversations because there was no place we could go to and have a confidential conversation. Because we weren't ready to announce ourselves to the world. (*Firm Founder interested in working with GUI Desktop Project*)

Private, 'off-channel' contact with firms violated community norms as project members pride themselves on the open and public nature of their development process. All code and dialogue justifying decisions about development and the group's organization are well documented on public mailing lists. Because all four projects are globally distributed, it is important for all discussions to happen through these channels where they can be accessible to all members. When most contributing members are not co-located, and all interactions occur on-line, information sharing can be very complete.

The use of off-line channels could threaten communal awareness of development and organizational activities and thus threaten democratic decision processes. This became apparent when firms threatened the community's authority by circumventing mailing lists and IRC channels to approach other firms about working with the community. For example, a firm that was interested in working with the GUI Desktop Project contacted other firms that had contacts contributing to the project, as opposed to contacting the community directly.

So if you know say, [Firm Theta] wanted to get involved in [GUI Desktop Project], right? Say the Foundation had existed, they could have talked to the Foundation. What they did instead was they talked to particular companies like ---, you know, for months and tried to plan everything in private, you know without [GUI Desktop Project] itself being involved. (*Sponsored Contributor, GUI Desktop Project*)

Even when firms approached community projects through appropriate channels, de facto and elected leaders³⁸ still felt uncomfortable speaking for the community, as they were all too aware of the varying attitudes project members held toward commercial participation in their projects. The informal decision making processes that guided technical development decisions did not seem adequate to informants faced with decisions that might affect fundamental aspects of the community structure.

At what point can you say the community is inviting [the firm] in?
(*Contributor, GUI Desktop Project*)

Projects that did not have mechanisms in place to make formal group decisions had difficulty representing the project to external parties. In articulating a corporate structure, project members also designed a mechanism to represent the project and a way to interact with firms that project members could authorize. A corporation not only provided a unified point of contact that firms could identify, but a mechanism, a formal channel, by which project members could represent the group.

The Linux Distribution Project's foundation was created for some of the same reasons as the other three, namely to protect volunteers from individual liability and to protect the group's intellectual property. However, it differs in important

³⁸ The Linux Distribution Project Leader is elected.

respects as it incorporated prior to the establishment of the open source frame and without the assistance of firms. The foundation affiliated with the Linux Distribution Project holds their assets and is only minimally involved with the direction of the project. Instead, the point of contact to represent the project is a project leader that is elected annually. However, the project created this position to solve a similar problem: to provide a way to represent the group that the outside world could identify.

We get questions like this – "We would like to have your CEO give us a call" and we get that regularly....and our basic thing is you obviously don't understand what [our project] is about, please read [the website] and if you have questions after that, feel free to send us another email. Right? Because people see [the project] as a distribution, they see the disks coming out, they see it all over the world. It is the second largest distribution in the world in terms of the user base according to the surveys. And it doesn't exist. And to somebody who is used to thinking of the traditional corporate sense that is nuts.... and we also get questions like "where are your corporate headquarters located" or things like that. (*Volunteer Contributor, Linux Distribution Project*)

This foundation has no members other than its board and its officers. Compared to the other three foundations, this foundation has the least direct involvement with a project. This project also works less closely with firms and is more closely linked with the free software frame than the open source frame, which helps explain these differences. Because this project had the least contact with firms and was less interested in working with firms, their foundation could assume a minimal role. While the Linux Distribution Project's precise mechanism to represent the project and its use of incorporation varies from the other projects, it shares a common need to specify a point of contact to represent the community and facilitate interactions between the online and the offline commercial world.

3. Defining a Role for Firms. The foundations set up to support community projects could now enter into agreements, such as intellectual property transactions, with firms. This was a new concept for online communities who formerly depended only on individual volunteers for contributions of code. The transition to a legal entity also changed project member's conception of the role firms could play. Firms could donate code and transfer code ownership rights to the corporation. This way, firms would be free to donate code to the foundation without being liable for the course of its future development.

In fact, when [Fortune 500 Firm Theta] donated the ---- code, the original --- code to us, and when [Fortune 500 Firm ABC] donated ----code and all these other things, they signed contracts that said we sign over copyright, not exclusively, they still get a copyright, but we have, we can consider that our code. And thus the [project] Foundation is liable for it. (*Contributor and Board Member, Webservice Project*)

An example of the type of language that would guide such an agreement is below.

This software consists of voluntary contributions made by many individuals on behalf of the [the project], and was originally based on software copyright (c) 1999, [Fortune 500 Firm ABC]. (*Excerpt from license agreement between project based foundation and firm*)

Community members in three of the projects did not want the creation of a foundation to change their membership policy toward firms. They wanted to maintain the principle that only individuals could become 'code committing'³⁹ members. Write access to the code base was earned based on one's demonstrated skill and experience over time. Two of the projects that were most supported by

³⁹ Code committing members are those that have 'write access' or the ability to commit code changes to the code repository.

corporate donors (The GUI Desktop and Standard Setting Projects) created formal channels for companies to voice their opinions on the development of the projects.

Well, there's a bunch of companies that joined [the advisory board], but they basically needed this channel – being on the advisory board and having their voice heard and other things. So that was the reason for its [the advisory board] creation. (*Founder, GUI Desktop Project*)

Firms could thus use their roles on Advisory Board or Committees to influence the technical direction of these projects. The Linux Distribution Project created a partner program to recognize contributions and sponsorships for firms, but did not create a forum for firms to express their views and opinions on development issues.

There is some evidence that the frame transformation in the spring of 1998 helped attract commercial attention⁴⁰. The new frame helped firms develop logics to rationalize their participation in community projects. Firms however had difficulty in assessing how to approach and work with community projects. They preferred to work with other legal entities like themselves. Three of the community projects had entertained the idea of incorporating, but the attention that arose from the new frame and the heightened interest from firms hastened the process of becoming a legal entity. Becoming a legal entity forced communities to articulate a means of representing themselves to the outside world and to define a role for commercial entities in their projects. As a legal entity, community projects could form binding agreements with firms and enjoy the benefits of incorporation. The corporation could hold the group's copyrights and trademarks and protect its members from

⁴⁰ The Linux Distribution Project did not fully adopt the open source frame and did not have trouble in attracting supporters. However, they had the least amount of firm participation of the projects studied here.

liability. However, project members were not interested in extracting profits for their work. The primary motivation of community members was not to become more 'firm-like', but to realign their balance of power with corporations.

What we were trying to do as a Foundation is have our own entity that could be on an equal footing with these companies that could represent the community interests, right? (*Sponsored Contributor, GUI Desktop Project*)

Most informants viewed incorporation as a necessary evil, as it was not their preferred work context. This reluctance affected the role foundations assumed which is discussed in the next section.

B. Preservation

Subsequent to the formation of foundations for each project, community members both embraced and distanced themselves from the incorporation. Not all project contributors welcomed the idea of incorporation. Some treated the idea with trepidation, if not outright hostility.

People in the [Linux Distribution Project] were hostile to the corporation ideas on a few fronts that we discussed. (*Former Volunteer Contributor, Linux Distribution Project*)

I was not convinced that we want a [GUI Desktop Project] Foundation. (*Volunteer Contributor, GUI Desktop Project*)

Online discussions over incorporation had helped project members identify a range of considerations that affected how they designed their foundations. Those who worked most closely on institution building were thus conscious of the need to avoid

being either too oriented toward commercial interests or too “heavy handed” in their design of a more formalized organization.

There was concern that we wanted to tread very carefully, to avoid the appearance that [The foundation’s] prime goal was to “raise money.” It was noted that [the foundation] would be looked at very closely for signs of “selling out” and that [the foundation] should tend towards conservatism in this area to avoid any concerns in the user community. (*Board Meeting Minutes, June 1, 1999, Webservice Project*)

Specific tactics to guard against the emergence of rationalistic controls and to protect against any firm from establishing a controlling interest accompanied this tone of treading carefully during the design of the foundations. I call these preservation tactics as they focused on preserving three areas of community life: work practices, technical direction, and goals. The tactics included: 1) limiting the authority and role of the corporation; 2) delimiting the role available to firms and 3) protecting the community’s intellectual property. Next, I describe these tactics in more detail.

1. Limit the Foundation’s Authority and Role. Incorporation required changes that were far more bureaucratic than members were accustomed, such as: defining a population of membership, determining board selection procedures, holding regular board and member meetings, publishing meeting minutes, establishing a bank account, and acquiring legal counsel. Members recognized that instilling the foundation with too much power might impose other changes with the specter of rationalistic controls such as: defined roles, fixed deadlines, and a directed as opposed to evolving development path. To avoid this, project members wanted to bound the foundation’s role and authority over technical aspects of the project.

Limiting the foundation's role was one way to meet the competing objectives of incorporating and preserving contributors' technical autonomy and community norms. By restricting the role of the foundation to primarily legal and administrative domains as opposed to technical domains, project members hoped to preserve their work, norms and culture and mitigate against the threat of rationalistic controls and undue influence from commercial interests. Guarding the technical domain and limiting the foundation's right to employ were two practices projects used to limit the role and authority of the newly formed foundations.

a. *Guarding the Technical Domain.* The charters of incorporation for all four projects include language that restricts the foundation's authority over technical decisions and rests this authority with those who work closest to the projects: individuals typically called maintainers. Maintainers may feed their work into larger modules that are managed by package managers, sub-project heads or release coordinators, depending on the project. Sub-projects within larger projects typically maintain distinct mailing lists. Technical discussions about what ideas and contributions should move the project forward occur on such lists.

The core of the technical domain lies in the right to accept or reject code contributions into the code repository. Contributions may "make it in", be sent back for revisions and reformatting, or be outright rejected by those who participate on a particular mailing list. The next level of the technical domain addresses decisions regarding the overall technical direction or architecture of the project and the determination of what constitutes a "release". A release is a full version of the software with all of its intended components that is released to the public at a point

when developers feel that it can be safely tested or used by others not directly involved in the project.

The core of the technical domain was largely protected from a foundation's influence. Informants did not feel as though project foundations had a role in guiding decisions about specific code contributions.

[The foundation] itself is just sort of in charge of ideals and assets, ideals and assets, rather than a project directly. That is the intent, that a project can just say 'look we need this kind of an organization behind us, but we don't want them telling us no, you are not going to release right now, you are going to release in two months' because it makes more sense you know. You have to let the projects develop how they want to develop. [The foundation] is just there to just do what the projects themselves can't do. Which is to be a legal entity and handle the kinds of things a legal entity can handle that the [Linux Distribution Project], since it doesn't exist, can't handle. (*Volunteer Contributor, Linux Distribution Project*)

Is this thing supposed to run [GUI Desktop Project]? People say, no, and people still say no. [GUI Desktop Project] is run by the maintainers of the packages. [GUI Desktop Project] is and that still is how it is. If you're the maintainer of a package, you own a package. And you know, not much we can do about that. So, people did not want the [GUI Desktop Project] Foundation to run [GUI Desktop Project]. (*Sponsored Contributor, GUI Desktop Project*)

Ensuring that technical decision-making would be unaffected by the foundation was important as it helped make incorporation palatable for project members that were skeptical or resistant to the idea. Informants involved in the construction of the corporate charters felt that retaining the technical autonomy of the volunteers who worked on these projects was critical to sustaining members' commitment to the project and to maintain current levels of effort. Any perceived reduction in autonomy could create a disincentive for contributors to work on the project.

In practice, autonomy at the second level of technical decision-making was more difficult, particularly for the one project (GUI Desktop Project) that explicitly and formally rested authority to manage release schedules with the board of their non-profit foundation.

The foundation bears the responsibility of coordinating each subsequent release of [GUI Desktop Project]. For each release, this will include setting a schedule (whether or not it is overlooked), choosing the set of modules which are a part of the release, and preparing the appropriate marketing materials. [GUI Desktop Project] is a loose collection of independent projects. The foundation will determine the set of modules which fall under the [project] umbrella. Most often, the foundation will endorse a project simply by including it in a release. In some cases, however, a project that is not scheduled to be included in any particular release will be designated as a part of [GUI Desktop Project]. In these ways, the foundation will be "defining [GUI Desktop Project]." (*GUI Desktop Project Charter, October, 23, 2000*)

This was the only project to explicitly grant this type of authority to their foundation. Several members on this project worried that release coordination authority would effectively blur the boundaries of organizational and technical decision-making and threaten members' control over the technical domain.

The reality is that, in my opinion, the thing [the foundation] is going to end up running [GUI Desktop Project]. And people don't want to say that because it just runs counter to the democratic values of the thing, but if you look at the combination of release coordination alone. Articulate the vision of [GUI Desktop Project] is another task that was cited. The release coordination alone gives you so much control, that you're effectively running the thing, because what you end up saying when you do a release is deciding what is a part of it and what is not a part of it? right? So that gives you a tremendous amount of control.... So the release coordination function inherently contains a fair amount of power and if you say it that way, then people start throwing fits, so what you say is, we're going to do release coordination, right? And the reality is that, in my opinion, over time the thing is going to take on more and more power, because it needs to. There needs to be a body, where, if you have a major problem, where it gets resolved. (*Sponsored Contributor, GUI Desktop Project, February 8, 2001*)

This informant viewed authority over release coordination as a step away from authority over the project. However, he did not perceive this expansion of the foundation's authority as wholly negative, as he thought it might enhance coordination and efficiency. Coordination and efficiency were more important to this project because it worked closely with firms. Projects that worked more closely with firms that were building business models using their work may have been pressured to become more predictable in their development progress, thus influencing the role that the foundation assumed.

By granting release coordination authority to their foundation, this project essentially granted their foundation the power to define the technical boundaries of their product, and by default, the boundaries of the group. Soon after adoption of the group's charter, board members approached their release responsibilities tentatively.

How do we make decisions on the API [Application Program Interface] freeze? The Board won't mandate but should suggest, and take the initiative to make a proposal. (*GUI Desktop Project Board Meeting, Project, February 14, 2001*)

However, in a few short months, the board appeared to gain confidence in exercising their authority over release coordination. The fears articulated by the sponsored contributor on February 8, 2001 were tested in the foundation's board meeting three months later.

We selected [---] and [---] to take the lead on building a Release Team and work with the folks who volunteered to put together a [GUI Desktop Project] 2.0 release plan. This will include a listing of all the top-level tasks that need to be accomplished, along with preliminary due-dates for each of

these and a full committee roster with specific responsibilities assigned to each team member. We'd like to see quick progress on this, and have at least the committee structure and responsibilities flushed out within 2 weeks. It is relatively clear that the work of doing the release coordination for [this release] will be larger than what was needed for [the last release]. (*GUI Desktop Project, Board Meeting, May 1, 2001*)

Without any further context, the discussion of project planning at this meeting could be mistaken for any meeting at a software firm. Board members selected team leads, asked for a list of tasks with set deadlines and expected swift progress. This is a marked deviation from the free software community's founding norm of volunteering on individual terms.

Because of the foundation's role in coordinating releases, members on this project had a more difficult time in guarding their technical domain. Some members pushed back on what they saw as too active a role for the foundation. They viewed the foundation's role in release coordination as disrespectful of the limits of its authority and a violation of community norms.

Informant: I don't know that the foundation is actually helping anyone or doing anyone any good really. It tries to, well I don't know, I think it probably is useful so long as it doesn't overstep its...I mean, fair enough, you want to have an organization that can say "This is [GUI Desktop Project]" or "That's not [GUI Desktop Project]". But they try to get into strategy and they try to produce a vision and it's not really their place to do...they're interfering in technical matters. Or feeling they ought to...(goes on)

Interviewer: So what should they be doing then?

Informant: Oh very little. (*Volunteer Contributor, GUI Desktop Project*)

Release coordination on the other three projects was managed by individual contributors who were not necessarily affiliated with the foundation, but either

volunteered for the role or were informally nominated based on their experience. Evidence of outright resistance to the foundation's authority only appeared on the GUI Desktop Project.

Of the four projects, the project with the least commercial involvement, the Linux Distribution Project, granted their foundation the smallest role in relation to the project. Informants from this project did not mention any infringement on the part of the foundation and some chose to ignore it. Linux Distribution Project's relation with their foundation was initially very contentious. Dissension over the power of the foundation's first board of directors led to the resignation of several of them and to a heated discussion and reconsideration of the foundation's role.

Then at some point, three of the board of directors at [the foundation] decided to resign after an argument with [Linux Distribution Project]. Where [Linux Distribution Project] was basically asking them what is the money, how are things working? [The foundation] should be working for [Linux Distribution Project]. The relationship between [the foundation] and [Linux Distribution Project] was unclear at that point. So three people resigned from the Board of Directors. (*Volunteer Contributor, Linux Distribution Project, February 19, 2001*)

At this point, a few members on Linux Distribution Project that cared most about the creation of the foundation solicited project members for feedback and revised the foundation's charter. The new charter more specifically limits the power of the Board of Directors and frames the foundation in terms of serving the needs of the project.

Interviewer: Tell me about redoing those [the project charter], what was that process like?

Informant: Well the first question was who is [the foundation]? What are they intended to do right? Because some people saw [the foundation] as controlling [the project] because they manage [the project's] money and what

not. But the [project's] sentiment was that [the foundation] exists to be a legal entity behind [the project], to hold the assets to be able to handle the money, to be a nonprofit organization that people can make donations to and get really good tax deductions. So basically [the foundation] is intended to be an umbrella organization for projects. And just to hold money and assets for them and if needed, provide things like mailing lists or servers. (*Volunteer Contributor, Linux Distribution Project, February 19, 2001*)

This framing was meant to reverse the power structure between the project and the foundation and signal to all members that the authority of the foundation depended upon and was recallable by project members at large.

The authority of Linux Distribution Project's foundation is not just limited, but stripped of power not explicitly granted to it by its members. Foundation members behave almost as legal executors that represent the will of the project. They do not have any power, coordinating or otherwise, above and beyond that of individual volunteer contributors to the project.

[The foundation] is sort of at arm's length from [the project] and [the project] very clearly manages itself and has a relationship with [the foundation] where [the foundation] holds property for [the project]. And that works. And [the foundation] has been very effective in collecting donations for [the project]. (*Former Project Leader and Volunteer Contributor, Linux Distribution Project, March 14, 2001*)

The language informants from this project used to talk about the project and the foundation also reflects the fact, that in their minds, these were two very distinct entities: the foundation and the project.

Due to initial concerns with the amount of power granted to the board, members on this project rearticulated a narrower role for their foundation. As a result, this foundation is not only the least active of the four, but also, the smallest in terms of membership. It has no direct members outside of the Foundation's Board

and corporate officers. Project members are not obliged to, but can elect to join the foundation. To date, few have done so. Membership in this foundation has not been as attractive to contributors as it has been for the other three projects primarily because of the limited role of the foundation. Project membership is however a carefully managed and complicated process. Potential new members must be sponsored by a current member; demonstrate their technical ability to contribute to specific packages on the project; read and sign their agreement with the project manual, social contract and constitution; and have their identity verified through a face to face encryption key signing process.

Project members who did not belong to the foundation thought of it only as an entity to serve the project, protect their intellectual property and help them insure against individual liability. Some individuals chose to ignore or distance themselves entirely.

[The foundation] is, its just an umbrella that people can hold over their head if they want to. And if they don't, they can ignore it. (*Linux Distribution Project Board Member and Volunteer Contributor*)

Limiting the foundation's authority and distancing oneself from the entity, allowed those who did not fully agree with the emerging new structure, to continue to work within the community. This was an alternative to possible responses to organizational dissatisfaction, namely exit and voice (Hirschman, 1970).

The difference in authority granted to the foundations for the Linux Distribution Project and the GUI Desktop Project may be related to differences in project members' stance toward commercial interests. The GUI Desktop Project had much closer relations with firms, provided firms with more access to their

organizational building process, and provided firms with a greater role in their new organizational structure than did the Linux Distribution Project. The Linux Distribution Project was the only project not to have any involvement from a Fortune 500 company in the crafting of their corporate bylaws as well as the only one that did not include a provision in their charter that allowed firms to participate in a formal way. This project also incorporated earlier than any of the other projects and did so prior to the dramatic rise in commercial interest in free software and open source.

A few theoretical propositions emerge from these comparisons. One, projects that minimize the role of their foundation and the power granted to their boards of directors may be better able to guard the technical domain from sources of influence. Two, projects that manage release authority independent of the foundation's structure, may be better able to preserve the technical autonomy of their members. Three, projects that work more closely with firms may face greater pressure to provide more centralized and formal coordination and thus, experience more difficulty in protecting the technical autonomy of their members. One explanation may be that firms prefer a more predictable development environment, and thus encourage the design of foundations that assume a more expansive role in technical development. The larger the role of the foundation, the more likely it was that rationalistic controls might emerge that could challenge the autonomy of individual members.

b. *Limiting Employment.* Another way that the foundation's authority could be limited was by not permitting it to employ staff. Two projects (the Webserver Project and the Linux Distribution Project) do not have foundations that employ

staff directly⁴¹ as a deliberate tactic to limit the growth in size and power of their respective foundations. Members on the Webserver Project felt that transitioning to an employer would change the nature of their work. They recognized that the act of employment would bring with it on-going responsibility for acquiring revenue and that this type of pressure might change what most members viewed as a pleasurable hobby into a responsibility.

Informant #1: My concern with a salaried person is that it puts the effort on us to make the money come in. We will have to put forth the effort to maintain fund raising, does the board want to do that?

Informant #2: Well are there things we want to do?

Informant #3: The pressure to constantly produce revenue would change the nature of this. I have plenty of that pressure elsewhere. Would be way....this is about hacking on code.....

Informant #4: We don't want to squander our revenue stream on salary people. What do we want to hire someone for? We code. We have fun coding. (*Webserver Project Members Meeting, April 3, 2001*)

While a simple reason for not hiring staff could be a lack of resources or a lack of need, interactions such as the one above, suggest that these decisions hinged on much more than either the availability of resources or assessments of a project's needs over and above what could be accomplished with volunteers. Nelson and Barley's study of the commodification of emergency medical technicians (1997) suggests that when unpaid work becomes paid, exchanges that were once linked to familial or communal roles become contractual (1997: 624).

⁴¹ These projects do, however, have contractors who help them with specific functions such as providing expertise on legal, tax and accounting matters.

A transition, from a communal role to a contractual role, was one that community members wanted to avoid. Informants from the Webserver Project recognized that hiring people to help with volunteer work could do two things. One, it could change volunteers' feelings toward the group and thus their motivations to contribute. If people were hired to do the same work volunteers did for free, volunteer contributors might grow to resent this differential. Two, it could create a set of responsibilities that would require on-going care and revenue, thus introducing survival concerns into the foundation's administration. Members on this project anticipated that by avoiding employment, they could preserve the social meaning that community projects had for them and limit any possible growth in bureaucracy.

Limiting employment was intended to safeguard foundations from becoming too professional, formal, and concerned with their own survival. There is some evidence that the presence of professional management in social movements is associated with the adoption of formal structures⁴² and tactics that aid in organizational maintenance (Staggenborg, 1988). All of the social movement organizations that moved to more formal structures in Staggenborg's study of the pro-choice movement did so under the leadership of professional managers (1988: 595). Movement entrepreneurs or founders preferred informal structures and resisted the creation of formalized social movements run by professional leaders (1988). Unlike Staggenborg's (1988) study, informants in this study did not resist the development of professional management and formal structures in order to exert personal control, but to promote and sustain a pluralistic and communal nature.

⁴² Social movement organizations with formal structures are defined by Staggenborg as those with established procedures to perform certain tasks routinely despite changes in leadership; defined criteria for membership; rules governing sub units, and a developed division of labor (1988: 587).

Egalitarianism and pluralism are an important part of the hacker ideology (Levy, 1994) and highly valued by these project groups.

Similarly, informants in this study, many of whom had been with the project since its origins, also feared that if they hired staff, the project might become more susceptible to organizational maintenance activities as opposed to activities directly related to the project. They feared what organizational theorists might call 'goal transformation' (Jenkins, 1977) or what political scientists might call Michels' "iron law of oligarchy" (1915/1968). Michels' influential, although overly deterministic, argument is that challenging groups will necessarily organize formally to gain the attention of established groups. The presumption is that powerless groups will gain more legitimacy, power and influence the more their form resembles that of established groups (Fligstein, 1990). Michels argues that scale and the need to coordinate will necessitate formal structures and the need for professional leadership. Leaders, once in power, will enjoy their power to the detriment of the causes and constituents they once represented and become more interested in organizational maintenance and preserving their links with the established elite (Michels 1915/1968). Thus, an oligarchical rule emerges that ceases to represent the will of founding members (Michels, 1915/1968).

A long-standing debate in the social movement literature has evolved around the question of whether social movement organizations, in order to gain power and legitimacy for their cause, will necessarily develop formal and bureaucratic organizations ruled by an oligarchy. Although many have criticized Michel's argument for its poor definition of terms and deterministic conclusions (Hands, 1971), Piven and Cloward (1977) and Cloward and Piven (1984) found some support

for the idea that, over time, social groups tend to formalize, become more susceptible to cooptation and less attentive to their founding goals and the needs of their members or constituency. Piven and Cloward (1977) argue that 'organization building' tasks can deflect the energy of social movements from pursuing their goals and transform their social concerns into concerns for organizational survival (Michels, 1915/1968).

The inevitability of this argument has been questioned and essentially disproved (Zald and Ash, 1966; Jenkins, 1977; Minkoff, 1999). Social movement theorists have found that neither conservatism (Jenkins, 1977; Minkoff, 1999) nor organizational maintenance are iron laws, but contingent on the particulars of the movement and its relation to its environment. The concerns raised by the 'iron law' thesis, were however, very similar to ones held by informants in this study. Zald and Ash (1966) argue that there are three processes represented by Michel's thesis, although they are empirically difficult to separate: goal transformation, a shift to organizational maintenance, and oligarchization. Informants worried about all three.

Three out of four projects considered the possibility that employment responsibilities could attenuate the community's goals.

What's very funny about this is that the [GUI Desktop Project] community was very adverse to the idea of having a staff in the [GUI Desktop Project] foundation, because they felt that would corrupt, you know, the purity of the thing? However they don't have a problem with someone on the corporate payroll doing that exact same function. (*Sponsored Contributor, GUI Desktop Project*)

In recapturing the discussions in favor of and opposing employment, the informant above wondered why employment was viewed so negatively, as in his eyes, being a sponsored contributor served no greater danger. Members on the GUI Desktop

Project initially opposed becoming an employer, but later gained support for it by arguing that the project would be better served if someone was dedicated to an administrative and fundraising role on behalf of the project, so as to allow project members to do what they do best, code. The Standard Setting Project, from its inception, planned to hire a professional leader to shape and lead their foundation, as the coordination complexities of building a standard that could gain the appropriate amount of attention, legitimacy and resources needed for a standard to become adopted was daunting.

The rationales informants offered for hiring staff related to meeting the demands of a commercial marketplace: to help provide a more predictable development cycle, to improve on areas volunteers were not interested in, engage in more coordinated marketing efforts, and mediate competing commercial interests on development specific issues. The need to present a coordinated face to the outside world was more important to projects that had greater commercial participation from firms, the Standard Setting Project in particular.

Informant: One of the things we really want to do, because we have a lot of volunteers, and people who are part-time at their companies just work on things, we've had a little bit of a continuity problem. Getting continuity. People will have 20% one week and then 0% for two weeks, and then 40% one week. And that slows things down a lot. It takes concentrated effort, especially with multiple people working on something, for extended periods of time, to get these things done. We want to hire developers and editors and engineers to push these things out faster.

Interviewer: They would be employed by...?

Informant: The [Foundation]. (*Sponsored Contributor, Standard Setting Project*)

Hiring people to coordinate and work on key areas could enable the project to progress in a more predictable way and provide a more stable and reliable platform

for firms who wished to use the community's code. Volunteers who contributed on their own terms were not always interested in working on areas that brought them less satisfaction. For example, documentation was an area that people described as often left lacking. Foundation professionals could also coordinate the project's marketing efforts and mediate competing commercial interests. The two projects that hired staff (the GUI Desktop and Standard Setting Projects) also worked with a greater number of Fortune 500 firms than projects that did not. However, those projects that hired staff shared the same concerns as those that did not, and were committed to keeping their staffing levels low to avoid creating unnecessary bureaucracy and reduce the chance of oligarchy.

The reluctance to hire people and build a strong professional organization is not surprising in a technical community that holds little regard for hierarchy and emphasizes individual talent and skill (Levy, 1994). It is also symptomatic of "new social movements" whose values and goals are rooted in identity as opposed to class distinctions (Calhoun, 1995). New social movements often defend or politicize specific spheres of life with a very limited scope, as opposed to representing all of the needs of their constituents⁴³ (Calhoun, 1995). One way in which these movements differ from traditional social movements is that the form their organization subsumes may be as important as the goals it purports. New social movement supporters want their organizations to "exemplify the values they seek to promulgate" (Calhoun, 1995: 192) similar to the Israeli kibbutzim that Simons and Ingram studied (1997).

⁴³ Women's rights, the environmental, right to life and right to choose movements would fall into this class of movements which Calhoun estimates as beginning to appear after 1968 (Calhoun, 1995).

Calhoun argues that NSMs have many similarities with traditional movements, but that they may differ in their commitment to direct as opposed to representative democracy and their resistance to involve professional movement staff (1995: 192). Other studies of ideologically based organizations (Haveman and Rao, 1997; Simons and Ingram; 1997) have found that the form an organization takes can be used to reflect a social group's frame and ideology or as Haveman and Rao (1997) call it, 'a theory in use'. For both the free software and open source movements, the form of organizing itself, the community managed project, expressed the community's preferred alternative to developing software within firms. To change that form risked changing the ideals of the movement.

The transition from a project constituted by an electronic mailing list to a foundation was not a decision taken lightly by either project. Guarding the technical domain and limiting the right of the foundation to become an employer were tactics community managed projects used to preserve their work practices, technical autonomy, and goals. Limiting the foundation's authority over technical dimensions of development work could be problematical on projects that granted foundations power to coordinate releases. The project that was most effective in limiting the authority and role of their foundation, the Linux Distribution Project, also had the least contact and support with commercial entities. While three of the four projects expressed goal displacement concerns associated with the hiring of professional staff, only two refrained from doing so. The projects that hired staff, worked with a greater number of firms and absorbed a larger coordinating role than those that did not. Those projects that avoided hiring professionals, had a stronger appreciation of

the responsibility associated with becoming an employer and its potential to affect the meaning they associated with their work than those who did.

2. Delimit the Roles Available to Firms. Another way that community projects mitigated against the threat of goal displacement and the creep of rationalistic controls was to delimit the roles available to firms. Two tactics project members used were: restricting firm representation on the foundation's board of directors and creating mechanisms for firms to voice their interests without decision making rights.

a. *Restricting Board Composition.* Project members insisted that sponsored contributors were no different from volunteer contributors. Both types of contributors had to earn their way into the project to gain access to the code repository. Project members wanted to maintain the sense that projects were composed of individuals, some of whom just happened to work at firms.

We're just contributors, we just happen to be united as a company but we're just individual contributors in the whole of the project. I mean, maybe we contribute a lot to a single thing in the system or we contribute patches here and there, but just from the perspective of [GUI Desktop Project], we're just contributors. (*Former Volunteer, Sponsored Contributor, GUI Desktop Project*)

However, this led to a source of conflict that touched all of the projects in some way. As more and more sponsored contributors joined the project, how could project members determine if a commercial controlling interest formed? At the membership level, this was difficult to assess as both sponsored and volunteer contributors often adopted project domain email address that masked the organizations with which they were affiliated. Informants did not worry that conflicts of interest at the individual level would affect code decisions. This may be because, as each individual's

autonomy was subject to defense in online forums, they were confident that any outside influence would be detected. If detected, it certainly would not be tolerated. Project members did however worry about a majority interest forming on the foundation's board of directors.

Corporate boards often protect against the formation of majorities by creating slated boards, designed to recognize different interests. However, this solution to maintaining pluralism was difficult for project members to conceptualize because it required members to recognize the affiliations of sponsored contributors. In other words, members had to first recognize potential opportunities for conflicts of interest in their current form before they could design mechanisms to manage them. The interaction among Webservice Project members shown below demonstrates this conceptual difficulty.

Project member #1: Up to this stage, we have been coming in as independent coders, but what if 7 members, a majority of the board are from the same company?

Project member #2: We trust each other, what would we do? Force people to step down?

Project member #3: Who controls the board? The members or the outside? (Angrily) I mean this could come about in the next few minutes! (*Member contributors on Webservice Project, Annual Board Meeting, April 3, 2001 in discussion just before foundation board elections*)

In the exchange above, member #1 asks how the group wants to handle the possibility that individuals elected by the group might be affiliated with the same firm; thereby forming a majority corporate interest. Member #2 recognizes that preventing such a thing from happening would require recognizing the firm affiliations of members as opposed to their individual status within the group. His

questioning of the group's foundation of trust suggests that such a change would alter the basis of trust from a communal one among individuals to one between agents and representatives. Member #3 does not identify a conflict of interest, and dislikes the idea of modifying the group's practices based on merely the appearance of a conflict of interest. He is angry that the group's trust in each other would not be enough to eradicate the possibility of conflicts of interest.

The conflict that surfaces in this exchange exposes the tension between individuals' desire to trust and recognize each other as a community of individuals, and their need to recognize that sponsorship of individuals could upset the balance of pluralism that they strove so carefully to maintain. Members who favored the idea of an amendment to their charter that could safeguard against the formation of majority interests favored the change, not out of a lack of trust for their current fellow members, but out of concern for the future. They wanted to create structures that would last independent of the current membership to protect against the possibility of a corporate 'take-over'. This indicates that project members did identify possible conflicts of interest that could stem from the sponsorship of the work formerly performed by volunteers. Project members did not take action at this meeting and the matter has been proposed for further review, as such a change would require a modification to the current charter. As there was no immediate threat of a majority formation, members agreed to rethink whether such a step was necessary.

However, two other projects adopted terms, as an explicit part of their charter, to ensure that contributors sponsored by the same firm could not hold a majority of board seats. The GUI Desktop Project resisted corporate influence to

create specific positions for firms and insisted on designing a board comprised of freely elected project members.

This is about openness and democracy and no corporate influence poisoning the whole thing, and the other half of your time, you spend with these corporate folks, right? Some of them [Fortune 500 Firm Theta] are pretty heavy handed, some of these folks are saying things like, 'if we don't have a board member, *we will not join this movement*. We must be on the board of directors. The board of directors should be composed of the company representatives.' Right? This is the sort of conversation, then you go, well, you know I'm not sure that our hacker community is ready for that.
(*Sponsored Contributor, GUI Desktop Project*)

First, it should be noted, this informant is describing the efforts of a large Fortune 500 firm, a dominant player in their industry, to gain access to the decision-making structures of a social movement. This scenario is quite a reversal from many assumptions of social movement theory and suggests that this challenging group may have accomplished an interesting power reversal. In the informant's recall of a conversation with representatives from this particular firm, they threatened *not to join the movement* (emphasis added) unless they were granted an explicit role in project governance. Such direct and explicit pressure from corporations to influence the design of the foundation and board selection may have made it easier for members on this project to adopt restrictions on board representation with fewer of the concerns demonstrated by the Webserver Project.

GUI Desktop Project revised their charter to lower the maximum number of people that could be associated with one company from 50% to 40%, despite the fact that GUI Desktop Project's charter forbids board members from acting as representatives of their employers. Even though individuals were not authorized to represent their employers, members on this project recognized that, to some extent,

a conflict of interest between corporate sponsors and the interests of the project was possible. Restricting firm affiliation in board membership added a layer of security to ensure that a majority interest did not emerge, even in places where corporate interests were not officially recognized.

The Standard Setting Project protected their board from majority influence by designing a slated board with allocations for different types of members: three corporate representatives, two non-profit representatives and four individual representatives. With this arrangement, in order for a firm to establish a majority, they would have to gain support from two additional members, either a non-profit organization or an individual. However, it was still possible for a firm to establish a majority if it sponsored contributors as individual members. Thus, in addition to maintaining pluralism in the types of members that could join this board, this project designed a way to ensure against a majority or what they called a 'control group' from forming across classes of membership. A control group could have no more than two members in any membership category.

Informant: We have something, a concept that we call a control group. It's actually probably the part that I am most proud of in the by-laws. It's not really a big thing. It's nothing dramatic, or insightful. But a control group is defined as any organization, any individual or organization that has some sort of control. We've talked about this before.

Interviewer: Yes, it's across all categories.

Informant: Right. A control group can have no more than two board seats in two separate control groups. That is two separate classes of membership. So, a control group can't have two representations in a single class of membership, but they can [have two members] across all classes of membership. If there a third one comes in, then there are provisions on how to get rid of the third.

Interviewer: Your reason for these two layers of security was?

Informant: To verify down the road when I'm no longer involved in this, and the people who are involved in it today are not involved in it, that there is not only a culture, but a structure for maintaining sanity, and goodness. The same reason why the constitution is written. (*Sponsored Contributor and Foundation Leader, Standard Setting Project*)

Similar to the Webservice Project, those who were most involved in institution building wanted to create resilient structures that would not depend upon the leaders that inspired them. The Standard Setting Project does not restrict firms from being members, nor does it prohibit individuals from representing their employers. Instead, the affiliations of the three different types of actors who might be concerned with the project are explicitly recognized. Board positions are balanced to ensure that a majority cannot form and encourages negotiation among different types of actors. This project was able to handle the possibility of conflicts of interest more explicitly than the other projects because, as a standard setting group, it had, from its inception been in the position of balancing the viewpoints of multiple parties.

While community projects preferred not to acknowledge the firm affiliations of sponsored contributors, some projects found that this was necessary in order to protect the project against the possibility of majority interests forming on their board. It also constituted a subtle reminder that corporate sponsorship could carry with it conflicts of interest, which also manifested itself in role conflict for sponsored contributors: a topic for discussion in Chapter VII. Projects that were most active in guarding against the possibility of conflicts between community and firm interests had been most exposed to ways in which these conflicts might occur.

b. *Creating Mechanisms for Voice.* As discussed previously, firms, as units, did not have direct technical decision-making rights on most community projects⁴⁴. However, firms that were integrating the work of community projects into their line of products and services wanted to establish a means by which they could voice their opinions on future development initiatives. Three out of four projects designed vehicles to provide firms with a mechanism to voice their opinions on the future technical development of the project without providing them technical decision-making rights. These mechanisms differed by project. The Linux Distribution Project created a partners program to acknowledge firms that supported them and a vehicle by which the opinions of partners could be acknowledged. The GUI Desktop Project created an advisory board for firms to discuss future developments of the community's work and charged firms for the privilege of belonging.

The advisory board is comprised of companies and organizations which want to further the progress of [GUI Desktop Project]. This is a forum for them to communicate with each other and the [GUI Desktop Project] Foundation membership. They have no power to make decisions about [GUI Desktop Project]. (*GUI Desktop Project, Project Charter, October 23, 2000*)

The Standard Setting Project allowed firms to have a voice and a decision-making role by allowing firms to join directly as members of the foundation for a fee that varies with different categories of membership privileges.

None of these mechanisms for voice committed community members to adjust their development practices or the technical direction of their work to the needs of corporate participants. Instead, they offered community members a means

⁴⁴ That is, with the exception of Standard Setting Project which allows firms to be (paying) members on the board.

to display their consideration for corporate participation without requisite commitment. It also provided a way for projects to recognize firms that were supporting them by sponsoring contributors or donating hardware or services. Community projects used these two mechanisms: restricting board composition and creating mechanisms for voice to delimit the role of firms on community projects. Delimiting the role of firms helped community projects to preserve their autonomy, sustain pluralism and prevent majority interests from influencing their work.

3. Protecting the Community's Intellectual Property. The first two types of preservation tactics discussed thus far were used to preserve project member's technical autonomy and balance competing interests. However, projects had a separate set of concerns with regards to the use and distribution of their software. Firms had the right to use community work in developing new products and services, but this was not an unlimited or unchecked right. Three of the four projects⁴⁵ had another set of concerns, namely how to protect their work against the threat of proprietary appropriation. I identified four tactics used by community managed projects to protect their intellectual property. Namely, they: a) adopted software licenses with distribution terms that restrict appropriation; b) encourage compliance with licensing terms through normative sanctioning in online public forums; c) trademark the brands and logos designed to represent their work; and d) transfer individual property rights to their foundations.

⁴⁵ The Webservice Project's license did not restrict proprietary appropriation of their work, provided that the project was given due credit. Their license is thus more lenient than the GNU GPL in allowing their code to be integrated with proprietary work. However, this project did use the other three tactics discussed in this section.

a. *Licensing Terms.* Three of the four projects use the GNU GPL, discussed in Chapter II, to license their work. To return to the most critical feature of the GNU GPL, article (2)(b) prevents free software from being appropriated by requiring derived works to be redistributed under the same terms.

You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all to third parties under the terms of this License (*Free Software Foundation, Section (2)(a), GNU General Public License, Version 2, June 1991*).

Project founders reported that they adopted the GNU GPL, explicitly with the protections offered by Section (2)(b) in mind.

Well, [GUI Desktop Project] was GPL'd mainly [to] enforce, to protect software that gets developed from developing proprietary features. (*Sponsored Contributor, GUI Desktop Project*)

Contributors were very much aware of the conditions of the GNU GPL and used it to protect their work.

The question is, how can these licensing terms be enforced? How can users who do not necessarily engage in transactions of exchange⁴⁶ establish agreement with these licenses? One way to establish agreement between users of GNU GPL software and the license terms is to make the freedoms to use, copy, and modify the software contingent upon agreement with the restrictions associated with it. For example, the terms of the license hold that:

⁴⁶ It could be argued that users who do not purchase or transact to use the software would not necessarily be bound to the terms of the license. Legal scholars such as (Lee, 1999) have argued that use and acceptance of the terms may be considered adequate consideration.

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this license. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it. (*Free Software Foundation, Section (4), GNU General Public License, Version 2, June 1991*).

These terms explain how agreement between public use of community managed software and the terms of the license could be legally established without a formal exchange, but not how they are enforced.

Informants enforced their licenses two ways: formally and informally. The formal enforcement mechanisms community members used to alert violators of their license were not so different from those used by firms. Contributors to community projects relied upon legal counsel and traditional mechanisms to enforce compliance with their licensing terms.

Interviewer: How do you plan to monitor and enforce the license?

Informant: Well, if you break the license, then you get a letter from the lawyer. It's a standard copyright violation. If you break the license, you have violated the copyright, just as if you had stolen music from Napster or illegal pirated DVDs. (*Volunteer Contributor, Webservice Project*)

Legally, copyright holders are the only ones with the right to enforce the licenses associated with their work (US PTO, 2002). However, informants felt that the task of monitoring and identifying license violations was a shared one. The institution founded primarily to help enforce the GNU GPL, the Free Software Foundation, encourages those who suspect license infringement to bring it to the attention of those with the rights to formally take action. "(R)eport it (license infringement).

First, check the facts as best you can. Then tell the publisher or copyright holder of the specific GPL-covered program.” (www.gnu.org/licenses/gpl-faq.html).

The GNU GPL conceived of in 1985 continues to be applied as a protective measure to preserve the continued availability of free software. It ensures that software licensed under its terms will be preserved indefinitely for the use of those who have contributed (as well as those who have not) by protecting it from appropriation. Although, firms may use and extend community software, they cannot close their code or make it proprietary.

Because of (2)(b), each contributor to a GPLd project is assured that she, and all other users, will be able to run, modify and redistribute the program indefinitely, that source code will always be available, and that, unlike commercial software, its longevity cannot be limited by the contingencies of the marketplace or the decisions of future developers. (Moglen, 1999)

Lee’s (1999) legal review indicates that if open source software lacked intellectual property protections, that there would be little or no legal way to prevent the closing of source code. This particular feature of the GNU GPL may be less attractive to firms that would like to make proprietary changes or extensions to GNU GPL code. For community managed projects, it functions as a preservation tactic that enables them to prevent the appropriation of their source code for private purposes by prohibiting certain types of usage and behavior while still allowing it to be freely distributable. It legislates temperance and compliance with community norms. The formal rights associated with the GNU GPL were well understood by informants whose projects adopted these terms. The bulk of compliance tactics were however more informal and based on normative controls rather than the formal controls the projects had the right to wield.

b. *Normative Control through Public Forums.* Informally, the mechanisms community members use to enforce the terms of their license are quite different from the mechanisms used by firms. First, although the GNU GPL is a legal mechanism, it was intentionally designed to codify a strong norm of reciprocity that has long been an important part of the programming culture (Levy, 1984/94; Stallman, 1999).

The idea of the GPL is that if you want to include our code in your program, your program must also be free software. *It is supposed to put pressure on you to release your program in a way that makes it part of our community.*" (Stallman, www.gnu.org/licenses/gpl-faq.html) (Italics added).

The GNU GPL is understood by informants as a normative mandate to encourage participation that has the added attraction of being legally enforceable, if violated. Legal scholars (Lee, 1999; Moglen, 1999) have also viewed the GNU GPL as a mechanism that 'cements' the cultural norms and values of the community and a way to counteract commercial pressures that might violate community norms.

However, in the eyes of informants, the GNU GPL's strength does not necessarily stem from its legality, but from the very public collective opinion of project members.

Interviewer: How do you enforce and monitor that [giving back to the community]?

Informant: So far no one has had to litigate....And also a very big thing that I have used, is they fight it out in the court of public opinion.

Interviewer: Postings?

Informant: Yeah which sometimes get me in a little trouble... (*Former Volunteer, Project Founder, Sponsored Contributor, Linux Distribution Project*)

The court of public opinion is held through Internet based online bulletin boards or mailing lists. Informants stressed that the primary vehicle by which they could enforce the terms of their license was by identifying and critiquing violations on online mailing lists and bulletin boards. Both projects maintain a number of mailing lists to which contributors, users and other interested parties post to on a frequent basis.

For example, the Linux Distribution Project maintains over 150 mailing lists that include 16 different languages. The largest mailing list, for project related announcements, increased from 5,000 members in Q3 of 1998 to 22,331 members in Q1 of 2002. The list that is most central to development currently has 1,863 members. Licensing and other legal issues became so frequent in this community, that a separate mailing list was created explicitly for legal issues 2 years after list data began to be actively recorded. Participation on this list has grown to 362 members over the past three years. In addition to project based mailing lists, most informants also reported that they participated in non-project discussion groups⁴⁷ where they discuss host of issues relating to their project, license terms and potential violations.

There are a couple of theoretical reasons why online discussion forums might have a strong normative effect on community members. First, postings to project mailing lists are public and not anonymous, so communications between any two members are available to all members. Prior research suggests that individuals contributing to a collective good do so interdependently and that they are affected by the nature and structure of their network relations (Fernandez and McAdam, 1988;

⁴⁷ Examples of these forums include: Slashdot, freshmeat, kuro5in, linuxworld, linuxtoday, linux journal.

McAdam and Paulsen, 1993; Gould, 1993; McAdam, 1996). The presence of others can motivate further contributions, despite free rider concerns (Guttman, 1987; Gould, 1993). Norms of fairness and efficacy can inspire individuals to contribute when other individuals in their network contribute (Gould, 1993). License violations committed by project members or others are thus highly visible and open to public sanctioning.

From an economic perspective, it is information, and not norms that can lead to a positive matching effect (Guttman, 1987). If people have reliable information on the contributions of others, then they may be more likely to “match” those contributions and contribute as well. Both sociological and economic explanations suggest that any mechanism that enhances the visibility of both norms and information might have a positive mutually reinforcing effect on encouraging individuals to contribute to collective goods. In an online context, information on not only the presence of others, but their behaviors is communicated in a way that may exacerbate behaviors associated with the presence of social networks.

Legal scholars such as Lessig (1999) and Lee (1999) have also argued that cultural norms are critical in regulating cyberspace behavior. However, one might wonder if such forums would have an effect on commercial actors that face greater economic incentives to violate the community’s licensing terms? Online forums may also be a less relevant reference public for firms than it is for community members. There are several ways to assess this. One way is to assess whether the community’s licensing terms and methods of enforcement are noticed by firms. A second is to ascertain whether such forums affect firm behavior. While such a thorough examination is beyond the scope of this paper, informants that represented Fortune

500 firms responded that their firms were well aware of both the need to comply with the licensing terms and norms of the community.

Informant: It would be nice for it [a Fortune 500's policy on open source software] to be a social contract, but I would be satisfied if it were to be good citizens of the open source community.....

Interviewer: How can they [Fortune 500 firm] be good citizens?

Informant: Don't break the GPL. Make open source software that we are using, support Linux across our lines, support open source drivers. Don't do closed source drivers. (*Fortune 500 firm representative, former open source project leader*)

Informant: We have been very careful to work as good citizens with the (---) group and the (---) people. At some point maybe our reputation will fall apart from the past and we will not have to worry so much about it....the main way is to be careful that we are open with them and if they don't like something, to fix it rather than just to say too bad. (*Senior Research Scientist, Fortune 500 firm*)

Informants thought that online public forums were an effective and powerful tool to ensure that others (in particular, firms) that used their software, did so in a way that did not violate community norms and licenses. This is suggestive evidence that commercial actors took this into consideration.

c. *Protection of Brand.* All four projects developed distinct brands and created logos to represent those brands. They also filed trademarks with the US Patent and Trademark Office (US PTO) to protect their brands. One project even filed for trademark rights in Europe. A trademark is a word, name, symbol or device that is used in trade with goods to indicate the source of the goods and to distinguish them from the goods of others (US PTO, 2001).

Why did open source projects worry about protecting their brand?

Contributors felt that this was particularly important as their software was

repackaged and redistributed by firms. Firms have the right to adopt open source software and repackage it to better fit their needs or the needs of their customers. While, it is in theory, legally possible for firms to effectively re-brand the work produced by particular projects, project contributors preferred to establish what marketing researchers might call co-branded relationships. When a product or service is co-branded, two contributing brands to the product or service are recognized.

Co-branding was attractive to contributors to open source projects for several reasons. First, contributors wanted to ensure that they received appropriate credit and recognition for their work. This required firms that were selling their software to incorporate appropriate acknowledgement of the project's contributions and of their trademark(s). When firms failed to do this, community members were unhappy and viewed that as a sign of unfair collaboration.

The only place you found in [firm name]'s [product], that said the word [Linux Distribution Project] was on the first page of their manual, we would like to thank [Linux Distribution Project] for their help. If you did not know what [Linux Distribution Project] was....Is that a group? A people? A piece of software? I would like to thank [Linux Distribution Project]– was that your script team? (with disgust) That was all they gave us. There was never a [Linux Distribution Project] logo, there was never a [Linux Distribution Project] anything. (*Sponsored Contributor and former Volunteer Contributor, Linux Distribution Project*)

In the quote above, the informant is concerned not about recognition for himself, but for the *project*. Individual recognition may be important for contributors within projects (Raymond, 1999; Lerner and Tirole, 2000), but in a commercial context, what mattered to informants was ensuring that the project, as a whole, received the appropriate recognition. The other insight that is apparent from this quote is that a

firm's acknowledgement of a project's efforts was a necessary, but not sufficient, step toward maintaining collegial firm and community relations. It was important for project members that the project brand, not contributors' names, be incorporated into the firm's product.

The second reason that the collective brand was so important was more pragmatic. Projects wanted to be recognized, but they did not want their work to be confused with that of firms that were distributing their work, or with other 'imposters' or hangers on who might try to create the impression that they were associated with the project. This was partly a matter of maintaining the project's identity and boundary and partly a matter of wanting to ensure that attributions, both positive and negative, for specific works were clear. For example, contributors did not want problems that might be related to a firm's product to be incorrectly associated with their code.

I also created at some point, because people were making them wrongthe CD vendors were originally making the masters and I decided that [the Linux Distribution Project] should create the masters because they were getting it wrong and we look bad when they do it. They made CDs that would not install for some reason. People would write to [us] and it would not be [our code]. So [the project] started making official CDs. And I attached the [project] trademark which, thank goodness, is a real trademark. I said you may call your product official [Linux distribution project] if it is made from our CD masters and if you include the source CD and the binary CD. And at that time there were only 2 CDs and that was great because all of the CD manufacturers wanted to be able to say that there's [CDs] was official [Linux distribution project]. So they just let us do all the mastering. And of course mastering did not cost anything. (*Former Volunteer Contributor and Project Leader, Linux distribution project*)

Creating a CD master allowed this project to retain some degree of control over the representation of their work and allowed them to protect their collective reputation. At the time of this writing, 126 vendors in 32 countries sold this Linux distribution⁴⁸.

The use of brands and trademarks by community managed projects suggests two things. It suggests that the collective reputation of the group may matter as much, if not more, to volunteer contributors than any one individual reputation. While several theorists have emphasized that recognition and one's reputation may be a critical motivator to inspire individuals to contribute to open source projects (Raymond, 1999; Weber, 2000; Lerner and Tirole, 1999), informants often talked about protecting, not their individual reputations, but the collective reputation of their project. Secondly, it suggests that community managed projects value and protect their identity as a community of developers as much as they do the results of their collective efforts.

d. *Reassignment of Individual Rights to the Collective.* Under US copyright law, copyrights are most effectively enforced when registered with the government (US PTO, 2002). Intellectual property rights are generally recognized as better protected when owned by a single legal entity as opposed to a nebulous collective. An open source project that is instantiated through mailing lists and web pages does not have any legal rights and cannot sign contracts or hold property. Non-profit foundations helped correct this imbalance, but they also provided a mechanism to hold the project's intellectual property.

So [the foundation] owns the [Linux Distribution Project] trademark or holds or owns and holds the trademark but [Linux Distribution Project] are

⁴⁸ Fifty-two percent of vendors distributing this software offer to collect monetary donations from their customers for the project.

the ones that say this is where we need to enforce it, this is where we let it go, this is an official logo, this is an unofficial logo and so forth. (*Volunteer Contributor and Board member, Linux Distribution Project*)

Problem is to defend the trademark, we should first incorporate, get money then and then try to register it for software, [the GUI Desktop Project] trademark. We could still try to claim trademark while getting the foundation incorporated and put it on the web site. (*GUI Desktop Project Board Meeting, November 28, 2000*)

The individual rights of developers over their work are transferred to the collective if they assign their copyrights to the foundation. The Standard Setting Project is developing its copyright policy and the Linux Distribution Project does not require individual contributors to sign over their copyrights. However, informants who contributed to the GUI Desktop Project and the Webserver Project signed over their copyrights to their foundation⁴⁹

The Free Software Foundation was the first foundation to adopt this policy. They extend copyright assignment from contributors one step further by also asking them to verify that their employers do not have rights to the code they are contributing.

In order to make sure that all of our copyrights can meet the record keeping and other requirements of registration, and in order to be able to enforce the GPL most effectively, FSF requires that each author of code incorporated in FSF projects provide a copyright assignment, and, where, appropriate, a disclaimer of any work-for-hire ownership claims by the programmer's employer." (Moglen, 2001).

⁴⁹ It is important to note that copyright policies vary by project. For example, individual contributors to the Linux kernel do not transfer their copyrights to any legal entity and there is thus no one single copyright holder of the code for the Linux kernel. The Linux kernel project leader argues that this is can also be a source of strength, as it means that a license change would be effectively impossible to negotiate among the many copyright holders that have contributed to the project (*Interview with Linus Torvalds, March 12, 2001*). This approach makes it effectively impossible to establish a single owner is an alternative to protecting projects by creating one specific owner. The Linux kernel project also does not have a foundation established behind it to which rights could be transferred.

Having a single and central copyright holder enables projects to more easily defend the copyright rather than requiring the swift cooperation of all contributors. It establishes one clear-cut owner of the copyright. It also helps to reinforce institutional as opposed to individual liability⁵⁰.

Work that is licensed, branded, and trademarked is usually done so to protect against unauthorized use, but community managed projects use these techniques to protect against unauthorized appropriation. Lee (1999) argues that open source licenses transfer intellectual property rights from the licensor to the licensee. An alternative conception is that the rights and privileges associated with these intellectual property mechanisms are unbundled and redistributed. For example, the GNU GPL grants the rights to use, modify, distribute, and perform to the licensee. However, these rights are not unbound, as licensees are not allowed to restrict access to the code or impinge on the freedoms of others to make use of the code. Licensees are granted the right to sell the community's work for a profit, but they must do so in a way that distinguishes the community's work and acknowledges their contribution.

This redistribution of rights would appear to shift the balance of power from the licensor to the licensee. What is unique about these innovative applications of traditional intellectual property rights is that, the rights retained by the licensor (community managed projects) are those rights that are most closely aligned with the licensor's values: the perpetuation of open code. Community projects use legal techniques traditionally used to restrict access to preserve access. However, the real

⁵⁰ Some legal opinions do not think that the foundation's protection from individual liability for the software would hold up in court unless individual members assigned their copyrights to the foundation.

strength of these mechanisms lies in the ability of a community to normatively enforce and sanction compliance with these conditions.

This section explained how community managed projects adapted their form to better accommodate the needs of commercial actors and markets, but also how they made changes to preserve their founding values. Through the processes of accommodation and preservation, a new organizing mechanism was formed: a private non-profit foundation. Chapter VI discusses how firms adapted to and resisted changes associated with the community development model. In Chapter VII, I compare how the processes of accommodation and preservation by both community managed projects and firms directly contributed to the design of these foundations and describe in more detail the new negotiated order that has emerged between community projects and firms.

Chapter VI. Firms

Firms also changed some of their work practices to accommodate the norms of community managed projects while trying to influence the evolution of projects and preserve their market influence. The firms discussed in this section refer to firms that approached or worked with the four projects examined as a part of this study. In the sample of 70 informants, 18 open source firms were represented including 8 of the 53 (15%) open source companies that received funds from venture capitalists. In addition, informants from three Fortune 500 firms that were dominant players in their markets worked with the projects in this study.

To participate in community projects, firms had to learn how to partner with a community of mostly individual volunteers as opposed to a firm. On all four projects, firms changed practices that were customary for collaborating with other firms but violated the norms of community projects. Given that community projects used alternative organizing mechanisms to develop and promulgate nonproprietary software, firm representatives recognized that community projects might not positively receive a large and highly visible corporate presence. Informants from firms who approached community projects spoke of the fear of being perceived by the community as “taking over”, “dictating”, or being “too aggressive”. These terms were also used by long standing members in the community who articulated similar concerns.

One of the things initially the executives at [Fortune 500 Firm] wanted was to have several core members. The word back [from project members] was, ‘you start giving us code that is meaningful and significant. We don’t want a

lot of your people contributing code, because that will overwhelm the group'.
(*Fortune 500 firm sponsor and contributor to Webservice Project*)

The tension between wanting to participate with community projects and the awareness that traditional tactics used in the commercial world would not be welcomed set the stage for firms to identify a new set of collaboration practices tailored for community based organizations as opposed to firms⁵¹.

A. Accommodation

Several of the practices that firms changed helped reduce the appearance of their size and power in order to gain the trust of project members. Firms changed to accommodate the norms of community projects by: 1) allowing their employees to contribute on individual terms; 2) releasing their employees from assigning the rights to their work to the firm; 3) releasing code that was once proprietary; and 4) adapting their work style to a new context of interaction. These changes are summarized in Table VI-1.

1. Individual Representation. Community project members were not all initially receptive to the idea of working with firms. The donation of code or sponsorship of a programmer was welcomed, but project members worried about the degree of firm influence on technical decisions. While the specific definition and

⁵¹ Smaller start-up firms founded by long standing community members did not face this problem to the degree that larger, older and more established corporate entities did. This may be because size was a proxy for power in the eyes of community members. Larger firms would thus be regarded as more likely to violate community values as they held and could exercise greater market power. However, tensions in initiating and maintaining a relationship between community and firms still existed in small and open source firms, but to a lesser degree. Those who were once volunteer members and founded open source firms were more familiar with community norms, but, similar to larger and Fortune 500 firms, open source firms still had to reconcile their new status as profit seeking entities within the context of a community project.

requirements of membership varied by project, experience working with the community, significant contributions of code, and the earned respect of one's peers were minimal requirements. Project members stipulated that individuals sponsored by firms who wanted to work on community projects had to be allowed to do so as individuals, not as representatives of the firm.

I think they understood this effectively that you know [Webserver Project] was not an industry consortium, right? It was a collection of individuals, so when an individual [Fortune 500 Firm ABC] engineer got core commit access, if that individual left [Fortune 500 Firm ABC] and went somewhere else to work on [the project], they would still have the same status within [the project]. And [the firm] would have to get someone else.” (*A Founder, Webserver Project*)

Informants on three projects worried that if they allowed firms to become project members, then their communal organization would be at risk of becoming a consortia or alliance of firms⁵².

Firms recognized that their size and power could be liabilities in this context. Despite the legal advantages that corporations enjoy, in this context, there were advantages to appearing to be more 'person-like' in order to negotiate access to a community software project.

Interviewer: Are you on [Standard Setting Project] as a representative of [the firm]?

Informant: No, and in fact, when I was employed, it was kind of stated to me that I'm expected to be kind of independent from them. (*Sponsored Contributor, Standard Setting Project*)

⁵² Two projects (the Standard Setting Project and the Webserver Project) now allow firms to become represented on their foundation's board.

Large Fortune 500 firms de-emphasized their size and economic power by initially limiting contacts with community projects to key individuals or small groups. This helped them appear less threatening to the project's independence and status as a community of individuals. Firms modified their approach to establishing partnerships in order to demonstrate their willingness to work with communities on their terms and to build trust in their intent.

We sent out a company wide memo before we announced saying that any [Websserver Project] staff had to go through, initially it was one person that was allowed to contribute. Then while we were negotiating we got a good feeling that we were going to get an ok, and we got an ok to have one guy start contributing. (*Fortune 500 firm Sponsor, Contributor to Websserver Project*)

In addition to segmenting smaller groups of employees to interact with community projects, firms released their employees from the obligation of representing their employer, by allowing employees to contribute to community based projects as individuals on the same terms as any other volunteer.

In practice, the de-coupling of an individual's obligation to represent their employer emerged as circumstances arose that challenged the traditional notions of what employers and employees owed each other. Employers who hired people who were formerly volunteers on projects had expectations of them that people who became sponsored contributors did not always feel were appropriate for working in the community. For example, initially one firm wanted to claim credit for the contributions of the project members they sponsored.

[The firm] had people working at [the project] attaching a disclaimer to their work "donated to by [Fortune 500 Firm ABC]". I told them I was not going

to attach that. Now they don't do that anymore. (*Former Volunteer, Sponsored Contributor, Fortune 500 Firm ABC, Webserver Project*)

Sponsored contributors at this firm resisted attaching a statement of their relationship with their employer to every email. Since email is the dominant communication medium, such an attachment could become a very public and salient reminder of an individual's employment status. This idea was not well received and it did not last long, as the informant above indicates. Contributors to community projects sponsored by firms helped firms gain entrance to community projects, and in defining their new role, negotiated the terms of a relationship that bridged community and firm norms. The protocols for the attribution of sponsored contributor's work were redefined for legal reasons as well.

2. Work Assignment Rights. The need to create new protocols of collaboration between community projects and firms occasioned changes in another aspect of the employment relationship: the treatment of employees' rights to their own work. When employers sponsored staff to work on community projects, they were confronted by a unique challenge – did the firm have a claim on those contributions? If firms attempted to establish a claim of ownership over the code contributed by their employees to community projects, neither their employees nor community members would likely accept those terms. This would also be inconsistent with the licensing terms associated with three of the four projects.

Some informants told of refusing to give up ownership rights to their work as a condition of employment.

Informant: [Project X] is separate. It was my project, and it was not owned by [the firm]. It was my project.

Interviewer: You took that job, but you retained control?

Informant: Yes, I will not take a job where the company says they want to buy the project, [they] want to own it. (*Project Founder, Sponsored Contributor*)

Informants on all four projects that were sponsored by firms renegotiated one of the most unquestioned practices in the software industry – the transfer of assignment of their works.

Firms that hire programmers typically ask them to sign an “Assignment of Pre-Employment Works” or a “Confidentiality and Invention Assignment Agreement”. This agreement is standard employment paperwork for many technical workers. It transfers ownership of all works created by the employee on company premises and using company resources to the company. This practice is common in the software industry and in university settings, but it is unknown how prevalent it is across other industries. An example of the language of such an agreement in the software industry is below.

I hereby grant, transfer, convey and assign to [Company], its successors and assigns, all right, title, and interest in and to all work and materials relating to my software, including the copyright, patent, trade secret rights, and all other right, title, and interest therein. (*From the Internet Legal Resource Guide, www.ilrg.com/forms/assignpre-emp.html*)

Informants reported that their employers revised these agreements so that they could contribute code to community projects. Employers thus formally gave up their rights to own the code their employees produced.

What [Fortune 500 firm ABC] did, when you start working for [them], they make you sign an IP [intellectual property] agreement that says basically we own everything you do. Well what [the firm] did when we started working on

[Websaver Project], is that we said anything that gets contributed back to [Websaver Project], [the firm] doesn't own. (*Sponsored Contributor to Websaver Project, Employed by Fortune 500 Firm ABC*)

When employers did not realize this, informants brought it to their attention.

Although firms knew how community projects treated their intellectual property and agreed to comply with those norms, a few informants reported that employers were sometimes caught unawares of the need to change this agreement. Intellectual property agreements were a standard operating procedure used in many firms, a taken for granted institution, that had to be modified to reflect new social arrangements between community projects and firms.

There was just a standard paper that everyone was supposed to sign. And I looked at it and said no I am not going to sign. And we changed like five words. And basically it was adding an 'except for Linux'. (*Project Founder, Linux*)

Some programmers hired to work on community projects also had the foresight to include clauses in their employment agreements that explicitly protected their rights to freely make technical decisions on community projects⁵³. Such a clause included language that allowed programmers to disregard their employer's advice on technical matters without establishing grounds for termination. This clause, in a sense, allows sponsored contributors unfettered technical autonomy on community projects. This is a radical departure from paperwork that typically grants individual employees few rights, if any, over their own work. Allowing employees to retain ownership of their work, despite the financial support associated with

company time and resources is a significant departure from standard procedures in the software industry. Firms in the software industry consider their code to be an asset, intellectual property upon which their business model depends. But, to collaborate with community managed projects, employers de-coupled their rights to employee works from their conditions of employment.

3. Release of Proprietary Code. In addition to allowing employees to represent themselves and own the rights to their own work, firms also released code to community projects that was previously developed under proprietary conditions. Releasing code that was formerly proprietary required firms to change the way they thought about and managed their intellectual property, often times necessitating a critical review of a firm's copyright and licensing terms with other partners. To release proprietary software under a license that was compatible with open source and free software licenses, firms first had to ensure that they owned all the rights to the code. Thirty party agreements with other partners could make clear determination of the ownership of code difficult. Once ownership was determined, firms could either re-license the code or license it under a dual license with terms that were compatible with community licenses.

Firms also contributed new code to help advance the speed of development of certain projects; to help strengthen interoperability across different platforms; to gain the community's trust; and to maintain a consistent code base. Firms understood that contributing code would help them earn the trust of the community,

⁵³ It should be noted that this practice was not regarded by informants as very common practice and was likely reserved for those leaders in the community with very successful projects, a great deal of experience and/or high status.

but they also cited pragmatic reasons for not wanting to maintain a separate development stream from the community's developed code base.

Informant: My official stance is fragmentation has never been an issue in the community that actually builds Linux.

Interviewer: And why is that?

Informant: Because who, or what company can afford the maintenance of a branch system of a piece of software? I can name two, and I can also name those two that would have to put lots of resources and an incredible amount of belief, blind faith that it will pay off in a few years. There are only two companies and one of them is Microsoft. (*Sponsored Contributor, Fortune 500 Firm, Standard Setting Project*)

Economists have wondered what prevents firms and individuals from 'fragmenting', 'forking' or from using 'free' code to create an entirely new development path. As the informant above indicates, maintenance of a code base separate from the community was viewed as a costly endeavor that did not offer them substantial benefits.

Informant: [The firm] didn't want to maintain a different set of patches from the normal [project] code base. So everything we did we wanted to give back to them....Remember the ultimate goal is to sell more software. That is why [the firm] did this. They didn't do it out of good faith. They had an ulterior motive of selling more...and the closer they could keep their version of [the code to the project], they could sell it that much easier." (*Sponsored Contributor, Webserver Project*)

Firms wanted their contributions to be returned to the community's development stream and often did so, even in cases where the community license would permit proprietary changes⁵⁴. As firms became familiar with the terms of the GNU GPL,

⁵⁴ The Webserver Project was licensed under a set of terms very different from the GNU GPL, and their license does allow proprietary extensions of their work.

they realized that extensions and improvements they made to the community's code would have to be provided back to the public community maintained code base. This required a more precise definition and understanding of what exactly could remain proprietary to the firm and what could be contributed to the community.

So drawing the lines as to where we give code back and help the cause and help our business and where we keep things proprietary is always a challenge.
(*Founder, open source firm, former Volunteer, Linux Distribution Project*)

Managing the legal and technical boundaries between proprietary and community code became another aspect of firms' relationship with the community for negotiation.

4. Adapt Work Style. Sponsored contributors who did not have prior community experience also changed their software development and communication patterns to adapt to community norms. On community managed projects, one person's idea was subject to critique and review by many others around the world, all of whom operated in a plurality of computing environments. Others (Raymond, 1999; Mockus, Fielding & Herbsleb, 2000) have argued that this far flung and diverse peer review process contributes to the high quality of the code developed on community projects. Those who proffered their ideas had to be prepared to defend them. Tactics based on power or authority that might otherwise be used to shape development decisions within a firm or among a firm's partners would not fare well in this environment.

They [the firm] understood very quickly that they couldn't start dictating. And you know big companies need to learn this when you start working with an open source group. You are not dictating how everything works. You are making suggestions and you are contributing the manpower to make it work

the way you want it to. (*Sponsored Contributor, Fortune 500 Firm ABC, Webserver Project*)

If it were an inside of [Fortune 500 firm Theta] project, I would be dictating what we are going to do, instead of negotiating what we are going to do. (*Former Volunteer, Sponsored Contributor at a Fortune 500 Firm Theta, Webserver Project*)

Firm informants felt that, aside from the legal challenges, one of the more difficult changes in their day-to-day work practice was adapting to the public nature of community based software development. Most, but not all, of the project mailing lists and the IRC channels are publicly accessible. Of the many mailing lists that constitute the project environment, only a very few have restricted access.

The shift to developing software in a public forum challenged the boundary of the firm by exposing firm employees to levels of visibility in their work environment to which they were unaccustomed.

The next thing we said was, we want this thing, not only to be GPL, but to be totally developed in the open and we want our source code to be sitting in an open CVS (Concurrent Versions System⁵⁵), right? So we did all that, which was actually pretty traumatic and difficult for us, because we went from thinking of, you know, open source as this source code will be available, and trying to figure out if we can sell it, to accepting that it needed to be GPL, to accepting, much later, that the thing needs to be in CVS at all times, so at any point in time, people can see exactly what we're up to, to accepting that it was actually going to be a part of [the project] proper, which means that, you have all these [project] hackers who get to yell at you and tell you if what you're doing is okay or not, right? (*Open Source Firm Founder, Sponsored Contributor to GUI Desktop Project*)

⁵⁵ Concurrent Versions System is an open-source network transparent version control system. It is an Internet based application that allows a group of distributed developers to work on the same code base with a check-out model. Developers check code out that they want to modify and then check it back in. This prevents people from making changes to the same piece of code. The code repository is publicly available, but only those with commit access to the code base can make changes to it.

A transparent development environment was a difficult adjustment for corporate informants who lacked prior experience with community development practices. It meant that changes and mistakes would be made in a setting that was publicly available and furthermore, that a much broader and more diverse audience would feel free to comment on those contributions regardless of their position within a hierarchy.

Firms interested in participating in community projects navigated through the legal and socio-economic institutions that distinguish firms from individuals. They adapted to the idea of cooperating with a loosely defined collective until community managed projects incorporated. Firms used several tactics to reduce the appearance of their size and power and appear more like the individuals that had originally formed these community projects. They allowed their employees to represent themselves and to own the rights to their work, released code that was proprietary and shifted to a more public forum for software development. Not only did members in the community notice that firms were changing traditional practices to collaborate with community projects, but so did industry analysts who watched these changes transpire very closely.

[Fortune 500 firm ABC] was very wise in realizing that they need to really position themselves effectively in the community given what a large company [Fortune 500 Firm ABC] is. There was a thousand pound gorilla, and they did not want to be perceived as driving Open Source, but just being a participant in the community. They very astutely figured out how to work with the community, rather than feel like they were driving the community. That is a very difficult thing to do. They make it look easy, but it is not easy. They obviously did enough due diligence to realize. (*Industry Analyst, Open Source Software Market Segment*)

The firm this analyst is referring to did, in the eyes of informants, adapt to community managed development norms better than any other firm. Informants from this firm agreed that these changes were difficult for the firm to make, but were glad that the community projects they were working with recognized their efforts.

It could be argued that the adaptations made by firms to work with community projects are the result of a set of rationalized trade-offs. Control and ownership of intellectual property is traded for access to technical talent and the ability to draw upon a constantly evolving code base. However, these changes were not made in nearly as a coordinated fashion as a retrospective accounting suggests. Firms did not approach community projects with a list of itemized practices they knew they had to change in order to participate in community development. The changes that informants discussed emerged over three years of dialogue and interaction. Informants from firms who pursued relationships with communities spoke frankly about 'not knowing what they were getting into' and their unawareness of the second order effects that community participation would require. Thus, these changes must be understood as the product of firms learning how to interact with a new community of practice (Wenger, 1998).

B. Preservation

Making changes in their work practices to collaborate with a community project did not preclude firms from wanting to influence the project's technical direction and management. As discussed in Chapter IV, firms were attracted to the talent and expertise of community members, the quality of their work and, most importantly, the market share projects managed to accumulate. But as firms began

integrating community developed code into their business models, they discovered that they had significantly less control over the direction and timing of development in a public community forum than they did with their internally developed proprietary software. In figuring out how to incorporate code initially created by and for individuals into commercial products, firms met with several challenges.

What they [Linux Distribution firms] are frustrated about is that they have so much trouble having any kind of influence on the development and direction of Linux. (*Sponsored Contributor, Standard Setting Project*)

One challenge was working with a group whose progress could be uneven.

Volunteer contributors did so in their free time, thus tasks were picked up and dropped depending on an individual's availability.

Most everyone is part-time on it [sub-project]. They fall behind because it is voluntary. That's the difficulty of working with a voluntary work force. (*Sponsored Contributor, Standard Setting Project*)

Uneven progress was not a problem for volunteer contributors as the code they created was for their own use. Informants eschewed deadlines as something that was associated with 'real work', this they did for fun. Deadlines violated the hacker's ethos and their occupational mandate: code was ready when those working on it felt it was ready and not before. Firms that were used to more predictable progress found themselves in the uneasy position of relying on the work of volunteers to meet the demands of the market. They essentially became interdependent with groups whose work was managed independent of market rhythms.

A second challenge occurred when project members made substantial changes to their software that affected software firms were developing. The coordination of code revisions among developers was typically managed via the check in and out of code through applications such as CVS. Using such a system, contributors around the world could work at any hour of the day or night revising code without working on the same code. However, because code revisions were not centrally coordinated, firms that relied upon certain application interfaces or ‘hooks’ to remain stable could become frustrated with the frequency of these revisions. This ‘instability’ in the code base could hinder firms from meeting internal and market deadlines. For example, a change in the names or location of community code files could create compatibility problems for large sections of firm developed code that depended upon those file names and locations. This might necessitate the need to rework a firm’s code and affect their ability to meet internal or customer deadlines.

If you're a company like [Open Source Firm #1] or [Fortune 500 firm], or [Open Source Firm #2], and you need to raise money, or meet quarterly objectives, you actually need to have a release schedule and a road map.... You come to work sometime on a Monday and they changed all the method names, you've [got to] change all of your software because they just totally changed the world from underneath you, right? It's a major problem for a company. (*Sponsored Contributor, GUI Desktop Project*)

The quote above suggests yet another factor that affected firm and community interactions: different patterns of work time. Volunteer contributors and sponsored contributors who worked even within the same time zone might work on the project at very different times. “Coming to work on Monday” to discover a great volume of changes could easily occur because of the different temporal structure of volunteer work. Volunteers were more likely to work on their projects after normal business

hours⁵⁶, while sponsored contributors might be more likely to work on their projects during business hours⁵⁷. Thus, not only did work patterns fluctuate at the whim of volunteers, but also, contributions could be heaviest when sponsored contributors were less likely to be working; leaving them constantly in the position of catching up.

The challenge for firms was to adapt their work practices to the norms of the community and yet introduce enough stability and predictability to the development process so that they could manage their own ability to meet market demand. Some saw a firm's need for a more stable and predictable development cycle to be inevitably at odds with the community's norms of unrestrained software development on individual terms.

And this is again part of the natural, there's an inherent and necessary conflict between an economic entity, like [Theta Corp.], and the community. (*Product Manager, Fortune 500 firm, Sponsored Contributor, GUI Desktop Project*)

Other firm representatives were more sanguine about the nature of this conflict and their ability to influence community managed projects. They recognized that by choosing to work with a community project, they chose to work on the community's terms, but they did not view this interdependence as necessarily resistant to management techniques, just requiring different ones.

Maybe the farmer has no control over the weather. But that doesn't mean they have no control over their farm. Right? So we, you know we read the weather reports, we do our best to make sure to make sure that whatever the weather is, we are well positioned and that expectations are properly set. And

⁵⁶ Some volunteers did manage to work on their community projects during business hours although it was not an authorized part of their jobs.

⁵⁷ This was more likely for sponsored contributors that came from a corporate background as opposed to sponsored contributors who were once volunteers. Sponsored contributors that were initially volunteers may be equally likely to work during non-business hours.

you know the crops actually make it to market. (*Chief Technology Officer, Open Source Firm; Supporter of GUI Desktop Project*)

The analogy this informant uses compares the degree of control that firms have over community software development to the degree of control that farmers have over the weather. The second part of the analogy, suggests that good farmers develop tactics to influence what they can. From the perspective of this Chief Technology Officer, firms could develop tactics to buffer them from the unpredictability of the community environment and to influence the direction and practice of development work - just like farmers attempting to maximize their harvest despite the vagaries of the weather.⁵⁸

Since the typical tactics firms might use to influence other firms could be ineffective in this context, several firms developed tactics for managing this lack of control. Firms wanted to preserve their market power by gaining greater predictability in the software development cycle and influencing project priorities to better fit with their products and services. I identified three primary tactics that firms used to accomplish this: 1) sponsor contributors in key roles; 2) donate resources to the project; and 3) sponsor or help shape the foundation. These tactics

⁵⁸ Another informant supporting a different project also described his firm's rationale for contributing to open source projects with an agrarian analogy.

We view it as watering our garden: that we have to take care of the open source part of this project as much as our business. (*Open Source Firm Founder, Former Volunteer, Webservice Project, March 13, 2001*)

The farmer, weather, and garden metaphors suggest a long-term decision horizon between investment outlays and outcomes, a perspective on investment that venture capitalists might term 'patient capital'. A long-term time horizon and a less explicit, if not weaker, tie between outlays and outcomes suggest a degree of patience not typically associated with traditional organizing mechanisms, such as alliances and joint ventures, used by firms to collaborate.

are summarized in Table VI-2. These phenomena, while evident to some degree on all four projects, was more profound on those projects that worked most closely with many different firms and provided firms with the highest degree of access to their organization, namely the GUI Desktop and Standard Setting Projects.

1. Sponsor Contributors in Key Roles. The most effective and most used tactic to exert influence on a community project was to contribute manpower by hiring or sponsoring existing project contributors. The previous section discussed changes firms made in their employment practices to sponsor contributors and also maintain their relationship as members of a community. Volunteer contributors who were hired to continue their work on community projects negotiated these accommodations prior to employment. Current employees that became community members modified their relationship with their employers, particularly with respect to intellectual property rights, *ex post*.

Why did firms give up some of their rights as employers? What did firms hope to gain by sponsoring contributors to work on software that the firm would never own or have rights to? Individuals with links to the community or prior experience on community projects were valuable to firms who wanted to partner with community projects: they could teach firm employees how to work within the community and provide deep technical knowledge of the project. All four projects had some sponsored contributors in leadership roles working on them and sponsored contributors on all four projects indicated that their corporate sponsors influenced the priorities of their work on the project to some degree. It is important to note, that the word influence is used here to indicate “the act or power of producing an effect without apparent exertion of force or direct exercise of

command” as opposed to the more pejorative “corrupt interference with authority for personal gain” (Merriam Webster, 2002).

Firm influence channeled through sponsored contributors occurred on two different dimensions: changes to the software itself and changes in the community’s development practices. Informants that were sponsored contributors reported that their employers asked them to shift their priorities; push innovations to improve the functioning of the program; or add improvements to expand the program’s interoperability. Informants spoke most frequently of being asked to shift their priorities to work on aspects of the project that were most important to the firm. Although, they recognized that their employer’s wishes could affect their work, they did not perceive this to be a direct threat to their technical autonomy.

Informant: I think, [Open Source Firm] does not have the ability to... they're not going to say this is the [---] feature list, you know.. they don't have any real influence on what [the program] will do as far as what features it will have or when it will be released. But what they can do is that [they] can ask, on company time, of course, that we work on one thing or another thing because that's the thing that they would like to see move forward.

Interviewer: So they can help, the distinction you're making is they can help prioritize something, or divert energy to it, but not necessarily change the requirements... how it's going to be shaped?

Informant: Exactly, I mean, you know, this is a public project. The goals for that are discussed in public, they're made you know, by, the community, the group of people that work on that project. And so that's not controlled by any company, but if there's ten features that we want and some company actually implements one of them and no one implements the other nine, that one will be the one that got implemented, right? You see what I'm saying? So the way that companies can sort of influence things are by saying these are the things that we think really need doing, do those things instead of the other things that would also be good. (*Sponsored Contributor, Board Member, GUI Desktop Project*)

This informant viewed the shifting of priorities as an almost natural consequence of agreeing to work for an employer on a community project. However, he was not concerned about influence over the project's technical direction, as a firm's ability to do so would be 'checked' by the plurality of the community in public forums. Thus what firms could influence was the level of effort and ordering of priorities in particular areas of the project.

Sponsored contributors on all projects were careful to make a distinction between 'priority setting' and direct influence on a project's features or goals.

Priority setting was viewed as an acceptable degree of firm influence.

The blueprints can change if the company or individual is willing to put the effort in...However, only to the extent that the rest of the community agrees to those principles...We have 100 line items that need to be done. I am drawing an analogy here. We have 100 line items that need to be done. If you are a member [of the foundation] you can arrange the order of those line items as one form of influence. (*Sponsored Contributor, Fortune 500 Firm ABC, Standard Setting Project*)

Shifting the priorities of a project could hasten its progress in a direction that was desired by a firm, but since all community members wanted to see their project advance to commercial grade, this type of influence was not perceived as a threat to the project's independence or their individual autonomy. Many saw the sponsorship of contributors as supporting the project's progress. As long as changes in project priorities benefited both the project and the firm, sponsored contributors did not feel as though their technical autonomy or the community's norms were violated.

Informant: There were some changes because I looked at what [Open Source Firm] wanted to do with the [Linux Distribution Project] which included putting it on our servers. The [Open Source Firm] didn't say you have to go do this or that directly, but it did influence my goals. I ended up working on [---] which isn't actually being used to the full extent in either [the project] or

[the firm] but one of the reasons I was pushed that way was because it will help [the firm] besides helping [the project] a lot... So you know you end up being pushed a little bit in different directions even if they don't say 'go work on this'. (*Former Volunteer, Sponsored Contributor, Linux Distribution Project*)

Informants who identified with the open source frame viewed attention to commercial priorities as important to expanding the relevance of their project to larger commercial audiences. Firm contributions that expanded the number and types of applications that could be compatible with a community software project could enhance its market share.

Corporate influence on the progress of community managed projects was not necessarily regarded negatively. In fact, informants viewed the efforts of firms in sponsoring and directing contributors on two projects (Websserver and GUI Desktop Projects) as critical to enhancing the performance of both projects and helping them build important releases.

Without [Fortune 500 Firm ABC], sorry there would not be a [version 2.0] right now. There just would not be, and I don't think there is anybody who would disagree. It is because [Fortune 500 Firm ABC] got involved that 2.0 had somebody pushing it, which was what it needed. (*Sponsored Contributor, Employee at Fortune 500 Firm ABC, Websserver Project*)

I don't think [Open Source Firm] puts any pressure on Linus to, you know, do a release. Whereas, in the case of [GUI Desktop Project], there was major pressure from relatively early on. The reality is [GUI Desktop Project] would not have been there if not for [Open Source Firm], right? Because it was all these guys with this energy and then [Open Source Firm] came along and hired five of them and that's how we all got, you know, to become the real thing, right? So [Open Source Firm] played a key role in the creation, in the initial stages of [Project GUI Desktop]. (*Former Volunteer, Sponsored Contributor, GUI Desktop Project*)

Firms often sponsored contributors to make frequent and steady contributions that advanced projects as a whole, but all projects experienced instances where a firm

used their influence to solely advance a firm's specific need. This was regarded warily.

Code contributions that were customized specifically for a firm's product were most likely to be perceived as a threat to the project's technical independence. The more firm specific the contribution was perceived to be, the more likely that other community members, volunteer or otherwise, would regard it with criticism or suspicion; and the least likely that the contribution would be whole heartedly accepted.

If it's a good change and somebody did the work [the Linux Distribution Project] would be happy to take it. It would only be a case [for not accepting code] when the company says 'going in with this specific library' and we'd say 'that's nasty' or when the whole system is specific to your company and is not going to help others and it will affect other systems. (*Former Volunteer, Sponsored Contributor, Linux Distribution Project*)

Alternatively, working on aspects of the project that were not specific to a firm was one way that sponsored contributors could demonstrate their intentions to further the project as a whole. Sponsored contributors from one firm deliberately expanded the realm of code they were working on to areas not explicitly related to their corporate interests in order to avoid the perception of only contributing code specific to them.

They were so skeptical of us here; it could take a little while. We had to give them a little, okay, these guys aren't so bad, and they are doing cool things. They are going to work on things that we care about." So we were really trying to also fix some bugs that were normal bugs, just not [Fortune 500 Firm] bugs. (*Sponsored Contributor, Fortune 500 Firm ABC*)

Several sponsored contributors in this firm who did not have prior experience as volunteers used this tactic early in their relationship with the project in order to gain the trust of the group.

Sponsored contributors who were formerly volunteers were aware that their sponsorship might affect the community's perception of their work and their allegiances. Volunteer contributors who became sponsored were careful to minimize any perception of their employer's influence over them or their work.

Well people in [Linux Distribution Project] were a little bit confused at first whether there would be a conflict of interest. You know, whether [the firm] would push the way but by now I think they pretty well understand that I do what I do for [the project] and just happen to be on [the firm's] time.
(*Sponsored contributor, Former Volunteer Contributor, Linux Distribution Project*)

Another way sponsored contributors could reduce the salience of their firm affiliation was by changing their email address from an employer based domain to a project based domain. This way, a contributor's corporate sponsorship became less visible as it was no longer a part of every communication. Every project provided email accounts and these were preferred to engaging in the community's online forum under an employer's email domain.

The ability of firms to influence the technical direction of a project was also bounded to the degree that sponsored contributors reinforced their status as individuals as opposed to firm representatives and resisted pressures that they felt encroached on their technical autonomy.

Interviewer: What did it feel like, turning a hobby into a job?

Informant: It did not change things so much. It was fun to be in a position to, to be in a position where I can tweak [Fortune 500 Firm ABC]'s nose. But

I am a [project] person first, a [firm] person second. I will resign from [the firm] rather than buckle to pressure for some bonehead decision.

Interviewer: What is an example of a bonehead decision?

Informant: A bonehead decision is them telling me I can't work on open software or to work on proprietary software or to vote in a specific way. *(Former Volunteer, Sponsored Contributor, Fortune 500 Firm ABC, Webserver Project)*

This informant, who had a long tenure on a community managed project, and then, to his surprise became a sponsored contributor, felt as though he had enough power to dictate the terms of his employment and that he would be resistant to forces to the contrary.

The sponsorship of key contributors on community managed projects was an important tactic firms used to help stabilize a project's development and influence project priorities. On two projects, firms made substantial contributions that catapulted progress toward a new release of the community's software. Community members were also however wary and watched the boundary between shifting priorities and setting direction closely. Attempts to introduce code that was too firm specific were rebuffed as were attempts to directly influence sponsored contributors on project direction.

2. Donate resources. A second tactic firms used to gain credibility as a committed partner to the community's software development efforts, was to donate resources to support the project. All four projects received some degree of technical resources such as servers and other hardware, bandwidth and teleconferencing services from firms.

[The Project] has a very good relationship with companies. Basically if we need hardware, whenever we really need something we can usually ask for it

and get it. Companies are very willing and open to sponsor. (*Volunteer Contributor, Board Member, Linux Distribution Project*)

In addition to technical services and support, all projects received some degree of direct financial contributions from firms. The two foundations that were formed during the height of commercial interest in Linux, the GUI Desktop Project and the Standard Setting Project, also created roles for firms that required financial contributions.

Well, we've hosted the servers since the start and I guess, we're going to pay the Foundation membership fee, which is like \$10,000 or something. Sometimes we'll sponsor things, or [for] example, we flew some people out to [the project meeting] last year and that type of thing. (*Sponsored Contributor, Board Member, GUI Desktop Project*)

The GUI desktop project charges \$10,000 for membership on their Advisory Board, which is “made up of companies and organizations which have a desire to participate in advising the foundation about releases and other decisions” (Project GUI Desktop Charter, October 23, 2000), but does not include decision-making rights.

Membership is open to any companies and groups that are interested in contributing the project but is subject to the approval of the board of directors. The standard setting project created a sliding fee for three different types of members: firms, non-profits and individuals. For \$100,000, the highest fee level, firms earn the right to be on the Executive Advisory Committee and are listed as sponsors of all the work groups. Neither the Webserver nor the Linux Distribution Project created roles for firms that required dues. These two projects, both of which were founded much earlier than the other two, also did not have as many firms closely involved in their institution building process.

The Linux Distribution Project was unique, however, in its ability to collect donations without charging firms a fee to participate in a formal way. Over one hundred vendors in over 30 countries sell this project's software on CDRoms⁵⁹. Of these vendors, over half of them will collect donations from their customers for the project in the foundation's name. Informants from this project viewed this relationship as a mutualistic one. That is, they did not mind if firms collected revenues and benefited directly from their collective efforts as long as firms, in turn, supported the project through small donations of hardware and other services.

Informant: We also get CDs donated regularly from companies that burn them. For instance if we go to a tradeshow, sometimes [Linux Distribution Project] will pay to have CDs burned to give out at the show. Other times we will just ask on the CD vendors mailing list. We will ask 'can somebody provide 1,000 CDs for us for next week's show in San Francisco?'

Interviewer: And they will?

Informant: Oh sure, well they are living off us as well, right? (*Volunteer Contributor, Linux Distribution Project*)

This informant acknowledges that firms are directly benefiting (or free riding) from their work, but yet does not indicate that he perceives this trade-off to be unfair. The implication is however, that if firms were not supportive in providing resources to the project, that the aura of mutualism would be diminished. The other three projects also received assistance from firms in the form of non-technical in-kind resources such as legal counsel, marketing help, travel expenses, and assistance with trade shows and exhibitions.

⁵⁹ The actual number of vendors who sell the Linux Distribution may be much higher as vendors have no requirement to report their commercial activities to the project.

Project informants viewed commercial donations as necessary to sustain the project as well as to maintain good working relations with firms who benefited from their efforts. Corporate supporters are typically acknowledged and thanked on project websites.

It requires ongoing involvement with the corporate community to get the resources to move forward, because you can't just get it from the development community. (*Sponsored Contributor, Foundation Leader, Standard Setting Project*)

It could be argued that these donations, like many other charitable donations that firms make for various causes, could be just that, donations to express and maintain good will in their business environment. There is no evidence that direct financial or in-kind resources donations from firms affected the technical direction of any of the projects.

However, a few informants viewed hardware donations as more strategically motivated than mere expressions of good will: they viewed such donations as, supportive, but also as a tactic to either promote a firm's hardware or to motivate programmers to use and work on their hardware.

I believe it [the provision of hardware] is a motivator. It is their [software vendors] way of getting what they need without having to confront the kernel developers directly and get an adverse reaction...and it is probably not going to cost them any more money than it would have for them to devote the resources internally to try and create, to try and make Linux better and submit them as patches and cross your fingers and hope they accept the patches you know. (*Sponsored Contributor, Standard Setting Project*)

Usually donations, if a corporation donates something they don't just want to give money, they want to say, 'here I am giving you hardware and this is what it will be used for'. Because at some point they have to justify that donation to their management as well. (*Volunteer Contributor, Linux Distribution Project*)

Specific hardware donations could be used to promote a firm's brand or to 'motivate' as one informant called it, programmers to work on their machines. As the quote in Table VI-2 indicates, 'motivating' programmers to work with a particular machine provided firms with another way to translate their commercial technical requirements into a more palatable form. Firms, perhaps rightly guessed, that volunteer hackers would not be as interested in direct requests for market driven changes as they were in solving particular intellectual or technical challenges.

The provision of hardware resources to community projects can be viewed as a strategic tactic used by firms as it also helped create greater stability in the hosting of projects. Resources were not discussed problematically by any of the informants on projects. Most felt that whatever was needed was eventually provided by someone. However, that could lead to some instability as supporters emerged and drifted away.

Interviewer: Where is [Linux Distribution Project]?

Informant: It moves. It is hosted by whoever is willing to host it for us. So every year or two whoever is hosting says we cant afford this anymore and it moves. (*Sponsored Contributor, Linux Distribution Project*)

On another project, a firm offered to provide hosting services to help the project gain greater technical stability and to improve their access to it.

We moved to a volunteer machine, somewhere in Texas. And [Open Source Firm], they said *they wanted to have a better connection*, [so] it was moved to [Open Source Firm] and ever since, it has stayed there. We have the option of moving it somewhere else, but it's fine there. [Open Source Firm] doesn't mind keeping it. (*Sponsored Contributor, Former Volunteer, GUI Desktop Project*)

3. Help Shape Foundation. Firms who wanted to work with the three community managed projects that were most amenable to commercial participation (Websserver, GUI Desktop and the Standard Setting Project, valued the creation of non-profit foundations as a means to enhance the project's legitimacy in the commercial world and facilitate legal transactions. Some firms also saw the emergence of non-profit corporate foundations as helpful in encouraging project stability and the transmission of more rationalistic or managerial based practices. As discussed in Chapter V, firms were catalysts facilitating incorporation on three out of four projects. However, where legal counsel was proffered, it was not considered to be a long term proposition by either party.

[Fortune 500 Firm ABC] provided attorneys to help us do the incorporation so beyond that today we now have secured other legal services and are getting to a place where we stand on our own a lot more. (*Sponsored Contributor, Former Volunteer, Open Source Firm Founder, Websserver Project*)

This type of support had the opportunity to create conflicts of interest as a large part of the development of the foundation's charters centered on defining levels of membership and participation and the rights associated with each. One project explicitly identified this type of support as a source of potential conflict of interest and worked to move toward greater independence from their sponsoring firm.

Attorney [---] has been doing an admirable job, but will not be able to act for us on issues that might have a conflict of interest with [Fortune 500 Firm ABC], and suggested we may want to find a counsel with specific expertise in the are of 501 © (3) non-profits. (*Websserver Board Meeting, April 13, 1999*)

However, on the GUI Desktop Project, firms that were helping with its incorporation exhibited direct pressure to create a board that better represented their

interests. By assisting with the design of the non-profit foundation, this firm hoped to affect its decision-making structure and ensure that their interests were represented. This did not come as a surprise to one informant who had worked on the foundations bylaws and found himself in the position of mediating between a supporting firm and the community.

So, the corporations, they want this industry consortium that they can run and control and they can do their corporate politics in. I mean...that's what you want if you're a company. You want a standard sitting body and you want to send all your folks to it and you want to try to control the thing as much as you can. (*Sponsored Contributor, GUI Desktop Project*)

Sponsoring firms were not necessarily effective in their attempts to influence the design of the GUI Desktop Project Foundation. This foundation's board structure was revised from initially maintaining that a firm could not sponsor more than 50% of board members to 40% of board members, thus ensuring that a majority influenced decision would be more difficult to obtain.

He [project founder] didn't allow them [sponsoring Fortune 500 firm] to do some of the things they would have loved to do had they written the charter. (*Sponsored Contributor, Volunteer Contributor, GUI Desktop Project*)

Informants did not view this firm as successful in achieving its desired level of influence over the design of the foundation, nor did its governance structure permit a controlling interest to emerge. However, one informant wondered how some sponsored contributors managed to get elected to board positions over those who had volunteered for a longer time within the community.

There are people who have poured, poured their free time into this project and, he was paid to do that, [I mean people who] really love it for what it is,

instead of it being a political end to a corporate means. I think it is really sad that people voted for him [corporate representative involved in charter development elected to a board position]. (*Former Volunteer, Sponsored Contributor, GUI Desktop Project*)

Other informants confirmed that they had supported and encouraged others to support the election of a corporate representative in order to pacify one particular firm and prevent more aggressive influence tactics from them in the future.

Given these tactics, what type of influence did firms achieve? Community members appreciated the support firms offered and felt that without their corporate sponsors, their projects would not have progressed as much as they did. They provided several examples where firm support sped the pace of development. Firms were most effective in exerting their influence when they hired volunteer contributors to community projects and modified their development priorities. Sponsored contributors were often in the position of balancing competing allegiances to firms and communities on both technical and organizational issues. The role conflict that this tension engendered will be discussed further in the next chapter.

Donations and other forms of in-kind support helped firms to subtly introduce their interests to a development context that was founded and managed based on individual priorities. 'Contributing back to the community' was a norm within the community that firms knew they had to comply with. Donations were one way to reciprocate for commercial use of community developed code. This was one way that firms who were earning profits from community managed software to make the relationship more mutualistic. This support could also be used strategically

to endear firms to community members who might otherwise not have made commercial interests a priority in their development progress.

This Chapter and Chapter V described how community managed projects and firms negotiated a new collaborative relationship in software development. Changes of accommodation helped firms and communities to narrow the gap in their differences, but where those differences threatened their core identity and objectives, both actors pursued preservation tactics to reinforce their founding state. I argue that these two forces, accommodation and preservation, helped shape the design of foundations that are intended to sustain pluralism and resist the emergence of controlling interests. The next chapter will discuss these four foundations in more detail, explore some of their more interesting differences, and more explicitly link these processes to the foundation's design. It will also compare the resulting organizational form, a private non-profit software foundation, to other organizational forms and assess differences and commonalities among them.

VIII. Discussion

This thesis began by arguing that theories of institutional change required a more nuanced understanding of human agency and the political processes that help supplant established institutions. The social movement vocabulary of framing, political opportunities and resource mobilization efforts was offered as one way to integrate a more active link between individual actors and institutional change. While organizational theorists have recently begun to use social movement theory (Davis and Thompson, 1994; Davis and McAdam, 2000; Rao, Morrill & Zald, 2000; Swaminathan and Wade, 2001) to explore industry dynamics, its empirical grounding derives more from social and political activism than it does from commercial contexts.

This research bridges this gap by examining what happens when a social movement shifts its loci from one rooted in social change to one that includes commercial activity. The context of this study was well suited to examine these processes as the free software movement's primary protest tactic, the production of free software, grew to attract the interest of firms. This research shows how interactions between challengers and defenders led to the creation of new social structures. It also shows how processes of accommodation and preservation influenced the recombination of organizational elements to produce a new form of organizing: private non-profit software foundations. The foundations support a new negotiated order that mediates competing logics.

This chapter links the processes and tactics described in earlier chapters to the design of foundations and examines the emergent negotiated order among firms,

community managed projects and their foundations. The chapter concludes by examining how this order might affect community projects, the founding movement, and sponsored contributors who work in both community and commercial worlds. First, we turn to social movement theory explanations of the political opportunity structure to explain the foundations for collaboration.

A. Political Opportunity and Context for Collaboration. There were a number of factors in the political, economic and social environment that fostered collaboration between community managed projects and firms that should be addressed before explaining how community managed projects and firms negotiated a working relationship. These factors shape what social movement theorists would call a social movement's political opportunity. Political opportunity is defined as "the degree to which groups are likely to be able to gain access to power and manipulate the political system" (Eisinger, 1973: 25). Jenkins and Perrow (1977) used this concept to explain why two farm movements with similar tactics, resources and leaders failed and the other succeeded. They argue that changes in the political environment were more important than factors internal to the movement. McAdam's study of the civil rights movement (1982) also draws upon this notion. McAdam makes the point that a political system can be more or less receptive to challenging groups at specific times. "The opportunities for a challenger to engage in successful collective action...vary greatly over time. And it is these variations that are held to be related to the ebb and flow of movement activity" (1982: 40-1).

Instead of a political opportunity structure defined by national politics and/or legislation, this political opportunity structure was defined by market dynamics. Variations in the market structure in the software industry fostered firm's

receptivity to free software and open source software. Industry dominants were frustrated by a monopoly hold on the software market. The presence of monopoly or market failure has been found to be conducive to the emergence of cooperatives (Sexton and Sexton, 1987). While it is in the interest of incumbents to deter the entrance of cooperatives, if cooperatives are partial as opposed to identical suppliers of the same good, incumbents are more likely to accommodate them. Firms may be more receptive to working with cooperatives if this introduces more choice to the market without providing goods similar to their own. Thus, the fact that software managed by communities did not typically target consumer markets coupled with the presence of monopoly failure might have provided an opportunity structure that encouraged non-monopoly holders to cooperate with a cooperative.

This political opportunity structure coupled with the technical success of the Linux operating system and the media attention its development model received helped create favorable conditions for a challenging group. A faction within the free software movement sensed this growing political opportunity, and seized it by transforming the free software frame and creating a spin-off movement, the open source movement, that was more hospitable to commercial interests. The frame transformation attracted further attention from the national media and inspired firms to rethink their attitude toward free software. These factors help explain the larger socio-economic context that made collaboration with a challenging group more attractive to defenders.

The problem with the notion of political opportunity structure is that, it can be used as a catch-all explanation to explain the success or failure of social movements: over attributing social movement outcomes to the socio-political

environment's receptivity. More recently social movement theorists have acknowledged that political opportunity structures are important, but caution that theorists should not be carried away with its seductive qualities (McAdam and Snow, 1997). In this study, the political opportunity of the market helped encourage firms to take an interest in community managed projects, but that does not predict the form of action that might follow. Firms, for example, could have used community managed software without necessarily collaborating with them.

Community managed projects held two bases of power that helped shift industry attention and interest in open source software to their consideration of community managed projects as a credible partner for collaboration. One source of power that community managed projects held was the size of their market share and their user and developer base. Three of the projects in this study held a significant share of their respective markets, one with over half of the webserver market. The standard setting project, which did not necessarily have a corresponding market share, had a different power base: relevance. Their standard would facilitate new entrants to the Linux market to establish compatibility with different Linux distributions.

Another source of power that community managed projects held was the protective mechanisms they used to preserve their software: legal and normative controls that encouraged users of their code to 'give back' to the project. The ability of project contributors to retain ownership and control over their intellectual property helped them to retain their market share and to establish them as a viable partner for collaboration. The use of copyright and trademarks, albeit for different purposes than they were intended, helped to establish a legal basis for the collective

management and ownership of software and provided a way for the community's software to survive in a commercial world.

Interviewer: How critical is the GPL to the moral and ethical foundation of..

Informant: Well the GPL is what makes free software survive in a copyrighted world. (*Open Source Initiative board member*)

Without the legal protection of their licensing terms, the community might not have been able to retain this power and firms might have been less willing to adapt their practices and policies to work on the community's terms, particularly with respect to intellectual property. Software licenses were an institution that firms used themselves, recognized and respected. Any disrespect of the community's terms might invite disrespect of their own licensing terms. These two power bases, market power and legal and normative control, helped offset technical communities' lack of economic and political power and established them as a viable commercial actor.

Veblen wrote that "one of the major problems facing American society is that its technological elites - i.e., its engineers - do not constitute an independent power base which shares in decisions governing the uses to which engineering talents are put" (cited in Perrucci, 1973: 121). He predicted the emergence of a "soviet of technicians", a revolutionary class that would try to recapture industry from business interests. The efforts of community managed projects may be one way to create a technical voice in society that can exert power independent of a firm. Van Maanen and Barley (1982) were quite prescient in their argument that the culture and control of occupational communities would become more critical to understanding work in a post-industrial society. However, the emergence of a technical community that

could establish and exert power enough to affect the practices of an established industry also depended upon material aspects of the technology itself. Without digital technologies constituting the context and product of this technical community, community members may have been less able to alter their power base and consequently alter dynamics between community managed projects and established firms.

Barley's analysis of how the micro-social dynamics occasioned by new technologies reverberate up levels of analysis (1990) includes an account of the ways in which a technology's material attributes affect work roles and relationships. Barley predicts that social patterns are likely to change when "materially induced changes in the non-relational aspects of roles spill over into the corresponding system of role changes" (1990: 99). While many popular and academic theorists have postulated that the Internet and World Wide Web will lead to the development of decentralized and new organizational forms much of these claims have been at the macro-social level, while much of the empirical research has been experimental at the individual or small work group level (O'Mahony and Barley, 1999). Few (Orlikowski, 1993; Orlikowski and Yates, 1994; Orlikowski et al, 1995; Orlikowski, 1996 are important exceptions) have heeded Barley's call for a more micro-social analysis (1990) of ways in which material changes in digital technologies reverberate up levels of analysis to contribute to new social structures.

The technical communities studied here were able to move programming work outside the firm largely because they could retain the rights to their source of expertise and because their work product could be cheaply digitally stored, accessed, and distributed in manner that would allow for collaboration and collective action.

The material attributes of the Internet enabled a powerless and challenging group to first coordinate and scale the production of software without capital resources. Contrary to popular beliefs that programmers are not socially inclined, these programmers did not just want to code alone, but work together. Armed with their talent and personal resources, individuals could, together create commercial grade code. Their evolving shared understanding of how their community of practice was working and developing code provided a context for collaboration (Brown and Dugoid, 2001; Wenger, 1999). These factors enabled communities to work together on projects that normally required the infrastructure of a firm and contributed to the renegotiation of what Barley would call relational roles (Nadel, 1957)⁶⁰ with firms. The digital nature of the product of this collaboration, its means of distribution, and the community's ability to retain rights to their work helped them build and maintain a base of power that occasioned the renegotiation of roles and social structures between technical communities and firms.

Theoretical Proposition #1: Technical communities that develop a large collective following and can retain the legal rights necessary to protect what they create can develop a base of power that makes change in social structures possible.

With a power base that was still unbalanced but more commiserate with firms, community managed projects moved beyond being merely interesting to firms, to becoming a potential partner with which to be reckoned. Firms who wanted to tap into the emerging Linux and open source markets were thus in the position of

⁶⁰ Relational roles cannot be played without an alter ego, a specific other who fills a complementary position in the social order. Non-relational roles are recurrent activities that fall within the purview of a particular position or job (Nadel, 1957 cited in Barley, 1990: 68-69).

learning to work with a new commercial actor. A favorable political opportunity structure, frame transformation, and legal foundation all contributed to help defenders in the software industry to reevaluate the challenging group, community managed projects.

However, social movement theorists typically use the word political opportunity to describe the conduciveness of the environment to a movement and its goals, not to collaborate with them. In order to use social movement concepts to explain industry dynamics in this environment, a shift from assuming that defenders or incumbents are in the position to grant something desired by challenging groups is required. Any perspective on the interactions between challenging and defending groups will be shaped by that assumption. When a challenging group has a respected, but different base of power, incumbent and challenging groups might be able to negotiate new grounds for interaction and the possibility for symbiotic or mutualistic relations to emerge may occur.

Theoretical Proposition #2: When a challenging group has a basis of power that differs from incumbent groups and interests incumbent groups, the two groups will be able to negotiate grounds for interaction.

B. The Negotiated Interaction Order. This research does not examine, at a very fine level, the strategies and tactics that helped community projects to recruit participants, coordinate their development, and grow their market share, all topics of equal interest. This research begins with mature projects that firms wanted to work with and examines what happened to the practices of communities and firms as a result. Firms and social movements are not new to each

other, as a long history of environmental activism could attest (Lounsbury, 2002; Bansal and Roth, 2000). They are however not used to working together in a highly interdependent development environment⁶¹. Firms that used community managed software, in effect allowed projects that grew out of a politically motivated social movement to become a part of their supply chain.

This interdependence also helped foster organizational innovation, for firms were not comfortable working with a form that they did not recognize and that embraced competing logics. Institution building is likely to occur where relationships are being forged in new, undefined social space (Fligstein and Mara-Drita, 1996). The changes and tactics pursued by both actors are presented in a manner that cannot do justice to their diachronic evolution over time. Without fine grained data on the timing and course of events, what emerges are the patterns and processes at a meso level. A meso level understanding of the interaction patterns between community projects and firms identifies the elements of their form and identity that each type of actor was willing to change and the elements they resisted changing.

Firms were initially cautious about how to approach community projects. Minimizing their size and power was a strategy to gain acceptance and trust within community projects. Allowing individual employees to represent themselves and retain rights to their own work required changes in employment contracts and in

⁶¹ I was not able to identify any examples that could parallel the interdependence between a social movement and a firm in a production environment that was apparent in this research in the literature. Habitat for Humanity, a service organization, might come close in that they are a non-profit organization producing goods with commercial resale value and they collaborate with donors. However, their mission is toward providing service, as opposed to a community of practice motivated by shared political beliefs. Furthermore, firms that they collaborate with do so in service missions, not as a part of their core business.

expectations about what employers and employees owe each other. In making these changes, firms also had to re-conceptualize the role of intellectual property as a source of competitive advantage. These changes allowed employees to act more like individuals, which facilitated community projects' acceptance of firms as legitimate partners that would not violate their normative order. In order to contribute code to community projects, firms adapted to the community's development context and style of working. This required adapting to public peer review of code decisions. Deference and accommodation to community norms and practices helped community members to accept firms as contributors to their projects. Community projects accepted the differing logics of firms and evaluated their intentions based on their ability and willingness to reciprocate with code and adapt to their practices.

Theoretical Proposition #3: Defenders of the established order will adapt to meet the terms of challenging groups if challenging groups have a differentiated power base that defenders value.

This political process of negotiating the terms of collaboration between community projects and firms helped shape the design of private non-profit foundations. Community projects adapted to the challenges of a commercial context by incorporating so that they could attract more resources, protect their members, and protect their work. Foundations could intercede between an online form that was not a legally recognized entity but more resonant with the movement's philosophy and the need for projects to represent themselves and protect their work in commercial markets. This leads to a slight modification of Gamson's (1975/1990) and Fligstein's prediction (1990) that one way to overcome the

asymmetry between established and challenging groups is for challenging groups to adopt the organizational form or pattern of established groups (1990).

Theoretical Proposition #4: When a challenging group enters a market, they will adopt some features that make them more like dominant forms.

However, adaptation to the dominant form will be limited on dimensions that conflict with a challenging group's political values. Community projects delimited roles for firms that did not include direct technical decision-making rights and adopting mechanisms to preserve pluralism, their technical autonomy, and the rights to their work. Similar competing tensions shaped the formation of the European Union, which intentionally has few enforcement powers in order to preserve national sovereignty (Fligstein and Mara-Drita, 1996). The vast majority of the directives given by the European Union are rules of exchange as opposed to directives, such as property rights, that are more tightly coupled with national sovereignty. Fligstein and Mara-Drita show that the Single Market Program's structure and function reflects compromises made by organized groups within Western Europe (1996). Similarly, this research shows that by creating an organization that had both adaptive and limiting qualities, community project members could adjudicate between the need to preserve their founding values and the need to adapt their structure to meet the needs of growing commercial markets.

Theoretical Proposition #5: If challenging groups have a power base that is distinct from the defending group, they may adopt the dominant form, but also adopt tactics to preserve their divergent interests.

Three years of adaptation and negotiation between technical communities and firms and the emergence of non-profit software foundations has helped define a new negotiated order between community managed projects and firms; one that would not be possible without digital technology and the ability to unbundle and rebundle associated legal rights. The ownership and maintenance of code is decoupled from its sale and distribution. The right to own the code is de-coupled from the right to earn a profit from it.

Foundations hold the assets and property rights of technical communities that produce software, but do not pay them. Nor do they collect revenues or distribute profits, or the work that is created. Firms are free to sell and distribute the community's work at a profit. They are also free to modify the work of the community as long as they respect the community licenses and contribute improvements back to the code base, where required. Firms cannot directly influence the direction of projects, but they can sponsor members who can make contributions on their behalf. Firms give up their right to own code on collaborations with community managed projects, but can still earn profits from its sale. Community members retain ownership of their work and the ability to set their own technical direction. In return, they are offered sponsorship and support. Digital technology and a new twist on copyright law occasion the creation of a new set of social structures that reflect compromises between competing institutional logics.

C. Comparison of Foundations. Competing processes of accommodation and preservation resulted in some interesting variances across the projects. Table VII-1 compares the attributes of the foundations across all four

projects. I grouped selected attributes of the foundation's role or governance structure that were closely linked to the processes of negotiation as either helping to preserve or accommodate the community's work. Three types of preserving attributes were identified: those that helped preserve the community's technical autonomy; preserve representation of the community; and helped support collective ownership. For example, limiting the role of the foundation from decisions in the technical domain was a feature of foundations that helped to preserve project member's technical autonomy. Creating a board that would be elected and that limited a majority form forming were attributes that helped preserve representation of the community.

Two types of accommodating features were identified: those that helped the project coordinate for commercial presence and allow firms an influential role on the project or the foundation. I consider these to be accommodating features as they were not a consistent part of the community development model or culture prior to collaboration with firms. Granting the foundation release coordination authority and the ability to set technical direction and employ staff were features that helped community projects to be more consistent and coordinated, a need that was not required when a community managed project was created by the community and for the community, but that materialized with the need to maintain a commercial presence.

All projects created foundations that have more preserving than accommodating features, which may not be surprising when considering that the new form of the foundation itself was an accommodation to the commercial environment that had grown around community projects. The project that

incorporated first and without the help of any firm, the Linux Distribution Project, has a very different foundation from the other three projects, with fewest of either type of attribute. As discussed earlier, this foundation plays a very small and almost negligible role for this project, acting primarily as a legal executor for the developers that continue to create a distribution that is targeted for technical audiences. They adopted only one accommodating feature and that is the creation of mechanisms to represent the project publicly. Preventing the potential infiltration of rationalistic controls, preserving their technical autonomy and sustaining pluralism were most important for informants in constructing their foundations. This project did not experience any direct threat from firms and did not experience pressure to improve their coordination for commercial presence. This project's earlier start and the fact that their development and distribution model was institutionalized prior to the rise in commercial interest in Linux may have helped insulate them from the influence of firms. A stronger antecedent of their minimalist form is perhaps this project's stronger affiliation with the free software frame.

The Webserver project which was founded and incorporated after the Linux Distribution Project, but earlier than either the GUI Desktop or Standard Setting projects also did not adopt very many accommodating features. Similar to the Linux Distribution, creating a legal entity to protect their intellectual property rights and represent the project were critical. Although a Fortune 500 firm helped catalyze the foundation's creation, I did not find evidence of direct pressure from firms in the design of their foundation. Their foundation does allow firms an influential role, by allowing board representatives to represent their employers, a change that came later to the organization. This project was also the only one that used a license that

allowed proprietary extensions of their work and had the largest market share.

Project members reported that there was more consensual agreement with the open source frame and less antagonism toward commercial interests on this project than on the other projects. This may have helped project members adapt to the role firms would play less problematically.

Interestingly enough, the two projects that had the most commercial involvement during their institutional building stage have more of both types of features. Both the GUI Desktop and Standard Setting Projects adopted a majority of the accommodating and preserving attributes discussed in this study. These two projects worked with more firms and in a more interdependent fashion than the other two projects. This may have affected the need to construct more complex mechanisms to balance competing interests and address community and firm interests. The GUI Desktop project adopted the most features that help projects coordinate for commercial presence and was also the only project that experienced direct pressure from firms in the design of the foundation's governance structure. Staggenborg (1988) found that when social movements formalize, they are better able to form and maintain coalitions, primarily because paid staff made coordination easier. This would help explain why both the GUI Desktop and Standard Setting Projects have paid staff, more coordination attributes, and more formalized firm roles.

The Standard Setting Project provides the most formalized roles for firms. As this project was trying to set a standard that would affect all Linux distributions, they worked with many diverse firms. The fact that balancing competing interests on this project were more explicitly raised as a direct part of the actual work of

creating a standard may have provided a context in which project members could be more explicit and intentional in designing an organization that could consider and balance competing interests. Their control group structure that limits the formation of a majority on the foundation's board across different categories of membership was also the most complex of the four projects.

These variations across projects suggest that the processes of accommodation and preservation affected each project a little differently. Those projects that were more accommodating to commercial interests have a foundation with a different structure and role than those projects that were less accommodating. Those projects that had a greater need to coordinate with firms granted firms more formalized positions and introduced more rationalistic controls into their foundation structure. Projects that were less accommodating to commercial interests have a foundation that most closely resembles the values of the founding movement. Minkoff (1999) has theorized that decisions that alter social movement's core identity may place them at greater risk of failure. While it would be premature to speculate about failure, there is some evidence that one project that was more accommodating was experiencing more difficulty in retaining one of the values they held dear, namely technical autonomy, but no evidence that it threatened the ideal of pluralism, primarily because of the preserving structures projects adopted.

Theoretical Proposition #6: The more challenging groups accommodate to resemble the form of defending groups, the more difficulty they will have in defending divergent interests.

D. Recombination. Variance in foundation role and structure suggest that different projects recombined existing organizational elements in distinctive

ways, but all projects borrowed traditional institutions and used them for new purposes. The non-profit corporate form was adopted for a new purpose: to allow the collective ownership and development of commercial grade software without profiting from it. DiMaggio and Anheir suggest that non-profits may be more likely to use donor dollars reliably (1990: 140); support professional autonomy (1990: 142); and that they may be better suited for certain goods that are perceived as requiring special protection from corruption (1990: 144). The fact that these four community managed projects managed to take software development out of the for-profit domain suggests, that regardless of their frame alignment, they no longer found software inappropriate for market exchange and wanted to protect it from the profit motive.

Other legal institutions, namely copyrights and trademarks, were adopted but inverted for the purposes of the movement. Copyright law is used to keep community software free from appropriation as opposed to restricting its distribution. The Copyright Act of 1790, the first US federal copyright act⁶², was conceived of as a statutory privilege given to authors for a limited time to prevent harmful monopoly, establish order in the book trade, and promote learning (Patterson, 1968). Copyright privileges have however, over the last two hundred years, come to be used to turn original works of authorship into marketable commodities (Rose, 1993; Lessig, 2002). The GNU General Public License (“copyleft”) borrows the technique of exerting rights over the works for an extended period of time, but exerts those rights to turn a commodity into a collective good.

⁶² This Act was replaced in 1976 with the New US Copyright Act that was amended in 1980 to include computer programs.

What is ironic is that the inversion of copyright was motivated by desires that were similar to early drafters of copyright law, to prevent harmful monopolies from forming.

Rao (1998) and Holm (1995) argue that the emergence of new forms do not arise from a vacuum, but must be constructed from prevalent cultural materials. Clemens refers to prevalent cultural materials as the "set of organizational models that are culturally or experientially available" as an "organizational repertoire" (1993: 758), similar to Swidler's (1986) conception of cultural toolkits for organizing. The existing repertoire of organizing mechanisms includes Internet mailing lists, incorporation, for profit firms, charitable non-profit organizations and cooperatives, project leaders, boards as well as legal mechanisms such as copyrights and trademarks. Members of community managed projects wanted to design a form that would be hardy enough to protect them in a commercial context but yet allow them to pursue their ideological goals and maintain their sense of identity as a community. Private non-profit software foundations that manage the production of what was once considered a commercial good reflect a recombination of organizing structures available from the existing repertoire but used for new purposes.

E. Comparison with Traditional Forms. These inversions in the use of traditional elements helped community managed projects engage in the political project of constructing a new type of form (Fligstein, 1996), one that would allow them to both adapt to the dominant form and protect their code and values in a commercial context. That is, to integrate a new way of organizing software development with the prevalent institutional order (Rao, 1998). Table VII-2 compares the resulting form, a private non-profit software foundation, with more

traditional forms. The foundations that emerged in this study are incorporated, were organized by and for individual members, produce benefits for the public and for a defined collective, own assets that are sold in commercial markets, may compete with firms, but do not redistribute profits to their members. The two forms that most closely resemble this form are those of a cooperative and a charitable organization. Table VII-2 shows where elements of both of these forms overlap with elements that have been incorporated into the non-profit software foundation.

Charitable organizations are non-profits that can be organized by individuals or firms and that produce benefits for the public or for a defined collective.

Cooperatives are non-profit organizations that produce benefits for a defined collective. Cooperatives are also different in that they can produce assets sold on commercial markets and may compete with firms. However they are organized by and for individual members and redistribute profits back to those members. The new form has some of the elements of a cooperative but does not redistribute profits back to its members. It also has some of the elements of a charitable organization, except that it owns assets sold on commercial markets that may compete with firms.

This research supports Rao (1998) and Havemen and Rao's (1997) findings that new organizational forms result from recombination of old and new institutional elements and that competing logics can affect the recombination process (Fligstein and Mara-Drita, 1996; White, 1992). It also supports Rao, Morrill and Zald's (2000) prediction that new organizational forms are produced by social movement processes when: normal incentives to create public goods are inadequate, market mechanisms fail to reduce social costs and when spin-off movements customize the master logic of the initiator movement. In this research, the creation of a spin-off

movement and new master frame and the motivation and beliefs of contributors who believed that proprietary software entailed social costs helped foment the creation of social structures that could support competing logics.

Most importantly it shows that process of accommodation and preservation were critical to the design of a new form of organizing. Several institutional researchers (Fligstein, 1996; Haveman & Rao, 97; Rao, 98; Davis and McAdam (2000); Swaminathan and Wade, (2000); Lounsbury, Hirsch & Ventresca, 1999) have theorized that the construction of new forms and institutions is a political process. This research opens up the black box, 'political process', to identify patterns that occurred between challenging and defending groups across all projects. I argue that recombinations of existing organizational elements will not be patternless, but will be marked a political process with competing tensions of accommodation and preservation.

Theoretical Proposition #7: Competing processes of mutual accommodation and preservation will affect the recombination of organizational elements in the creation of new organizational forms.

The role of a social movement in the construction of a new organizational form also validates Rao, Morrill and Zald's (2000) prediction that new organizational forms were likely to emerge "when spin-off movements customize the master logic driving an initiator movement to a new locale." (2000: 264). The frame transformation and spin-off movement helped create a logic by which collaboration with firms could be reconceptualized and planted the seeds for institutional innovation.

F. Efficacy of Foundations. How did the creation of foundations affect projects and their interactions with firms? Because this study was undertaken

as two of the foundations were emerging, it is difficult to determine with any precision how successful the foundations set up by projects were at representing community interests or at mediating firm and community interests. Informants did feel that the foundations played an important role in facilitating communication between communities and firms and helping to avoid potential problems and conflicts.

There have been a lot of potential problems that have been headed off. Just because on the list or our weekly calls, we were able to discuss something before it became a problem. And that is one of the things the community is not going to see and not going to realize but they sure do benefit from.
(*Sponsored Contributor, Standard Setting Project*)

This informant raises an interesting point, if the creation of the foundation and formal leadership roles probably did more to prevent problems from happening than it did anything else, its effects would be invisible to other project members. Since communities wanted to limit the role of their foundations, an undetectable effect could be evaluated as a sign of a successful design.

Project leaders recognized that changes associated with the creation of foundations could be distasteful to some of their members. One leader viewed a successful organizational design as one that “members could live with”.

I think to a large extent at least as far as I can tell we have created an organization that can live with the community and the community can live with it and work together towards maintain our software over a long period.
(*Volunteer Contributor, Webserver Project*)

An organization that its members could tolerate required minimal bureaucracy and rationalistic controls. Those most involved in the institutional building process worked hard to avoid this and were successful on three out of four projects. There

was no evidence to support Michel's prediction of the emergence of an elite few who work with incumbent groups to improve their legitimacy to the detriment of the wishes of ordinary members (1911/1949).

Rao (1998) postulates that new organizational forms become operative only when they embody a truce among contending social factions. Many informants spoke of the new relationship between community managed projects and firms on mutualistic terms. Explanations of mutualism often focused on the different types of competencies and resources that communities and firms could bring to bear on technical problems. For example, informants spoke of the different lens firms brought to development work and felt that volunteer contributors and firms were interested in different things.

There has not been that many clashes because commercial people don't tend to but in on stuff that the technical people are interested in. Because commercial people very seldom actually have strong interests about what the implementation details should be. While the technical people do have very strong opinions there and they usually articulate why and convince the commercial people why they are right in theory. But the technical people are not at all interested in the stuff that commercial products end up having to do. And there is not that much overlap. (*Founder, Linux Project*)

This project founder posits that one pinnacle of good working relations between community managed projects and firms was the fact that firms could make the contributions they needed to grow commercial markets without threatening community members' technical autonomy. The fact that he thought there was little overlap between the interests of community members and firms was viewed as helpful in maintaining peaceful and mutualistic relations. Firms that sponsored contributors on three other projects agreed with the same principle.

I think our main contribution is that we are using [Linux Distribution Project] and we are looking at [it] from a commercial point of view. And making improvements to make it more attractive to companies as an alternative to the commercial systems. So we are doing work that a nonprofit group is not necessarily interested in doing and looking at [the project] from a different point of view. So our hope is that by doing that, we are going to be able to help [the project] improve and expand its audience beyond where it is now. *(Former Leader, Linux Distribution Project, Open Source Firm Founder)*

To the degree that firms collaborating on these projects wanted the same goals as community members (to expand their audience or market share) and to the degree that each type of actor wanted to work on different aspects of the project, symbiotic relations were possible. Maintenance of this balance was understood to be tentative and to require social structures to maintain pluralism, representation of the community and protect community developed code from appropriation.

Leaders on two projects made the point that project contributions from firms were sometimes overlooked or undervalued and that project members needed to better appreciate the roles played by firms.

There is one last topic I want to talk about, which is finding a role for companies in your project...one interesting metric would be to find out how much of the work was done by people working as a hobby in their own spare time and how much of it was someone who was paid in some way or another to be responsible for the software itself...I would, if I throw out a guess that it is more than 50%. *(Websserver Project Founder, Former Volunteer, Presently Sponsored Contributor, August, 2000)*

People have the incorrect impression that open source software is primarily about people on weekends that are doing something out of the goodness of their heart, and while there certainly is some of that, and that's sort of how I started doing it, I mean I did that for a year, and I think a lot of people do, [but] that is certainly not the whole thing. You get substantial contributions from commercial entities. *(Sponsored Contributor, Project Leader, GUI Desktop Project, February 2, 2001)*

Sponsored contributors were on the front lines of helping to foster this symbiosis, because as this informant explains, many former community volunteers now worked for firms on the same projects.

Interviewer: How would you describe community corporate relations as they stand now?

Informant: I think that it has happened gradually enough that people have had time to get used to the idea. There certainly used to be a lot of friction, there are still people that think commercialism is bad. At least all the developers I talk to are fairly happy with commercial efforts and most of them tend to work for commercial companies now. (*Sponsored Contributor, Former Volunteer, Linux, March 12, 2001*)

G. Sponsored Contributors and Role Conflict Many informants felt that the introduction of more sponsored contributors helped their project's progress but they also feared that it could threaten their ability to maintain pluralism and the emergence of dominant controlling interests. As discussed previously, sponsored contributors strove to maintain their identity and status as individuals within the project environment, but many, particularly those who were formerly volunteers, felt a degree of strain from bridging two distinct worlds: the online project community and the firms in which they worked. One institutional entrepreneur who was primarily responsible for structuring the Standard Setting Project's foundation looked upon his role as a mediator, bridging and translating between two different worlds.

Now I have to set expectations accordingly and make sure that this company plays nice, that these Open Source developers don't feel negated, that they are still relevant, that they are being heard, that we haven't sold out. Also, at the same [I have to make sure that] these Open Source developers play nice with the corporations. (*Sponsored Contributor, Fortune 500 firm, Standard Setting Project*)

In his role as a leader of the new foundation, this informant recognized that he would have to help both firm and community members to adjust to a new logic of cooperation in order to continue to sustain symbiotic relations.

Another source of strain that informants (not necessarily those in leadership positions) mentioned was the need to cooperate with firm internal deadlines and requirements and the need to cooperate with community norms that regarded time constraints as an anathema to their development environment.

Informant: There was a deadline, that I had to have a supportable thing...but I couldn't announce it.

Interviewer: How do you have a deadline in a community context?

Informant: Well I couldn't announce it because it was an internal [Firm Theta] deadline, so how I had to walk this balancing act, of because I have write access to the code, it was really easy for me to tag the source code when it looked stable and when it passed all my tests and just didn't contribute anything for a couple of days. (laughs) (*Sponsored Contributor, Fortune 500 firm, Former Volunteer, Webservice Project, April 3, 2001*)

The degree of access this informant had enabled him to withhold contributing to the code base in order to improve its stability and more easily integrate it with the firm's pending deadline. This behavior, what the informant referred to as 'a balancing act', enabled him to create a temporary 'buffer zone' between community and firm activities that allowed him to synchronize two very different temporal orders. Attempts to introduce a structured temporal order to community development activities would be regarded as an imposition of commercial interests. Buffering eased the degree to which the firm's activities depended upon the community's progress and helped to ease individual role stress from working in two different

temporal orders. Sponsored contributors' adoption of buffering tactics enabled community managed projects to maintain their own organic temporal rhythms while also allowing firms to build a business model using their code.

Managing the boundary between a firm's technical strategy and project evolution was a second source of strain mentioned by sponsored contributors who were formerly volunteers.

Informant: But the reason I was doing it was for [the firm]. Which to me is the difference. I try to keep the trees separate. Because the stuff that I do for [for the firm], I might not, I have to think of myself as two different persons. Would I as the maintainer of the [project] accept this stuff that I as an employee am doing for [the firm's] project?

Interviewer: Interesting. So you sort of have to step outside your body?

Informant: Right yeah and say ok, I mean I trust myself. But in that sense it's easy for me to know whether I trust the technology. But at the same time, I have to ask myself, [does] this actually make sense outside of [the firm]? Because if it does not make sense outside of [the firm] it does not make sense to put it in the [project]. (*Sponsored Contributor, Former Volunteer, Linux, March 12, 2001*)

Sponsored contributors spoke stridently of their ability to withstand any influence from their employers over project direction, and yet freely admitted that their priorities as a sponsored contributor changed from when they were a volunteer. Few felt that their technical autonomy was sacrificed, but several provided evidence of strain associated with this new role. How the growing prevalence of sponsored contributors, a phenomenon that was emerging as this field work was undertaken, affects the continued evolution and governance of community managed projects and the work of sponsored contributors is worthy of further study.

Changes in the proportion of volunteer and sponsored contributors over time, could influence not just a project's ability to maintain pluralism but also provide an important source of future changes in the project organizing structures. For example, an increasing proportion of sponsored contributors could change the distribution of programmers that are co-located and distributed, which in turn, could affect the sharing of information that occurred on-line.

Problem is, people are very focused on a single project usually within the boundaries of their companies. It also doesn't help the 'hacking together' mentality which is one of the forces of the project. (*GUI Desktop Project, Board Meeting, April 8, 2001*)

As this informant indicates, as more and more sponsored contributors joined one firm, the proportion of their offline work interactions might increase, thereby reducing the proportion of interactions recorded online. This could change information sharing and learning dynamics, as interactions occurring within a firm are not accessible to distributed members of the group. Prior to corporate sponsorship, volunteers were equally distant and distributed as there was no physical space to support congregation. Thus, greater concentration of sponsored contributors in a few firms could lead to disparities in information sharing and contribute to project conflict.

H. Founding Movement: Frame Elasticity and Dilution. Finally, how have changes in form, organizing logic and frame affected the founding free software movement? As mentioned in Chapter II and demonstrated in Table II-2, the open source frame differed from the free software frame by providing a conception of non-proprietary software that was not inconsistent with market logics and was thus more inclusive of commercial interests. This conception emphasized pragmatic

benefits associated with open code over the moral and political philosophy that guided the forming of the free software movement.

The recycling movement went through a similar transformation in frame and logic: shifting from a vision of recycling as a means to restructure capitalistic production and consumption to a technocratic logic that could promote a profitable recycling industry (Lounsbury, 2002). Lounsbury shows how this change in logics helped recycling grow from a marginal practice promoted by activists to a major for-profit industry (2002). This research has argued that a similar change in logics: from a frame that was hostile to commercial interests to one that was more market friendly played a necessary, but not sufficient, role in attracting commercial attention and helping to create a new set of social structures that could support community and firm collaboration.

As evidenced in Figure II-2, the free software frame has not received the same increase in media attention, as did the open source frame. Media attention of free software has been fairly stable since its inception, while media attention of open source grew at a rapid rate. However, from the perspective of the founding movement, the new frame has much less narrative fidelity to the original goals of the movement and is thus more open to the risk of cooptation or goal transformation (the deliberate redefinition of organizational goals) (Jenkins, 1977).

Informant: The open source movement is where the co-opt sign is. You have heard of a kick me [sign]?

Interviewer: What do you mean by that?

Informant: It is sort of designed to be co-opted.

Interviewer: Designed to be co-opted. Ok. What are the cracks that you see that other people are not seeing?

Informant: Well I don't think they are not seeing it, I just don't think, if they don't share the idealism, than the co-optation won't bother them, it just seems natural. (*Richard Stallman, Free Software Movement Founder*)

Stallman, the founder of the free software movement, equates the development of the open source frame and the emergence of a spin-off movement with cooptation. His analysis suggests that the emergence of a movement that is more hospitable to commercial activity is implicitly seeking co-optation and thus it is to be expected. To co-opt something is to take it over or appropriate it, but the term is also often used to refer to goal displacement or transformation. Appropriation in this sense would not refer to the appropriation of code, but of the term, meaning, and logic imbued in the free software movement and its ability to attract supporters.

The free software movement founder fears that the new frame has diluted, if not transformed, the goals that Stallman laid out for the free software movement.

Informant: We developed this operating system so we could have freedom. These people [open source movement] paid no attention to the idea of freedom, but they liked the operating system so when they saw developments of non-free software that would increase the popularity of the system, they celebrated it, even those developments did not do anything for freedom, and in fact weaned, tended in the direction against freedom. I saw that they had been completely co-opted. That their focus on popularity had led them to approve of things that were bad for freedom. One step is to forget about the goal and fix on the sub-goal. You know confusion with means and ends. The second step is, when in pursuit of that sub goal, you start going against the original goal. And that had actually happened.

Interviewer: Oh, so the definition of co-optation you are applying would be going against [the goal]?

Informant: Being used for something that is bad which is against the deeper goal that should be there. I think that is the definition of cooptation. Is there a better definition of cooptation? (*Richard Stallman, Founder, Free Software Movement*)

The concerns articulated by the founder of the free software movement are supported by a large literature that has found that groups advocating social change become less confrontational with established elites in order to improve their prospects for survival (e.g. Michels, 1911/1962; Piven and Cloward, 1977). In order to garner support from elites, challenging groups may recast a protest strategy to one of advocacy and thus attenuate, if not transform, their goals for social change (Michels, 1911/1962; Piven and Cloward, 1977; Jenkins, 1977). Minkoff's study of women's and radical minority organizations suggests that a shift from protest toward a more conventional strategy could confer greater legitimacy, but that it could also limit a movement's ability to mobilize in the future (1999). She argues that changes to a movement's core beliefs that signal greater conformity to established institutions and less confrontational models of action will not provide significant survival advantages to movement organizations in the long term (1999: 1697).

This literature suggests that challenging groups' courting of incumbents has a cost: one that is also evident in this study. Jenkin's study of the National Council of Churches postulates that if the tendency toward goal transformation and cooptation by elites is true, then accommodation to the larger social order would become the overriding rule (or outcome) for groups advocating social change (Jenkins, 1977: 570). Instead, he finds that the development of professional social movement staff helped reinvigorate the movement's goals in a more radical than conservative direction (1977). Unlike Jenkins' study, this research did not find evidence of greater radicalism with the creation of a new frame and professionalization of the movement, but it did find evidence of accommodation, coupled with preservation, for both challenging and incumbent actors. The open source frame, which began as

a marketing program and has evolved to become a social movement that includes firms and commercial activity has strayed from the original goals of the founding movement and this has led to a new negotiated order with firms. It is difficult to label this as cooptation. It looks more like change on the part of both types of actors with power bases structured on different sources.

I argue that these competing forces provide the creative abrasion (Hirschberg, 1998) needed to recombine existing elements to create new organizing mechanisms that are different from the established order and yet can form a foothold of acceptance. Lounsbury has encouraged (2000) scholars to move away from the ideologically rooted and overly simplistic assumption that the professionalization of social movements leads to co-optation and to think instead of how social movements change over time. In this research, social change activists and industry dominants affected each other's logic and form of organizing. Points of resistance and adaptation fostered new institutional logics that defy either classification at this point in time.

It is clear that the free software frame has been transformed and that in doing so, new goals have been adopted that value commercial acceptance in addition to non-proprietary code. This has in turn affected industry dynamics.

Informant: Why I say the revolution is over? It is because the alternative, which is proprietary software, is just not viable and most people know it. So, in the long term, open source, well it is going to be evolutionary, because it will change a whole lot, in ways you can't imagine. I cannot imagine open source dying. *I can imagine that it will evolve in ways that five years from now, Richard Stallman will stand up and say "this isn't open source". (italics added)*

Interviewer: Like he is doing right now?

Informant: Well, right that is true, safe prediction because he is already doing that. But, I meant that everyone would nod and say 'you are right, it is not

[open source], it has changed'. (*Senior Research Scientist, Fortune 500 firm, July 18, 2000*)

To the degree that the open source frame becomes increasingly elastic in its use and application by those new to the movement, it may have a more difficult time in maintaining fidelity to the founding movement's goals. This, in turn, may affect the degree of cooperation between the two movements. Prior research on social movements has shown that factionalism within movement has an adverse affect as it diverts resources from use in struggle against opponents outside the group (Frey et al, 1992). The free software movement has little danger of cooptation: its frame has stayed very consistent since its inception. However, it has had a more difficult time in retaining supporters in light of the attention and use that the open source frame has been able to purchase. This is consistent with Frey et al's finding that it is hard for groups with displacement goals to succeed in achieving new advantages (Frey, et al, 1992).

In order to gain acceptance from established elites in the software industry, the meaning and goals of the free software movement were purposely diluted. While this could be considered cooptation of the free software frame, it nonetheless, has provided a source of innovation and helped inspire the creation of new logics for organizing between community managed projects and firms. To the degree that community managed projects pursue tactics to preserve their culture, work practices and code, and can maintain their base of power, they may be able to sustain the symbiotic balance of this order. To the degree that the open source term, meaning and logic becomes so elastic that it grows to include principles embracing proprietary software, than this symbiosis may be on shakier ground.

To understand the outcomes that are possible when a challenging group strives to gain legitimacy, attention, and respect from established institutions and carve a foothold for divergent social change, social movement outcomes must be re-conceptualized beyond co-optation or the realization of narrowly defined goals so that the complexity that occurs when activists engage with entrenched institutional actors is captured. Social movement scholars have, too often, “turned off their cameras when protesters leave the street”⁶³ instead of turning them toward the equally interesting dynamics that may be happening at the negotiating table inside.

⁶³ Dick Scott mentioned that one of the things Doug McAdam said that he would have liked to have done differently in his landmark studies of the civil rights movement was “not to turn off my camera when people left the street” at the Social Movements and Organization Theory Conference at the University of Michigan, May 10-11, 2002.

Chapter VIII. Implications

People have this idea, in the open source world, that things are self-organizing. It's this great myth -- the mythology of the Internet. The Internet just kind of happens - thousands of people around the world. Well, actually no, the Department of Defense blew a couple billion dollars, right? And all these companies executed these core technologies and these other folks, you have the W3C and all this other shit and you have structure here actually helping make it happen. So that's the mythology, the same mythology in the Linux organization, same in [GUI Desktop Project]. It's like the mythology is, we are hackers, we do this stuff and we don't need marketing weenies and we don't need an organizational structure or an organizational chart, because we are meritocracy and we hash things out by ourselves. Well, actually that's not what happens... (*Sponsored Contributor, GUI Desktop Project, February 8, 2001*)

What happens is that community managed projects build social structures in ways that both replicate established institutions and alter them on dimensions that are inconsistent with their goals and identity. This informant explains that a community project's survival depends upon organizing structures and support from sponsors, a fact oft overlooked by the public, but recognized by social movement theorists who have long studied challenging groups. This research examined the interactions between community managed projects and firms and the organizing mechanisms designed to facilitate their cooperation. It shows the micro-interactions that helped build an institutional space that could support the competing logics of firms and community managed software projects. The internal organizing mechanisms projects use to manage their boundaries, membership and decision-making processes have not been a focus, but are of equal interest and the subject for future research.

The context of community software development has not been a focus of this paper, but is worth returning to when discussing implications of this research.

Most of the interactions described here occurred in an online context. Many people who informed this research and worked on the same projects had never met each other. Researchers of online communities⁶⁴ have emphasized their virtuality (Rheingold, 2000), and their ability to create alternative identities or realities (Turkle, 1997). Online communities that have productive aims as well as social orientations offer a different lens into the junction of online and offline worlds. Examining the interactions between free forming online communities and established institutions may prove to be fruitful grounds for future study as it is this nexus that may best inform the emergence of new forms of organizing.

As the informant on the prior page indicates, the public perception of these projects as ephemeral and lacking in any kind of formal structure is very far from the truth. The four projects studied here produce commercial grade software, cooperate with firms, own trademarks and copyrights, are incorporated, are represented at trade shows and have sophisticated governance structures to guide their technical and organizational development. These projects make use of many of the same techniques firms use to develop software, attract users and to promulgate and protect their work. They however, do so with a different goal in mind.

A. Contributions. This research builds on existing work to further confirm that social movements can provide a source of diversity that fosters organizational innovation (Rao, Morrill, Zald, 2000; Rao, 1998; Clemens, 1993) and that the process of institution building itself resembles social movement processes (Fligstein, 1996; Fligstein and Mara-Drita, 1996). Similar to the findings of Rao (1998) and Haveman

⁶⁴ The term online community here is used to refer to pre-institutional, unincorporated emergent collectives as opposed to corporate sponsored and designed online communities.

and Rao (1997), hybrid forms emerge when competing institutional logics provide the variation necessary to enable the recombination necessary to construct new forms.

Understanding where new forms of organizing derive from is important because new types of forms contribute to diversity in the population. Diversity expands the menu of models or repertoires available to policy makers and institutional entrepreneurs to solve social problems (Rao, 1998; DiMaggio and Anheier, 1990). DiMaggio and Anheier argue that non-profits in particular “contribute to pluralism by creating centers of influence outside the state and provide vehicles through which disenfranchised groups may organize” (1990: 151). Understanding one pathway that explains the gestation of new forms contributes to a critical deficit in institutional theory. This account focuses on groups with very divergent power bases and resources and explains how agency, interests, power, and rights contribute to the development of new social structures. In doing so, it offers insight as to the structuration processes that underlie the creation of new social structures that may eventually come to be taken for granted and replicated.

This research also contributes to social movement theory by offering a more nuanced understanding of social movement outcomes that are possible when movements “leave the street”. Examination of the consequences of social movement activity is critically under developed (Giugni, 1998; Cress and Snow, 2000) partly because social movement theorists usually define outcomes narrowly (Giugni, 1998) and partly because of the unit of analysis favored by movement theorists (Cress and Snow, 2000). Examination of macro level social movement activity and their effect on the passage of national policies as opposed to the micro-

interactional level of social movements at the local level may have led theorists to miss identifying other types of possible outcomes (Cress and Snow, 2000). In Cress and Snow's study of homeless social movements, they found several different types of outcomes (representation, resources, rights, and relief) and multiple pathways by which they were achieved (2000).

Furthermore, Cress and Snow argue that because outcome attainment is too often analyzed in a correlate fashion, the ways in which different types of outcomes may interact and combine with one another are unexplored (2000). While both Voss (1996) and Cress and Snow (2000) have called attention to the need to study social movement activity at the local and organizational level, few have done so. Rao, Morrill and Zald have also suggested that a more micro level of analysis attending to the emergence and impact of social movements as they are experienced on the front lines of organizations (2000: 276) would be an area fruitful for further study. Their recommendation of a strategy of comparative case studies on the impact of social movements and new forms is one that has been pursued in this research.

B. Sustainability. This research captures an emerging phenomena at a particular point in time. Thus, it is important to postulate factors that may affect the sustainability of these new social structures and the negotiated order that emerged between community managed projects and firms. Cooperatives and strong collectivist organizations do not typically enjoy a high rate of organizational survival (Kanter, 1968; Rothschild-Witt, 1979; Miller, 1981; Simons and Ingram, 1997). Most often this is due to internal reasons rather than antagonism from the capitalist class (Miller, 1981). In Simons and Ingram's study of Israeli kibbutz (1997), of 1,448 cooperatives founded between 1920 and 1992, 1,255 (87%) of them failed. In

Miller's analysis of the demise of British cooperates in the nineteenth century (1981), conflict over member share arrangements that separated capital and labor contributed to their demise. However, Miller's assumption that profits from cooperatives would be distributed to members and that all members would try to maximize their personal incomes and (1981) is less relevant for this context.

Without the expectation of redistributing profits and the shared goal of producing open source software, community managed projects and their foundations may avoid some of the pitfalls that cooperatives have historically experienced. A second factor that may contribute to the sustainability of community projects and their foundations is the narrowness of the domain of social life to which they apply. In prior studies of collectivist organizations (Kanter, 1968; Rothschild-Witt, 1979; Swidler, 1979), these organizations were organized around domains that were all encompassing. Community managed software projects place no more demands on their members other than to produce high quality code in accordance with the norms and practices that each project has developed. Contributors are free to live, work, and engage in all other social activities according to their individual wishes. The narrow domain and aspatial character of community managed projects may allow for more individual differences to coexist with the shared goals of a collectivist group⁶⁵. This may help diffuse or prevent the type of internal conflict that has historically contributed toward the demise of collectivist organizations.

If we reconceptualize community managed projects as a collective managing a common pool resource, the work of Ostrom (1990; 1999) and colleagues (Ostrom

⁶⁵ Shared goals within a community managed project is not meant to imply agreement on means and methods, but agreement on working toward the success of the project.

et al, 1999) would lend hope to the sustainability of community managed software projects. Ostrom and her colleagues (Ostrom, 1990; Ostrom et al, 1994; Ostrom et al, 1999; Ostrom, 1999) have theoretically and empirically challenged Hardin's assumptions inherent in his famous treatise, '*The Tragedy of the Commons*, (1968) as well as his dismal conclusions. Hardin theorized, that if all individuals maximized their gains when drawing from nonrenewable resources, that these resources would, over time, diminish. Because the individual cost of using the resource would always be less than the collective cost, individuals would have no incentive to show temperance. People following their own short-term interests would produce outcomes that were not in anyone's long-term interests (Ostrom et al, 1999). The outcome Hardin feared was the ultimate destruction of natural resources and thus 'the tragedy of the commons'.

Ostrom and colleagues argue that this metaphor not only mischaracterizes the problem of common pool resources, but that it leads to incorrect conclusions. The tragedy of the commons only becomes a tragedy if the actors using the commons are "norm-free maximizers of immediate gains, who will not cooperate to overcome the common dilemmas they face" (Ostrom, 1999: 493). "Viewing resource users as trapped in a tragedy of their own making is consistent with contemporary textbooks on resource economics and the predictions derived from non-cooperative game theory for finitely repeated dilemmas" (Ostrom et al, 1994). The common resource pool perspective advocated by Ostrom and her colleagues is one that recognizes human actors as capable of cooperating with each other and of establishing norms and social mechanisms to encourage and reinforce cooperative behavior.

Recent reviews of field and experimental studies (Ostrom, 1999; Ostrom et al, 1999) indicate that groups can learn to problem solve and develop solutions to manage common goods in sustainable ways. For example, Sneath's study (1998) shows how local property collectives in Mongolia managed pastoral land with much less degradation than state-owned or centralized solutions (Ostrom et al, 1999: 278). In one study of inshore fisheries, 33 out of 44 user groups used different combinations of boundary rules to regulate the use of available resources (Schlager, 1994: 258). For example, 30 out of 33 groups limited fishing to local users with residential status. Groups with boundary rules managed their resources more effectively than groups without such rules (Schlager, 1994). These studies demonstrate that localized users of common natural resources can develop sustainable institutions to manage these resources and that these mechanisms can outperform more centralized solutions. Thus, and thankfully so, the tragedy of the commons may be overstated.

A few general principles can be gleaned from studies of common pool natural resources and compared with tactics used by open source projects to protect their work. Mechanisms to manage common pool resources usually either restrict access to the resource or create incentives to use the resource with more temperance (Ostrom, 1999). Groups that are able to identify each other are more likely than groups of strangers to draw on trust, reciprocity, and reputation to develop norms that limit use (Ostrom et al, 1999: 279; Ostrom, 1999). Groups that are better able to monitor and coordinate activities in order to solve common pool resource problems are more likely to develop mechanisms that can help them sustainably manage their resources. Users that highly value the resource and envision a long

term future of using or working with the resource are more likely to help define, use, and enforce mechanisms to control the use of the resource.

These principles also point to some fundamental differences between common pool resources and community managed software. Unlike common pool resources, community managed software is publicly available. Resources do not need to be redistributed or limited on any scheduled basis. However, community managed software projects do meet several of the other conditions that have been found to facilitate the sustainability of common pool resources. Namely, community managed software is developed in a context where everyone is identifiable and there are high monitoring and coordinating capabilities. Contributors highly value the resource and view their participation on projects as a long term project which may foster the sustainability of mechanisms to manage the resource.

The most severe challenges that could threaten the sustainability of the projects in this study are more likely to stem from external as opposed to internal threats. Firms are likely to become more involved in working with community managed projects and sponsoring them in a more official and explicit way. Industry analysts expect a compound rate of growth (CAGR) of 29.3% for Linux operating system and subsystem shipments through 2006 (Gilen, 2002). This would increase revenues from 69.5 million to 291.4 million (Gilen, 2002).

I think that to continue to evolve and service the demands that are made on it [the project], it needs corporate involvement....I believe that open source needs commercial involvement and it's the commercial involvement and a healthy, successful commercial company involved in a healthy, successful open source project that really feed off each other (*Open Source firm Founder, Former Volunteer, Webserver Project, March 13, 2001*)

Because commercial interest and involvement in community managed projects is likely to grow, the sustainability of this form of organizing and the negotiated order in general will rest largely upon the ability to balance competing interests. The design and scope of projects and their foundations will continue to evolve but as White (1992) and Rao (1998) predict, this will be the outcome of “contending attempts at control and competing quests to impose a preferred definition of the identity of the constituencies that benefit from the form” (Rao, 1998: 920). The ability of these emerging governance models to withstand dominance and promote pluralism among firms and independent community members will be critical. This is consistent with Ostrom’s (1999) argument that polycentric systems, systems where individuals are able to organize not just one but multiple governing authorities at different scales, can be effective in helping collectives to manage common pool resources.

C. Limitations. The projects discussed here are mature, successful community managed projects. Less successful projects may not have the same experiences of the projects discussed in this study. The population of informants that participated in this study is weighted toward those in leadership roles on the projects studied, and the experiences of those on the periphery of these projects may differ. Community managed projects that do not produce commercial grade software or develop a base of power through their claim of market share or other means are unlikely to command the interest of firms and are equally unlikely to develop mechanisms that require coordination for commercial presence. These projects were developing software that were viewed as more infrastructural than consumer oriented. The dynamics between community managed projects and firms

might vary depending on the degree to which community managed projects moved toward consumer applications. This may be an important distinction in evaluating the reach of these findings to other projects.

The firm tactics and changes discussed in this study are those used by firms that approached the four community projects in this study and do not limit the number or types of changes firms may have made in relation to their work on community projects. This research presumes the interest of firms and begins at a stage where firms were interested enough in community managed software to entertain the idea of working with them. It does not address reactions from firms that did not approach or want to establish relations with community managed projects. The degree to which changes in firm practices that informants shared were disseminated within firms is difficult to establish. Some of the firms that approached these projects also maintained businesses that depended upon proprietary control of software. The degree to which the practices and changes discussed here diffused to other business areas is unknown.

However, it would be safe to surmise that these practices had an effect on firms' conceptions of software and their ownership rights. Corporate attorneys, responsible for other types of intellectual property decisions within firms, were involved in helping to craft some of these changes firms made to adapt to community managed projects. Corporate informants who went through internal procedures to help these changes come to pass perceived them to be significant.

D. Generalizability. The challenging groups in this study had several criteria that make them unique from most of the challenging groups that social movement theorists have studied. One, digital technology supported collaboration,

coordination and distribution. Two, the rights of ownership were contestable and, because the intellectual property in question was digital, these rights could be unbundled and rebundled. The ability to de-couple and recombine ownership rights enabled the separation of software development from the management of distribution. This, in turn, altered the capital needs of community managed software projects and enabled them to survive with minimal support while enhancing their reach to new supporters. Three, the prior two factors enabled this challenging group to reach a growing audience of supporters, users, and developers which helped them to sustain and expand a base of power that was independent of and not controllable by incumbent firms.

These conditions may limit the relevance of these findings to contexts outside of software development. However, I would not be surprised to find similar phenomenon occurring in an industry like the music industry. While artists and recording studios have very different employment relations and contracting mechanisms than do technical workers, the music industry shares some similar qualities with the software industry: a digital product; contestable property rights; a growing political process, and the ability to draw upon a power base (fans) that is independent from and not controllable by incumbents (established firms in the music industry).

E. Alternative Explanations. What type of alternative explanations might explain these research findings? One plausible alternative is that community managed software projects have ‘sold-out’ and that this is a story of co-optation. In Chasin’s study of the lesbian and gay movement (2000), she examines the conflict between a challenging group’s quest for acceptance and their desire for progressive

social change. Chasin points to the ways in which a challenging group that is gaining the attention and interest of incumbent groups may be compromising itself and the type of strings that may be attached to gaining resources and legitimacy. Simons and Ingram's study of (1997) Israeli kibbutz demonstrates that a challenging group's ideological principles may be moderated by external political and economic pressures such as the need to acquire resources. In their study of Israeli kibbutzim, socialist organizations in a capitalist system faced economic incentives to change their form.

The free software and open source movements face similar dilemmas.

However, the preservation tactics that are pursued by both movements leads me to effectively rule out this alternative explanation. With regard to development practices and the management of the code, community based informants thought that firms had made greater concessions in changing their business practices to work within their context than the changes they made to adapt to a commercial context. With regard to the meaning and interpretation of the terms of free software and the founding goals of the movement, informants feared that goal distortion, if not displacement, had occurred.

Since community managed projects required little capital to sustain themselves, community managed projects did not face the same pressures as prior studies of challenging groups to obtain resources from elites at the expense of their ideological commitments. Their power base and legal and normative controls helped them to adapt to a new audience, without relinquishing dimensions critical to their identity and mission. This analysis does not mean that cooptation will not happen, just that there is not enough evidence to currently support that argument. In her study of women's groups, Clemens (1993) claims that a challenging group's

adaptation of organizational models that are both familiar and novel may “bring about changes in the taken-for-granted rules about what political organization is and what it is for”, if not ultimate outcomes (1993: 792).

A second alternative explanation is that the foundations that community managed projects developed are not critical to the negotiated order that has emerged between communities and firms or to the success of community managed projects. Across the four projects there was variance in the size, power and role played by the foundations that projects created: varying from a minimal role for the Linux Distribution Project to helping with release coordination on the GUI Desktop Project and mediating between different firms on standard setting in the Standard Setting Project. I would argue however, that even if foundations do not play a very large role for projects like the Linux Distribution Project, their very existence is a nod to the need to reconcile two competing orders: the logic of online mailing lists and a form that the offline commercial world could legally recognize. In this study, I interpreted the community’s efforts to reduce the scope and power of their foundations as a tactic to preserve their culture, identity and mission.

A third alternative explanation is that scale and growth played a larger role in the development of formal structures to represent projects to the public than any need to address competing logics or interests. While all projects grew over time, and faced strains from their growing membership, this was not reported by informants to be a driving consideration in incorporating or adjusting their logics for organizing in relation to their stance with the commercial environment. All reported the existence of some type and degree of commercial pressures.

This research suggests that the interstices between technical communities and firms constitute loci for innovation and the articulation and dissemination of technical standards that are worthy of further study. Further research is needed to understand the internal governing mechanisms central to the sustenance of community managed projects. Lessig has argued that code can constitute a means of governance and he asks “if the architecture is a kind of law – then as in any law-making contest, we should be asking, who are these law makers, and how do they make the law” (1999: 105). If community managed projects are central to the development and maintenance of the Internet and World Wide Web (Berners-Lee et al, 2000), than understanding their governance and their relations with firms will be central to understanding the evolution of the infrastructure on which a post-industrial social and economic order rests.

References

- Abbott, Andrew. 1991. "The Future of Professions: Occupation and Expertise in the Age of the Organization." In S.R. Barley and P. Tolbert (Eds.), *Research in the Sociology of Organizations* 8: 17-42.
- Abbott, Andrew. 1992. "An Old Institutionalism Reads the New Institutionalism." *Contemporary Sociology* 21: 754-756.
- Adler, Patricia, Peter Adler, and Andrea Fontana. 1987. "Everyday Life Sociology." *Annual Review of Sociology* 13: 217-235.
- Aldrich, Howard E and Fiol, C Marlene. 1994. "Fools Rush In? The Institutional Context of Industry Creation." *Academy of Management Review* 19: 645-670.
- Arensberg, Conrad M. 1954. "The Community-Study Method." *American Journal of Sociology* 60 (2): 9-1.
- Arensberg, Conrad M. and Solon T. Kimball. 1968. "Community Study: Retrospect and Prospect." *American Journal of Sociology* 73 (6): 691-705.
- Bansal, Pratima and Kendall Roth. 2000. "Why Companies Go Green: A Model of Ecological Responsiveness." *Academy of Management Journal* 43 (4): 717-736.
- Barley, Stephen R. 1990. "The Alignment of Technology and Structure through Roles and Networks." *Administrative Science Quarterly* 35: 61-103.
- Barley, Stephen R. and Pamela S. Tolbert. 1997. "Institutionalization and Structuration: Studying the Links between Action and Institution." *Organization Studies* 18: 93-117.
- Baum, Joel A. C. and Christine Oliver, "Institutional Linkages and Organizational Mortality." *Administrative Science Quarterly* 36: 187-118.
- Behlendorf, Brian. 1999. "Open Source as a Business Strategy." In *Open Sources*. Edited by Chris DiBona, Sam Ockman, and Mark Stone. Sebastopol, CA: O'Reilly.
- Benford, R. D. and D. A. Snow. 2000. "Framing Processes and Social Movements: An Overview and Assessment." *Annual Review of Sociology* 26: 611-39.
- Benford, Robert D. 1993. "Frame Disputes within the Nuclear Disarmament Movement," *Social Forces* 71: 677-701.
- Berners-Lee, Timothy, Mark Fishetti, and Michael L. Dertouzos. 2000. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. New York: HarperBusiness.

- Bessen, James. 2001. Open Source Software: Free Provision of a Complex Public Good. Unpublished working paper available at www.opensource.mit.edu.
- Blau, Judith R. 1996. "Organizations as overlapping jurisdictions: Restoring reason in organizational accounts." *Administrative Science Quarterly* 41:172-179.
- Brint, Steven and Jerome Karabel. 1991. "Institutional Origins and Transformations: The Case of American Community Colleges". In *The New Institutionalism in Organizational Analysis*. Edited by Walter W. and Paul J. DiMaggio Powell. Chicago and London: University of Chicago Press.
- Brooks, Frederick P. Jr. 1975. *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley.
- Broom, Leonard and Phillip Selznick. 1955. *Sociology: A Text with Adapted Readings*. New York: Row, Peterson.
- Brown, John Seeley and Paul Duguid. 2000. *The Social Life of Information*. Boston, MA: Harvard Business School Press.
- Burns, Tom. 1961. "Micropolitics: Mechanisms of Institutional Change," *Administrative Science Quarterly* 6: 257-281.
- Calhoun, Craig. 1995. "New Social Movements of the Early Nineteenth Century". In *Repertoires & Cycles of Collective Action*. Edited by Mark Traugott. Durham, NC: Duke University Press.
- Chasin, Alexandra. 2000. *Selling Out: The Gay and Lesbian Movement Goes to Market*. St. Martins Press.
- Clemens, Elisabeth S. 1993. "Organizational Repertoires and Institutional Change: Women's Groups and the Transformation of U.S. Politics, 1890-1920." *American Journal of Sociology* 98: 755-798.
- Clemens, Elisabeth S. and James M. Cook. 1999. "Politics and Institutionalism: Explaining Durability and change." *Annual Review of Sociology* 25: 411-466.
- Cloward, Richard A. and Frances Fox Piven. "Disruption and Organization: A Rejoinder," *Theory and Society* 13: 587-99.
- Coleman, James S. 1986. "Social Theory, Social Research, and a Theory of Action," *American Journal of Sociology* 91: 1309-35.
- Constant, David, Lee Sproull, and Sara Kiesler. 1997. The Kindness of Strangers: On the Usefulness of Electronic Weak Ties for Technical Advice. In *Culture of the Internet*. Edited by Sara Kiesler. Mahwah, N.J: Lawrence Erlbaum Associates, Publishers.
- Cress, Daniel M. and David A. Snow. 2000. "The Outcomes of Homeless

- Mobilization: The Influence of Organization, Disruption, Political Mediation, and Framing." *American Journal of Sociology* 105: 1063-1104.
- Dacin, M. Tina, Jerry Goodstein, and W. Richard Scott. 2002. "Institutional Theory and Institutional Change: Introduction to the Special Research Forum." *Academy of Management Journal* 45: 45-56.
- Daft, Richard L. and A. Y. Lewin. 1993. "Where are the Theories for the "New" Organizational Forms? An Editorial Essay." *Organization Science* 4: i-vi.
- Davis, Gerald F and Thompson, Tracy A. 1994. "A Social Movement Perspective on Corporate Control." *Administrative Science Quarterly* 39:141-173.
- Davis, Gerald R. and Doug McAdam. 2000. "Corporations, Classes, and Social Movements After Managerialism." *Research in Organizational Behavior* 22: 193-236.
- Dibona, Chris, Sam Ockman, and Mark Stone. 1999. *Open Sources*. Sebastopol, CA: O'Reilly.
- DiMaggio, Paul. 1991. "Constructing an Organizational Field as a Professional Project: U.S. Art Museums, 1920-1940." In Powell, Walter W and DiMaggio, Paul. *The New Institutionalism in Organizational Analysis*. Chicago, University of Chicago Press. Pp 267-292.
- DiMaggio, Paul and Powell, Walter. 1991. "The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields." In Powell, Walter W and DiMaggio, Paul. "The New Institutionalism in Organizational Analysis." Chicago, University of Chicago Press. Pp 63-82.
- DiMaggio, Paul J. 1988. "Interest and Agency in Institutional Theory." In *Institutional Patterns and Organizations: Culture and Environment*. Edited by L. G. Zucker. Cambridge, MA: Ballinger.
- DiMaggio, Paul J. and Walter Powell. 1983. "The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields." *American Sociological Review* 48: 147-160.
- DiMaggio, Paul J. and Helmut K. Anheier. 1990. "The Sociology of Nonprofit Organizations and Sectors." *Annual Review of Sociology* 16: 137-159.
- Eisenger, Peter K. 1973. "The Conditions of Protest Behavior in American Cities." *American Political Science Review* 67: 11-68.
- Eisenhardt, Kathleen M. 1989. "Building Theories from Case Study Research." *Academy of Management Review* 14: 532-550.
- Eisinger, P. K. "The Conditions of Protest Behavior in American Cities." *American Political Science Review* 67: 11-28.

- Elsbach Kimberly D. 1994. "Managing Organizational Legitimacy in the California Cattle," *Administrative Science Quarterly* 39: 57-88.
- Elster, Jon. 1998. "A Plea for Mechanisms." In *Social Mechanisms: An Analytical Approach to Social Theory*. Edited by Peter Hedstrom and Richard Swedberg. New York: Cambridge University Press.
- Fernandez, R. M. and D. McAdam. 1988. "Social Networks and Social Movements: Multiorganizational Fields and Recruitment to Mississippi Freedom Summer," *Sociological Forum* 3: 441-460.
- Fligstein, Neil. 1985. "The Spread of the Multidivisional Form Among Large Firms, 1919-1979." *American Sociological Review* 50: 377-391.
- Fligstein, Neil. 1987. "The Intra-organizational Power Struggle: The Rise of Finance Presidents in Large Corporations," *American Sociological Review* 52: 44-58.
- Fligstein, Neil and Kenneth Dauber. 1989. "Structural Change in Corporate Organization." *Annual Review of Sociology* 15: 73-96 .
- Fligstein, Neil. 1990. "The Transformation of Corporate Control - Introduction." In *The Transformation of Corporate Control* . Edited by Neil Fligstein. Cambridge, Mass: Harvard University Press.
- Fligstein, Neil and Mara-Drita, Iona. 1996. "How to Make a Market: Reflections on the Attempt to Create a Single Market in the European Union. *American Journal of Sociology* 102: 1-33.
- Fligstein, Neil. 1996. "Markets as Politics: a Political-cultural Approach to Market Institutions." *American Sociological Review* 61: 656-673.
- Fligstein, Neil. 2001. "Social Skill and the Theory of Fields." Working Paper, University of California at Berkeley.
- Frey, R. Scott, Thomas Dietz, and Linda Kalof. 1992. "Characteristics of Successful American Protest Groups: Another Look at Gamson's *Strategy of Social Protest*." *American Journal of Sociology* 98: 368-387.
- Galaskiewicz, Joseph. 1985. *Social Organization of an Urban Grants Economy a Study of Business Philanthropy and Nonprofit Organizations*. Orlando: Academic Press.
- Galaskiewicz, Joseph and Ronald S. Burt. 1991. "Inter-organization Contagion in Corporate Philanthropy." *Administrative Science Quarterly* 36: 88-105.
- Galvin, Christopher J. 1999. "A Revolution from Within: Open Source Business Models in Infrastructure Software." Hambrecht and Quist Industry Report.
- Gamson, William A. 1992. "The Social Psychology of Collective Action." In *Frontiers of Social Movement Theory*. Edited by Aldon D. Morris and Carol McClurg

- Mueller. New Haven, Conn: Yale University Press.
- Gamson, William A. 1975. *The Strategy of Social Protest*. Belmont, Ca: Wadsworth.
- Garud, Raghu, Sanjay Jain, and Arun Kumaraswamy. 2002. "Institutional Entrepreneurship in the Sponsorship of Common Technological Standards: The Case of Sun Microsystems and Java." *Academy of Management Journal* 45: 196-214.
- Giddens, Anthony. 1984. *The Constitution of Society*. Berkeley, CA: University of California Press.
- Giugni, Marco G. 1998 "Was it Worth the Effort? The Outcomes and Consequences of Social Movements." *Annual Review of Sociology* 98: 371-93.
- Glaser, Barney G and Strauss, Anselm L. *The Discovery of Grounded Theory; Strategies for Qualitative Research*. Chicago: Aldine Pub. Co.
- Goffman, Erving. 1974. *Frame Analysis*. Cambridge: Harvard University Press.
- Goodrick, Elizabeth and Salancik, Gerald R. 1996. "Organizational Discretion in Responding to Institutional Practices: Hospitals and Cesarean Births." *Administrative Science Quarterly* 41: 1-28.
- Gould, R. V. 1993. "Collective Action and Network Structure." *American Sociological Review* 58: 182-196.
- Greenwood, Royston and C. R. Hinings. 1996. "Understanding Radical Organizational Change: Bringing Together the Old and New Institutionalism." *Academy of Management Review* 21: 1022-1054.
- Greenwood, Royston and C. R. Hinings. "Understanding Strategic Change: The Contribution of Archetypes." *Academy of Management Journal* 36: 1052-1081.
- Greenwood, Royston, Roy Suddaby, and C. R. Hinings. 2002. "Theorizing Change: The Role of Professional Associations in the Transformation of Institutional Fields." *Academy of Management Journal* 45: 58-80.
- Guttman, J. M. 1987. "A Non-Cournot Model of Voluntary Collective Action." *Economica* 54: 1-19.
- Hardin, Garrett. 1968. "The Tragedy of the Commons." *Science* 162: 1243-1248 .
- Haunschild, Pamela R. 1993. "Interorganizational Imitation The Impact of Interlocks on Corporate Acquisition Activity." *Administrative Science Quarterly* 38: 564-592.
- Haveman, Heather A. and Hayagreeva Rao. 1997. "Structuring a Theory of Moral Sentiments: Institutional and Organizational Coevolution in the Early Thrift

- Industry," *American Journal of Sociology* 102: 1606-1651.
- Hedstrom, Peter and Richard Swedberg. 1998. "Social Mechanisms: An Introductory Essay." Hedstrom, Peter and Richard Swedberg. *Social Mechanisms: An Analytical Approach to Social Theory*. Cambridge, Cambridge University Press. Pp 1-31.
- Hirsch, Paul. 1998. "Rejoinder to Scott." *American Journal of Sociology* 103 (4): 1049-1051.
- Hirsch, Paul M. 1975. "Organizational Effectiveness and the Institutional Environment." *Administrative Science Quarterly* 20: 327-344 .
- Hirsch, Paul M. 1997. "Sociology Without Social Structure: Neo-institutional Theory Meets Brave New World" *American Journal of Sociology* 102 : 1702-1723.
- Hirsch, Paul M. and Michael Lounsbury. 1997. "Ending the Family Quarrel: Toward a Reconciliation of "Old" and "New" Institutionalisms." *American Behavioral Scientist* 40: 406-418.
- Hirschman, Albert O. "Exit, Voice, and Loyalty; Responses to Decline in Firms, Organizations, and States." *Industrial Relations and Labor Economics* , Cambridge, MA: Harvard University Press.
- Hirshberg, Jeremy. 1998. *The Creative Priority: Driving Innovation in the Real World*. New York: Harper Business.
- Hoffman, Andrew J. 1999. "Institutional Evolution and Change: Environmentalism and the U.S. Chemical Industry." *Academy of Management Journal* 42: 351-371.
- Holm, Peter. 1995. "The Dynamics of Institutionalization: Transformation Processes in Norwegian Fisheries." *Administrative Science Quarterly* 40: 398-422.
- Homans, George C. 1986. "Fifty Years of Sociology." *Annual Review of Sociology* 12: xiii-xxx.
- Hughes, Everett C. 1936. "The Ecological Aspects of Institutions." *American Sociological Review* 1: 180-189.
- Hughes, Everett C. 1937. "Institutional Office and the Person." *American Journal of Sociology* 43: 404-413.
- Jenkins, Craig J. 1977. "Radical Transformation of Organizational Goals." *Administrative Science Quarterly* 22 (4): 568-586.
- Jepperson, Ronald. 1991. "Institutions, Institutional Effects, and Institutionalism." Powell, Walter W and DiMaggio, Paul. *The New Institutionalism in Organizational Analysis*. Chicago, University of Chicago Press. Pp 143-163.

- Johnson, Justin Pappas. 2001. "Economics of Open Source Software." Working Paper, Available at www.opensource.mit.edu.
- Kanter, Rosabeth M. 1968. "Commitment and Social Organization: a Study of Commitment Mechanisms in Utopian Communities." *American Sociological Review* 33: 499-517.
- Kraatz, Matthew S. and James H. Moore. 2002. "Executive Migration and Institutional Change." *Academy of Management Journal* 45: 120-143.
- Kraatz, Matthew S. and Edward J. Zajac. 1996. "Exploring the Limits of the New Institutionalism: The Causes and Consequences of Illegitimate Organizational Change." *American Sociological Review* 61: 812-836.
- Lakhani, K. and Von Hippel, E. 2000. "How Open Source Software Works: "Free" User-to-User Assistance." MIT Sloan School of Management Working Paper available at www.opensource.mit.edu.
- Lakhani, Karim and Wolf, Bob. 2002. "BCG Hacker Survey." MIT Sloan School of Management Working Paper.
- Lawrence, Thomas B., Cynthia Hardy, and Nelson Phillips. 2002. "Institutional Effects of Interorganizational Collaboration: The Emergence of Proto-Institutions." *Academy of Management Journal* 45: 281-290.
- Lee, Gwendolyn and Cole, R. E. 2000. "The Linux Kernel Development as a Model of Knowledge Creation." Working Paper available at www.opensource.mit.edu.
- Lee, Steve H. 1999. "Open Source Software Licensing." Harvard Law School.
- Lerner, Josh and Jean Tirole. 2000. "The Simple Economics of Open Source." *National Bureau of Economic Research Working Paper no. W7600*.
- Lessig, Lawrence. 1999. "Open Code and Open Societies: Values of Internet Governance." *Chicago-Kent Law Review* 74: 101-116.
- Levy, Steven. 1994. *Hackers: Heroes of the Computer Revolution*. New York: Penguin Books.
- Jonathan Littman. "Linus Torvalds." *Upside*, 10 August 1999.
- Lounsbury, Michael. 1997. "Exploring the Institutional Tool Kit: The Rise of Recycling in the U.S. Solid Waste Field." *American Behavioral Scientist* 40: 465-477.
- Lounsbury, Michael. 2001. "Institutional Sources of Practice Variation: Staffing College and University Recycling Programs." *Administrative Science Quarterly* 46: 29-56.

- Lounsbury, Michael. 2002. "Institutional Transformation and Status Mobility: The Professionalization of the Field of Finance." *Academy of Management Journal* 45: 255-266.
- Lounsbury, Michael. "Institutional Variation in the Evolution of Social Movements: The Spread of Recycling Advocacy Groups." Paper Presented at the Organizations and Social Movement Conference at the University of Michigan May 9, 2002.
- MacCormack, A and Herman, K. "Red Hat and the Linux Revolution." Harvard Business Review Case No 9-600-009. 2000.
- Malone, T., J. Yates, and R. I. Benjamin. 1987. "Electronic Markets and Hierarchies." *Communications of the ACM*.
- Malone, T. W. "Is Empowerment 'Just a Fad?' Control, Decision-Making and Information Technology." *Sloan Management Review* Winter (1997).
- Malone, T. W. and R. J. Laubacher. 1998. "The Dawn of the E-Lance Economy." *Harvard Business Review* September-October: 145-152.
- Marx, Gary T. and James L. Wood. 1975. "Strands of Theory and Research in Collective Behavior." *Annual Review of Sociology* 1: 363-428.
- Marx, Karl. 1976. *Capital -Vol. I*. London: Penguin Classics.
- McAdam, Doug. 1995. "Initiator" and "Spin-off" Movements: Diffusion Processes in Protest Cycles. In *Repertoires & Cycles of Collective Action*. Edited by Mark Traugott. Durham, NC: Duke University Press.
- McAdam, Doug. 1982. *Political Process and the Development of Black Insurgency, 1930-1970*. Chicago, IL: University of Chicago Press.
- McAdam, Doug. 1986. "Recruitment to High Risk Activism. The Case of Freedom Summer." *American Journal of Sociology* 92: 64-90.
- McAdam, Doug. 1983. "Tactical Innovation and the Pace of Insurgency." *American Sociological Review* 48: 735-754.
- McAdam, Doug, John D. McCarthy, and Mayer N. Zald. 1996. "Introduction: Opportunities, Mobilizing Structures and Framing Processes - Toward a Synthetic, Comparative Perspective on Social Movements." In *Comparative Perspectives on Social Movements*. Edited by Doug John D. McCarthy and Mayer N. Zald McAdam. Cambridge: Cambridge University Press.
- McAdam, Doug, John D. McCarthy, and Mayer N. Zald. "Social Movements." In *Handbook of Sociology*. Edited by N. Smelser. Beverly Hills, CA: Sage.
- McAdam, Doug and Ronnelle Paulsen. 1993. "Specifying the Relationship Between

- Social Ties and Activism." *American Journal of Sociology* 99: 640-667.
- McAdam, Doug and Scott, W. Richard. "Organizations and Movements." Presented at the Conference on Organizations and Social Movements at the University of Michigan Business School, May 9 2002.
- McCarthy, J. D. and Mayer N. Zald. 1977. "Resource Mobilization and Social Movements: A Partial Theory." *American Journal of Sociology* 82: 1212-1241 (1977).
- Merton, Robert K. 1949. *Social Theory and Social Structure*. Glencoe, IL: The Free Press.
- Meyer, John and Rowan, Brian. 1991. "Institutionalized Organizations: Formal Structure as Myth and Ceremony." In Powell, Walter W and DiMaggio, Paul. *The New Institutionalism in Organizational Analysis*. Chicago, University of Chicago Press. Pp 41-62.
- Meyer, John W and Brian Rowan. 1977. "Institutionalized Organizations: Formal Structure as Myth and Ceremony." *American Journal of Sociology* 83: 340-63.
- Michels, Robert. 1911. *Political Parties: A Sociological Study of the Oligarchical Tendencies of Modern Democracy*. New York, NY: The Free Press.
- Miller, David. 1981. "Market Neutrality and the Failure of Co-Operatives." *British Journal of Political Science* 11: 309-329.
- Mills, Harlan D. 1971. "Top-down Programming in Large Systems." In *Debugging Techniques in Large Systems*. Edited by R. Rustin. Englewood Cliffs, NJ: Prentice-Hall.
- Minkoff, Debra C. 1991. "Bending with the Wind: Strategic Change and Adaptation by Women's and Radical Minority Organizations." *American Journal of Sociology* 104: 1666-1703.
- Mockus, A., Fielding, Roy, and Herbsleb, James. "A Case Study of Open Source Software Development: The Apache Server." Proceedings of the International Conference on Software Engineering.
- Moglen, Eben. 1999. "Anarchism Triumphant: Free Software and the Death of Copyright," *Saggi, Conferenze E Seminari (38), Roma*.
- Moody, Glyn. 2001. *Rebel Code: Inside Linux and the Open Source Revolution*. New York, NY: Harper Collins.
- Moon, Jae Yun and Lee Sproull. 2002. "Essence of Distributed Work: The Case of the Linux Kernel." In *Distributed Work*. Edited by Pamela Hinds and Sara Kiesler. Cambridge, MA: MIT Press.
- Morris, Aldon. 1993. "Birmingham Confrontation Reconsidered." *American*

Sociological Review 58: 621-36.

- Nadel, Siegfried F. 1957. *The Theory of Social Structure*. Glencoe, Ill: Free Press.
- Nelsen, Bonalyn J and Barley, Stephen R. 1997. "For love or money? Commodification and the construction of an occupational mandate." *Administrative Science Quarterly* 42: 619-653.
- O'Mahony Siobhan and Barley Stephen R. 1999. "Do Digital Telecommunications Affect Work And Organization? The state of our knowledge," *Research in Organizational Behavior* 21: 125-161.
- Oliver, Christine. 1991. "Strategic Responses to Institutional Processes." *Academy of Management Review* 16: 145-179.
- Olson, Mancur. 1965. *The Logic of Collective Action*. New York: Schocken.
- Orlikowski, Wanda J. 1993. "CASE tools as Organizational Change Investigating Incremental and Radical Changes in Systems Development." *MIS Quarterly* 17: 309-340.
- Orlikowski, Wanda J. 1996. "Improvising Organizational Transformation Over Time A Situated Change Perspective." *Information Systems Research: ISR: a Journal of the Institute of Management Sciences* 7: 63-92.
- Orlikowski, Wanda J. and Yates, JoAnne. 1994. "Genre repertoire: The structuring of communicative practices in organizations." *Administrative Science Quarterly* 39: 541-574.
- Orlikowski, Wanda J. et al. 1995. "Shaping electronic communication: the meta-structuring of technology in the context of use," *Organization Science* 6 (4): 423-444 (1995).
- Ostrom, Elinor. 1999. "Coping With Tragedies of the Commons." *American Review of Political Science* 2: 493-535.
- Ostrom, Elinor. 1990. *Governing the Commons: The Evolution of Institutions for Collective Action*. New York, NY: Cambridge University Press.
- Ostrom, Elinor et al. 1999. " Revisiting the Commons: Local Lessons, Global Challenges." *Science* 284: 278-282.
- Palmer, Donald A., Jennings P. Devereaux, and Zhou Xueguang. 1993. "Late Adoption of the Multidivisional Form by Large US Corporations: Institutional, Political, and Economic Accounts." *Administrative Science Quarterly* 38: 100-131.
- Park, Robert E. 1921. "Sociology and the Social Sciences: the Group Concept and Social Research." *American Journal of Sociology* 27: 169-183.

- Parsons, Talcott. 1956. "A Society of Organizations." *Administrative Science Quarterly* 1: 63-85.
- Patterson, Lyman Ray. 1968. *Copyright in Historical Perspective*. Nashville, TN: Vanderbilt University Press.
- Perens, Bruce. 1999. The Open Source Definition. In *Open Sources*. Edited by Chris Ockman Sam Stone Mark DiBona. Sebastopol, CA: O'Reilly.
- Perrucci Robert. 1973. Engineering: Professional Servant of Power. In *The Professions and their Prospects*. Edited by Eliot Freidson. Beverly Hills : Sage Publications.
- Pfeffer, Jeffrey and Gerald R Salancik. 1978. *The External Control of Organizations: a Resource Dependence Perspective*. New York: Harper & Row.
- Piven, Frances Fox and Richard A. Cloward. 1977. *Poor People's Movements: Why They Succeed, How They Fail*. New York, NY: Pantheon Books.
- Rao, Hayagreeva. 1998. "Caveat Emptor: The Construction of Nonprofit Consumer Watchdog Organizations." *American Journal of Sociology* 103: 912-961.
- Rao, Hayagreeva, Calvin Morrill, and Mayer N. Zald. 2000. "Power Plays: How Social Movements and Collective Action Create New Organizational Forms." *Research in Organizational Behavior* 22: 237-281.
- Raymond, Eric S. 1999. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly & Associates.
- Rheingold, Howard. 2000. *The Virtual Community*. Cambridge, MA: The MIT Press.
- Rose, Mark. 1993. *Authors and Owners: The Invention of Copyright*. Cambridge, MA: Harvard University Press.
- Rosenberg, Donald K. 2000. *Open Source: The Unauthorized White Papers*. Vol. . ed. Edited by . Foster City, CA: IDG Books Worldwide, Inc.
- Rothschild-Whitt, Joyce. 1979. "The Collectivist Organization: An Alternative to Rational-Bureaucratic Models." *American Sociological Review* 44: 509-527.
- Schlager, E. "Fishers' Institutional Responses to Common-Pool Resource Dilemmas." In *Rules, Games, and Common-Pool Resources*. Edited by E. Ostrom, R. Gardner, and J. M. Walker. Ann Arbor, MI: University of Michigan Press.
- Schneiberg, Marc and Elisabeth S. Clemens. forthcoming. "The Typical Tools for the Job: Research Strategies in Institutional Analysis." In *How Institutions Change*. Edited by Walter W. Powell and Dan L. Jones. Chicago: University of Chicago Press.
- Scott, W. Richard. "A Call for Two-Way Traffic: Improving the Connection

- Between Social Movement and Organization/Institutional Theory." Stanford University. A paper presented for presentation at the "Zaldfest": A conference to honor Mayer D. Zald, University of Michigan, September 17 and 18, 1999.
- Scott, W. Richard. 1995. *Institutions and Organizations*. Thousand Oaks, CA: Sage.
- Scott, W. Richard et al. 2000. *Institutional Change and Healthcare Organizations: From Professional Dominance to Managed Care*. First ed. Chicago: University of Chicago Press.
- Selznick, Philip. 1996. "Institutionalism "old" and "new". " *Administrative Science Quarterly* 41: 270-277.
- Selznick, Philip. 1949. *TVA and the Grass Roots: A Study of Politics and Organization* . Berkeley: University of California Press.
- Selznick, Phillip. 1992. *The Moral Commonwealth: Social Theory and the Promise of Community*. Berkeley: University of California Press.
- Sewell, William H. Jr. 1992. "A Theory of Structure: Duality, Agency, and Transformation." *American Journal of Sociology* 98: 1-29.
- Sexton, Richard J. and Terri A. Sexton. 1987. "Cooperatives as Entrants," *The RAND Journal of Economics* 18: 581-595.
- Simons, Tal and Ingram, Paul. 1997. "Organization and Ideology: Kibbutzim and hired labor, 1951-1965." *Administrative Science Quarterly* 42: 784-814.
- Sneath, D. 1998. "State Policy and Pasture Degradation in Inner Asia." *Science* 281: 1147.
- Snow, David A. and Robert D. Benford. 1992. "Master Frames and Cycles of Protest." In *Frontiers of Social Movement Theory*. Edited by Aldon D. Morris and Carol M. Mueller. New Haven: Yale University Press.
- Snow, David A. et al. 1986. "Frame Alignment Processes, Micro-mobilization, and Movement." *American Sociological Review* 51: 464-481.
- Sorensen, Aage B. 1998. "Theoretical Mechanisms and the Empirical Study of Social Processes." In *Social Mechanisms: An Analytical Approach to Social Theory*. Edited by Peter Hedstrom and Richard Swedberg. New York: Cambridge University Press.
- Spradley, James P. 1979. *The Ethnographic Interview*. Orlando: Harcourt Brace Jovanovich.
- Staber, U. H. 1989. "Organizational Foundings in the Cooperative Sector in Atlantic Canada: An Ecological Perspective." *Organization Studies* 10: 383-405.

- Staggenborg, Suzanne. 1988. "The Consequences of Professionalization and Formalization in the Pro-Choice Movement." *American Sociological Review* 53: 585-605.
- Stallman, Richard. 1999. "The GNU Operating System and the Free Software Movement." In *Open Sources*. Edited by Chris DiBona, Sam Ockman, and Mark Stone. Sebastopol, CA: O'Reilly.
- Stern, Robert N. and Stephen R. Barley. 1996. "Organizations and Social Systems: Organization Theory's Neglected Mandate." *Administrative Science Quarterly* 41: 146-162.
- Strauss, Anselm and Juliet Corbin. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA: Sage Publications.
- Strauss, Anselm L. 1987. *Qualitative Analysis for Social Scientists*. New York: Cambridge University Press.
- Swaminathan, Anand and James B. Wade. 2001. "Social Movement Theory and the Evolution of New Organizational Forms.: In *The Entrepreneurship Dynamic in Industry Evolution*. Edited by C. B. Schoonhoven and E. Romanelli. Palo Alto, CA: Stanford University Press.
- Swidler, Ann. 1986. "Culture in Action: Symbols and Strategies." *American Sociological Review* 51: 273-286.
- Swidler, Ann. 1979. *Organization without Authority: Dilemmas of Social Control in Free Schools*. Cambridge, MA: Harvard University Press.
- Tiemann, Michael. 1999. "Future of Cygnus Solutions: An Entrepreneur's Account." In *Open Sources*. Edited by Chris DiBona, Sam Ockman, and Mark Stone. Sebastopol, CA: O'Reilly.
- Tolbert, Pamela S. and Lynne G. Zucker. 1983. "Institutional Sources of Change in the Formal Structure of Organizations: The Diffusion of Civil Service Reform, 1880-1935." *Administrative Science Quarterly* 28: 296-313.
- Torvalds, Linus. 1999. "The Linux Edge." In *Open Sources*. Edited by Chris Ockman Sam Stone Mark DiBona. Sebastopol, CA: O'Reilly.
- Townley, Barbara. 2002. "The Role of Competing Rationalities in Institutional Change." *Academy of Management Journal* 45: 163-179.
- Tuomi, Ilkka. 2000. "Internet, Innovation, and Open Source: Actors in the Network." ITRA, The Finnish National Fund for Research and Development Working Paper.
- Turkle, Sherry. 1997. *Life on the Screen: Identity in the Age of the Internet*. Touchstone Books.

- Van Maanen, J. and S. R. Barley. 1984. "Occupational Communities: Culture and Control in Organizations." *Research in Organizational Behavior* 6: 287-365.
- Von Hippel, Eric. 2001. "Open Source Shows the Way: Innovation By and For Users - No Manufacturer Required." *Sloan Management Review* 42: 82-85.
- Voss, Kim. 1996. "The Collapse of a Social Movement: The Interplay of Mobilizing Structures, Framing, and Political Opportunities in the Knights of Labor." In *Comparative Perspectives on Social Movements*. Edited by Doug McAdam, John D. McCarthy, and Mayer n. Zald. New York: Cambridge University Press.
- Voss, Kim and Rachel Sherman. 2000. "Breaking the Iron Law of Oligarchy: Union Revitalization in the American Labor Movement." *American Journal of Sociology* 106: 303-349.
- Wade, James B., Anand Swaminathan, and Michael Scott Saxon. 1998. "Normative and Resource Flow Consequences of Local Regulations in the American Brewing Industry, 1845-1918." *Administrative Science Quarterly* 43: 905-935.
- Weick, Karl E. 1974. "Amendments to Organizational Theorizing." *Academy of Management Journal* 17: 487-502.
- Wenger, Etienne. 1998. *Communities of Practice: Learning, Meaning, and Identity*. Vol. . ed. Edited by . Cambridge, UK: Cambridge University Press.
- Westphal, James D, Gulati, Ranjay, and Shortell, Stephen M. 1997. "Customization or conformity? An institutional and network perspective on the content and consequences of TQM adoption." *Administrative Science Quarterly* 42: 366-394.
- White, Harrison. 1992. *Identity and Control*. Princeton, NJ: Princeton University Press.
- Yin, Robert K. 1981. "The Case Study Crisis Some Answers." *Administrative Science Quarterly* 26: 58-65.
- Zald, Mayer N. and Roberta Ash. 1966. "Social Movements Organizations: Growth, Decay and Change." *Social Forces* 44: 327-41.
- Zander and Kogut. 1995. "Knowledge and the Speed of Transfer and Imitation of Organizational Capabilities: An Empirical Test." *Organization Science* 6.
- Zilber, Tammar B. 2002. "Institutionalization as an Interplay between Actions, Meanings, and Actors: The Case of a Rape Crisis Center in Israel." *Academy of Management Journal* 45: 234-254.
- Zucker Lynne G. 1977. "The Role of Institutionalization in Cultural Persistence," *American Sociological Review* 42: 726-743.

Table II-1: Comparison of Traditional and Community Software Development Models

	Traditional Software Development	Open Source Development
Initiator/ Owner	Proprietary firm initiates project and owns the intellectual property associated with it.	Individuals or firms initiate projects and may retain copyright or transfer it to a non-profit.
Team/ Membership Composition	Organized by a firm appointed Project Manager – may include organizational members and contractors who are provided access to proprietary data.	Anyone can participate from anywhere in the world. Core team emerges based on merit and effort. Decision-making at higher levels by invitation from the core team.
Technical Context	Development site is hosted within organizational firewalls.	Development site is hosted by an individual or non-profit. Firms may donate resources.
Resources	Firms dedicate labor and technical resources. Project may halt or change based on changes in strategy or the market.	Labor is mostly self-selected volunteers. Technical resources may be donated. Project will continue as long as there is interest.
Technical Specifications	Technical requirements are based on customer analysis with input from marketing at early stages in the project.	Ideas emerge from wide pool of contributors and are accepted or rejected by the core team throughout the project lifecycle based on their ability to help the project.
Release Patterns	Customer requests for changes may be incorporated into new versions. There may be a long lag time between versions with little on-going development visible.	New versions are 'released early and often'. Development between versions is transparent. Stable and non-stable versions are available – work in progress is visible on the project site.
Distribution Terms	Proprietary licenses and copyrights that restrict use, modification, and distribution predominate. Only binary code is shipped.	Source code is available over the Internet. Copies can also be sold. OS licenses permit change and redistribution with some user requirements.
Customers	Governments, education, large firms, small and medium firms, and consumers.	Other open source developers, firms in the computer industry and technical users.
User/ Developer Relations	Relations with other developers and firms working on related applications are managed from a corporate perspective. They have little access to work in progress.	Expanded base of users is critical for beta-testing and bug fixing work in progress. Few fixed deadlines limit participation. Users may progress to developers if qualified.

Table II-2: Frame Comparison

Frame Characteristic	Free Software Frame	Open Source Frame
Diagnosis (Problem identification and locus of attribution)	Proprietary software is immoral – all software should be free	Open source software should be more widely adopted - it leads to better quality code
Elasticity/Inclusivity	Rigid	Flexible
Resonance – Commercial	Low	High
Narrative Fidelity to Founding Movement	High	Low

Insert Figure II-1

Insert Figure II-2

Table III-1: Theoretical Rationale for Project Selection

Project Case Study	Theoretical Rationale
Websaver Project – a project to develop and maintain a commercial grade web server	One of the first to forge formal commercial relations with a Fortune 500 firm, this project sheds light on how community structures interact with corporate structures.
GUI Desktop Project – a project to develop a user friendly graphical user interface for the Linux desktop	Because further commercial adoption of Linux depends on GUI improvements, innovation in this area is critical. It has also caught the attention of several large firms who would like to improve user interfaces. As the only application project, this project offers contrast with other projects at the infrastructural layer.
Standard Setting Project – a project to develop standards for all Linux distributions	An open source project dedicated to creating a standard to improve interoperability among Linux distributions, this project will have crosscutting effects on both community and corporate projects.
Linux Distribution Project – a noncommercial Linux distribution	The largest and only non-commercial Linux distribution, this project has a stronger identification with the free software frame and provides greater contrast to projects that have more interactions with commercial entities.

Figure III-1: Estimated Project Affinity for Commercial Relations

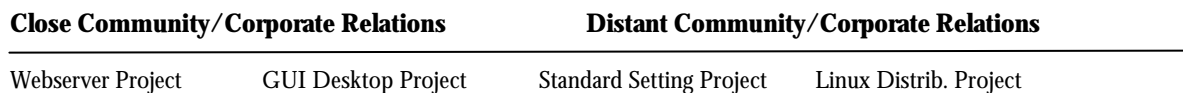


Table III-2: Project Attributes

Project Names/ Project Attributes	Webserver	GUI Desktop	Standard Setting	Linux Distribution
Project Mission/Goal	To create a commercial grade freely available webservice	To build a free and easy to use desktop environment	To develop & promote standards to increase compatibility among Linux distributions	To develop a free non-commercial operating system
Date Project Founded	Feb-95	Aug-97	Jun-95	Aug-93
Date of first release	April-95	Jun-98	May-98	Jan-94
License(s) used	BSD type	GPL, LGPL	GPL	GPL
Foundation Formed	Yes	Yes	Yes	Yes
Type of Corporation	Public Benefit	Public Benefit	Mutual Benefit	Public Benefit
Date Incorporated	Jun-99	Aug-00	May-00	Jun-97
Non-Profit Status	501c(3)	501c(3)	501c(6)	501c(3)
Date Awarded	Apr-01	pending	2001	Jun-99
Membership Association	Yes	Yes	Yes	Yes
Companies as members	No	No	Yes	No
Board officers	Elected	Elected	Elected	Apptd
Number of Board Directors	9	11	9	8
Slated Board	No	Yes	3-C/2-NP/4-I	No
Fortune 500 participation	Yes	Yes	Yes	Yes
Corporate Roles	Firms can be represented by individuals on boards	Advisory Board	Advisory Committee	Partner Program

C = Corporate
 NP = Non-Profit
 I = Independent

Table III-3: Informant Demographics

	Percent
Male	0.93
Female	0.07
Independent	0.36
Corporate Sponsored	0.64
Identified through face to face contact	0.47
Identified through Internet or introductions	0.53
Interviewed Face to Face	0.84
Interviewed By Phone	0.16
Participation in Trust Network*	0.41
(Ratings of those participating)	
Observer	0.03
Apprentice	0.07
Journeyer	0.34
Master	0.55
N = 70	
<p>* Some members of the free software and open source communities participate in a trust network called Advogato that allows community members to evaluate each other. Individuals can rank each other's technical ability at observer, apprentice, journeyer, and master levels. The trust network was not used in the formation of this sample as it was discovered during the course of the study. An ex postea search on informants that participated in this study indicates that 41% of them participated in the network. This data is included to show how those informants who participated on this study were regarded by their peers.</p>	

Table V-1: Changes in Community Practices

Traditional Practice	Adapted Practice	In Practice	Community Projects			
			Web-server	Linux Distribution	GUI Desktop	Standard Setting
Free software frame is organizing construct	Embrace open source frame	“By doing so [changing from free software to open source], we both safeguarded the hacker community’s interests and made open source attractive to software producers and consumers as a brand.” (<i>An open source movement Founder</i>)	I, D, S	----	----	I, D
Loosely defined community with no legal status	Incorporate/ Create board of directors	“The community isn’t enough in itself. You also need the legal protections of a real corporation. You need money to actually hire lawyers to deal with the things that we don’t want to deal with. We need to be able to sign contracts.” (<i>A Founder and Board member, Webservice Project</i>)	I, D, S	I, D, S	I, D, S	I, D, S
Only individuals can contribute to community projects	Allow firm participation or engage in agreements with firms	“They [Fortune 500 firm ABC] had to earn their way in, and they did. Over the course of the few months, they you know made solid contributions of code, from [firm] addresses.” (<i>A Project Founder and Board member, Webservice Project</i>)	I, D	I, D	I, D, S	I, D, S
Volunteers contribute on their own terms	Formalize mechanisms to represent project	“People want to talk to [Linux Distribution Project] so what are they going to try and contact? They are going to contact the project leader....A lot of people still don’t understand the open source hierarchy. So they want to talk to the boss, the guy in charge. Who is that? I don’t know, we will say he is.” (<i>Sponsored Contributor, Linux Distribution Project</i>)	I, D, S	I, D, S	I, D, S	I, D, S

Key

- Supported by Interview data = I
- Supported by Project Documentation = D
- Supported by Secondary Sources = S
- Not Supported = ----

Table V-2: Community Preservation Tactics

Tactic	In Practice	Community Projects			
		Web-server	Linux Distribution	GUI Desktop	Standard Setting
Limit authority and role of foundation	“Heavy bureaucracy is not in our DNA. And it shouldn’t be. So let’s not try to graft an administrative superstructure over the community we’ve built.” (<i>GUI Desktop Project Charter</i>)	I, D	I, D	I, D	I, D
Delimit roles available to firms	“I did have this idea that business did have to cooperate using our rules. But it was later when we started to apply it and we started to have companies participate that we realized you know this is something really good” (<i>Former Project Leader, Linux Distribution Project</i>)	I, D	I, D	I, D	I, D
Protect community’s intellectual property	“There have been cases where people have called their consulting company [Webserver Project] or put [Webserver Project] on a book or software that made it sound like something that we were doing and it just, we have gone out and essentially defended the brand.” (<i>Founding Member, Webserver Project</i>)	I, D	I, D	I, D	I, D

Key

- Supported by Interview data = I
- Supported by Project Documentation = D
- Supported by Secondary Sources = S
- Not Supported = ----

Table VI-1: Changes in Firm Practices

Traditional Practice	Adapted Practice	In Practice	Community Managed Projects			
			Web-server	Linux Distribution	GUI Desktop	Standard Setting
Employees represent the firm	Employers allow individuals to represent themselves	“[Fortune 500 firm ABC], they understood that the [firm] programmers were not representing [the firm] on this mailing list.” <i>(Former Volunteer, Sponsored Contributor, Webserver Project)</i>	I, D	I, D	I, D	I, D
Employees sign the rights to their work to employers	Employers allow individuals to retain work assignment rights	“The only thing down in paper that I require is basically saying that the work I do on [the project] is my work. The work I do for [Qualtech] is the property of [Qualtech] corporation on [Qualtech] issues. But when I work on [the project]... I have paperwork that says that that is mine.” <i>(Former Volunteer, Sponsored Contributor, Linux Distribution Project)</i>	I	I	I	I
Proprietary code is kept within the firm	Firms release code to community projects	“We employ several [Linux Distribution Project] developers and while they are working on projects that are interesting to us, there is a lot of overlap, in terms of what [the project] can use. So all of the work we have done on [the project] we are offering to [them].” <i>(Firm Founder, Sponsored Contributor, Former Volunteer, Linux Distribution Project)</i>	I, D, S	I, D, S	I, D, S	I, D, S
Public/private realms are clearly demarcated - firms may partner with firms but development is private	Firms adapt work style to program and work with a collective in public forums	You can browse the archives of any CVS lists. You can see every change that our developers mark. This is naked programming. We’re developing, sort of, completely in the open, and what we do at some point, we’ll publish our commercial release dates. <i>(Product Manager, Fortune 500 firm, Sponsored Contributor, GUI Desktop Project)</i>	I, D	----	I, D, S	I, D, S

Key

- Supported by Interview data = I
- Supported by Project Documentation = D
- Supported by Secondary Sources = S
- Not Supported = ---

Table VI-2: Firm Preservation Tactics

Tactic	In Practice	Community Managed Projects			
		Web-server	Linux Distribution	GUI Desktop	Standard Setting
Sponsor contributors in key roles to shape priorities	<p>“Now as much as we were individuals we did not have carte blanche to do what we wanted to do. We were working for [Fortune 500 Firm ABC]. So you know, [the firm] said here is a project, get this into the core.”</p> <p><i>(Sponsored Contributor, Webserver project)</i></p>	I, D	I, D	I, D	I, D
Donate resources	<p>“Now it is an intellectual challenge, it is no longer commercial pressure. See [Fortune 500 firm] going [to the community] to say we need A, B, and C is commercial pressure. And they are hostile to that [Fortune 500 firm] donating money to a lab, that creates an intellectual challenge and so, it is not commercial pressure any more, they are once removed from pressuring kernel developers.”</p> <p><i>(Sponsored Contributor, Standard Setting Project)</i></p>	I, D, S	I, D, S	I, D, S	I, D, S
Shape foundation/ Gain access to the board	<p>“At the board level, what’s going on is that we have folks that are theoretically representing only themselves, as a practical matter. If I’m on there, right? I care deeply about [the project], but I’m sitting here all day long, and I also care deeply about [Firm Zeta]. So, I definitely stand up for [Firm Zeta’s] interests.”</p> <p><i>(Sponsored Contributor, Board Member, GUI Desktop Project)</i></p>	I, D	----	I, D, S	I, D, S

Key
 Supported by Interview data = I
 Supported by Project Documentation = D
 Supported by Secondary Sources = S
 Not Supported = ----

Table VII-1: Comparison of Foundation Attributes

		Project Names				
		Linux Distrib- ution	Web- server	GUI Desktop	Standard Setting	Adoption Ratio
Preserving Attributes of Foundation						
	Function					
1	Has no direct authority on technical decisions	✍	✍	✍	✍	3/4
2	Elects board		✍	✍	✍	3/4
3	Limits corporate interests on board			✍	✍	2/4
4	Holds trademarks and brand identity rights	✍	✍	✍	✍	4/4
5	Holds copyright and code ownership rights		✍	✍		2/4
6	Creates a role for firms without decision rights	✍		✍	✍	3/4
# of Preserving Attributes Adopted		3	4	5	5	
		Project Names				
		Linux Distrib- ution	Web- server	GUI Desktop	Standard Setting	Adoption Ratio
Accommodating Attributes of Foundation						
	Function					
1	Has release coordination authority			✍		1/4
2	Can set technical direction			✍		1/4
3	Employs staff			✍	✍	2/4
4	Can represent project/public relations responsibility	✍	✍	✍	✍	4/4
5	Allows firms to be members				✍	1/4
6	Board members can represent employers		✍		✍	2/4
# of Accomodating Attributes Adopted		1	2	4	4	

Table VII-2: Comparison of Organizational Forms

Organizational Attributes	Firm	Coop- erative	Private Software Foundation	Charitable Organ- ization	Industry Consortia	Trade Association	Employee Association	Union	Internet mailing list
IRS Non-Profit Classification	N/A	521a	501c3, 501c6	501c3	501c6	501c6	501c4, 501c9	501c5	N/A
Incorporated	✘	✘	✘	✘	✘	✘	✘	✘	
Produces goods/service for profit	✘								
Represents firms	✘			✘	✘	✘			
Organized by and for individual members		✘	✘	✘			✘	✘	✘
Produces benefits for the public			✘	✘					
Produces benefits for a defined collective		✘	✘	✘	✘	✘	✘	✘	✘
Redistributes profits	✘	✘							
Owens assets sold on commercial market	✘	✘	✘						
May compete with firms	✘	✘	✘						

Notes

Shaded area is the new form: private non-profit software foundations

Areas outlined in black are attributes that have been recombined to shape this new organizational form

Figure II-1: Frame Transformation and Media Mentions

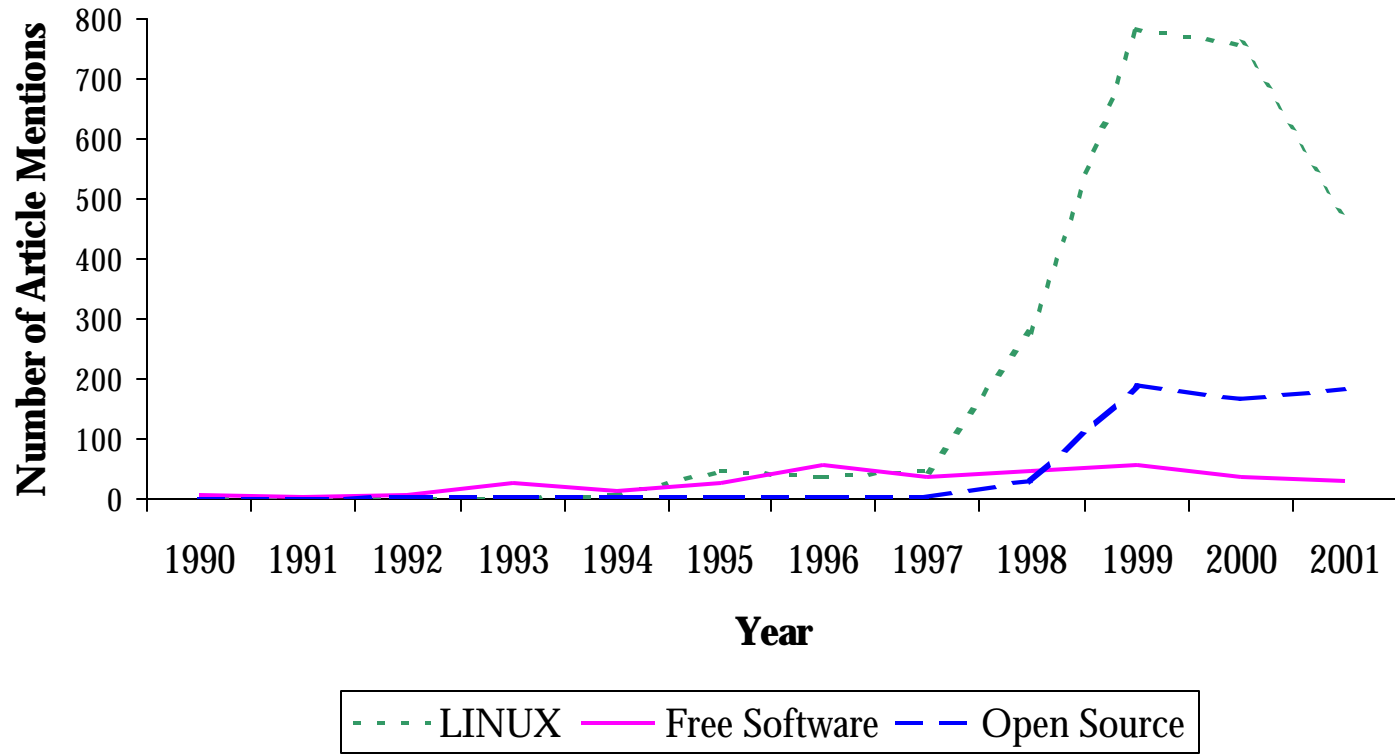


Figure II-2: Open Source and Linux Firms Receiving Round One Venture Capital Funding

