THE IMPACT OF IDEOLOGY ON EFFECTIVENESS IN OPEN SOURCE SOFTWARE DEVELOPMENT TEAMS

Katherine J. Stewart (kstewart@rhsmith.umd.edu)
Department of Decision and Information Technologies
R. H. Smith School of Business
University of Maryland
College Park, MD 20742
301-405-0576 (phone)
301-405-8655 (fax)

Sanjay Gosain (sgosain@rhsmith.umd.edu)
Department of Decision and Information Technologies
R. H. Smith School of Business
University of Maryland
College Park, MD 20742
301-405-3224 (phone)
301-405-8655 (fax)

Last Revised April 2005
Forthcoming in *MIS Quarterly* in 2006

THE IMPACT OF IDEOLOGY ON EFFECTIVENESS IN OPEN SOURCE SOFTWARE DEVELOPMENT TEAMS[1]

ACKNOWLEDGEMENTS: We thank the senior editor on the manuscript, V. Sambamurthy, and the anonymous associate editor and reviewers for their many insightful suggestions on earlier versions of this paper. We also thank Ritu Agarwal, Marc-David Seidel, and Paul Zantek for helpful comments. The first author's work on this research was partially supported by National Science Foundation award IIS-0347376. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily represent the views of the National Science Foundation.

Katherine J. Stewart is an assistant professor of information systems at the R. H. Smith School of Business, University of Maryland at College Park. She received her Ph.D. from the University of Texas at Austin in 2000. Her current research focuses primarily on trust in virtual settings and organizational issues related to the development of open source software.

Sanjay Gosain is an assistant professor of information systems at the R. H. Smith School of Business, University of Maryland at College Park. He received his Ph.D. from the University of Southern California in 2000. His current research focuses primarily on issues related to the development of open source software and design of IT infrastructure.

[1] An earlier version of this paper was presented in the research-in-progress track at the International Conference on Information Systems, December, 2001.

THE IMPACT OF IDEOLOGY ON EFFECTIVENESS IN OPEN SOURCE SOFTWARE DEVELOPMENT TEAMS

ABSTRACT

The emerging work on understanding open source software has questioned what leads to effectiveness in OSS development teams in the absence of formal controls, and it has pointed to the importance of ideology. This paper develops a framework of the OSS community ideology (including specific norms, beliefs, and values) and a theoretical model to show how adherence to components of the ideology impacts effectiveness in OSS teams. The model is based on the idea that the tenets of the OSS ideology motivate behaviors that enhance cognitive trust and communication quality and encourage identification with the project team, which enhances affective trust. Trust and communication in turn impact OSS team effectiveness. The research considers two kinds of effectiveness in OSS teams, the attraction and retention of developer input and the generation of project outputs. Hypotheses regarding antecedents to each are developed. Hypotheses are tested using survey and objective data on OSS projects. Results support the main thesis that OSS team members' adherence to the tenets of the OSS community ideology impacts OSS team effectiveness and reveal that different components impact effectiveness in different ways. Of particular interest is the finding that adherence to some ideological components was beneficial to the effectiveness of the team in terms of attracting and retaining input, but detrimental to the output of the team. Theoretical and practical implications are discussed.

KEYWORDS: Open Source Software, Trust, Ideology, Communication, Virtual Teams

ISRL categories: DA01, DA06, DD02

THE IMPACT OF IDEOLOGY ON EFFECTIVENESS IN OPEN SOURCE SOFTWARE DEVELOPMENT TEAMS

INTRODUCTION

While the practices associated with Open Source Software (OSS) development have been in use for decades, recent years have seen a surge of interest in OSS across developers, businesses, governments, and researchers. Tens of thousands of projects have been registered on Sourceforge, a website devoted to supporting OSS development (http://www.sourceforge.net), and many large companies have become involved in OSS[2]. Similarly, government institutions have begun trying to create effective policy with regard to open source software and development practices[3]. As the phenomenon of OSS has grown, researchers in many fields ranging from sociology to computer science, economics, and information systems have started studying it, and a rich base of literature has begun to appear.

This literature often views OSS teams as kinds of virtual teams or organizations (Crowston and Scozzi 2002; Gallivan 2001; Malone and Laubacher 1998; Markus et al. 2000), but with some unique aspects. OSS teams are often composed of volunteers working without financial remuneration directly tied to their contributions, and their output (e.g., source code) is generally made available to any interested users with little or no charge. Further, these teams usually work without the formal a priori requirements that tend to be used to guide and evaluate commercial software development efforts (Scacchi 2002). These unique aspects of OSS have led many to question why developers contribute (Hars and Ou 2002) and how their dispersed efforts are controlled in order to result in viable outputs (Gallivan 2001).

[2] IBM contributed code fixes and features to Apache (Ljungberg 2000). Dell, HP, and Oracle support versions of their products for the Linux platform (Gallivan 2001).
[3] A number of third world countries, for example, are proposing to use GNU/Linux for low cost computing platforms. (http://www.linuxjournal.com/article.php?sid=6049)

Answers to these questions may lie, in part, in another unique feature of OSS development: the ideology of the OSS development community. Researchers and developers have identified ideological tenets associated with the open source development community in which OSS teams function, and several have suggested that it may be the ideology that facilitates team effectiveness in the absence of traditional organizational incentives and controls (Bergquist and Ljungberg 2001; Markus et al. 2000; Raymond 2001). This research investigates that proposition by identifying the key ideological tenets of the OSS community and examining means by which they may facilitate three important outcomes related to OSS team effectiveness: attraction and retention of developers to projects, devotion of effort by project developers, and the completion of recognized work tasks. The basic thesis underlying the study is that OSS team members' adherence to the tenets of the overarching OSS community ideology may have a positive effect on these outcomes because the ideology fosters trust and good communication practices by encouraging behaviors and orientations that are beneficial to the team's work.

The next section provides a brief discussion of the nature of ideology and the role it plays in OSS teams. The paper then summarizes the main tenets of the OSS ideology and discusses the relevance of the three aforementioned effectiveness outcomes in OSS teams. This discussion is used as the basis for developing hypotheses regarding the effects of a teams' adherence to these particular ideological tenets on team members' motivations and behaviors relevant to the effectiveness of the team. The methods section describes how the model was tested by combining survey and public data on OSS projects. The discussion section summarizes the results and outlines implications, limitations, and contributions of the research.

OSS IDEOLOGY AND OSS TEAM EFFECTIVENESS

"there are norms common to nearly all mature, sustained open source projects" (Bretthauer 2002: 3)

> "[open source movements are] loosely coupled communities kept together by strong common values" (Ljungberg 2000: 208)
>
> "contributors…are motivated by the personal benefit of using an improved software product and by social values such as altruism, reputation, and ideology" (Markus et al. 2000: 14)

As the quotations demonstrate, the ideology associated with OSS development has been widely suggested as a lynchpin in enabling OSS efforts. We employ Trice & Beyer's (1993: 33) definition of ideology as "shared, relatively coherently interrelated sets of emotionally charged beliefs, values, and norms that bind some people together and help them make sense of their worlds." *Beliefs* refer to understandings of causal relationships, *values* refer to preferences for some behaviors or outcomes over others, and *norms* refer to behavioral expectations (Trice and Beyer 1993).

Though ideology is by definition shared, adherence to a particular ideological tenet may vary across subgroups. Ideology is an aspect of culture (Trice and Beyer 1993). Just as the ideology of an organizational culture is influenced by the larger national culture in which it exists (Hofstede 1980; Hofstede et al. 1990), organizational subunits may have their own unique subcultures, varying in their ideological content. Open source developers have often been said to constitute a community, identifiable by its common ideology (Ljungberg 2000). At the same time, subgroups are formed around projects within that community, and such subgroups may vary in the extent to which they conform to the overarching community ideology. Ljungberg (2000: 210) suggests that commitment to the ideology varies widely across developers: "At one end of the spectrum there is great zeal…at the other end there is no big deal about the ideology."

In settings such as OSS development where behavior may be hard to observe and it is not feasible to institute formal controls, ideology may be important as the vehicle of clan control (Barker 1993). Rather than dictating specific actions or manipulating tangible rewards, clan

control occurs when shared understandings define appropriate ways to behave and respond to situations in the group (Kirsch 1997; Ouchi 1979). In order for it to result in positive team outcomes, ideology should motivate behaviors and responses that are likely to be productive for the team, constrain opportunistic behavior, and apply in a broad variety of situations to orient team members. Below we first identify norms, beliefs, and values of the larger OSS community and then discuss how these norms, beliefs and values influence developers' motivations to produce behavior that enhances the effectiveness of an OSS project team.

Content of the OSS Ideology

We took a two-pronged approach to understanding the content of the OSS community ideology by deriving norms, beliefs, and values from both narratives of the open-source community including commentaries by icons such as Richard Stallman[4] and Eric Raymond[5] as well as prior research by academics and practitioners (Bretthauer 2002; Elliott and Scacchi 2003; Feller and Fitzgerald 2002; Gosain 2003; Lerner and Tirole 2002; Ljungberg 2000; Markus et al. 2000). The narratives have been shaped by community members' input over time, making them representative of the overarching OSS ideology[6]. The identification of the tenets of OSS ideology draws upon socio-political ideological discourse analysis, which proposes that ideologies are both developed and expressed in language and communication and may be "uncovered" by close reading (van Dijk 1995: 135). Table 1 summarizes the norms, beliefs, and values uncovered in the review and provides a quote to illustrate each one. Identification of norms, beliefs, and values was limited to those that are widely claimed to be held in the community, not including those that are specific only to one or a small subset of projects. Similar

[4] Richard Stallman is the founder of the Gnu Project, launched in 1984 to develop the free operating system GNU. He is also founder of the Free Software Foundation and creator of the copyleft concept in software licensing. The Free Software adherents are considered especially ideologically "zealous."

[5] Eric Raymond has been an influential member of many OSS projects, and he is the author of highly influential papers including "The Cathedral and the Bazaar," which influenced Netscape's decision to get involved in OSS, and "The Halloween Documents," which presents a critique of Microsoft's response to OSS.

[6] See, for example, http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s16.html

to other settings, these ideological tenets provide a "general cultural repertoire" from which teams may select those that best realize their goals and interests and use them as building blocks for team ideology (van Dijk 1995: 138). Table 1 is thus proposed as a preliminary framework, not one that necessarily encompasses all norms, beliefs, and values that may be relevant in any particular OSS project.

Table 1. Tenets of Open Source Ideology

OSS Norms	**Illustrative Quotes from Open Source Narratives**
1. Forking – there is a norm against forking a project, which refers to splitting the project into two or more projects developed separately.	…`forking' almost never happens. Splits in major projects have been rare, and always accompanied by re-labeling and a large volume of public self-justification. It is clear that, in such cases as the GNU Emacs/XEmacs split, or the gcc/egcs split, or the various fissionings of the BSD splinter groups, that the splitters felt they were going against a fairly powerful community norm (Raymond 1998)
2. Distribution - there is a norm against distributing code changes without going through the proper channels.	The taboos of a culture throw its norms into sharp relief. Distributing changes to a project without the cooperation of the moderators is frowned upon, except in special cases like essentially trivial porting fixes. (Raymond 1998)
3. Named Credit – There is a norm against removing someone's name from a project without that person's consent.	Removing a person's name from a project history, credits or maintainer list is absolutely *not done* without the person's explicit consent. (Raymond 1998)
OSS Beliefs	
4. Code Quality – open source development methods produce better code than closed source	Perhaps in the end the open-source culture will triumph not because cooperation is morally right or software 'hoarding' is morally wrong [...] but simply because the closed-source world cannot win an evolutionary arms race with open-source communities that can put orders of magnitude more skilled time into a problem. (Raymond 2000)
5. Software Freedom – outcomes are better when code is freely available	Restrictions on the distribution and modification of the program cannot facilitate its use. They can only interfere. So the effect can only be negative. (Stallman 1992)
6. Information Freedom – outcomes are better when information is freely available	I believe that all generally useful information should be free. By 'free' I am not referring to price, but rather to the freedom to copy the information and to adapt it to one's own uses.When information is generally useful, redistributing it makes humanity wealthier no matter who is distributing and no matter who is receiving. (Stallman quoted in (Denning 1990))
7. Bug fixing – The more people working on the code, the more quickly bugs will be found and fixed.	Peer review is essential. Given enough co-developers, problems will be characterized quickly and the fix obvious to someone: "Given enough eyeballs, all bugs are shallow". (Raymond 2000)
8. Practicality – practical work is more useful than theoretical discussion	[I]n reading things from Linus in the past, he does seem to value working code much more than complaints or rants or high-level descriptions. Show me the code, I think he said. Find a way to make it work and make it work. (Kuwabara 1999)

9. Status Attainment – Status is achieved through community recognition	When you play the hacker[1] game, you learn to keep score primarily by what other hackers think of your skill (this is why you aren't really a hacker until other hackers consistently call you one). (Raymond 2003)
OSS Values	
10. Sharing – sharing information is important.	Programmers have the duty to encourage others to share, redistribute, study, and improve the software we write. (Stallman 1992)
11. Helping – aiding others is important.	Hackers solve problems and build things, and they believe in freedom and voluntary mutual help. To behave like a hacker, you have to believe that the thinking time of other hackers is precious — so much so that it's almost a moral duty for you to share information, solve problems and then give the solutions away just so other hackers can solve new problems instead of having to perpetually re-address old ones. (Raymond 2003)
12. Technical knowledge - Technical knowledge is highly valued.	But be aware that you won't reach the skill level of a hacker or even merely a programmer if you only know one or two languages — you need to learn how to think about programming problems in a general way, independent of any one language. To be a real hacker, you need to get to the point where you can learn a new language in days by relating what's in the manual to what you already know. This means you should learn several very different languages. (Raymond 2003)
13. Learning – there is a value on learning for its own sake.	I am a hacker. That is to say, I enjoy playing with computers -- working with, learning about, and writing clever computer programs. (Raymond 2000)
14. Cooperation – voluntary cooperation is important.	We must start sending the message that a good citizen is one who cooperates when appropriate, not one who is successful at taking from others. I hope that the free software movement will contribute to this: at least in one area, we will replace the jungle with a more efficient system which encourages and runs on voluntary cooperation. (Stallman 1992)
15. Reputation - reputation gained by participating in open source projects is valuable.	For examined in this way, it is quite clear that the society of open-source hackers is in fact a gift culture. Within it, there is no serious shortage of the `survival necessities'—disk space, network bandwidth, computing power. Software is freely shared. This abundance creates a situation in which the only available measure of competitive success is reputation among one's peers. (Raymond 1998)

Effectiveness in OSS Development Teams

As with the study of commercial software development contexts, it is important in the open source context to consider a multidimensional view of team effectiveness (Crowston et al. 2003). We consider two aspects of OSS project effectiveness: the extent to which a project attracts input from the development community and the extent to which it produces observable

[1] While the term "hacker" has a pejorative connotation in popular usage, it conveys a positive identity in the OSS community.

outputs such as the addition of new features to the software or the fixing of software bugs. While commercial projects have employees paid and directed though formalized mechanisms, a critical step in becoming effective in an OSS project is to attract developers and motivate their input to the project (Mockus et al. 2002; Sturmer 2005; von Krogh et al. 2003). Without people donating their efforts voluntarily, an OSS project has little chance of success, thus the amount of input to a project (i.e., how many people devote how much effort) is an important aspect of effectiveness. The number of developers that have been attracted and retained to work on the team (*team size*) and an estimate of the amount of *effort* those developers have devoted to the team are the two constructs related to an OSS team's input effectiveness used in this study.

Given that meeting budgets and requirements may not be relevant in OSS (Scacchi 2002), more appropriate indicators of outcome success may be those related to the ongoing productivity of the team. Software development teams rely on change management systems to organize and track their development work, with actual changes to code being associated with a basic unit of work called a modification request. Studies of software development in distributed contexts (Herbsleb and Mockus 2003) and analyses of OSS projects such as Apache and Mozilla (Mockus et al. 2002) have used responses to modification requests as indicators of work accomplishment. The extent to which the team completes identified work tasks (*task completion*) is the output effectiveness construct used in this research.

The next section first develops hypotheses regarding antecedents to the input effectiveness of an OSS team and then focuses on hypotheses regarding the output effectiveness of the team. The theoretical development focuses on the role of shared OSS norms, beliefs, and values in enhancing developers' identification with the team and their motivation to behave in ways that are beneficial to OSS team effectiveness outcomes. The essential argument is that in teams that adhere to the OSS ideological tenets identified in table 1, individuals develop a

stronger sense of identification with the team and their behavior is more predictable and helpful. This creates a connection between adherence to the ideological tenets, the level of trust, and the quality of communication within the team. Trust and communication in turn enhance team effectiveness. The proposed relationships are depicted fully in figure 1.

THEORY DEVELOPMENT AND HYPOTHESES

There are many reasons that potential developers may be attracted to an OSS project, join and remain involved in an OSS project, and voluntarily devote their efforts to working on the project. These include their assessments of the likely usefulness of the project software, potential reputational benefits from project involvement, and the extent to which involvement in the project brings psychological or emotional benefits (Hars and Ou 2002). All of these factors may be affected by the level of trust that exists in the OSS team.

Trust as an Antecedent to OSS Team Input Effectiveness

Trust has been viewed as "the extent to which a person is confident in, and willing to act on the basis of, the words, actions, and decisions of another" (McAllister 1995: 25), and this definition has been generalized to the level of a team (Jarvenpaa et al. 1998). This view of trust is appropriate in the setting of OSS teams because the value of one member's contributions depends in part on the efforts and contributions of others (e.g., insofar as code contributed by different members must function together). Also, the overall value of the software to a developer is contingent on the contributions of others and the decisions made regarding how those contributions are combined. Similarly, the words, actions, and decisions of others may affect the emotional and psychological consequences that an individual experiences from participation. Prior work on virtual teams has shown that low levels of trust may be associated with decreases

in project participation and contribution among team members (Jarvenpaa and Leidner 1999), and that OSS projects may have high levels of turnover (von Krogh et al. 2003).

McAllister (1995) argues that affective and cognitive components of trust are important. *Affective trust* stems from emotional attachment between a trustor and a trust target and may therefore be most relevant to potential developers' psychological and emotional reasons for joining, staying with, and contributing to OSS teams. Affective trust is persistent, leading to the discounting of events that are contrary to ongoing harmony (Holmes and Rempel 1989). For example, the following extract from developer archives of the Jakarta POI project demonstrates how an experienced member responded positively (with an offer of mentorship and encouragement to increase efforts) to a negative behavior on the part of an inexperienced member (missing a deadline).

[name removed], So did you meet your deadline? I hope you are still alive to read this :-) [……] I can help you meet your next deadline if you give 100% of your work time to HWPF. I don't know if this is possible because I don't know what else you're working on. I can assign you enough work to keep you busy all day, every day. I need your total commitment. Did you ever see the Karate Kid? I will be like Mr. Miagi and you will be like Daniel-son.

Interactions such as these may not only benefit the team by retaining and motivating existing members, but, because this communication can be observed by potential members who have not yet joined the team, the positive affective climate demonstrated may actually help to attract new developers. Potential new members can see by reading such communications that current team members provide a supportive atmosphere, and that new members may receive opportunities for mentoring. Von Krogh et al. (2003) show that individuals observe and participate in public project communication forums prior to becoming members of an OSS project development team, supporting the contention that the climate in the team may be an antecedent to the growth of team membership.

Hypothesis 1: Affective trust among OSS team members will have a positive effect on the team's input effectiveness by increasing (a) team size and (b) team effort.

Cognitive trust relies on a rational assessment of the target by the trustor (McAllister 1995), and such assessments may relate to the evaluation of competence of the trust target as well as the probability of opportunistic behavior by the target. Thus cognitive trust may be most important to the utility and reputation motivations for developers to join, stay with, and exert efforts on OSS projects. These motivations may be affected because the usefulness of an OSS product to one developer depends in part on the inputs of other developers. If a contributor has high cognitive trust in other team members then he will expect those inputs to be useful and the project to sustain ongoing success, thereby making it more worthwhile to devote his efforts to the project. A contributor will not want to waste effort on a project that he thinks is going to fail because such a project is not likely to be useful to him in the future nor to enhance his reputation. Because of the public nature of OSS team interactions and outputs, cognitive trustworthiness may be assessed by both current members and potential members. In addition to increasing the estimate of the utility likely to result from contributing to an OSS team, cognitive trust also reduces the perceived need for and thus the expenditure of effort on monitoring and defensive behaviors (McAllister 1995). Reducing behaviors that are not directly productive should free more resources to be devoted to the team's tasks.

Hypothesis 2: Cognitive trust among OSS team members will have a positive effect on the team's input effectiveness by increasing (a) team size and (b) team effort.

Impacts of OSS Norms, Beliefs, and Values on Trust in OSS Teams

Cognitive trust relies on rational assessments of competence and behavior and may develop through the prediction process when a trustor is able to forecast that a target will behave

in a desirable fashion (Doney and Cannon 1997). OSS norms, beliefs, and values support the predictability of behavior that will be beneficial to team members and the assessment of others as competent. Shared norms in general tend to result in behavior patterns that are relatively stable and expected by a team's members (Bettenhausen and Murnighan 1991). In the case of OSS, norms are designed to protect developers' interests by assuring they get credit for their work (Named Credit norm) and maximizing the value of that work (Forking norms). The forking norm minimizes the hazard of competing versions of projects, which may reduce reputational benefits to contributors by decreasing the audience for each fork (Kogut and Metiu 2001).

Shared beliefs provide a kind of mutual knowledge about cause-effect relationships (Hardin and Higgins 1996), which enables rational assessment of other team members' behaviors based on an understanding of the reasoning driving these behaviors. The OSS beliefs about bug fixing, practicality, and status attainment are shared beliefs about how and why better outcomes are obtained from open source community practices. These beliefs guide the specific means by which OSS work is conducted, and in some cases may provide an explanation for behaviors enacted by team members that might otherwise be interpreted negatively. For example, when a developer submits code with defects for peer review, instead of interpreting this as an indication that the developer is not competent to fix the bugs, it may be interpreted in light of the common understanding that allowing multiple others to help fix bugs is more efficient. Sharing beliefs about OSS processes and their associated outcomes is, therefore, likely to result in developers having greater cognitive trust in the ability of others.

OSS values may also enhance cognitive trust by leading members to perform behaviors that are evaluated by others as indicative of competence and ability. When team members adhere to the values of attaining technical knowledge, learning, and enhancing their reputations, they will act in ways that will increase their knowledge and competence. When they also adhere to

the values of helping, sharing, and cooperating they will provide to each other evidence of that knowledge and competence. Adhering to these values will also allow contributors to predict how others will respond to their requests. Because predictability and assessment of the ability of others are at the root of cognitive trust, subscribing to these OSS values should enhance cognitive trust in the team.

Hypothesis 3: An OSS teams' member's adherence to the norms of the OSS community will have a positive effect on cognitive trust among the team's members.

Hypothesis 4: An OSS team's members' adherence to the beliefs of the OSS community will have a positive effect on cognitive trust among the team's members.

Hypothesis 5: An OSS team's members' adherence to the values of the OSS community will have a positive effect on cognitive trust among the team's members.

While cognitive trust relies on rational assessments of others, affective trust rests on emotional attachments to others. Such emotional attachments may be created and enhanced via group identification (Bagozzi and Dholakia 2002; Tajfel 1978), and group norms and values have been highlighted as important to forming positive group identification in virtual settings (Dholakia et al. 2004). Dholakia et al. (2004) argue that norms are important to identification in virtual settings because they tend to be most accessible or inferable based on the group's archived communication and understanding them will facilitate potential members' categorization of themselves as part of the group. Postmes et al. (2000) point out that in order to have a positive effect, the group's norms must be congruent with individual's motivations. As explained above, the OSS norms are consistent with reputation and utility-based motivations of OSS developers. Thus adherence to the OSS norms may be expected to enhance identification with the group and thereby the emotional attachment that underlies affective trust.

Values may drive identification with virtual groups because when a member or potential member finds that her values are congruent with those of others in the group, she is more likely to categorize herself as a member of the group (Bagozzi and Dholakia 2002). Dholakia et al. (2004) found that values related to individual instrumental purposes such as giving and receiving information in a virtual group had a positive effect on identification with the group. In OSS, the values related to technical knowledge, learning, and reputation all focus attention on common personal instrumental goals of contributors.

Hypothesis 6: An OSS team's members' adherence to the norms of the OSS community will have a positive effect on affective trust among the team's members.

Hypothesis 7: An OSS team's members' adherence to the values of the OSS community will have a positive effect on affective trust among the team's members.

Antecedents to OSS Team Output Effectiveness

Input-process-outcome models of team work (Ilgen et al. 2005; Martins et al. 2004) imply that OSS team output effectiveness is likely to be influenced by the effectiveness of the team in attracting and retaining input. Prior research has indicated that OSS developers may tend to specialize by working on small parts of an overall project (Koch and Schneider 2000; von Krogh et al. 2003). In this case, having a larger pool of developers implies a larger pool of available specializations and a greater chance that the skills and interest needed for completing a variety of tasks will be available. Likewise, the more effort that the team members devote to the project, the greater the number of tasks they will be able to complete.

Hypothesis 8: OSS project (a) team size and (b) team effort will have a positive effect on the team's level of task completion.

In addition to being a function of the input to a team, the extent to which an OSS team achieves high levels of task completion may be influenced by the quality of communication in the team. Globally distributed software development teams have been found to face significant communication related challenges (Montoya-Weiss et al. 2001), yet communication is essential for coordinating work among team members (Ancona and Caldwell 1992).

Communication in OSS Teams

Because software development is socially complex and communication-intensive, requiring precise communication for tasks such as requirements elicitation and project coordination, communication breakdowns are likely to occur (Carmel 1999). Poor communication can create task overlaps and duplication of effort, making code integration more difficult. The importance of timely and relevant communication has been recognized by some in the OSS community, as demonstrated in the following excerpt from the Mozilla web site (http://www.mozilla.org/get-involved.html):

Whenever you begin work on something that's more than a day's work, we strongly recommend that you let people know about it. Publicizing your efforts lets others know what you're working on so they can coordinate their work with yours, offer help, and avoid duplicating your work. It opens your work up for peer review, and provides an archive so the same issues don't need to be repeated. Also, if someone else is working on the same thing, they can let you know and save you a lot of work.

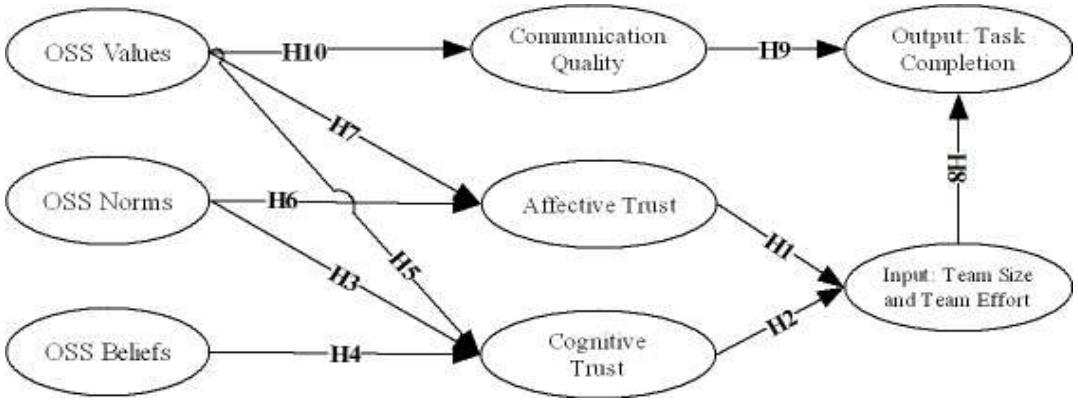
As this excerpt implies, it is not only the amount of communication, but its timeliness and helpfulness that are important to outcomes in OSS work. Previous work on communication in virtual teams also implies that it is not just the amount but the quality of communication that is important to outcomes. Communication is most likely to enhance team effectiveness if it is timely, helpful, and relevant to the task at hand (Jarvenpaa and Leidner 1999), which we refer to as being of high quality.

Hypothesis 9: Quality of communication among an OSS team's members will have a positive effect on the team's level of task completion.

While the importance of communication to effectiveness is well established, the means by which quality communication may be achieved in OSS teams is not. Though the review presented in table 1 did not uncover overarching OSS community norms or beliefs directly related to the quality of communication, there are several values that directly bear on communication practices in OSS development. The more participants value learning, building their technical knowledge, and enhancing their reputation, the more likely they should be to exchange useful information with each other. Shared OSS values will enhance quality of communication because the more team members value being helpful and cooperating, the more attentive they will be to providing timely and pertinent replies to each other's queries. Thus the values listed in table 1 should serve to focus team members on communicating useful information in a timely manner.

Hypothesis 10: An OSS teams' member's adherence to the values of the OSS community will have a positive effect on the quality of communication among the team's members.

Figure 1: Research Model



METHODS

Most studies of OSS to-date have focused on case studies of the largest, most well-known projects such as Linux and Apache (e.g., Gallivan 2001; Mockus et al. 2002). While these projects are interesting and important, they may not be representative of the majority of OSS projects, which have attained much lower levels of participation and prominence (Krishnamurthy 2002). One of the goals in this research was to collect data on a larger set of projects that may be more representative of the wider population of open source development efforts. To do this we collected data related to OSS projects hosted on Sourceforge (www.sourceforge.net). Sourceforge's mission is to enrich the Open Source community by providing a centralized infrastructure for developers to control and manage OSS development. As of April 2005, Sourceforge hosted more than 98,000 projects and more than one million registered users. In addition to hosting a large segment of the population of OSS projects, Sourceforge provides these projects with a standard technology toolset, thereby reducing variance in team effectiveness that may be due to differences in technology used to support workflow, code distribution, versioning, etc.

The unit of analysis is an OSS development team working on a specific software application. Several approaches have been suggested to measure team-level constructs – assessing individual perceptions as a representation of team beliefs (e.g., Crocker and Luhtanen 1990), aggregation of individual assessments (Guzzo et al. 1993), and team discussion to arrive at a common assessment (Gibson et al. 2000). Unfortunately, none of these approaches are ideal for assessing team level attributes (Bar-Tal 2000; Gibson et al. 2000). Given the practical

difficulty of accessing all members of OSS teams, we used a key informant approach (Seidler 1974) combined with collection of publicly available project data from the Sourceforge website.

Using key informants to collect data about larger social entities is a common practice in organizational research (see Nelson and Cooprider (1996) at the department level, Capron and Hulland (1999) at the firm level, and Sparrowe et al. (2001) at the team level). The use of key informants requires a deliberate strategy to access respondents that possess special qualifications pertinent to the research such as status, experience, or specialized knowledge (Segars 1998). Research on the effectiveness of software development projects typically elicits project-level data from project managers (e.g., Ethiraj et al. 2005; Gopal et al. 2003). In the F/OSS domain project administrators represent an equivalent role that is best suited to providing details of a project. Thus we targeted project administrators, as they are positioned to be most familiar with the teams' internal dynamics, activities, and accomplishments. Also, because founders and key leaders shape organizational ideology (Hofstede et al. 1990), such individuals may be better suited to report on the ideology of their teams than other individual members.

Data Collection

Data was collected using two surveys and from the Sourceforge website. The first survey included open-ended questions asking project administrators about their projects, teams, and development efforts. Administrators of 48 Sourceforge projects were contacted using email. Eighteen administrators completed the questionnaire. The qualitative data collected on this survey was then used to develop the wording of closed-ended ideology items to be included on a second survey, which also measured trust and communication for hypothesis testing.

A different set of projects was targeted for the second survey. We selected projects from two categories on Sourceforge – communications (BBS, chat, and ICQ) and multimedia (audio, video, and graphics 3D rendering). We limited the sample to two similar domains to control for

differences across projects in very different product categories. After selecting categories, we ensured that the projects had some activity in the past week in terms of contributions to the code repository; requests for bug-fixes, support, patches or features; or in terms of page views. This was done to ensure that the sample included ongoing projects that had not been abandoned by their developers. Finally, projects were required to have at least four developers to be included in the study because the model is concerned with team processes and dynamics. One hundred fifty projects met all criteria. A subset was randomly selected to pilot test the survey. For the pilot test, respondents were asked to complete all Likert scale items and answer open ended questions asking if any of the items were unclear, if they had problems understanding or answering any questions, or if there were ways the survey itself could be improved. Sixteen administrators responded, and none of them indicated any problems in the survey. Personalized invitations were sent to the remaining project administrators in the sample requesting their participation.

Participants were offered a chance to win a $100 lottery as well as an opportunity to be informed of the results. Because many individuals are involved in multiple open source projects, the survey instructed administrators involved with multiple projects to respond with reference to the software project with which they were most involved – this resulted in seven respondents reporting on projects not in the communications or multimedia categories. These seven projects focused on providing software to support other functional applications, therefore they were categorized as utility applications. Fifty-one administrators responded to the survey. After ANOVAs confirmed that the pilot test respondents did not differ significantly on any of the outcome measures from the respondents in the second round, the two sets of responses were merged for analysis, resulting in a sample size of 67 (an overall response rate of 44.7%).

Operationalization of Constructs

Ideology, Trust, and Communication

We followed Hofstede et al. (1990) by first elaborating specific values, norms, and beliefs (listed in table 1) and then crafting Likert scale items to quantitatively measure the extent to which individuals felt each ideological tenet was adhered to in his or her team. In order to maintain a manageable survey length, and because there are no prior empirical investigations of OSS ideology to draw on, one item was created to represent each norm, belief, and value. Exploratory factor analysis was used to assess the extent to which these items loaded onto underlying constructs representing adherence to sets of norms, beliefs, and values.

Scales from earlier studies were adapted to measure cognitive trust and affective trust (McAllister 1995). The scales were reworded to reflect the change from the original inter-personal level to a team-level assessment (Ammeter 2000). We drew upon the work of Jarvenpaa & Leidner (1999) to create Likert scale items measuring the administrator's perception of the quality of communication in the team. All scale items are included in the appendix.

Team Effectiveness

Team size was measured as the number of developers associated with the project. *Team effort* was measured as an estimate by the project administrator of the total number of work weeks devoted to the project. The Sourceforge site tracks the number of requests for bug fixes, patches, support, and new features on each project as well as the number of such requests that have an uncompleted (open) status. We calculated *task completion* as the percentage of tasks completed: (total requests – requests open)/total requests * 100, or zero for projects with no task requests. This operationalization is in line with other studies of software teams that use change requests as a measure of software work accomplishment (Herbsleb and Mockus 2003). Because

it includes requests for features in addition to requests for changes to existing code, the task completion measure is relevant even at relatively early stages of project development.

Control Variables

All of these measures are likely to be a function of how long the project has been in existence, therefore *project age* (number of months since project inception as reported by the administrator) was used as a control variable. Both input and output measures of effectiveness may be a function of the development stage of the project because, for example, projects in earlier stages may be less certain to provide utility, reducing motivations for input and projects in the later stages may have defined roles to more efficiently handle task requests. Therefore the *development stage* of the project (reported on Sourceforge) was used as a control variable in the model. Development stage ranges from 1 for projects in the planning stage to 6 for projects that have reached a mature state. McAllister (1995) argued and demonstrated that cognitive trust is an antecedent to affective trust, therefore a path between cognitive trust and affective trust was included in the model. Team effort may be partially a function of the number of team members, therefore a path was included to control for the effect of team size on team effort.

Analysis

Data were analyzed in two stages. Because survey items were developed or adapted for the study, they were first subjected to an exploratory factor analysis. This allowed assessment of the extent to which the OSS values, norms, and beliefs may represent deeper underlying constructs or may be effectively independent of one another. Past studies have suggested the use of factor-analytic procedures to gauge the best representation of the underlying correlational structure of measures of ideological viewpoints (cf. Tetlock 2000). In the second stage of the analysis measurement properties were further assessed and hypotheses were tested using the Partial Least Squares (PLS) approach. PLS was used because it is more appropriate than

alternatives, such as LISREL, when sample sizes are small and models are complex, the goal of the research is explaining variance, and measures are not well established (Fornell and Bookstein 1982; Gefen et al. 2000).

RESULTS

Descriptive statistics and correlations among constructs are shown in table 2. Because data was obtained on projects in three different categories (35 multimedia, 25 communications, and 7 utilities) we conducted ANOVAs to determine whether the projects were significantly different on variables of interest across categories. Analysis of variance tests did not show a significant difference in mean values of any of the dependent variables across the categories, thus the three sets of projects were pooled for analysis ($p = .065$ for team size, $p = .516$ for team effort, $p = .885$ for task completion).

Factor Analyses

Harman's one factor test provided no evidence of a common method problem (the largest factor accounted for less than 25% of the overall variance). Given the constraints of the sample size relative to the number of items measured, we followed other recent work (cf. Kirsch et al. 2002) and conducted factor analyses on subsets of items to assess convergent and divergent validity. Items were grouped into analyses based on their expected loading (i.e., keeping all items for a single scale and for closely related constructs together) and following guidelines regarding an acceptable observation-to-item ratio of approximately 5 to 1 (Stevens 1996).

Two analyses were run, one including survey items related to affective trust, cognitive trust, and communication quality, a second including survey items related to norms, beliefs, and values. Principal components analysis with varimax rotation was used. Stevens (1996) recommends using items with factor loadings above 0.40 for interpretation purposes. Following

this guideline, three items aimed at assessing cognitive trust, one item aimed at assessing affective trust, and one item aimed at the communication quality scale were dropped. After items were dropped, the remaining affective trust, cognitive trust, and communication quality items all loaded highly on the expected factor with acceptable cross-loadings (see table 2).

Table 2. Descriptive Statistics and Correlations among Constructs[1]

Construct (# items)	Mean	Std Dev	Reliability	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
1.Communication Quality (3)	5.84	0.94	.90	0.86													
2. Forking Norm (1)	4.48	1.64	-	-0.01	1.00												
3. Distribution Norm (1)	4.85	1.60	-	-0.04	0.00	1.00											
4. Named credit Norm (1)	2.22	1.74	-	0.02	0.00	0.02	1.00										
5. Process Beliefs (3)	4.95	1.17	.69	0.19	-0.11	-0.04	0.03	0.69									
6. Freedom beliefs (2)	5.49	1.20	.79	-0.07	0.04	-0.12	-0.04	0.44	0.82								
7. Collaborative Values (3)	6.18	0.73	.85	0.43	-0.12	-0.02	-0.20	0.35	0.11	0.81							
8. Individual Values (3)	5.98	0.76	.76	0.28	-0.21	0.02	0.06	0.23	-0.03	0.45	0.72						
9. Affective trust (4)	5.18	1.07	.88	0.32	0.19	0.19	-0.23	0.21	0.11	0.53	0.35	0.77					
10. Cognitive trust (3)	5.47	1.07	.89	0.48	0.02	0.11	-0.27	0.37	-0.07	0.58	0.22	0.50	0.85				
11. Team Size (1)	8.25	14.74	-	0.14	0.12	0.00	-0.07	0.10	-0.13	0.01	0.03	0.17	0.25	1.00			
12. Team Effort (1)	197.94	306.95	-	0.19	0.34	0.12	0.02	-0.27	-0.27	-0.11	-0.13	0.26	0.07	0.22	1.00		
13. Task Completion (1)	42.33	34.55	-	0.33	-0.20	0.00	0.10	0.16	0.07	-0.05	0.16	0.06	0.10	0.08	0.20	1.00	
14. Development Stage (1)	4.02	1.31	-	0.15	0.07	0.19	-0.06	0.00	-0.03	-0.04	0.12	0.07	0.14	0.00	0.33	0.24	1.00
15. Project Age (1)	25.97	23.91	-	0.19	0.27	0.06	-0.03	-0.05	0.01	-0.05	-0.13	0.16	0.00	-0.08	0.62	0.09	0.36

[1]*Values on the diagonal are the square-root of the average variance extracted for each construct (AVE)*

Variables 1-10 were measured using 7 point Likert scale items. Team size is the number of developers, team effort was an estimate of work weeks devoted to the project, task completion is a percentage, development stage is on a 1-6 scale, and project age is in months.

The factor analysis of the ideology items was exploratory in that there was no prior work to suggest an a priori grouping of the items. Instead, the analysis was used to determine whether they should be treated as representing a single coherent set or multiple constructs. The analysis including all ideology items indicated that beliefs and values tended to load together onto different factors, but no two of the items representing norms loaded on the same factor. Each norm loaded highly on a different factor seemingly indicating that they may represent relatively independent rules for behavior. This would be in keeping with van Dijk's (1995) assertion that groups may, to some extent, pick and choose tenets from the larger organizational ideology in which they function. Norms were therefore treated separately in subsequent analyses.

Having removed norms, the factor analysis was re-run to examine beliefs and values. The item representing the value of practice over theory did not load highly on any factor and was dropped. Remaining items loaded clearly onto 4 factors, two representing values and two representing beliefs. For values, the three items reflecting cooperation, sharing, and helping loaded on one factor and the three items focusing on learning, expanding technical knowledge, and enhancing reputation loaded on a second factor. Because the first set of values all focus on interaction among team members, we labeled this factor *collaborative values*. Because the second set focus on individual goals associated with participating in OSS, we labeled these *individual values*. The split of the values into these two factors is consistent with the distinction between collectivistic and individualistic dimensions that has been recognized in past work on employee values in organizations (cf. Moorman and Blakely 1995). Beliefs also loaded onto two factors. These were beliefs about better quality outcomes resulting from the processes used in OSS development (code quality, bug fixing, status attainment) and beliefs regarding the benefits

of freedom to outcomes in OSS development (software freedom, information freedom). These are labeled *process beliefs* and *freedom beliefs* respectively.

In the final analyses (see table 3), all items loaded most highly on their respective factors and loaded less highly on other factors. In two cases where loadings were above 0.4 on alternate factors, the items loaded at least 0.6 on the expected factor and at least 0.2 higher on the main versus the cross-loaded factor. Given that the cross-loadings in the factor analysis results are in line with prior work (e.g., Agarwal and Karahanna 2000), that the items did clearly load most highly on a single factor, and that deleting them would reduce the reliability of the scales, we retained these items. The wording of all items, including those deleted, is shown in the appendix.

Table 3. Factor Analysis Loadings

Item	1	2	3
Affective Trust 1	**.837**	-.018	.275
Affective Trust 2	**.724**	.258	-.157
Affective Trust 3	**.689**	.461	-.056
Affective Trust 4	**.684**	.175	.388
Cognitive Trust 1	.266	**.714**	.331
Cognitive Trust 2	.133	**.826**	.290
Cognitive Trust 3	.187	**.769**	.143
Communication Quality 1	.282	.052	**.847**
Communication Quality 2	.069	.334	**.790**
Communication Quality 3	-.093	.340	**.787**

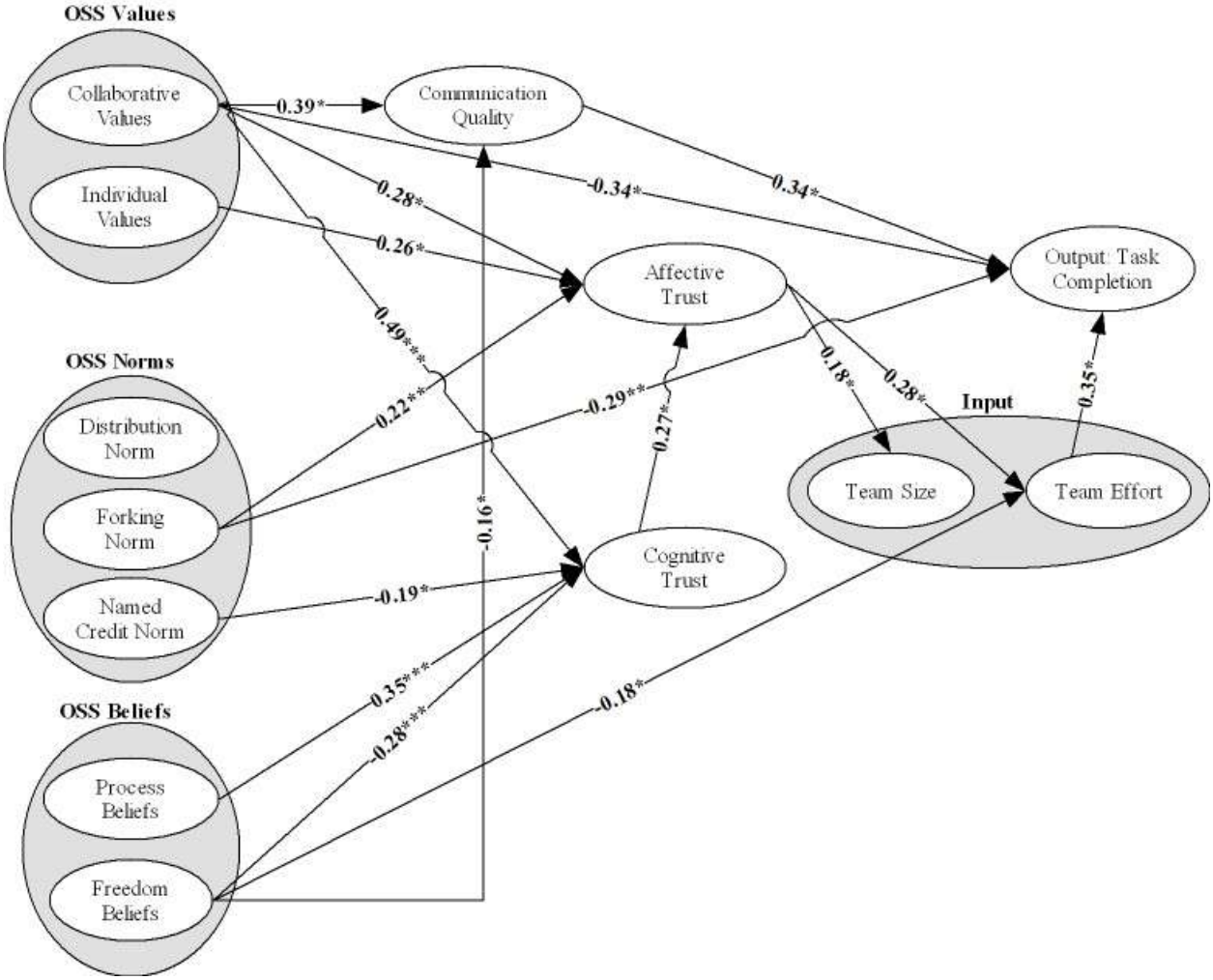
Item	1	2	3	4
Code Quality	**.609**	.107	.003	.308
Bug Fixing	**.698**	.055	-.002	.234
Status Attainment	**.767**	.014	.216	-.357
Software Freedom	-.001	**.868**	.161	-.026
Information Freedom	.467	**.707**	-.137	.125
Technical Knowledge	.209	-.389	**.446**	.365
Learning	-.073	.045	**.808**	.272
Reputation	.126	.063	**.859**	.000
Cooperation	.000	-.095	.398	**.727**
Sharing	.350	.096	.006	**.670**
Helping	.042	.006	.116	**.854**

PLS Measurement Model

While the factor analysis results provide evidence of convergent and divergent validity, in the PLS methodology (cf. Agarwal and Karahanna 2000) divergent validity is typically assessed using the average variance extracted (i.e., the average variance shared between a construct and its measures, AVE). The squareroot of this measure should be greater than the variance shared between the construct and other constructs in the model (Fornell 1981). Table 2 shows that this is the case. The table also displays the composite reliability, which is calculated as explained in (Agarwal and Karahanna 2000). Though reliability for process beliefs was slightly low at 0.69, all other scales were well above the typical threshold of 0.7.

The results of the PLS analysis are shown in figure 2 and table 4. The path coefficients in the model were assessed using the jackknife routine. To allow for the possibility of effects other than those hypothesized, we tested a saturated model, including paths from all independent variables to each of the measures of effectiveness. To present an uncluttered picture, control variables and insignificant paths are omitted from the figure. The coefficients for all variables and paths are displayed in table 4. Table 5 summarizes the results of all hypothesis tests.

Figure 2: Structural Model



****p< 0.001 , ** p< 0.01, * p< 0.05*
Note: Named Credit is measured using a reverse coded item

Table 4. Structural Model

	Path Coefficients
Communication Quality (R^2=0.28**)**	
Forking Norm	0.02
Distribution Norm	-0.06
Named Credit Norm (r)	0.09
Process Beliefs	0.11
Freedom Beliefs	-0.16*
Collaborative Values	0.39*
Individual Values	0.10
Project age	0.23**
Affective Trust (R^2=0.49)	
Forking Norm	0.22**
Distribution Norm	0.17
Named Credit Norm (r)	-0.11
Process Beliefs	-0.07
Freedom Beliefs	0.14
Collaborative Values	0.28*
Individual Values	0.26*
Cognitive Trust	0.27*
Project Age	0.13*
Cognitive Trust (R^2=0.48)	
Forking Norm	0.12
Distribution Norm	0.11
Named Credit Norm (r)	-0.19*
Process Beliefs	0.35***
Freedom Beliefs	-0.28***
Collaborative Values	0.49***
Individual Values	-0.06
Project Age	-0.01

	Path Coefficients
Team Size (R^2=0.16)	
Communication Quality	0.09
Forking Norm	0.10
Distribution Norm	-0.06
Named Credit Norm (r)	-0.05
Process Beliefs	0.14
Freedom Beliefs	-0.17
Collaborative Values	-0.28
Individual Values	-0.01
Cognitive Trust	0.21
Affective Trust	0.18*
Project Age	-0.15
Development stage	-0.02
Team Effort (R^2=0.62)	
Communication Quality	0.05
Forking Norm	0.08
Distribution Norm	-0.02
Named Credit Norm (r)	0.10
Process Beliefs	-0.19
Freedom Beliefs	-0.18**
Collaborative Values	-0.08
Individual Values	-0.11
Cognitive Trust	0.00
Affective trust	0.28**
Team Size	0.21
Project Age	0.49**
Development stage	0.13*

	Path Coefficients
Task Completion (R^2=0.33)	
Communication Quality	0.34**
Forking Norm	-0.29**
Distribution Norm	-0.04
Named Credit Norm (r)	0.03
Process Beliefs	0.15
Freedom Beliefs	0.17
Collaborative Values	-0.34*
Individual Values	0.13
Cognitive Trust	0.04
Affective Trust	0.01
Team Size	-0.03
Team Effort	0.35**
Project Age	-0.16
Development stage	0.13

****p< 0.001 , ** p< 0.01, * p< 0.05*

(r) reverse-coded item

Table 5. Results of Hypotheses Tests

Hypothesis	Result
H1a. Affective Trust → Team Size	Supported
H1b. Affective Trust → Team Effort	Supported
H2a. Cognitive Trust → Team Size	Not Supported
H2b. Cognitive Trust → Team Effort	Not Supported
H3. OSS Norms → Cognitive Trust	Supported for Named Credit Norm only
H4. OSS Beliefs → Cognitive Trust	Supported for Process Beliefs, Contradicted for Freedom Beliefs
H5. OSS Values → Cognitive Trust	Supported for Collaborative Values only
H6. OSS Norms → Affective Trust	Supported for Forking Norm only
H7. OSS Values → Affective Trust	Supported
H8a. Team Size → Task Completion	Not Supported
H8b. Team Effort → Task Completion	Supported
H9. Communication Quality → Task Completion	Supported
H10. OSS Values → Communication Quality	Supported for Collaborative Values only

DISCUSSION

Results supported the main thesis of the paper that adherence to the ideological tenets of the open source community are important to the effectiveness of OSS development teams by supporting trust and communication quality in the teams. In addition to identifying how tenets of the ideology were grouped into components and which components of the ideology had a positive effect, the analysis revealed some unanticipated findings regarding negative results from adherence to some ideological components. This section synthesizes the findings and theoretical implications with regard to the content and structure of the OSS ideology, the positive effects of components of the ideology, and the negative effects of ideological components. Practical implications are then considered. The paper concludes with a summary of limitations, contributions, and implications for future research.

Structure and Impact of the OSS Ideology

Table 1 identified several norms, beliefs, and values associated with the OSS community. Exploratory factor analysis indicated that these were grouped into a smaller number of

components. These were collaborative values (helping, sharing, cooperation), individual values (learning, technical knowledge, reputation), OSS process beliefs (bugs fixing, code quality, status attainment), and beliefs regarding the importance of freedom to outcomes in OSS (free information, free software). Results showed that norms (forking, named credit, and distribution) were adopted independently by OSS teams rather than forming into larger components.

With the exception of the distribution norm, each of the ideological components played a significant role in determining effectiveness in the OSS teams studied. For collaborative values, individual values, the forking norm, the named credit norm, and process beliefs, the overall pattern of results supported the arguments in the paper that these ideological components are important to effectiveness because of the positive impacts they have on trust and communication quality. However, two sets of negative effects were also revealed. First, freedom beliefs were shown to have only negative effects. Second, collaborative values and the forking norm had negative effects on task completion in addition to the positive effects they had on trust and communication quality. The next section discusses positive effects on OSS team effectiveness and the subsequent section focuses on interpreting the negative effects.

Drivers of OSS Team Effectiveness

One conclusion that can be drawn from the study is that affective trust was the main driver of the input effectiveness of OSS teams in both size and effort. There were two ways ideology supported affective trust. Collaborative values, individual values, and the forking norm had direct effects on affective trust, supporting arguments regarding the enhancing effect that these may have on identification with the team. One reason the forking norm impacted affective trust and the other norms did not may be that this norm effectively creates a barrier to exit from the team because it forbids starting a second team using the same code base. Groups that are less permeable (in this case, difficult to leave) have been shown to be perceived as stronger social

units, which enhances identification (Lickel et al. 2000). The other norms focus on behavior within the team rather than keeping the team together. Of these, the naming norm may be most directly relevant to cognitive trust because the naming norm is most directly tied to protecting individuals' interests, and assessments of trust tend to focus on expected individual outcomes from interaction. The distribution norm, by contrast, protects the interests of the project as a whole.

Results showed that cognitive trust was a significant antecedent to affective trust, providing the second path through which ideological components had a positive impact. The fact that cognitive trust acted through affective trust rather than having a direct effect on team size or effort may indicate that potential utility or reputation benefits alone are not enough to drive participation; developers may only devote their efforts to a project if affective feelings in the group are also positive.

As hypothesized, both team effort and communication quality drove the output effectiveness measure, task completion. The absence of an effect of team size on task completion failed to support the reasoning that having a greater number of developers available to complete tasks will result in greater output effectiveness on this measure. There may be at least two possible reasons that this relationship did not hold. First, it could be that teams with more members did not in fact have a wider range of expertise available but rather they included members with essentially overlapping areas of knowledge. Second, large teams may consume substantial resources to manage social interaction and enable coordination, and they may be prone to social loafing (Guzzo and Dickson 1996).

Detractors from OSS Team Effectiveness

In addition to the positive effects of collaborative values and the forking norm discussed above, these also had direct negative effects on task completion. Both could be a result of these ideological components imposing a "consensus-building cost." If team members believe that forking is acceptable (i.e. they go against the norm), they may be less likely to care about avoiding conflict or building consensus. As a result, in situations involving conflicting ideas, these teams may be less likely to wait to build consensus before closing a request. Instead, requests can be dealt with immediately with the option for disagreeing members to potentially fork the project. This explanation is suggested by comments from a manager for a Microsoft product that has been open-sourced, wherein he suggests that forking, despite its downsides, provides a "safety valve" for developers if conflicts arise (Mensching 2004).

Similarly, the more a team subscribes to collaborative values, the more they focus on attending to the needs of team members, which may detract from a focus on the task. Rather than closing task requests quickly, the team may spend time trying to build consensus on how requests should be handled, and they may leave requests open until agreement is reached. Our overall interpretation is that there may be a trade-off involved in fostering these ideological components: the positive affective trust built by these components helps to make the group membership valuable to developers and encourage their input, however task completion is slowed because of the increased desire to build consensus before tasks are closed.

While most ideological components had some positive effects, freedom beliefs had only negative effects. Higher levels of adherence to these beliefs negatively impacted communication quality, cognitive trust, and team effort. One explanation for this pattern of effects may lie in the fact that the strength of such beliefs has been seen as a differentiating factor among members of the broader community. Richard Stallman notes that developers who subscribe strongly to the freedom beliefs also believe that "*non-free software is a social problem and free software is the*

solution.[1]" These beliefs may therefore drive individuals to participate more broadly by spreading their effort across multiple projects, reducing their contribution to any one. That is, those who believe strongly in freedom as a crucial aspect of software development may be driven more by a desire to act as agents of social change than they are by the desire to further the objectives of a particular project. This could lead to reduced effort for the project. Because members' attention is spread across many projects the timing of their participation in any one may also be less predictable thereby reducing cognitive trust and communication quality.

In summary, there appear to be two ways that ideological adherence can detract from the effectiveness outcomes studied. Collaborative values and adherence to the forking norm may overemphasize the importance of acting only after the group is in agreement thereby solidifying dedication to the group but slowing progress in task completion. Adherence to the freedom beliefs may overemphasize participation in the larger OSS community rather than a particular OSS team project, leading members to spread their efforts over more projects.

Limitations and Future Research

The analyses revealed several unexpected results including negative effects of some ideological components. Above, we presented some possible explanations for these patterns of results, however the data does not allow us to test those explanations. Future research is needed to explore the mediating mechanisms between these ideological components and their outcomes.

Identification of Ideological Tenets

While table 1 represents a more comprehensive framework of the OSS ideology than has been developed elsewhere, it has some limitations. Because ideological tenets may sometimes

[1] http://www.gnu.org/philosophy/free-software-for-freedom.html

remain unexpressed (van Dijk 1995), there could be other important ideological tenets that were not revealed by the review of OSS discourse. More thorough research is needed to refine and validate this initial ideological framework. A further limitation is related to the measurement of adherence to ideological components. Single items to measure agreement with each norm, value, and belief were created. Exploratory factor analysis was used to uncover the interrelationships among the norms, values, and beliefs. Future work may use this as a basis for developing and validating more comprehensive scales to measure the ideological components identified. Conclusions regarding the structure of the OSS ideology based on this study must be regarded with caution until they can be confirmed or refined by future work.

Data Collection and Measurement

While the theoretical development argues for the causal nature of the relationships empirically examined, the research methodology does not allow us to establish temporal precedence. For example, trust could influence team members' perceptions of the extent to which they share common ideological views. However, norm formation has been reported to take place early (Bettenhausen and Murnighan 1985), while trust has generally been argued to develop over time (e.g., Doney and Cannon 1997). Ideology is seen to constitute the environment in which attitudes are formed (Scarbrough 1990). Further, since developers are drawn from the larger OSS community, they have likely been exposed to the OSS ideology before joining a team, thus reverse causality seems unlikely in this case.

The sample was limited to projects that had at least four team members because of the focus on team level constructs. This led to exclusion of projects that had very low levels of effectiveness in terms of attracting and retaining participants, and may thereby limit the generalizeability of results to such teams. Collecting measures of multiple constructs through the survey raises the possibility of common methods bias. Results showed sufficient discriminant

validity, and drawing measures of effectiveness from Sourceforge helps to limit this concern. In gathering the survey data project administrators were used as key informants because they were in a position to report on how the team functions as a whole and to have the most significant impact on the teams' dynamics (thus their perceptions may be of greater interest than the perceptions of other individual team members). While these respondents were able to tell us about the teams, perceptions may vary across team members, and administrators may tend to have an upwardly biased view of desirable qualities such as trust. Factors related to difficulties in observation or assessment may also distort the judgment of key informants (Bagozzi et al. 1991). Though it is not uncommon for research on software development to use project managers as key informants (Ethiraj et al. 2005; Gopal et al. 2003), we hope future studies will be able to elicit broader participation from team members.

The measure of task completion overcomes known problems with other indicators of software project performance such as lines of code (Von Hipple and Von Krogh 2003). However this and the other measures of team effectiveness did not take into account any specific evaluation of quality. For example, it is possible teams could report tasks as having been completed without their having been completed in an appropriate manner. Also, the timing of data collection could be important because task identification and completion may follow cycles related to new software releases. We hope that future efforts might seek more discriminating measures of OSS team effectiveness and incorporate more dynamic models to capture effects of the timing of key events in projects.

Other Antecedents to Effectiveness

This study sought to add to the literature on open source development by examining ideology and related social-psychological antecedents to effectiveness. While results indicated these are important factors, they do not provide the complete picture. Though some variance due

to other factors was controlled for (by limiting the sample), the model did not include structural, technical, socio-cognitive, or economic factors that past studies have implied may be important (Crowston and Kammerer 1998; Faraj and Sproull 2000; Lerner and Tirole 2002). For example, all of the sampled projects used the same technical toolset provided by Sourceforge. Projects that use other toolsets to conduct their work may face different challenges, and variance in outcomes due to differing technical platforms may be an interesting area for future work. Similarly, role specialization and differentiation in teams may be helpful in dealing with task complexity and could be particularly important to understand how these factors impact effectiveness because of the potentially dynamic nature of team membership (von Krogh et al. 2003). Because members may come and go at will, clear role specification could be especially important to developing swift trust among members (Meyerson et al. 1996). It may be interesting for future work to consider these sets of factors in concert and assess how they may interact with one another in determining OSS team outcomes.

Practical Implications

In addition to the research implications discussed above, results have several implications for organizations interested in utilizing OSS development practices or adopting OSS software and for the administrators of OSS projects. Scarbrough (1990) proposes that ideologies are analogous to maps specifying the terrain (e.g., belief systems), destinations (e.g., values), and signposts (e.g., norms) that orient actors. Identification of the main ideological components in OSS teams and the different means by which they affect outcomes provides a preliminary map for organizations wishing to leverage or understand OSS software development. This may be valuable to organizations because those interacting with the community are more likely to be successful if they are perceived as understanding and being sympathetic to the community's beliefs, values, and norms, and because it provides insight to guide specific decisions.

Many corporations have begun attempting to leverage open source development teams to produce software. Sun Microsystems, for example, set up the OpenOffice.org site to enhance Sun's Star Office suite. This study suggests that corporations wishing to attract the work of open source developers may benefit most by nurturing individual values, the named credit norm, and process beliefs as these factors were found to have positive effects leading to input measures without any simultaneous negative effects. While encouraging collaborative values and adherence to the forking norm may also result in increased input, and collaborative norms especially are important to fostering quality communication, organizations should be aware of the potential "trap" posed by coinciding negative effects on task completion. Organizations or project administrators may wish to investigate ways to mitigate these negative effects, perhaps by instituting policies to limit the amount of time spent building consensus before action is taken or by setting target deadlines for tasks to be handled.

Beyond providing guidance on what factors may enhance the effectiveness measures included in the study, the results point to the fact that organizations and project administrators may benefit from concentrating on multiple outcomes. Lots of attention had been paid to the size of OSS development teams and the importance of attracting developers to an OSS project (cf. Krishnamurthy 2002; von Krogh et al. 2003). However the study results suggest that large team sizes may not increase other outcomes of interest in that team size did not have a significant effect on either effort or task completion. Administrators should be aware that simply attracting developers may not guarantee the project will succeed in other ways.

The measure of task completion corresponds closely to practitioners' concerns about OSS in that responsiveness to support and change requirements (e.g., security patches) are two of the most frequently cited concerns of IT professionals considering using OSS (Smith 2002). By providing some understanding of the factors that influence the responsiveness of an OSS project

team to these concerns, the results of the study may better enable organizations to evaluate a particular OSS software product. For example, organizations may wish to assess the extent to which the team developing the software they are interested in adheres to particular components of OSS ideology linked to task completion because results indicate such teams may be more responsive to requests made by the organization.

Finally, it remains to be seen whether the implications for OSS projects may be applied to closed source development contexts. However, these results should encourage researchers and managers to explore whether more traditional software development projects may benefit by, for example, instituting some form of the named credit norm or by selecting team members who subscribe to the individual values prevalent in the OSS community.

CONCLUSION

The emerging work on understanding open source software has argued for the importance of understanding what leads to effectiveness in OSS development (Crowston and Scozzi 2002; Ghosh 2002), has pointed to ideology and communication as probable factors in impacting effectiveness (cf. Elliott and Scacchi 2003; Markus et al. 2000), and has presented conflicting opinions about the role of trust in determining effectiveness (Gallivan 2001; Ljungberg 2000). For all the discussion of these issues, there has been no comprehensive elaboration of what constitutes OSS ideology, and little empirical data has been brought to bear on examining antecedents of OSS team effectiveness. This paper has added to the literature by developing a framework of the main tenets of the OSS ideology and suggesting which tenets may be grouped to form relevant sets of values and beliefs that tend to coincide in teams. The study provided initial empirical evidence that some shared ideological components have a positive impact on team effectiveness by enhancing trust and communication quality, while some have negative

effects by either reducing input to the team or by reducing task completion. We hope that these findings encourage other researchers to delve more deeply into the varying roles that ideology plays in OSS projects.

APPENDIX Scales

Communication Quality

Think about the communication among participants in this project. Please use the following scale to rate how frequently each kind of communication occurs:
1= never 2=very infrequently 3=infrequently 4=sometimes 5=frequently 6=very frequently 7=all the time

1. People on this team answer each other's questions in a timely manner.
2. Team members' responses to each other's questions are correct and useful.
3. People on this team answer each other's questions in a thoughtful manner.
4. There is a long lag time between someone asking a question and someone else on the team responding*. *[Item dropped]*

Open Source Ideology

Please indicate the extent to which the values, beliefs, and norms of the group are represented by each of the following statements:
1= strongly disagree 2= disagree 3=disagree somewhat 4=neither agree nor disagree 5=agree somewhat 6= agree 7= strongly agree

OSS Norms

1. Members of this group think that it is wrong to fork a project.
2. Members of this group believe it is inappropriate to distribute code changes without going through the proper channels.
3. Members of this group think it is OK to remove someone's name from a project without that person's consent*.

OSS Beliefs

1. Members of this group believe that the best code wins out in the end.
2. Members of this group believe free software is better than commercial software.
3. Members of this group think information should be free.
4. Members of this group believe that with enough people working on a project, any bug can be quickly found and fixed.
5. Members of this project think that practice is better than theory. *[Item dropped]*
6. Members of this group believe that you only become a hacker when others call you a hacker.

OSS Values

1. Members of this group value sharing knowledge.
2. Members of this group believe in helping others.
3. Members of this group place great value on technical knowledge.
4. Members of this group are driven by a desire to learn new things.
5. Members of this group think cooperation is important.
6. Members of this group value the reputation they gain by participating in open source projects.

Affective Trust

Each of the statements below refers to how the participants in your project feel about each other. Please indicate the extent to which you agree or disagree with each statement about the group using the following scale:
1= strongly disagree 2= disagree 3=disagree somewhat 4=neither agree nor disagree 5=agree somewhat 6= agree 7= strongly agree

1. Members of the team have made considerable emotional investments in our working relationships.
2. Members of the team have a sharing relationship with each other. We can freely share our ideas, feelings, and hopes.
3. On this team we can talk freely with each other about difficulties we are having and know that others will want to listen.

4. Members of the team would feel a sense of loss we could no longer work together.
5. If a member for this group shared problems with other members, they would respond constructively and caringly. *[Item dropped]*

Cognitive Trust

Each of the statements below refers to how the participants in your project feel about each other. Please indicate the extent to which you agree or disagree with each statement about the group using the following scale:
1= strongly disagree 2= disagree 3=disagree somewhat 4=neither agree nor disagree 5=agree somewhat 6= agree 7= strongly agree

1. Members of the team know that everyone on the team approaches their work with professionalism and dedication.
2. Given the track records of the team members, we see no reason to doubt each other's competence and preparation for a job.
3. Members of the team believe they will be able to rely on other members of the team not to make a job more difficult by careless work.
4. Members of the team are concerned with monitoring each other's work*. *[Item dropped]*
5. Members of the team believe that other members should be trusted and respected as coworkers. *[Item dropped]*
6. Members of the team consider each other to be trustworthy. *[Item dropped]*

* Reverse Coded Items

REFERENCES

Agarwal, R., and Karahanna, E. "Time Flies When You Are Having Fun: Cognitive Absorption and Beliefs About Information Technology Usage," *MIS Quarterly* (24:4), 2000, pp 665-694.

Ammeter, A.P. "Determinants of Interpersonal Trust in Workgroup Relationships," University of Texas Management thesis, 2000.

Ancona, D.G., and Caldwell, D.F. "Demography and Design: Predictors of New Product Team Performance," *Organization Science* (3:3), 1992, pp 321-341.

Bagozzi, R.P., and Dholakia, U.M. "Intentional Social Action in Virtual Communities," *Journal of Interactive Marketing* (16:2), 2002, pp 2.

Bagozzi, R.P., Yi, Y., and Phillips, L.W. "Assessing Construct Validity in Organizational Research," *Administrative Science Quarterly* (36:3), 1991, pp 421-438.

Bar-Tal, D. *Shared Beliefs in a Society - Social Psychological Analysis* Sage, Thousand Oaks, CA, 2000.

Barker, J.R. "Tightening the Iron Cage: Concertive Control in Self-Managing Teams," *Administrative Science Quarterly* (38:3), 1993, pp 321-341.

Bergquist, M., and Ljungberg, J. "The Power of Gifts: Organizing Social Relationships in Open Source Communities," *Information Systems Journal* (11:4), 2001, pp 305-320.

Bettenhausen, K., and Murnighan, J.K. "The Emergence of Norms in Competitive Decision-Making Groups," *Administrative Science Quarterly* (30:3), 1985, pp 318-335.

Bettenhausen, K.L., and Murnighan, J.K. "The Development of an Intragroup Norm and the Effects of Interpersonal and Structural Challenges," *Administrative Science Quarterly* (36:1), 1991, pp 20-35.

Bretthauer, D. "Open Source Software: A History," *Information Technology and Libraries* (21:1), 2002, pp 3-11.

Capron, L., and Hulland, J. "Redeployment of Brands, Sales Forces, and General Marketing Management Expertise Following Horizontal Acquisitions: A Resource-Based View," *Journal of Marketing* (63:2), 1999, pp 41-54.

Carmel, E. *Global Software Teams: Collaborating across Borders and Time Zones* Prentice Hall, Upper Saddle River, NJ, 1999.

Crocker, J., and Luhtanen, R. "Collective Self-Esteem and in-Group Bias," *Journal of Personality and Social Psychology* (58:1), 1990, pp 60-47.

Crowston, K., Annabi, H., and Howison, J. "Defining Open Source Software Project Success," International Conference on Information Systems, Seattle, WA, 2003.

Crowston, K., and Kammerer, E.E. "Coordination and Collective Mind in Software Requirements Development," *IBM Systems Journal* (37:2), 1998, pp 227-246.

Crowston, K., and Scozzi, B. "Open Source Software Projects as Virtual Organizations: Competency Rallying for Software Development," *IEE Proceedings Software* (149:1), 2002, pp 3-17.

Denning, D.E. "Concerning Hackers Who Break into Computer Systems," 13th National Computer Security Conference, Washington, D.C., 1990, pp. 653-664.

Dholakia, U.M., Bagozzi, R.P., and Pearo, L.K. "A Social Influence Model of Consumer Participation in Network- and Small-Group-Based Virtual Communities," *International Journal of Research in Marketing* (21:3), 2004, pp 241.

Doney, P.M., and Cannon, J.P. "An Examination of the Nature of Trust in Buyer-Seller Relationships," *Journal of Marketing* (61:2), 1997, pp 35-51.

Elliott, M.S., and Scacchi, W. "Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration.," GROUP, 2003, pp. 21-30.

Ethiraj, S.K., Kale, P., Krishnan, M.S., and Singh, J.V. "Where Do Capabilities Come from and How Do They Matter? A Study in the Software Services Industry," *Strategic Management Journal* (26:1), 2005, pp 25-45.

Faraj, S., and Sproull, L. "Coordinating Expertise in Software Development Teams," *Management Science* (46:12), 2000, pp 1554-1568.

Feller, J., and Fitzgerald, B. *Understanding Open Source Software Development* Addison-Wesley, London, 2002.
Fornell, C., and Bookstein, F.L. "Two Structural Equation Models: Lisrel and Pls Applied to Consumer Exit-Voice Theory," *Journal of Marketing Research* (19:4), 1982, pp 440-452.
Gallivan, M.J. "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies," *Information Systems Journal* (11:4), 2001, pp 277-304.
Gefen, D., Straub, D., and Boudreau, M.-C. "Structural Equation Modeling and Regression: Guidelines for Research Practice," *Communications of the AIS* (4:7), 2000.
Ghosh, R.A. "Free/Libre Open Source Software: Survey and Study," in: *Workshop on Advancing the Research Agenda on Free/Open Source Software*, Brussels, Netherlands, 2002.
Gibson, C.B., Randel, A.E., and Earley, P.C. "Understanding Group Efficacy: An Empirical Test of Multiple Assessment Methods," *Group & Organization Management* (25:1), 2000, pp 67-97.
Gopal, A., Sivaramakrishnan, K., Krishnan, M.S., and Mukhopadhyay, T. "Contracts in Offshore Software Development: An Empirical Analysis," *Management Science* (49:12), 2003, pp 1671-1683.
Gosain, S. "Looking through a Window on Open Source Culture: Lessons for Community Infrastructure Design," *Systemes d'Information et Management* (8:1), 2003, pp 11-42.
Guzzo, R.A., and Dickson, M.W. "Teams in Organizations: Recent Research on Performance and Effectiveness," in: *Annual Review of Psychology,* J.T. Spence, J.M. Darley and D.J. Foss (eds.), Annual Reviews, Palo Alto, CA, 1996, pp. 307-338.
Guzzo, R.A., Yost, P.R., Campbell, R.J., and Shea, G.P. "Potency in Groups: Articulating a Construct," *British Journal of Social Psychology* (32:1), 1993, pp 87-106.
Hardin, C., and Higgins, E.T. "Shared Reality: How Social Verification Makes the Subjective Objective," in: *Handbook of Motivation and Cognition,* R.M. Sorrentino and E.T. Higgins (eds.), Guilford Press, New York, 1996, pp. 28-84.
Hars, A., and Ou, S. "Working for Free? Motivations for Participating in Open Source Projects," *International Journal of Electronic Commerce* (6:3), 2002, pp 25-39.
Herbsleb, J.D., and Mockus, A. "An Empirical Study of Speed and Communication in Globally-Distributed Software Development," *IEEE Transactions on Software Engineering* (29:6), 2003, pp 1-14.
Hofstede, G. *Culture's Consequences* Sage, Beverly Hills, CA, 1980.
Hofstede, G., Neuijen, B., Ohayv, D.D., and Sanders, G. "Measuring Organizational Cultures: A Qualitative and Quantitative Study across Twenty Cases," *Administrative Science Quarterly* (35:2), 1990, pp 286-316.
Holmes, J.G., and Rempel, J.K. "Trust in Close Relationships," in: *Close Relationships,* C. Henrick (ed.), Sage, Newbury Park, CA, 1989, pp. 187-202.
Ilgen, D.R., Hollenbeck, J.R., Johnson, M., and Jundt, D. "Teams in Organizations: From Input-Process-Output Models to Imoi Models," *Annual Review of Psychology* (56:1), 2005, pp 517.
Jarvenpaa, S., and Leidner, D. "Communication and Trust in Global Virtual Teams," *Organization Science* (10:6), 1999, pp 791-815.
Kirsch, L.J. "Portfolios of Control Modes and Is Project Management," *Information Systems Research* (8:3), 1997, pp 215-239.
Kirsch, L.J., Sambamurthy, V., Ko, D.-G., and Purvis, R.L. "Controlling Information Systems Development Projects: The View from the Client," *Management Science* (48:4), 2002, pp 484-498.
Koch, S., and Schneider, G. "Results from Software Engineering Research into Open Source Development Projects Using Public Data," *Diskussion zum Tagigkeitsfeld Informationverarbeitung und Informationswirtschaft* (22), 2000, pp 1-16.
Kogut, B., and Metiu, A. "Open Source Software Development and Distributed Innovation," *Oxford Review of Economic Policy* (17:2), 2001, pp 248-264.

Krishnamurthy, S. "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects," *First Monday* (7:6), 2002.
Kuwabara, K. "Linux: A Bazaar at the Edge of Chaos," First Monday, 1999.
Lerner, J., and Tirole, J. "Some Simple Economics of Open Source," *Journal of Industrial Economics* (50:2), 2002, pp 197-234.
Lickel, B., Hamilton, D.L., Wieczorkowska, G., Lewis, A., Sherman, S.J., and Uhles, A.N. "Varieties of Groups and the Perception of Group Entitativity," *Journal of Personality and Social Psychology* (78), 2000, pp 223-246.
Ljungberg, J. "Open Source Movements as a Model for Organizing," *European Journal of Information Systems* (9:4), 2000, pp 208-216.
Malone, T.W., and Laubacher, R.J. "The Dawn of the E-Lance Economy," *Harvard Business Review* (76:5), 1998, pp 144-152.
Markus, M.L., Manville, B., and Agres, C.E. "What Makes a Virtual Organization Work?" *Sloan Management Review*), 2000, pp 13-26.
Martins, L.L., Gilson, L.L., and Maynard, M.T. "Virtual Teams: What Do We Know and Where Do We Go from Here?" *Journal of Management* (30:6), 2004, pp 805.
McAllister, D.J. "Affect- and Cognition-Based Trust as Foundations for Interpersonal Cooperation in Organizations," *Academy of Management Journal* (38:1), 1995, pp 24-59.
Mensching, R. "Interview with Rob Mensching of Microsoft's Wix Project," OSDir, 2004.
Meyerson, D., Weick, K., and Kramer, R.M. "Swift Trust and Temporary Groups," in: *Trust in Organizations,* R.M. Kramer and T.R. Tyler (eds.), Sage, Thousand Oaks, CA, 1996, pp. 166-195.
Mockus, A., Fielding, R.T., and Herbsleb, J.D. "Two Case Studies of Open Source Software Development: Apache and Mozilla," *ACM Transactions on Software Engineering and Methodology* (11:3), 2002, pp 309-346.
Montoya-Weiss, M.M., Massey, A.P., and Song, M. "Getting It Together: Temporal Coordination and Conflict Management in Global Virtual Teams," *Academy of Management Journal* (44:6), 2001, pp 1251-1262.
Moorman, R.H., and Blakely, G.L. "Individualism-Collectivism as an Individual Difference Predictor of Organizational Citizenship Behaviors," *Journal of Organizational Behavior* (16:2), 1995, pp 127-142.
Nelson, K.M., and Cooprider, J.G. "The Contribution of Shared Knowledge to Is Group Performance," *MIS Quarterly* (20:4), 1996, pp 409-432.
Ouchi, W.G. "A Conceptual Framework for the Design of Organizational Control Mechanisms," *Management Science* (25:9), 1979, pp 833-848.
Postmes, T., Spears, R., and Lea, M. "The Formation of Group Norms in Computer-Mediated Communication," *Human Communication Research* (26:3), 2000, pp 341-371.
Raymond, E. "Homesteading the Noosphere," 1998.
Raymond, E. "The Cathedral and the Bazaar," 2000.
Raymond, E.S. "The New Hacker Dictionary," 2000.
Raymond, E.S. *The Cathedral and the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary* O'Reilly, Sabastopol, CA, 2001.
Raymond, E.S. "How to Become a Hacker.," 2003.
Scacchi, W. "Understanding the Requirements for Developing Open Source Software Systems," *IEE Proceedings on Software* (149:1), 2002, pp 24-39.
Scarbrough, E. "Attitudes, Social Representations, and Ideology," in: *The Social Psychology of Widespread Beliefs,* C. Fraser and G. Gaskell (eds.), Oxford University Press, Oxford, UK, 1990, pp. 99-117.
Segars, A.H., and Grover, V "Strategic Information Systems Planning Success: An Investigation of the Construct and Its Measurement," *MIS Quarterly* (22:2), 1998, pp 139-163.

Seidler, J. "On Using Informants: A Technique for Collecting Quantitative Data and Controlling Measurement Data in Organization Analysis," *American Sociological Review* (39:12), 1974, pp 816-831.
Smith, T. "Open Source: Enterprise Ready - with Qualifiers," TheOpenEnterprise.com, 2002.
Sparrowe, R.T., Liden, R.C., Wayne, S.J., and Kraimer, M.L. "Social Networks and the Performance of Individuals and Groups," *Academy of Management Journal* (44:2), 2001, pp 316-325.
Stallman "Why Software Should Be Free," 1992.
Stevens, J. *Applied Multivariate Statistics for the Social Sciences* Lawrence Erlbaum Associates, Mahwah, NJ, 1996.
Sturmer, M. "Open Source Community Building," University of Bern Institue of Information Systems thesis, 2005.
Tajfel, H. "Social Categozation, Social Identity, and Social Comparison.," in: *Differentiation between Social Groups,* H. Tajfel (ed.), Academic Press, London, 1978, pp. 61-76.
Tetlock, P.E. "Cognitive Biases and Organizational Correctives: Do Both Disease and Cure Depend on the Politics of the Beholder?" *Administrative Science Quarterly* (45:2), 2000, pp 293-328.
Trice, H.M., and Beyer, J.M. *The Cultures of Work Organizations* Prentice Hall, Englewood Cliffs, NJ, 1993.
van Dijk, T.A. "Ideological Discourse Analysis," *New Courant* (4), 1995, pp 135-161.
Von Hipple, E., and Von Krogh, G. "Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science," *Organization Science* (14:2), 2003, pp 209-223.
von Krogh, G., Spaeth, S., and Lakhani, K.R. "Community, Joining, and Specialization in Open Source Software Innovation: A Case Study," (32:7), 2003, pp 1217.