

**OPEN SOURCE SOFTWARE:**

5        **WHAT WE KNOW (AND DO NOT KNOW) ABOUT MOTIVATIONS TO  
CONTRIBUTE\***

Georg von Krogh

Stefan Haefliger

10        Sebastian Spaeth

Martin Wallin

Department of Management, Technology, and Economics

ETH Zurich

Switzerland

15

September 2008

---

\* We gratefully acknowledge the feedback from reviewers and participants at the DRUID Conference 2008, the University of Gothenburg research seminar, and the Open and User Innovation Workshop 2008 at Harvard Business School. The last three authors are to be considered joint second authors, with names reported in alphabetical order. The corresponding author is Sebastian Spaeth: [sspaeth@ethz.ch](mailto:sspaeth@ethz.ch) or +41 44 632 8837.



## INTRODUCTION

Over the last 15 years, OSS products have made successful inroads into many information  
45 systems (IS) segments, attracting millions of users. Today, firms are both heavy users of OSS  
products and contributors to them. As a result, IS managers are increasingly dependent on  
development resources outside their direct control. Thus, IS managers are concerned with  
what motivates outside contributors to participate in the creation of a public good such as  
OSS. For example, if a firm decides to invest millions of dollars in order to migrate its  
50 servers to a Linux-IBM system, managers will want to know to what extent Linux will  
continue to receive contributions by volunteers, how the software will evolve, and if Linux  
and other OSS projects manage regularly to release new and improved versions of the  
software.

The question of why people contribute to a public good through acting collectively  
55 has been much researched in economics, sociology and psychology (Olson, 1965; Melucci,  
1999; Oliver, 1993; Klandermans, 1997). A key research question initiating IS-related  
research on motivations to contribute to OSS development was formulated by Lerner and  
Tirole (2002): Why would thousands of top-notch software developers contribute for free to  
the creation of a public good? Lerner and Tirole's question posed huge challenges for  
60 scholars who studied IS development within firms that systematically use pay and careers to  
incentivize developers. The question was attractive because answering it would mean  
explaining a major phenomenon of high academic and practical interest. Theory and research  
on motivation shed light on critical issues such as the emergence and growth of OSS projects,  
their organization, and sustainability (Ulhoi, 2004; West and O'Mahoney, 2005; Markus,  
65 2007).

One of the first objectives of this paper is to review major contributions to the

literature on motivation to contribute to the development of OSS. We seek to provide interested scholars with an entrée to the literature, and provide a platform for new research on the topic. Through an examination of previous work, we identify research gaps and contribute  
70 a set of new research questions. We show that theory and research on motivations is organized in two phases. In phase one, academic work attempted to explain why voluntary, unpaid developers contribute to OSS—a direct response to the original question by Lerner and Tirole (2002). In their pioneering work, Lerner and Tirole used economic theory to propose that signaling incentives were available and strong in the labor market for software  
75 developers and that developers contribute to OSS in order to increase their human capital. In phase two, researchers ventured beyond straightforward connections between motivation and contribution by studying the interrelationship of motivations, contributions, and institutional arrangements. Institutional arrangements such as community sponsorships and license restrictiveness impact individuals' motivation (Shah, 2006; Roberts et al., 2006).

80 Analyzing work in the two phases, we show that research must grapple with a larger and thus-far hidden issue: the relationship between institutions that provide incentives such as careers and pay for developers, and practices that incentivize developers through the pursuit of a craft, access to a broader array of compensation, and even moral concerns. Thus, we argue for a third phase of research, where IS scholars begin to examine the nature of  
85 contributions and the motivation to contribute, as they emerge from the tension between institutions and practices. We introduce the sociology of Alasdair MacIntyre (for an introduction to this, see Knight, 1998) in order to shed new light on how developers shape norms, values, and commitments related to the excellence of the goods they produce. MacIntyre theorizes that an institution offering pay and career advancement is limited in  
90 motivating practitioners to achieve excellence in their products (using MacIntyre's theory in a

strict sense, institutions tend to destroy these efforts). For example, consider Lerner and Tirole's proposition about the signaling incentive. On the one hand, firms may tend to reward developers who are most "visible" and "productive" in terms of coding volume. From the vantage point of MacIntyre's theory, when such institutional incentives are strong, developers  
95 may find it more beneficial to contribute a large volume of software across many projects rather than providing small, targeted, high quality (but obscure) software patches. On the other hand, social practices with strong values and norms attached to public goods can motivate people to strive for excellence in products and processes, contributing a small amount of innovative software code (Hertel et al., 2003).

100 Our paper is organized as follows: the next section describes the method used for this review and examines research in phase one. Subsequently, we review work in phase two. Next, we introduce a new research agenda based on the sociology of Alasdair MacIntyre, and develop a matrix linking context to motivations. This section develops a set of research questions aimed at sustaining and developing research on OSS development. The last section  
105 contains a short discussion and conclusion.

## **TWO PHASES OF PRIOR RESEARCH ON OPEN SOURCE MOTIVATION**

### **Review Method**

110 Academic work on motivation in OSS originates in different disciplines, including organization and management theory, anthropology, economics, psychology, and sociology, and applies a variety of qualitative and quantitative research designs. We decided to adopt synthetic, qualitative, and critical analysis methods in our review. There were four main steps in the identification of material . First, we identified articles listed in the ISI Database that

115 contained the keywords “open source software” and “motivation.” Second, we reviewed the  
reference list of these articles to identify new sources not listed in the ISI or relevant ISI  
papers excluded by the search criterion. Third, we browsed online paper repositories (e.g.  
www.opensource.mit.edu) to identify articles that matched the ISI search criterion. Fourth,  
we included topical conference papers and book chapters that were known to us and our  
120 colleagues in the field (and which had escaped the previous identification). This provided  
comprehensive material for the review and allowed us to identify a dominant taxonomy in the  
literature to cluster the motivation factors identified. Theoretical and conceptual papers were  
examined for motivational factors that were used to create theory. Motivational factors that  
proved relevant in empirical papers were also included in the review. The inclusion of  
125 empirical studies is comprehensive, whereas the inclusion of purely theoretical contributions  
is more selective. If a study used a different terminology, but the specific motivation seemed  
sufficiently close to an existing one in the taxonomy, it was merged into the existing category.

### **Phase One: The Motivation to Contribute**

130 Studies of individual motivation to contribute to OSS development can be grouped into  
categories depending on the stream of the literature that forms the basis for theory building,  
data gathering, and analysis. While various frameworks have been used, the most frequent  
has been the distinction between intrinsic and extrinsic motivation in self-determination  
theory. This distinction is based on different reasons that bring about action. An activity is  
135 extrinsically motivated when it is done in order to obtain some separable outcome, whereas  
an activity is intrinsically motivated when the activity is done for the inherent interest or joy  
of performing it (Deci and Ryan, 1985). A number of empirical studies have shown that OSS  
developers have both intrinsic and extrinsic motivations for contributing to its development

(Hars and Ou, 2002; Lakhani and Wolf, 2005; Roberts et al., 2006; Wu et al., 2007).

140 Following the work by Lindenberg (2001) and Lakhani and Wolf (2005), Osterloh and Rota  
(2007) distinguished between enjoyment-based intrinsic motivation and obligation-  
/community-based intrinsic motivation. The latter paper provided a theoretical overview,  
whereas Lakhani and Wolf presented data showing that both types of intrinsic motivation as  
well as extrinsic motivation impacted on work effort. Wu et al. (2007) also used the  
145 framework to explain the continued intention to contribute to OSS projects. Hars and Ou  
(2002) suggested that intrinsically motivated contributors spend more time and effort in open  
source projects than extrinsically motivated developers, but did not examine this empirically.  
Other empirical studies concentrated on intrinsic motivation rather than extrinsic motivation.  
For example, Lakhani and von Hippel (2003) linked feelings of competence and fun to  
150 willingness to help other developers. This work stands in contrast to Lerner and Tirole's  
explanation based on extrinsic motivation (2002). Authors have also viewed motivation in  
relation to reciprocity, for example giving software patches as "gifts" to the community  
(Bergquist and Ljungberg, 2001; Wu et al., 2007), reciprocal helping behavior (i. e. helping  
because having been helped or expecting to be helped) (Lakhani and von Hippel, 2003), or  
155 status motivation (Roberts et al., 2006).

While alternative frameworks have been proposed, they are often closely related to  
Deci and Ryan's original framework of extrinsic and intrinsic motivation. For example,  
Bonaccorsi and Rossi (2006) distinguished between economic, social and technological  
motivations, building on a taxonomy proposed by Feller and Fitzgerald (2002). Economic  
160 motivation is similar to extrinsic motivation, and social motivation is close to intrinsic  
motivation. However, the authors also suggest a third type—"technological motivation"—  
that includes benefits from learning and working with a "bleeding-edge" technology.

Attempts toward a broader and integrative framework can be found in Hemetsberger (2004) and Hertel et al. (2003). Hemetsberger viewed motivation as “self-interest” and “others-orientation.” Self-interest was further divided into task- and product-related motivation (corresponding to intrinsic motivation) and others-orientation, including long-term utilitarian goals and social significance (corresponding to extrinsic motivation), was divided into internalized group goals and values, and socio-emotional relationships. Hertel et al. (2003) extended a model of voluntary action in social movements proposed by Klandermans (1997). The authors included collective, norm-oriented, reward, and identification motives, and combined these with the “VIST-model” (Hertel, 2002) that explains individual motivation in virtual teams<sup>1</sup>.

There are three reasons why the extrinsic/intrinsic framework became dominant in the study of motivations to contribute to OSS. First, the framework is widely used in psychology and economics to explain why and how much people contribute to economic activity (for overviews see Deci and Ryan, 1985; Benabou and Tirole, 2003; Frey, 1997). It is backed up by strong empirical research findings and guided by many years of experience of various empirical research designs. Second, using one dominant framework allowed for the comparison of results across project samples. Third, as observed by Lerner and Tirole (2002), OSS is a successful outcome of collective action by people who may or may not be paid for their contributions. While IS development normally hinges on the work of paid developers, the context of OSS represents an interesting phenomenon for the study of complementary intrinsic motives in collective action (e.g., Lakhani and Wolf, 2005; Hars and Ou, 2002).

Most of the work on motivation based on the extrinsic/intrinsic framework can be grouped into four categories of intrinsic motivation (Table 1), four categories of internalized

---

<sup>1</sup> VIST stands for valence (the subjective evaluation of goals), instrumentality (the perceived importance of one’s own contribution), self-efficacy (team members’ perceived ability to perform the required activities for the team task), and trust (the expectation of reciprocity rather than exploitation).

extrinsic motivation (Table 2), and two categories of extrinsic motivation (Table 3). Some motivations are by definition extrinsic but contributors could internalize them, so that they are perceived as self-regulating behavior rather than external impositions (Roberts et al., 2006; Deci and Ryan, 1987). Internalized extrinsic motivations include reputation,  
190 reciprocity, learning, and own-use value. Pure extrinsic motivations include careers and pay.

--- insert tables 1-3 about here ---

While the research in phase one generated a clear link between extrinsic and intrinsic  
195 motives and contributions, it did not relate individual motivation to the quality of the contributions made. Nor did it connect individual motivation to institutional arrangements of OSS development. However, decades of research into other forms of collective action, ranging from lobbying and preservation of natural resources, to money collection for a good cause, show that institutions and individual motivations are interrelated (Morris and Mueller,  
200 1992). This phase failed to generate a robust generalizable theory that spawned a larger population of open source projects, because as Figure 1.3 shows, data were collected within very few projects, and so researchers could not observe the relative influence of institutions on motivation. Yet OSS development is a form of collective action that shapes institutions, and which in turn enables individuals to contribute (von Hippel and von Krogh, 2003). Phase  
205 two research tries to remedy this weakness by explicating the relationship between institutions and motivation.

### **Phase Two: Motivation and Institutions**

Institutional arrangements such as organizational sponsorship (Shah, 2006), technical and

social barriers to joining a developer community (von Krogh et al., 2003), and software  
210 licensing (Stewart et al., 2006) may be outside the direct control of single individual  
contributors, but these still influence motivations to contribute. In self-determination theory,  
extrinsic and intrinsic motivation is both a predictor and an outcome of institutional  
arrangements, depending on “the nature of the study and the way and time in which self-  
determined motivation is measured” (Sheldon and Krieger, 2007: 885). While some authors  
215 discuss the interrelationship between the motivation of OSS developers and the institutional  
arrangements that impact on the development, most of this research is recent and difficult to  
categorize (von Hippel and von Krogh, 2003; von Krogh and von Hippel, 2006; Shah, 2006).  
Most authors cast their own framework across institutions and individuals. Therefore, our  
review proceeds along seven kinds of institutional arrangement proposed within this  
220 literature, and then discusses the special role of crowding theory (Frey and Jegen, 2001),  
which explains mutual adjustments among motivation factors given specific institutional  
arrangements. For a definition of “institutions” we follow North (1986), who includes  
regularities in repetitive interactions among individuals, thus, customs and rules that “provide  
a set of incentives and disincentives for individuals” (North, 1986: 231).

225         The seven kinds of institutional arrangement identified are: governance, community  
sponsorship, license restriction, the provision of rewards, infrastructure and tools, code  
architecture and design, and social and technical exposure to a specific community (see Table  
4).

230 --- insert table 4 here ---

**Governance.** Markus (2007) defined OSS governance as “the means of achieving the

direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of an OSS development project to which they jointly contribute” (2007: 152). While governance in OSS has been described in terms of structure, practices, 235 rules, and norms, it leaves unanswered an important question that relates directly to motivation: what is the source of control in OSS development communities? (Markus, 2007: 153). This question points to the organizational sponsorship of OSS development. Shah (2006) distinguished between open and gated source communities, where gated referred to 240 limited accessibility to the development process due to firm sponsorship (and control. She found that, in the long run, developers who were mainly motivated by use value tended to contribute to gated source communities, whereas developers mainly motivated by enjoyment contributed to open source communities. Considerations of fairness and reciprocity explain this behavior. The developers were aware of the property rights situation in the gated source 245 communities and suspected firms of acting “strategically,” that is, neglecting the needs of the community and pursuing the firm’s own interests. Thus, developers in gated source communities contributed only if they derived direct use value from their work (Shah, 2006: 1011).

250 **Community Sponsorship.** Stewart et al. (2006) distinguished between market (e.g., firms) and non-market (e.g., universities) sponsors and concluded that developers perceive cues as to the project’s future, and hence incentives to contribute, from the type of sponsor.

**License Restriction.** Stewart et al. (2006) found that license restriction impacts on 255 developers’ motivations. The effects, however, are interrelated. They found that open source projects with a non-market sponsor attract greater development activity than projects without

a sponsor. Less restrictive licenses tend to attract more development activity to a project (Stewart et al., 2006; Fershtman and Gandal, 2004). However, “the presence of a non-market sponsor may alleviate concerns as to the project’s future in the same way as a restrictive  
260 license would, in the sense that the restrictive license is not perceived as necessary to protect the developers’ interests” (Stewart et al., 2006: 141).

**The Provision of Rewards.** Pay is frequently used for examining extrinsic motivation (Roberts et al., 2006). Alexy and Leitner (2007) studied the impact of rewards and payment  
265 norms within a community on developers’ intentions to contribute to a (fictitious) OSS project. They found that developers’ total motivation increased when they were offered a monetary reward on completion. When the reward was offered, a no-payment norm did not lower total motivation but a payment norm increased it. With the reward, intrinsic motivation increased given a payment norm and decreased given a no-payment norm (Alexy and Leitner,  
270 2007).

Hann et al. (2002) showed that the degree of contribution did not impact on Apache developers’ salaries (i.e., did not raise them). However, a higher rank within the Apache community significantly correlated with higher wages, suggesting that the programmers’ position (or role) within the community signals productivity to the employer. This finding  
275 confirms the signaling argument, first advanced by Lerner and Tirole (2002).

**Infrastructure and Tools.** Information technology, including low-cost access to bandwidth and communication and coordination infrastructure, enables individual contributions in two ways. First, the infrastructure lowers the cost of sharing work and provides easy access to  
280 others’ work (von Hippel and von Krogh, 2003). Second, tools such as versioning control,

repositories, and mailing lists enable direct learning through feedback and coordination (Lee and Cole, 2003).

**Code Architecture and Design.** Research on contribution levels has also taken a new  
285 direction, inspired by the work of Franke and von Hippel (2003), who demonstrated that the technical design of OSS relates to developer motivation. This work prepared the ground for other researchers, such as Baldwin and Clark (2006), to investigate the relationship between design and motivation. These authors developed a game-theoretic model that takes into account the extent to which a software architecture is “modular.” Their analysis showed that  
290 developers have less incentive to “free-ride,” and thus contribute more to OSS with a modular architecture.

**Social and Technical Exposure to a Specific Community.** The exposure to a community over time creates opportunities in terms of advancement within the social structure of the  
295 community (Rullani, 2007; von Krogh et al., 2003) and in terms of insights that can lead to more challenging tasks (Shah, 2006). In order to get their work accepted by established developers and be able to exert influence on the technical design in the project, newcomers have to adhere to behavioral scripts for joining the community. Adherence to “joining scripts” requires time for developers to advance to community leadership or other central positions  
300 (von Krogh et al., 2003; O’Mahony and Ferraro, 2007). A central position within the community represents an incentive for developers. There are more frequent and direct learning opportunities in feedback from many other community members, and developers can influence the technical agenda (Spaeth et al., 2008). Rullani (2007) found that contributions to community activities can be explained by the developers’ social exposure to the

305 community over time. Similarly, Shah (2006) maintained that long-term developers’  
accumulation of knowledge provided opportunities to take up more challenging tasks or tasks  
that require broader knowledge of the code base (multiple modules and components).

The motivations and institutional arrangements in OSS development are also central  
310 in crowding theory, which predicts mutual adjustments among motivations, particularly  
between intrinsic and extrinsic motivation. Luthiger and Jungwirth (2007) showed that 42%  
of the contributions to OSS development emanated from paid working time. Several studies  
agree that paid contributors amount to approximately 40% of all developers (Lakhani and  
Wolf, 2005; Hertel et al., 2003; Hars and Ou, 2002). Counter to economic theory, introducing  
315 pecuniary incentives may not always increase supply of a public good (Frey and Jegen,  
2001). Osterloh and Rota (2007) argued that the profit motive may crowd out sharing of  
software and other knowledge among developers based on reciprocity. Crowding out of  
intrinsic motivations to contribute to OSS development, however, has not occurred due to a  
balance between intrinsic and extrinsic incentives and due to the “pro-social intrinsic  
320 motivation of a sufficient number of participants to contribute to the second order public  
good (enforcing the rules of cooperation)” (Osterloh and Rota, 2007: 169). Indeed, Lakhani  
and Wolf (2005) found that intrinsic and extrinsic motivations co-existed, and Roberts et al.  
(2006) detected no crowding out of intrinsic motivation by extrinsic motivation. However,  
they found that extrinsic motivation did crowd out own-use value motivation (Roberts et al.,  
325 2006). Increased reputation and career concerns (“status motivation”) enhanced developers’  
intrinsic motivations (Roberts et al., 2006).

Finally, Shah’s research showed that own-use value was the initial motivation for  
developers to join an OSS community. However, fun and enjoyment motivated their long-

term contribution. These findings support a conjecture initially proposed by von Hippel and  
330 von Krogh (2003; see also Elster, 1986; Lattemann and Stieglitz, 2005), that the continued  
contribution to OSS development requires stronger and different motives than those present  
in joining. As developers start to work their way into the social practice of OSS development,  
their motivations to continue contributing change.

To summarize, research in phase one failed to create a general theory of motivation in  
335 OSS or to connect motivation with the nature or quality of contributions. Yet, as scholars  
acknowledge, the number of messages sent or the amount of code written cannot approximate  
the quality of the contributions to software (e.g., Koch and Schneider, 2002; von Krogh et al.  
2003). Moreover, institutional arrangement impacts on developers' motivations, as confirmed  
by research in phase two. While this literature proposes seven dimensions of institutional  
340 arrangement, little is known about how individuals or organizations balance the effects and  
increase developers' motivations to contribute high-quality work. In order to address this  
question, it will be beneficial to conduct research on the social practice of OSS development.  
The interplay between institutions and motivation represents one aspect of the social practice  
that favors this focus (MacIntyre, 1984: 181). The emergence of institutions through social  
345 practice is yet not well understood in OSS, and in the remainder of this paper we outline an  
agenda for this research.

### **TOWARD A NEW RESEARCH AGENDA**

Research in phases one and two neglected important aspects of social practice, including the  
ethics and virtues that guide the work of engineers and software developers (Martin, 2002;  
350 2000; Latour, 1999; Friedman and Kahn, 1994). It is reasonable to assume that such elements  
also influence developers' motivations to contribute to OSS (recall that Deci and Ryan (2000)  
defined motivation as being moved to do something). In our review of phase one research, we

showed the dominant role played by self-determination theory in explaining developers' contributions. To date, the framework of intrinsic and extrinsic motivation is detached from  
355 the nature of work and could be applied to any type of human activity. However, the context of work strongly relates to motivations (Morgeson and Humphrey, 2006) and below we explain how context can be brought into motivational studies on OSS through the work of Alasdair MacIntyre.

### 360 **MacIntyre's Views on Social Practice, Institutions and Goods**

In recent years, scholars have taken an increasing interest in the influence of social practices on the evolution of economy and society, often referred to as the "practice turn" in the social sciences (Schatzki et al., 2001). The practice turn has also influenced theorizing in IS development and use (e.g. Suchman et al. 1999, Orlikowski, 2000) but so far, it has not been  
365 applied to OSS development for solving the conundrum about how this context produces particular motivations. Alasdair MacIntyre's seminal work *After Virtue* (1984) presents a theory of increasing importance for understanding social practices, their relationship with institutions, and how they influence individual motivation (Knight, 1998).<sup>2</sup> MacIntyre defines a social practice as "any coherent and complex form of socially established cooperative  
370 human activity through which goods internal to that form of activity are realized in the course of trying to achieve those standards of excellence which are appropriate to, and partly definitive of, that form of activity, with the result that human powers to achieve excellence, and human conceptions of the ends and goods involved, are systematically extended"

---

2 The theory has also found some use in organization studies (e.g., Beadle and Moore, 2006), but as noted by Dawson and Bartholomew (2003) applying it is not unproblematic since MacIntyre heavily criticizes contemporary notions of management and firms as institutions. While this concern must be taken seriously, we will show that the strength of MacIntyre's concept of social practices is that it derives explanatory power from highlighting distinct differences between social practices and institutions.

(MacIntyre, 1984: 187). This definition can be applied to a range of professions such as  
375 architecture, medicine, journalism, science, and the arts, with the precondition that a social  
practice should have wide and positive effects for humankind.<sup>3</sup> Individuals are motivated to  
act through the provision of goods and MacIntyre proposes that social practices create two  
types of goods: external and internal. External goods include capital, status, or power, which  
are privately owned by individuals and organizations. Internal goods are partly defined by the  
380 social practice. Internal goods are “public goods” and thus of benefit to the wider community.  
For example, knowledge is considered an internal good of science. Analogously, software  
publicly available under an open source license is the internal good of OSS development.  
MacIntyre discusses institutions such as governments, political organizations, and firms, and  
sees these as a prerequisite for the organization and sustenance of social practices. However,  
385 he also warns that because institutional goals may conflict with the internal goods of the  
social practice, institutions may constrain or corrupt social practices and demotivate or  
demoralize practitioners. This is the case when institutions pursue limited goals aimed at  
external goods (e.g., excessive profits) at the expense of internal goods that motivate  
practitioners (e.g., achieving excellence in a craft).<sup>4</sup> A strength of MacIntyre’s work is that it

---

3 A social practice is the basis for making decisions about which virtues are called for in particular circumstances and the best way of enacting them (Noel, 1999; Fowers, 2003; Rämö, 2004). Dunne (1993) refers to “ethical knowledge” that directs “ethical action.” Aristotle’s discussion of the “good” refers to man’s practice “of his soul’s faculties in conformity with excellence or virtue” (Aristotle, 1941: 33). The good becomes a metaphysical goal, like truth, justice, and beauty, toward which people strive by adjusting their lives and actions. However, in a modern world “good” may be contested by people who pursue different goals. The very standards that define what is good may be subject to different interests and, therefore, judgment itself will be judged as more or less virtuous (whether something is judged good or bad, right or wrong). MacIntyre argues we should rather understand the common good as internal to a social practice, as a goal to be achieved by its practitioners (Knight, 1998).

4 MacIntyre’s theory emanates from a criticism of the work by Aristotle on ethics and virtues in political leadership. MacIntyre notes that practitioners achieve excellence of character or virtue in pursuing internal goods. To act in a virtuous manner is to emulate the rules of morality rather than simply abiding by them because one is commanded to do so. A social practice, therefore, is a “school of virtue,” where practitioners learn aspects of the internal good, such as ethical reasoning, argumentation, criteria for excellence and product quality, rules of communication, and so forth. Justice, courage, truthfulness, and above all love for the social practice, are cultivated through practitioners’ participation. Practitioners discover and commit to goals that lie beyond their own selfish, short-term, needs and desires. They also realize that they can only achieve the internal goods that are of value to themselves, their social practice, and the wider society, when they emulate the standards of excellence already established within the practice. According to MacIntyre, to

390 highlights such distinct differences between social practices and institutions.

An interpreter of MacIntyre's work, Mike W. Martin (2000; 2002) analyzes motives related to technical and other professional work. Martin distinguishes between craft motives, compensation motives, and motives of moral concern. Craft motives are desires to achieve expertise and to manifest technical skills, theoretical understanding and creativity. When  
395 committed practitioners achieve a standard of excellence, they experience a deep sense of satisfaction. When they do not, however, they have a sense of failure, shame, or regret. Craft motives are associated with individuals embracing professional ideals, such as technical standards of excellence, which evoke their interests and talents with sustained challenge and complexity (Martin, 2000:22). The craft motive is aimed outwards toward other people and  
400 social practices.

Compensation motives cover desires for social rewards such as money, power, authority, good reputation, and job stability. Phase one research on motivations in OSS covers compensation motives, in particular extrinsic and internalized intrinsic elements including pay, career, reputation, and reciprocity. The immediate aim of compensation motives is to  
405 reap benefits for oneself. Martin, however, adds that compensation motives are not purely self-interested but relate to the desire to support family and friends, philanthropic activities, and obtaining resources to help others.

Motives of moral concern include caring for other people, in the sense of having an active desire to promote their well-being for their own sake. Martin uses the example of  
410 medical doctors who act in accordance with ethical codes when they care for their patients' well being. Motives of moral concern also include a desire to maintain moral integrity and self-respect. These also include desires to meet professional responsibilities and maintain

---

pursue internal goods (excellence) is therefore synonymous with cultivating virtues by subordinating oneself and one's relations with others to the reasoning that is internal to the social practice.

personal integrity. In addition, such motives are embedded in virtues like honesty, truthfulness, trustworthiness, justice, courage, loyalty, benevolence, etc.

415

These three motives cannot be viewed in isolation from a social practice (Martin, 2000). In contrast to the application of self-determination theory, Martin's framework of motives relates more directly to technical work and social practice. Therefore, it might be an important complement to motivation studies thus far <sup>5</sup>.

420

MacIntyre argues that there are important, symbiotic relationships but also strong potential tensions between social practices and institutions. Whereas institutions provide practitioners with external goods that satisfy their compensation motives, they may fail to support or corrupt social practices by overshadowing or conflicting with craft motives and motives of moral concern. Much of MacIntyre's critique of institutions aims at management (MacIntyre, 1984). Managers justify their power over others and monetary compensation by implementing techniques and systems for social change. However, because of the complexity of organizations, techniques and systems seldom lead to predictable outcomes, and hence the basis for justification must be false. This critique echoes Robey and Markus's (1984) much cited analysis on the unpredictable outcomes of the design of management information system (MIS). The implementation of planned social change can do more harm than good in helping the social practice's capacity for producing internal goods. Moreover, because

425

430

---

5 Martin also acknowledges that his framework relates to self-determination theory, and suggests the three categories of motivation are all a "wellspring of intrinsic satisfaction" (2000: 24). However, he also suggests that craft motives and motives of moral concern are likely to have the greatest overlap with intrinsic motivation, because the former are directly aimed at the development of a profession, and the latter tend to be directed toward those who are being served by the work of practitioners. Compensation motives, in turn, would overlap more strongly with extrinsic motivation, but here there are also social rewards that are more deeply rooted in the work itself. For example, developers may achieve reputation or status through work performance rather than pay or career. On the one hand Martin provides us with a more fine-grained understanding of intrinsic motivation and an alternative take on extrinsic motivation; on the other, his motive categories can be viewed as an alternative to self-determination theory with a clearer link to social practice. Thus, his categories serve an important analytical purpose in this paper.

managers are often motivated by compensation motives and want to create and appropriate excessive external goods, their actions may undermine practitioners' craft-based and moral concern motives, and ultimately destroy the fabric of the social practice.<sup>6</sup>

435 Finally, MacIntyre's theory predicts that when practitioners (whose motives are influenced by their social practice, virtues, and the provision of internal goods) experience conflicts between internal and external goods, they begin to act collectively. In effect, their collective action gives rise to new institutions that are more efficient and effective in support of social practices. This explains the proliferation of new institutions in society.

440

### **MacIntyre's Theory Applied to Motivations to Contribute**

With one exception (Martin, 2002), MacIntyre's work has not been widely applied to theories of innovation and technology management, and we are not aware of any application to research on IS and OSS development. In this section we seek to show that individual  
445 motivation to create OSS influences, and is influenced by, social practice and institutions (e.g., firms or type of license). We start by outlining individual motives related to developer' technical work and professions.

Martin (2002) extends MacIntyre's view of social practice and goods to technology. He views technological innovation as a social practice in itself, with participants including  
450 firms, users, or other citizens. Technologies fulfilling basic human needs (e.g., communication, entertainment, housing, transport) result from complex cooperative social

---

6 MacIntyre (1984) also raises doubts that management can be considered a social practice because this presumes attention to the creation of internal goods and well being of humanity. Managers' concerns are with techniques and systems for transforming raw materials into products, unskilled labor into skilled labor, and investment into profit (p. 30). Managers decide and act to achieve a desired end state through social change, but are blind to other concerns such as the wider effect of their actions on humanity. Thus, MacIntyre concludes they are not able to engage in moral debates about their own actions. MacIntyre's harsh critique neglects the empirical fact that managers are often concerned about the consequences of their actions and repeatedly engage in a wider discourse with society about the nature of their decisions and activities and the purpose of the institutions they run (see Dawson and Bartholomew, 2003).

activities conducted by many individuals and groups working to standards of excellence (internal good). OSS development is a social practice that involves designers, coders, and testers, but also users and those who are indirectly affected by the technology. The social  
455 practice builds on and extends knowledge of software development, including developers' own experience, algorithms, rules for documenting and testing, design sheets, product roadmaps, artifacts, software modules, languages, and so on.

The social practice of software development can be institutionalized in a firm that sells licenses for the use of the software it develops. The firm applies behavioral rules of  
460 software engineering that use project management techniques and administrative structures (Scacchi, 2002; Kohanski, 2000).<sup>7</sup>

Because of its ubiquitous presence and the wide-ranging impact of IS in all aspects of contemporary human life, software development, like many other areas of engineering and technology (Latour, 1996; van den Burg and van Borg, 2005), should be considered a social  
465 practice in MacIntyre's sense of the concept. However, Scacchi (2002) has written on the difference between traditional software engineering and OSS development practices. He notes that "[OSS developers] enact teamwork structures and relatively flat, peer-oriented, decentralized community forms that reduce/supplant functional organizational forms inherent in traditional [software engineering] techniques that increased bureaucratic tendencies. [OSS]  
470 avoids reliance on formal project management techniques and administrative structures that pervade industrial [software engineering] projects" (Scacchi, 2002: 3). Thus, the social

---

<sup>7</sup> Martin also points out that craft, compensation, and moral concern motives tend to be tightly interwoven in the technical work of engineers and other professions. Moral concerns and compensation motives may be well aligned. Likewise, examples are given where compensation motives give rise to moral concern motives. While this runs counter to previous theory, which warns of ways in which extrinsic motivations could crowd out intrinsic motivations in open source software (Osterloh and Rota, 2007), it supports the result in Roberts et al. (2006) that extrinsic and intrinsic motivations can coexist and explain higher participation levels. Using the current framework, an important issue is the extent to which craft and moral concern motives are reinforced by developers who are both paid and following career paths. At least, when they are paid, developers do not need to worry about their ability to sustain their contribution to the open source software project.

practice is distinctly related to the technical object. Mackenzie (2005) suggests that the OSS code itself, with modular, functional, and transparent objects, gives rise to a social practice outlining rules for behavior. The technical object of the software code requires developers to  
475 behave in a specific way when creating and maintaining it, for example, modularizing, reusing, keeping to the API specification, documenting, and so on (Baldwin and Clark, 2006; Baldwin, 2008). To be an OSS developer means to engage in a social practice and adhere to its specific rules of software development because they enable the creation of a high quality product for its users (von Hippel, 2001). At the same time, according to Scacchi, it also  
480 means abandoning other rules related to institutions of software engineering prevalent in industry.

In IS development, Nelson et al. (2000) found that different factors impact on motivation, contingent on whether people were working in software support or development. OSS, like IS development, provides various levels of satisfaction for developers. For  
485 example, in a study of Gentoo Linux, Monteiro et al. (2006) found that a technical infrastructure (IRC) gives rise to communication rituals that move developers to continue their contributions. These rituals keep developers hooked on development work and prevent their defection.

Following Mackenzie and Martin, we propose that when developers create software  
490 and share it with others using an OSS repository and communication platforms, they gradually develop an understanding of the standards of excellence of the technology object simultaneously with appropriate rules for behavior. As for technology, such standards could be an upper limit for the lines of code needed to solve a mathematical problem, proper ways of documenting code, or of testing before releasing a work product. As for behavior, rules  
495 could include the need to learn what more experienced developers do before contributing

comments or software code, the need to understand an evolving software architecture before contributing, the cycles of testing needed before an official release of the product, and so on. Several of these rules, including observing and learning before contributing software code and technical comments, were revealed by Kuk (2006). Thus, becoming a developer of OSS  
500 means learning the rules of the social practice and its standards of excellence with regards to technology. Martin (2002: 556) suggests: “These (social) practices enter centrally into defining a way of life, with technological development entering even more centrally into the ways of life of engineers. (The technologies are parts of or aspects of ways of life.)” In many people this learning process also means creating a sense of moral obligation for supporting  
505 and further developing the social practice. For example, a form of “moral obligation” is found in the early release of software patches, an integral part of the social practice of open software development (Raymond, 1999). Through early release of prototypical software, dialogue, bug fixing, and technical fine tuning, developers create software that represents “mental models” or the designs, unarticulated needs, and aspirations of current and future  
510 users (von Hippel, 1998; von Hippel, 2003; von Krogh et al., 2003). Motivations of moral concern lead people to empathy, which software developers need to achieve standards of excellence within the social practice (Klein and Herskovitz, 2007). Personal commitment goes beyond commitment to the software product, and extends to external goods like profit, the social practice that constitutes the activity of OSS development, and its broader social  
515 effects. From this perspective, OSS development as social practice is meaningful work and a way of life with all its attendant fascinations — pleasure, challenges, and other inherent benefits.

This discussion shows that developers’ craft motives, compensation motives, and motives of moral concern are tightly interwoven with the nature of work and the goods

520 produced in the social practice of OSS development. When proposing that social practice gives rise to motivations, we also assume that developers can reflect on their own work in relation to the internal good of the social practice.<sup>8</sup> This self-reflection may also ultimately shape their character (Habermas 1991: 163; Flyvbjerg, 2001). Once developers promote the social practice, are motivated by it, and reflect on their actions (Calhoun, 1988), they also  
525 gain insights about their values in relation to that social practice, for example whether or not its virtues fit their own personal virtues. In the next section, we draw upon this important insight in analyzing the emergence of the Free Software Foundation.

### **The Emergence of an Institution**

530 In the 1960s and 1970s, much software development was carried out by scientists and engineers in academic and corporate research laboratories. It is a normal part of the developers' social practice to give and exchange software they had written, in order to modify and build upon each other's software, both individually and collectively, and in turn to make their modifications freely available. The virtue of active and intense sharing was considered  
535 important for learning, efficiency in the development of code, better bug-free products, and, overall, the development of the software engineering profession.

The virtue of sharing work in software development was very strong among a group of developers at MIT's Artificial Intelligence Laboratory during this period (Levy, 1984). The first conflicts between the institutional goals and the virtues of social practice can be traced  
540 back to the 1980s. At this time, MIT decided to license some of the code created by this group to a commercial firm. In accordance with its commercial interests (external goods), the

---

8 However, there is no objective or universally true point of reference from which to understand technology development. Discussing his concept of habitus, Bourdieu (1990: 91) underscored the difficulty of finding such a vantage point: "...there is every reason to think that as soon as he reflects on his practice, the agent loses any chance of expressing the truth of his practice..."

firm restricted access to the source code for that software to the original MIT developers, creating a great deal of frustration and irritation among them. This incident is a good illustration of the kind of conflicts MacIntyre draws our attention to. Whereas software developers at the time considered virtues such as openness, learning, and knowledge-sharing a prerequisite for the social practice's creation of internal goods (including product excellence, professional development, and benefits for others), they felt institutional concerns for the constraint of external goods, which they feared would eventually harm the practice. Richard Stallman, at the time a programmer at MIT's Artificial Intelligence Laboratory, was distressed by the institutional pressure to restrict access to source code and sell software through licensing. He believed it would harm the software engineering profession and hinder humanity's rapidly growing need for better and better technologies. Stallman viewed these practices as "morally wrong" impingements upon the rights of software users to learn and create freely. In his own words, faced with the collapse of his community's social practice, he was faced with "a stark moral choice" (Stallman, 2002):

With my community gone, to continue as before was impossible. ... The easy choice was to join the proprietary software world, signing non-disclosure agreements and promising not to help my fellow hackers. Most likely I would also be developing software that was released under a non-disclosure agreement, thus adding to the pressure on other people to betray their fellows too. I could have made money this way, and perhaps amused myself writing code. But I knew at the end of my career, I would look back on years of building walls to divide people, and I feel I had spent my life making the world a worse place ... So I looked for a way a programmer could do something for the good. I asked myself, was there a program or programs I could write so as to make the community possible again?

Stallman's response to this challenge was to create an institutional arrangement as an alternative to the firm, called the Free Software Foundation. The purpose of the foundation

was to preserve free access to software developed by people who shared the virtues of the  
570 practice. The legal mechanism he developed to support this idea was the GNU General Public  
License, which can be affixed to a piece of software by a developer, and which guarantees a  
number of basic rights to all future developers and users. These include the right to download  
for free, study, and modify the source code, and the right to redistribute to others modified or  
unmodified versions of the software for free. Stallman firmly believed that this license and  
575 the new institution of the Free Software Foundation would support the social practice of  
software development and eventually help create excellent products of benefit to society.<sup>9</sup>

As this analysis shows, craft motives and moral concerns alongside compensation  
motives set in motion Free and OSS development. Stallman was concerned about the  
advancement of craft in software development, as well as the moral choices that new  
580 institutional goals imposed on software developers. His institutional alternative was created  
to preserve the social practice's ability to create internal goods alongside external goods  
pursued by the software industry. Later observers refer to Stallman's institutional alternative  
as a "Free software" ideology that motivates developers to join and contribute to OSS  
(Stewart et al., 2006). However, the origin of collective action was the social practice of  
585 software development that shaped craft motives and moral concerns and prompted Stallman  
to create an institutional alternative. To paraphrase an insight from MacIntyre's theory,  
anything that does not promote the common good, such as the appropriation of collectively  
developed software code, may not be properly regarded as social practice.

For example, an internal good can be a practitioner's pursuit of excellence in software  
590 development — it is against this backdrop we should understand Stallman's moves. Thus, it

---

9 Because of his skills and the need to ensure further openness, Stallman first started work on the GNU operating system, which emulated the design of Unix. In the years to follow, different licenses and organizations emerged that presented variations on this basic idea, such as the Open Source Software Movement (Perens, 1999).

is important to note that the new institution of Free software sprang out of craft motives and practitioners' moral concerns, and these motives are shaped by the social practice of software development. Rather than ascribing to the view that the new institution originates in an overarching ideology that motivates people to act, we conclude that the concern for product quality, work, and the wider implications of Free and OSS gives rise to new institutions. This conclusion differs sharply from other accounts of OSS, like Moody's *Rebel Code* (2001), or social movements that confront the establishment of the software industry on ideological grounds (Stewart et al. 2006; see also the discussion of ideology and collective action in von Hippel and von Krogh, 2003). Finally, consistent with the theory, if the institutions of Free and OSS fail to support social practices that create internal and external goods and enable developers' craft motivations and motivations of moral concern (alongside compensation motives), new institutional alternatives will emerge that do the job better.

### ENTERING PHASE THREE

From this discussion, we can now formulate a set of questions for IS research. What we term "phase three" of research on motivation consists of research areas with corresponding research questions. We found from our review that the links and tensions between individual motivation and social context in OSS development is underdeveloped and often completely missing from the literature in phases one and two. In phase three, we need to make explicit the link between individual motives to contribute and the social context in which those contributions are made. We propose an organizing framework, a motivation-context matrix linking motivations in OSS development that accounts for Martin's categories to the social context in MacIntyre's theory (see Table 5). The columns represent individual motivations: craft motives, compensation motives (including non-financial compensation), and moral

615 concerns. The motivations extend beyond the intrinsic-extrinsic dichotomy and provide a finer grained categorization in the area of intrinsic motivation related to work. The rows consist of the three key conceptual building blocks of MacIntyre's theory: social practice, (internal and external) goods, and institutions.

Highlighting the social aspects of motivation in this manner has two main  
620 consequences. First, people's motivations are interwoven with the social practice in which the work is done. Since social practices and institutions mutually support or constrain each other, motives must be understood by taking into account the institutional arrangements through which social practices evolve. Thus, individual motivation is both enabled and constrained by social practices, goods, and institutions. Second, social context is the antecedent to individual  
625 motivation, and vice versa. We discuss this consequence later, looking at the emergence of institutions.

The motivation-context framework is not explanatory. Its purpose is to assist in generating and classifying new, challenging, and fruitful avenues for future research. The following sections are structured along the rows of the framework, i.e. starting with the social  
630 practice in OSS development, continuing with goods, and finishing with institutions.

--- insert table 5 here ---

### **Social Practice**

635 Developers care about their social practice (here, OSS development) and would like to adhere to its rules and standards of excellence, and improve these as well. Over time, the social practice generates and alters motivation (Shah, 2006). However, multiple motives play a role (Roberts et al., 2006). For each motive, IS researchers should ask how social practice and

motivation mutually impact each other. Table 6 provides an overview of pertinent questions.

640           The desire to achieve expertise, to demonstrate technical skills, theoretical  
understanding and creativity, are individual motivations to participate in the social practice.  
Vice versa, participation in the social practice shapes the craft motive by exposing the  
individual to the intricacies software development. Is producing excellent software code  
sufficient adherence to the standards of excellence within the social practice? Anecdotal  
645 evidence suggests not: in 2000, one of the hackers in the Freenet project rewrote large  
portions of the core code privately over a couple of weeks, without community involvement.  
When he deemed it finished, he ripped out the existing code base and replaced it with his  
code. While his work was acknowledged as technically high-quality source code, the  
community at large was not happy with the lack of transparency and community involvement  
650 in his actions. Because they had not been involved in the development process, other  
developers found it hard to grasp how the new code worked and criticized the development  
process. This developer had clearly not followed the rules and standards of excellent craft  
within the social practice, despite having single-handedly developed a technologically  
excellent solution.

655           Do compensation motives and craft values reinforce or diminish each other? For  
example, developers with a high own-use value (solving their problems) might be more  
interested in fixing the one specific bug or adding the one feature they crave, than striving to  
produce beautiful code. Also, in combination with institutional arrangements, (financial)  
compensation might reduce the level of participation in the social practice. On the other hand,  
660 exposure to the social practice over time might change the perception of compensation values  
for internal or external goods.

Caring for others, moral integrity, self-respect, and adhering to work ethics are issues

of moral concern. In the context of open source, researchers will need to identify the dimensions of moral concern within the social practice (Himanen, 2001). Can we, as is usually done, simply assume that “Free software” and “open source” development are equivalent and cluster them together in research, especially when examining motivations? While both factions might share standards and values on craft, their moral considerations deviate (Stewart and Gosain, 2006). Since the developers’ “love” for social practice impacts on their motivations and vice versa, there might not be only one social practice (open source) —Free software might represent a related, yet separate social practice. Designing institutions that cater to these social practices would then require different approaches.

Finally, an important question: do moral obligations lead to common and accepted rules for behavior? For example, do moral obligations lead developers to create reusable components, even if this means an increased burden of maintaining them? (“This is just how you have to do it in open source;” “I have an obligation to fix bugs in my code, even if they only bother other users and not me.”) Hitherto, morality in open source development has often been seen as an existing but unimportant aspect. However, if moral considerations do shape the practices that help to create superior software, they are much more worthwhile to examine.

680

--- insert table 6 here ---

### **Internal and External Goods**

In MacIntyre’s work, internal and external goods constitute a conduit connecting social practice and institutions by representing the output and direct consequences of the latter. These goods are collective goods in production but not necessarily in consumption (use),

which means that an external good such as status is a collective good in production (all participating in attributing status to people), although it may be privately appropriated (private in consumption by the high status developers). At the same time, open source code  
690 can be both collective in production as well as in consumption. Future research should explore how the goods relate to different motives and how the motives impact on the characteristics and quality of the goods produced. This research will remedy a problem in phases one and two regarding lack of focus on the quality of contributions. Table 7 contains some sample questions.

695 As far as craft motives are concerned, the goods created as part of the social practice are selected by the developers. The self-selection of tasks by developers of OSS represents an important organizational characteristic of OSS development communities (Yamauchi et al., 2000) and plays a central role in the reuse of software (Haefliger et al., 2008). How does the type of software created (the development agenda) impact on contributors' craft motives?  
700 Vice versa, how do craft motives affect the software development agenda? Craft motives drive the creation of high quality work, and the perceived quality of the goods produced may relate to the motivation to contribute quality. How do individual perceptions of quality relate to the quality of goods produced by the community? What is the relationship between (technical) quality and other quality elements of goods produced? One suggested strategy to  
705 investigate software quality involves reuse intensity in the software component market. Frequently (re)used software components signal high component quality (Ahsrafi, 2003; Lim, 1994).

Differences in the perception of quality between individuals and the social practice  
710 may be mended or aggravated by goods, and this should be a focus in future work.

715 Researchers should also explore potential conflict between the quality demands of the social practice and the individual's compensation motives. Here one should keep in mind that compensation can arrive as a financial reward or as use value (see our discussion of phase one research on page 6). Given that use value represents an important motivation for contributors to OSS development, we should ask: how does the use value depend on the characteristic and quality of the goods produced?

720 Moral concerns deal with the well being of others and how these concerns motivate individuals to act (or not). In relation to internal and external goods, this motivation gives rise to a number of research questions. First, moral concerns in OSS development suggest the contributor's motivation to contribute is partly shaped by how his or her resulting actions impact the well being of others, and thus ought to include aspects of helping behavior with no direct and personal payoff. Helping behavior, in turn, can result in the production of internal and external goods. Thus, the following research question is relevant: how is the type of internal good produced through helping other individuals different from other contributions?

725 Second, research should explore how certain goods constrain or enable helping behavior. The answer may lead to insights as to why certain types of goods elicit or allay moral concerns.

730 Third, future work should also explore the tensions and possible conflicts between the need for caring and the pursuit of external goods such as status, power, or profit. Thus, we propose the following research question: what are the potential conflicts between moral concerns and the production and consumption of external goods?

--- insert table 7 here ---

## **Institutions**

735 OSS communities represent a novel organization for the social practice of software development (Scacchi, 2002). They are not associated with one employer or workplace, but integrate individual contributions into a common pool, and assume ownership of their output (O'Mahony and Ferraro, 2007). The emergence of communities as institutions gives rise to a number of research questions that relate to individual developer motivations. On the one  
740 hand, research needs to investigate the emergence of OSS institutions. On the other hand, future academic work needs to target the relationships between different existing and new institutions (e.g. communities and firms). Lastly, open questions relate to the interplay between institutions and moral concerns. A pertinent sample of research questions is shown in Table 8.

745           The craft motivation to contribute to internal goods of the social practice of OSS may lead to institutions such as the community organization in OSS. This process is not yet well understood. O'Mahony and Ferraro's (2007) work on the emergence of governance represents a step toward phase three research. Governance constitutes a central problem of community organization and the authors show that the allocation of authority is rooted in the  
750 values of the social practice of OSS development. These values typically resemble what we call "craft motives" of developers. Community leaders receive limited authority over technical matters and must defer to the majority of the rest of the developers if necessary (O'Mahony and Ferraro, 2007). When and how do individual motives give rise to such institutions and sources of authority? In the literature on collective action, institutional  
755 change is often seen as the result of political struggles over the framing of issues and values (Hargrave and Van de Ven, 2006). We suggest distinguishing between institutional change as understood in the political science literature (e.g. as a process of framing) on the one hand,

and on the other, institutional change explained by craft motives rooted in social practice. The two approaches may account for different models (or forms) of collective action.

760           Institutions compete for talent and, within them, individuals compete for the resources that the institutions provides (status, pay, etc.). Motivations connect directly to the way in which institutions structure work, solve ethical issues, and compensate individual achievements and effort (Alexy and Leitner, 2007). OSS communities may emerge where firms fail to balance motivations, for example by providing hackers with the necessary  
765 freedoms associated with Free software. Vice versa, firms may succeed in attracting and keeping talent if compensation motivations are compatible with craft motives and moral concerns.

          Some institutions may be more receptive to developers' moral concerns, such as consensus-oriented decision making or democratic legitimacy (MacIntyre, 1984). But beyond  
770 compatibility, research needs to explore institutional arrangements that actively engage individuals with strong motivations rooted in moral concerns. Also, it should attempt to answer how institutions are established and sustained, based on moral concerns. Finally, academic work should examine a conundrum: if moral concern accounts for a significant share of individual contributions to OSS, how does competition between institutions (firms,  
775 the Free Software Foundation, Apache Foundation, etc.) affect sensitivity to individuals' moral concerns?

--- insert table 8 here ---

780           Phase three of research on motivation needs to connect the social practice of OSS development and IS development institutions with the individual motivations that, on the one

hand, derive from this social practice and, on the other, generate institutions that support and constrain this social practice. The research matrix that spans types of motivations, social practice, goods, and institutions may inspire future research to take on fundamental problems  
785 of collective action, agency, and institutional change and their relation to individual motivation embedded in the nature of work.

## CONCLUSION

The objective of this paper is twofold: to provide a state-of-the-art review of what is known about motivations to contribute to OSS development, and to reinvigorate the research field by  
790 providing a framework with a set of new research questions. Our review of motivation studies in OSS development segmented previous work in three phases, phase three of which includes suggestions for future research. Phase one contains studies on types of motivation factors (intrinsic, extrinsic, and internalized extrinsic) that incentivize developers to contribute their time and effort to the development of OSS. The motivation factors identified  
795 in this work warrant further research into the institutional arrangements surrounding OSS development. Phase two contains literature that begins to connect motivation and institutional arrangements in OSS development, and studies that uncover crowding effects between motivational factors, e.g., how institutions such as firms impact on motivation among OSS developers. Phase three complements work on motivation and institutional arrangements with  
800 a social practice perspective on OSS development. Here we introduce and apply the theory of social practices formulated by McIntyre (1984) and a motivational framework suggested by Martin (2000, 2002).

OSS does not exist only in competition to proprietary commercial software development. It complements it, in so far as it secures the social practices through which  
805 certain standards of excellence in software development can be further nurtured and

developed. Therefore, OSS is a social practice where professional craft and moral concern (and compensation) motivations guide practitioners in learning and innovating beyond what may possible within other institutions, such as the software firm. The internet has enabled the evolution of such social practices, where thousands of software developers can interact, learn  
810 and formulate ever higher standards of excellence.

The history of Free and OSS development demonstrates how the social practice of OSS development was born in response to institutional pressure in commercial software development, creating its own institutions that protect the virtues associated with the social practice. MacIntyre's theory applied to this phenomenon elucidates the role of individual  
815 motivations in carrying forward a social practice that in turn complements and gives rise to new institutions. Thus, understanding individual motivation in conjunction with the social practice could lead to fruitful research on institutional change beyond software. Other perspectives on the emergence of institutions (see Hargrave and Van de Ven, 2006, for a review) argue convincingly that the struggle between factions and social movements brings  
820 about change and institutional innovation. However, framing and political struggles hardly grasp the diversity of people's motivations to contribute and the role technology might play in this process. As demonstrated in the case of the Free Software Foundation, notions of quality that drive social practice to generate and maintain its standards of excellence can be powerful forces for change. The views are compatible, in that a social practice of OSS development  
825 can become a social movement (von Hippel and von Krogh, 2003; Hertel et al., 2003), and eventually an institution with the primary goal of protecting work against appropriation by third parties (O'Mahony, 2003). The views are also complementary, in that collective action emphasizes the "struggle over meanings of new issues and technologies" (Hargrave and Van de Ven, 2006: 884) across social movements, whereas social practice focuses on the quest for

830 better products, more profound knowledge, and improved collaboration in creating internal  
goods. The research questions that can best be tackled with this perspective are subsumed  
under phase three.

We have only begun to understand the surface of motivations for developers to  
contribute to OSS development. There are still vast research opportunities in phases one and  
835 two. The motivations across OSS projects could be productively investigated using self-  
determination theory. Likewise, the understanding of how institutional arrangements in OSS  
relate to developers' motivations is only beginning to emerge. However, we believe the  
greatest research opportunities are a set of research questions found in the intersection  
between social practices, institutions, and individual motivations. We welcome all to  
840 participate in this quest.

## REFERENCES

- Alexy, O., and Leitner, M. 2007. "Norms, Rewards, and their Effect on the Motivation of Open Source Software Developers," available online from <http://opensource.mit.edu>, Working Paper.
- 845 Aristotle 1941. R. McKeon, eds. *Basic Works Of Aristotle*. New York: Random House.
- Ashrafi, N. 2003. "The Impact Of Software Process Improvement On Quality: In Theory And Practice", *Information and Management* (40:7), 677-690.
- Baldwin, C. Y. 2008. "Where do transactions come from? Modularity, transactions, and the boundaries of firms," *Industrial and Corporate Change* (17:1), pp. 155-195.
- 850 Baldwin, C. Y., and Clark, K. B. 2006 "The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?" *Management Science* (52:7), pp. 1116-1127.
- Beadle, R., and Moore, G. 2006. "MacIntyre on virtue and organization," *Organization Studies* (27:3), pp. 323-340.
- 855 Benabou, R., and Tirole, J. 2003. "Intrinsic and Extrinsic Motivation," *Review of Economic Studies* (70:3), pp. 489-520.
- Benkler, Y. 2002. "Coase's Penguin, or Linux and the Nature of the Firm," *Yale Law Journal* (112:3), pp. 369-447.
- Bergquist, M., and Ljungberg, J. 2001. "The Power of Gifts: Organizing Social Relationships in Open Source Communities," *Information Systems Journal* (11:4), pp. 305-320.
- 860 Bonaccorsi, A., and Rossi, C. 2006. "Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business," *Knowledge, Technology, and Policy* (18:4), pp. 40-64.
- Bourdieu, P. 1977. *Outline of a theory of practice*, Cambridge, UK: Cambridge University Press.
- 865 Calhoun, C. 1988. "The radicalism of tradition: Community strength or venerable disguise and borrowed language?" *American Journal of Sociology* (88:5), pp. 886-924.
- Csikszentmihályi, M. 1975. *Beyond Boredom and Anxiety*, San Francisco, CA: Jossey-Bass.
- Csikszentmihályi, M. 1990. *Flow: The Psychology of Optimal Experience*, New York: Harper and Row.
- 870 David, P. A., Waterman, A., and Arora, S. 2003. "FLOSS-US: The Free/Libre/Open Source Software Survey for 2003," Technical report, SIEPR Stanford Institute for Economic and Policy Research, Stanford CA.
- Dawson, D., and Bartholomew, C. 2003. "Virtues, managers and business people: Finding a place for MacIntyre in a business context," *Journal of Business Ethics* (48:2), pp. 127-138.
- 875 Deci, E. L., and Ryan, R. M. 1985. *Intrinsic Motivation and Self-Determination in Human Behavior*, New York: Plenum.
- Deci, E. L., and Ryan, R. M. 1987 "The Support of Autonomy and the Control of Behavior," *Journal of Personality and Social Psychology* (53:6), pp. 1024-1037.
- 880 Dunne, J. 1993. *The rough ground: "Phronesis" and "Techne" in Modern Philosophy and in Aristotle*. South Bend: University of Notre Dame Press.

- Elster, J. (1986), *An Introduction to Karl Marx*, Cambridge, UK: Cambridge University Press.
- 885 Feller, J., and Fitzgerald, B. 2002. *Understanding Open Source Software Development*, London, UK: Addison-Wesley.
- Fershtman, C., and Gandal, N. 2004. "The Determinants of Output Per Contributor in Open Source Projects: An Empirical Examination," available at SSRN: <http://ssrn.com/abstract=515282>, CEPR Discussion Paper No. 4329.
- 890 Flyvbjerg, B. 2001. *Making social science matter. Why social inquiry fails and how it can succeed again*, Cambridge, UK: Cambridge University Press.
- Fortes, M. 1969. *Kinship and the social order: the legacy of Lewis Henry Morgan*, Chicago: Aldine.
- Fowers, B. J. 2003. "Reason and Human Finitude," *American Behavioral Scientist* (47:4), pp. 415-426.
- 895 Franke, N., and von Hippel, E. 2003. "Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software," *Research Policy* (32:7), pp. 1199-1215.
- Frey, B. S. 1997. *Not Just for the Money: An Economic Theory of Personal Motivation*, Edward Elgar.
- 900 Frey, B. S., and Jegen, R. 2001. "Motivation crowding theory," *Journal of Economic Surveys* (15), pp. 589-610.
- Frey, B. S., and Meier, S. 2004. "Pro-social behavior in a natural setting," *Journal of Economic Behavior and Organization* (54), pp. 65-88.
- 905 Friedman, B., and Kahn, P. 2004. "Educating computer scientists: Linking the social and the technical," *Communications of the ACM* (37:1), 64-70.
- Ghosh, R. A. 2005. "Understanding Free Software Developers: Findings from the FLOSS Study," in *Perspectives on Free and Open Source Software*, J. Feller, B. Fitzgerald, S. Hissam, and K. R. Lakhani, (eds.), Cambridge MA:MIT Press.
- 910 Ghosh, R. A., Glott, R., Krieger, B., and Robles, G. 2002. "Free/Libre and Open Source Software: Survey and Study (FLOSS)," <http://www.infonomics.nl/FLOSS/report/>
- Habermas, J. 1991. *On The Logic Of The Social Sciences*. Cambridge, MA: MIT Press.
- Haefliger, S., von Krogh, G., and Spaeth, S., 2008. "Code Reuse in Open Source Software Development," *Management Science* (54:1), pp. 180-193.
- 915 Hann, I., Roberts, J., Slaughter, S. A., and Fielding, R. 2002. "Economic Incentives for Participating in Open Source Software Project," Proceedings of the 22nd Internat. Conf. on Inform. Systems, Barcelona.
- Hargrave, T.J., Van de Ven, A.H. 2006. A collective action model of institutional innovation. *Academy of Management Review* (31:4), pp. 864-888.
- 920 Hars, A., and Ou, S. 2002. "Working for free? Motivations for participating in open-source projects," *International Journal of Electronic Commerce* (6), pp. 25-39.
- Heider, F. 1958. *The Psychology of Interpersonal Relations*, Nre York: Wiley.
- Hemetsberger, A. 2004. "When Consumers Produce on the Internet: The Relationship between Cognitive-affective, Socially-based, and Behavioral Involvement of Prosumers," available from: <http://opensource.mit.edu/papers/hemetsberger1.pdf>, Working Paper.
- 925

- Hertel, G. 2002. "Management virtueller Teams auf der Basis sozialpsychologischer Modelle," in *Sozialpsychologie Wirtschaftlicher Prozesse*, E. H. Witte (ed.), Lengerich, Germany: Pabst Publishers, pp. 172-202.
- 930 Hertel, G., Niedner, S., and Herrmann, S. 2003. "Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux Kernel," *Research Policy* (32:7), pp. 1159-1177.
- Himanen, P. 2001. *The Hacker Ethic and the Spirit of the Information Age*, London, UK: Secker and Warburg.
- 935 Klandermans, B. 1997. *The Social Psychology of Protest*, Oxford, UK: Basil Blackwell.
- Klein, E., and Herskovitz, P. 2007. "Philosophy of science underpinnings of prototype validation," *Information Systems Journal* (17:1), pp. 111-132.
- Knight, K. (ed.). 1998. *The Macintyre Reader*, Notre Dame, Illinois: University of Notre Dame Press.
- 940 Koch, S., and Schneider, G. 2002. "Effort, Cooperation and Coordination in an Open Source Software Project: GNOME," *Information Systems Journal* (12:1), pp. 27-42.
- Kohanski, D. 2000. *Moths in the Machine: The Power and Perils of Programming*, New York: St. Martin's Press.
- Krebs, D. L. 1970. "Altruism - Examination of Concept and a Review of Literature,"
- 945 *Psychological Bulletin* (73:4), pp. 258-302.
- Kuk, G. 2006. "Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List," *Management Science* (52:7), pp. 1031-1042.
- Lakhani, K. R., and von Hippel, E. 2003. "How Open Source Software Works: 'Free' User-to-User Assistance," *Research Policy* (32:6), pp. 923-943.
- 950 Lakhani, K. R., and Wolf, R. G. 2005. "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," in *Perspectives on Free and Open Source Software*, J. Feller, B. Fitzgerald, S. Hissam, and K. R. Lakhani, (eds.), Cambridge MA: MIT Press, pp. 3-22.
- Latour, B. 1996. *Aramis, or the love of technology*, Cambridge, MA: Harvard University Press.
- 955 Latour, B. 1999. *Pandora's hope: essays on the reality of science studies*, Harvard, Cambridge: Harvard University Press.
- Lattemann, C., and Stieglitz, S. 2005. "Framework for Governance in Open Source Communities," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, IEEE Computer Society, p. 192b.
- 960 Lave, J., and Wenger, E. 1991. *Situated Learning: Legitimate Peripheral Participation*, Cambridge, UK: Cambridge University Press.
- Lee, G. K., and Cole, R. E. 2003. "From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development," *Organization Science* (14:6), pp. 633-649.
- 965 Lerner, J., and Tirole, J. 2002. "Some Simple Economics of Open Source," *Journal of Industrial Economics* (50:2), pp. 197-234.
- Levy, S. 1984. *Hackers: Heroes of the Computer Revolution*, New York: Anchor Press/Doubleday.
- 970 Lim, W. C. 1994. "Effects Of Reuse On Quality, Productivity, And Economics," *IEEE*

- Software* (11:5), pp. 23-30.
- Lindenberg, S. M. 2001. "Intrinsic Motivation in a New Light," *Kyklos* (54:2/3), pp. 317-342.
- Luthiger, B., and Jungwirth, C. 2007. "Pervasive fun," *First Monday* (12: 1).
- LWN.net. 2007. "Who wrote 2.6.23?," published on September 11, 2007 by Jonathan Corbet,  
975 available at <http://lwn.net/Articles/247582/>.
- MacIntyre, A. 1984. *After Virtue: A Study in Moral Theory*, Notre Dame, Illinois: University of Notre Dame Press.
- MacKenzie, A. 2005. "The performativity of code," *Theory, Culture, and Society* (22: 1), pp. 71-92.
- 980 Markus, M. L. 2007. "The governance of free/open source software projects: monolithic, multidimensional, or configurational?," *Journal of Management and Governance* (11:2), pp. 151-163.
- Martin, M. W. 2000. *Meaningful Work: Rethinking Professional Ethics*, Oxford, UK: Oxford University Press.
- 985 Martin, M. W. 2002. "Personal meaning and ethics in engineering," *Science and Engineering Ethics* (8:4), pp. 545-560.
- Mauss, M. 1959. *The gift. The form and the reason for exchange in archaic societies*, London, UK: Routledge.
- Melucci, A. 1999. *Challenging codes: collective action in the information age*. Cambridge,  
990 UK: Cambridge University Press.
- Monteiro, E., Østerlie, T., Rolland, K., and Røyrvik, E. 2006. "Keeping it going: The Everyday Practices of Open Source Software," Norwegian University of Science and Technology, Working Paper.
- Moody, G. 2001. *Rebel Code: Linux and the Open Source Revolution*, London, UK: Penguin.
- 995 Morgeson, F. P., and Humphrey, S. E. 2006. "The Work Design Questionnaire (WDQ): developing and validating a comprehensive measure for assessing job design and the nature of work," *Journal of Applied Psychology* (91:6), pp. 1321-1339.
- Morris, A., and Mueller, C. M. (eds.). 1992. *Frontiers in social movement theory*, New Haven: Yale University Press.
- 1000 Nelson, K., Nadkarni, S., Narayanan, V., and Ghods, M. 2000. "Understanding software operations support expertise: A revealed causal mapping approach," *MIS Quarterly* (24:3), pp. 475-507.
- Noel, J. 1999. "Phronesis and Phantasia: Teaching with Wisdom and Imagination," *Journal of Philosophy of Education* (33:2), pp. 277-286.
- 1005 North, D. C. 1986. "The New Institutional Economics," *Journal of Institutional and Theoretical Economics* (142), pp. 230-237.
- O'Mahony, S. C. 2003. "Guarding the Commons: How Community Managed Software Projects Protect Their Work," *Research Policy* (32:7), pp. 1179-1198.
- O'Mahony, S., and Ferraro, F. 2007. "The Emergence of Governance in an Open Source  
1010 Community," *Academy of Management Journal* (50:5), pp. 1079-1106.
- Oliver, P. E. 1993. "Formal models of collective action," *Annual Review of Sociology*, (19), pp. v271-300.
- Olson, M. 1965. *The logic of collective action*. Cambridge, MA: Harvard University Press.
- Orlikowski, W. 2000. "Using technology and constituting structures: A practice lens for

- 1015 studying technology in organizations,” *Organization Science*, (11:4), pp. 404-428.
- Osterloh, M., and Rota, S. G. 2007. “Open Source Software Development - Just Another Case of Collective Invention?,” *Research Policy* (36:2), pp. 157-171.
- Perens, B. 1999. “The Open Source Definition,” in *Open Sources: Voices from the Open Source Revolution*, M. Stone, C. DiBona, and S. Ockman, (eds.), Sebastopol, CA: O’Reilly and Associates, Inc., pp. 171-188.
- 1020 Rämö. H. 2004. Spatio-temporal notions and environmental issues. *Organization* (11:6), pp. 849-872.
- Raymond, E. S. 1998. “Homesteading the Noosphere,” *FirstMonday* (3:10).
- Raymond, E. S. 1999. *The Cathedral and the Bazaar*, Sebastopol, CA: O’Reilly.
- 1025 Roberts, J. A., Hann, I., and Slaughter, S. A. 2006. “Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects,” *Management Science* (52:7), pp. 984-999.
- Robey, D., and Markus, M. L. 1984. “Rituals in Information System Design,” *MIS Quarterly* (8:1), pp. 5-15.
- 1030 Rullani, F. 2007. “Dragging developers towards the core. How the Free/Libre/Open Source Software community enhances developers’ contribution,” EURAM annual meeting, available from <http://opensource.mit.edu>.
- Scacchi, W. 2002. “Understanding Requirements for Developing Open Source Software Systems,” *IEE Proceedings - Software* (149:1), pp. 24-39.
- 1035 Schatzki, T.R., K. Knorr-Cetina, and E. von Savigny, (eds.). 2001. *The Practice Turn in Contemporary Theory*, London: Routledge.
- Shah, S. K. 2006. “Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development,” *Management Science* (52:7), pp. 1000-1014.
- Sheldon, K. M., and Krieger, L. S. 2007. “Understanding the Negative Effects of Legal Education on Law Students: A Longitudinal Test of Self-Determination Theory,”
- 1040 *Personality and Social Psychology Bulletin* (33:6), pp. 883-897.
- Spaeth, S., Haefliger, S., von Krogh, G., and Birgit, R. 2008. “Communal Resources in Open Source Software Development,” *Information Research* (13:1).
- Stallman, R. 1999. “The GNU Operating System and the Free Software Movement,” in *Open Sources: Voices from the Open Source Revolution*, M. Stone, C. DiBona, and S. Ockman, (eds.), Sebastopol, CA: O’Reilly and Associates, Inc., pp. 53-70.
- 1045 Stallman, R. 2002. “The GNU Project,” <http://www.gnu.org/gnu/thegnuproject.html>.
- Stewart, K. J., and Gosain, S. 2006. “The Impact of Ideology on Effectiveness in Open Source Software Development Teams,” *MIS Quarterly* (30:2), pp. 291-314.
- 1050 Stewart, D. 2005. “Social Status in an Open-Source Community,” *American Sociological Review* (70:5), pp. 823-842.
- Stewart, K. J., Ammeter, A. P., and Maruping, L. M. 2006. “Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects,” *Information Systems Research* (17:2), pp. 126-144.
- 1055 Suchman, L., Blomberg, J., Orr, J. E. and Trigg, R. 1999. “Reconstructing technologies a social practice,” *American behavioral Scientist* (43:3), pp. 392-408.
- Torvalds, L. and Diamond, D. 2001. *Just for Fun*, London, UK: Texere.
- Ulhoi, J. P. 2004. “Open Source Development: a Hybrid in Innovation and Management

- Theory,” *Management Decision* (42), pp. 1095-1114.
- 1060 Van den Burg, S., and van Borg, A. 2005. “Understanding moral responsibility in the design of trailers,” *Science and Engineering Ethics* (11 :2), pp. 235-256.
- von Hippel, E. 1998. “Economics of product development by users: The impact of ‘sticky’ local information,” *Management Science* (44:5), pp. 629-644.
- 1065 von Hippel, E., and von Krogh, G. 2003. “Open Source Software and the ‘Private-Collective’ Innovation Model: Issues for Organization Science,” *Organization Science* (14:2), pp. 209-223.
- von Krogh, G. 1998. “Care in knowledge creation,” *California Management Review* (49:3), 133-153.
- 1070 von Krogh, G., Spaeth, S., and Lakhani, K. 2003. “Community, Joining, and Specialization in Open Source Software Innovation: A Case Study,” *Research Policy* (32:7), pp. 1217-1241.
- von Krogh, G., and von Hippel, E. 2006. “The Promise of Research on Open Source Software,” *Management Science* (52:7), pp. 975-983.
- 1075 West, J., and O’Mahony, S. C. 2005. “Contrasting Community Building in Sponsored and Community Founded Open Source Projects,” in *Proceedings of the 38th Annual Hawaii International conference on System Sciences (Jan 2005)*.
- Wu, C., Gerlach, J. H., and Young, C. E. 2007. “An Empirical Analysis of Open Source Software Developers’ Motivations and Continuance Intentions,” *Information and Management* (44:3), pp. 253-262.
- 1080 Yamauchi, Y., Yokozawa, M., Shinohara, T., and Ishida, T. 2000. “Collaboration with Lean Media: How Open-Source Software Succeeds,” ACM Conference on Computer Supported Cooperative Work. Philadelphia, PA.
- Ye, Y., and Kishida, K. 2003. “Toward an Understanding of the Motivation of Open Source Software Developers,” *Proceedings of 2003 International Conference on Software Engineering (ICSE2003)*, Portland, Oregon, pp. 419-429.
- 1085 Zeitlyn, D. 2003. “Gift economies in the development of open source software: anthropological reflections,” *Research Policy* (32:7), pp. 1287-1291.

## APPENDIX

**Table 1. Intrinsic Motivation**

<i>Motivation</i>	<i>Concept</i>	<i>Key empirical findings</i>
Ideology	Ideology has been quoted as a major reason for starting the GNU project, one of the earliest coherent attempts to write software under an explicitly open license (Stallman, 1999). The extent to which contributors adhere to ideology is usually captured by items such as: “software should be free for all,” “free to modify and redistribute,” or that “open source code should replace proprietary software.”	<ul style="list-style-type: none"> <li>•Support for ideological motives (Ghosh, 2005; David et al., 2003; Lakhani and Wolf, 2005).</li> <li>•Weak support (Hemetsberger, 2004).</li> <li>•Hertel et al. (2003) found a positive, significant relationship between social and political motives with accepted source code patches and lines of code contributions.</li> <li>•Stewart and Gosain (2006) found that open source developers’ adherence to the community ideology (defined as “open source” norms, values, and beliefs) impacts on team effectiveness.</li> </ul>
Altruism	Altruism is the selfless concern for the welfare of others. A typical altruistic act consists of three characteristics: “a) it is an end in itself; it is not directed at gain, b) is emitted voluntarily, and c) does good” (Heider, 1958 in Krebs, 1970, p.259). Due to the self-containment of an altruistic act, it fits well with the category of intrinsic motivation, and several authors have used the concept of altruism to explain code contribution of OSS developers.	<p>Osterloh and Rota (2007) suggested that altruistic behavior caused by “pro-social motives” influences developers to contribute to OSS development. The “pro-social motive” is a type of intrinsic motivation (Lindenberg, 2001), which the authors link to open source contributions.</p> <ul style="list-style-type: none"> <li>•Wu et al. (2007) investigated the intention of OSS developers to continue their involvement in future projects. Their structural equation model shows that altruism in the form of helping behavior influences the developers’ continuance, only if mediated by the developers’ satisfaction.</li> <li>•Hemetsberger (2004) reported that 22% of developers ranked altruism as a motivational factor to contribute. Hemetsberger also attempted to differentiate between types of developers and found that the importance of altruism in explaining contributions is stronger for people who contribute a lot (30.7%), compared to medium contributors (23.9%) and low contributors (6.5%).</li> <li>•Hars and Ou (2002) reported that altruism motivated developers to contribute. 16.5% of the survey participants rated high on altruism. Student and hobby programmers rated altruism the highest at 24.2%, followed by salaried and contract programmers at 11.1%, whereas only 7.7% of the programmers paid for open source development were driven by altruistic motivations.</li> <li>•Selfish behavior to such a degree as to rule out altruistic behavior as an important characteristic of OSS development (Ghosh, 2005).</li> </ul>
Kinship amity	The concept of kinship amity (Fortes, 1969) has been related to the concept of the gift economy (Zeitlyn, 2003). However, kinship amity differs from the gift economy since the former does not assume reciprocity in social relations. For example, in families (kin) there is no calculated economic relationship. Kinship amity thus also differs from altruism as a motive to contribute, because it is restricted to the group to which one	<ul style="list-style-type: none"> <li>•Zeitlyn (2003) first suggested kinship amity as a motive in OSS and as an explanation for why people contribute to OSS.</li> <li>•Hemetsberger (2004), using concepts such as “group boundaries” and “group bonds,” found a weak relationship between kinship amity and developers’ level of contribution.</li> <li>•Lakhani and Wolf’s survey (2005) identified kinship amity as an important motive for contributing, and showed it is an important determinant of the effort invested (hours per week).</li> <li>•Hars and Ou (2002), studying “community identification,” found a correlation between kinship amity and the number of hours per week spent on OSS contribution.</li> <li>•Hertel et al. (2003) tested the relationship between kinship amity and the number of accepted</li> </ul>

	<p>belongs, such as the OSS community. In our review, some equivalent constructs are subsumed under kinship amity, for example, the frequently cited motivation for OSS contributors, community identification (e.g., Hars and Ou, 2002). Community identification instills a feeling of belonging to a certain group, and urges people to help others in that group.</p>	<p>patches and lines of code, and found it to be positive and significant.</p>
<p>Enjoyment and fun</p>	<p>Enjoyment and fun have been said to motivate contributors to open source projects. One of the main drivers of the so called “hacker culture” emerging during the 1980s was for developers to enjoy the playfulness and experimentation with hardware and software (Levy, 1984; Torvalds and Diamond, 2001)</p>	<ul style="list-style-type: none"> <li>•Benkler (2002) and Osterloh and Rota (2007) suggested that enjoyment plays an important role in OSS.</li> <li>•Lakhani and von Hippel (2003) showed in their survey that developers considered enjoyment and fun important when conducting technically challenging tasks, whereas mundane tasks, such as helping users to install software, required different motives.</li> <li>•Luthiger and Jungwirth (2007) conducted the most comprehensive study focusing exclusively on fun and enjoyment motivations. Their survey of 1,330 open source developers revealed that the fun factor had a significantly positive effect on both the number of hours spent on a project as well as on developers’ intention to participate in the future. Fun accounted for 28% of the effort in terms of number of hours dedicated to projects.</li> <li>•Lakhani and Wolf (2005) found that developers deemed enjoyment-based motivation an important source of motivation. In their research, high levels of enjoyment also increased the hours per week that developers spent on a project.</li> <li>•Hertel et al. (2003), measuring the number of accepted patches and lines of code in the Linux project, also found a significant positive impact of fun and enjoyment.</li> <li>•Hemetsberger’s (2004) survey identified modest positive impact of enjoyment on contributions to projects.</li> <li>•Roberts et al. (2006) could not identify a significant link between intrinsic enjoyment and the number of accepted patches and lines of code.</li> </ul>

Table 2. Internalized Extrinsic Motivation

Motivation	Concept	Key empirical findings
Reputation	<p>Based on the literature, reputation can be classified as “peer reputation” and “outside reputation.” Peer reputation is usually targeted to community insiders (peers, or kin) and potential employers who perceive peer reputation to signal potential talent. Very few studies have considered reputation directed to the outside of the community and not targeted towards potential future employers. Outside reputation is concerned with the expected reactions and prestige awarded to the contributors, by significant others such as friends or relatives.</p>	<ul style="list-style-type: none"> <li>•Raymond’s (1998) essay “Homesteading the Noosphere” linked reputation to reciprocity in the gift economy and described it as the “major motivation” for developers.</li> <li>•Stewart (2005) and Lerner and Tirole (2002) proposed peer reputation as a fundamental motivation.</li> <li>•Osterloh and Rota (2007) termed this motivation “ego gratification,” which could easily be confused with intrinsically motivated self-determination (the happiness of having/being able to achieve something, see Deci and Ryan, 1987). However, the authors classify it as an extrinsic signaling incentive, aimed at increasing one’s own labor market value.</li> <li>•Lakhani and von Hippel (2003) differentiated peer reputation further. They proposed that peer reputation motivates “gratifying” technical tasks, while it fails to motivate the “necessary but mundane tasks” that are an inherent part of each software project.</li> <li>•Lattemann and Stieglitz (2005) proposed that contributors’ roles are related to motivations. In their view, programmers (rather than bug fixers, or managers) were motivated through peer reputation.</li> <li>•Spaeth et al. (2008) argued that some motives are formed as by-products of contributions. In their empirical study of the Freenet project, the authors found that higher levels of contributions provided more peer reputation, such as positive mentioning in e-mail lists.</li> <li>•The surveys by Lakhani and Wolf (2005), Hemetsberger (2004), Hars and Ou (2002), Ghosh (2005) reported peer reputation as an driver for participation.</li> <li>•Hars and Ou (2002) identified a weak but existing relationship between reputation and the number of hours invested.</li> <li>•Lakhani and Wolf (2005) found peer reputation to be the fourth biggest determinant of invested effort.</li> <li>•Roberts et al. (2006) measured the accepted lines of code. They identified a significant positive relationship between peer reputation motives and accepted code in the Apache project.</li> <li>•Stewart (2005), who examined an online community dedicated to free software developers, found that a newcomer’s reputation tends to stabilize some time after joining. To date, this is the only study that has examined the dynamics of developers’ reputation in OSS development projects.</li> <li>•Hemetsberger (2004) found a weak relationship between outside reputation and participation.</li> <li>•Hertel et al. (2003) tested the impact of outside reputation on the number of accepted patches and lines of code. <i>Ceteris paribus</i> they found a significant positive impact of outside motivation on accepted code.</li> </ul>
Gift	Originally a concept from anthropology (Mauss,	•Bergquist and Ljungberg (2001) suggested reciprocity as a motivation for contributions to OSS.

economy/ Reciprocity	1959), several authors discussed the logic of gift giving in the context of OSS development (Raymond, 1999; Bergquist and Ljungberg, 2001; Zeitlyn, 2003). Viewing OSS development as a gift economy asserts that developers give code to others expecting gifts in return. The corresponding internalized, extrinsic motivation can be termed “reciprocity.”	<ul style="list-style-type: none"> <li>•Hemetsberger (2004) and Lakhani and Wolf (2005) confirmed reciprocity in empirical studies that found moderate support, while David et al. (2003) found strong support.</li> <li>•In their survey, Lakhani and von Hippel (2003) found that reciprocity motivated developers to perform mundane tasks. It seems that if people have been helped by other contributors in the past, they are more inclined to reciprocate as they gain experience and knowledge.</li> </ul>
Learning	The motive to acquire new skills or to learn in OSS development appears in almost every contribution to this review article. However, the definition of learning was often vague and referred to survey items such as “improve programming skills” (the opportunity to learn from the experience of writing software and the feedback provided by the peers who tested, integrated, and commented upon the software published). As a single motivation, learning—without a specification of subject and object—runs the danger of oversimplifying or making empirical results non-comparable.	<ul style="list-style-type: none"> <li>•Survey studies confirmed that learning motivated individuals to participate in OSS development (Ghosh, 2005; Hemetsberger, 2004; Lakhani and Wolf, 2005), with particularly strong results in the two studies by David et al. (2003) and Hars and Ou (2002).</li> <li>•Wu et al. (2007) found that learning motives led to the intention to participate.</li> <li>•In the survey by Robert et al. (2006), the effort spent on developing OSS could be explained by learning and the accepted patches and lines of code written were (significantly) positively impacted by learning.</li> <li>•Ye and Kishida (2003) suggested the consideration of legitimate peripheral learning based on the work by Lave and Wenger (1991) to explain increasing levels of participation over time (see also Rullani, 2007).</li> </ul>
Own use value	Own-use value refers to internalized extrinsic motives to create OSS for contributors’ personal use.	<ul style="list-style-type: none"> <li>•Raymond (1999), Lakhani and von Hippel (2003), and Osterloh and Rota (2007) suggested developers of OSS “scratch their itch” by developing software they find useful, by fixing bugs, and by adding features they need.</li> <li>•Surveys by Ghosh (2005), David et al. (2003), Hars and Ou (2002), Lakhani and Wolf (2005), as well as Hemetsberger (2004) identified own-use value as a motive for participating in the development of OSS.</li> <li>•Wu et al. (2007) found own-use value was connected to the intention to participate.</li> <li>•Lakhani and von Hippel (2003) identified own-use value as a motive for taking on mundane tasks.</li> <li>•Hars and Ou (2002) reported that developers attributed a high score to own-use value as their motive regarding actual effort measured in hours spend per week.</li> <li>•Hertel et al. (2003) reported that own-use value had a significant effect on accepted patches and lines of code contributed.</li> <li>•Roberts et al. (2006) reported that own-use value exerted a significant <i>negative</i> impact on the level of participation in the Apache project, also measured in accepted patches and lines of code. One explanation offered is that developers motivated by own-use value worked “eclectically”: they would fix bugs that annoyed them and then leave the development again, rather than remaining as long-term developers. This behavior would result in relatively low total contributions to one project.</li> </ul>

- |  |  |  |
|--|--|--|
|  |  | <ul style="list-style-type: none"><li>•Lattemann and Stieglitz (2005) proposed that own-use value may impact on OSS development via the roles individuals assume in communities. Contributors who mainly fix bugs may be particularly motivated by own-use value, whereas others such as managers (maintainers), might be more motivated by pay.</li></ul> |
|--|--|--|

**Table 3. Extrinsic Motivation**

<i>Motivation</i>	<i>Concept</i>	<i>Key empirical findings</i>
Career	Lerner and Tirole (2002) first suggested studying the signaling behavior of OSS developers. Their proposition, derived from economic literature, stated that individual developers would be motivated by career concerns when developing OSS. By publishing software that was free for all to inspect, they could signal their talent to potential employers and thus increase their value in the labor market.	<ul style="list-style-type: none"> <li>•Lakhani and Wolf (2005) and Hemetsberger (2004) found weak support for career concerns as motivation.</li> <li>•Hars and Ou (2002) and Ghosh (2005) found more substantial support for career concerns as motivation.</li> <li>• Wu et al. (2007) found career concerns related to intended participation.</li> <li>•Hars and Ou (2002) found strong support of the motivation for efforts measured in hours per week spent on OSS development.</li> <li>•Roberts et al. (2006) and Hertel et al. (2003) documented a positive and significant relationship between career concerns, accepted code patches and lines of code.</li> <li>•Hars and Ou (2002) report that career concerns played a more important role for paid participation in OSS development than for unpaid participation.</li> </ul>
Pay	A significant minority (approximately 40 %) of contributors are paid to participate in OSS projects (Lakhani and Wolf, 2005). An examination of contributions to the Linux kernel found only 9% of the developers involved worked in their own time (LWN.net, 2007).	<ul style="list-style-type: none"> <li>•Lakhani and Wolf (2005) examined the degree of participation and its link to financial motives. They concluded that the financial subsidy of these projects was substantial. For example, paid contributors dedicated 17.7 hours per week on all FOSS projects they participated in, while volunteers contributed 11.7 hours per week. As programmers often participate in several projects simultaneously, Lakhani and Wolf reported the results for the focal project of the programmers as well. These results showed a similar pattern: 10.3 hour per week for the paid contributor and 5.7 hours per week for the volunteer. The differences between the groups were found to be significant.</li> <li>•Surveys by Hars and Ou (2002), Hertel et al. (2003), and Luthiger and Jungwirth (2007) reported findings consistent with Lakhani and Wolf (2005).</li> <li>•Dahlander and Wallin (2006) found that paid developers contribute more to development discussions in terms of emails sent than unpaid developers.</li> </ul>

**Table 4. Institutional Arrangements**

<i>Institutional arrangement</i>	<i>Concept</i>	<i>Key references</i>
Governance	“The means of achieving the direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of an OSS development project to which they jointly contribute” (Markus, 2007: 152).	<ul style="list-style-type: none"> <li>•Markus, 2007</li> <li>•Shah, 2006</li> <li>•O’Mahony and Ferraro, 2007</li> <li>•Spaeth et al., 2008</li> </ul>
Community sponsorship	Sponsorship can refer to control by an organization, such as a firm, over the development process, source code accessibility, or code ownership.	<ul style="list-style-type: none"> <li>•Stewart et al., 2006</li> <li>•Shah, 2006</li> </ul>
License restriction	Choice of open source license: due to its viral properties, the GNU GPL is considered the most restrictive license. Restriction is usually thought of as limiting the commercial exploitation of the source code, i.e. the option to combine OSS with proprietary software for sale.	<ul style="list-style-type: none"> <li>•Stewart et al., 2006</li> <li>•Fershtman and Gandal, 2004</li> </ul>
Provision of rewards	Incentive structure and level of income.	<ul style="list-style-type: none"> <li>•Roberts et al., 2006</li> <li>•Alexy and Leitner, 2007</li> <li>•Hann et al., 2002</li> <li>•Lerner and Tirole, 2002</li> </ul>
Infrastructure	Information technology, including low-cost access to bandwidth, versioning tools, mailing lists, etc.	<ul style="list-style-type: none"> <li>•Lee and Cole, 2003</li> <li>•von Hippel and von Krogh, 2003</li> </ul>
Code architecture and design	Technical design and modularity of the code base.	<ul style="list-style-type: none"> <li>•Franke and von Hippel, 2003</li> <li>•Baldwin and Clark, 2006</li> </ul>
Social and technical exposure to the community	Exposure over time creates opportunities for advancement and work on more challenging tasks that require deep insights into the community’s workings.	<ul style="list-style-type: none"> <li>•Rullani, 2007</li> <li>•von Krogh et al., 2003</li> <li>•O’Mahony and Ferraro, 2007</li> <li>•Shah, 2006</li> </ul>

**Table 5. Outline of the Motivation-Context Matrix**

		Individual		
		Craft motives	Compensation motives	Moral concerns
Collective	Social practice			
	Goods			
	Institutions			

**Table 6. Motivation-Context Matrix: Social Practice**

	Craft motives	Compensation motives	Moral concerns
Social practice	<ul style="list-style-type: none"> <li>•How do newcomers learn the standards of excellence in the social practice?</li> <li>•How does the social practice generate and alter craft motives?</li> <li>•Is producing excellent software code sufficient to adhere to the standards of excellence within the social practice?</li> </ul>	<ul style="list-style-type: none"> <li>•What effect has financial compensation on participation in the social practice?</li> <li>•Does compensation impact on the standards of excellence of the social practice?</li> <li>•Do compensation motives and craft values reinforce or diminish each other?</li> <li>•Does exposure to the social practice change the perception of compensation over time?</li> </ul>	<ul style="list-style-type: none"> <li>•What are the dimensions of moral concerns within the social practice?</li> <li>•Do moral considerations result from or precede participation in a social practice?</li> <li>•How are conflicts of interest between members of the social practice and other stakeholders resolved?</li> <li>•Are moral obligations a strong factor impacting on behavior within the social practice (i.e., this is how things are done)?</li> </ul>

**Table 7. Motivation-Context Matrix: Internal and External Goods**

	Craft motives	Compensation motives	Moral concerns
Internal and external goods	<ul style="list-style-type: none"> <li>•How does the software to be created (the development agenda) impact on contributors' craft motives?</li> <li>•Vice versa, how do craft motives affect the software development agenda?</li> <li>•How do individual perceptions of quality relate to the quality of goods produced by the community?</li> <li>•What is the relationship between (technical) quality and other quality elements of goods produced?</li> </ul>	<ul style="list-style-type: none"> <li>•What are the potential conflicts between the motivation for individual compensation and the quality, or standard of quality, of the goods produced?</li> <li>•How does the use value depend on the characteristic and quality of goods produced?</li> </ul>	<ul style="list-style-type: none"> <li>•How is the type of internal goods produced through helping other individuals different from other contributions?</li> <li>•How and why do certain types of goods elicit or allay moral concerns?</li> <li>•What are the potential conflicts between moral concerns and the production and consumption of external goods?</li> </ul>

**Table 8. Motivation-Context Matrix: Institutions**

	Craft motives	Compensation motives	Moral concerns
Institutions	<ul style="list-style-type: none"> <li>•When do craft motives warrant the foundation of new institutions?</li> <li>•How can institutions support rather than corrupt craft motives?</li> <li>•How can craft motives be distinguished from framing strategies to determine the direction of institutional change?</li> <li>•Which institutional characteristics, such as governance structures, are best compatible with craft motives?</li> </ul>	<ul style="list-style-type: none"> <li>•How can compensation structures be designed to remain compatible with other motives?</li> <li>•How can institutions balance financial and non-financial compensations?</li> <li>•How can competing institutions maintain long-term relationships with volunteers?</li> <li>•How can firms and communities interact, given radically different compensation structures for members?</li> </ul>	<ul style="list-style-type: none"> <li>•Which institutional mechanisms foster integration and respect for their members' moral concerns?</li> <li>•What is the role of moral concern in establishing and sustaining institutions?</li> <li>•How can conflicts between moral concerns and institutions be resolved?</li> <li>•How do institutions compare when catering to moral concerns?</li> </ul>