Communications of the Association for Information Systems

Volume 27 Article 30

10-2010

Understanding Developers' Motives in Open Source Projects: A Multi-Theoretical Framework

Hind Benbya
Department of Information Systems, GSCM-Montpellier Business School, h.benbya@supco-montpellier.fr

Nassim Belbaly
Department of Information Systems, GSCM-Montpellier Business School

Recommended Citation
Benbya, Hind and Belbaly, Nassim (2010) "Understanding Developers' Motives in Open Source Projects: A Multi-Theoretical Framework," *Communications of the Association for Information Systems*: Vol. 27 , Article 30.
DOI: 10.17705/1CAIS.02730
Available at: https://aisel.aisnet.org/cais/vol27/iss1/30

Communications of the Association for Information Systems

CAIS

Understanding Developers' Motives in Open Source Projects: A Multi-Theoretical Framework

Hind Benbya

Department of Information Systems, GSCM-Montpellier Business School

Benbya@gmail.com; h.benbya@supco-montpellier.fr

Nassim Belbaly

Department of Information Systems, GSCM-Montpellier Business School

Abstract:

The question of why so many developers dedicate time and effort into contributing to Open Source Projects (OSP) is one of the most intriguing questions in OS research. Several preliminary studies have theorized about and empirically examined this challenging question. They suggest a variety of reasons to explain this phenomenon but mostly rely on self-determination theory, with its extrinsic-intrinsic dichotomy, to explain the heterogeneous and complex nature of motivation in OSS. This article provides an alternative, theory-driven approach, whereby three different, yet complementary, theories of motivation are combined, to explain the participation and outcomes of developers in OSP. More specifically, our multi-theoretical framework is based on social exchange theory, goal-orientation, and expectancy theory. An empirical test of the model is provided within the context of SourceForge.net. The results offer new theoretical and practical insights into developers' motivation and how it affects their participation and outcomes.

Keywords: Open Source Software, motivation, incentives, participation, multi-theory, goal-orientation, social-exchange, expectancy, outcomes

Volume 27, Article 30, pp. 589-610, October 2010

Communications of the Association for Information Systems

I. INTRODUCTION

Over the past decade, there has been a phenomenal increase in the adoption of Open-Source Software (OSS) by both firms and governments. OS is largely recognized today as an alternative way of developing and distributing software of high quality at relatively lower costs when compared with proprietary approaches. The majority of OSP participants, however, remain volunteers who supply their work for free, and many of them agree to have their contributions licensed in such a way that is difficult for them to profit directly from the resulting software product.

This puzzling phenomenon of developers' contributions to OSP without a clear return on their invested time and effort has energized considerable conceptual and empirical research [for a review see Bonaccorsi and Rossi, 2006; Benbya and Belbaly, 2011]. Some researchers have proposed that developers participate to gain selective, transactional benefits such as career opportunities, reputation, and status [Lerner and Tirole, 2002; Hertel et al., 2003]. Others, in contrast, find that participation in OSP is mainly driven by altruism and ideology [Stewart et al., 2006; Bagozzi and Dholakia, 2006]. Recent empirical findings strongly suggest that the functioning of these systems is driven by mixed and heterogeneous motivations [e.g., Shah, 2006; Roberts et al., 2006]. Consequently, optimizing on only one dimension might have the effect of limiting participation. While these studies provide a good understanding of some underlying rationale for voluntary participation, there are still some limitations in the existing literature.

First, most empirical research to date relies mainly on self-determination theory (SDT) and its extrinsic-intrinsic dichotomy to explain the complex and heterogeneous nature of OSS motivation [e.g., Hars and Ou, 2002; Lakhani and Wolf, 2005; Bitzer, 2005; Shah, 2006; Roberts et al., 2006; Ke and Zhang, 2009]. While these studies find that both intrinsic and extrinsic motivational components are important, evidence on their relative value is mixed, and this taxonomy may not sufficiently explain the multifaceted and complex nature of developer motivation [Krishnamurthy, 2006]. Investigating how different, but complementary, motivation theories—other than SDT—together affect participation could offer novel insights into the complex relationships between developers' motives and shed critical light on the result of previous studies.

A second limitation of the existing literature is that we know very little about how the motivational factors identified affect participation. Previous empirical studies investigating the relationship between motivation and participation in OSS focus almost exclusively on the extent of participation (*effort)* [e.g., Hars and Ou, 2002; Hertel et al., 2003; Ke and Zhang, 2009]. The *type* of participation, however, has been largely ignored. To the best of our knowledge, there is no prior study examining how individual motivation relates both to the *type* and *extent* (*effort)* of participation. Understanding not only why developers participate, but also how they participate, can help bring more clarity to the OSS motivation black box.

Finally, there is a lack of research examining the effect of motivation on participation outcomes. Several outcomes (e.g., learning, career benefits, performance) have been suggested to result from developer participation in OSP. However, with the exception of the study of Roberts et al. [2006], which analyzes motivation in relation to participation and its effect on individual ranking progress (status), most research to date considers motivation only in relation to participation (*effort),* and very rarely in relation to its effects.

The present study addresses these shortfalls in the literature by drawing on three complementary motivational theories and examining how together they explain the participation and outcomes of developers in OSP. We seek to investigate the following research question from an empirical perspective: How do differences in motivation affect the extent and type of participation in OSP and its outcomes?

To address this question: Based on an analysis of motivation theories and findings from previous studies on motivation in OSP, we develop a multi-theoretical framework in which social exchange theory, expectancy theory, and goal-orientation are combined to explain participation (both *type* and *extent)* and how that framework relates to two outcomes (learning and satisfaction). These theories are selected because of their relevance to the OSS context and applicability to provide better explanations of developers' motives than could be achieved by each theory applied separately. The research model is tested empirically through data collected from OSP hosted on SourceForge.net.

In the following sections of this article, we first present our review of the literature (Section II) and describe our conceptual model (Section III). We then describe our research sample and methods (Section IV). Next, we report our findings with respect to developers' motivation, participation levels, and outcomes (Section V). Finally, we discuss both the theoretical and practical implications of our findings (Section VI).

II. REVIEW OF THE LITERATURE

To develop our multi-theoretical framework, we started with an analysis of existing motivation theories in order to identify potential frameworks that can fit with the OSS context. We then discuss findings from previous studies on OSP motivation and map them to categories of motives relevant to OSS.

Motivation Theories Relevant to the OSS Context

There are a variety of constructs posited by motivation theorists to explain how motivation influences choice, persistence, and performance. Our objective in this phase was not to provide an exhaustive or systematic review of the literature on motivation theories. The scientific study of motivation began in 1930s and arose from varied traditions, resulting in several theories [e.g., Ambrose and Kulik, 1999]. Rather, our aim was to identify from published reviews major motivation theories relevant to the OSS context and be able to map them later to categories of motives that fit OSS developers. Central to our analysis of motivation reviews is a focus on theories used to explain autonomous work and task-related motivation. Autonomy involves acting with a sense of having a choice and is characteristic of OSS communities in which the locus of control and management lies with the individual participants who decide for themselves the terms of interaction, self-select tasks, and make contributions that benefit others. The resulting analysis reveals the existence of several motivation theories other than SDT (Table 1). Among these theories, goal-orientation, expectancy, and social exchange seem particularly relevant to the OSS context and can provide useful insights into developers' participation. They view people as naturally inclined to act on their inner and outer environments, engage in activities that interest them, and move toward personal and interpersonal coherence. By contrast, reinforcement theories view people as passively waiting for disequilibrium, that is, they have to be pushed or prodded to act. Consequently, they do not fit with the OSS context in which participants are mostly volunteers who participate with no promise of a direct financial reward for their efforts. We briefly describe these theories in Table 1 and discuss how they underpin our work in the following section.

Table 1: Motivation Theories Definition

Motivation Theories	Definition	Author
Goal-orientation	Goal theory is based on the premise that people are motivated to reach goals. They will consequently direct their behavior in pursuit of these goals.	Nicholls [1984]
Expectancy	Expectancy theory views behavior as purposeful and is largely based on conscious intentions. When applied to the workplace, it considers employees to rationally evaluate various on-the-job work behaviors (e.g., working harder) and then choose those they believe will lead to their most valued work-related rewards and outcomes (e.g., a promotion).	Vroom [1964]
Self-determination (SDT)	Self-determination theory differentiates between intrinsic and extrinsic motivation. Intrinsically motivated behaviors are those that are freely engaged out of interest. Extrinsic motivation, in contrast, requires an instrumentality between the activity and some separable consequences, such as tangible or verbal rewards	Deci and Ryan [1980]
Social Exchange	Social exchange theory proposes that social behavior is the result of an exchange process between parties. The motivation behind social exchanges is considered as a process of cost-benefit analyses in which people make decisions based on their individual satisfaction level within the relationship.	Blau [1964]
Reinforcement	Reinforcement theory is concerned with controlling behavior to increase the probability of a consequence to occur in the future.	Skinner [1953]

An Analysis of Motivation in OSS

The purpose of this review was twofold. First, to summarize what we know about the motivation of developers in OSP. Second, to identify major approaches and theories used to study OSS motivation, in relation to participation and its outcomes, to extend our knowledge on the unique aspects of OSP. While our objective is not to perform an exhaustive review or meta-analysis of a specific domain of enquiry, it is necessary to examine as much related literature as possible to synthesize the findings of empirical research on OSS motivation.

To find published research on OSS motivation, full-text searches in numerous online databases (EBSCO Host, ABI Inform) were performed using multiple keywords, such as "motivation in open source," "open source communities motivation," "open source incentives," etc. Our primary searches for literature focused on journals within the IS discipline. However, since the topic of open source motivation has been of interest to many disciplines (especially in economics), we did not restrict our analysis to IS journals. Special issues on OSS were also examined to ensure that applicable studies were included. A total of forty-five empirical and conceptual papers were identified. These papers were published between 2002 and 2009. From this collection of papers, only papers reporting *empirical* results (both quantitative and qualitative research) of interrelationships among motivations and participation and/or outcomes are included for further analysis; this yielded a total of eighteen papers. Appendix 1 summarizes the main results of these studies.

First, empirical studies on OSS motivation have identified a variety of reasons for developers' participation. OSS participants are sometimes motivated by self-interest, for example, seeking to enhance their reputation, or to gain other personal benefits [Lerner and Tirole, 2002; von Hippel and von Krogh, 2003]. At other times, participants act selflessly and appear to be motivated by altruism [Hars and Ou, 2002], or, they identify closely with an OSS community, feeling obligation to other community members [Stewart and Gosain, 2006].

Although researchers have identified a variety of reasons to explain developers' participation in OSP, these motives generally fall under two categories. The first category is based on traditional economic theory and reflects self-interested behavior and developers' desires to increase their future potential earnings (e.g., learning, career benefits). The second category is based on the literature on social movement and gift economies and reflects developers' interest to engage in cooperative behaviors and contribute for the benefits of others (e.g., reciprocity, ideology).

To summarize both dimensions and explain the heterogeneous nature of motivation in OSS, researchers have generally relied on self-determination theory (SDT) with its extrinsic/intrinsic dichotomy. However, evidence on the relative value of these dimensions is mixed and this taxonomy may not be sufficient to explain the multifaceted nature of developer motivation [Krishnamurthy, 2006; Freeman, 2007]. For example, Lakhani and Wolf [2005] find intrinsic motivation in the form of enjoyment to be the main driver of participation in OSP, while Hars and Ou [2001] report that extrinsic motivation (expected future returns and personal needs for software) plays a greater role in explaining participation. By taking a multi-theoretical perspective, we recognize that participation in OSP is driven by heterogeneous motives and that no single theoretical framework can provide a complete explanation. Instead, different motivation theories may be better at explaining participation and the degree to which it affects their outcomes.

III. A MULTI-THEORETICAL FRAMEWORK OF DEVELOPERS' MOTIVATION IN OSS

Drawing on three distinct motivational theories—goal-orientation, expectancy, and social exchange—we develop a research model to investigate the black box of motivation's effects on participation and its subsequent outcomes (Figure 1). Each of these theories is well suited for understanding a specific dimension of motivation in OSP. First, we use goal-orientation and expectancy theories to account for the rational and conscious choices of developers to contribute to OSP. Goal-orientation distinguishes between two goals: learning and performance [Dweck and Legget, 1988]. Learning goals involve personal enhancement, while performance goals are similar to ego-involved goals and focus on developers desires to demonstrate their level of ability.

Apart from their learning and performance goals, developers' participation in OSP can be related to other expected benefits (Appendix 1). Specifically, "private rewards" (more likely to accrue to individuals who actively participate) are mainly related to career benefits and professional effectiveness. To address these professional expectations not accounted for by goal-orientation, we use expectancy theory [Vroom, 1964]. Expectancy theory suggests that a person's expectation of specific outcomes (expectancy) and the attractiveness of those outcomes (valence) will affect their participation levels and outcomes.

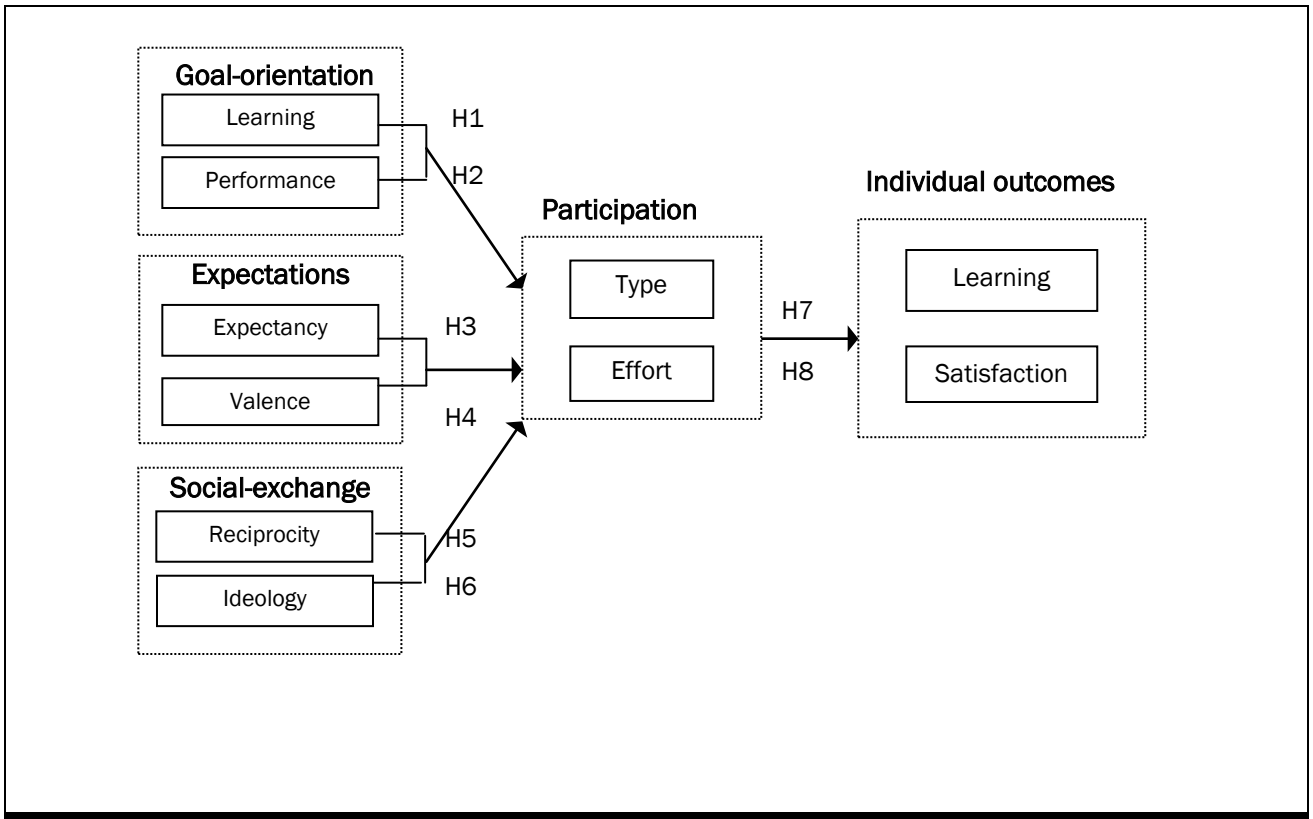

Figure 1. A Multi-Theoretical Framework of Developer Motivation in OSS

In the second section of the model, we use social exchange theory to better understand why developers contribute their own valuable knowledge and put effort into the activities of OSP for the benefit of others. Specifically, two main dimensions can be associated with social exchange: reciprocity and ideology. Reciprocity suggests that developers are involved in an exchange relationship because they expect their contributions to be reciprocated [Gouldner, 1960]. Ideology indicates developers' affiliation to a particular social group together with some emotional or value significance to him of this group membership [Tajfel, 1972].

Figure 1 depicts the research model. Specifically, It details how developers' differences in goal-orientation (learning versus performance), expectations (expectancy and valence), along with pro-social behavior (reciprocity and ideology), act jointly to influence participation (type and extent) and two of its outcomes (learning and satisfaction). Next, we develop the specific hypotheses that form the basis of our model. We then provide a test to the individual hypotheses of the model using survey data collected in the second phase of the study.

Goal Orientation Theory

Goal orientation is a social cognitive theory that holds that variations in behavior are not necessarily a result of high or low absolute amounts of motivation, but are a manifestation of the qualitatively different goals adopted by individuals [Roberts, 1992, 2001]. Goal-orientation suggests that individuals adopt goals that will most closely reflect their cognitive beliefs about what is required to maximize achievement in that particular context. The original theory and research by Nicholls [1984], Dweck and Elliot [1983], Dweck and Leggett [1988] distinguish between two types of goals: *Learning* versus *Performance*.

In the first, *learning goal orientation*, individuals are concerned with increasing their competence and the acquisition of new skills. In the other category, *performance goal orientation*, individuals are focused on the demonstration and verification of their ability that can be achieved by seeking favorable evaluations of their competence. While OSP is one of these settings in which developers' motivation is shaped by their learning and performance orientation, research into OSP motivation has not investigated the relationship between developers' individual differences in goal orientation. This is important to do since a participant goal orientation likely influences his participation and, in turn, his/her outcomes. Specifically, goal orientation has been related to a number of adaptive outcomes, including effort, persistence, and better performance [Printrich, 2000].

However, prior research into OSP suggests that engaging in intellectual pursuits and solving challenging problems is one of the main drivers of participation [Lakhani and Wolf, 2005]. This dimension characterizes individuals with a

high learning goal orientation who pursue an adaptive response pattern in which they persist, escalate effort, and report enjoying the challenge. They believe their abilities are malleable and approach tasks with an intention of enhancing their knowledge and competencies. OSP represent an ideal context for developers to share knowledge, expertise, and technical crafts [Raymond, 1999; Kogut and Metiu, 2001]. Any developer can decide to contribute to the software development process, through the OSP platform, and aim at enhancing his competence and/or acquiring new skills. A developer can make use of the OS platform communication channels for coordinating the code development effort, as well as exchanging with peers. Besides the code development activities, knowledge exchanges over these communication networks are an important source of learning for a developer [Weber, 2004]. These arguments suggest the following hypothesis:

Hypothesis 1: Developers' learning orientation positively influences their participation in OSP.

Apart from the learning orientation, developers' participation in OSP can also be shaped by their performance orientation. Indeed, Dweck [1990] notes that a person may operate in both systems of learning and performance goals. Performance oriented individuals, are concerned about their ability and performance relative to others. They approach tasks as a normative social comparison with others in a desire of public recognition, and for "outperforming others as a means to aggrandize one's ability status at the expense of peers" [Covington, 2000, p. 174]. This dimension refers to "ego-gratification," in which developers are likened to craftsmen who want others to admire their artistic style of coding [Raymond, 1999]. This suggests that developers may join OSP to demonstrate to themselves—and to others—their level of ability and establish the adequacy of their ability in the eyes of other developers. However, he also clearly emphasizes how the OS community's internal market in reputation exerts a subtle pressure on people not to launch development efforts they're not competent to follow through on. Therefore, because of the voluntariness of individuals in OSP, we expect learning-oriented individuals to participate more than performance-oriented individuals. Based on the above arguments, we hypothesize that:

Hypothesis 2: Developers' performance orientation positively influences their participation in OSP.

Expectancy Theory

Expectancy theory [Vroom, 1964] suggests that people are motivated to perform an activity, such as contributing software, when they anticipate they will achieve the things they want from doing so. The strength of a motive and a person's choice of the extent of effort invested is governed jointly by the person's expectation of specific outcomes resulting from their actions (expectancy) and by the attractiveness of those outcomes (valence) Bandura [1997]. The mathematical product of expectancy and valence can be used to predict the need (force) for a person to perform a particular act. According to expectancy theory, as expectancy increases, motivation to perform the action increases. It is, therefore, assumed that if an individual expects that contributing to an activity will result in valued outcomes, then he/she will be more motivated to expend effort and demonstrate higher levels of motivation in performing such activity.

Expectancy theory applied to OSP refers to developers' beliefs that the extent and type of effort involved in contributing to OSP will result in professional opportunities and outcomes. Valence refers to the personal relevance or importance of these expectations to OSP developers. An increase in any of these two beliefs should result in more participation.

Hypothesis 3: The higher the level of developers' expectations from the OSP, the higher is their level of participation.

Hypothesis 4: The higher the valence of developers' expectations, the higher is their level of participation in OSP.

Social Exchange Theory

Social exchange theory [Blau, 1964] was developed to explain why individuals engage in cooperative behaviors that are not formally rewarded by the organization, which captures the essence of developers' contributions to OSP in the context of this study. Essentially, in a social exchange relationship, an individual willingly makes a contribution to an organization or another individual as a gesture of goodwill, either based on a trust that this contribution will be reciprocated with an equal exchange at some point in the future or to fulfill an obligation that resulted from a gain received from a previous exchange. At the heart of social exchange theory is the idea that when an individual receives a favor from another party, there is an expectation of some future return, although when it will occur and what form it will take is usually unstated [Blau, 1964].

In his discussion of social exchange theory, Blau [1964] argued that individuals involved in an exchange relationship expect that a balance will be maintained in the exchange between parties.

If the exchange becomes unbalanced, the exchange partners will feel obliged to engage in activities or further exchanges in order to bring the exchange relationship back into balance. This obligation to reciprocate has been termed the *norm of reciprocity* [Gouldner, 1960], which refers to the social obligation created when an individual receives some benefit from the act of another and there is an expectation of some future return.

With respect to OSP, social exchange theory suggests that a developer with valuable knowledge that could be reused by another developer would be motivated to contribute. The benefit received through reusing the software by another developer will result in a sense of obligation to reciprocate, especially in a way that will benefit the partners in the exchange [Blau, 1964; Gouldner, 1960]. Thus, theoretical and empirical research suggest that developers who benefit from accessing and using the code in the OSP will reciprocate in a way that would benefit the project or those who contributed to it. The primary means by which recipients can provide such a benefit is by contributing to the OSP. These arguments suggest the following hypothesis:

Hypothesis 5: *Reciprocity positively influences developers' participation in OSP.*

Another form of social incentives for developers is related to the ideology associated with OSP development. Ideology or identity indicates affiliation to a particular social group, together with some emotional or value significance to him of this group membership [Tajfel, 1972, p. 292; Hogg and Terry, 2000, p. 122]. In this definition, two elements can be distinguished: the first component conveys the extent to which an individual perceives him/herself as belonging to the group, being interwined with the fate of the group, and being a typical member of it. The second component is related to a feeling of pride of belonging to the organization or feeling acknowledged in the organization [Tajfel, 1972, p. 24]. While the latter dimension is mainly tied to the organization it can clearly refer to a group or community.

In the context of OSP, the team's beliefs are the glue that holds developers together. These beliefs guide the specific means by which OSP development is conducted and may provide, according to Stewart and Gosain [2006], an explanation of behaviors enacted by team members that might otherwise be interpreted negatively. This social influence exerted by the OSP community is essential for software projects to be developed and to persist without a central authority or commercial objectives [Scacchi et al., 2006; Gallivan, 2001]. Stewart and Gosain [2006], distinguish among norms, values, and beliefs. Beliefs are the basic assumptions referring to the underlying philosophy of the community and belong to our conceptualization of identification. Based on this, we hypothesize that:

Hypothesis 6: *The higher the level of identification of developers with OSP, the higher is their level participation.*

Individual Outcomes

While many instrumental and non-instrumental outcomes may result from developers' contributions to OSP, we focus on two outcomes: learning and satisfaction. We define learning outcomes as the extent to which an individual's cognitive structures have improved over time, and we focus on three distinct types of learning: replication, adaptation, and innovation [Gray and Meister, 2004]. Satisfaction, on the other hand, refers to the perception of contentment developers derive from their participation to OSP. We consider both their satisfaction with the team and the project advancement.

Learning Outcomes

Participation has been suggested to affect learning outcomes by improving the utilization of existing knowledge and changing the manner in which day-to-day work is conducted. Because individuals with a strong participation invest more attention in learning, they are more likely to extract new knowledge from the activities they are already performing.

Recent findings from an empirical analysis of OSP find both knowledge creation and transfer to be possible in this context [Hemetsberger and Reinhardt, 2006]. These authors explain that some functionalities available online compensate for the absence of face-to-face interactions. For instance, commentaries added in code programs and concurrent version systems (CVS) allow developers to review the process that lies behind the code developed by others. Enabling developers to review the whole history of code development allows them, consequently, to be engaged in reflective observation and to learn from the improvements and errors made previously. This implies that developers with a higher level of participation are more likely to experience higher levels of learning outcomes. To

distinguish learning outcome levels, we based our analysis on three dimensions put forward by Gray and Meister [2004]. The first one, *replication* refers to the exploitation and reuse of existing knowledge and results in efficiencies of not having to re-create knowledge that already exists. *Adaptation* refers to incremental changes in causal structures, paralleling the ongoing evolution of work in response to new developments. Finally, *innovation* refers to radical, discontinuous change. These three classes of cognitive change constitute learning outcomes in our study.

Hypothesis 7: The higher the participation level of developers in OSP, the higher are their learning outcomes.

Satisfaction Outcomes

Apart from the learning outcomes outlined above, our research considers developers' satisfaction as an important dimension in the outcomes they achieve [Crowston, 2006]. The most motivated developers will have a positive behavior toward OSP and their satisfaction with these projects will be as a result higher. Developers' satisfaction may entail different dimensions. A developer can be more or less satisfied with the outcome of the project itself. Taking into account this dimension is of critical importance, as several OSP are abandoned before their closure or result in outcomes that differ largely from initially assigned objectives [Scacchi, 2002].

The absence of a central authority in project teams, of formal deadlines, and of monetary rewards implies that developers have an important autonomy in fulfilling their tasks. Thus, beyond defined project objectives, developers can afford to decide themselves what modules should be developed or not with regard to their perceived interests in the open source community. Similarly, OSP developers' satisfaction depends also on their social experience with the project team. Based on this, we hypothesize that:

Hypothesis 8: The higher the participation of developers in OSP, the higher is their level of satisfaction.

IV. DATA AND METHODS

This section describes our research approach, instrument development and data collection processes. To test the proposed research model, we adopted the survey method for data collection and examined our hypotheses by applying the partial least squares (PLS) method to the collected data. Our unit of analysis, as outlined earlier, is the individual developer.

Data Collection

Our theoretical model stipulates measurement in various time periods. Motivation leads to participation, and participation is an antecedent of performance. Naturally, the data underlying these constructs has to reflect this sequence. In our research design this requires the collection of outcomes' data after the measurement of the motivation and participation constructs. We collected data in two times periods. In Period 1 (2007–2008), we measured the motivation and participation of developers in OSP. In Period 2 (2008), seven months after the first phase of data collection, we measured the outcomes of developers' participation in OSP. This temporal distinction among our measures of motivation, participation, and performance is consistent with the general model of motivation and performance in psychology [e.g., Mitchell and Daniels, 2003] in which the relationship among motivation, participation, and performance is properly considered as a sequence and not as simultaneous events.

The dataset we employ in our analysis consists of OSP hosted on SourceForge.net under the category of software development in 2007–2008. We have chosen SourceForge, because it is the world's largest OSS development website. As of 2007, SourceForge had more than 124,900 projects and more than 1.3 million registered users. SourceForge.net provides free hosting to OSP development through a standard technology toolset reducing, consequently, variance in participation that may be due to differences in technology used to support workflow, code distribution, versioning, etc.

We selected projects from one category on SourceForge—software development (code generator, design, and framework)—and we limited the sample to one similar domain (enterprise application development). This approach has been used in previous OSP studies [e.g., Stewart and Gosain, 2006] and provides an appropriate way to control for differences across projects in very different product categories. After selecting categories, we ensured that the projects had some activity in the past week in terms of contributions to the code repository; requests for bug fixes, support, patches, or features; or in terms of page views. In total, fifty projects met all criteria. A subset of these projects was randomly selected to pilot test the survey. We have selected in each project the member of the project whose role/position was stated project administrator as he/she would be able to provide the requisite perspective and provide support for the study needed in order to reach the other contributors. Twelve developers responded, and none of them indicated any problems in the survey.

Personalized invitations were then sent to the remaining contributors in the sample requesting their participation. In all, 122 contributors responded to our two-times-period survey from a sample size of 320 (an overall response rate of 38.12 percent). Males represented 94 percent of respondents and females 6 percent. Only 12 percent of respondents received a monetary reward for their contribution to the OSP activities. The sample composition reflects the variety of profiles of those taking part: 2.2 percent are unemployed, 57.6 are employed, 14.1 are self-employed, and 26.1 are students.

Measures

The measurement items in the questionnaire were adapted from existing validated and well-tested scales in the literature. The scales had been proved to have good validity and reliability. In the questionnaire, all items were measured with seven-point Likert scales ranging from "strongly disagree" to "strongly agree". Appendix 2 shows all the measurement items applied in the data collection.

Participation

We consider two types of participation in OSP: (1) the level of contribution to the different activities of the OSP and (2) the effort developers put in the project. The level of contribution to the OSP has been operationalized as rating of behaviors. We used five-items from the list of activities performed by developers in the context of OSP [Zhao and Deek, 2004]. Specifically, their measure suggests five items: find bugs, find usability problems, suggest new features, review and inspect source code, submit source code. Effort on the other hand, refers to the number of hours per week spent on a project. This measure has been used in previous OSP studies [e.g., Hars and Ou, 2002; Lakhani and Von Hippel, 2003] and provides an appropriate proxy for participants' contribution to OSP. Survey respondents were asked how many hours per week they spent working on their current OSP.

Goal orientation was assessed using five-items adapted from Dweck and Leggett [1998] to fit the OSP context of this study. Participants were required to indicate their level of agreement with various statements ranging from 1 "strongly disagree" to 7 "strongly agree." Two items were designed to assess an individual's performance goal orientation. An example of an item was "I feel very good when I know I have outperformed other developers in the project." The remaining items assessed learning goal orientation. An example was "I'm willing to select a challenging work assignment from the OS platform that I can learn a lot from."

Expectancy was measured using two items reflecting developers' professional expectations from the OSP [Vroom, 1964]. Participants were asked to rate their degree of satisfaction with two statements related to their expected professional outcomes from OSP using a 7-point Likert scale. An example of an item was "working on the OSP would enhance my career advantages."

Valence or personal relevance [Vroom, 1964] was measured using three items of a bipolar scale reflecting how important developers considered the professional expected outcomes they can derive from OSP ranging from 1 "extremely undesirable" to 7 "extremely desirable." An example of an item was "the professional outcomes I can achieve from the OSP mean a lot to me."

Reciprocity measures were adapted from Constant et al. [1996]. We used two items to measure developers' belief in the reciprocal effect of OSP. An example of an item was "I know that other members of the OSP will help me, so it's only fair to help other members."

Ideology was measured with four items adapted from Stewart and Gosain [2006]. An example of an item was "I believe free software is better than commercial software."

The *learning outcomes*— replication, adaptation, and innovation—were adapted from Gray and Meister [2004].

Satisfaction was measured with four questions reflecting the degree of contentment that developers may derive from their participation in OSP. Following Crowston et al. [2003], we considered both their satisfaction with the project and with the team. An example of an item was "I'm satisfied with the results achieved from the project to date." All the questions used in this study are detailed in Appendix 2.

Minimizing Potential Biases in the Survey Data

Since self-reported data were collected in one context for all variables involved from the same source, there were some possible sources of biases, such as multicollinearity and common-method.

Collinearity (or multicollinearity) exists among two or more independent variables that are highly correlated. It can be seen as a useful signal that there is conceptual redundancy among the chosen indicators, particularly for very

closely worded perceptual measurement items. Such redundancy needs to be identified at the time of interpreting results. To assure the validity of our analysis, it is imperative to establish that there is no collinearity among the independent variables and no nonlinear relations between dependent and independent variables. Indeed, in our case, correlations among independent variables (Appendix 4) are less than 0.5, indicating no problem with multicollinearity. Examination of residual plots revealed no nonlinear relationships.

Common-method bias refers to the amount of spurious covariance shared among variables because of the common method used in collecting data [Buckley et al., 1990]. To minimize such bias in our data, apart from the time separation between the collection of outcomes data, motivation and participation data, we designed the survey to decouple responses between dependent and independent variables.

First, we included a section with several questions unrelated to this study in between those related to the independent and dependent variables. Additionally, the electronic version of the survey automatically forwarded the respondents to the appropriate next questions based on their responses to previous questions, reducing the likelihood that their responses to later questions would lead them to amend their earlier responses, as they could not go back and change their previous answers. The electronic formal also meant that when they began entering their responses to the early questions, they did not know that questions about type and extent of participation would appear later on and didn't know that the second part of the questionnaire (received later) was related to outcomes, so they could not anticipate this. While the survey design minimized the likelihood that answers to earlier and later questions might have influenced each other, to confirm that common-method bias had been avoided, we conducted a factor analysis. A common-method bias would be suspected if all the variables loaded into a single factor, indicating that the respondents answered all questions in the same way, and if pairs of variables that might be expected to be similarly affected by a common-methods bias loaded into the same factors. The factor analysis indicated no evidence of a common-methods bias in the dataset.

V. RESULTS

Partial Least Squares (PLS), was utilized to assess the measurement scales and proposed hypotheses. PLS is a multivariate technique for testing the psychometric properties of the scales and for estimating the relationships between parameters of a structural model. PLS can be used to analyze measurement and structural models with multi-item constructs, including direct, indirect, and interaction effects, and is widely used in IS research [Chin and Todd, 1995]. There were two stages for data analysis. In stage 1 all the instruments were assessed in a measurement model for reliability and validity. In stage 2, the proposed model and hypotheses were tested, with the individual path coefficients and variance explained in the dependent variables examined in the structural model.

Measurement Model

The first step in PLS is to assess the convergent and discriminant validity of the measurement scales. Convergent validity was assessed by (1) reliability of items, (2) composite reliability of constructs, and (3) average variance extracted (AVE). Items reliability was assessed by each item's loading on its corresponding construct. A rule of thumb suggests that the item loading should exceed 0.70. As can be seen in Appendix 4, the loadings (in boldface) for all items exceeded 0.70 [Fornell and Larcker, 1981].

Composite reliability is recommended to be 0.70 or higher. Appendix 3 shows that the composite reliabilities (CR) of all the constructs exceeded 0.70, with the lowest value being 0.83 for ideology. AVE measures the amount of variance that a construct captures from its indicators relative to the amount due to measurement error. It is recommended that it should exceed 0.50 [Chin, 1998]. Appendix 3 shows that all the AVEs of all constructs exceeded 0.50, with the lowest value at 0.60. Hence, all three conditions for convergent validity were met.

Discriminant validity indicates the extent to which a given construct is different from other constructs. One criterion for adequate discriminant validity is that the construct should share more variance with its measures than with other constructs in the model [Barclay et al., 1995]. We used Fornell and Larcker's recommendation that the square root of the AVE for each construct should exceed the correlations between this construct and all the other constructs [Chin, 1998]. In Appendix 3, the boldface numbers on the diagonals are the square root of the AVEs. Off-diagonal elements are the correlations among constructs. All diagonal numbers are much greater than the corresponding off-diagonal ones, indicating satisfactory discriminant validity of all the constructs.

Another criterion for discriminant validity is that no measurement item should load more highly on any construct other than the construct it intends to measure. An examination of loadings and cross-factor loadings showed that all items satisfied this guideline (see Appendix 4). The results indicate that all items had loadings above 0.7 on their respective constructs, and cross-loadings below 0.5 thresholds of item reliability and discriminant validity recommended by Hair et al. [1998].

Hypotheses and Model Testing

We tested our hypotheses by examining the size and significance of structural paths in the PLS analysis. The explanatory power of the structural model is evaluated by looking at the R^2 value of the dependent constructs: participation and outcomes. Because we measure participation in two ways, type of participation and effort, we present two sets of results. Next, we present results for type of participation. To examine the specific hypotheses, we assessed the t-statistics for the standardized path coefficients and calculated p-values based on a two-tail test with significance level of .05. Table 2 presents the results of the PLS analysis used to test the model.

Table 2: Individual Motivation and Participation Results

		Participation			
		type		effort	
	Constructs	**β**	**t-statistics**	**β**	**t-statistics**
H1	Learning	0.28**	2.75	0.31**	2.94
H2	Performance	–0.06	0.56	–0.06	0.50
H3	Expectations	0.24**	2.86	0.12	1.06
H4	Valence	0.05	0.20	0.00	0.00
H5	Reciprocity	0.23**	2.84	0.07	1.15
H6	Ideology	0.17*	2.15	0.16*	2.12

* p <.05, ** p < .01

Links to the Type of Participation

The R^2 for the relationship between the motivations and type of participation model was 0.19. We proposed direct links between learning (H1) and performance (H2) orientations, professional expectations (H3), valence (H4), reciprocity (H5), ideology (H6) and the type of participation to OSP. The path between learning orientation and type of participation was positive and significant ($\beta = 0.28$, $p < 0.01$), while the path for performance orientation was not. Hypothesis 3 and 4 suggested a link among professional expectations, valence, and the type of participation. Our results showed a positive significant link between professional expectations and type of particiipation ($\beta = 0.24$, $p < 0.01$). However, no link was found between valence and type of participation. Finally, Hypotheses 5 and 6 suggested a link between dimensions of social exchange, namely, reciprocity and ideology and type of participation. The results show a positive significant path between reciprocity and type of participation ($\beta = 0.23$, $p < 0.01$), and ideology and type of participation ($\beta = 0.17$, $p < 0.05$).

Links to Effort

The R^2 for the effort model was 0.27. We proposed direct links among learning (H1) and performance (H2) orientations, professional expectations (H3), valence (H4), reciprocity (H5), ideology (H6), and average time spent per week on the OSP. The path between learning orientation and effort was positive and significant ($\beta = 0.31$, $p < 0.01$), while the path for performance orientation was not supported. This suggests that developers motivated by learning, persist, escalate effort, and consequently, spend more time on the OSP. The other significant path relates ideology to effort ($\beta = 0.16$, $p < 0.05$). The other hypothesized relationships between effort and valence, expectations, reciprocity, and performance orientation were not supported.

Links to Outcomes

The other set of remaining hypotheses relate participation to two types of outcomes: (1) learning and (2) satisfaction. We find participation both type ($\beta = 0.19$, $p < 0.05$) and effort ($\beta = 0.16$, $p < 0.05$) to be related to learning outcomes. Our results show that satisfaction, however, is not related to participation (Table 3).

Table 3: Participation and Outcomes Results

		Learning		Satisfaction	
		B	**t-statistic**	**B**	**t-statistic**
H7	Type	0.19*	1.98	0.01	0.33
H8	Effort	0.16*	1.97	0.08	0.71

* p < .05

Communications of the Association for Information Systems

VI. DISCUSSION

Summary of Findings

The aim of this study was to develop and test a multi-theoretical model of developers' motivation to investigate how their differences in motivation affect their *extent* and *type* of participation in OSP and its outcomes. By relying on three different motivation theories—goal-orientation, social exchange, and expectancy—opening the black box of participation (*extent and type*) and two of its outcomes (learning and satisfaction), our objective is to offer a complementary theoretical and analytical lens to provide a better explanation of the divergent motives of developers' contribution to OSP. Our results provide support for the theoretical model and qualified support for most of our hypothesized relationships (Table 4).

Table 4: Results Summary

Hypothesis		Supported?	
		Participation	
		Type	*effort*
H1	Developers' learning orientation positively influences their participation in OSP.	Yes	Yes
H2	Developers' performance orientation positively influences their participation in OSP.	No	No
H3	Developers' expectations will positively influence their participation in OSP.	Yes	No
H4	Developers' valence will positively influence their participation in OSP.	No	No
H5	Reciprocity positively influences developers' participation in OSP.	Yes	No
H6	Ideology positively influences developers' participation in OSP.	Yes	Yes
H7	Participation is positively related to learning outcomes.	Yes	Yes
H8	Participation is positively related to satisfaction outcomes.	No	No

First, we find learning goals to be a significant predictor of developers' contribution to OSP. Developers join and contribute to OSP in the aim of increasing their competency, understanding, and appreciation for code development. We find developers' learning orientation to be associated not only with effort and type of participation, but also with learning outcomes. In this latter connection, learning-oriented developers tend to believe that effort is the key to success. They engage in the activity of code development for its own sake (as an end in and of itself) and try hard when faced with obstacles and difficulty. As a consequence, they achieve higher learning outcomes. This study's results are consistent with prior research on goal-orientation research [e.g., Nicholls, 1984] and on how learning goals favor deep-level, strategic-processing of information, which in turn leads to increased achievement. Performance oriented developers will not invest their time and contribute more intensely to the different activities of the OSP, however. This finding seems to provide support to the argument that ego-oriented developers tend to withdraw from tasks or to reduce their effort when faced with difficulty or defeat in order to protect their self-esteem.

Our results also provide support to the relationship between professional expectations and type of participation, but we find no support for a relationship between effort and professional expectations. This implies that contributors' desires to further their careers or develop professional opportunities may enhance their interest in contributing to OSP. However, these advantages seem to depend on the type of participation rather than on time spent on the project. In hindsight, this may not be too surprising. Making contributions to the OSP can help developers achieve higher status or obtain better career opportunities, but this depends on the type of contribution. This is in line with previous research that suggests it is improbable for participants to advance in the Apache meritocracy without substantive and sustained software code contributions [Roberts et al., 2006]. In addition, the type of participation rather than the estimated time spent on the project is what makes a difference in terms of professional expectations.

In addition to goal-orientation and expectations, our results provide some evidence that social exchange plays an important role in developers' participation to OSP. Specifically, we find both reciprocity and ideology to affect participation. Reciprocity seems to be associated with the type of contribution but not with the time spent on the OSP. Developers will solve a particular bug or add a particular feature because they expect other members of the OSP to act in a similar way. The type of contribution (e.g., source code submitted or reviewed) is what matters to developers, rather than time spent on the OSP. Finally, we find ideological beliefs, those largely attributed to the emergence of the OSP movement, to be associated with both type of participation and effort. This dimension has been suggested by several previous studies to explain developers' commitment and participation to OSP [e.g., Stewart and Gosain, 2006].

Implications

Theoretical Implications

The overarching objective of this research was to move closer to the development of an integrated model of the motivational mechanisms that help explain the participation of developers in OSP and its subsequent outcomes. What is novel about our perspective is that (1) we combine three motivation theories: goal-orientation, expectancy, and social exchange, each well suited for understanding a specific element of motivation; and (2) we relate these theories to both participation (extent and type) and two of its outcomes (learning and satisfaction). Our first theoretical contribution comes in the form of demonstrating the necessity to treat motivation as a multidimensional concept that is better predicted by integrating different motivation theories. This was the first study to integrate three motivational theories in a single model to examine how together they affect participation and its outcomes. Our research findings reveal that learning goals, professional expectations, ideology, and reciprocity are all predictors of participation in OSP that should be integrated rather than examined separately. This also may explain some of the inconsistent and controversial findings on the importance of different motivational mechanisms [Ke and Zhang, 2009].

Our second contribution consists in demonstrating that understanding participation in OSP requires opening the black-box of this concept and integrating both its type and extent (effort). Previous studies examined these concepts separately or focused solely on effort. Consequently, findings need to be cautiously interpreted, as our research findings indicate that different types of motives have different effects on type and extent of participation.

The third contribution of this study demonstrates that learning is an important outcome of developers in OSP. Through their participation in OSP, developers improve their understanding and usage of existing knowledge and extract new knowledge, and their cognitive structures improve over time.

Practical Implications

Our study has important implications for information systems managers on how to successfully attract and motivate developers to contribute to software development projects. While it's focused on a particular application of OSP, its findings suggest that managers of software developers, whether open source or proprietary, have a much broader range of incentives upon which to draw than just financial compensation.

In particular, we investigated the potential of six incentive mechanisms on developers' contributions:

1. *Ideology*: believing in software freedom for its collective development to occur
2. *Reciprocity*: related to the "give and take" aspect of the community
3. *Expectations*: receiving professional opportunities from their contributions to OSP
4. *Valence*: relevance and importance of the expected outcomes to developers
5. *Performance orientation*: comparing themselves to other developers
6. *Learning orientation*: learning in the aim of enhancing their own competence

Our findings reveal that OSP represent a setting in which economic, social, and psychological motives can coincide. Participation to software development activities in OSP do not provide just a short-term incentive to contribute and enhance code, but has an impact on developers' participation and learning outcomes. We find the aforementioned incentives to affect differently the participation of developers in the context of OSP. The primary motives of developers' participation in such projects are learning, reciprocity, career benefits, and ideology. Consequently, managers of software developers should prompt an interest in learning among developers through providing them with challenging projects and work assignments through which they can learn new skills and avoid routine work. They can set up transparent coordination and communication processes in which developers can interact. In this way, they can enhance not only the level of participation of developers but their learning outcomes as well. Commercial firms might also try to emulate open source development by fostering the mutual norm of reciprocity. It is not surprising that perception of fairness in OSP, represented by developers' ability to select the project and tasks they want to participate in, as well as the use of a neutral and accessible platform, such as SourceForge, is crucial for fostering reciprocity and weighs heavily into developers' decision to work with others.

Developers do not like to feel exploited; if they believe that the other members of the project will not contribute equally, the norm of reciprocity is violated. Managers should, consequently, enforce the adherence to this mutual norm of reciprocity, making cooperation between developers a goal as well as a part of the evaluation and success of the project. Finally, the promotion of widespread code sharing within the company between groups and

departments may enable them to reduce code duplication, broaden a programmer's audience. Thus, while some of the benefits conferred from participation in OSP may be less concrete in nature, in that they are mainly related to goal setting, and managers may find it difficult to act upon them. The majority of motives studied can be used by software managers also in commercial firms to motivate developers.

Limitations and Future Research

The results from this study add to the growing literature on motivation of developers in OSP by providing insights into the motivational mechanisms of OSP and how they affect their participation and outcomes. In particular, we integrate several existing theories of motivation, open the participation blax-box, and explain how motivation affects differently the type and extent of participation of developers in OSP and its outcomes. The findings of this study provide several opportunities for future research. First, we used subjective measures for both participation and outcomes. Although there was a time separation between our measurement of motivations and outcomes of developers in OSP, we relied on validated scales to measure our questions. Employing more objective data for measuring participation, to understand, for example, how it evolves over time, would provide further insight into developer participation.

Second, in this study, we limited our investigation to learning outcomes and found that developers' effort and level of contribution to the different activities of the OSP are associated with higher learning outcomes: adaptation, replication, and innovation. Future studies may investigate which other outcomes are more likely to be associated with participation. Outcomes such as career benefits have been suggested to be related to participation and require further investigation.

Third, we tested our model on a sample size of 122 developers working on enterprise application development on SourceForge. While we focused on a particular domain (software development frameworks and tools for enterprise application development); this domain remains a dominant application in SourceForge, and its adoption by firms has increased considerably in the last years [IDC, 2009]. Future studies, should investigate further how companies adopt/adapt these applications to their specific needs.

REFERENCES

Ambrose, M.L. and C.T. Kulik (1999) "Old Friends, New Faces: Motivation Research in the 1990s", *Journal of Management* (25)3, pp. 231–292.

Añez, J. (1999) "Programmers or Entrepreneurs?" *Developer News* (November 1).

Bagozzi, R. and U. Dholakia (2006) "Open Source Software User Communities: A Study of Participation in Linux User Groups", *Management Science* (52)7, pp. 1099–1115.

Bandura, A. (1997) *Self-Efficacy: The Exercise of Control,* New York: Freeman.

Barclay, D., C. Higgins, R. Thompson (1995) "The Partial Least Squares (PLS) Approach to Causal Modeling: Personal Computer Adoption and Use as an Illustration", *Technology Studies* (2)2, pp. 285–309.

Benbya, H. and N. Belbaly (eds.) (2011) *Successful OSS Project Design and Implementation*, UK: Gower Publishing, Forthcoming.

Benbya, H. and B. McKelvey (2006) "Toward a Complexity Theory of Information Systems Development", *Information Technology and People* (19)1, pp. 12–34.

Bitzer, J, W. Schrettl, P. Schroder (2007) "Intrinsic Motivation in Open Source Software Development", *Journal of Comparative Economics* (35), pp. 160–169.

Blau, P. (1964) *Exchange and Power in Social Life,* New York, NY: Wiley.

Bonaccorsi, A. and C. Rossi (2006) "Motivations to Take Part in the Open Source Movement", *Knowledge, Technology & Policy* (18)4, pp. 40–64.

Chin, W.W. (1998) "The Partial Least Squares Approach for Structural Equation Modeling" in Marcoulides, G.A. (ed.) *Modern Methods for Business Research,* Mahwah, NJ: Lawrence Erlbaum Associates, pp. 295–336.

Chin, W. and P. Todd (1995) "On the Use, Usefulness, and Ease of Use of Structural Equation Modeling in MIS Research: A Note of Caution", *MIS Quarterly* (19), pp. 237–246.

Constant, D., L. Sproull, S. Kiesler (1996) "The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice", *Organization Science* (7)2, pp. 119–135.

Covington, M.V. (2000) "Goal Theory, Motivation, and School Achievement: An Integrative Review", *Annual Review of Psychology* (51)1, pp. 171–190.

Crowston, K., J. Howison, H. Annabi (2006) "Information Systems Success in Free and Open Source Software Development: Theory and Measures", *Software Process: Improvement and Practice*, Special Issue on Free/Open Source Software Processes (11)2, pp. 123–148.

David, P.A., A.H. Waterman, S. Arora (2003) *First Full Findings from FLOSS-US: The Free/Libre and Open Source Developer Survey for 2003,* Stanford Project on Open Source Software, http://www.stanford.edu/group/floss-us/stats/ (current January 3, 2005).

Deci, E.L. and R.M. Ryan (eds.) (2002) *Handbook of Self-Determination Research*, Rochester, NY: University of Rochester Press.

Dweck, C.S. (1999) *Self-Theories: Their Role in Motivation, Personality, and Development*, Philadelphia, PA: The Psychology Press.

Dweck, C.S. and S. Elliott (1983) "Achievement Motivation" in Mussen, P. (ed.) *Handbook of Child Psychology: Socialization, Personality, and Social Development*, New York: Wiley, pp. 643–691.

Dweck, C.S. and E.L. Leggett (1988) "A Social Cognitive Approach to Motivation and Personality", *Psychological Review* (95), pp. 256–273.

Dwyer, G.P. (1999) *The Economics of Open Source and Free Software,* Working paper, Federal Reserve Bank of Atlanta.

Fang, Y. and D. Neufeld (2009) "Understanding Sustained Participation in Open Source Software Projects", *Journal of Management Information Systems* (25)4, pp. 9–50.

Ford, J.K. et al. (1998) "Relationships of Goal Orientation, Meta-Cognitive Activity, and Practice Strategies with Learning Outcomes and Transfer", *Journal of Applied Psychology* (83), pp. 218–233.

Fornell, C. and D. Larcker (1981) "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error", *Journal of Marketing Research* (18), pp. 39–50.

Freeman, S. (2007) "The Material and Social Dynamics of Motivation: Contributions to Open Source Language Technology Development", *Science Studies* (20)2, pp. 55–77.

Gallivan, M. (2001) "Striking a Balance Between Trust and Control in a Virtual Organization", *Information Systems Journal* (11), pp. 277–304.

Ghosh, R. et al. (2002) "Survey of Developers. Free/Libre and Open Source Software: Survey and Study", FLOSS, Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Gouldner, A. (1960) "The Norm of Reciprocity: A Preliminary Statement", *American Sociological Review* (25)2, pp. 161–178.

Gray, P. and D. Meister (2004) "Knowledge Sourcing Effectiveness", *Management Science* (50)6, pp. 821–834.

Hair, J. et al. (1998) *Multivariate Data Analysis, 5th edition,* Upper Saddle River, NJ: Prentice-Hall.

Hars, A. and S. Ou (2002) "Working for Free? Motivations for Participating in Open-Source Projects", *International Journal of Electronic Commerce* (6)3, pp. 25–39.

Haruvy, E., A. Prasad, S. Sethi (2003) "Harvesting Altruism in Open Source Software Development", *Journal of Optimization Theory and Applications* (118)2, pp. 381–416.

Hemetsberger, A. and C. Reinhardt (2006) "Learning and Knowledge-Building in Communities", *Management Learning* (37)2, pp. 187–214.

Hertel, G., S. Niedner, S. Hermann (2003) "Motivation of Software Developers in the Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel", *Research Policy* (32)7, pp. 1159–1177.

Hogg, M.A. and D.J. Terry (2000) "Social Identity and Self-Categorization Processes in Organizational Contexts", *Academy of Management Review* (25)1, pp. 121–140.

Johnson, J.P. (2002) "Open Source Software: Private Provision of a Public Good", *Journal of Economics and Management Strategy* (11)4, pp. 637–662.

Ke, W. and P. Zhang (2009) "Motivations in OSS Communities: The Mediating Role of Effort Intensity and Goal Commitment", *International Journal of Electronic Commerce* (13)4, pp. 39–66.

Kogut, B. and A. Metiu (2001) "Open-Source Software Development and Distributed Innovation", *Oxford Review of Economic Policy* (17)2, pp. 248–264.

Krishnamurthy, S. (2006) "On the Intrinsic and Extrinsic Motivation of Free/Libre/Open Source (FLOSS) Developers", *Knowledge, Technology and Policy* (18)4, pp. 17–39.

Lakhani, K. and E. Von Hippel (2003) "How Open Source Software Works: 'Free' User-to-User Assistance", *Research Policy* (32)6, pp. 923–943.

Lakhani, K. and R. Wolf (2005) "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects" in Feller, J.R. et al. (eds.) *Perspectives on Free and Open Source Software,* Cambridge, MA: MIT Press, pp. 3–21.

Lattermann, C. and S. Stieglitz (2005) "Framework for Governance in Open Source Communities", *Proceedings of the 38th Hawaii International Conference on System Sciences*.

Lerner, J. and J. Tirole (2002) "Some Simple Economics of the Open Source", *The Journal of Industrial Economics*, 50(2), pp.197–234.

Lerner, J. and J. Tirole (2005) "The Economics of Technology Sharing: Open Source and Beyond", *Journal of Economic Perspectives* (19)2, pp. 99–120.

Mitchell, T.R. and D. Daniels (2003) "Motivation" in Borman, W. et al. (eds) *Handbook of Psychology: Industrial and Organizational Psychology* (12), New York, NY: John Wiley & Sons, pp. 225–254.

Nicholls, J.G. (1984) "Conceptions of Ability and Achievement Motivation" in Ames, R. and C. Ames (eds.) *Research on Motivation in Education: Student Motivation,* New York, NY: Academic Press, pp. 39–73.

Oreg, S. and O. Nov (2008) "Exploring Motivations for Contributing to Open Source Initiatives: The Roles of Contribution Context and Personal Values", *Computers in Human Behavior* (24)5, pp. 2055–2073.

Printrich, P. (2000) "Multiple Goals, Multiple Pathways: The Role of Goal Orientation in Learning and Achievement", *Journal of Educational Psychology* (92), pp. 545–555.

Raymond, E. (1999) *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary,* Sebastopolous, CA: O'Reilly & Associates.

Roberts, G.C. (1992) "Motivation in Sport and Exercise: Conceptual Constraints and Convergence" in Roberts, G.C. (ed.) *Motivation in Sport and Exercise,* Champaign, IL: Human Kinetics Books, pp. 3–30.

Roberts, G.C. (2001) "Understanding the Dynamics of Motivation in Physical Activity: The Influence of Achievement Goals on Motivational Processes" in Roberts, G.C. (ed.) *Advances in Motivation in Sport and Exercise* Champaign, IL: Human Kinetics, pp. 1–50.

Roberts, J.A., H. Hann, S. Slaughter (2006) "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study", *Management Science* (52)7, pp. 984–999.

Scacchi, W. (2002) "Understanding the Requirements for Developing Open Source Software Systems", *IEEE Proceedings Software* (149)1, pp. 24–39.

Scacchi, W. (2006) "Understanding Free/Open Source Software Development Processes", *Software Process—Improvement and Practice* (11)2, pp.95-105.

Shah, S.K. (2006) "Motivation, Governance, and the Viability of Hybrid Forms", *Management Science* (52)7, pp. 1000–1014.

Shaw, M.H. (1976) *Group Dynamics: The Psychology of Small Group Behavior*, New York, NY: McGraw-Hill.

Speier, C., I. Vessey, J.S. Valacich (2003) "The Effects of Interruptions, Task Complexity, and Information Presentation on Computer-Supported Decision-Making Performance", *Decision Sciences* (34)4, pp. 771–797.

Stallman, R. (1999) "The GNU Operating System and the Free Software Movement", in DiBona, C., S. Ockman, M. Stone (eds.) *Open Sources: Voices from the Open Source Revolution*, Sebastopol, CA: O'Reilly.

Stewart, K. and S. Gosain (2006) "The Impact of Ideology on Effectiveness in Open Source Software Development Teams", *MIS Quarterly* (30)2, pp. 291–314.

Tajfel, H. (1970) "Experiments in Intergroup Discrimination", *Scientific American* (223)5, pp. 96–102.

Vroom, V. (1964) *Work and Motivation,* New York, NY: Wiley.

Waltz, D.B., J.J. Elam, B. Cunis (1993) "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration", *Communications of the ACM* (36)10, pp. 62–80.

Wang, C.K.J. and S.J.H. Biddle (2001) “Young People’s Motivational Profiles in Physical Activity: A Cluster Analysis”, *Journal of Sport and Exercise Psychology* (23), pp. 1–22.

Weber, S. (2004) *The Success of Open Source,* Cambridge, MA: Harvard University Press.

Wood, R.E. (1986) “Task Complexity: Definition of the Construct”, *Organizational Behavior and Human Decision Processes* (37), pp. 60–82.

Wu, C.G., J.H. Gerlach, C.E. Young (2007) “An Empirical Analysis of Open Source Software Developers’ Motivations and Continuance Intentions”, *Information & Management* (44)3, pp. 253–262.

Zeitlyn, D. (2003) “Gift Economies in the Development of Open Source Software: Anthropological Reflections”, *Research Policy* (32)7, pp. 1287–1291.

Zhao, L. and F. Deek (2004) “User Collaboration in Open Source Software Development”, *Electronic Markets* (14)2, pp. 89–103.

APPENDIX 1

Analysis of Empirical Motivation Studies in OSS				
Authors	**Motivation Theory**	**Motives**	**Participation**	**Outcomes**
Hars and Ou (2002)	_	Altruism, identification, future rewards, personal needs	Effort	_
Ghosh et al. (2002)	_	Learn, share knowledge, improve products, cooperate	Effort	_
Lerner and Tirole, (2002)	Labor economics	Career concerns, ego-gratification	Type of tasks	Credit, reputation
Zeitlyn et al. (2003)	_	OSS community, promoting free software, reputation, fun	General contribution	_
Hertel et al. (2003)	Extrinsic-intrinsic (SDT)	Identification, norm and social motives, hedonic and pragmatic	Effort	_
Lakhani and Von Hippel (2003)	Extrinsic-intrinsic (SDT)	Problem-solving time, learning	Provision of online support	_
Lakhani and Wolf (2005)	Extrinsic-intrinsic (SDT)	Enjoyment, user needs, programming skills	Effort	_
Roberts et al. (2006)	Extrinsic-intrinsic (SDT)	Extrinsic (pay, status, use-value)/intrinsic (enjoyment, satisfaction for competence, control and autonomy)	Code submitted & accepted	_
Shah (2006)	Extrinsic-intrinsic (SDT)	Need for software, reciprocity	Code creation	Ranking progress
Bagozzi and Dholakia (2006)	_	Attitude, emotions, identification	Provision of support	Status
Stewart et al. (2006)	_	Ideology (values, norms, beliefs)	Effort, task completion	_
Bitzer et al. (2007)	Extrinsic-intrinsic (SDT)	Signaling, need for software, the fun of play, gift culture	Contribution the project	_
Wu et al. (2007)	Expectancy theory	Helping, human capital, career benefits, personal needs	Intention to continue	_
Ke and Zhang (2009)	Extrinsic-intrinsic (SDT)	Ideology, social identification	Effort	_
Oreg and Nov (2008)	_	Reputation, self-development, altruism	_	Task performance
Xu et al. (2009)	_	Interpersonal relationship, software need, ideology, leadership	Involvement	_
Fang and Neufeld (2009)	Legitimate peripheral participation	Situated learning, identity construction	Advising others, improving code	Sustained participation

APPENDIX 2

Questionnaire Items			
Variables	**Definition**	**Questions**	**Source**
Learning orientation	A learning goal orientation orients developers to acquire new skills and improve their ability.	• I'm willing to select a challenging work assignment from the OSP that I can learn a lot from. • I often look for opportunities in OSP to develop new skills and knowledge. • I enjoy challenging difficult tasks in OSP development where I'll learn new skills.	Dweck and Leggett [1998]
Performance orientation	A performance goal orientation orients developers to achieve a positive evaluation of their current abilities and performance from others.	• I feel good when I know I have outperformed other developers in the project. • The OSP provides a good mean to compare my performance with other developers.	Dweck and Leggett [1998]
Expectancy	Reflects developers expectations to enhance their professional opportunities	• Working on the OSP would enhance my career advantages. • Through my contributions to the OSP, I improve my professional experience toward potential employers.	Lerner and Tirole [2002]
Valence	Reflects personal relevance and importance of the expected professional outcomes to developers	• The professional outcomes I can achieve from OSP are relevant to me. • The professional outcomes I can achieve from OSP matters to me. • The professional outcomes I can achieve from OSP means a lot to me.	Vroom [1964]
Reciprocity	A mutual or cooperative interchange where individuals receiving some benefit engage in activities or further exchanges in order to bring the exchange relationship back into balance	• I trust that someone would help me, if I were in a similar situation. • I know that other members of the OSP will help me, so it's only fair to help other members.	Constant et al. [1996]
Ideology	Ideology or identity indicates affiliation to a particular social group together with some emotional or value significance to him of this group membership.	• I believe free software is better than commercial software. • I think information should be free. • I believe that, with enough people working on a project, any bug can be quickly found and fixed.	Stewart and Gosain [2006]
Participation—Type	Level of contribution to the different activities of the OSP	• Find bugs • Find usability problems • Suggest new features, • Review and inspect source code • Submit source code.	Zhao and Deek [2004]
Participation—Effort	The effort developers put in the project	• The number of hours per week spent on a project	Hars and Ou [2002], Lakhani and Von Hippel, [2003]
Learning	The extent to which an individual's cognitive structures have improved over time	• I now have a much better understanding of the right way to do my work that I did before joining the OSP. • Compared to before joining the OSP, I now know much more about proven methods and procedures. • I have been revising and adapting my knowledge to keep up with changes in versions in the OSP. • New developments in the OSP have caused me to revisit and update my knowledge. • Since joining the OSP, I have thought of some revolutionary ways that my job could be improved.	Gray and Meister [2004]
Satisfaction	Level of contentment from the team and overall project progress	• I'm satisfied with the results achieved from the project to date. • I'm satisfied with progress achieved within this project. • I'm satisfied with the contributions of the team members. • I'm satisfied with the team's effort to achieve the project objectives.	Crowston et al. [2003]

APPENDIX 3

Means, SD, Internal Consistencies and Correlations of Constructs														
	Mean	***Std Dev***	***ICR***	***AVE***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
1 (LO)	5.32	1.59	0.93	0.83	**0.91**									
2 (PO)	5.14	1.4	0.90	0.82	-0.06	**0.90**								
3 (EXP)	4.89	1.38	0.90	0.82	0.42	0.12	**0.90**							
4 (VAL)	5.94	1.06	0.91	0.77	0.32	0.25	0.23	**0.87**						
5 (REC)	4.50	1.39	0.92	0.85	0.43	0.03	0.24	0.30	**0.92**					
6 (IDE)	4.59	1.68	0.83	0.62	0.34	0.05	0.25	0.09	0.06	**0.78**				
7 (CON)	4.61	1.9	0.85	0.60	0.24	-0.04	0.22	0.07	0.21	0.19	**0.77**			
8 (EFF)	3.50	1.6	n/a	n/a	0.32	-0.09	0.11	0.10	0.09	0.16	0.27	**n/a**		
9 (LOUT)	5.16	1.48	0.90	n/a	0.45	-0.11	0.28	0.24	0.30	0.22	0.19	0.25	**n/a**	
10 (SAT)	5.40	1.25	0.89	0.63	0.15	0.10	0.23	0.16	0.24	0.12	0.04	0.04	0.14	0.79

APPENDIX 4

Loadings and Cross-Loadings										
Loadings and cross-loadings	**Learning orientation (LO)**	**Performance orientation (PO)**	**Reci-procity (RP)**	**Ideo-logy (ID)**	**Expec-tancy (EX)**	**Val-ence (VA)**	**Participa-tion Type (PT)**	**Effort (EF)**	**Lear-ning (L0)**	**Satisfaction (SA)**
LO1	**0.92**	–0.09	0.41	0.22	0.31	0.35	0.27	0.34	0.38	0.13
LO2	**0.88**	–0.03	0.35	0.36	0.45	0.26	0.22	0.28	0.42	0.11
L03	**0.96**	–0.05	0.43	0.38	0.41	0.31	0.17	0.15	0.47	0.17
P01	–0.02	**0.86**	0.48	0.11	0.36	0.25	0.00	0.04	–0.11	0.08
P02	–0.07	**0.96**	0.37	0.01	0.42	0.19	0.06	0.12	–0.11	0.09
RP1	0.41	0.06	**0.91**	0.05	0.25	0.30	0.19	0.09	0.31	0.12
RP2	0.39	0.04	**0.94**	0.05	0.22	0.32	0.15	0.12	0.32	0.31
ID1	0.26	0.04	0.01	**0.79**	0.11	0.05	0.20	0.14	0.27	0.15
ID2	0.31	0.06	0.02	**0.76**	0.18	0.10	0.19	0.18	0.12	0.10
ID3	0.28	0.11	0.12	**0.81**	0.29	0.10	0.13	0.13	0.23	0.12
EX1	0.41	0.18	0.49	0.21	**0.87**	0.14	0.24	0.10	0.20	0.22
EX2	0.38	0.13	0.39	0.24	**0.95**	0.12	0.20	0.12	0.33	0.27
VA1	0.36	0.26	0.24	0.05	0.16	**0.85**	0.07	0.05	0.34	0.15
VA2	0.27	0.30	0.30	0.10	0.21	**0.89**	0.07	0.08	0.17	0.10
VA3	0.32	0.20	0.35	0.10	0.24	**0.89**	0.08	0.15	0.31	0.16
TA1	0.13	–0.10	0.15	0.06	0.11	0.03	**0.73**	0.16	0.10	0.00
TA2	0.15	–0.06	0.16	0.16	0.11	0.01	**0.77**	0.26	0.15	0.05
TA3	0.26	–0.04	0.17	0.15	0.24	0.04	**0.84**	0.22	0.22	0.03
TA4	0.27	–0.10	0.20	0.14	0.26	0.06	**0.83**	0.26	0.17	0.02
EF	0. 32	–0.09	0.09	0.16	0.11	0.10	0.29	**1.00**	0.25	0.04
LO1	0.44	–0.19	0.23	0.22	0.28	0.03	0.17	0.24	**0.84**	0.36
L02	0.40	–0.18	0.26	0.24	0.24	0.02	0.16	0.15	**0.89**	0.41
L03	0.47	–0.14	0.20	0.27	0.24	0.08	0.22	0.27	**0.84**	0.33
LO4	0.45	–0.14	0.30	0.21	0.25	0.21	0.21	0.17	**0.82**	0.35
LO5	0.23	–0.15	0.41	0.20	0.22	0.23	0.18	0.16	**0.71**	0.34
SA1	0.10	0.12	0.20	0.07	0.23	0.10	0.07	0.15	0.31	**0.81**
SA2	0.27	0.13	0.22	0.13	0.25	0.08	0.08	0.00	0.41	**0.88**
SA3	0.01	0.10	0.19	0.05	0.25	0.13	0.10	0.10	0.28	**0.71**
SA4	0.27	0.15	0.16	0.06	0.11	0.11	0.07	0.08	0.36	**0.82**

ABOUT THE AUTHORS

Hind Benbya is Associate Professor of Information Technology Management at GSCM-Montpellier Business School, France. Her research consists in developing frameworks and tools for implementing and evaluating complex IT systems (i.e., Knowledge Management Systems, Internal Knowledge Markets, and Co-creation Communities). Her publications appeared or are forthcoming in journals such as *MIT Sloan Management Review*, *Information Technology and People*, *Communications of the Association for Information Systems*, *Journal of Information Technology*. She has published: *Knowledge Management Systems Implementation: Lessons from the Silicon Valley* and co-edited *Successful OSS Project Design and Implementation*.

Nassim Belbaly is Associate Professor of Information Technology Management and Dean of Academic Affairs at GSCM-Montpellier Business School, France. Before joining GSCM, he was Visiting Scholar and Principal Researcher at the Anderson Business School at UCLA, Los Angeles. His current research interests are Knowledge Management Systems, Open Source Systems, coopetitive innovation, and design science research. He publishes in peer review journals in Information Systems and management relative to his research fields.

Communications of the Association for Information Systems

ISSN: 1529-3181