Understanding Sustained Participation in Open Source Software Projects

YULIN FANG AND DERRICK NEUFELD

YULIN FANG is an Assistant Professor in the Department of Information Systems, City University of Hong Kong. He earned his Ph.D. at Richard Ivey School of Business, University of Western Ontario, London, Ontario. His current research is focused on knowledge management, virtual teams, and open source software projects. He has published papers in journals such as *Strategic Management Journal, Journal of Management Information Systems, Journal of Management Studies, Journal of the American Society for Information Science and Technology, Information & Management, Communications of the AIS,* and others.

DERRICK NEUFELD is an Associate Professor of Information Systems at the Richard Ivey School of Business, University of Western Ontario, London, Ontario. His research examines IT-facilitated distributed work arrangements, including consequences for employees (telecommuting), groups (virtual teamwork), and managers (remote leadership). Dr. Neufeld's research has appeared in more than a dozen journals, including *European Journal of Information Systems, Leadership Quarterly, Information & Management, Communications of the AIS, Journal of Information Technology,* and *Journal of Engineering & Technology Management.*

ABSTRACT: Prior research into open source software (OSS) developer participation has emphasized individuals' motivations for joining these volunteer communities, but it has failed to explain why people stay or leave in the long run. Building upon Lave and Wenger's theory of legitimate peripheral participation (LPP), this paper offers a longitudinal investigation of one OSS community in which sustained participation is hypothesized to be associated with the coevolution of two major elements of LPP theory: "situated learning" (the process of acting knowledgeably and purposefully in the world) and "identity construction" (the process of being identified within the community). To test this hypothesis, data were collected from multiple sources, including online public project documents, electronic mail messages, tracker messages, and log files. Results from qualitative analyses revealed that initial conditions to participate did not effectively predict long-term participation, but that situated learning and identity construction behaviors were positively linked to sustained participation. Furthermore, this study reveals that sustained participants distinguished themselves by consistently engaging in situated learning that both made conceptual (advising others) and practical contributions (improving the code). Implications and future research are discussed.

KEY WORDS AND PHRASES: communities of practice, legitimate peripheral participation, open source projects, open source software community, qualitative study.

Journal of Management Information Systems / Spring 2009, Vol. 25, No. 4, pp. 9–50.

0742–1222 / 2009 $9.50 + 0.00.
DOI 10.2753/MIS0742-1222250401

THE OPEN SOURCE SOFTWARE (OSS) development model originated in the 1970s, in part as a defensive reaction to the move by some private software companies to appropriate publicly available software into their proprietary applications [56]. Over the last decade, this intriguing software development paradigm has attracted academic and corporate attention [54, 58]. The OSS model represents an alternative development mechanism that can potentially result in higher-quality software at lower costs when compared with proprietary approaches [46]. Fitzgerald [18] has argued that OSS efforts have now metamorphosed into mainstream and commercially viable forms, a phenomenon he labeled "OSS 2.0." Numerous OSS projects have achieved remarkable adoption success. For example, nearly 70 percent of all the pages sent through the Web are delivered via open source Apache server software [17]. Gartner Group estimates that the market for OSS information technology (IT) services will reach $4.3 billion by 2010. Furthermore, 60 percent of the largest companies in North America planned to implement OSS applications, half of these for mission critical needs [52].

Despite the notable success stories, many more OSS projects have failed, frequently due to insufficient volunteer participation [13, 27, 37]. OSS communities cannot survive or thrive without individual developer contributions [49]. Because participants are often self-employed freelancers and volunteers rather than traditional employees, it is impossible to rely on standard employment contracts and incentives to motivate and retain them [37]. While this challenging "voluntary participation" dynamic has energized considerable research into OSS developer participation [62], most research to date has tended to focus on identifying individuals' initial reasons for getting involved—sometimes with an implicit assumption that these reasons might also be related to extended participation, but only rarely with any explicit consideration of long-term effects. Thus, the objective of this study is to address the research question:

> *RQ: What mechanisms sustain long-term voluntary developer participation in OSS communities?*

Considering that 80 percent of OSS projects fade away due to insufficient long-term participation [9],[1] additional research into this question is imperative [2, 61].

We answer this question as follows. We first review the prior OSS literature, paying particular attention to factors associated with developers' initial versus sustained participation. The theory of legitimate peripheral participation (LPP) [31] is introduced to provide a theoretical backdrop. One main hypothesis relevant to LPP in the OSS community context is developed, and then two alternative hypotheses are drawn from the extant OSS literature. We then describe the qualitative, longitudinal case study methodology used to examine the hypotheses, and present our results. Finally, we discuss how the results extend our theoretical understanding of the nature of sustained participation in OSS communities.

Literature Review—Developer Participation in Open Source Communities

AN OSS PROJECT INVOLVES A DECENTRALIZED COMMUNITY of volunteer developers who collaborate to produce a software product using Internet-based tools such as e-mail,

Table 1. Motives to Participate in Open Source Software Projects

Motives to participate	Related literature
Software use value	[20, 25, 35, 49, 55, 60, 63, 65]
Status and recognition	[25, 28, 35, 47, 49, 55, 63]
Learning	[25, 26, 28, 61, 63, 65]
Personal enjoyment	[25, 28, 49, 55, 65]
Reciprocity	[25, 35, 55]
Getting paid	[25, 49]
Sense of ownership and control	[28, 39, 61]
Career advancement	[25, 35, 55, 65]
Free software ideology	[2, 9, 28, 29, 57]
Social identity	[2, 25]

mailing lists, Web-based concurrent versioning systems (CVS), and bug reporting software. To date, OSS researchers and practitioners have been primarily interested in three subareas of research: (1) developer motivations to participate; (2) competitive dynamics; and (3) innovation processes, governance, and organization (see [62] for a summary of these areas). While much additional research is required within each area, the focus of the present study is on developer participation, and more precisely on sustained long-term participation.

The existing literature has focused on developer initial motivation and access. Factors associated with participation have included software use value [20, 25, 55, 60, 63], learning and personal enjoyment [25, 29, 55], recognition and reputation [25, 29, 35, 47, 49, 55, 63], personal ownership and control [39, 63], career advancement opportunities [25, 35, 55, 65], the free software ideology [9, 57], and desire for social identity [2] (see Table 1). Some researchers have proposed that developers may participate to gain selective, transactional benefits beyond the software itself; for example, by demonstrating their superior programming skills, they may receive career advancement opportunities or improved status within the community [25, 36]. This selective benefit perspective has been characterized as a "private-collective" model of innovation incentives [61]. In contrast, others have taken a social-psychology perspective and proposed that social drivers such as trust and identification, as captured in the free software ideology, constitute the most important motives for participation [2, 57]. Yet others have applied extrinsic and intrinsic motivation theories to understand why developers participate in OSS communities [49]. While these studies provide a good understanding of some underlying rationale for voluntary participation, they do not necessarily explain whether driving factors for sustained participation differ from those for initial participation. In fact, recent research suggests that the original motivating benefits behind OSS volunteerism may erode over time [61], implying that these motivations might not be sufficient for maintaining long-term sustained developer participation.

Available research on OSS participation has also tended to attribute participation to developer accessibility to the OSS community. Although IT makes it very easy for an interested individual to gain basic access to an OSS project's documentation, source code, and communication histories via the Internet (e.g., see www.sourceforge.net), new and unproven participants are not usually granted authority to make substantive changes. Individuals' access control is regulated by more established participants (i.e., "core developers" [33]). For instance, von Krogh and his colleagues [63] found that while initial access to the community was open to everyone, access to certain areas (e.g., CVS) was restricted to those who took on key technical activities and demonstrated advanced technical knowledge. This being said, although the extant literature shows that participants with higher access privilege participate more actively [33], we do not yet know whether or how an initial participant's access condition evolves over time, or how such changes are associated with sustained participation behavior. In this sense, most prior research on developer participation focuses almost exclusively on initial "necessary condition" (i.e., motivation, access) for participation behavior. Sustained participation behavior has been largely ignored.

Among the few studies that have begun to explore sustained participation behaviors, Shah [55] found that long-term participants enjoyed programming and interacting with the rest of the community (i.e., labeled as "hobbyists"), whereas short-term participants were typically driven by an immediate need for software (i.e., use value). While Shah's study did not distinguish whether long-term participants entered the community as hobbyists from the start, or became hobbyists gradually over time, it did conclude that initial participation in an OSS project was predominantly needs driven. This implies that most participants enter the community with a use value motive, but that this motivation may change for some of them over time (e.g., from short-term needs driven, to long-term personal enjoyment). It is not clear what underlying processes are associated with such a transformation. In a similar vein, Bagozzi and Dholakia reported that experienced participants were motivated by personal enjoyment from engaging in community activities (i.e., they see their contribution as "enjoyable joint activities to be done with one's friends"), in contrast to the extrinsic motivations of novice participants whose contributions are "driven by specific task-oriented goals typically of a personal nature" [2, p. 1111]. They also suggested that community experiences strengthened developers' sense of identification, leading to continued participation, and that complex social interactions and group dynamics seemed to underlie developers' community experiences. However, due to the cross-sectional, quantitative nature of the study, they did not fully unearth the mechanisms of such ongoing interactions that contributed to strengthened identity and sustainability of developer participation.

Recently, von Hippel and von Krogh envisioned a "social integration" perspective to explain how individuals become sustained participants in OSS communities [61, p. 218]. This process recognizes that the momentum for participation might not reside within individuals, but emerges from ongoing social interactions with the community. For instance, peer recognition/status motives (as previously studied) do not exist in a vacuum, but are enacted through interactions between the focal developer and other community members. Developers whose identities are increasingly recognized and integrated within the community may enjoy greater benefits and therefore exhibit a stronger desire to continue to participate [64]. As another example, unlike passive learning in schools or training workshops, learning and competence development in an OSS community requires a developer to actively contribute to the construction of the software. As such, community members' learning and peer recognition (i.e., identity development) are closely linked with their social contexts. To develop deeper insights into social mechanisms associated with members' continuous participation in OSS communities, we consider in the next section a dynamic and process-oriented theoretical lens—legitimate peripheral participation [31].

Theoretical Background—Legitimate Peripheral Participation

The theory of LPP primarily explains how participation, situated learning, and identity construction interrelate and coevolve as an individual engages in a community of practice [23, 64]. LPP theory has faced some critique [10, 11, 49],[2] but it has been broadly embraced as an attractive alternative to the traditional cognitive theories of learning since it provides a cogent, contextually situated explanation for learning and knowledge development [7, 43, 59, 64]. There are three main elements to the theory—participation, situated learning, and identity construction [31].

First, *participation* is a central concept, since it is through participation that both situated learning and social identity develop within communities of practice [31]. From the LPP perspective, participation refers "not just to local events of engagement in certain activities with certain people, but a more encompassing process of being active participants in the *practices* of social communities and constructing *identities* in relation to these communities" [64, p. 4, emphasis original]. Thus, participation in communities of practice in a broad sense refers not only to actions or events per se, but also to the vehicle through which one's practices and identity is brought into being [41, 64]. In order to conceptually distinguish participation from the constructs of situated learning and identity construction, for the purpose of this study, we follow the normative definition of participation as "taking presence" in a community, and sustainability of participation as "length of presence" in the community.

Second, LPP theory and its later extensions and applications (e.g., [7, 41, 50]) suggest that learning is situated in participants' everyday practice. *Situated learning* is concerned with the theoretically generative interconnections between persons, actions, knowing, and the surrounding social world [31]. The concept of situated learning in LPP theory emphasizes the social and situational aspects of learning [31, 53], rather than the conventional individual, cognitive aspects [21, 38]. This social and situational orientation argues that situated learning is process oriented and has a strong flavor of practicing (or doing). Situated learning is not something one does when one does nothing else, but rather is "the practice of intervening knowledgeably and purposefully in the world" [41, p. 252], through meaningful practices such as questioning, proposing ideas, discussing issues, seeking feedback, and making behavioral changes [16]. Moreover, situated learning is related to developing or shaping one's competence in relation to certain performance criteria that are valued and recognized by others in the social order, and hence is specific to the local context [24]. To "know" (through situated learning) is to be capable of *practicing* as a competent member in the complex web of local relationships among people and activities [5, 19]. Thus, situated learning is an ongoing social accomplishment, continually constituted by the everyday meaningful practices in the particular social and cultural context within which the actor is situated.

Third, *identity construction* is a core element that is enacted through continuous participation [31]. A member's participation in a community involves the construction of his or her identity; that is, a process of understanding who one is, what one can do, and to what extent one becomes more or less legitimized and valued by the other members [23]. This concept rests on theories of social identity, with special emphasis on two processes of identity construction: identity-regulation (an individual's response to regulation originating from or mediated through the organization, such as recruitment, induction, and promotion acts) and identity-work (an individual's continuous efforts to form, maintain, or revise his or her perception of him- or herself in relation to his or her social context) [1]. Through these two processes, individuals come to embrace or reject opportunities to participate in their community of practice, depending on the resonance of those possibilities with their current sense of self [23]. Identity-regulation and identity-work are enacted in two respective ways: (1) through access to control community resources, and (2) through access to an understanding of community artifacts [31]. Access to control community resources is about authority and what one is allowed to do (for newcomers to become participants in a community of practice, the organization must grant them access to various resources—e.g., other community members, information, community tools, and opportunities for participation). Access to an understanding of a community's artifacts speaks to an individual's competence in acting upon the artifacts emerging within or employed by the community (e.g., history of practice, cultural and procedural norms [31]). As new members become increasingly established in the community, their access to resources and understanding escalates, and this in turn stimulates them to participate even more actively in the future.

Identity construction and situated learning are seen in the LPP theory as mutually influential, collectively contributing to and enabled by ongoing participation. Through situated learning, the member's competencies become increasingly recognized by the community (via identity-regulation) and by the member (via identity-work). Enhanced identity in turn provides the member with more legitimacy and opportunities to develop and demonstrate his or her potential competence and align it with what is valued in the community (i.e., situated learning). The more an individual participates by being present in the community, the more he or she engages in situated learning (and contributes to the situated learning of others), the more he or she is seen as an established participant by the community and by him- or herself, and the more likely the individual is to maintain his or her presence in the community. Through situated learning, identity construction and continuous participation, the individual moves from being a "peripheral" participant who spends a lot of time on the sidelines, to a "full" participant who is an essential player.

Research Hypotheses—Sustained Participation in OSS Communities

Lave and Wenger's [31] theory originated from observations of informal professional communities of practice, and has been extended to divergent organizational contexts (e.g., [8, 40, 41, 50]). However, it has not yet been effectively applied in the OSS community setting. We argue that OSS communities, while dispersed and supported by IT, constitute a variation on the social configuration of communities of practice in which situated learning and identity construction coevolve. The community of practice literature suggests that new members often start out as peripheral participants; they share a common identity, communicate with other members, access a shared repertoire of social and technological artifacts, and have similar epistemological roots. Similarly, OSS developers typically begin as peripheral participants (e.g., they might publicly register as a developer in the developer's mailing list, or contribute to the tracker community by doing peripheral jobs such as reporting bugs, collecting feature requests, offering patches, or providing usage feedback [63]). As with members in communities of practice, OSS developers normally share a common set of ideologies, norms, beliefs, and values [57] that facilitate the enactment of a strong social collective enterprise [9]. OSS developers learn from one another by communicating and sharing knowledge and experience, and by addressing common coding challenges.

Moreover, OSS developers in a community possess a shared pool of technological and social artifacts. For instance, mailing lists provide a historical log of all communications among developers, past and present. CVS data provide a complete history of all changes to the software program that occurred since its inception. Bug reporting software keeps track of software errors as they are identified over time. Tracker software logs new features requested by users and developers. Details of tasks pending completion are also available for visitors to read and act upon. With these tools in hand, peripheral participants have access to a nearly complete history of practical and cultural artifacts, enabling and deepening their engagement. Since the structural characteristics of an OSS community appear closely related to those of a community of practice, we believe that the LPP theoretical lens presents itself as an appropriate perspective through which to offer insights into the participation dynamics in the OSS context.

According to LPP theory, situated learning and identity construction are essential elements of healthy OSS communities. Since situated learning that takes place within a community is largely performance driven and context specific to everyday practice, OSS developers acquire skills and make meaningful contributions to the community by situating their learning in collective software coding processes over time. They usually arrive with a priori knowledge about coding that is needed in the community, gradually situate their knowledge within a specific problem/function domain that concerns the OSS community [63], and then contribute by developing solutions to the challenges at hand [55]. Like most other teamwork aimed at solving a collective technical challenge [16], OSS developers manifest their situated learning by undertaking a variety of behaviors such as questioning, making suggestions, reporting bugs, discussing issues, seeking/providing feedback, writing new code, and making code changes—all depending on the level of access privilege they have been granted.

Through situated learning, OSS participants develop a better sense of what initiatives are more likely to be appreciated and valued by the community in general and by project leaders in particular. As they address real concerns through practices that are meaningful to the OSS community, participants progressively enact their identities. That is, when an aspiring member has made significant contributions to the software development process (through situated learning), established OSS developers with core responsibilities may bestow certain privileges on them (e.g., CVS write permissions, authorship, delegated responsibility for a module, inclusion of contributed code to the CVS), and other participants may express appreciation (e.g., peer recognition through such means as positive feedback, praise, and appreciation) [33]. Such recognition reflects a process of identity construction/identity-regulation [34], and promotes positive self-perceptions of one's standing in the community (i.e., enhanced identity-work).

A developer's identity construction in an OSS community is in turn important for successive situated learning. As de Laat argued, "while tasks may be chosen voluntarily, access to files has to be granted, new code for inclusion be approved of, code freezes be respected, tools and procedures be accepted. This is, however, not the usual hierarchy, but one of esteem . . . if volunteers want to obtain esteem from the project leader(s) of a particular project, they have no choice but to play by their rules, or leave" [14, p. 170]. OSS developers earn and maintain their esteem by engaging in desirable practices; the more they do so, the more opportunities they will have to continue to do so. Conversely, if a developer's identity is less recognized in the community, he or she will have fewer opportunities to engage in the community and build his or her esteem. We thus argue that one's status and peer recognition (i.e., his or her identity in the community) coevolve with situated learning, determining the developer's sustained participation. The foregoing is captured in one main hypothesis (developed from LPP theory):

> *Hypothesis 1: Sustained participation in an OSS community is positively associated with the coevolution of (a) identity construction and (b) situated learning.*

Furthermore, we introduce two alternative hypotheses that are implied in some of the existing OSS literature, for the purpose of increasing the analytical rigor and predictive power of the study [32]. As discussed previously, initial motivation and initial access are the two major factors for OSS developer participation [33, 62]. Although a variety of motivations for participation have been identified [62], it remains implicit that they are associated with sustained participation. A common argument is that developers are motivated to join a community due to selective benefits [61], such as use value [29] or personal enjoyment [25]. Despite a recent dissenting view [61], it is plausible to extend this argument to sustained participation as developers could be motivated to remain in the community so as to retain these benefits. Following this argument, we propose:

Hypothesis 2: Sustained participation is positively associated with initial motivation of developers.

Accessibility defines what a participant is authorized to do in an OSS community. The extant literature shows that participants with higher access privileges have higher status and peer recognition in the community, and therefore enjoy greater benefits, participate more actively [33], and exhibit a stronger desire to continue participating [64]. Therefore,

Hypothesis 3: Sustained participation is positively associated with initial access level of developers.

Research Method

To address the research question, we designed and conducted a longitudinal, qualitative case study analysis using documents posted in one OSS development community. The case study method entails "an empirical inquiry that investigates a contemporary phenomenon within its real-life context" [66, p. 13]. Case research is useful when a phenomenon is broad and complex; when a holistic, in-depth investigation is needed; and when the phenomenon cannot be studied outside the context in which it occurs [4, 66]. It is therefore a sound methodology for this study, which requires an in-depth understanding of the dynamic, complex participation processes that cannot be studied outside the specific OSS context.

Research Site

Several characteristics were identified at the outset to assist in selecting an ideal OSS project to serve as the subject for our study. First, we wanted to focus on a project that had the potential to provide long-lasting value, and avoid "faddish" projects that might quickly become irrelevant. Second, we wished to explore an active project for which a substantial and relatively complete volume of historical data existed. Third, because we wanted to examine variance in developer sustained participation, it was important that the chosen project involve developers with varying degrees of length of participation.

Taking this list of desirable features to the popular OSS portal SourceForge.net, we quickly settled in on phpMyAdmin as an ideal study candidate. phpMyAdmin met the first criterion: it was highly regarded in the OSS community, it was consistently ranked among the top ten most active projects at SourceForge (i.e., SourceForge.net, an Internet-based virtual community portal that supports OSS project development), and it was considered a "must have" program for those who rely on both the PHP and MySQL software technologies. The project easily met the second criterion; phpMyAdmin was a healthy and busy project with a large volume of mailing list messages and CVS commits, and was being steadily contributed to over time with no major interruptions and structural changes in progress. Finally, our initial inspection of the public data revealed that phpMyAdmin developers varied substantially in their participation behaviors.

The purpose of the phpMyAdmin project was to develop a Web-based graphical system for managing MySQL databases. The software provides sophisticated database management features (e.g., allowing the user to easily create and manipulate MySQL database structures, execute SQL statements, and clean and manage data). This project was founded by Tobias Ratschiller in September 1998, lay dormant for approximately two years, and then in March 2001 was resuscitated by Olivier Mueller and two other developers. A total of 715 individuals were registered project members during the data collection period (years 2001–4), and 16 of these had been granted "write" permission, allowing them to update and add to the base of distributable source code. As of 2007, the current stable version of phpMyAdmin had been downloaded from SourceForge.net over 12 million times, and translated into 50 different languages (www.phpmyadmin.net/home_page/team.php).

By sampling developers from within a single OSS community, we eliminated from consideration potential community-level differences such as governance mechanisms [55], code architecture [3], and license design and organizational sponsorship [58], and focused entirely on individual-level differences in relation to the same OSS context. The 2001–4 time frame allowed us to avoid the kinds of institutionalization and governance challenges that often occur at the early stages of an OSS project [14] (i.e., the first stable version of the software was released in 2001, an event that indicated the project had passed its infancy stage). Moreover, no significant governance-related statements were detected in the public data during the study period (e.g., regarding changes in developer status rules), further supporting the conclusion that institutional stability had emerged.

Data Sources

Having access to multiple sources of data is critical to qualitative research, because it enables triangulation and validation of theoretical concepts [66]. Data for this study were assembled from the following four primary sources:

1. Project documentation and related reports. Data collected from the official Web site (http://sourceforge.net/projects/phpmyadmin) included project descriptions, charters, bylaws, and meeting minutes. This public documentation allowed us to develop a holistic understanding of the project in terms of product specifications, development status, user feedback, major developer profiles, and external news reports.
2. E-mail messages. We collected and read 4,332 project-specific e-mail messages that appeared on the public mailing list between March 18, 2001, and June 18, 2004. Reading these messages allowed us to become familiar with the technology as well as with the types, quantity, content, and specific contributions of different developers.
3. Tracker messages. A total of 3,443 messages from a variety of trackers were collected in the focal period, including bug reports (1498), feature requests

(629), patches (223), support requests (532), and translations (561). Peripheral participants might begin participation from contributing to trackers before committing themselves as developers, and developers might visit trackers to collect information from users and interact with peripheral participants.

4. CVS log files. Over 33,000 CVS entries were logged during the study period, specifying what specific code changes were made, who made them, when the changes took effect, and so on. The CVS log information provided a complete history of code changes and allowed us to construct a longitudinal account of developers' contributions.

Data Coding and Analysis

To explain variations in sustained participation among developers, as well as the dynamic processes among sustained participation, situated learning, and identity construction, we adopted a data analysis procedure that reconciled variance and process strategies based on the contingency approach described by Sabherwal and Robey [51]. That is, contingency variables were used to determine the appropriateness of alternative process models (e.g., [22]). We first categorized developers in the OSS community according to the length of their past sustained participation (i.e., using a variance approach), and then analyzed the dynamics between situated learning and identity construction in each categorized group (i.e., using a process approach). Following Yin [66], our rationale for grouping OSS developers was to maximize the empirical variation in the central concept of interest—sustained participation—and then to do a cross-group comparison of patterns. For each group, we analyzed individual cases for concepts and processes of interest that were transparently observable, and looked for similar patterns using within-case analyses [45].

To differentiate length of participation, participation activity for all community members was first summarized using a combination of three indicators: frequency of mailing list submissions, frequency of tracker submissions, and frequency of CVS commits (see Table 2), which effectively captured participants' immediate presence in the community. Each community member was assigned one of the following codes for each quarterly period: (C) core participation (i.e., in this quarter, the individual had contributed at least one CVS commit, representing the most critical development activity), (A) active participation (i.e., in this quarter the individual had contributed no CVS commits, but more than ten e-mail or tracker messages, which represented above-average participation), and (P) peripheral participation (i.e., in this quarter the individual had contributed no CVS commits, and fewer than ten e-mail or tracker messages, which represented below-average participation). All other cells in the matrix were left blank. By differentiating core, active, and peripheral participation levels, we pursued a more fine-grained assessment of participation than what has been attempted to date (e.g., [33, 55]).

Using the quarterly core, active, and peripheral assignments, we classified developers into one of three overall groupings representing three different lengths of presence in the community. The first group consisted of developers who were engaged in core or

Table 2. Developers' Participation Behavior

Type	E-mail total	Tracker total	CVS total	Name	Quarter (March 18, 2001–June 18, 2004)													
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
Sustained participants (*N* = 9)	2,833 (67%)	2,449 (71%)	33,230 (99%)	S1	C	C	C	C	C	C	C	C	C	C	C	C	C	C
				S2	P	P	P	P	P	C	C	C	C	C	C	C	C	C
				S3	P	P	P	P	C	C	C	C	C	C	C	C	C	C
				S4	C	C	C	C	C	C	C	P	P	A	—	—	—	—
				S5	C	C	C	P	C	C	C	C	C	C	—	C	—	—
				S6	—	—	P	P	P	P	P	C	C	C	C	C	C	C
				S7	P	P	P	P	C	C	C	C	P	P	P	—	—	—
				S8	C	C	C	C	C	C	—	C	—	—	—	P	C	—
				S9	C	C	C	C	—	C	—	—	—	—	—	—	—	—
Unsustained participants (*N* = 7)	228 (5.4%)	218 (6.5%)	208 (1%)	U1	C	C	C	P	P	—	—	—	P	P	P	—	—	—
				U2	P	C	C	—	—	—	—	—	—	—	—	—	—	—
				U3	P	C	P	—	—	—	—	—	—	—	—	—	—	—
				U4	C	C	C	—	—	P	—	—	—	—	—	—	—	—
				U5	P	P	C	C	P	—	—	—	—	P	—	—	—	—
				U6	—	A	A	—	—	—	—	—	—	—	—	—	—	—
				U7	—	A	A	—	—	—	—	—	—	—	—	—	—	—

Inactive	1,271	776	0	IN1	—	—	—	A	P	P	P	P	—	—	—	—	—	—
participants	(27.6%)	(22.5%)		IN2	—	A	P	—	—	—	—	—	—	—	—	—	—	—
(*N* = 699)				IN3	—	A	P	P	—	—	—	—	—	—	—	—	—	—
				IN4	—	A	—	—	—	—	—	—	—	—	—	—	P	—
				IN5	—	A	P	—	—	—	—	—	—	—	—	—	—	—
				IN6	—	—	—	—	—	—	—	—	—	A	—	—	—	—
				IN7	—	—	—	—	—	—	—	—	—	A	—	—	—	—
				IN8	—	—	—	—	—	—	—	P	P	—	—	—	—	—
				IN9	—	—	—	—	—	—	—	A	—	—	—	—	—	—
				IN10	—	—	—	—	—	—	—	—	P	A	—	—	—	—
				IN11	—	—	P	P	P	—	—	—	—	—	—	—	—	—
				IN12	P	P	—	—	—	—	—	—	P	P	—	—	—	—
				IN13	—	—	—	—	—	—	—	—	—	A	—	—	—	—
				IN14	—	—	P	P	—	—	—	—	—	—	—	—	—	—
				IN15	—	—	—	—	—	—	—	—	—	—	—	—	P	—
				+ 684 others	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Notes: C = core; A = active; P = peripheral; — = nonparticipation.

active participation for four or more consecutive quarters during the 14-quarter study period (the label for this group was "sustained participants" or SUS).[3] The second group included developers who were core or active participants for two or three consecutive quarters (this group was labeled "unsustained participants" or UNS). The third group included the remaining developers, who engaged with the community lightly or irregularly, for no more than a single quarter of active participation (this group was labeled "inactive participants" or INA). This procedure revealed a small number of SUS and UNS participants ($N = 9$ and $N = 7$, respectively), and a large majority of INA members ($N = 699$)—a distribution that is consistent with other OSS studies [33, 55].

The next step was to analyze individual cases within the SUS, UNS, and INA groups. We evaluated the nine SUS cases, seven UNS cases, and randomly sampled nine additional cases in the INA group. We coded for text that related to situated learning[4] and identity construction in order to examine the main hypothesis (H1). With the understanding that situated learning represents the practice of intervening knowledgeably in one's social context [41], we identified two types of situated learning behaviors: *practical* (e.g., contributing applied software code) and *conceptual* (e.g., contributing thoughts and ideas), representing direct and indirect intellectual inputs to the focal OSS software development effort. A variety of categories for practical intellectual inputs were observed, including offering bug fixes, developing patches, modifying codes, and coding new features. The categories for conceptual intellectual inputs included questioning, answering technical questions, proposing ideas for improvement, reporting bugs, requesting features, suggesting patches, discussing issues with other participants, providing user support, and providing usage feedback. Recognizing that identity construction can represent either identity-regulation or identity-work [1], we coded for them accordingly. As discussed earlier, identity-regulation refers to identity changes regulated in the local social context [31] and is enacted as access to control in the OSS community. Coding categories identified here included changes in authorship status, changes in CVS writing permissions, changes in work scope (additional tasks assigned/allocated), and verbal recognition from peers. Identity-work refers to identity changes perceived by the focal individual as a result of access to understanding of the community artifacts [31]. We thus coded it as self-assessment of one's own work progress, and self-perception of intellectual challenges being faced.

We also coded developers' initial motivations and access in order to assess the two alternative hypotheses implied in the existing OSS literature (H2 and H3). During the coding process, special attention was paid to text that was proximal to the dates of either of the following two incidents: (1) when developers first began participating in the OSS community (because participants' initial motivations were more likely to be shared at that time), and (2) when developers "upgraded" or "downgraded" their level of participation as manifested by a change in their submission status, such as moving from core to active participation (because status change is a major form of identity construction and was most likely to be evidenced at that time). To account for the time dimension of each case, we traced the chain of the coded incidences for each individual using processual narratives, a tactic commonly used in qualitative research to address the temporal sequences of events [44, 66].

Finally, in order to detect within- and between-group similarities and differences, we conducted a series of comparisons at two levels, to examine convergence of participation patterns of individuals within the same group (i.e., SUS/UNS/INA), as well as divergence of patterns across the three groups.

Results

ANALYSES REVEALED WITHIN-GROUP SIMILARITIES as well as cross-group differences between the SUS/UNS/INA categories (see Table 3), discussed below.

Developer Participation Across Groups

Substantial differences in participation behaviors existed across the three groups. The SUS group contributed 67 percent of the total e-mail messages (average of 315 per person), 71 percent of tracker assignments (average of 272 per person), and 99 percent of the total CVS commits (average of 3,692 per person). Members of the UNS group contributed 5 percent of the e-mails (average of 33 per person), 6.5 percent of the tracker assignments (average of 31 per person), and 1 percent of the CVS commits (average of 30 per person). The INA group contributed 28 percent of the e-mails (average of 1.8 per person), 22.5 percent of the tracker assignments (average of 1.1 per person), and none of the CVS commits. The fact that a majority of the code writing work was completed by a very small number of core developers is consistent with the existing literature [33, 55]. To understand the reason for such a participation divide across the groups, particularly the underlying mechanisms differentiating their lengths of presence, the following sections trace developer participation histories with a view to understanding how individuals got started with the project, and how their involvement in the community was motivated and subsequently changed over time.

Motivation to Participate

Software use value was a major motive across the three groups. For instance, the developer who reinitiated and registered the phpMyAdmin project (participant S8) commented in a Web-based press release that it was the potential, practical value of the project that motivated him to get involved: “For me it started when I noticed the phpMyAdmin project was ‘clinically dead’: no new release . . . for the last 18 months, no feedback about submitted patches, etc. . . . [phpMyAdmin could] take away all the pain of creating and maintaining a MySQL database . . . [so I] asked Tobias for his permission to take over the project (he accepted).”

S5, another core developer in the SUS group, expressed his practical interest in contributing to the project in his early messages posted in the mailing list: “I have also put a lot of work into my school webpage, as I did all the programming behind it, and I’m hosting it for my school at the moment . . . the real stuff uses PhP & MySQL to provide the functionality & content.”

Table 3. Cross-Group Comparison

			Starting conditions		Legitimate peripheral participation	
Group	Participation sustainability	Quantity of participation	Access (when joining)	Motivation	Situated learning	Identity construction
SUS (*N* = 9)	Continuous presence for at least four consecutive quarters	Contributed 67 percent e-mail messages, 71 percent tracker assignments, 99 percent CVS updates	Three had CVS write access as administrators; remaining six started with peripheral access and later upgraded to core	Use value (dominant); Personal enjoyment (secondary)	All nine members of this group engaged deeply in the project, advising others by sharing their technical expertise (82 percent community interactions), made substantive contributions to the software code (99 percent CVS updates).	Positive peer recognition and status in the community.

UNS (*N* = 7)	Continuous presence between two and three consecutive quarters	Contributed 5 percent e-mail messages, 6 percent tracker assignments, 1 percent CVS updates	All started with peripheral access; five were later upgraded to core	Use value	All seven UNS members engaged with the OSS community by advising others (80 percent community interactions). However, two did not contribute codes, and the remaining five did contribute, in each case only for a limited capacity (1 percent CVS updates).	Positive peer recognition and status for a limited period of time, but eventually spiraled down (to termination).
INA (*N* = 699)	Presence for less than two consecutive quarters	Contributed 28 percent e-mail messages, 22 percent tracker assignments, zero CVS updates	All started with peripheral access; zero were upgraded to core	Use value	These members rarely engaged in or contributed to the community by advising others (50 percent community interactions) or coding (zero CVS updates).	Peer recognition and status remained low throughout (to termination).

Developers in the UNS and INA groups indicated a similar use value rationale for their participation. For instance, one member of the UNS group reported the following in her first e-mail message: "When I first met phpMyAdmin, I found it to be the essential tool I use almost all the time. Now I wish it [were] easier to use than Access, even with its web-interface limitation. So comes my feature request. . . ." Likewise, developer IN3, a member of the INA group, expressed his interest in the project, as well as a desire to see it released in his own language (i.e., use value): "Guys, I am 'new.' . . . I have seen you working hard for this release. . . . I am interested in join[ing] the project. . . . [It would be] good to make it in another language. . . ."

It was noteworthy that only two developers out of nine in the SUS group expressed any degree of personal enjoyment as a motive. Developer S4 noted: "In addition to my dissertation, I enjoyed exploring . . . ," and developer S3 commented: "I also joined a few other Php projects at SF. It is fun. . . ." Even for these two developers, use value was the primary motive.

In summary, we found that use value was a common, dominant initial motivation to participate across the three groups, regardless of length of presence in the OSS community. Personal enjoyment was also found to be a secondary motive in a minority of cases. Thus, this sample did not provide any striking evidence to support H2, that initial motivation to participate differs by length of presence. Sustained, unsustained, and inactive participants all seem to arrive with similar initial motivations (i.e., software use value).

Initial Access

Open source projects by definition are designed to provide people with an opportunity to gain legitimate peripheral access to a community. Indeed, newcomers are encouraged to engage peripherally by contributing to the tracker (e.g., to report a bug or request new features), and are also free to participate more substantially in the developer mailing list directly (e.g., to discuss core development issues). Certain developers are also awarded higher-level administrative access to the project on their initial arrival (i.e., CVS writing privilege).

Whereas all 699 developers in the INA group began as peripheral participants (i.e., with no CVS write permission), nearly half (seven out of 16) of the developers in the SUS and UNS groups were granted CVS write permission at the outset. However, analysis of the discussion threads revealed that these seven individuals were "veterans" with substantial prior experience. For instance, one core developer sent an e-mail to introduce a new member to the core development team: "I'm proud to announce another new developer: the biggest contributor to the 'bugs' and 'feature requests' trackers at this time, i.e. [U4]." This indicated that U4 had participated peripherally in the tracker community. Because of his solid reputation, U4 was granted CVS writing privilege almost immediately after he rejoined the resuscitated community. Similarly, apart from the three SUS administrators (S1, S8, and S4) who automatically claimed CVS privilege upon taking over the project, two additional core developers were given CVS access very quickly because of their heavy involvement before the project was resuscitated, implying that these developers also had prior experience as peripheral participants. Overall, it appeared that all of the developers across the three groups began with peripheral access (except for the three administrators). Despite a peripheral start, these SUS group members still managed to sustain their participation by gaining and subsequently keeping CVS access privilege. By contrast, none of the five developers in the UNS group who were granted CVS writing privileges ever became sustained participants. Thus, we found little evidence to support H3, the claim that length of participation is related to initial access privilege. Participants with the same initial peripheral status varied in their tenure.

Situated Learning and Identity Construction

While neither developer motivation nor access adequately explained sustained participation differences in the sample, the three groups did demonstrate differences in terms of situated learning and identity construction behaviors. Developers in the SUS group experienced repeated, coevolving situated learning and identity construction events over time. These developers were all highly active situated learners who not only discussed software issues with others (i.e., "advisors" who provided qualified conceptual intellectual inputs, accounting for 82 percent of their community interactions), but also engaged directly in rewriting the software (i.e., "coders," responsible for 99 percent of all code changes). As a result, their contributions were consistently well recognized by the community in general and by the administrators in particular. In contrast, those in the UNS group offered very few code changes (1 percent of code changes), though they did contribute to varying degrees as advisors (accounting for nearly 80 percent of their community interactions). As such, their situated learning activity was for the most part conceptual rather than material, and they received relatively less recognition from the community in terms of CVS privileges. Finally, those in the INA group did not provide as much contribution either as advisors (accounting for an average of 50 percent of their community interactions) or coders (zero code contribution). We analyzed the activities of individuals in each group to better understand these differences. Illustrative examples are presented below.

Sustained Participation (SUS)

S5 was a persistent core participant from Q1 through Q12. Prior to joining the project, S5 had developed deep software development expertise using a variety of different programming languages. As indicated earlier, he joined phpMyAdmin for use value as well as personal interest. After S5 introduced himself at Q1, the project leader (S8) legitimized him with a warm personal welcome, and then asked S5 his opinion about an important organizational element: "Welcome to you ☺ Please document every changes and updates in the changelog, and on the devel list, thanks! And for the user-admin stuff, maybe that working on a separate tree at the beginning (phpmyadmin-devel) would be a good idea, what do you think?"

Starting with the first few assignments, S5 immediately immersed himself in situated learning in practice—questioning, proposing ideas, reporting bugs, providing feedback to others' inquiries, discussing issues with other core participants, and modifying the code. A striking feature of his situated learning was that he was both providing ideas (i.e., acting as a advisor) and implementing ideas (i.e., acting as a coder), whether an idea was generated by him or was confirmed by the community. For instance, among his early activities during Q1 and Q2, S5 recommended a proven software application to manage a TODO list, a topic that involved nearly all the key core developers and generated a long discussion thread: "I think a to do list would be a really helpful tool, so that we can be more productive with the project. I would propose the use of devtodo, but I'm not sure what tools the rest of you use while working on the code. I'm using it for my phpDNS project already, and it works nicely over CVS, making it easy for my co-developers to arrange things to get done." He was also actively involved with the tracker community, submitting many feature requests and bug reports between 2001 and 2003. By engaging himself in activities that generated new ideas, thoughts, and feedback for other participants, his perceived expert identity within the community was strengthened.

S5's situated learning, however, went beyond subject expertise. He also proved himself to be a competent software coder. He actively opened himself up to new opportunities, gained exposure to a wide range of functions, took on new commitments, and consistently delivered on those commitments. For instance, S5 personally completed an essential feature module ("Feature 414807") in Q2. He fixed a large number of bugs and delivered several important patches during the first four quarters, all of which were submitted by others. He proactively repaired a synchronization crisis between language files, and explored a synchronization problem between CVS trees. The results he delivered were successfully incorporated into the software. Because his situated learning activities involved both "thinking" and "doing," S5 was able to demonstrate his competencies in ways that were important, relevant, and consequential to this specific OSS project. This signaled S5's increased understanding of the community's work, and his identity in the community was positively affected (through identity-work).

S5's efforts were recognized by his core-developer peers, and remained so over time (for instance, another core developer, S1, wrote in the public mailing list in Q1, "Thanks [S5] for the new table maintenance features"). Yet his transformed identity was not limited to verbal recognition in the community, but also manifested itself through expanded access to control features and functions over time. For instance, privilege to access certain features was requested and granted in Q2: "Please mark Feature 414807 . . . as closed as I have completed it and placed it in the CVS. Can you also please mark Feature 419841 . . . as assigned to me, as I intend to work on it next and it should be done in the next few days." Project administrator S4 responded: "Done. I've also updated your profile so you [can] do it yourself now." This provides another example of S5's identity being positively affected (in this case, through identity-regulation).

Iterative situated learning, recognizing, and role transforming events were clearly evident throughout S5's engagement with the project, and led to further opportunities for him to engage (an ongoing positive cycle or "spiral-up" LPP process). We found similar LPP patterns across the other developers in the SUS group (see Table 4), further demonstrating the dynamics of a sustained LPP cycle.

Unsustained Participation (UNS)

Members of the UNS group may be characterized as being strong in one area of situated learning (i.e., conceptual "thinking"), but weak in the other (i.e., practical "doing"). One case in point: U4 was a core participant when he joined the developer community from Q1 through Q3, a peripheral participant in Q6, and a nonparticipant thereafter. He worked as an assistant researcher for a government organization. U4's motivation to join the project was similar to S5's (i.e., use value), and he was also warmly welcomed upon his arrival. Thanks to his earlier contribution in the tracker community, U4 was quickly granted CVS write privileges at Q1, and this identity transformation apparently stimulated him to participate actively in the project. As a core participant, he proactively interacted with other core participants in the early months via the mailing list and trackers, sharing his ideas and suggestions by submitting multiple bug reports, feature requests, and patch requests. He was also moderately involved in the mailing list, where he discussed technical issues with other developers (six messages in the first quarter). While these situated learning actions showed off his technical expertise, he was also free to actively contribute to the software code as a developer. At the beginning, he did engage himself with certain programming tasks (such as developing an edit demo function). However, he seemed limited in his capacity to manage concurrent or conflicting coding tasks. According to one self-report in Q2: "I agree that the change [of a particular function] won't be very complicated but now I am working with my in-place edit feature [an alternative/competing function]. So I will work on cloning later unless someone does [the next job in the pipeline]." The change he referred to was not implemented (i.e., it was left open in the tracker community). In fact, U4 completed only two of the six feature requests he had proposed during the first three quarters (four were left open), and he did not complete any of the patch requests or bug reports that were submitted by himself or by others.

U4's situated learning behavior in the community demonstrated a clear divide between "thinking" and "doing," thus creating a barrier to his identity transformation process. While his contribution to the project in the first quarter was recognized within the community, and he was formally acknowledged in the author team list for the first release of the 2001 version (i.e., upgraded identity-regulation), the administrators' technical discussions with him decreased noticeably from Q2 onward. At that point he did not request or claim any additional coding tasks, nor was he asked to take any on (i.e., limited work scope, stagnating identity-regulation). Because he failed to demonstrate himself as a competent or reliable coder, his identity in the community began to flag. Consequently, he was not given more opportunities to engage in situated learning. Citing a career switch, U4 suddenly excused himself from the community in Q3: "I must apologize for having abandon[ed] my duty as a team member. I have switched job from a (sort of) programmer to a lecturer at a new university. I will start

Table 4. Within-Group Analysis of OSS Developers' Participation Dynamics (SUS Group)

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
S1	Core Q1–Q14	Core	Use value: "I also develop internal apps in PhP, and I introduced PhP to some teachers here. We all use phpmyadmin, of course :)"	As the formal leader of the project, S1 had full access to the entire development project and assumed the role of overseeing the direction of the project, dividing work, and making job requests. He demonstrated a holistic understanding of the overall direction of the project. His participation included giving advice to developers, answering technical questions, and making final decisions on functionality. Unlike his predecessors who left the project unattended for a long time, he consistently delivered code and managed the entire project with other core developers. This earned him great respect and recognition within the community.
S2	Peripheral Q1–Q5; core Q6–Q14	Peripheral	Use value: [PHP relates to S2's professional work]	S2 started working as bug reporter (Q1–Q5). In this role, he was exposed to a variety of different code packages. His understanding of the product was enhanced over time as his responsibilities were expanded to meet his (growing) capabilities, resulting in even more opportunities (Q6–Q14).
S3	Peripheral Q1–Q4; core Q5–Q14	Peripheral	Enjoyment: "I also joined a few other Php projects at SF. It is fun . . ." Use value: [S3 worked on a PHP/ PostgreSQL based database application at his regular job]	When peripheral in Q1–Q4, S3 volunteered to take on more responsibilities (e.g., "If you have not time to work on a second branch, it doesn't matter. It's not necessary that we all work on the bugfix branch, Loïc and me could do it. . . . If you let me, I'd modify the code to use TRUNCATE instead of DELETE"). He quickly proved that he could successfully deliver on his promises. His situated learning experience, through voluntary active contribution to the code, eventually resulted in his being granted full access rights to the project in Q5, which made it possible for him to keep contributing more directly (Q5–Q14).

S4	Core Q1–Q7; peripheral Q8–Q9; active Q10; then terminated	Core	Enjoyment: “In addition to my dissertation, I enjoyed exploring here . . .” Use value: [As a cofounder, S4 recognized the potential value of phpMyAdmin to user-friendliness of MySQL]	S4 had full access to the project from day one, and acted as an advisor who fielded other developers’ questions and concerns. He earned recognition within the community through his active participation in constructing the product and responding to ad hoc questions. Like S1, his full control of and knowledge about the product was consistent throughout.
S5	Core Q1–Q3, Q5–Q10, Q12	Peripheral	Use value: “[phpMyAdmin could] take away all the pain of creating and maintaining a MySQL database . . . [so I] asked Tobias for his permission to take over the project.”	S5 received a welcome from a core developer at the beginning (S8: “Welcome to you ☺”), and was given responsibilities immediately from Q1 (“Please document every changes and updates in the changelog, and on the devel list, thanks! And for the user-admin stuff, maybe that working on a separate tree at the beginning (phpmyadmin-devel) would be a good idea, what do you think?”). S5 committed himself to participating in this work by sharing ideas (as an advisor) as well as making the changes (as a coder), and his peers regularly recognized his performance. As his reputation improved over time, so did his responsibilities.

(continues)

Table 4. Continued

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
S6	Peripheral Q3–Q7; Core Q8–Q14	Peripheral	Use value: "[I am] often lazy about documentation [at work], and phpMyAdmin helps out with this."	S6 proved himself to be a competent advisor and productive coder from the day he joined the developer community in Q3. He actively participated in discussions with the core members and continuously took on responsibilities to delivering new code. He created database layouts and entity-relationship models (Q5), built a patch for the vertical display of table rows (Q7), built a Javascript-based query window and SQL history (Q9), and completed vertical display of column properties pages (Q12), to name a few. He was fully recognized as a core developer with CVS access from Q7 onward, for his direct and continuous contributions to the code base.
S7	Peripheral Q1–Q4; core Q5–Q8; peripheral Q9–Q11	Peripheral	Use value: [S7 was a PHP programmer]	S7 joined the developer community as a peripheral participant without CVS access (Q1–Q4). Over time at the periphery, he demonstrated to the administrators his competence and value to the project. He shared ideas and reported bugs, and then volunteered to fix them. For instance, after pointing out a weakness in the user interface in Q4 ("PMA always shows the SQL it created, so couldn't we just add a small checkbox and a textinput giving the user the possibility to add a variablename and then to have a second output of the SQL"), he followed up with a software patch. Because of his contributions to the code, the administrators decided to include him as "part of the dev. team. Now let's hope that the sourceforge people give him cvs access quickly" at Q5. Once he had CVS access, S7's contributions became even more regular (he had ongoing responsibility for developing and maintaining key features such as linking column in printview [Q5], automatic joins in QBE [Q6], and relation views [Q7]).

S8	Core Q1–Q6	Core	Use value: “[phpMyAdmin could] take away all the pain of creating and maintaining a MySQL database.”	S8 had full administrative access to the project from day one, and acted as an advisor who regularly fielded other developers’ questions and concerns. In addition, he consistently coordinated with the other two administrators by maintaining the software code base and taking charge of numerous bug fixes and features that were left unclaimed by other developers. Thanks to his role as a competent advisor and productive coder, he was consistently honored as one of the founders of the revived project.
S9	Core Q1–Q4, Q6	Peripheral	Use value: [S9 was a mySQL administrator]	S9 contributed to the tracker community as a peripheral participant and was being included as a core member after the community was resuscitated. He was an active bug fixer for bugs reported by himself, as well as those assigned by the administrators. He not only delivered on his promises, he also helped others with suggestions and patches. For instance, his response to a problem faced by one participant, “i have change your code in tbl_properties.php3 to change the table type in depending of capability of the mysql installation. i hope it works :-)” (Q2) indicating that he not only provided suggestions (as an advisor) but also worked out solution (as a coder). As such, he was a highly respected developer in the community. In addition, he was actively involved in developing module tasks related to rewriting the dump code for php4 (Q1–Q4) and MySQL table statistics (Q6), and was formally recognized as a major contributor for these two contributions.

giving my first lecture 'Calculus I' next month and after a few months (or years ^_^), I may have some of my students coding in an open source project as a homework. Not sure if this is a valid excuse anyway. :)" Subsequently U4 became an INA participant between Q4 and Q5.

When U4 attempted to rejoin the project in Q6, he was quickly confronted with two stark realizations. First, his understanding of the programming code had become severely outdated, and he admitted that his knowledge and skills were no longer sufficient—in other words, U4 experienced downgraded identity-work ("It is to my surprise to see the project progress so much. I need to spend some time to study the code again. So much features . . ."). This gap disqualified him from taking on the "thinking" expert role he enjoyed previously. Second, due to his lack of presence for the preceding year, U4 had lost his CVS write privileges, that is, downgraded identity-regulation (AM: "btw [by the way] who has removed [U4] from the developers list at SF? :o)." S1: "I did. We will put him back on the list if he becomes a regular contributor.").

Reduced competence and opportunity for practicing meaningfully in the community in terms of thinking and doing, and degraded identity recognition, had a negative effect on U4's LPP cycle. Subsequently, U4 disappeared from the project after Q6—he no longer participated in either the mailing lists or CVS trees. While it was an external intervention (career change) that drew U4 away from the community, it was the absence of ongoing involvement that led to his absolute withdrawal from the community. We found a similar spiral-up-and-down LPP pattern across the other six developers in the UNS group (see Table 5); that is, an early positive LPP cycle triggered by demonstration of software competence and expertise, followed by failure to deliver competently, and ultimately ending in complete withdrawal.

Inactive Participation (INA)

The LPP cycle revealed in the SUS group was reversed in the INA group. The participants in this group failed to achieve recognized performance as either thinking experts, or competent coders. For example, while developer IN3 joined the community with a strong motivation to take part in the project, his initial LPP experience was not very smooth. Like most others, his work scope was initially restricted to translating programming languages, and reporting and fixing bugs. IN3 began his participation with a simple bug report in the mailing list, but did not offer any suggestions about how to fix it and thus missed the opportunity to demonstrate his technical expertise. One week later, IN3 asked a developer to consider translating the software into Spanish (again, without offering direct involvement). The quality of IN3's contributions appeared to be weak, and he expressed a lack of confidence about what he was doing (i.e., his identity-work came into question). In Q2, IN3 criticized himself for a mistake he had apparently made: "I AM A LAMER!! I use[d] php3 files not php but for some reason there IS a .php file on my dir (I AM TERRIBLY SORRY). I was editing .php instead of php3." Thus, IN3's credibility in the community was compromised, and his work was not widely recognized or appreciated. Nonetheless, IN3 expressed interest in getting more involved in Q3: "Give me a task, and evaluate the way I do it . . . I am

Table 5. Within-Group Analysis of OSS Developers' Participation Dynamics (UNS Group)

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
U1	Core Q1–Q3; peripheral or less thereafter	Core	Use value: "I'm the webmaster of phpIndex, a french site about PhP."	Thanks to his strong background experience with PHP and previous contributions to the tracker community, U1 was granted CVS access as recognition by the administrators as soon as he rejoined the developer mailing list (Q1). He communicated and cooperated heavily with other core members such as S4 and S8 by providing valuable suggestions and comments, and his competency as an advisor became widely recognized in the community (Q1–Q2). To acknowledge his contribution, U1 was acknowledged as one of the authors in the 2001 release. However, U1 failed to deliver as much as expected as a core developer with CVS access, and in Q6, he suddenly left the project with no explanation. When he returned in Q9, he had lost his CVS privileges due to his lack of continued contribution to the code base. U1 participated for three more quarters, offering opinions and advice (Q9–Q11), and at one point made a casual bid to regain CVS privilege in Q9: "I don't know if I still have access to the CVS, so, I'm not sure I can commit these changes (if you are agree with, of course)." However, U1 did not receive any response to this proposal, and he left with no further messages.

(continues)

Table 5. Continued

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
U2	Core Q2–Q3; then terminated	Peripheral	Use value: [Software was instrumental to him and relates to his competence]	U2 was a core developer before the project was taken over and moved to the SF Web site. As with U3, U2's understanding about particular modules was appreciated and welcomed, and by Q2, he had gained significant responsibilities and high-level access. However, his permissions were inadvertently revoked in Q3 when the project was moved to a new CVS server (a move that U2 had previously argued against: "So I think you have to have some really good reason for doing this—otherwise I strongly suggest that we keep the development in the current CVS control system"). As a result, his suggestion became less appreciated. He stopped contributing to both the forum and the code base, and his participation terminated completely after Q3.
U3	Core Q2; peripheral Q3; then terminated	Peripheral	Recognition: "Version 2.1.0.1 [of one module] was by me and I'm glad to see after a whole year that you included my fixes to start this branch on sourceforge."	U3 was a core developer before the project was moved to the SourceForge.net Web site, and so he was familiar with the code and many of its intended functionalities (initial high identity-work). His prior understanding helped him to provide useful suggestions and comments to others' problems (i.e., advising), which gained him recognition in the community in Q1 and granting of CVS access privileges in Q2 (identity-regulation). However, since his knowledge was limited to the module he had originally coded, other members of the project team questioned his value when they moved on to new modules in Q3 (reduced identity-regulation). U3 seemed to lose interest and gradually faded away.

U4	Core Q1–Q3; peripheral Q6; then terminated	Peripheral	Use value: "I found it to be the essential tool I [can] use almost all the time . . ."	U4 was a high performer in the tracker community where he actively detected and reported bugs and came up with good feature requests (advising). U4 received an enthusiastic welcome from core participants, and was granted CVS write privileges almost immediately (Q1) (S4: "I'm proud to announce another new developer: the biggest contributor to the "bugs" and "feature requests" trackers at this time, i.e. [U4].") U4 engaged intensively in the project, actively expanded his work scope and responsibilities, and received increased recognition in the community (higher identity-regulation) in Q1. Over time (from Q2) he was considered a "thinker," providing suggestions to others but not doing much coding. Furthermore, he withdrew from the community for two quarters due to a career change (Q4-Q5). Due to his delayed performance in contributing to the code base, his CVS privileges were cancelled (downgraded identity-regulation). Upon returning, his understanding of the project lagged far behind what was needed to keep him as a valuable advisor (downgraded identity-work). U4's participation terminated after Q6.
U5	Peripheral Q1–Q2; Active Q3–Q4; Peripheral Q5	Active, Peripheral	Use value: [U5 was a SQL database user at her daily work]	U5 joined the community when the project was taken over by S8. She was experienced in php programming and was willing to help others by providing technical advice (advising). For instance, administrator S4 broadcasted a call for help in Q2: "Please test this new rc and test it again: we want the 2.2.6 official release to be a "reference" version with as less bug as possible." U5 quickly followed up with test results: "it will log you in ok [when you log in]. Then when you want to logout, it pops up the login box again." She further pointed out a minor issue and provided a quick solution to it: "now, if at this point you wanted to log back in, it will not work. . . . so how about a logout page telling 'you have successfully logged out' and then adding a link to log back in if you logged out accidentally". Her prompt reply with valued advice gained her praise in the community in Q3 (upgraded identity-regulation). However, she ultimately never submitted any patches or illustrative code (no coding), and her status in the community faltered. She eventually quit participating in Q5.

(continues)

Table 5. Continued

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
U6	Active Q2–Q3; peripheral or less thereafter	Peripheral	Use value: [U6's job was related to PHP programming	U6 joined the project and worked on the SQL database in Q2. Although he did not have CVS permissions, he actively responded to questions and concerns with useful ideas (advising). His understanding of the database and advisory value to the community was further enhanced through healthy mailing list communication with the core developers. However, in Q3, U6's participation was actively blocked by a more experienced core developer (with whom he had had a difference of technical opinion regarding a particular feature in the database), and his role as a leading advisor in SQL was challenged. In addition, he did not prove his competence in delivering code. This eventually left U6 with an unsolvable technical problem, and he withdrew from the community in Q3.
U7	Active Q2–Q3	Active	Use value: [U7 was translating the software to French and he was a user of the tool]	U7 joined the mailing list mainly to translate the software to French in Q2. He was also involved in reporting issues and providing feedback based on his experience in using phpMyAdmin (advising). As a user/translator, his feedback was primarily related to translation and provision of comments (limited works cope). For instance, at Q1, S1 was planning a version release and broadcasted an e-mail to the mailing list soliciting developers' feedback so that "nobody would get hurt [because of the release]." U7 followed up with several comments, such as, "Warning, before 2.2.0 is out we need to translate the new strings (and there are quite a few . . .)," and he offered to do the translation. His advice and follow-up actions received favorable acknowledgment in the community in Q3 (upgraded identity-regulation). However, U7's contribution remained limited to issue reporting and translating. Because he was not able to contribute directly to the code, he received no additional recognition such as CVS rights, or expanded coding responsibilities, and thus no continuously transformed identity. He quit after Q3.

all ears. . . . This time I'm not being THAT lame. . . . I swear it used to work under r2." However, no developers responded to his request (i.e., stagnating identity-regulation). IN3 subsequently retreated fully, and did not return.

Likewise, IN9 popped in with a message requesting an additional feature in Q9, but without a clear solution: "What about something like this. Ideally the rotate headers should be done without a page reload but I didn't find a way to do that. The repeat headers should also be displayed without a page reload, but this may take some time to implement and error test." Ultimately, IN9 did not take on any work (limited conceptual or practical contribution), nor was he given any opportunities to perform (restricted work scope). We found similar stagnating, spiral-down LPP patterns across the other developers in the INA group (see Table 6). The INA participants' initial role was initially limited, but they were still fully legitimate in the community. Although these individuals had opportunities to participate, learn in practice, and construct a public identity, they did not perform to the expectations of the core developers as either competent "thinkers" (who could identify problems and offer useful suggestions), or competent "doers" (who were capable of fixing problems directly).

Discussion and Conclusion

The above analyses offer several important findings. First, the results largely invalidate the two hypotheses derived from the existing OSS literature on participation. Results show that initial motivations and accessibility, which has been linked to OSS developers' participation by prior research (e.g., [20, 25, 55, 60, 63]), failed to explain sustained developer participation. Software use value was found to be a universal motivation for joining an OSS community for individuals in all three groups (SUS, UNS, and INA), and thus contradicts H2. Similarly, with the exception of project administrators who assumed project ownership at the very beginning, developers in all three groups joined the project with legitimate but peripheral access to the tracker and developer mailing list communities. There was no striking difference in access levels among the three groups (i.e., most developers began with peripheral access), in contradiction to H3. Rejection of H2 and H3 adds support to von Hippel and von Krogh [61], who suggested that OSS participants' initial conditions might not necessarily lead to sustained participation. In other words, while participants' initial motivation and access may provide the necessary conditions for new members to start participating, they do not provide a sufficient explanation for sustained participation.

At the same time, our findings offer credible support for the main hypothesis, developed from the theory of legitimate peripheral participation [31]. Cross-case comparisons revealed key differences among the three group settings in terms of the LPP processes (as summarized in Table 3). The INA group exhibited a quick spiral-down process of situated learning and identity construction. These members did not make substantial contributions in terms of suggestions and ideas, nor did they contribute much directly to the software code, and they withdrew from the community shortly after arrival. The UNS group was characterized by a temporary spiral-up of situated learning and identity construction in earlier stages, but then withdrawal and eventual

Table 6. Within-Group Analysis of OSS Developers' Participation Dynamics (INA Group)

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
IN2	Active Q2; peripheral Q3; then terminated	Active, peripheral	Use value: "I was using phpmyadmin with different user account [at work] . . ."	IN2 entered the community as a bug reporter in Q2, "When you click on the database, it gives a confusing error that no tables are found . . . which is not really correct (also giving a mysql error output)." He tried offering a solution, "It should say that the user is restricted to use the database. . . . This might be more of an idea, but here is a patch to correct this." However, S8 could not get IN2's patch to work: "failed. Doesn't seem to work . . . at least on my test system." IN2 quit immediately thereafter (in Q3).
IN3	Active Q2; peripheral Q3–Q4; then terminated	Active, peripheral	Use value: "I am interested in join[ing] the project. . . . Good to make it [the software] in another [my] language . . ."	IN3's initial access was restricted to the mailing and discussion lists, and his efforts were solely focused on translating the package into another language (Q2). His recognition was negatively affected when he made a coding mistake in Q2: "I AM A LAMER!! I use php3 files not php but for some reason there IS a .php file on my dir (I AM TERRIBLY SORRY). I was editing .php instead of php3" (i.e., downgraded identity regulation). He was then kept peripheral in Q3–Q4. IN3 left the project in Q4.
IN4	Active Q2	Active	Use value: "I have just done a similar upgrade on my own web site."	IN4 began participating by posting a question: "I don't know what to do about this that makes sense. This must be occurring if the footer is included, but the header is not. (Should this scenario even be happening?)" He did not provide any suggestions about how to fix the problem (neither advising nor coding). Later on, he pointed out another problem in the software and contributed some code: "I have just done a similar upgrade on my own web site and have seen enormous speed increase from the site." However, when S1 tested the patches, he found they did not work. IN4 terminated his participation after this encounter.

IN5	Active Q2; peripheral Q3	Active, peripheral	Use value: [IN5 was an end user of the database tool]	IN5 acted as a commenter from the time he joined the OSS community. On the second software release in Q2 that incorporated a few new changes he commented, "Looks much better—I like the ability to select which tab to default to." "I like the LeftFrameLight, especially with lots of db's, just found initial orientation a bit hard." However, he did not offer suggestions to others for software improvement or do anything to the code on his own (neither advising nor coding). His participation ended after Q3.
IN9	Active Q8; then terminated	Peripheral	Use value: "I'm interested in integrating phpMyAdmin with . . ."	IN9 was active in reporting errors and suggesting fixes and interface improvements for a brief time in Q8. His participation was initially frequent and strong, but he made several errors in Q8 that were noted by the core development team, and his suggestions were not seen as useful in the community (downgraded identity-regulation). At one point, S4 declared: "Your page just causes JS errors here (with IE 6.0 SP1), nothing else." IN9 quickly stopped contributing immediately after this communiqué.
IN11	Peripheral Q3–Q5; then terminated	Peripheral	Use value: [IN11 was in charge of phpMyAdmin users in a department]	IN11's participation in the project remained passive. He did not appear to engage in personal participation or learning by solving problems himself. Instead, IN11 reported bugs that were reported to him by his customers, with the apparent expectation that somebody else would explore and resolve them (e.g., "Can someone else try to recreate this? If so please tell me I'm not insane." Q3). Thus, he did not contribute to code nor provide advice to others' problems. IN11 left the project in Q5.

(continues)

Table 6. Continued

Name	Participation behavior	Initial access	Initial motivation	LPP cycle (situated learning ↔ identity construction ↔ sustained participation)
IN12	Peripheral Q1–Q2, Q9–Q10; then terminated	Peripheral	Use value: "I have been looking into using system calls to deal with database backup/ restoration."	IN12 joined the development project specifically for the purpose of "looking into using system calls to deal with database backup/restoration" (Q1). However, the first job assigned by the project administrator appeared to exceed IN12's capabilities (Q2). After several rounds of dialogue with the core development team, IN12 did not make any conceptual or practical contribution to the community, and thereafter withdrew from the project.
IN14	Peripheral Q3, Q4	Peripheral	Use value: [Joined the community in order to help his customers in his daily job]: "A user of 2.2.0 and 2.2.1 came to me today because the edit row page kept on coming up incomplete."	IN14 entered the community with a bug report (Q3): ". . . at line 235 of db_stats. php3 of the 2.2.0 release the link to the database doesn't keep the current server/ language in the URL. Most recent version of CVS doesn't have this fixed either." However, he did not provide any alternatives to fixing the problem (no advising or coding). He later reported another problem as referred to by one of his customers, but again did not provide any concrete suggestions in the form of conceptual ideas or software coding. Shortly thereafter he left the community (Q4).
IN15	Peripheral Q13; then terminated	Peripheral	Use value: [PHP logo design relates to his work as a professor]	IN15's work was largely constrained to designing a logo for the project (Q13); no additional responsibilities were provided from the core team. The logo design task did not progress smoothly, and numerous major revisions were suggested. IN15's original logo idea was eventually rejected and replaced with an entirely new design, thus his performance was not recognized by the community. IN15's contributions terminated in Q13.

termination. Developers in this group predominantly focused on proposing ideas/suggestions (i.e., thinking), but did not contribute substantively to the software code at a level that met the expectations of the community (i.e., doing). However, developers in the SUS group demonstrated a spiral-up process of situated learning and identity construction, in which they engaged significantly in both conceptual "thinking" activities (e.g., proposing ideas, reporting bugs, discussing solutions, requesting features) and practical "doing" activities (e.g., fixing bugs, developing patches, modifying codes). The community recognized these behaviors by granting more resources, privileges, and responsibilities. In this sense, situated learning was as much about constructing positive social relations as it was about creating a useful software application [42].

Theoretical Implications

This study has several theoretical implications. First, it begins to squarely examine the important question of sustained developer participation in OSS communities, whereas prior research has focused on initial participation only [62] or at best speculated about "social integration" [61]. To our knowledge, no theoretical or empirical work has been done to understand how this social interaction sustains developer participation. This paper provides insights into the specific nature of this social interaction (i.e., situated learning, identity construction), rooted in LPP theory, and demonstrates that sustained participation is a natural outcome emerging from successful social interaction between a participant and his or her community. This finding complements and extends the work of Bagozzi and Dholakia [2] and Shah [55], who reported that short-term OSS participants were needs driven whereas long-term participants were enjoyment driven. We extend this work by suggesting that a participant may arrive with use value (needs) as the predominant motivation, but over time repeated positive situated learning and identity construction social interactions can transform this short-term, utility focus into a long-term, esteem-oriented identification with the community. These findings also complement the early work [2] that reported that more experienced (long-term) members were more strongly identified with the community. Our study extends this work by articulating the underlying social interaction processes of this relationship that, if successfully managed, would transform a newcomer into an experienced, recognized participant in the community. The explanatory power of the LPP perspective on sustained participation is further enhanced through the testing and rejection of two alternative hypotheses originating in the OSS literature [32].

Second, our study extends the LPP perspective in the specific context of OSS communities by revealing that LPP behaviors in the OSS community tend to be distinctly contribution oriented. The OSS community only integrated and retained participants who demonstrated readily evident conceptual ("thinking") and practical ("doing") contributions. This contribution orientation effectively distinguishes the OSS community from other communities where legitimate peripheral participation was originally examined (e.g., the butcher and midwife communities [31], the employee community in an insurance claims processing department [64], the service technician community in Xerox [7]). In these traditional communities, the social relationships among participants are either apprenticeship oriented [31] or collegial oriented [7, 64]. Participants usually grow their identity and competence by working on peripheral tasks under the guidance of experienced masters [31], or by socializing face-to-face with relevant others who have similar skill sets and concerns [7, 64]. In these communities, individuals take on the role of legitimate "learners" or "competence builders"; distinct contributions are not a prerequisite of tenure. The OSS community, however, seems to emerge as a special configuration of communities of practice where participants are expected to perform in a "contested field." To remain as long-term participants they must arrive with relevant skills and competences that are already useful to the community, and immediately and persistently contribute to the community by providing advice to others and writing software code. In this sense, we contend that situated learning in an OSS is not about learning programming skills (echoing recent OSS field studies [15, 55]), but rather is about engaging the community in a knowledgeable way, and constructing a unique social identity, through "thinking" (advising) and "doing" (coding) contributions. This participation process ensures that the most competent and valuable participants earn their esteemed status and subsequently stay in the community.

Third, our findings establish initial motivation and access as necessary, yet not sufficient, conditions for sustained participation. A necessary precondition of success for the contribution-oriented participation and retention process is abundant availability of willing and capable candidates. The OSS community achieves this by making initial access extremely easy for individuals, most of whom are motivated by the use value as evidenced in this study as well as prior literature [20, 25, 35, 49, 55, 60, 63, 65]. This configuration is again differentiated from traditional communities, where membership is not highly restricted but still confined by certain explicit requirements (e.g., within a department [64], an organization [7], or a geographic area [31]). Unrestricted Internet-based access, and the ability to motivate participants (e.g., hit a programmer's "personal itch" [46]), makes participation possible from anywhere and at any time, resulting in a large initial candidate pool.

Summarizing these findings, we arrive at a more nuanced theoretical understanding of the underlying mechanism of sustained OSS participation, and suggest the conceptual model illustrated in Figure 1. Initial access and motivations are the precursors to peripheral participation in an OSS community. They are necessary, but not sufficient, for sustained participation. Once a member begins to participate, the LPP process ensues. This is an iterative process of situated learning by advising (conceptual "thinking") and coding (practical "doing"), and identity construction through community recognition (identity-regulation) and self-perception (identity-work). As this process is positively reinforced over time, sustained participation results. If the process is not positively reinforced, participation decreases over time.

Methodological Implications

This study adopts a procedure that reconciles variance and process strategies. Information systems researchers commonly see these two strategies as mutually exclusive and difficult to reconcile. Sabherwal and Robey [51] convincingly demonstrated that

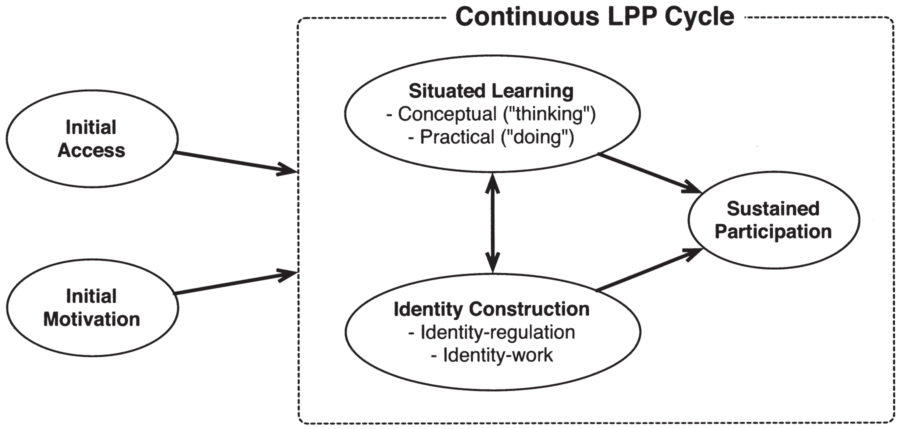

Figure 1. A Conceptual Process Model of Sustained Participation in the OSS Community

three viable options exist to reconcile the two strategies, but empirical studies deploying them to date have been limited. Our study adopts the contingency approach to reconciling variance and process strategies, and establishes that variations in sustained participation behavior are associated with different process dynamics, thus serving as a new application of the variance–process combined strategy.

This paper also demonstrates the potential of conducting qualitative research on OSS communities using publicly available online archival data. Most existing OSS studies have used either anecdotal narratives or quantitative survey methods, with a few limited exceptions (e.g., [33, 55]). Qualitative methods can provide a process-oriented view to explain "why" and "how" [66], and meaningfully account for complex relationships among concepts. The rich descriptions of online archival data provide a truthful and effectively complete account of everyday communications, and thus afford an opportunity for researchers to validate existing theories or develop "substantive" theories that stay close to the original data [30]. These data also make it possible to develop longitudinal accounts of virtual ways of working. While the existing OSS literature has begun to explore this methodological option (e.g., [55]), we believe that online archives hold much promise for advancing knowledge, and deserve more attention.

Practical Implications

This study has several important practical implications for OSS administrators and participants. For administrators, the study suggests that project success may demand the sustained participation of a small number of "core" developers who possess strong technology skills and proven records to play a vital role in the project. To identify and retain these high-caliber individuals, the first step is to attract a large pool of potential candidates by keeping the community free to access (open) and the target software widely relevant and useful. Administrators must be able to easily assess and recognize valuable contributions, for example, by granting distinct access privileges (thus "priming" a positive LPP cycle). For OSS participants, our findings suggested that OSS communities might be more of a "contest" field than a "playing" field. In order to grow and thrive in an OSS community, initial motivation and access to the community are insufficient conditions. Participants must be impelled to make direct (software coding) contributions in addition to indirect (advisory) contributions. Of these two means, direct contribution to the code is the most straightforward way to develop a renowned status in the community. Indirect contribution is helpful for growing one's status and gaining recognition in the short term, but does not appear to be sustainable in the long term.

Limitations and Future Research

Limitations in this work suggest several opportunities for future research. The conceptual model (see Figure 1) should be empirically retested in multiple different OSS communities. Additional primary data collection is another possibility (e.g., conducting interviews with OSS developers); such data could be triangulated against secondary data sources. Future research could also use quantitative methods to extend the generalizability of this research study (e.g., a structured survey could be used to capture participants' identity construction and situated learning practices). Likewise, social network analysis [12] could be used to quantify online activities (e.g., tracker forums, mailing lists, CVS contributions), and then associate these social interactions and structures with participation tenure.

The present study focuses on understanding variations in individual developers' sustained participation in one community, and so does not examine community-level factors. Future research could examine community-level factors that might influence individual developer sustained participation by studying and comparing multiple communities simultaneously (e.g., see [55]). Third, this study did not explicitly address potential effects of power-relations in influencing developer participation, because power is not accounted for in LPP theory. However, the issues of identity and access do appear to have implications for power-relations in Lave and Wenger's theoretical account of situated learning, in that power is implicitly exercised to "foster or impede access to, and continuing membership of, communities of practice, as distilled in the phrase 'legitimate peripheral participation'" [10, p. 285]. It is difficult for a peripheral participant to participate, to learn a new practice, or to be an identified member of a community if power-relations impede or deny access to components of the community such as resources, technology, and established members [10]. Future research may adopt an alternative theoretical lens, such as Bourdieu's theory of practice [6], to account for possible power tensions among developers in the same field of practice.

In conclusion, this study has examined the question, *what mechanisms sustain long-term voluntary developer participation in OSS communities?* Results revealed that situated learning and identity construction activities, as predicted by LPP theory, were linked to sustained participation. The results also showed OSS communities to be a special configuration of communities of practice that are distinctly contribution oriented. By consistently encouraging and rewarding both conceptual and practical contributions, OSS administrators can promote long-term sustained developer participation.

Acknowledgments: Earlier versions of this paper were presented at the 2006 Hawaii International Conference on System Sciences and at the research seminar series of the City University of Hong Kong. The authors thank Vladimir Zwass, the two anonymous reviewers, Blaize Reich, Sonali Shah, and conference/seminar participants for their helpful comments. This research was supported by the General Research Fund from the University Research Council, Strategic Research Grant at the City University of Hong Kong, and research funding from the Richard Ivey School of Business.

Notes

1. See "Unmaintained Free Software" at www.unmaintained-free-software.org/wiki/Main_Page.

2. According to recent critiques of LPP theory [10, 48], popularized versions of situated learning and LPP [7, 8] have tended to ignore several important contextual factors under which a community of practice is formed, such as power dynamics, trust, and predispositions, among others [48]. Since our study represents one of the first attempts to apply LPP theory to the OSS context, we have focused on the mainstream concepts and their interrelations as suggested by Brown and Duguid [7], Lave and Wenger [31], and Wenger [64]. We hope that future research will extend this work to account for these sorts of limiting factors.

3. We set the four-quarter heuristic based on a research article by Colazo and Fang [9], which reported that the average tenure for core developers across 62 popular OSS projects was 369 days (i.e., approximately four quarters).

4. Situated learning captures various types of participation (i.e., *what* developers participated in), whereas participation sustainability is differentiated by length of continuous presence (i.e., *how long* developers consistently participated).

References

1. Alvesson, M., and Willmott, H. Identity regulation as organizational control: Producing the appropriate individual. *Journal of Management Studies, 39,* 5 (2002), 619–644.

2. Bagozzi, R.P., and Dholakia, U.M. Open source software user communities: A study of participation in Linux user groups. *Management Science, 52,* 7 (2006), 1099–1115.

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

4. Benbasat, I.; Goldstein, D.K.; and Mead, M. The case research strategy in studies of information systems. *MIS Quarterly, 11,* 3 (1987), 369–386.

5. Boland, R.J., and Tenkasi, R.V. Perspective making and perspective taking in communities of knowing. *Organization Science, 6,* 4 (1995), 350–372.

6. Bourdieu, P. *Outline of a Theory of Practice.* Cambridge: Cambridge University Press, 1977.

7. Brown, J.S., and Duguid, P. Organizational learning and communities of practice: Toward a unified view of working, learning, and innovation. *Organization Science, 2,* 1 (1991), 40–57.

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

9. Colazo, J., and Fang, Y. The impact of license choice on open source software development activities. *Journal of the American Society for Information Science and Technology* (2009), forthcoming.

10. Contu, A., and Willmott, H., Re-embedding situatedness: The importance of power relations in learning theory. *Organization Science, 14,* 3 (2003), 283–296.

11. Cox, A. What are communities of practice? A comparative review of four seminal works. *Journal of Information Science, 31,* 6 (2005), 527–540.

12. Cross, R., and Parker, A. *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations.* Boston: Harvard Business School Press, 2004.

13. Crowston, K.; Annabi, H.; and Howison, J. Defining open source software project success. In A. Massey and S. March (eds.), *Proceedings of the 2003 International Conference on Information Systems.* Atlanta: Association for Information Systems, 2003, pp. 1–14.

14. de Laat, P.B. Governance of open source software: State of the art. *Journal of Management and Governance, 11,* 2 (2007), 115–117.

15. Ducheneaut, N. Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work, 14* (2005), 323–368.

16. Edmondson, A. Psychological safety and learning behavior in work team. *Administrative Science Quarterly, 44,* 2 (1999), 350–383.

17. February 2004 Web server survey. Netcraft, Bath, England (February 2004) (available at http://news.netcraft.com/archives/2004/02/01/february_2004_web_server_survey.html).

18. Fitzgerald, B. The transformation of open source software. *MIS Quarterly, 30,* 3 (2006), 587–598.

19. Fox, S. Communities of practice: Foucault and actor network theory. *Journal of Management Studies, 37,* 6 (2000), 853–868.

20. Franke, N., and von Hippel, E. Satisfying heterogeneous user needs via innovation toolkits: The case of Apache security software. *Research Policy, 32,* 7 (2003), 1199–1215.

21. Friedlander, F. Patterns of individual and organizational learning. In S. Srivastva (ed.), *The Executive Mind: New Insights on Managerial Thought and Action.* San Francisco: Jossey-Bass, 1983, pp. 78–94.

22. Goodhue, D.L.; Quillard, J.A.; and Rockart, J.F. Managing the data resource: A contingency perspective. *MIS Quarterly, 12,* 3 (1988), 373–392.

23. Handley, K.; Sturdy, A.; Fincham, R.; and Clark, T. Within and beyond communities of practice: Making sense of learning through participation, identity and practice. *Journal of Management Studies, 43,* 3 (2006), 641–653.

24. Henderson, J.C., and Lentz, C.M.A. Learning, working, and innovation: A case study in the insurance industry. *Journal of Management Information Systems, 12,* 3 (Winter 1995–96), 43–64.

25. Hertel, G.; Niedner, S.; and Herrmann, S. Motivation of software developers in open source projects: An Internet-based survey of contributors to the Linux kernel. *Research Policy, 32,* 7 (2003), 1159–1177.

26. Himanen, P. *The Hacker Ethic and the Spirit of the Information Age.* London: Secker & Warburg, 2001.

27. Krishnamurthy, S. Cave or community? An empirical examination of 100 mature open source projects. *First Monday, 7,* 6 (June 2002) (available at http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881).

28. Lakhani, K.R., and von Hippel, E. How open source works: "Free" user-to-user assistance. *Research Policy, 32,* 6 (2003), 923–943.

29. Lakhani, K.R., and Wolf, B. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerals, S.A. Hissam, and K.R. Lakhani (eds.), *Perspectives on Free and Open Source Software.* Boston: MIT Press, 2005, pp. 3–23.

30. Langley, A. Strategies for theorizing from process data. *Academy of Management Review, 24,* 4 (1999), 691–710.

31. Lave, J., and Wenger, E. *Situated Learning—Legitimate Peripheral Participation.* Cambridge: Cambridge University Press, 1990.

32. Lee, A.S. A scientific methodology for MIS case studies. *MIS Quarterly, 13,* 1 (1989), 33–52.

33. Lee, G.K., and Cole, R.E. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science, 14,* 6 (2003), 633–649.

34. Lee, P.C.B. Social support and leaving intention among computer professionals. *Information & Management, 41,* 3 (2004), 323–334.

35. Lerner, J., and Tirole, J. The simple economics of open source. Working Paper no. W7600, National Bureau of Economic Research, Cambridge, MA, 2000.

36. Lerner, J., and Tirole, J. The scope of open source licensing. *Journal of Law, Economics, and Organization, 21,* 1 (2005), 20–56.

37. Markus, M.L.; Manville, B.; and Agres, C.E. What makes a virtual organization work. *California Management Review, 42,* 1 (Fall 2000), 13–26.

38. Merriam, S., and Caffarella, R.S. *Learning in Adulthood—A Comprehensive Guide.* San Francisco: Jossey-Bass, 1998.

39. Moon, J.Y., and Sproull, L. Essence of distributed work: The case of the Linux kernel. *First Monday, 5,* 11 (2000) (available at http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1479/1394).

40. Nidumolu, S.R.; Subramani, M.; and Aldrich, A. Situated learning and the situated knowledge web: Exploring the ground beneath knowledge management. *Journal of Management Information Systems, 18,* 1 (Summer 2001), 115–150.

41. Orlikowski, W.J. Knowing in practice: Enacting a collective capability in distributed organizing. *Organization Science, 13,* 3 (2002), 249–273.

42. Orr, J. Sharing knowledge, celebrating identity: War stories and community memory in a service culture. In D.S. Middleton and D. Edwards (eds.), *Collective Remembering: Memory in Society.* Beverly Hills, CA: Sage, 1990, pp. 169–189.

43. Orr, J. *Talking About Machines: An Ethnography of a Modern Job.* Ithaca, NY: IRL Press, 1996.

44. Paré, G., and Elam, J.J. Using case study research to build theories of IT implementation. In A.S. Lee, J. Liebenau, and J.I. Gross (eds.), *Information Systems and Qualitative Research.* London: Chapman and Hall, 1997, pp. 542–568.

45. Pettigrew, A. Longitudinal field research on change: Theory and practice. *Organization Science, 1,* 3 (1990), 267–292.

46. Raymond, E.S. *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary,* 2d ed. Cambridge, MA: O'Reilly, 2001.

47. Raymond, E.S., and Trader, W.C. Linux and open-source success. *IEEE Software, 16,* 1 (1999), 85–89.

48. Roberts, J. Limits to communities of practice. *Journal of Management Studies, 43,* 3 (2006), 623–639.

49. Roberts, J.A.; Hann, I.-H.; and Slaughter, S.A. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science, 52,* 7 (2006), 984–999.

50. Robey, D.; Khoo, H.M.; and Powers, C. Situated learning in cross-functional virtual teams. *IEEE Transaction on Professional Communication, 43,* 1 (2000), 51–66.

51. Sabherwal, R., and Robey, D. Reconciling variance and process strategies for studying information system development. *Information Systems Research, 6,* 4 (1995), 303–327.

52. Schadler, T. *Open Source Moves into the Mainstream.* Cambridge, MA: Forrester Research, 2004.

53. Schön, D.A. *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books. 1983.

54. Sen, R. A strategic analysis of competition between open source and proprietary software. *Journal of Management Information Systems, 24,* 1 (Summer 2007), 233–257.

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

56. Stallman, R.M.; Gay, J.; and Lessig, L. *Free Software, Free Society: Selected Essays of Richard M. Stallman.* Boston: GNU Press, 2002.

57. Stewart, K.J., and Gosain, S. The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly, 30,* 2 (2006), 291–314.

58. Stewart, K.J.; Ammeter, A.P.; and Maruping, L.M. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research, 17,* 2 (2006), 126–144.

59. Thompson, M. Structural and epistemic parameters in communities of practice. *Organization Science, 16,* 2 (2005), 151–164.

60. von Hippel, E. Innovation by user communities: Learning from open source software. *MIT Sloan Management Review, 42,* 4 (Summer 2001), 82–86.

61. von Hippel, E., and von Krogh, G. Open source software and the “private-collective” innovation model: Issues for organization science. *Organization Science, 14,* 2 (2003), 209–223.

62. von Krogh, G., and von Hippel, E. The promise of research on open source software. *Management Science, 52,* 7 (2006), 975–983.

63. von Krogh, G.; Spaeth, S.; and Lakhani, K. Community, joining and specialization in open source software innovation. *Research Policy, 32,* 7 (2003), 1217–1241.

64. Wenger, E.C. *Communities of Practice: Learning, Meaning, and Identity.* Cambridge: Cambridge University Press, 1998.

65. Wu, C.-G.; Gerlach, J.H.; and Young, C.E. An empirical analysis of open source software developers’ motivations and continuance intentions. *Information & Management, 44,* 3 (2007), 253–262.

66. Yin, R.K. *Case Study Research: Design and Method,* 2d ed. London: Sage, 2003.