Paper to be presented at the

35th DRUID Celebration Conference 2013, Barcelona, Spain, June 17-19

Open Source Participation Behavior - A Review and Introduction of a Participation Lifecycle Model

Daniel Ehls
Hamburg University of Technology
Technology and Innovation Management
daniel.ehls@tuhh.de

Cornelius Herstatt
Hamburg University of Technology
Technology and Innovation Management
cornelius.herstatt@tuhh.de

Abstract

Open Source has proven its impact on business and puzzled researchers. However, little is known about participation behavior or limited to participants traits or participation rationales. We review existing research and propose a coherent participation lifecycle model. The framework illustrates the stages a volunteer progresses through from pre-contribution steps, for instance community selection, to initial participation, e.g. community joining, to sustained participation. The model guides researchers in an end-to-end participation lifecycle framework, highlights certain influences and socialization impact, and creates consensus in research. We also draw the attention to further research areas and propose a research agenda in regards to open source participation behavior.

Jelcodes:Z00,Z00

Abstract:
Open Source has proven its impact on business and puzzled researchers. However, little is known about participation behavior or limited to participants traits or participation rationales. We review existing research and propose a coherent participation lifecycle model. The framework illustrates the stages a volunteer progresses through from pre-contribution steps, for instance community selection, to initial participation, e.g. community joining, to sustained participation. The model guides researchers in an end-to-end participation lifecycle framework, highlights certain influences and socialization impact, and creates consensus in research. We also draw the attention to further research areas and propose a research agenda in regards to open source participation behavior.

Open Source Participation Behavior
- A Review and Introduction of a Participation Lifecycle Model

Market analysis provides stunning figures for openly developed products resulting of the behavior of participants. As of June 2012, Apache has a market share for web server software of 64.33%, followed by Microsoft with 13.76%.[1] Total factory revenues in the worldwide server market of $11.9 billion break down to 16.9% for Linux, 21.8% for Unix and 48.5% for Windows in the first quarter of 2011.[2] However, initial software deployment on sold servers does not represent actual market share, but stresses the high share of open source products in commercial distribution. An actual market share of used server software is obtained by measuring internet traffic. These analyses for the usage of operating systems reveal a market share of only 36.1% for Windows, but 32.6% for Linux, and 31.2% for other Unix distributions.[3] These iconic examples of open product creation have fortified the phenomenon of open source innovation. Innovative goods are produced by volunteers[4], who "program to solve their own as well as shared technical problems, and freely reveal their innovations without appropriating private returns from selling the software" (von Hippel and von Krogh 2003, p. 209). Volunteers create a product, make it publicly available, relinquish most of their IPR, and do not receive a direct compensation (Hars and Ou 2002). This distinctive participants' behavior of voluntary contribution and free revealing has puzzled researchers as open source projects deviate sharply from the private model (participants receive no private rents) and the collective model of innovation (participants receive no subsidy). Moreover, the behavior of participants is outstanding in terms of the locus of innovation: distributed volunteers working together in self-governing communities rather than under contract for firms. The locus of knowledge creation shifts outside the boundaries of the firm and there is no contractual member commitment. Above participant behavior is

1 http://news.netcraft.com/, retrieved 15 June 2012.
2 http://www.idc.com/getdoc.jsp?containerId=prUS22841411, retrieved 30 September 2012.
3 http://w3techs.com/technologies/overview/operating_system/all, https://secure1.securityspace.com/s_survey/data/200907/index.html. http://www.gartner.com/it/page.jsp?id=1654914. Retrieved 30 September 2012.
4 Volunteering exhibits "any activity in which time is given freely to benefit another person, group, or organization" (Wilson 2000, p. 215). Volunteers are neither contractually obliged to participate in communities, nor are they directed by formal hierarchical control (Setia et al. 2012); however, volunteers are not precluded from benefiting from their work (Wilson 2000).

no single case. The Linux Kernel project has more than 1,000 contributors. The large number of open source volunteers is furthermore illustrated by the open source directory SourceForge. As of January 2013 sourceForge lists more than 324,000 projects with a total of over 3.4 million developers on the supply side, and on the demand side more than 4 million downloads a day connecting 46 million consumers.[5] Further projects not listed on SourceForge like the Open Directory Project or Wikipedia consist of more than 10,000 contributors (Magnus Cedergren 2003; Glott et al. 2010).

It is for these reasons that the open source phenomenon attracts the interests of scholars, governments and businesses, and substantiates the relevance for management and science. However, surprisingly less attention has been given to understand participation behavior in more detail or provide a coherent community participation framework as addressed in this study.

1 Research Gap and Objective

Open source exhibits a trans-disciplinary phenomenon (von Krogh and Spaeth 2007), bringing together several research approaches from economic and social perspectives. Up to now, participation research in open source has targeted three main aspects: Participants' traits, participation rationales, and participation involvement. Research about participants' traits and rationales was among the first questions asked and reveals participation types, roles, activity, and motivations. Participation involvement has focused on describing community joining, how participants move from periphery to center, and on membership turnover. Nevertheless, current research suffers from certain limitations. A review of participation behavior, and especially participation involvement is missing. Moreover, participation behavior is only explained in separated contributions, but a coherent framework is lacking.

The understanding of participation behavior would enhance our knowledge multifold. It targets the discussion of community membership dynamics, including attracting volunteers and understanding leaving, thus how to keep innovative input and capture their value. It also shed light on the diversity of volunteers, including their idiosyncratic participation benefits and different requirements, hence the heterogeneity of volunteers. Finally, understanding behavior helps to identify root causes of participants' actions and steer volunteers.

We target this unaddressed puzzle and review the existing open source literature in terms of the research questions: What do we know about participation behavior? We highlight the key contributions in the areas participants' traits, participation rationales, and participation involvement. Based on the findings we unite different participation involvement views and move the conversation forward with a coherent participation lifecycle model. This study does not aim to provide a comprehensive review about the open source phenomenon in general,[6]

5 Source: http://sourceforge.net/about, retrieved 14 January 2013.

6 For a more comprehensive review of the emergence of the research field, see for example von Krogh and von Hippel (2006); Dahlander and Gann (2010); Raasch et al. (2012); Crowston et al. (2012).

but to take stock of the latest research in view of open source participation. Thus, this study aims to clarify the lines of open source participation behavior. A first time review concentrating on participation behavior beyond volunteers participation rationales opens the discussion of volunteer behavior from a higher level. We order existing research, compare study findings, and provide a fresh look on participation with a participation lifecycle model. The model integrates certain perspectives and aims to create consensus between these participation views, but also to provide guidance for membership lifecycle considerations. Finally, we stimulate research; one the one hand with the introduction of a new participation phase, on the other hand we propose a research agenda to further strengthen our understanding of participation behavior.
The next section briefly summarizes existing research of participants' traits and rationales.

2 Open Source Participation: Traits and Rationales

Research in open source participation has so far focused on two main aspects: Participants' traits and participation rationales. This section condenses the two fields and points to further literature. Participant's traits describe the participants' types, roles and activity. Participation rationales summarize the key findings of participation motives.

2.1 Participants' Traits: Types, Roles and Activity

"I don't know who these crazy people are who want to write, read and even revise all that code without being paid anything for it at all," writes Glass (1999, p. 104). In 1999, open source was predominantly associated with software development. Open source community members were described as "hackers," including a positive connotation and badge of honor (Raymond 1999; Lakhani and Wolf 2003), participants in a "gift" culture (Bergquist and Ljungberg 2001), or as "geeks" (Pavlicek 2000). Open source conventions were a "meeting place between the informality of geek culture and the buttoned-down business world" (Deckmyn 2002).

Several studies have enriched the picture of a technically skilled contributor working with peers and creating a subculture. Three principal types of contributors have been identified: Individual contributors, non-profit organizations, and for-profit firms.
Individual contributors are participants without affiliations, for example, students, academics and hobbyists. In software communities, students account for 14% and hobbyists for 25% (Hars and Ou 2002). According to Lakhani and Wolf (2003), students represent 20% and academic researchers 7% of the sample. In content communities, Schroer and Hertel (2009) calculated a student share of 32%.
Professional participation is a further participant type (Bonaccorsi et al. 2006; Henkel 2006; Rolandsson et al. 2011). Netscape offered its browser Mozilla under an open source license, but continued to support the project. Linux Kernel 3.2 is written by 1316 developers, including 226 known companies. The top ten firms participating in the Linux Kernel project

account for over 60% of the total contributions; paid developers even account for 75% of all kernel developments (Linux Foundation 2012). Hars and Ou (2002) disclose that 16% of their study respondents are directly paid for their contribution and account for 38% of total contribution efforts. Lakhani and Wolf (2003) report that 53% of survey respondents contribute during paid working time, whereby 70% of those 53% are supported by their supervisors. Hence, approximately 37% of total respondents indicate tolerated firm contributions. With respect to content creation, the literature is silent for firm participation. Yet, some indications of firm support are present. The non-profit Wikipedia foundation is the organizational sponsor of Wikipedia. The Open Directory Project is owned by Netscape, and Freebase is owned by Google. These "men on the inside" examples (Dahlander and Wallin 2006) reveal the strategic influence of firms in open source software communities including its significant amount of contribution and sponsorship.

In terms of demographic diversity, open source participants differ in a wide range of aspects including age, gender and additional educational background.
The age of software contributors ranges from 14 to 73 years (Ghosh et al. 2002), with a mean age of 27 (Ghosh et al. 2002) to 32 years (Oreg and Nov 2008). Gender diversity is strongly biased toward male participation and female programmers remain rare (Rolandsson et al. 2011). The share of male participation ranges from 91% (Hertel et al. 2003) to 98% (Oreg and Nov 2008). The age of open source content participants varies from 16 to 70 years, with a population mean age of 33 years (Schroer and Hertel 2009). Even more extremely distributed are the worldwide Wikipedia study findings, revealing an age range from 10 to 85 years, with a mean age of 25 years (Glott et al. 2010). Male participants in open content represent 75% (Glott et al. 2010) up to 91% of participants (Oreg and Nov 2008).
Regarding educational background, participants are often knowledgeable people (Bryant et al. 2005) with 26 months' experience in contributing to wikis in general, reading 3.4 different wikis daily, and contributing to 1.5 wikis (Majchrzak et al. 2006). The distribution of Wikipedia contributors in terms of education is 33% with secondary education, 26% undergraduates, and 23% Masters and Doctors (of Philosophy) (Glott et al. 2010). In software samples, 51% of contributors had university-level training, 9% had on-the-job training, and 40% were self-taught. Most participants had an undergraduate degree followed by people with a Masters degree (Hars and Ou 2002; Ghosh et al. 2002). Following the discussion, there exists ample heterogeneity within open source participants, but research stops frequently at the a descriptive level.

2.2 Participants Rationales: Motives

Lerner and Tirole (2002, p. 198) are among the first to ask the question: "Why should thousands of top-notch programmers contribute freely to the provision of a public good?" Their question has triggered a plethora of research and encourages studies to clarify online

field support or mundane tasks (Lakhani and von Hippel 2003), progression of users to leaders (Dahlander and O'Mahony 2011), and organizational involvement (Henkel 2006) in terms of why volunteers participate.

Lerner and Tirole (2002) reveal in their qualitative study that benefits for the contributors are essential for participation. Contributors are motivated by opportunities to solve information technology problems and gain reputational benefits. Many contributors later become employees of commercial partnering organizations. Hars and Ou (2002) have conducted one of the first quantitative studies explaining participation in open source projects. Their survey reveals intrinsic motivation and altruism, as well as the role of external rewards, such as expected future returns and personal needs. Lakhani and von Hippel (2003) extended the scope of participation from direct code contribution to user-to-user assistance. Their survey of field support within the Apache community found as reasons reciprocity, helpfulness, reputation, career prospects and intrinsic motives. They indicate that participation could be due to it being part of the job. Following the above seminal publications, further studies support the reported motives for contribution. These publications apply further methods, for example netnography (Janzik et al. 2011), or target different participation stages, for instance enduring participation (Wu et al. 2007). Only a few studies touch on the influence of license regime (Stewart et al. 2006), access regulations (Shah 2006), and interaction with organizations (Jeppesen and Frederiksen 2006) - thus behavior beyond participation rationales. For an extended review of participation rationales see e.g. von Krogh et al. (2012).

In detail studying motivations to contribute reveals two remarkable aspects: A Motivation-effort correlation and a shift in participation rationales.
The correlation between the effort level of participation and identified motivations differs strongly. Social motivations like altruism and ideology are usually present, but only explain little or no participation effort in open source. In other words, even if social motives are prevalent and highly appreciated, these motives do not support increased contribution. Hars and Ou (2002) as well as Bagozzi and Dholakia (2006) prove this relationship in open source software; Nov and Kuk (2008) as well as Schroer and Hertel (2009) highlight this insight in open content projects. In contrast, strong correlations exist between the level of effort invested and own need, reputation, learning, fun, and payment.
Second, a shift in the motivation to participate is discovered. A transformation of the member base is observed from participants who are less profit driven to seekers of stronger commercial benefits, as well as from open source being an ideological movement to it being a serious business opportunity (Rolandsson et al. 2011; Fitzgerald 2006; West 2003; Bonaccorsi and Rossi 2003). The reasons for this transformation are explained by increased firm participation, changing user bases, and the implementation of commercial licenses.
Considering these aspects of user motivation transformation and motivation-effort correlation, altruistic behavior may “at most” be relevant for hobbyists, but cannot explain

significant resource investments (Bonaccorsi and Rossi 2003). The identified motivations need to be reinterpreted in a commercial context (Rolandsson et al. 2011).

3 Participation Involvement: Stages and Steps

Participation involvement has focused on describing how participants move from periphery to center, and on membership turnover. Moving from periphery to center is understood as a socialization process. A new joiner learns community behavior (rules, norms) while working together with other members (Hinds and Bailey 2003). The socialization process includes two critical steps: Initial participation and sustained participation. This two-step approach covers the stages through which a user progresses and reflects the different circumstances in the beginning and with long-term participation. Initial reasons to join and contribute to a community differ strongly from those in subsequent stages (Fang and Neufeld 2009; Dahlander and O'Mahony 2011). Satisfaction of needs and technical contributions are central in the beginning, but sustained participation depends on community identity construction and socialization. In other words, initial (peripheral) participation is open to everybody, but sustained (central) participation is limited to selected core members. This section reviews participation involvement according to initial participation including the community joining script as well as sustained participation and progressing towards center. Also, we describe an community exit option: terminating participation and community leaving.

3.1 Initial Participation and the Community Joining Script

Initial participation describes the "strategies and processes by which new people join the existing community of [software] developers, and how they initially contribute code" (von Krogh et al. 2003, p. 1217). New people start by spending a significant amount of time silently observing the community, an activity referred to as 'lurking',[7] before contributing for the first time to the community. Lurking ranges from a couple of weeks to several months and is explained as the gaining of sufficient understanding to contribute to the technical discussion. A joiner is defined as a person who emerges from a larger group of peripheral participants and eventually earns source code database editing rights (von Krogh et al. 2003). Joining interactions refer to this early contribution stage, describing the steps to reach developer status and join the group of further other developers. Becoming a developer expresses a status change of participants from mainly community-observing to active community participation with code repository modification access (von Krogh et al. 2003; Ducheneaut 2005). To gain developer status a 'joining script' behavior of peripheral community members is identified.

The joining script behavior is defined as the "level and type of activity a joiner goes through to become a member of the developer community" (von Krogh et al. 2003. p. 1227). The level of activity expresses "the intensity of effort until a joiner is granted developer's status" (von Krogh et al. 2003. p. 1227). Contrasting emails of joiners who become developers and joiners who do not become developers, von Krogh et al. reveal significantly different behaviors with respect to the level and type of activity. Future developers tend to report bugs (9.6% to 3.3%), offer bug fixes (4.8% to 1.4%), and participate in general technical discussion (43.0% to 27.6%). On the other hand, list participants give more usage feedback (9.9% to 1.4%), request more help (2.2% to 0%), and refer more often to other projects (4.3% to 0%) than upcoming developers (von Krogh et al. 2003). Combining these activities to a 'joining script' construct, von Krogh et al. propose that contributors who follow the script are more likely to gain developer privileges. Prospective developers start lurking silently to understand the project and learn technical details. Afterwards, they provide hands-on solutions to technical problems rather than wide-ranging feedback. As a developer interviewee confirmed: "I started working with it. I saw these problems. I fixed them. Here they are. That person gets in" (von Krogh et al. 2003, p. 1229).

Ducheneaut (2005) additionally examines contributor socialization within a Python project and shows distinct steps of a developer trajectory. First a user monitors the development mailing list in order to "assimilate the norms and values of the community and analyze the activity of the experts" (p. 349). The second step represents bug reporting and simultaneously suggesting patches. While following this trajectory, the participant gains a reputation for meaningful contributions within the community, socializes, and finally becomes

7 Lurkers are passive, not visible free riders, but also listening subscribers to the development mailing lists. Their level of contributions ranges from "never" (Kollock and Smith 1996) to "minimal" (Nonnecke and Preece 2000). Lurkers account for approximately 90% of all people who use online communities (Nonnecke and Preece 2000). However, while not contributing, lurkers often spread news by word of mouth and use the community product, hence increase community traffic and market share.

a patcher. The third step is obtaining code repository access and directly fixing bugs. The user has now moved from lurking the community to actively developing the community and has reached developer status. The contributor has demonstrated sufficient technical skills to move to a privileged group, progressed in socialization, and is able to "identify important controversies and enroll a network of allies to attack the problem" (Ducheneaut 2005, p. 345). While doing so, the contributor has started the next step in socialization and progresses towards center and 'sustained participation.'

3.2 Sustained Participation and Progressing Towards Center

Motivation to join a community ranges from altruism to one's self-satisfaction to reputation and payment. However, altruistic and idealistic motives hardly correlate with participation efforts (Hars and Ou 2002; Bagozzi and Dholakia 2006; Nov and Kuk 2008; Schroer and Hertel 2009). Furthermore, initial conditions for participation do not predict long-term participation (Fang and Neufeld 2009), and 80% of open source software projects fade away (Colazo and Fang 2009). Communities rely on trustworthy key persons, but as everyone can join, even under different avatars, the participants' potential is hard to evaluate. Communities therefore give full access and key roles only after an evaluation period and assimilate joiners gradually into the project (O'Mahony and Ferrarao 2007; Preece and Shneiderman 2009). A two-tier developer structure is observed: peripheral developers and core developers (Lee and Cole 2003; Fang and Neufeld 2009; Ducheneaut 2005; Dahlander and O'Mahony 2011). Peripheral developers report bugs, suggest changes, participate in technical discussions and provide pieces of content. They accomplished the joining script and now have first code repository access to fix bugs.[8] Core developers have full code repository access, oversee modules, moderate the community, and craft the project. They contribute a substantial amount technically as well as additionally holding administrative roles and lateral authority. Core developers' driving motivations to participate turn out to be different. Long-term participants are driven by enjoyment of programming and community interaction; in contrast, short-term participants are driven by need and use value (Shah 2006). Bagozzi and Dholakia (2006) support these findings; novice participants are typically driven by extrinsic motivation, whereas experienced participants are self-motivated by their enjoyment and by being part of the community.

While a peripheral developer has already gained some reputation for meaningful suggestions and parallel technical contributions, to become a core developer a common developer has to "demonstrate a higher level of mastery by taking charge of a sub-module" (Ducheneaut 2005, p.351). After gaining first repository access, subsequent steps for successfully progressing towards the center are taking charge of a module-size project, and developing this project. These steps include a much greater interaction with the community, gathering

8 Phases may be more nuanced. Examples and sub-phase steps are reported by Ducheneaut (2005) and include, for example, direct code repository access or contributing via admins regarding development. However, due to the flat hierarchies observable in open source communities, this paper simplifies these steps to elemental levels and key principles.

support for the project, and defending it publicly. Obtaining the approval of the core members for module integration represents the next step. At this stage, the developer is very likely to gain full code repository access and has connected intensively with the core developers and the entire community. Connecting within the community is essential to gaining lateral authority and progressing to a core developer position (Dahlander and O'Mahony 2011). Technical contribution explains the progression of individuals at an early stage, but not at a later stage after gaining developer status. To acquire authority roles beyond the developer status, coordination work and the spanning of subproject communication boundaries are significant predictors to further progress (Dahlander and O'Mahony 2011).

Summarizing the community integration process, the participant progresses from observing experts and assimilating community norms (lurking), to providing significant technical contribution and ongoing community interaction (developing), to emerge as a go-to identity and being responsible for modules (administrator). This socialization process of building an identity and learning from peers is found in software (Fang and Neufeld 2009; Qureshi and Fang 2011) and content (Bryant et al. 2005) communities. Nevertheless, a participant can also terminate participation and leave the community.

3.3 Terminating Participation and Community Leaving

Membership retention represents an important component for open source communities, which can explain community failure or prosperity (Butler 2001; Oh and Jeon 2007). Half of the registered open source community members stop contributing after their initial posting (Ducheneaut 2005), and most developers, even core members, leave the project within one year (Shah 2006). Community participants, in contrast to traditional firms' employees, do not have a formal contract with the community. They are free to leave and can easily vote with their feet.

From a social capital perspective it is argued that the more members are present within a network, the more potential and assets can be mobilized, and the more valuable it is. Social capital and the naturally evolving ties represent an essential aspect in open source projects. Tan et al. (2007) find that the stronger the cohesive member ties, the more productive the group is. If members leave a community, the network becomes smaller and social capital, including contributing resources and cohesive ties, is reduced. Members depending on each other notice the lack of a connection as soon as a partnering role is no longer occupied. Participation rationales indicate that community involvement is due to existing community members and a sense of belonging to the community. Departure of (core) members may signal dissatisfaction, reduced commitment to the community project and project failure, triggering other members to rethink their participation (Jones 1986). Oh and Jeon (2007) prove these rationales within an open source software community. Supported by herding theory, they discover a snowball effect, that the decisions of members to leave the community is heavily influenced by neighboring members. Besides the members and their

ties, leavers additionally take away the gained knowledge and experience. Even when explicit knowledge is documented, tacit knowledge vanishes. Departure of members hence reduces the benefits and contribution motivation of the remaining participants (Butler 2001).

However, positive support for membership turnover exists, too. Membership fluidity facilitates a dynamic exchange of resources, including cognitive verve in terms of creativity, passion, and social identity (Faraj et al. 2011). Turnover allows new members to join and experienced developers can progress to core developer roles. Even though virtual communities are not limited in size, core developer roles are rare, and prospective joiners avoid high communication levels and communities that are too full (Butler 2001; Kuk 2006). Ransbotham and Kane (2011)[9] offer two empirical findings for the distinct phases of knowledge creation and knowledge retention. First, they provide evidence that moderate levels of turnover correlate positively with project success. While some retention stability is required to keep the community knowledge, turnover facilitates the gaining of new knowledge for the community, because members appear to concentrate on content creation but put less effort into preserving that content. Second, their longitudinal study of featured Wikipedia articles reveals a curvilinear relationship of effective collaboration, in particular between the turnover of Wikipedia editors and the quality of an article. More experienced editors increase the likelihood of raising an article in quality up to a specific point, but after that, editors with average experience decrease the quality of the article.
Concluding the discussion, membership turnover is an essential element in the community joining process and the individual membership lifecycle. Detailed knowledge of why participants leave a community provides direct insights into areas for improvement in order to control member retention, understand member behavior, and derive implications for successful community management.

This section has described certain participation involvement stages and steps. However, the stages and steps are isolated, an overview is missing. We aim to combine the stages and steps and propose a participation lifecycle model.

4 Developing a Participation Lifecycle Model

4.1 Developing a Contribution Framework

Combining the previous phases, we introduce the contribution framework. The framework incorporates the previously described phases through which a community contributor progresses and additionally includes the steps describing how to progress. Currently, the literature is silent with respect to proposing a contribution framework. Research on community joining is fragmented and progressing is discussed in isolated researc stages.

9 Ransbotham and Kane (2011) provide a detailed overview of antecedents and consequences of community leaving and retention, also with regard to organizational theory beyond open source.

Only single phases or steps – for example how to progress from one phase to another – are described. We order research and address that gap with a contribution model. The model does not only describe single phases, but untangles and connects previous research into a coherent framework.

A notable example proposing a framework is the 'Reader-to-Leader Framework' (Preece and Shneiderman 2009). The framework provides four distinct phases (Reader, Contributor, Collaborator, Leader) describing how a user becomes a leader. While steps back and forth between the phases are possible and the phase descriptions highlight participant activities and key motivations within the phase, the model shows that only a fraction of users progresses to the next phase. It fails to describe how and why a user progresses to the next step. It implies that motivations change and accordingly a user progresses due to altered usability and sociability factors that influence a certain phase. Moreover, it lacks a leaving phase, and thus an important phase for explaining membership turnover and retention.

Reviewing the above discussion about initial participation and sustained participation, a participant passes certain phases. Participants commence by lurking a community, followed by active contribution (developing), and finally progress to administrator status and governing the community. These phases are connected by steps, enabling progress to the next phase. Progressing from lurking to contributing is explained by the 'joining script' construct (initial contribution), and advancing to administrator status by enrollment of key allies with respect to coordinating work and gaining lateral authority (sustained participation). Thus, while technical contributions are important to receiving developer status, coordination work and spanning subproject communication boundaries are key elements to progressing to administrator status. Consequently, socialization starts as soon as a user decides to follow a community by (unconsciously) learning norms and values expressed in community behavior. Combining these steps and phases leads to the contribution model. The model is illustrated in figure 1 below. The steps between the phases are represented by pentagons and symbolize the connections between phases. Two types of steps exist: promotion step and exit step. The exit step includes leaving. Leaving can occur during all phases, thus every phase is connected to the leaving step. The promotion steps include the joining script and lateral authority progression. The phases are lurking, developing, and administrating. Socialization takes place during all phases and increases with progression toward the center.

From a theory point of view, the model, and in particular progressing to the center of the community and identity construction, is based on 'situated learning theory' (Lave and Wenger 1991). Situated learning theory explains socialization and increasing community interactions, including learning from each other and building up an identity. Identity construction is "the process of being identified within the community," and 'situated learning' is the "process of acting knowledgeably and purposefully in the world" (Fang and Neufeld

2009, p. 9). Especially by learning from higher ranked developers, joiners gain valuable community insights. For example, socializing with core developers strengthens their skills and joiners can get social support up to receiving patronage for subprojects (Qureshi and Fang 2011; Brown and Duguid 1991). As Lee and Cole report, "the learning process uses the cultural artifacts as an educational tool. On the one hand, the publicly archived criticisms help individuals to learn from their peers how to improve their next submission. On the other hand, they serve as documented texts to train other developers observing the peer review process. As developers learn from their own and others' prior successes and failures, they can sort themselves into tasks appropriate to their skills, move up to more challenging tasks, and/or generate better variations of the source code" (Lee and Cole 2003, p. 644).

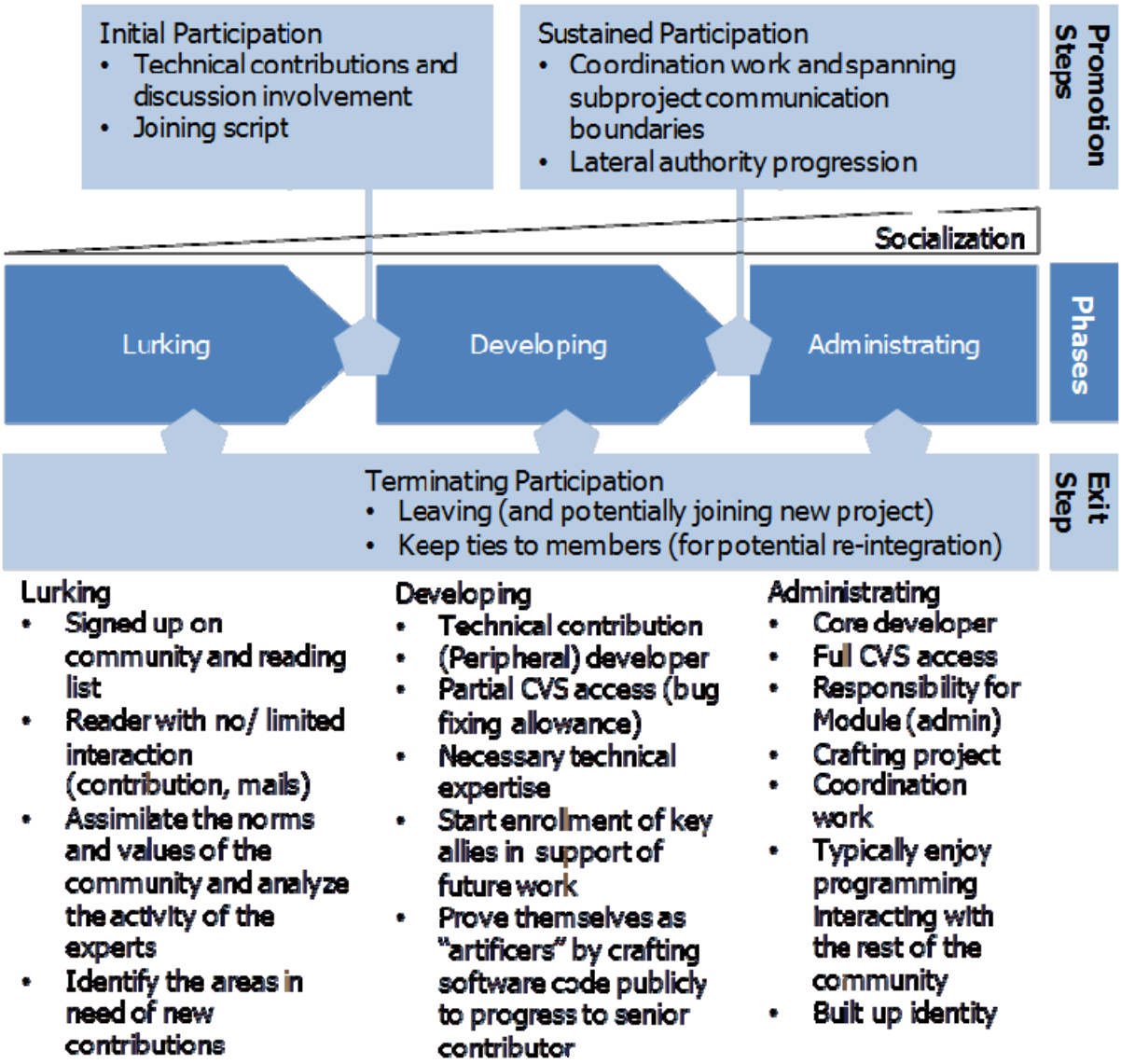

Figure 1: Contribution Framework of Open Source Projects

The above model covers the process from lurking to leading to leaving, and combines essential elements discussed within open source research with respect to the time users are

associated with the community. However, anecdotal evidence and wider literature points to a phase even before a user is affiliated with the community: a pre-participation phase.

4.2 Extending the Contribution Framework with Pre-Participation

Reflecting upon open source from a broader perspective and beyond contribution activities, point to activities before being affiliated with the community. Trusov et al. show that word of mouth has a positive effect on member acquisition. Word of mouth "referrals have substantially longer carryover effects than traditional marketing actions and produce substantially higher response elasticities" (Trusov et al. 2009, p. 90). Hahn et al. (2008) reveal the relationship of prior collaborative ties as an explanation for project selection. The project selection likelihood grows if the prospective joiner is already familiar with the new project founder, being a past collaborator. Similarly, Kuk (2006) reports highly strategic project selection by users in order to succeed. Users enter a reciprocal interdependent relationship in order to connect to further developers. Shah (2006) supports community evaluation, describing that strangers familiarize themselves with the specific project context and make conscious decisions to join and use a community. Various externally observable community characteristics (project tenure, size, intended audience, types of software, and programming language) are instrumental to project success (Crowston and Scozzi 2002).
Including above activities into the contribution framework calls for a further step. The integration of an 'awareness' concept before the lurking phase is required: A pre-participation step. A pre-participation step by users takes place before the user joins a community or starts lurking. The step includes community identification and evaluation. The awareness step thus catches actions by users in choosing a community. Choosing then include gathering information about a community, exploration initiatives (e.g. search engine usage, friend referral), and initial community evaluation. As soon as a user comes back regularly to the community and intensifies community observation, the user progresses to the lurking phase and is following the community.

Moreover, the potential of a prospective joiner to select a community exists before lurking; a personal 'scratching' exists. As Raymond reports, "every good work of software starts by scratching a developer's personal itch" (Raymond 1999, section: 'The Mail Must Get Through'). Individual motivations to participate in a community trigger a community project selection and spark a fire for community joining activities. Consequently, we term the phase before lurking the 'scratching' phase. The scratching phase describes the situation where a user is not following, and has not chosen a community yet, but already carries the intention and motives to participate, triggering subsequent actions. The phase hence incorporates the unmatched need recognized by the user as well as the problem awareness. The phase takes into account the intention of users to do an activity independent of the subsequent actually-conducted behavior. The phase represents the actions of the user while outside the community, in terms of not following and not being registered. Thus, the scratching phase

precedes the lurking phase. Both phases are connected by actions to select the community – the awareness step – and extend the contribution model to a participation lifecycle model.

4.3 Introducing the Participation Lifecycle Model

The previously introduced contribution model describes the activities of a member while within the community, or at least connected by observing the community. The above rationales regarding a community awareness step, including a scratching phase, suggest an extension of the contribution model to before the user's connecting with the community. Merging the awareness steps and scratching phase with the contribution model results in an end-to-end participation lifecycle model. The entire participation lifecycle model is illustrated in figure 2. The contribution model is extended by the awareness step and the scratching phase.

The model shows an unidirectional flow indicating the member progression, however leaving may take place during all phases; also, stops and steps backwards can occur. Leaving ranges from stepping back from more advanced roles to more initial roles (e.g. from developer to lurker), to losing affiliations or a complete exit out of the community. The phases do not symbolize a one-way road, but represent essential stages through which a user moves gradually back and forth. Moreover, there may be certain key steps where a user is likely to be recognized as being promoted (e.g. having received initial code repository access or module responsibility). However there exists no formal process or credential to reach a certain phase.

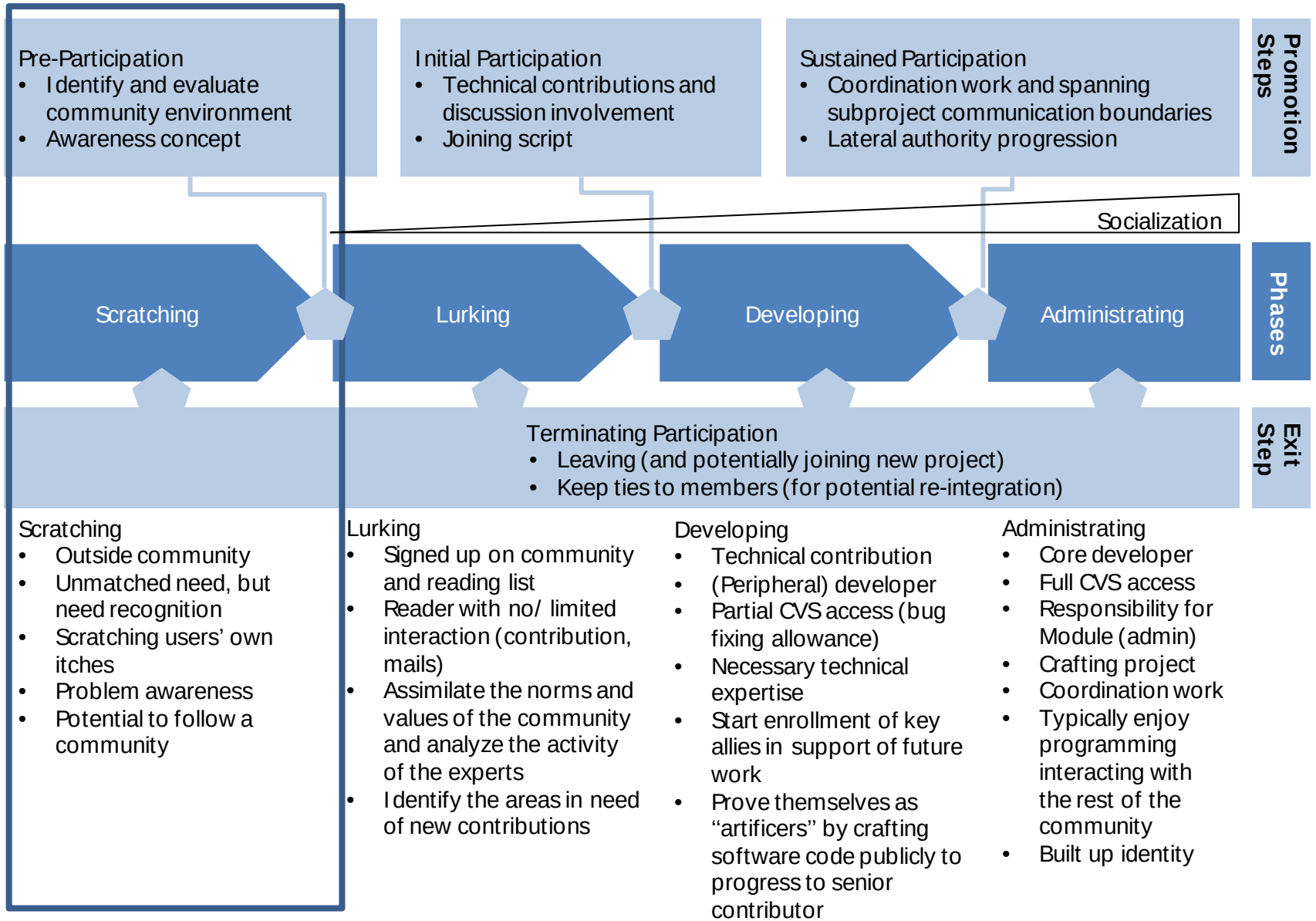

Figure 3: Participation Lifecycle Model

Progressing to a certain phase depends on individual skills, in particular technical and socialization skills, invested effort driven by participation rationales, and previous member experience. Shah (2006), who draws on research on motivation by Roberts et al. (2006) and user background by Hertel et al. (2003), supports the heterogeneous progressing of members. Due to considerable variations in user characteristics, the socialization process varies for different users. New joiners with less experience may need more time to grasp community norms and to socialize with other members (Shah 2006). Beginners may lurk silently for a longer time. Herraiz et al. (2006) as well as Shibuya and Tamai (2009) provide empirical evidence. Herraiz et al. discover two different joining patterns: one exhibits a "sudden integration", while the second follows a "step-by-step" pattern. Step-by-step integration is observed for volunteer participants. Sudden integration is observed for firm participants and only in the coder project. Shibuya and Tamai identify the same patterns and reveal an even more determining aspect than being hired. Hired participants are already familiar with the project and are known within the community. They have previously worked together in other (sub) projects. These findings point to the relevance of a pre-project contribution phase and leaving. Former collaborators quit one project (leaving phase), but still carry motives to participate (being paid, scratching phase). Former project ties of users and community experience enable them to progress differently compared to strangers, and significantly shorten, or even skip, the lurking phase. Leaving does not necessarily mean losing connection with selected developers, but rather stepping back from stronger contribution while keeping future opportunities and developer networks open. Consequently, leaving and scratching do not represent opposite phases, but can be neighbors connected by activities out of the community (e.g. word of mouth) and can close the loop within the entire participation lifecycle.

5 Conclusion

To the best of our knowledge, this is the first review considering open source participation. In particular, we go beyond highlighting research on individuals' traits and participation rationales and point to individual member behavior and a conceptual model of participation involvement. We contribute with the development of a participation lifecycle model. Up to now, certain unconnected concepts of joining, socialization and lateral authority exist within the literature. This study orders these concepts and develops a coherent participation lifecycle model. The model describes certain phases with activities, as well as concepts, that influence the member. It highlights the social interactions strongly influencing a member. We introduce a pre-joining phase with conscious decision processes addressing actions before a user is affiliated with the community. The phase fills a gap and connects leaving and joining actions. While the joining script (von Krogh et al. 2003) describes the progress from lurking to coding, and the lateral authority concept (Dahlander and O'Mahony 2011) describes the steps towards administrating, the pre-joining actions represent the actions from scratching to

lurking, a joiner awareness concept. The lifecycle model can guide researchers analyzing community participation. We draw their attention to carefully considering the participation phases and influences upon them, especially in view of interaction effects, participation phases and specifying research localizations. Future research may benefit from the lifecycle model for an end-to-end understanding of the participation process. Moreover, the participation lifecycle model creates consensus in so far isolated and cluttered research, integrates certain participation perspectives, and solves the puzzle of an end-to-end membership perspective. It includes steps and phases describing the member behavior in joining, contributing, and leaving a community. While the model combines several loose research contributions, introduces a new phase, and provides an overview of the socialization process, it still leaves open future research avenues and has several limitations. We focus strongly on open source innovation, and thus limit the research scope to this field. The model is conceptual and not empirical tested - but based on established research. We do not use a strict bibliometric analysis for review, but use forward and backward citations to find appropriate papers within the yet limited number of available papers.[10] Thus, future research is needed for model validation and detailing.

6 Future Research Directions

Besides the consolidation of existing research and the development of a participation lifecycle model, we also aim to stimulate research on individual behavior in open source innovation. Currently the literature has just sporadically considered member behavior. We point in particular to four areas: The participation lifecycle model, participation rationales, open source transformation, and fairness.

We start to ameliorate the lack of behavioral research by ordering research and proposing a participation lifecycle model focusing on member behavior. The model combines anchored concepts of open source research and includes new phases and steps. However, further research is required, on the one hand, for model corroboration, and on the other hand, for completing the phases with more characteristics. For example, what exactly happens within the awareness phase and how do users decide in favor of a community? What are the preconditions for joining and what are the barriers for participation? Do users reflect on their contribution and how carefully do joiners consider the attributes of communities in their joining decisions? Additionally, more empirical support is needed to strengthen the model and detail the transitional steps. Only a few papers (Oh and Jeon 2007; Harhoff and Mayrhofer 2010) consider why members leave a community or change to other communities. Thus ex-post contribution decisions may help us to understand participation and sustain competitive strength.

[10] As of February 2013, Google Scholar provides for the research query ["open source"+"participation behavior"] only 113 results, including results not hitting the intended study scope.

Member behavior can also enhance understanding in regards to participation rationales. Research to identify the motivation for participation was among the very first questions asked. Most behavioral research concentrates on why do members participate and on the relation between participation motivation and contribution actions, but there are further open areas for future research. Interaction between motivations and behavior when subject to contradictory motivations are less taken into account. Firms increasingly utilize open source communities and user motivation shifts from being mainly altruistically and idealistically driven to being more benefit-oriented. Both aspects create a tension within the community due to the different stakeholders and intentions. While 'crowding-out' effects are considered, little is known about how contributors decide if they embody several (contrasting) motivations and responsibilities. How does a contributor behave who aims to gain reputation within the community, but at the same time is employed at a firm not allowing contributions? How are conflicting incentives satisfied?[11] Analyzing user motivation beyond the motivational level is rare. Community heterogeneity (in types of users and their motivation) is mostly neglected but is essential as it triggers participation. Furthermore, von Krogh et al. (2012) highlight the need to link motivations to institutional settings. Motivations should not be analyzed independently, but with respect to contextual settings. They address this research gap commenting that "most of this work is recent and difficult to categorize" (p. 645), but at the same time, social practices are strongly related to contributors' motivations and individual behavior.

The ongoing utilization and expansion of open source, including its communities, creates new challenges. A transformation from ideology-driven participants to a large number of commercially motivated users and the participation of firms is one example (West 2003; Bonaccorsi and Rossi 2003; Fitzgerald 2006; Rolandsson et al. 2011). Most studies treat contributors and communities as one-dimensional, ignoring the presence of differently motivated contributors, and do not break down the groups or distinguish between them. Another challenge is simply the availability of more communities (Oh and Jeon 2007; Dahlander and Magnusson 2005). Decisions under the assumption of available opportunities and particularly different participation options are not yet considered. This research gap increases due to the momentum of open source, the opening of firms, and the ongoing emergence of communities. A "vast number of projects competes for the attention and interest of the developers and users" (Dahlander and Magnusson 2005, p. 489) and rivalry

11 Notable exception is the work by Henkel (2006) who analyses the selective revealing of employed contributors, and Rolandsson et al. (2011) who examine programmers' behavior where both firm and community-based modes of production exist.

for donated labor is increasing (West and O'Mahony 2005). In the same vein, Harhoff and Mayrhofer propose that "competition for particularly productive or influential community members will increase, and that migration of important users will be an important phenomenon in community-driven innovation" (Harhoff and Mayrhofer 2010, p. 34). How do volunteers behave as soon as there are alternatives? What are preconditions and preferred factors for participation? Seen from another viewpoint, the question about obstacles and minimum requirements for participation remains largely unanswered. Preconditions exist in the form of low costs for the contributors, modular architecture in bite-size pieces, and low costs of integration (Tapscott and Williams 2008). Participation barriers exist in ease of coding, altering and integrating modules, variability of coding language, and independent working of modules (e.g. von Krogh et al. 2003; Glott et al. 2010). Legal restrictions imposed by organizations hinder user innovation (Braun and Herstatt 2009). However, gaining insights into further, subtle mechanisms for (non-) participation provides insights into how to (un-) trigger innovation and (un-) build barriers. What are the conditions leading to free revealing and private collective innovation? What are the minimum levels for participation? Moreover, much research has concentrated on lighthouse projects like Linux or Apache. These projects often belong to established business applications and are populated with a disproportionately high amount of paid contributors.[12] It is questionable whether these top projects are representative of the entire open source domain, or if they represent the tip of the iceberg. Research in open content is still nascent. Research is rare or the studies concentrate on one research object, Wikipedia. What about (neglected) entertainment communities? Are software, content, business and entertainment communities alike?

Fairness, understood as interpersonal relative payoff (Loewenstein et al. 1989), seems to influence volunteering in an open collaborative innovation environment (Harhoff and Mayrhofer 2010; Nov and Kuk 2008; Shah 2006), and organizational behavior (Colquitt et al. 2006). However, participation research widely ignores this aspect. As "the economic environment determines whether the fair types or the selfish types dominate equilibrium behavior" (Fehr and Schmidt 1999, p. 817), an open private-collective collaboration may represent an interesting example for studying Pareto efficiency conditions, in regards to increased individual and social welfare. Within the realm of heterogeneous actors aiming at contrary objectives, integrating behavioral aspects seems to be a fruitful field for further research.

12 Considering Linux as the most business-oriented project with a high degree of commercial input and Apache closely following.

References

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

Bergquist, M., and Ljungberg, J. 2001. "The power of gifts: organizing social relationships in open source communities," *Information Systems Journal* (11:4), pp. 305–320.

Bonaccorsi, A., and Rossi, C. 2003. "Why Open Source software can succeed: Open Source Software Development," *Research Policy* (32:7), pp. 1243–1258.

Bonaccorsi, A., Giannangeli, S., and Rossi, C. 2006. "Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry," *Management Science* (52:7), pp. 1085–1098.

Braun, V., and Herstatt, C. 2009. *User-Innovation: Barriers to Democratization and IP Licensing*, New York: Routledge.

Brown, J. S., and Duguid, P. 1991. "Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation," *Organization Science* (2:1), pp. 40–57.

Bryant, S. L., Forte, A., and Bruckman, A. 2005. "Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia," in *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, Sanibel Island, Florida, USA: ACM, pp. 1–10.

Butler, B. S. 2001. "Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures," *Information Systems Research* (12:4), pp. 346–362.

Colazo, J., and Fang, Y. 2009. "Impact of license choice on Open Source Software development activity," *Journal of the American Society for Information Science and Technology* (60:5), pp. 997–1011.

Colquitt, J. A., Scott, B., Judge, T., and Shaw, J. 2006. "Justice and personality: Using integrative theories to derive moderators of justice effects," *Organizational Behavior and Human Decision Processes* (100:1), pp. 110–127.

Crowston, K., and Scozzi, B. 2002. "Open source software projects as virtual organisations: competency rallying for software development," *IEE Proceedings Software* (149:1), pp. 3–17.

Crowston, K., Wei, K., Howison, J., and Wiggins, A. 2012. "Free/Libre open-source software development: What we know and what we do not know," *ACM Computing Surveys* (44:2), pp. 7:1–7:35.

Dahlander, L., and Gann, D. M. 2010. "How open is innovation?" *Research Policy* (39:6), pp. 699–710.

Dahlander, L., and Magnusson, M. 2005. "Relationships between open source software companies and communities: Observations from Nordic firms," *Research Policy* (34:4), pp. 481–493.

Dahlander, L., and O'Mahony, S. 2011. "Progressing to the Center: Coordinating Project Work," *Organization Science* (22:4), pp. 961–979.

Dahlander, L., and Wallin, M. W. 2006. "A man on the inside Unlocking communities as complementary assets," *Research Policy* (35:8), pp. 1243–1259.

Deckmyn, D. 2002. *Suits, Geeks Seek Open Source Entente*. http://www.computerworld.com.au/article/65320/suits_geeks_seek_open_source_entente/. Accessed 1 October 2012.

Ducheneaut, N. 2005. "Socialization in an open source software community: A socio-technical analysis," *Computer Supported Cooperative Work (CSCW)* (14:4), pp. 323–368.

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

Faraj, S., Jarvenpaa, S., and Majchrzak, A. 2011. "Knowledge collaboration in online communities," *Organization Science* (22:5), pp. 1224–1239.

Fehr, E., and Schmidt, K. 1999. "A theory of fairness, competition, and cooperation," *The Quarterly Journal of Economics* (114:3), pp. 817–868.

Fitzgerald, B. 2006. "The Transformation of Open Source Software," *MIS Quarterly* (30:3), pp. 587–598.

Ghosh, R. A., Glott, R., Krieger, B., and Robles, G. 2002. *Free/Libre and Open Source Software: Survey and Study.Part IV: Survey of Developers*. http://www.flossproject.org/report/Final4.htm. Accessed 30 September 2012.

Glass, R. L. 1999. "The loyal opposition of open source, Linux…and hype: Software, IEEE," *Software, IEEE* (16:1), pp. 126–127.

Glott, R., Schmidt, P., and Ghosh, R. A. 2010. *Wikipedia Survey – Overview of Results*. http://www.wikipediasurvey.org/docs/Wikipedia_Overview_15March2010-FINAL.pdf. Accessed 30 October 2012.

Hahn, J., Moon, J. Y., and Zhang, C. 2008. "Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties," *Information Systems Research* (19:3), pp. 369–391.

Harhoff, D., and Mayrhofer, P. 2010. "Managing User Communities and Hybrid Innovation Processes: Concepts and Design Implications," *Organizational Dynamics* (39:2), pp. 137–144.

Hars, A., and Ou, S. 2001. "Working for Free? Motivations for Participating in Open-Source Projects," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34*, Washington, DC: IEEE Computer Society, pp. 25-39.

Healy, K., and Schussman, A. 2003. "The ecology of open-source software development," Department of Sociology. University of Arizona, Tucson.

Henkel, J. 2006. "Selective revealing in open innovation processes: The case of embedded Linux," *Research Policy* (35:7), pp. 953–969.

Herraiz, I., Robles, G., Amor, J. J., Romera, T., and Barahona, J. M. G. 2006. "The processes of joining in global distributed software projects," in *Proceedings of the 2006 international workshop on Global software development for the practitioner*, Shanghai, China: ACM, pp. 27–33.

Hertel, G., Niedner, S., and Herrmann, S. 2003. “Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel,” *Research Policy* (32:7), pp. 1159–1177.

Hinds, P., and Bailey, D. 2003. “Out of sight, out of sync: Understanding conflict in distributed teams,” *Organization Science* (14:6), pp. 615–632.

Hippel, E. A. von, and Krogh, G. von 2003. “Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science,” *Organization Science* (14:2), pp. 209–223.

Janzik, L., Herstatt, C., and Raasch, A.-C. 2011. “Warum Kunden in Online-Communities innovieren: Ergebnisse einer Motivanalyse,” *Kundenintegration 2.0* (5), pp. 47–81.

Jeppesen, L. B., and Frederiksen, L. 2006. “Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments,” *Organization Science* (17:1), pp. 45–63.

Jones, C. 1986. *Programming productivity*, New York: McGraw-Hill.

Kollock, P. 1998. “The economies of online cooperation: Gifts and public goods in cyberspace,” in *Communities in cyberspace*, M. A. Smith, and P. Kollock (eds.), London: Routledge, pp. 220–239.

Kollock, P., and Smith, M. A. 1996. “Managing the Virtual Commons: Cooperation and Conflict in Computer Communities,” in *Computer-Mediated Communication: Linguistic, Social, and Cross Cultural Perspectives*, S. C. Herring (ed.), Amsterdam: John Benjamins, pp. 109–128.

Krogh, G. von, and Hippel, E. A. von 2006. “The Promise of Research on Open Source Software,” *Management Science* (52:7), pp. 975–983.

Krogh, G. von, and Späth, S. 2007. “The open source software phenomenon: Characteristics that promote research,” *The Journal of Strategic Information Systems* (16:3), pp. 236–253.

Krogh, G. von, Haefliger, S., Späth, S., and Wallin, M. W. 2012. “Carrots and rainbows: Motivation and social practice in open source software development,” *MIS Quarterly* (36:2), pp. 649–676.

Krogh, G. von, Späth, S., and Lakhani, K. R. 2003. “Community, joining, and specialization in open source software innovation: a case study: Open Source Software Development,” *Research Policy* (32:7), pp. 1217–1241.

Kuk, G. 2006. “Strategic interaction and knowledge sharing in the KDE developer mailing list,” *Management Science* (52:7), pp. 1031–1042.

Lakhani, K. R., and Hippel, E. A. von 2003. “How open source software works: “free” user-to-user assistance,” *Research Policy* (32:6), pp. 923–943.

Lakhani, K. R., and Tushman, M. 2012. “Open Innovation and Organizational Boundaries: The Impact of Task Decomposition and Knowledge Distribution on the Locus of Innovation,” *HBS Working Papers* 12-57, Harvard Business School, Cambridge.

Lakhani, K. R., and Wolf, R. 2005. “Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects,” in *Perspectives on Free*

and Open Source Software, J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani (eds.), Boston: MIT Press.

Lave, J., and Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*, Cambridge: Cambridge University Press.

Lee, G. K., and Cole, R. E. 2003. "From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development," *Organization Science* (14:6), pp. 633–649.

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

Linux Foundation 2012. *The Linux Foundation Releases Annual Linux Development Report*. http://www.linuxfoundation.org/news-media/announcements/2012/04/linux-foundation-releases-annual-linux-development-report. Accessed 1 October 2012.

Loewenstein, G., Thompson, L., and Bazerman, M. 1989. "Social utility and decision making in interpersonal contexts," *Journal of Personality and Social Psychology* (57:3), p. 426.

Majchrzak, A., Wagner, C., and Yates, D. 2006. "Corporate wiki users: results of a survey," in *Proceedings of the 2006 international symposium on Wikis*, New York, NY, USA: ACM, pp. 99–104.

Nonnecke, B., and Preece, J. 2000. "Lurker demographics: counting the silent," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, The Hague, The Netherlands: ACM, pp. 73–80.

Nov, O., and Kuk, G. 2008. "Open source content contributors' response to free-riding: The effect of personality and context," *Computers in Human Behavior* (24:6), pp. 2848–2861.

Oh, W., and Jeon, S. 2007. "Membership herding and network stability in the open source community: The Ising perspective," *Management Science* (53:7), pp. 1086–1101.

O'Mahony, S., and Ferraro, F. 2007. "The Emergence of Governance in an Open Source Community," *Academy of Management Journal* (50:5), pp. 1079–1106.

Oreg, S., and Nov, O. 2008. "Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values," *Computers in Human Behavior* (24:5), pp. 2055–2073.

Pavlicek, R. 2000. *Embracing Insanity: Open Source Software Development*, Indianapolis: Sams.

Preece, J., and Shneiderman, B. 2009. "The reader-to-leader framework: Motivating technology-mediated social participation," *AIS Transactions on Human-Computer Interaction* (1:1), pp. 13–32.

Qureshi, I., and Fang, Y. 2011. "Socialization in open source software projects: A growth mixture modeling approach," *Organizational Research Methods* (14:1), pp. 208–238.

Raasch, A.-C., Lee, V., Späth, S., and Herstatt, C. 2012. "The rise and fall of interdisciplinary research: The case of open source innovation," *Working Paper,* Hamburg University of Technology, Hamburg.

Ransbotham, S., and Kane, G. 2011. "Membership turnover and collaboration success in online communities: Explaining rises and falls from grace in Wikipedia," *MIS Quarterly* (35:3), pp. 613–627.

Raymond, E. S. 1999. "Articles - The Cathedral and the Bazaar," *Knowledge, Technology and Policy* (12:3), pp. 23–49.

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

Rolandsson, B., Bergquist, M., and Ljungberg, J. 2011. "Open source in the firm: Opening up professional practices of software development," *Research Policy* (40:4), pp. 576–587.

Schroer, J., and Hertel, G. 2009. "Voluntary Engagement in an Open Web-Based Encyclopedia: Wikipedians and Why They Do It: Media Psychology," *Media Psychology* (12:1), pp. 96–120.

Setia, P., Rajagopalan, B., Sambamurthy, V., and Calantone, R. 2012. "How Peripheral Developers Contribute to Open-Source Software Development," *Information Systems Research* (23:1), pp. 144–163.

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

Shibuya, B., and Tamai, T. 2009. "Understanding the process of participating in open source communities," in *ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009. FLOSS '09.,* Vancouver, BC, pp. 1–6.

Stewart, K. J., Ammeter, A. P., and Maruping, L. M. 2006. "Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects," *Information Systems Research* (17:2), pp. 126–144.

Tan, Y., Mookerjee, V., and Singh, P. 2007. "Social capital, structural holes and team composition: Collaborative networks of the open source software community," in *Proceedings of the International Conference 2007*.

Tapscott, D., and Williams, A. 2008. *Wikinomics: How mass collaboration changes everything*, New York: Portfolio Trade.

Trusov, M., Bucklin, R. E., and Pauwels, K. 2009. "Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site," *Journal of Marketing* (73:5), pp. 90–102.

West, J. 2003. "How open is open enough?: Melding proprietary and open source platform strategies: Open Source Software Development," *Research Policy* (32:7), pp. 1259–1285.

West, J., and O'Mahony, S. 2004. "Contrasting Community Building in Sponsored and Community Founded Open Source Projects," *San Jose State University, Faculty Publications* Paper 2, San Jose.

Williams, R. L., and Cothrel, J. 2000. "Four Smart Ways To Run Online Communities," *Sloan Management Review* (41:4), pp. 81–91.

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