Decision Support Systems 64 (2014) 142–151

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Uncovering the relationship between OSS user support networks and OSS popularity

Juliana Sutanto [a,*], Atreyi Kankanhalli [b,1], Bernard C.Y. Tan [b,2]

[a] Department of Management, Technology, and Economics, ETH Zurich, Weinbergstrasse 56/58, 8092 Zurich, Switzerland
[b] Department of Information Systems, School of Computing, National University of Singapore, 15 Computing Drive, Singapore 117418, Singapore

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form 29 May 2014
Accepted 31 May 2014
Available online 10 June 2014

Keywords:
OSS user support
Structural embeddedness
Junctional embeddedness
In-degree centralization
Betweenness centralization
OSS popularity

ABSTRACT

The open source model of software development has become an attractive alternative to the traditional proprietary approach. However, the incomplete understanding of the phenomenon has continued to prompt researchers to investigate factors that could increase the use and popularity of open source software (OSS). While a key antecedent for OSS use highlighted in the previous literature is the software quality, we propose that effective online user support is also necessary to increase its popularity. As an understudied area, this paper seeks to understand the role of online user support networks in facilitating OSS use. Based on the network embeddedness theory, it suggests that properties of the user support network i.e., variation in structural and junctional embeddedness, measured as the in-degree and betweenness centralizations respectively, would affect OSS popularity in terms of the number of active users and downloads of the software. Testing on a sample of 176 OSS projects from Sourceforge.net, we showed that a negative quadratic relationship exists between the variation in structural embeddedness of the OSS user support network and the software popularity. Further, as hypothesized, the variation in junctional embeddedness was found to positively impact the OSS popularity. Theoretical and practical implications of the study are discussed.

1. Introduction

Open source software (OSS) is having a substantial impact on software and its development processes with a large number of software products now containing open source components. OSS is unique in its production, modification, and distribution. A key characteristic of OSS is that it is often developed in a public and collaborative manner. Also, owners typically make their source code available and impose special licenses that allow users to access, change, and sometimes even distribute the software [33,54]. However, the high failure rate of OSS projects [12] has motivated substantial previous research that focused on understanding how to improve the quality of OSS to increase the possibility of success. For instance, previous studies have examined the effects of the technical aspects of the OSS (e.g. [19,32]), the role of learning tools (e.g. [25]), the empowerment (e.g. [27]) and work effectiveness of OSS developers (e.g. [43,48]) on OSS development and quality. Nevertheless, even if an OSS project is completed by generating a stable working version, it may not be able to acquire a critical mass of users, resulting in many sizeable OSS projects being sterile [12].

While it is clear from the previous literature that a key criterion for users to adopt an OSS is the quality of the software e.g. [46], this study proposes that an effective online user support community is also necessary to induce OSS use and popularity, where popularity is typically measured in terms of the number of active users or downloads [5]. As per social influence principles, once individuals download the OSS and obtain useful support, they can then positively impact others to download and use the software as well [13]. However, when people encounter difficulty in utilizing the software and are unable to obtain the necessary support, they may abandon it and influence others in a negative way. The unfavorable reviews of the online user support could propagate to others, thus creating a vicious cycle. Potential users can also be influenced by the user support they observe in the community.

To investigate the importance of the online user support community in increasing OSS use, we examine how user support takes place in OSS. User support involves the provision of assistance to those facing difficulty with the software either due to its defects or the extent of their experience [14, 31]. For commercial software, user queries are usually handled by dedicated user support staff members. In contrast, for most OSS, volunteers who can either be the *developers* or *active users*[3] provide answers to queries posted in the community discussion forum

* Corresponding author. Tel.: +41 44 6323750.
E-mail addresses: jsutanto@ethz.ch (J. Sutanto), atreyi@comp.nus.edu.sg (A. Kankanhalli), btan@comp.nus.edu.sg (B.C.Y. Tan).
[1] Tel.: +65 65164865.
[2] Tel.: +65 65166868.

[3] Active users are knowledgeable non-developer community members who post knowledge contributing messages, e.g. responses to queries. Passive users are community members who either post only knowledge acquiring messages, e.g. queries, or never post messages.

http://dx.doi.org/10.1016/j.dss.2014.05.014
0167-9236/

[44]. However, only a small percentage of projects are found to have effective user support communities [57].

In this study, we posit that a reason for OSS to be unpopular is because of the user support network's inability to deliver effective support. Building on the network embeddedness theory [16,51], we expect user support and consequently OSS use to depend on the structural embeddedness as well as junctional embeddedness of members in the OSS support community. In the context of our study, we consider that structural embeddedness, which captures the extent to which an entity is entrenched in a network of relationships [49], represents the information load of members in the user support community. On the other hand, junctional embeddedness, which refers to the extent to which an entity connects other entities [17], is considered to represent members' capacity to mobilize other volunteers. Members' information load and mobilization capacity should determine how well user support can be delivered and thereby the OSS popularity. As in previous studies, structural and junctional embeddedness are measured through the centralization properties of the network [17] — in this case the OSS user support network.

Through a sample of 176 OSS projects from Sourceforge.net (the largest OSS development website), this study examines the effects of structural and junctional embeddedness of the user support network on the OSS popularity. As a theoretical contribution, this study adds to extant OSS literature that mainly examined individual e.g., competency [6], or team-level e.g., team trust and common ideology [48], antecedents of OSS use and popularity, by investigating the impacts of the user support network through a network embeddedness perspective. For OSS project owners, the findings intend to provide guidance on structuring their online user support communities in order to increase the OSS popularity.

2. Conceptual background

In this section, we first review the previous literature on OSS development and user support to highlight the research gap that our study seeks to address. We then describe the network embeddedness theory as the foundation for identifying the antecedents of OSS popularity in terms of properties i.e., structural and junctional embeddedness, of the user support network.

2.1. Prior literature on OSS development and user support

With the growth in the open source movement, there has been a corresponding interest in open source research. As a result, a fairly large body of knowledge has accumulated in this area — see Crowston et al. [7] for a review of empirical studies and Aksulu and Wade [3] for an overall review of OSS literature. The latter study categorized previous OSS literature into seven patterns or themes i.e., conceptual, performance metrics, legal and regulatory, OSS production, OSS applications, OSS diffusion, and beyond software (e.g., innovation, education). Our research falls within the OSS diffusion theme and the OSS implementation communities and networks sub-theme as per their categorization. With regard to the OSS diffusion theme of research, Aksulu and Wade [3] pointed out that the majority of the prior studies have focused on software development. By contrast, there is a paucity of research on what happens after an OSS has been produced. Under the sub-category of OSS implementation communities and networks, the importance of user support community groups has been noted [3] but there is a lack of investigation of the provision of such user support. Indeed, our own review suggests that phenomena related to online user support of OSS have received limited research attention, with exceptions being the work of Singh et al. [44] who analyzed the types of questions asked and answers provided in eight communities and the study by Lakhani and von Hippel [31] on the motivations of asking questions and providing answers in the Apache help forum.

Particularly, while it is apparent that OSS adoption is influenced by the quality of the software [46,47], we expect that user support should also influence OSS popularity. In the IS literature, the provision of user support has been considered salient to ensuring overall user satisfaction of a new IS application [8,55]. However, the nature of user support in OSS varies from previously studied commercial IS where dedicated user support staff members are usually deployed to answer queries.

When seeking assistance, OSS users (potential or actual) typically browse through the user support discussion forum. If they don't find the relevant support information, they can broadcast their queries for response by community members. Alternatively, they can 'publicly' ask a particular developer or active user to answer their queries. The 'targeted' developers or active users may choose to address the query themselves or refer the query to others who are knowledgeable on the topic or ignore the query. With the delivery of online user support being a community effort, the properties of the community that determine how user queries are addressed should be important for OSS popularity. To examine the way in which such community help can be facilitated, this study builds on the network embeddedness theory.

2.2. Network embeddedness theory

The network embeddedness theory suggests that entities do not act atomically but rather are embedded in a network of relationships that are important determinants of their behavior and capabilities [16,49]. Network embeddedness has been used to explain phenomenon such as organizational performance (e.g. [40,51]), alliance formation (e.g. [2]), knowledge exchange and contribution (e.g. [39,53]), and individual innovation (e.g. [50]).

In examining OSS networks, Grewal et al. [17] defined the concept of embeddedness as the architecture of network ties and rationalized various forms of network embeddedness in previous studies. Accordingly, the study derived three sub constructs of network embeddedness, i.e., structural, junctional, and positional embeddedness. Structural embeddedness denotes the extent to which an entity is entrenched in a network of relationships [49], while junctional embeddedness refers to the extent to which the entity connects other entities [17]. In this study, we did not make use of the concept of positional embeddedness, which is often not differentiated from structural embeddedness [18].

Structural embeddedness has been related to a member's behavior, such as knowledge contribution [53], in online communities. The rationale is that individuals who are centrally embedded in a collective are likely to have developed a cooperation habit of contributing knowledge and are more likely than others to understand and comply with group norms and expectations [53]. However, since contributions in communities such as OSS user support forums are voluntary, when the number of queries increases, it is important to understand whether the structurally embedded individuals would still be willing to contribute their time and effort in responding to all the questions.

Junctional embeddedness has been linked to a member's knowledge reach in organizational communities [50]. This is because individuals who are in-between many other individuals in a network are likely to gain knowledge from diverse sources. Besides being beneficial for personal innovation, knowledge reach may also be advantageous for cooperation and coordinated actions [50]. Prior literature suggests that these benefits are conditional on the nature of the bridging ties i.e., whether there exists a common language across the ties [39,50]. Since each OSS has their respective user support forum, common language within the forum is self-maintained and may not be an issue in the context of our study. A relevant question to examine is whether it is useful to have many junctionally embedded individuals who are able to mobilize other members to respond to queries in the OSS support forum. We present our research hypotheses in the next section.

3. Hypothesis development

Consider a well-known active user or developer in an OSS community whom many people publicly ask for help. This person represents an entity with high structural embeddedness i.e., is connected to a large

number of other entities. The degree of structural embeddedness is likely to determine the information load of the member. As a volunteer, there is no obligation for the targeted member to provide answers to user support requests. If flooded with user queries, he or she may choose to ignore or refer the questions to others. Conversely, when structural embeddedness is low, with few connections to other members and few requests for help coming in, the member may not feel motivated or responsible to respond to the queries. This is because humans have limited motivation to process either too much or too little information [10,28]. Specifically, the level of human information processing is an interactive consequence of dispositional and conditional factors [41]. Too high or too low information load (a conditional factor) would decrease individuals' drive or motivation (a dispositional factor) to process information [15].

Accordingly, previous literature has noted that human information processing systems appear to gradually warm up as the input increases, until they are handling as much as they can, then they fall off [11]. Thus, in response to increases in support requests, a member may initially increase their information processing. However, if the support requests keep increasing and finally exceed the capacity of the individual, the trend will cease. Instead, the member will be likely to decrease their information processing. One of the most common ways of reducing information processing is by means of selective attention [15, 41]. Attention can be thought of as the portal through which information reaches the brain [28]. Selective attention or attention-gating refers to the differential processing of simultaneously received information or messages such that only some part of the information is let through [26]. It is an essential means of dealing with information overload where some incoming information is selected for further analysis while other information is ignored or attenuated and lost [10]. In the context of our study, members of an OSS user support community with high structural embeddedness may initially provide answers to the user support requests publicly directed to them. However, if the increasing number of requests exceeds their capacity, they may begin to selectively respond to some user queries and not all.

Extending this concept to the network level, when members in the OSS support network direct all their queries to a few individuals, there will be a large variation in the number of incoming queries with a few members having high structural embeddedness while the majority of the members have low structural embeddedness. In such a situation where there are only a few individuals with high structural embeddedness, they may easily experience information overload. If they attempt to process numerous messages, they may be cognitively overloaded and their information processing capacity can become compromised [11]. Thus, they may employ selective attention as a means of reducing this adverse state [15] and neglect to respond to some user queries. When current users experience and/or observe such ineffective user support with unanswered queries in the forum, they may spread negative word of mouth that influences others not to download and use the OSS. Also, potential users who observe ineffective user support may be de-motivated to use the OSS.

On the other hand, when members often broadcast support requests to the whole forum, there are marginal differences in the number of incoming user support requests among the members in the network i.e., low variation in structural embeddedness. Here, too, there may be a few individuals (e.g., core developers) who effectively answer the broadcast queries, as a result of which users feel it unnecessary to target their queries. Hence broadcast questions may become the norm in the forum. However over time, these few individuals who answer questions are likely to experience information overload and desist from answering. With the reduced motivation for answering the broadcast queries, many queries will be left unanswered which leads to ineffective user support. As a result of experiencing and/or observing such ineffective user support, current users may abandon the software and negatively influence others about the OSS. Potential users may also decide against downloading and using the OSS when they realize that its user support is ineffective.

The optimal situation is therefore when the differences in the structural embeddedness among all members are moderate. In this case, user support queries are sometimes directed to a few well-known experts in the network and sometimes broadcast to the whole network. Since the experts may not be overloaded by user support queries in this situation, community members should receive answers to their queries which, in turn, is likely to have a favorable effect on OSS use and popularity. Specifically, when current users experience and/or observe effective user support in the forum, they may influence others in a positive way who may then be motivated to download and use the OSS. Also, when potential users observe effective user support, they may be motivated to use the OSS.

The above discussion for high, low, and medium conditions suggests that the variation in the structural embeddedness of the members in an OSS user support network has an inverted U relationship with the OSS popularity. Thus, we hypothesize:

H1. Variation in the structural embeddedness of the members in an OSS user support network has a negative quadratic relationship with the OSS popularity.

The first hypothesis emphasizes the importance of managing the information distribution within an OSS user support community so that queries received by the relatively small number of expert individuals are neither too few nor too many to keep them from actively contributing towards user support and at the same time users' queries do not go unanswered. However, how could the information distribution in an online discussion forum be managed in this fashion? We argue that this could happen when members have the capacity to mobilize one another to address a user's query (see example in Appendix A). Particularly, members who “stand in-between” other pairs of individuals are salient for facilitating information exchange between them. In other words, their junctional embeddedness is high in terms of connecting other people.

In general, a community's capacity to mobilize its members to solve collective problems and maintain or improve its well-being depends on leveraging the human, organizational, and social capital resources existing within the community [35]. In the context of OSS user support communities, the expertise and ties of the volunteers, i.e. developers and active users, can be leveraged to maintain and enhance the delivery of user support, which is the chief purpose of such a community. An active user/developer with high junctional embeddedness may be able to easily mobilize other volunteers to respond to some of the user queries that are publicly directed to him or her. However, another active user with low junctional embeddedness may find it difficult to mobilize others to address user queries.

Previous studies have conceptually related network configuration to the efficiency and effectiveness of information transfer and coordination in virtual networks (e.g. [1, 36]). Similarly, extending the concept of junctional embeddedness to the network level, we believe that the variation in this property across the network will influence the effectiveness of coordinating responses to user queries. When there is low variation in the members' junctional embeddedness, there could be a number of members attempting to mobilize others to respond to user queries, which in turn can cause a disorganized response. People may get multiple referrals for the same information or some queries may not be covered resulting in ineffective support. This may upset current users who are waiting for answers to their problems, and generate negative signals for current or potential users who are browsing through the support forum. Consequently, current users may abandon the software and negatively influence others from using it, while potential users may be de-motivated to adopt the software.

On the other hand, when there is a large variation in the junctional embeddedness of the individuals in the network, this indicates a

centralized control of information flow. Network theory and diffusion of innovation research have shown that such centralization allows for the mobilization of network members [52]. The few members with high junctional embeddedness can act as controllers to make sure that users' queries are being addressed. This avoids the chaotic situation of having too many or too few members suggesting others to address user queries. Thus, users who post enquiries as well as those who observe the support forum will perceive that the community has the capacity for user support and thereby could be motivated to use the software. They can also influence others, who may in turn adopt the OSS. Accordingly we hypothesize:

H2. Variation in the junctional embeddedness of the members in an OSS user support network has a positive relationship with the OSS popularity.

4. Research methodology

In this section, we describe the data collection and sampling of OSS projects for our study, the operationalization of the independent variables (structural and junctional embeddedness), and the measures for the dependent and control variables as well as the data coding for the independent variables.

4.1. Data collection and sampling

The unit of analysis in our study is an OSS project with its user support forum. We collected the sample of OSS projects to test our hypotheses as follows. From among the numerous OSS hosts, SourceForge was chosen as the platform to sample the OSS projects for our data for several reasons. First and foremost, SourceForge is the world's premier OSS development website. It is the largest repository of open source programming code and applications available on the Internet, and hosts more OSS projects than any other site or network worldwide. Second, SourceForge has been studied by a number of OSS researchers (e.g. [5,17,42,48]), thus allowing for accumulation of research results.

While being aware of the above advantages, we also realize certain limitations of SourceForge data. For instance, 1) some projects are now defunct, 2) data from earlier systems has been included, and 3) some projects are also hosted in other sources besides SourceForge [23]. To avoid these potential problems, we carefully screened the sampled projects using the following steps. First, considering that the differences among OSS product categories may affect the results, we selected our OSS projects from two popular product categories, i.e., multimedia and communication. This step ensured that the samples were comparable as well as controlled for differences across projects in different product categories [48]. Second, since one of the measures of the dependent variable is through the number of file downloads, the samples were further refined by deleting those OSS projects which did not release a single file. Third, we filtered out those projects that had other distribution channel(s) outside SourceForge (e.g., offline distribution via CDs, or online distribution in other websites) to minimize noise in the measurement of the dependent variable. Fourth, as in the case of Stewart and Gosain's study [48], because we are concerned with community processes and dynamics, we filtered out those projects comprising fewer than four members. Fifth, since our study concerns the social network of the OSS user support/help forums, we removed those projects that did not have help forums, or those that had help forums outside of SourceForge. After filtering the projects, we ended up with 711 multimedia projects and 744 communication projects to sample from.

Next, we randomly sampled 100 projects from each category which provides an adequate sample for testing our hypotheses and reduces the coding burden as compared to examining all the projects. The user support forums of each of the sampled projects were downloaded with Telepro. The user support communication logs of the sampled OSS were collected from their inception date. When constructing the network graphs of the sampled projects, we encountered anonymous replies in the help forums of a few projects. To minimize noise in the construction of the networks, we first removed those projects that had more than 10% of anonymous replies (16 multimedia and 8 communication projects) in their help forums and then deleted the anonymous replies in the remaining projects. These processes left us with 84 multimedia projects and 92 communication projects. The characteristics of the final sample are shown in Table 1.

4.2. Operationalization of the independent variables

In previous studies [17,49], network embeddedness has been captured through the notion of centrality that indicates the "importance" or "visibility" of network members (e.g. [21]). Specifically, Grewal et al. [17] used degree centrality to measure structural embeddedness and betweenness centrality to assess junctional embeddedness of developers in OSS communities. In our study we extend these concepts to the network level (i.e., centralization) since we aim to examine how different user support networks for OSS projects enhance their popularity. Thus, the *variations in the structural and junctional embeddedness* of the members of the OSS user support network are measured in terms of the *in-degree and betweenness centralization* respectively.

In an OSS user support forum, a member can browse the forum hoping to find relevant answers, or make an enquiry for help. When making an enquiry, the member can either broadcast the message or 'publicly talk' to another member. For example, Tom may broadcast his query to the community as shown below:

> *What server is used to host the traffic? If we use this internal on our LAN, is there any guarantee it will not go outside of our network to the Internet?*
> or Danny may 'publicly talk' to Jenny:
> *Jenny, have you managed to get this working? I am having the same problem.*

The above communications can be represented in the social network with an arrow pointing from Danny to Jenny; but no arrow pointing from Tom to Danny or Jenny. In this manner, all the messages posted in the user support forum can be represented and the centralization properties computed as discussed in the next sub-section.

In-degree centralization of an OSS user support network is an indicator of the degree to which an entire network is centered around a few people. *Betweenness centralization* of a network measures to what extent there is a centralized control of information flow in the network [37]. The equations for in-degree and betweenness centralizations of a network are shown in Appendix B.

4.3. Remaining measures and data coding

All model variables were assessed based on the OSS project records sampled from SourceForge as described in Section 4.1. To compute the independent variables' measures, i.e., *in-degree* and *betweenness centralization*, we utilized UCINET 6 to analyze the network matrices of each project. Data coding to create the network matrices was done by reading the threads of each OSS user support forum, one at a time. In the beginning, one author and a coder coded the same five projects separately. Although the inter-rater reliability was high (slightly less than 80%), the disagreements in the coding results were discussed until both parties reached consensus. Afterwards, the coder proceeded to code the rest of the projects. Subsequently, the first author randomly checked a number of projects from the sample and no discrepancies were found in the coding.

We now provide a simple example to show how the centralization measures were computed from the forum message logs. It is important

Table 1
Characteristics of the sampled OSS projects.

Project characteristics		Number of projects (total = 176)	Percentage of targeted queries
Project age (in years)	<1 year	12 (6.8%)	Mean: 3.42%, median: 0%
	1–<3 years	27 (15.3%)	Mean: 4.15%, median: 2.65%
	3–<6 years	90 (51.1%)	Mean: 2.43%, median: 1.05%
	6–9 years	47 (26.7%)	Mean: 4.76%, median: 1.54%
Project development stage	Planning	6 (3.4%)	Mean: 0.64%, median: 0%
	Pre-alpha	8 (4.5%)	Mean: 7.75%, median: 1.23%
	Alpha	27 (15.3%)	Mean: 3.34%, median: 1.34%
	Beta	53 (30.1%)	Mean: 3.13%, median: 1.61%
	Production/stable	73 (41.6%)	Mean: 3.15%, median: 1.53%
	Mature	9 (5.1%)	Mean: 3.65%, median: 1.03%
Number of developers	<11	156 (88.6%)	Mean: 3.60%, median: 1.47%
	11–20	9 (5.1%)	Mean: 2.98%, median: 1.60%
	21–30	5 (2.8%)	Mean: 2.15%, median: 0.55%
	>30	6 (3.4%)	Mean: 0.48%, median: 0.28%
Community size	<50	138 (78.4%)	Mean: 3.98%, median: 1.54%
	51–100	17 (9.7%)	Mean: 1.68%, median: 1.34%
	101–150	8 (4.5%)	Mean: 2.76%, median: 1.98%
	>150	13 (7.3%)	Mean: 0.78%, median: 0.44%
Intended audience	End-users	142 (80.7%)	Mean: 3.26%, median: 1.36%
	Developers	34 (19.3%)	Mean: 3.52%, median: 1.55%

to stress here that in an OSS user support forum, a member can either broadcast his/her message or 'publicly talk' to another member, as shown in the following logs:

Chris:
Is it possible to force use an "ini" file instead of the registry?
Michael:
This is currently not possible without source code changes and I do not see a reason why settings should be saved in "ini" by default…
Remo:
Dear Michael,Isn't it much better not to save settings in the registry? … I used Friendly Chat before and they saved all settings in an "ini" file. ….
Michael:
Well, Remo, in fact it doesn't matter where the data is saved.

The above forum log shows a situation where Chris broadcast his query, Michael responded to Chris' query, Remo asked a question to Michael, and Michael responded to Remo's question. Based on the formula shown in Appendix B, the in-degree centralization of the user support network in this example is equal to 1.0. Chris and Remo have no potential to bridge any pair while Michael can potentially bridge Chris and Remo. Based on the formula shown in Appendix B, the betweenness centralization of the network in this example is equal to 0.5.

Further, one measure for the dependent variable i.e., the *number of downloads* was collected from the statistics page of each OSS project, while the other i.e., *the number of active users* [27] was computed as those users who are active in contributing to the forum.

Apart from the independent and dependent variables, we included six control variables in our model tests, i.e., *project category*, *project age*, *development stage*, *number of developers*, *number of bugs solved*, and *intended audience*. We controlled for *project category* (1 = communication, 2 = multimedia) and *development stage* (1 = planning, 2 = pre-alpha, 3 = alpha, 4 = beta, 5 = production/stable, 6 = mature) to make sure that the hypothesis testing results were not category or stage specific. Since OSS projects that have been around longer may tend to be downloaded more times, we also controlled for the project age. *Project age* refers to the duration of the project in days starting from its registration date till the date of our data collection. Further, because an OSS with more developers may be downloaded more often [29], we controlled for the *number of developers* in our hypothesis testing. We did not control for the OSS user support *community size* because the calculation of in-degree and betweenness centralization takes into consideration the size of the network. We controlled for the *number of bugs solved* because, as mentioned before, previous studies have shown that OSS adoption is influenced by the quality of the software [46,47]. Last, we controlled for the *intended audience* (end-users or developers) since this may also impact the OSS popularity.

5. Data analysis and findings

The hypotheses were tested using negative binomial regression analysis [22] since the dependent variable is measured using counts of the number of downloads or the number of active users. Tables 2 and 3 show the descriptive statistics of the variables and their correlation matrix respectively.

The correlation matrix shows that no pair of independent or control variables were correlated at 0.60 or above (the maximum correlation was 0.27), which indicates that multicollinearity problem is unlikely. Another test of multicollinearity was performed by computing the variance inflation factor (VIF). Since the highest VIF is 1.3, no multicollinearity problem was indicated [9].

We also tested for endogeneity since the OSS support network includes people who have already downloaded the software. Two-stage least square (2SLS) regression is employed when there is a potential recursivity, i.e., the possibility that the error term of the dependent variable is correlated with the independent variables. To perform this test, we first identified the instrumental variables. There are two main criteria for good instrumental variables: 1) they are uncorrelated with the error term of the dependent variable, and 2) they are fully correlated with the independent variables [34]. It is also important that the instrumental variables should be of similar statistical nature as the independent variables. The number of instruments must be at least as large as the number of potentially problematic independent variables i.e., 2 in our study. We considered other social network properties (besides in-degree and betweenness centralization) as potential instrumental

Table 2
Descriptive statistics of model variables.

Variables	N	Min.	Max.	Mean	Std. dev.
Number of downloads	176	12.06	786388.00	40217.36	102450.54
Number of active users	176	4.00	1099.00	48.32	118.51
InDegree-centralization (%)	176	0.00	22.22	5.23	5.87
Betweenness-centralization (%)	176	0.00	57.76	5.05	10.18
Development stage	176	1.00	6.00	4.17	1.11
Project age (in months)	176	3.00	88.00	54.40	23.00
Number of developers	176	2.00	74.00	8.95	8.16
Number of bugs solved	176	0.00	947.00	32.26	98.81

Table 3
Correlations of independent and control variables.

	1	2	3	4	5	6	7
InDegree-centralization (1)	1.00	0.22	0.13	0.00	−0.21	-0.09	0.01
Between.-centralization (2)		1.00	0.11	0.09	0.12	0.27	0.05
Development stage (3)			1.00	0.20	−0.04	0.10	0.08
Project age (4)				1.00	0.24	0.10	0.16
Number of developers (5)					1.00	0.26	0.07
Number of bugs solved (6)						1.00	0.04
Intended audience (7)							1.00

variables. Among the candidates are out-degree centralization and core-periphery.

Out-degree centralization is an indicator of the degree to which an entire network is focused around a few people that frequently respond to other people's messages. *Core-periphery* of a network measures the extent to which the network revolves around a set of central nodes (the core) who are well-connected with each other, and also with the periphery. The equations for out-degree centralization and core-periphery of a network are shown in Appendix B. These two social network properties do not directly reflect the information load and mobilization capacity of the OSS user support network, but they are related and of similar nature to the in-degree and betweenness centralizations which make them good candidates for our instrumental variables. Out-degree centralization and core-periphery are significantly related to in-degree centralization (both are sig. at 0.000) as well as to betweenness centralization (sig. at 0.000 and 0.012 respectively). This fulfills the second criterion of instrumental variables. To check whether they fulfill the first criterion, we regressed both of them on the error terms of H1and then H2. We found that they are statistically independent of the error terms, which means that any transformations of the instrumental variables will also be uncorrelated with the error terms. Finally, we performed 2SLS regression consisting of the dependent variables, independent variables, and instrumental variables for H1 and H2. We obtained the residuals for each suspected endogenous variables, and then added the residuals to the regression analysis [56]. Since the coefficients of these residuals are not significant, we can conclude that there is no endogeneity problem [56].[4] We then proceeded with the hypotheses testing using negative binomial regression analysis with the R software.

The model met the three assumptions of negative binomial regression analysis [22]: 1) expectation of residuals equals zero, 2) true autocorrelation of residuals is not greater than zero, and 3) variance of residuals is constant. The control variables were entered in step 1, and the independent variables were added in step 2. The changes in Chi-square values between the two steps were noted. The results are shown in Table 4. The AIC values show that Model 2 (control + independent variables) is preferred since it is the one with the lowest AIC. Similarly, the theta parameter that reflects the inverse of dispersion is largest for Model 2. Last, 2 × log-likelihood is used to compute Chi-square change, where a higher value indicates a better model. Thus, all three GFI are better for Model 2.

The results show that the in-degree centralization of OSS user support forums has a negative quadratic effect (inverted U relationship) on the number of downloads and the number of active users.[5] This indicates that the variation in structural embeddedness of the user support network has a negative quadratic influence on the OSS popularity (*H1 is supported*). Further, the betweenness centralization has a positive effect on the number of downloads and active users. In other words, the variation in the junctional embeddedness of the user support network has a positive impact on the OSS popularity (*H2 is supported*).

6. Discussion and implications

6.1. Discussion of results

Based on the data analysis, our study found that an OSS user support forum with moderate variation in structural embeddedness (in-degree centralization) and high variation in junctional embeddedness (betweenness centralization) is optimal to delivering user support, which is associated with higher popularity of the OSS. When a large number of users 'publicly' refer to a few people for support (high variation in structural embeddedness), these members may suffer from information overload which consequently may lead to selective attention [15]. They may ignore some enquiries leading to ineffective user support for the OSS. When users typically broadcast support requests (low variation in structural embeddedness), again, a few individuals (e.g., core developers) may effectively answer the broadcast queries, as a result of which users feel it unnecessary to target their queries. However over time, these few respondents are likely to experience information overload and desist from answering. As their motivation for answering the broadcast queries decreases, many queries will be left unanswered which leads to ineffective user support. Thus a moderate variation in structural embeddedness of the user support network is found to be effective.

Further, to moderate the situation, there is a need for control of the information distribution through a few members who have the ability to mobilize other members whom they are connected to. However, when there are many people who could act as gatekeepers (low variation in junctional embeddedness), user queries may not be addressed effectively because there could be several 'referrals' which in turn might confuse members. In such network structures there are usually several disconnected sub-networks with their own gatekeepers which can result in multiple referrals and inefficiencies in knowledge dissemination. When an OSS community is unable to promptly address support queries, it will eventually affect software popularity. Thus, a high degree of variation in junctional embeddedness of the user support network is found to be optimal.

With regard to the control variables, we found that development stage and number of bugs solved affect number of downloads, while all controls except intended audience affect number of active users. A particularly interesting finding is that coefficients for the number of downloads increase until the OSS projects reach the production stage and then decline when they are at the mature stage (see Table 4). This could be because as the software is stabilizing, there may be many releases, and thus the number of downloads increases. However when it finally stabilizes, there may not be so many releases, and thus the number of downloads decreases. Nevertheless, our hypotheses are still supported when controlling for the maturity of the software. In fact, despite the effects of the development stage, project age (which also reflects the maturity of the project), and the number of bugs solved (which reflects the quality of the software), the hypothesized underlying dynamics in the OSS user support network still hold.

6.2. Limitations and future research

The findings of this study should be interpreted in view of its limitations. First, this study is based on a cross-sectional snapshot of the OSS user support forums. Consequently, it may not be able to detect the effects of changing structural and junctional embeddedness towards the varying popularity (number of downloads or active users) over time.

[4] We accounted for endogeneity both conceptually and empirically. First, those who have downloaded the software may influence others about it based in part on the effectiveness of user support that they have observed or experienced in the forum. Additionally, we also observed potential users (people who had not yet downloaded the software) posting messages in the user support forum. These members are thus part of the nodes of the constructed OSS user support network. Further, there are typically many lurkers who have not yet downloaded the OSS who observe the postings in the forum and form opinions about the software. Second, we tested for endogeneity using the two-stage least square regression approach suggested in Wooldridge [56].

[5] It is debatable whether the linear term of a variable should be included when testing for its quadratic effect (http://www.statisticalhorizons.com/multicollinearity). For this reason, we also did a robustness check by adding the linear term of in-degree centralization in the model. The results are the same, i.e., the quadratic effect of in-degree centralization is negatively related to both dependent variables at $p \leq 0.001$.

Table 4
Hypothesis testing results.

	Model 1 (controls)		Model 2 (controls + independent variables)	
	# Downloads	# Active users	# Downloads	# Active users
Controls				
Project category				
Multimedia	0.094 (0.222)	−0.258 (0.151)	0.021 (0218)	−0.292 (0.136)
Development stage				
Pre-alpha	2.244 (1.044)*	0.800 (0.537)	2.2217 (1.034)*	0.751 (0486)
Alpha	3.260 (0.943)***	0.676 (0.459)	3.379 (0.935)***	0.776 (0.417)
Beta	4.147 (0.916)***	1.119 (0.433)**	4.156 (0.911)***	1.052 (0.396)**
Production	4.391 (0.915)***	1.376 (0.430)**	4.381 (0.910)***	1.315 (0.393)***
Mature	3.326 (1.024)**	0.460 (0.531)**	3.484 (1.017)***	0.630 (0.483)
Project age	−0.007 (0.005)	0.014 (0.003)***	−0.005 (0.005)	0.015 (0.003)***
Number of developers	0.007 (0.014)	0.042 (0.009)***	0.002 (0.014)	0.037 (0.008)***
Number of bugs solved	0.005 (0.001)***	0.005 (0.001)***	0.004 (0.001)***	0.003 (0.001)***
Intended audience				
Developers	−0.361 (0.283)	−0.120 (0.192)	−0.456 (0.278)	−0.181 (0.174)
Independent variables				
InDegree-centralization^2			−0.003 (0.001)**	−0.003 (0.001)***
Betweenness-centraliz.			0.028 (0.001)**	0.040 (0.007)***
Goodness of fit				
AIC value	1345	1573	1339	1528
Theta estimate	0.537 (0.055)	1.162 (0.115)	0.565 (0.058)	1.475 (0.151)
2 × log-likelihood	−1321.007	−1548.599	−1310.926	−1499.967
Chi-square change			10.081***	48.632***

* Significant at $p \leq 0.05$.
** Significant at $p \leq 0.01$.
*** Significant at $p \leq 0.001$.

A longitudinal study would provide an opportunity for future research to test the dynamic and feedback effects. However, this effort could be challenging necessitating that researchers have to periodically capture and code the data over a substantial amount of time.

Second, although the number of downloads is a useful proxy for OSS popularity, not all downloads of a particular version of the software may be associated with unique users. Future research may utilize survey methodology to capture the perceptual popularity of the software. Nevertheless, by utilizing objective measures of OSS network embeddedness and OSS popularity, this study is able to avoid the problems of perceptual measures such as respondent bias and common method error. We also mitigate the problem by using another objective proxy for OSS popularity i.e., number of active users, which provides similar results to the number of downloads.

6.3. Contributions and implications

This study offers several contributions for researchers. While a large number of previous studies have investigated phenomena related to OSS development and a few studies noted the importance of user support groups after the OSS has been produced [3], this study adds to the OSS literature by examining the provision of OSS user support and its effect on OSS popularity. Other than the quality of the software studied previously, our findings show that the availability of user support is important for OSS popularity and adoption. These results contribute to our existing studies of OSS user support, which have so far analyzed the nature and motivations of question asking and responding [31,44].

Further, as the delivery of OSS user support is a community effort, this study examines the relationship between community or network-level properties (structural and junctional embeddedness) and OSS popularity. Since previous studies have mainly explored the effect of team attributes such as trust [48] and individual antecedents such as members' competency [6] on OSS development, this is a novel contribution to the literature on network properties of OSS communities (e.g. [20,43]), particularly for user support networks. When seeking assistance, OSS users can either broadcast their queries or 'publicly' ask a particular community member for the information. In both situations, central members may choose to directly answer the queries or refer them to others or ignore them. Through our study we formally propose and test that variation in structural embeddedness (measured by in-degree centralization which is indicative of information load) and variation in junctional embeddedness (measured by betweenness centralization which is indicative of mobilization capacity) of the OSS user support network affect OSS popularity. In this manner, we also extend the application of the network embeddedness theory to the context of OSS user support.

Our study found an inverted U relationship between variation in structural embeddedness of the user support network and OSS popularity. Lakhani and von Hippel [31] reported that it is of utmost importance for OSS users to gain valuable information from reading about problems that other users are encountering. When the variation in structural embeddedness of an OSS user support network is low or high, there could be many unanswered queries, which may dissatisfy both current and potential OSS users. Accordingly, current users may abandon the software and negatively influence others from using it, while potential users may be de-motivated from adopting the software. This situation could be moderated by having information control, which is reflected by the variation in junctional embeddedness of the user support network. Our study found a positive linear relationship for the variation in junctional embeddedness of the user support network and OSS popularity. Under the conditions of high variation in junctional embeddedness and moderate variation in structural embeddedness, current and potential users will be satisfied with the OSS user support and a virtuous cycle could ensue to enhance the OSS popularity.

Last, our study adds to prior OSS literature [30] that shows relationships between embeddedness and project outcomes. Examining developers' discussion threads of new features and applications, Kuk [30] also found an inverted U relationship between structural embeddedness and knowledge sharing. Although the contexts differ, both studies examined communication structural embeddedness in OSS projects and highlighted the importance of participation inequality in OSS development as well as in OSS user support network. Together, these findings challenge the "bazaar" view of OSS projects described previously [38].

Besides contributing to OSS research, the results have practical implications for OSS project owners. To successfully complete an information transaction on an OSS help forum, three tasks must be performed: (1) a question must be posted, (2) the information sought must be matched to an appropriate and willing provider of information, and (3) an answer must be provided [31]. When most of the user queries are broadcast without intended recipients (*low variation in structural embeddedness*), the potential information providers must identify the questions that they can and are willing to answer. Also, they may be de-motivated by the request not being specifically addressed to them or expect others to answer the query. Consequently, they may not be willing to answer some of the users' queries even if they know the answers. On the other hand, when most of the users habitually and 'publicly' refer to a few individuals for their queries (*high variation in structural embeddedness*), these few individuals may have to spend a substantial amount of time to provide answers to each query. As a result, these individuals may deliberately ignore some queries. Thus, it is important to have members who can mobilize other individuals to whom they are connected to address users' queries. However, when there are multiple gatekeepers (*low variation in junctional embeddedness*), information seekers and providers may receive a number of 'referrals' which in turn may confuse them.

OSS project owners would benefit from monitoring and shaping the structure of the user support forums to prevent information overload while developing the mobilization capacity to deliver user support. Research in the area of personalization could provide possible solutions to overcome the problem of information overload [45]. A personalization tool that learns the preferences of individual members over time by monitoring their online interactions (e.g., pages browsed, queries ignored, queries answered) can then use this information to filter out irrelevant forum pages and unsuitable queries for that expert, thus preventing these members from experiencing information overload. A frequently-asked-questions (FAQ) thread may also partially shift the burden of matching the questions and answers from information providers to information seekers.

Another technological advancement that may facilitate the process of mobilizing community capacity in addressing members' queries is buddysourcing. As compared to the more well-known crowdsourcing concept that depends on people's attention to the requests in the online community [24], buddysourcing technology learns the social networks of each members and their expertise which in turn allows the automatic forwarding of queries that are directed to a particular member to his/her 'expert' friends. The above technologies have the potential to alleviate information overload as well as mobilize the OSS user support community in addressing user queries.

In the sampled user support forums we observed that many threads overlap, since anyone can create a new thread. Following from our observations, we suggest that OSS project owners create sub forums within a user support forum and designate moderators for the forum. If a user creates a thread in an inappropriate sub forum, the moderators should move the thread to a more relevant sub forum. If there are overlapping threads, the moderators should merge the threads. Over time, experts in different sub forums can be easily identified by users (and moderators); thus allowing them to know whom to direct their queries without overloading a particular individual or repeatedly broadcasting their queries and hoping that someone would answer them. These moderators can also act as controllers with the right to create a new sub forum when the current forum becomes too crowded. By doing so, more specialized experts can be identified by users for better match of solutions to queries.

7. Conclusion

With their various advantages, use of OSS is growing rapidly. This study attempts to add to our understanding of why certain OSS are popular by employing a novel, user support perspective. The research built on the network embeddedness theory to investigate the effect of structural and junctional embeddedness of an OSS user support forum on the OSS popularity. Thus, the study explained the effect of community-level factors of OSS user support in contrast to previous literature that mainly focused on individual and team-level antecedents of OSS development.

The findings indicate that both the variations in the structural embeddedness and the junctional embeddedness of the OSS user support forum exhibit significant relationships with OSS popularity. Moderate degree of variation of members' structural embeddedness and high degree of variation of members' junctional embeddedness allow for effective user support by letting experts respond to user queries or forward them to other relevant members if they are unavailable or unqualified to respond. Based on these results, this study provides recommendations for OSS project owners on how to structure their user support forums.

Appendix A. Example of community mobilization

The following log is an evidence of an individual (in this case, SV) mobilizing other individuals to address queries.

Problem with files larger than 500 kbytes
By: GT
Everything works fine for small files but I have a problem when uploading files larger than 500 K. I get the following error message:
Request entity too large! The POST method does not allow the data transmitted, or the data volume exceeds the capacity limit.
If you think this is a server error, please contact the webmaster. Error 413 …
Here's my configuration: …
Any idea why that is happening?
Thanks
RE: Problem with files larger than 500 kbytes
By: SV
I have no idea, but if you send me your php.ini I'll compare it to mine.
RE: Problem with files larger than 500 kbytes
By: HT
I have the same problem. I'm running: …
RE: Problem with files larger than 500 kbytes
By: DG
I also have the same problem. Same specs as HT.
RE: Problem with files larger than 500 kbytes
By: SV
What version (of the software)? Anyone with a lesser version of Apache or php have this problem?
RE: Problem with files larger than 500 kbytes
By: HT
It's version … in my case.
RE: Problem with files larger than 500 kbytes
By: LM
SV, I had same problem, though I did not get any error message. Moved back to Redhat 7.3 and whatever default Apache server and PHP came with it. No problem with older versions. Upload just about any size file.
RE: Problem with files larger than 500 kbytes
By: PB
Yep, SV. Ever since I upgraded to RH8.0 from 7.3, I have hit the same problem. I am using a plain vanila RH install as follows …
RE: Problem with files larger than 500 kbytes
By: SV
Can one of you do a google search, to see if this is a known problem with Apache or php or RedHat 8.0?
RE: Problem with files larger than 500 kbytes
By: PB
Yep, you are right, Google is great. The answer is: …
RE: Problem with files larger than 500 kbytes
By: JS
I found the same solution as you, but I just have a new problem. When I try to upload at 11.8 Mb pdt file I get:
Warning: Unable to open for reading: No such file or directory in/var/www/html/extranet/dbmodify.php on line 312 …"
RE: Problem with files larger than 500 kbytes
By: SV
What version of (the software)? Did you increase the config/php max file size as well?
RE: Problem with files larger than 500 kbytes

(*continued on next page*)

Appendix A. (*continued*)

By: MT
I tried to upload a very big file, about 16 MB. I did all changes and now the error message is:
Fatal error: Maximum execution time of 30 s exceeded in c:|apache|htdocs|intranet|dbmodify.php on line 12
… Where is the problem? Apache, php or (the software)?
RE: Problem with files larger than 500 kbytes
By: SV
I doubt it's (the software) in this case, Apache is timing out …
Anyone else have any idea? PB?
RE: Problem with files larger than 500 kbytes
By: PB
I just double-checked my config by uploading a 41.1 MB file and it worked ok. My exact config is posted to: …
May I suggest double checking the params back to this known working setup and see what happens.
Otherwise it has to be related to a very slow connection.
RE: Problem with files larger than 500 kbytes
By: MT
I did the changes and it's worked. Thanks
RE: Problem with files larger than 500 kbytes
By: SV
Thanks (PB) for stepping in.

Appendix B. Social network indices

Index	Definition	Formula
In-degree centralization	The variation in the in-degree centralities of vertices divided by the maximum possible in-degree variation in a network of the same size	$InDegreeCentralization(D) = \frac{1}{(n-1)(n-2)}\sum_{i=1}^{n}(d^* - d_i)$ where n = the total number of nodes in the network, d_i = the number of edges converging on a given node, and $d^* = max_i\, d_i$
Betweenness centralization	The variation in the betweenness centralities of vertices divided by the maximum possible variations in betweenness centrality scores in a network of the same size	$BetweennessCentralization(B) = \frac{1}{(n-1)^2(n-2)}\sum_{i=1}^{n}(b^* - b_i)$ where n = the total number of nodes in the network, b_i = the proportion of shortest (geodesic) paths between other nodes that a given node resides on, and $b^* = max_i\, b_i$
Out-degree centralization	The variation in the out-degree centralities of vertices divided by the maximum possible out-degree variation in a network of the same size	$OutDegreeCentralization(D) = \frac{1}{(n-1)(n-2)}\sum_{i=1}^{n}(d^* - d_i)$ where n = the total number of nodes in the network, d_i = the number of edges diverging from a given node, and $d^* = max_i\, d_i$
Core-periphery	Inequality in coreness scores of the nodes in a network is used to assess overall network-level structural inequality or core-periphery	The "coreness" score of each node quantifies how "close" the node is to the dense core of the network. The algorithm developed by Borgatti and Everett [4] is used to assign coreness scores and the Gini coefficient is used to measure inequality in the scores. Gini coefficient is a commonly used measure of inequality that varies from 0 (perfect equality) to 1 (perfect inequality).

References

[1] M.K. Ahuja, K.M. Carley, Network structure in virtual organizations, J. Comput.-Mediat. Commun. 3 (4) (1998) (Retrieved from: jcmc.indiana.edu/vol3/issue4/ahuja.html).
[2] G. Ahuja, F. Polidoro Jr., W. Mitchell, Structural homophily or social asymmetry? The formation of alliances by poorly embedded firms, Strateg. Manag. J. 30 (9) (2009) 941–958.
[3] A. Aksulu, M.R. Wade, A comprehensive review and synthesis of open source research, J. AIS 11 (11) (2010) 576–656.
[4] S.P. Borgatti, M.G. Everett, Models of core/periphery structures, Soc. Networks 21 (1999) 375–395.
[5] K. Crowston, J. Howison, H. Annabi, Information systems success in free and OSS development: theory and measures, Softw. Process. Improv. Pract. 11 (2) (2006) 123–148.
[6] K. Crowston, B. Scozzi, Open source software projects as virtual organizations: competency rallying for software development, IEE Proc. Softw. 149 (1) (2002) 3–17.
[7] K. Crowston, K. Wei, J. Howison, A. Wiggins, Free/libre open source software development: what we know and what we do not know, ACM Comput. Surv. 44 (2012) 2.
[8] W.H. DeLone, E.R. McLean, The DeLone and McLean model of information systems success: a ten-year update, J. Manag. Inf. Syst. 19 (4) (2003) 9–30.
[9] A. Diamantopoulos, J.A. Siguaw, Formative versus reflective indicators in organizational measure development: a comparison and empirical illustration, Br. J. Manag. 17 (2006) 263–282.
[10] B.A. Dosher, G. Sperling, A century of human information-processing theory, in: J. Hochberg (Ed.), Perception and Cognition at Century's End, Academic Press, New York, 1998, pp. 199–252.
[11] M.J. Eppler, J. Mengis, Concept of information overload: a review of literature from organization science, accounting, marketing, MIS, and related disciplines, Inf. Soc. 20 (5) (2004) 325–344.
[12] B. Fitzgerald, Open source software adoption: anatomy of success and failure, Int. J. Open Source Softw. Processes 1 (1) (2009) 1–23.
[13] M.D. Gallego, P.L. Luna, S. Bueno, User acceptance model of open source software, Comput. Hum. Behav. 24 (5) (2008) 2199–2216.
[14] X. Ge, Y. Dong, K. Huang, Shared knowledge construction process in an open-source software development community: an investigation of the gallery community, Proceedings of the 7th International Conference on Learning Sciences, 2006, pp. 189–195.
[15] R.G. Geen, Social motivation, Annu. Rev. Psychol. 42 (1991) 377–399.
[16] M. Granovetter, Economic action and social structure: the problem of embeddedness, Am. J. Sociol. 9 (1985) 481–493.
[17] R. Grewal, G.L. Lilien, G. Mallapragada, Location, location, location: how network embeddedness affects project success in open source systems, Manag. Sci. 52 (7) (2006) 1043–1056.
[18] R. Gulati, Alliances and network, Strateg. Manag. J. 19 (4) (1998) 293–317.
[19] S. Haefliger, G. von Krogh, S. Spaeth, Code reuse in OSS, Manag. Sci. 54 (1) (2008) 180–193.
[20] J. Hahn, J.Y. Moon, C. Zhang, Emergence of new project teams from open source software developer networks: impact of prior collaboration ties, Inf. Syst. Res. 19 (3) (2008) 369–391.
[21] V. Hamilton, D.M. Warburton, Human Stress and Cognition, John Wiley, New York, 1979.
[22] J.M. Hilbe, Negative Binomial Regression, Cambridge University Press, U.K., 2011.
[23] J. Howison, K. Crowston, The perils and pitfalls of mining SourceForge, Presented at the Workshop on Mining Software Repositories, 26th International Conference on Software Engineering, Edinburgh, Scotland, 2004, (Retrieved from: citeseer.ist.psu.edu/howison04perils.html).
[24] B.A. Huberman, D.M. Romero, F. Wu, Crowdsourcing, attention, and productivity, J. Inf. Sci. 35 (6) (2009) 758–765.
[25] C.L. Huntley, Organizational learning in open-source software projects: an analysis of debugging data, IEEE Trans. Eng. Manag. 50 (4) (2003) 485–493.
[26] W.A. Johnston, V.A. Dark, Selective attention, Annu. Rev. Psychol. 37 (1986) 43–75.
[27] W. Ke, P. Zhang, Effects of empowerment on performance in open-source software projects, IEEE Trans. Eng. Manag. 58 (2) (2011) 334–346.
[28] T. Klingberg, The Overflowing Brain: Information Overload and the Limits of Working Memory, Oxford University Press, New York, 2009.
[29] S. Krishnamurthy, Cave or community? An empirical examination of 100 mature OS projects, First Monday 7 (6) (2002) (Retrieved from: firstmonday.org/issues/issue7_6/krishnamurthy/index.htm).
[30] G. Kuk, Strategic interaction and knowledge sharing in the KDE Developer mailing list, Manag. Sci. 52 (7) (2006) 1031–1042.
[31] K.R. Lakhani, E. von Hippel, How open source software works: free user-to-user assistance, Res. Policy 32 (6) (2003) 923–943.
[32] J. Li, R. Conradi, C. Bunse, M. Torchiano, O. Slyngstad, M. Morisio, Development with off-the-shelf components: 10 facts, IEEE Softw. 26 (2) (2009) 80–87.
[33] J. Lindman, M. Rossi, A. Paajanen, Matching open source software licenses with corresponding business models, IEEE Softw. 28 (4) (2011) 31–35.
[34] D. McFadden, D. McFadden, Instrumental Variables, 1999. (Retrieved from: elsa.berkeley.edu/~mcfadden/e240b_f01/ch4.pdf).
[35] S.E. Meyer, Building Community Capacity: The Potential of Community Foundations, Minneapolis, Rainbow Research, 1994.
[36] J. Nahapiet, S. Ghoshal, Social capital, intellectual capital, and the organizational advantage, Acad. Manag. Rev. 23 (2) (1998) 242–266.
[37] W.D. Nooy, A. Mrvar, V. Batagelj, Exploratory Social Network Analysis with Pajek, Cambridge University Press, New York, 2005.
[38] E. Raymond, The cathedral and the bazaar, First Monday 3 (1998) 3-2 (Retrieved from: firstmonday.org/ojs/index.php/fm/article/view/578/499issues/issue3_3/raymond/).
[39] R. Reagans, B. McEvily, Network structure and knowledge transfer: the effects of cohesion and range, Adm. Sci. Q. 48 (2) (2003) 240–267.

[40] T. Rowley, D. Behrens, D. Krackhardt, Redundant governance structures: an analysis of structural and relational embeddedness in the steel and semiconductor industries, Strateg. Manag. J. 21 (3) (2000) 369–386.
[41] H.M. Schroder, M.J. Driver, S. Streufert, Human Information Processing, Holt, Rinehart and Winston, New York, 1967.
[42] S.K. Shah, Motivation, governance, and the viability of hybrid forms in open source software development, Manag. Sci. 52 (7) (2006) 1000–1014.
[43] P.V. Singh, Small-world effect: the influence of macro-level properties of developer collaboration networks on open-source project success, ACM Trans. Softw. Eng. Methodol. 20 (2) (2010) 1–27.
[44] V. Singh, M.B. Twidale, D.M. Nichols, Users of open source software — how do they get help? Proceedings of Hawaii International Conference on Systems Science, 2009, pp. 1–10.
[45] B. Smyth, Personalization and collaboration in social search, Presented at the 7th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP), 2009.
[46] S.Y. Sohn, M.S. Mok, A strategic analysis for successful open source software utilization based on a structural equation model, J. Syst. Softw. 81 (6) (2008) 1014–1024.
[47] I. Stamelos, L. Angelis, A. Oikonomou, G.L. Bleris, Code quality analysis in open-source software development, Inf. Syst. J. 12 (1) (2002) 43–60.
[48] K.J. Stewart, S. Gosain, The impact of ideology on effectiveness in open source software development teams, MIS Q. 30 (2) (2006) 291–314.
[49] X. Tang, A. Rai, J. Wareham, Bridging and bonding in exchange networks: a structural embeddedness perspective of B2B digital intermediation, IEEE Trans. Eng. Manag. 58 (1) (2011) 4–20.
[50] M. Tortoriello, D. Krackhardt, Activating cross-boundary knowledge: the role of Simmelian ties in the generation of innovations, Acad. Manag. J. 53 (1) (2010) 167–181.
[51] B. Uzzi, The sources and consequences of embeddedness for the economic performance of organizations: the network effect, Am. Sociol. Rev. 61 (4) (1996) 674–698.
[52] T.W. Valente, Network Models of the Diffusion of Innovations, Hampton Press, Cresskill, NJ, 1995.
[53] M.M. Wasko, S. Faraj, Why should I share? Examining social capital and knowledge contribution in electronic networks of practice, MIS Q. 29 (1) (2005) 35–57.
[54] R.T. Watson, M. Boudreau, P.T. York, M.E. Greiner, D. Wynn, The business of open source, Commun. ACM 51 (4) (2008) 41–46.
[55] B.H. Wixom, P.A. Todd, A theoretical integration of user satisfaction and technology acceptance, Inf. Syst. Res. 16 (1) (2005) 85–102.
[56] J.M. Wooldridge, Introductory Econometrics, 3rd Edition South-Western Cengage Learning, 2005.
[57] R. Zachary, Six options for open-source support, Information Week (1 March 2006).

Juliana Sutanto received the B. Comp (Hons.) and Ph.D. degrees from the National University of Singapore. Currently, she is an Assistant Professor and Chair of the Management Information Systems Group, Department of Management, Technology, and Economics, ETH Zurich, Zurich, Switzerland. Her research interests include virtual teams, virtual communities, and social networking. Her work has been published in the Management Information Systems Quarterly, IEEE Transactions on Engineering Management, Journal of Management Information Systems, and others.

Atreyi Kankanhalli is an associate professor in the Department of Information Systems at the National University of Singapore. Her research interests are knowledge management, virtual teams, and information technology in the service sectors. Her work has appeared in MIS Quarterly, Journal of Management Information Systems, IEEE Transactions on Engineering Management, International Journal of Human Computer Studies, and others. She serves on the editorial boards for several information systems journals, including MIS Quarterly.

Bernard C. Y. Tan is a professor in the Department of Information Systems, National University of Singapore. Currently, he is a Vice Provost of the National University of Singapore. His current research interests are knowledge management, virtual communities, and information privacy. He has served or is serving on the editorial boards of MIS Quarterly, Management Science, Journal of AIS, Journal of Management Information Systems, IEEE Transactions on Engineering Management, Information and Management, and others.