

# Social Network Analysis on Communications for Knowledge Collaboration in OSS Communities

Takeshi Kakimoto    Yasutaka Kamei    Masao Ohira    Ken-ichi Matsumoto  
Graduate School of Information Science  
Nara Institute of Science and Technology  
8916-5 Takayama Ikoma Nara Japan 630-0192  
{takesi-k, yasuta-k, masao, matumoto}@is.naist.jp

## Abstract

*Knowledge collaboration is the key for success of open source software (OSS) communities, because not all members have knowledge and skills necessary for software development. Generally, members in OSS communities communicate for knowledge collaboration using communication tools (e.g. mailing lists, discussion forums, bug tracking systems, and so on) so that geographically distributed members collaborate and coordinate their work. In this paper, we apply social network analysis to the data accumulated in communication tools. We analyzed relationships between the density of social networks and OSS releases by time series analysis of 4 OSS communities in SourceForge.net, in order to investigate the quality of communications for knowledge collaboration. The analysis results showed that communications among community members with a variety of roles are active before/after OSS release in communities where knowledge collaboration is going well.*

## 1. Introduction

Nowadays software developers continuously require a considerable amount of new and diverse knowledge about technologies for software development such as programming languages and components libraries, since such technologies have been evolving from day to day and the past knowledge about them cannot be used soon. In this situation, an individual developer cannot possess every kind of knowledge about latest technologies needed for software development. Knowledge collaboration [11] is not desirable but necessary for modern software development.

Especially, open source software (OSS) development communities rely on knowledge collaboration among community members who have a variety of roles such as com-

munity leaders, developers, bug reporters, passive users and so forth [7, 12], because OSS communities, differently from traditional software development organizations, cannot recruit members who have sufficient skills and knowledge required for building software systems in advance.

In typical OSS communities where community members are geographically distributed, knowledge collaboration takes place through using collaboration tools such as version control systems, bug tracking systems, and mailing lists. Based on the data stored in the collaboration tools, prior studies discussed the model of collaboration processes in distributed environments [10], the efficiency of communication and coordination in distributed software development [4], the benefits of OSS style software development [6], communications metrics for knowing the quality of group work [2] and so forth.

In this paper, we would like to investigate the quality of communications for knowledge collaboration by analyzing the data from communication tools used for distributed software development and the data denoting the success and failure of knowledge collaboration (e.g. number of software releases and number of software downloads). In OSS development, community members rarely meet to discuss but communicate heavily using electronic media (e.g. mailing lists and forums). So, we supposed that we might comprehend the success and failure of knowledge collaboration from the quality of communications among community members through collaborative communication media.

As an approach to inspecting the quality of communications for knowledge collaboration, we use social network analysis (SNA) [8, 9], especially the density of social networks which is a measure to know the quality of social relationships among people (e.g. intimacy or solidarity among people). In this paper, we applied SNA methods to the communication data stored in forums for OSS communities in SourceForge.net (SF.net)<sup>1</sup>.

<sup>1</sup>SourceForge.net, <http://sourceforge.net/>

In what follows, in Section 2 we hypothesize on communications for knowledge collaboration, more specifically, how knowledge collaboration in OSS communities is conducted using electronic communication media. Section 3 describes density of social networks, which is a measure for SNA. In section 4 we analyze 4 OSS communities in SF.net. Section 5 is the results of our analysis. We discuss the results and our hypothesis in Section 6. Section 7 concludes the paper.

## 2. Communications for Knowledge Collaboration in OSS Communities

In this section, we discuss communications for knowledge collaboration in OSS communities. Typical OSS communities where community members are geographically distributed and rarely meet to discuss together, heavily relies on collaboration tools such as version control systems and bug tracking systems and electronic communication media such as mailing lists and forums to precede their knowledge collaboration. Yamauchi et al. [10] had conducted two case studies to investigate how OSS development communities achieve smooth coordination and effective collaboration. One of the findings of the case studies was that collaboration and communication tools (e.g. CVS, TODO lists and Mailing lists) were used in a good balance between centralization and spontaneity [10].

In this paper we would like to focus on the quality of communications for knowledge collaboration through communication media. In OSS development, communications for knowledge collaboration involve a variety of people. For instance, software developers discuss technological problems, bug reporters point out bugs of released software, end-users request developers to add new features and so forth. It is important for knowledge collaboration to involve such a variety of community members because “voice” from bug reporters and end users often makes OSS reliable and innovative, and motivates OSS developers to develop further OSS[3].

Figure 1 shows a simple model on a cycle of knowledge collaboration in OSS development. Before OSS released, OSS developers discuss their products and related problems (development period). After OSS released, users ask questions on usage of the products to other users or developers and also report bugs or requests of a new features to developers (feedback period). Again, developers discuss the reported bugs and requested features, and then modify and refine their products. This would be a simple view of a cycle of OSS development but an important aspect of knowledge collaboration, because an end user would not use the products if s/he can get help from other community members, a bug reporter would not report bugs if developers do not modify reported bugs, and developers would not continue

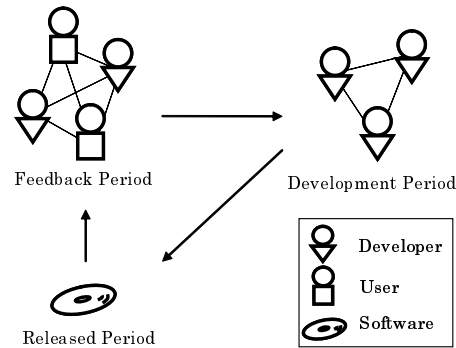


Figure 1. Cycle of Knowledge Collaboration

to create software products if no one use them. Here we can make a hypothesis on communications for knowledge collaboration in OSS development communities as follows.

**Hypothesis:** Communications are actively encouraged before/after OSS released, especially among community members with a variety of roles but not among particular members.

We thought that we might be able to know the success and failure of knowledge collaboration or “health condition” in OSS communities by analyzing the quality of communications among community members before/after OSS released. The next section describes use of the density of social networks which is our approach to investigating the quality of communications in OSS communities.

## 3. Density of Social Networks

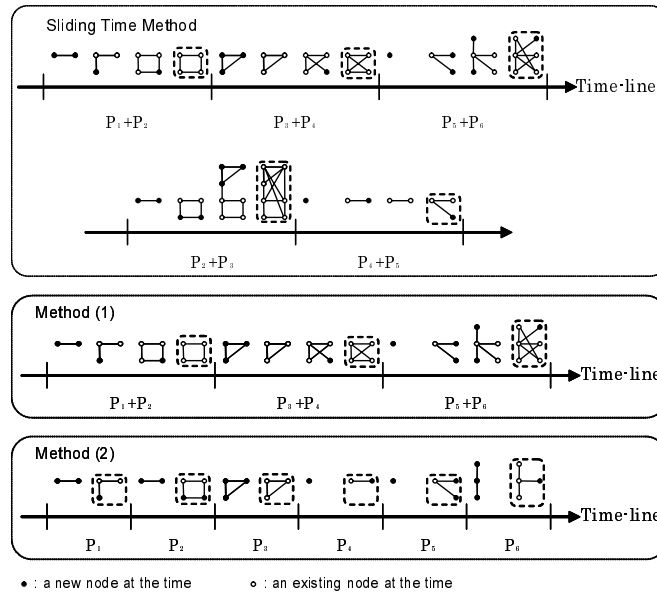
Using the density of social networks in social network analysis (SNA) is a simple way to know the quality of social relationships among people [8, 9]. Social relationships can be graphed as social networks, which consist of persons (nodes) and their relationships (edges).

The density of social networks is defined as the number of lines (edges) in social networks, expressed as a proportion of the maximum possible number of lines [8, 9]. The formula for the density of social networks is

$$ND = \frac{2l}{n(n-1)} \quad (1)$$

where  $l$  is the number of lines (edges) in the networks and  $n$  is the number of nodes in the networks. The values of  $ND$  (network density) can be from 0 to 1.

If social networks show low density, the social relationships tend to be “open” which means *a large, open, diverse, and externally focused relationships* [1]. If social



**Figure 2. Calculation methods for the density**

networks indicate high density, the social relationships often have characteristics of “closed” which means a *small, closed, homogeneous, and internally focused network* [1].

In this paper we apply SNA to the communication data stored in communication tools such as mailing lists and forums (bulletin board systems) to know the quality of communications for knowledge collaboration in OSS development. In this case, social relationships can be defined by posts and replies. Community members (e.g., developers, end-users, bug reporters, and so on) discuss issues related to OSS development. If a member (A) posts a message to a forum for a community ( $C_i$ ) and a member (B) replies the message, then it can be assumed that there is a social relation between A and B in  $C_i$ . Therefore, the density of the social relationships (i.e. social networks) will be high when community members mutually discuss a topic in a forum, but it will be low when no one post a reply message even if there are a number of posted messages in a forum.

Although the activeness of an online community, in general, can be measured by the amount of communications among community members, we do not use the number of posted messages to a forum to know the quality of communications from the above reason. We also do not use the number of replies to know how community members mutually discuss issues because only a handful of members often reply to posted messages in an online forum [5]. We expect that the density of social networks is better to know whether communications for knowledge collaboration are going well or not.

## 4. Analysis on The Quality of Communications for Knowledge Collaborations

### 4.1. Dataset

We collected the data involving public forums and OSSs released in 4 OSS communities for the time interval between December 1, 1999 and December 31, 2005. These communities were selected as target communities for analysis because they indicated characteristic measurements results (e.g. a large number of developers, downloads, or posts). We did not collect the data of mailing lists because the mailing lists were not used for communications among community members but mainly for announcements of OSS releases or archives of CVS logs. The data on public forums includes ID of each posted message, user’s name who posted messages, the date of messages posted, ID of each replied message, and ID of each OSS community. The data on released OSS includes the number of developers in each community, the start date of each community, the number of downloads, the number of average downloads per a day, version numbers of released OSS, the release date of OSS, and ID of each OSS community.

### 4.2. Analysis Procedure

The followings show the procedure of our analysis using social network analysis (SNA) [8, 9].

**Preparation** Before calculating the density of social networks, firstly we need to define social networks in

**Table 1. Characteristics of target communities**

Community	Num. of developers	Density of all periods	Num. of posts	Date of communities started	Num. of downloads	Num. of average downloads per a day
Community A	138	0.022	174	04-Jun-01	28,265	16.92
Community B	1	0.013	165	07-Oct-04	7,734,629	17188.06
Community C	11	0.007	766	05-Dec-99	26,000,000	11878.12
Community D	3	0.500	203	29-Dec-03	156	0.21

the context of our analysis. As described before, our aim of using the density of social networks is to know the quality of communications for knowledge collaboration. We use the communication data made from discussions (messages) in forums.

From messages in forums for a target community<sup>2</sup>, we identify who posted a message to the forums (node A) and who replied to the message (node B). Then we regard the relation between the poster (node A) and the respondent (node B) as an edge, by threading relationships between posts and replies as social networks. Repeating this for all messages in forums of a target community, we can graph the relationships as social networks and calculate the density of the social networks.

#### Calculations of network density by a certain period

Calculating the density of social networks from all the data is inadequate, because the density is calculated from a snapshot of structures of social networks at a certain point while structures of social networks change over time. Therefore, time series analysis is necessary to know changes of the quality of communications among community members, that is, changes of the density of social networks. In order to see temporal changes of the density of social networks, we have to fix a particular time interval.

We calculate the density of social networks from social networks for a period  $P$  in a way that slides a  $\frac{P}{2}$  interval (sliding time method) in this paper. Figure 2 shows calculation methods for the density of social networks. The density of a social network for a certain period is calculated from the structure of the network at the end of the period.

The sliding time method in this paper is sensitive to changes of network structures than method (1) and (2) which not overlap neighboring periods. For example, communications are active in the period of  $P_2 + P_3$ .

However, method (1) can not reflect such the activeness. Method (2) which divides the period in half also can not reflect the activeness because it can only show small changes.

In this paper, the density of social networks is calculated by one and a half month ( $P = 3$  months). The reason why we fix 3 months is we considered that one topic in a forum is finished about 3 months. We need further consideration for this period or a way to fix an appropriate period.

**Time series analysis** We analyze relationships between the density of social networks and OSS releases in order to verify our hypothesis. Changes of the density of social networks in time series are used in the analysis. The number of posters who posted messages (i.e. nodes), links among posters (i.e. edges), and posted messages are also used.

#### 4.3. Target Communities

In this paper, we analyze 4 characteristic communities. Table 1 shows the measurement results of each community. In what follows, we describe an overview of each community, which consists of characteristic measurement results, developing software, and usages of forums.

**Community A** Community A has a number of developers. This community has been developing an operating system for controlling small electronic devices. The posted messages to the forum of community C consist of questions on implementation from developers. This community is currently working on own web site but not on SF.net.

**Community B** Community B has only one developer but provides a tool downloaded by a large number of users. This community provides windows installers for image manipulation software which is originally developed for UNIX. The posted messages are only from users.

<sup>2</sup>A community can have several forums for different purposes of discussions

**Community C** The tool created in community C is downloaded by a large number of users. Community C has been providing a CD ripping tool. The posted messages to the forums of the community consist of posts regarding implementation of software, questions on released software, bug reports, and requests for new features. Both developers and users often post to the forums. Anonymous users who do not have user ID of SF.net also use them.

**Community D** The characteristic measurement results of community D are that the network density is very high and the number of downloads and posters is very small. Community D creates an OpenGL viewer with command line tools. The forums of this community are used only by developers excepting one post by a user.

## 5. Analysis Results

Figure 3 shows time series graphs for 4 target communities. The horizontal axis shows time sequence, the vertical axis at the right side is values of the density of social networks, and the vertical axis at the left side is the number of posters, links among posters and posts for each period. Dashed lines mean the date of OSS release.

**Community A** The following pattern of the changes of the density in community A was repeated. At the initial phase of the community started, values of the density became high. Then, values of the density were decreased as the community progressed. Finally, values of the density became zero. Version 0.6.0 and version 1.0 were released when values of the density became zero. Values of the density before OSS releases were higher than that for OSS release periods excepting version 0.6.1. The number of posts after OSS releases was larger than that for OSS release periods. Posted messages before OSS releases were mainly from developers and posted messages after OSS releases were from users.

**Community B** In community B, when the density was increasing or high, new versions were released in a short interval. On the other hand, when the density was decreasing, new versions were released in a long interval. Values of the density after OSS releases were higher than that for OSS released periods in most cases. When the number of posts was small in a long interval, community B did not release software products. Developers did not post a message. All messages were posted by users.

**Community C** In community C, values of the density before OSS releases were higher than that for released

periods in all cases. Values of the density after OSS releases were also higher than that for released periods excepting version 1.50. The degree of increases of density values after releases is decreasing as the community progressed. The number of posts after OSS releases was larger than that before OSS releases. Posted messages before OSS releases were from developers and that after OSS releases were from developers and users.

**Community D** The number of posters is small against the number of posts in the community D. All messages were posted by developers. In the version 0.1 release, values of the density after the release were higher than that for the released period. No developers posted messages after September 2004.

## 6. Discussion

The analysis results excepting community D showed that values of the density before OSS releases are high in the community that has a number of posts from developers (community A, C). And, values of the density after OSS releases are high in the community that has a number of posts from users (community B, C).

In the community C that meets both the conditions, values of the density before and after OSS releases are higher than values of the density for released periods. In other words, communications among community members are active before and after released periods in community C. On the other hand, community D that is not the case with these conditions seems to be stagnant as the number of downloads and posters was very small and OSS was released after the version 0.1. Therefore, we consider that our hypotheses are true for communities where knowledge collaboration among community members with a variety of roles is going well.

One of the advantages of using the density of social networks is that we can know community members mutually discuss issues. If the number of posts is large but the density is low, it would mean that many members post messages but do not receive replies from other members. The density may be used for an indicator which reflects a state of knowledge collaboration in their community. So community leaders or managers can help others discuss when the density is very low.

However, we need to note that values of the density are very sensitive against changes of the number of nodes (posters in this paper). Values of the density often show extremely high when the number of nodes is very small. It was very difficult to understand what high values of the density mean in our case study. For instance, the first local peak of the density value (08/25/01) in community C

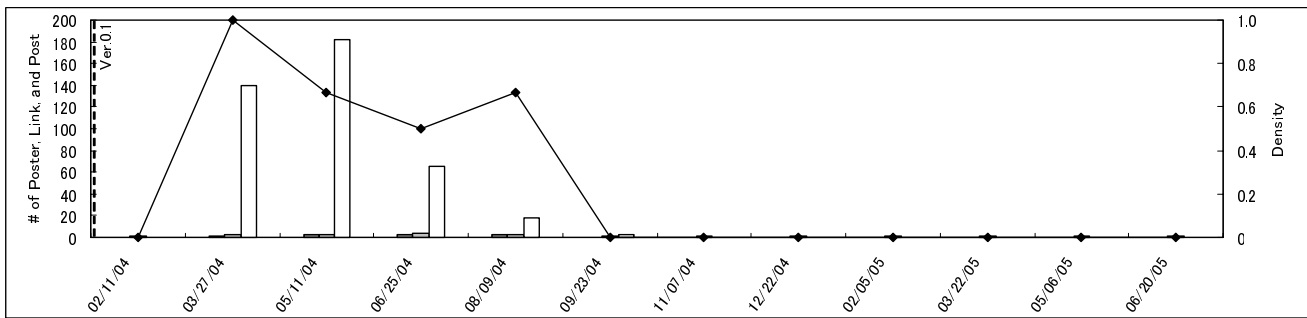
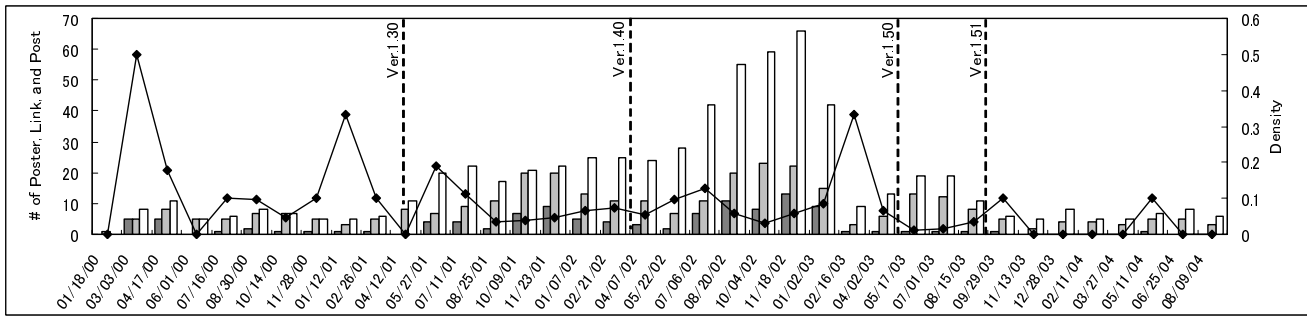
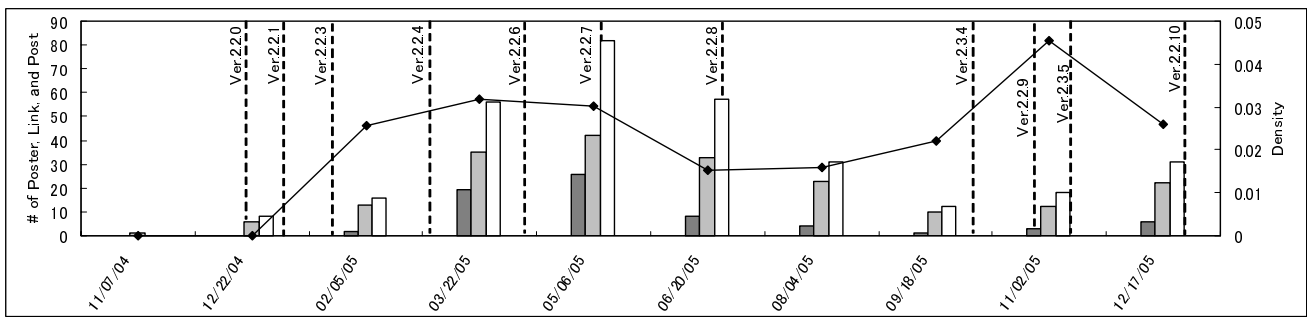
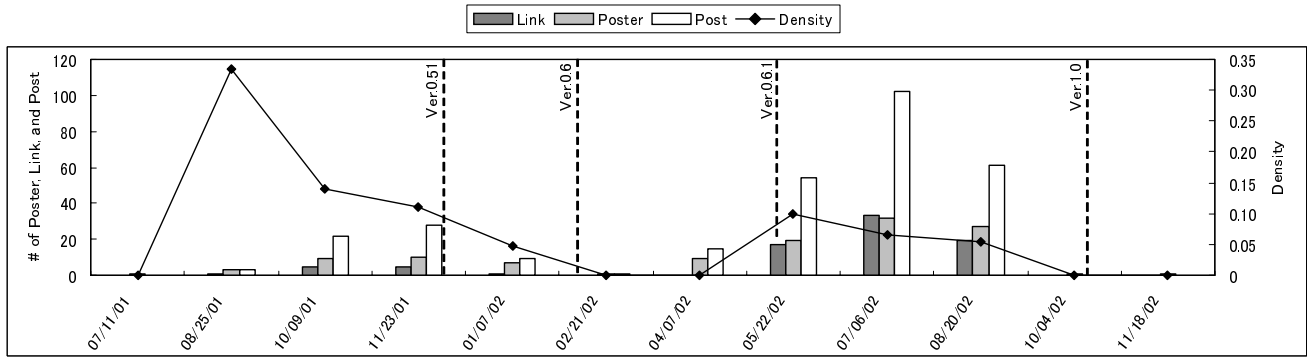


Figure 3. Analysis Results

does not mean knowledge collaboration is going well because only a small number of particular members discuss. This is applicable to community D while community D has a number of posts. In the future, we need to improve this difficulty in using the density.

## 7. Conclusions and Future Work

In this paper, we investigated the quality of communications for knowledge collaboration by time series analysis using the density of social networks. From the results of analyzing changes of the density in 4 OSS communities, our hypothesis (communications are actively encouraged before/after OSS released, especially among community members with a variety of roles but not among particular members.) was partly verified.

In the future, we will analyze the data by separating developers from end users to distinguish between development periods and feedback periods in more detail. And, we also need to analyze the data by considering structures of social relationships among community members though we did not include them in this paper.

**Acknowledgments** We would like to thank Shinsuke Matsumoto for helping us analyze OSS communities. This work is supported by the EASE (Empirical Approach to Software Engineering) community in the Comprehensive Development of e-Society Foundation Software program and Grant-in-aid for Scientific Research (B) 17300007, 2006 and for Young Scientists (B), 17700111, 2006, by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

## References

- [1] W. E. Baker. *Achieving Success Through Social Capital*. John Wiley & Sons Inc., 2000.
- [2] A. H. Dutoit and B. Bruegge. Communication metrics for software development. *IEEE Transactions on Software Engineering (TSE)*, 24(8):615–628, 1998.
- [3] J. Feller and B. Fitzgerald. *Understanding Open Source Software Development*. Addison-Wesley, 2002.
- [4] J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering (TSE)*, 29(6):481–494, June 2003.
- [5] K. R. Lakhani and E. von Hippel. How open source software works: “free” user-to-user assistance. *Research Policy*, 32(6):923–943, June 2003.
- [6] A. Mockus, R. T. Fielding, and J. D. Herbsleb. Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3):309–346, 2002.
- [7] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye. Evolution patterns of open-source software systems and communities. In *Proceedings of the International Workshop on Principles of Software Evolution (IWPSE’02)*, pages 76–85, New York, NY, USA, 2002. ACM Press.
- [8] J. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, 2000.
- [9] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [10] Y. Yamauchi, M. Yokozawa, T. Shinohara, and T. Ishida. Collaboration with lean media: how open-source software succeeds. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW’00)*, pages 329–338, New York, NY, USA, 2000. ACM Press.
- [11] Y. Ye. Dimensions and forms of knowledge collaboration in software development. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC’05)*, pages 805–812, Taipei, Taiwan, December 2005. IEEE Computer Society.
- [12] Y. Ye and K. Kishida. Toward an understanding of the motivation open source software developers. In *Proceedings of the 25th International Conference on Software Engineering (ICSE’03)*, pages 419–429, Washington, DC, USA, 2003. IEEE Computer Society.