

Exploration of the Open Source Software Community

Jin Xu
University of Notre Dame
jxu1@nd.edu

Gregory Madey
University of Notre Dame
gmadey@nd.edu

Abstract

The OSS community can be considered as a complex, self-organizing system. These systems are typically comprised of large numbers of locally interacting elements. Developers are main components in this network. The interaction between developers forms a collaborative social network. Study of the roles of developers and their activities can help us determine the development of projects. In this paper, we perform a quantitative analysis of Open Source Software developers by studying the whole developer community at SourceForge. Our research provides topological and evolutionary statistics for the OSS developer social network, which is helpful to understand the OSS phenomenon. Our work shows that OSS developer network is a scale free network.

Contact:
Jin Xu
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN46556

Tel: 1-574-631-7596
Fax: 1-574-631-9260
Email: jxu1@nd.edu

Key Words: Open Source Software, social network

Acknowledgement: Thanks to SourceForge for providing us data
Support: This research was funded in part by the NSF Award-0222829, from the Digital Society & Technologies Program, CISE/IIS.

Exploration of the Open Source Software Community

Jin Xu, Gregory Madey

The OSS movement is a phenomenon that challenges many traditional theories in economics, software engineering, business strategy, and IT management. The OSS community has developed a substantial amount of the infrastructure of the Internet, and has several outstanding technical achievements, including Apache, Perl, Linux, etc. These programs were written, developed, and debugged largely by part time contributors, who in most cases were not paid for their work, and without the benefit of any traditional project management techniques. A research study of how the OSS community functions may help IT planners make more informed decisions and develop more effective strategies for using OSS software.

The OSS community can be considered as a complex, self-organizing system [Madey 2004]. These systems are typically comprised of large numbers of locally interacting elements. The Open Source Software (OSS) development movement is a classic example of a dynamic social network; it is also a prototype of a complex evolving network. Developers are main components in this network. As shown in Figure 1, many developers may participate in one project. A developer may join many projects. The interaction between developers forms a collaborative social network. Study of the roles of developers and their activities can help us determine the development of projects.

Some researchers have begun to study OSS developers. Nakakoji et al. [Nakakoji 2002] classify OSS community members into different roles and study the influences of different members on the OSS system and the community in three OSS projects. A modified classification is presented by Xu [Xu 2003] to redefine OSS member roles which will be discussed in the next section. Crowston et al. [Crowston 2002] studied the OSS development teams on success factors for distributed work teams. By studying Linux Software Maps (LSMs), Dempsey et al. [Dempsey 2002] analyze the body of all extant LSMs at a Linux site to obtain information on the nature of Linux contributions and their contributors. Data mining techniques were used by Xu et al. to find patterns in the OSS developers' community [Xu1 2003]. Gao et al. [Gao 2003, Xu2 2003] simulate activities of core developers on SourceForge hosted projects.

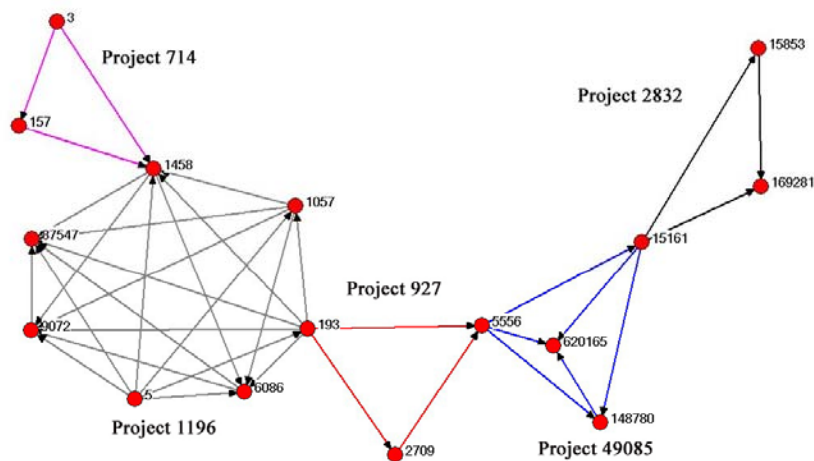


Figure 1: Developer Social Network, Linked by Joint Project Membership — Cluster of Size 16

(This graph is drawn by using UCINET [Ucinet])

All of these previous studies are either qualitative classifications or are performed on a small set of sample projects. In this paper, we perform a quantitative analysis of Open Source Software developers by studying the whole developer community at SourceForge. Our research provides topological and evolutionary statistics for the OSS developer social network, which is helpful to understand the OSS phenomenon. The work in this paper is the preliminary stage of our OSS community study. Based on these statistic data, we will develop agent-based models to simulate the development of the OSS community.

The rest of this paper is organized as follows: the next section describes the properties of OSS developer network; the third section classifies roles of developers by their activities in projects; Then, data collection and mining process are presented; Based on the collected data, statistic analysis is performed on the SourceForge developer community; lastly, conclusions and future work are given.

OSS Developer Network

The OSS developer network is a scale free network whose degree distribution follows a power law. According to Barabasi and Albert [Barabasi 1999], such a network possesses two properties:

- Unlike random networks which have a fixed number of nodes that are randomly connected, the network grows by the sequential addition of new nodes. In our OSS developer network, with the development of projects, developers sequentially join in projects.
- Unlike random networks in which the probability of two nodes being connected is independent of the nodes' degree, there exists "richer gets richer" phenomenon in scale free networks. The probability of two nodes being connected is related to the nodes' degree, which is called preferential attachment. In OSS, developers tend to choose more popular projects to participate.

OSS Developers Classification

An OSS developer community is composed of a group of loosely-connected contributors with some central coordinators and decision makers. According to Xu [Xu 2003], the OSS community can be classified as different roles:

- User Group
 1. Passive User: Passive Users have no direct contribution other than forming a larger user base. They just download source code and use it for their needs.
 2. Active User: Active users discover and report bugs, suggest new features, and exchange other information by posting messages to forums or mailing lists.
- Developer Group
 1. Peripheral Developer: Peripheral Developers irregularly fix bugs, add features, provide support, write documents, and exchange other information.
 2. Central Developer: Central Developers regularly fix bugs, add features, submit patches, provide support, write documents and exchange other information.
 3. Core Developer: Core Developers extensively contribute to projects, manage CVS releases and coordinate peripheral developers and central developers.
 4. Project Leader: Project leaders guide the vision and direction of the project.

In our study of SourceForge, the OSS developer community is defined to include all above roles except passive users. Thus, our OSS developer community is comprised of project leaders who are also called project administrators, core developers (also called member developers) who regularly contribute on projects and manage CVS releases, co-developers including both peripheral developers and central developers, and active users who have some contributions but not modifying the source code.

Data Collection

There are several web sites which host OSS projects. With around 79,000 projects and 830,000 users, SourceForge.net [Sourceforge], sponsored by VA Software, is the largest OSS development and collaboration site which offers a centralized place for OSS developers to control and manage OSS development by providing project web server, tracker, mailing lists, discussion boards, and software releases, etc. This site provides highly detailed information about projects and developers, including project characteristics, developers' activities, and "top ranked" developers. By studying SourceForge, we can explore developers' behaviors and projects' growth.

We gathered data from the 2003 data dump provided by SourceForge. The data dump is stored in a PostgreSQL relational database. The data dump contains information about the community, projects, and developers. We examined those data to characterize the entire SourceForge community, across multiple numbers of projects, investigating behaviors and mechanisms at the project and developer levels.

There are two ways to collect developers' information from SourceForge. The first way is to check each project's homepage because project leaders and core developers are listed for each project. However, there is no information about other developers such as peripheral developers and active users. Although we cannot directly get other developers' information from the SourceForge web site, we can gather their information by collecting their activities such as bug reports, patch submissions, and forum discussions. These activities are recorded in the database. By mining SourceForge database, we can retrieve developers who participate in a specific project and classify their roles according to their activities. We analyzed several tables in the SourceForge 2003 data dump. Two main tables we investigated are *artifact* containing information about developers' activities and *forum* containing their open forum discussions. By processing those tables, we can get developers' participation activities for each project. We used three-step data integration and data reduction to process data. Data integration combines data from multiple sources into a coherent store. Data reduction is used to reduce the huge data set to a smaller representative subset according to developers' role in a project.

Analysis of the SourceForge Developer Community

We classified developer roles in SourceForge as follows: project leaders are administrators in each project; core developers are members who control CVS releases and are listed in each project; co-developers (central and peripheral developers) are people who are assigned to tasks such as bug fixing and document writing, but are not listed as project leaders and core developers; active users are those who submit requests and post messages, but are not included in project leaders, core developers and co-developers; passive users are gotten by excluding all developers from all users. Figure 3 shows the distribution of developers in the whole SourceForge community. About 65% of the community is passive users who have no direct contributions to the development of projects. Among developers, there are 28.4% project leaders, 15.5% core developers, 33.9% central/peripheral developers and 22.2% active users. We observed that the central/peripheral developers have almost the same percentage as the sum of project leaders and core developers. This is because a large portion of projects on SourceForge are not so popular that almost all developers are initiators. (Detailed analysis of specific projects is under investigation.)

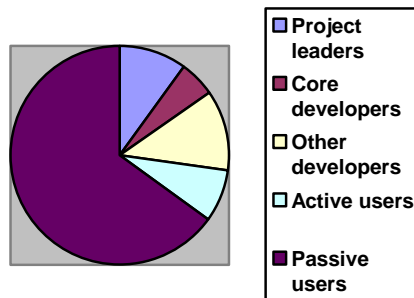


Figure 3: Distribution of SourceForge Community

Degree distribution is the frequency of the index value throughout the network. Degree distribution was believed to be a normal distribution, but Albert and Barabasi recently found it fit a power law distribution in many real networks [Albert 1999]. Figure 4 gives developer distributions in SourceForge community. The X coordinate is the number of projects in which each developer participated, and the Y coordinate is the number of developers in the related categories. The right sub-graph shows the distribution based on the log scale. From the figure, we can observe that the developer distribution matches the power law. Such power law distribution proves that the SourceForge developer network is a scale free network. In this network, developers sequentially choose more popular projects to join. Thus, a popular project tends to attract more and more developers, while less popular project sometimes can not even survive after a while. (More results will be presented during the conference.)

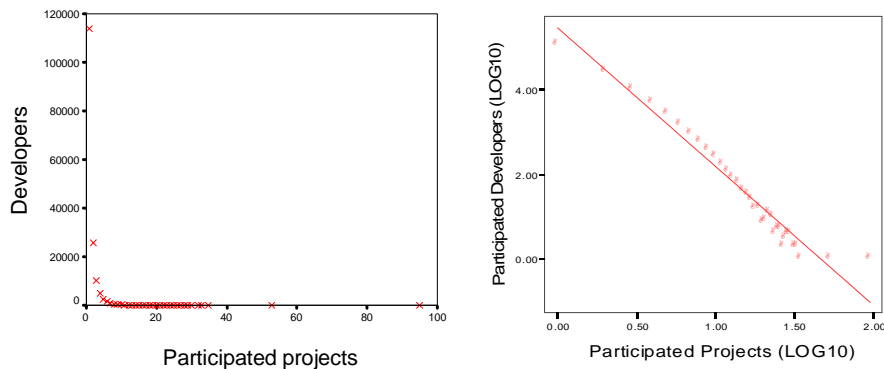


Figure 4: Degree Distribution of Developers

Conclusions

In this paper, we classify and study Open Source Software developer network of SourceForge. The data collection design and process are described. By gathering data from SourceForge 2003 data dump, we perform a quantitative analysis of OSS developers' community. Our research provides useful information to study the development of OSS projects. Future work will focus on the simulation of OSS developer network based on the statistic results in this paper.

References

- [Albert 1999] Albert R., Jeong H., Barabasi A. L., "Diameter of the World Wide Web", *Nature*, V. 401, P. 130-131, 1999.
- [Barabasi 1999] Barabasi A. L., Albert R., "Emergence of Scaling in Random Networks", *Science*, v. 286, p. 509-512, 1999.
- [Dempsey 2002] Dempsey B. J., Weiss D., Jones P., Greenberg, J. "Who is an Open Source Software Developer?", *Communications of the ACM*, v. 45, n. 2, p. 67-72, February 2002.
- [Gao 2003] Gao Y. Q., Vincent F., Madey G., "Analysis and Modeling of the Open Source Software Community", North American Association for Computational Social and Organizational Science (NAACSOS 2003), Pittsburgh, PA, 2003.
- [Madey 2004] Madey G., Freeh V., and Tynan R., "Modeling the F/OSS Community: A Quantitative Investigation," in *Free/Open Source Software Development*, ed., Stephan Koch, Idea Publishing, 2004.
- [Nakakoji 2002] Nakakoji K., Yamamoto Y., Kishida K., Ye Y., "Evolution Patterns of Open-source Software Systems and Communities", *Proceedings of The International Workshop on Principles of Software Evolution*, Orlando Florida, May 19-20, 2002.
- [Sourceforge] <http://www.sourceforge.net>
- [Ucinet] Borgatti, S.P., Everett, M.G. Freeman, L.C., "Ucinet for Windows: Software for Social Network Analysis", Harvard, MA: Analytic Technologies, 2002.
- [Xu 2003] Xu N., "An Exploratory Study of Open Source Software Based on Public Project Archives", Thesis, the John Molson School of Business, Concordia University, Canada, 2003.
- [Xu1 2003] Xu J., Huang Y., Madey G., "A Research Support System Framework for Web Data Mining", *Workshop on Applications, Products and Services of Web-based Support Systems at the Joint International Conference on Web Intelligence (2003 IEEE/WIC) and Intelligent Agent Technology*, Halifax, Canada, October 2003.
- [Xu2 2003] Xu J., Gao Y., Goett J., Madey G., "A Multi-Model Docking Experiment of Dynamic Social Network Simulations", *Agents2003*, Chicago, IL, October 2003