www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13549-13552

Online Community of Open Code Development Project

Rehan Khan[1]

[1]*Computer Science Department, Maharshi Dayanand University*
Rohtak, Haryana, India
rehankhan5348@gmail.com

Abstract—Today's first technical need is a free platform for usability and if this platform could be customized by the user itself then it creates a unique community named as Open Source. This community provides many different technicalities in operating systems, applications, software, gaming etc. How this community takes its members and let them open their codes to the world for better use and modification. This is what we actually need to know about, how the biggest free and open source community gets its work, develops it, distributes it, use it and modify it without directly monitoring by the profit industries. Their work interacts with many e-commerce or e-businesses missions. Today every computer science student, lecturer, professor or administration should aware of this free and open source development phenomenon to participate with ongoing easy, batter, secure and modifiable environment which cost nothing but gives us millions.

Keywords—Open Source Software, Social Network, Developers

I. INTRODUCTION

The OSS movement is a phenomenon that challenges many traditional theories in economics, software engineering, business strategy, and IT management. Thousands of software programmers are spending tremendous amounts of time and effort writing and debugging software, most often with no direct monetary compensation. The programs, some of which are extremely large and complex, are written without the benefit of traditional project management, change tracking, or error checking techniques. Since the programmers are working outside of a traditional organizational reward structure, accountability is an issue as well. A significant portion of internet e-commerce runs on OSS, and thus many firms have little choice but to trust mission-critical e-commerce systems to run on such software, requiring IT management to deal with new types of socio-technical problems. A better understanding of how the OSS community functions may help IT planners make more informed decisions and develop more effective strategies for using OSS software. I hypothesize that open source software development can be modeled as self-organizing, collaboration, social networks. There is define two software developers to be connected part of a collaboration social network if they are members of the same project, or are connected by a chain of connected developers at SourceForge.net. Project sizes, developer project participation, and clusters of connected developers are analyzed. I find evidence to support our hypothesis, primarily in the presence of power-law relationships on project sizes (number of developers per project), project membership (number of projects joined by a developer), and cluster sizes. Potential implications for IT researchers, IT managers, and governmental policy makers are discussed.

II. AN OVERVIEW OF SOCIAL DEVELOPER NETWORK TYPES

A. Heterogeneous Social Network

A heterogeneous social network holds topological information and relational information and consists of more entity types. Each vertex is recognized using its neighbor relations. A relation is combination of directly and indirectly connected nodes and connectors. Heterogeneous Social Network HSN (Vertices, Edge, Label) is directed labeled graph, where Vertices is a limited group of nodes, Label is limited set and Vertices x Label x Vertices is a limited group of edges. Most algorithm for example Meta-path analysis, ranking, similarity analysis and group similar activities are operation mainly performed in this network.

B. Homogeneous Social Network

Homogeneous social network assumes single type of vertices and edge relations. This kind of analysis method generally produces loss of information in the process. In homogeneous social network, vertices denote entity and edge shows relations. Available algorithm and methods for example ranking, similarity search, clustering and group similar activities and association relation prediction.

C. Multidimensional Social Network

Denotes multiple kinds of relationship, Multi-dimensional network containing every dimension constituting user association at each site e.g. Facebook, Twitter, YouTube and Orkut etc. Users relatedness with each other on several activities leads to multiple network. Establish community relation in multi-dimensional network by calculating the similarity between two items in some dimension (entity) or different dimension (entity) from the network based in probability distribution of each dimension or entity. Their result shows that purposed algorithm is effective and efficient.

III. OPEN SOURCE SOFTWARE DEVELOPMENTS

Open source software is by definition software for which users have access to the source code. This distinguishes it from the recent common practice by commercial software publishers of only releasing the binary executable versions of the software. Most open source software is also distributed at no cost with limited restrictions on how it can be used; hence the term free when used to describe open source carries two meanings: 1) free of cost and 2) free to do with the software as you wish (i.e., most importantly free to read the code).

Case studies documenting the open source software development model, point to potential lessons and benefits that may be of value to corporate IT. It is claimed that open source development produces more bug-free code, faster, than closed proprietary developed code, although this has yet to be conclusively demonstrated. Open source software development teams, are generally comprised of volunteers working not for monetary return, but for the enjoyment and pride of being part of a successful virtual software development project. Team members often come from around the world and rarely meet one another face-to-face. The open source projects are self-organized, employ extremely rapid code evolution, massive peer code review, and rapid releases of prototype code. Many of these practices are counter intuitive and the opposite of what conventional software engineering holds as the correct processes for the production of high quality code.

The Open Source Software movement is a prototypical example of a decentralized self-organizing process. There is no central control or central planning. It challenges conventional economic assumptions, it turns conventional software engineering and project management principles inside out, it threatens traditional proprietary software business strategies, and it presents new legal and government policy questions regarding software licensing and intellectual property. Moreover, OSS is a major component of the IT infrastructure enabling global e-Commerce. Open source software including BIND, sendmail, Apache, Linux, INN, GNU utilities, MySQL, PostgreSQL, and Perl are critical components of the Internet. They enable major services hosted on the Internet, e.g., e-mail, WWW, e-Commerce, domain name lookup. The Netcraft.com survey of 36.6 million web servers worldwide reports an over 60% market share for the open-source web-server Apache (Netcraft, 2002) and now it is 90%.

IV. DATA COLLECTION AND ANALYSIS

I gathered data monthly over the 3 month period at SourceForge, a web-based project support site sponsored by VA Software and owned & operated by Slashdot Media. SourceForge provides project management tools, bug tracking, mail list services, discussion forums, version control software for over 3.7 million open source developers, participating on over 430,000 projects and serves more than 4,800,000 downloads a day. I note that not all open source projects are registered with SourceForge; many high profile projects maintain their own developer sites, e.g., Apache, Perl, sendmail, Linux. But some large projects have moved to SourceForge (e.g. Samba) and I speculate that there are many smaller projects that have not joined SourceForge. My assumption is that the projects at SourceForge are representative of the overall open source movement, in part because of its popularity and the large number of projects and developers registered there.

The primary data required for this research is a table consisting of records with two fields: project number and developer ID. Because projects can have many developers and developers can be on many projects, neither field is unique primary key. Thus the composite key composed of both attributes serves as a primary key. Each project in SourceForge has a unique project number. Additionally, each developer is assigned a unique ID when registering with SourceForge.

A web crawler traversed the SourceForge web server to collect the necessary data. All project home pages in SourceForge have a similar top-level design. Many of these pages are dynamically generated from a database. In particular, the developers belonging to a project are found by issuing the following request:

http://sourceforge.net/project/memberlist.php?group_id=projnum

A simple shell script fetches each project's developer page, and then parses the HTML, extracting the names of the developers. A python program parses the HTML source. It outputs one line for each developer, which contains the project number and the developers ID.

The above script (and auxiliary programs) creates a file of project numbers and developer IDs. Below is an extract of this file:

```
8001|dev378

8001|dev8975

8001|dev9972

8002|dev27650

8005|dev31351

8006|dev12509

8007|dev19395

8007|dev4622

8007|dev35611
```

V. OSS NETWORK STRUCTURE

The structural data was collected at SourceForge.net, the largest Open Source Foundry (SourceForge, 2014). SourceForge is a free hosting service for Open Source projects which offers, among other things, web site hosting, mailing lists, bug tracking, message forums, and task management software.

I found the model of OSS developers and projects as a network in two complementary ways. First, each developer is a node in the network; an edge exists between nodes if both developers are on the same project as shown in Figure 1. In that figure we observe two linchpin developers, dev[58] and dev[46], who tie 5 separate projects into a cluster of 24 developers. This representation is analogous to movie actors as nodes and movies as links, or research paper authors as nodes and joint authorship as a link in the collaboration networks discussed above. The second way uses projects as nodes. My initial analysis of the structural data shows that the developer collaboration network at SourceForge fits a power-law model, as determined by ordinary least squares (OLS) regression in log-log coordinates. The project-size (number of developers on the project) and the number of projects per developer (total number of projects-joined by a developer) have power-law distributions. The solid line is the OLS regression line though the data, with an adjusted R = .93 for the project-size data, and an adjusted R = .97 for the projects-joined data. This power-law distribution is often a property of such self-organizing systems.

The small projects have not had time to attract linchpin developers to tie the many projects into a large cluster linking a large number of registered developers. Alternatively, the SourceForge site is serving a critical role, by linking developers that might not normally be connected. This suggests that a longitudinal study of the growth of the open source network is needed to follow the attachment, detachment, and evolution of that network.

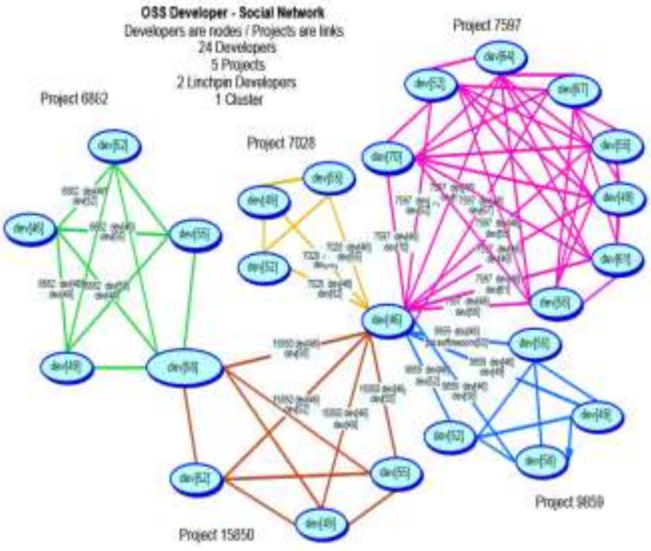

Figure 1. Developer Social-Network, Linked by Joint Project Membership, Cluster of Size 24

In most collaboration networks that have been studied, links are persistent. For example, actors are linked if they appeared together in a movie. This link is never removed, and it does not require any additional effort by the actor to maintain. But in the developer network, the link exists only while the developer is a member of a project. Thus the links in this network are transient, and node degree (number of edges) of developers is much smaller. Over the 2 months of my initial survey, the "busiest" developer (on the most projects) has fluctuated between membership on a low of 17 and a high of 27 projects, while the number of developers belonging to exactly one project has remained a nearly constant 80%.

SourceForge rates projects by activity, such as downloads, updates, and page hits. This allows one to find distinguishing characteristic of top projects. Looking at the top 10 projects for the month of August 2014, I find that there is an average of 29 developers per project, 20 times the SourceForge average. Also these developers belong to 25% more projects than the overall population. Finally, 70% of the developers on top projects belong to exactly one project (versus 80% overall), but the maximum is only 12.

VI. RESULTS

The results support our speculation that the open source movement is not a random graph (i.e., new nodes attach to existing nodes with uniform probabilities), but a graph displaying preferential attachment of new nodes (i.e., some nodes have higher probability of attachment than others). This typically happens under situations of positive feedback or increasing returns and is sometimes call the rich-get-richer effect or the band-wagon effect. In the open source movement, initial success breeds more success because developers prefer to be part of a successful project. Since the developers are free to self-organize and select the projects they choose to join, it is reasonable to expect that some projects will be more visible or more attractive than others, hence some projects will grow disproportionately larger than expected under random growth. Additional data collection and analysis will be needed to resolve this question. Should open source networks be determined to have these power-law relationships, then by inheritance the new results of the ongoing research on similar networks (research authors, actors, the Internet, Web pages, etc.) would also apply to the open source movement. This would enable better modeling of the processes associated with this development strategy, hence supporting research in this area and possibly enabling better implementation of such open source development strategies by IT organizations. For example, we observe the importance of the linchpin nodes in growing larger clusters. In the case of open source development, these developers play a similar role to the gatekeepers in organizational studies on technology diffusion. These linchpin developers may need to be identified, nurtured, and supported in their role of facilitating the diffusion of ideas and technology between disparate development groups.

VII. SUMMARIES AND CONCLUSION

I describe an empirical study of the open source projects registered at SourceForge. I believe they are representative of open source movement worldwide. Those projects were modeled as a collaborative social network, with developers as nodes and joint membership in projects as links between the nodes. Analysis of the data displays a heavily skewed distribution, which has a good fit to a power-law relationship. Previous studies of social and collaborative networks with similar properties are believed to grow not as random networks, but as preferentially connected networks. My study suggests that the same may be true of the open source movement. If this observation is true, then the active research on other such social networks may produce insights that may be applied to further research on open source software.

Several assumptions and limitations are present in the study. I assume that the projects at SourceForge are representative of open source projects in general. This needs to be confirmed. Although I have collected monthly data over 4 months, my analysis only looked at a monthly snapshot of the open source network at SourceForge. Once several months of data are collected, a longitudinal and dynamic analysis may provide better understanding of how node attach and detach from the network. Data on developers who dropped off of projects was not analyzed. We consider only the linking relationship of joint project membership; many of the developers are linked through other relationships, e.g., shared subscriptions to newsletters, listerservs, or reading common web pages. The effect of those other linking relationships, along with the effect of SourceForge itself, should be further investigated. Is the open source movement highly fragmented with SourceForge helping to link those fragments together into a larger connected collaborative cluster?

REFERANCES

[1] Greg Mandy, Vincent Freeh, Renee Tynan “Open Source Software Development Phenomenon” Eight Americas Conference On Information System-2002

[2] Anil Kumar “Evolution Of Social Developer Network In OSS: Survey” International Journal of Research in Engineering and Technology-2014

[3] Chang-Te Li and Shou-De Lin “Centrality Analysis, Role-based Clustering, and Egocentric Abstraction For Heterogeneous Social Networks” International Conference on Social Computing (PASAT, SocialCom), IEEE-2012

[4] Lei Tang, Xufei Wang and Haun Liu “Community Detection via Heterogeneous Interaction Analysis. Knowledge Discovery and Data Mining (DMKD)-2012

[5] Xutao Li, Ng, M.K., Yunming Ye “Finding Community Structure in Multi-dimensional Networks” Knowledge and Data Engineering, IEEE-2014

[6] SourceForge "SourceForge Home," (August 2014)

[7] Barabasi, A.L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., Viscek, T. "Evolution of the Social Network of Scientific Collaborations," (April 10, 2001)

[8] Bollinger, T. "Linux and Open-Source Success: Interview with Eric. S. Raymond," IEEE Computer, 1999.

[9] Charles, J. "Open Source: Netscape Pops the Hood," IEEE Software, 1998.

[10] Edwards, J. "The Changing Face of Freeware," Computer, 1998.

[11] Faloutsos, M., Faloutsos, P., Faloutsos, C. "On Power-Law Relationships of the Internet Topology," Cambridge, MA, 1999.

[12] Feller, J.F., B. "A Framework Analysis of the Open Source Software Development Paradigm," 2000, Brisbane, Australia, 2000.

[13] Hars, A., Ou, S. "Working for free? Motivations of participating in Open Source Projects," Proceedings of the Hawaii International Conference on Systems Sciences, 2001.

[14] Hecker, F. "Setting up Shop: The Business of Open-Source Software," IEEE Software, 1999.

[15] Jin, E.M., Girvan, M., Newman, M. E. J. "The Structure of Growing Social Networks," Santa Fe:01-06-032, 2011.

[16] Jorgensen, N. "Putting it all in the Trunk: Incremental Software Development in the FreeBSD Open Source Project," Information Systems Journal 2013.

[17] Koch, S., Schneider, G. "Effort, Co-operation and Co-ordination in an Open Source Software Project: GNOME," Information Systems Journal 2013.

[18] Madey, G., Freeh, V., Tynan, R. "Agent-Based Modeling of Open Source using Swarm," AMCIS2002, Dallas, TX, 2002. Netcraft "Netcraft.com Web Server survey," (Feb. 16, 2013).

[19] O’Reilly, T. "Lessons from Open-Source Software Development," Communications of the ACM (42:4), 1999, pp. 33-37.

[20] Scacchi, W. "Understanding the Requirements for Developing Open Source Software Systems,"

[21] Sharma, S., Sugumaram, V., Rajagopalan, B. "A Framework for Creating Hybrid-Open Source Software Communities," Information Systems Journal 2013.

[22] "Open Source Software: A Status Report," IEEE Software, 2010.