Workings of Collective Intelligence within Open Source Communities

Everett Stiles and Xiaohui Cui

Electrical Engineering and Computer Science Department,
University of Tennessee,
Knoxville, TN 37996
estiles@utk.edu
Computational Sciences and Engineering Division,
Oak Ridge National Laboratory,
Oak Ridge, TN 37831-6085
cuix@ornl.gov

Abstract. Open source communities have been of great interest for researchers recently, yet little can be agreed upon when it comes to developers motives. While it has been shown that participants are mostly driven to contribute based on work related needs, it has also been shown that they contribute to fulfill an ideological purpose. We believe that the majority of participants contribute to satisfy their own personal goals. We reveal how developers function as a collective intelligence by modeling the open source community as a disjoint group of contributors. We show that most developers contribute to only one project and only to a small portion of its source code. We demonstrated that useful functionality of most OSS software is an emergent phenomenon created by a collection of developers with different motivations and personal goals.

Keywords: open source software, developers, collective intelligence, sourceforge, concurrent versions system.

1 Introduction

It has been proposed that open source software is heavily reliant on the community that developed it [2]. Without the collective effort, social interactions, and group influences of open source communities, the development and adoption of said software would not occur [2]. The view of developers laboring together to achieve a common purpose is an attractive, if unproven, assumption. Previous research has revealed that the population of open source developers is disjoint with the largest cluster consisting of only 25% of the total number of developers [12]. Even though their conclusions will be shown as incorrect, they did present evidence that suggested that the open source community does not function as a whole. We wish to address this inaccurate portrayal of the open source community directly and demonstrate, using the popular open source development website SourceForge, that virtually all developers within the open source community work independent of one another.

S.-K. Chai, J.J. Salerno, and P.L. Mabry (Eds.): SBP 2010, LNCS 6007, pp. 282–289, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Our hypothesis is the open source community and their achievements can be viewed as a collective intelligence model. Developers spend their time attempting to achieve their own goals, while their contributions collect and emerge as a related collection of useful functionality that can compete with peripheral software. A few developers, which we have named long term contributors, bind this functionality into a marketable application and do their best to maintain it. The open source software community is the collective workings of individual developers with different motivations and personal goals. In this paper, we presented our research results to demonstrate that this collective intelligence model is a reasonably accurate portrayal of the open source community thereby, correcting the held assumption that it is the open source communitys intention to labor together to achieve a common ideological goal.

2 Related Work

Open source communities have been researched greatly over the last couple of years, but not much research on developers motives has been agreed upon. Views that explain participation have ranged from the free software ideology [15] to enjoyment and creativity [4][5][16]. Other views that have been propagated include career concerns, learning, and reputation [7][11]; affiliation and identity [8]; and the developers desire to fulfill personal needs [3][9][10][13]. Although survey evidence exists to support each of these theories [6][8], we feel that the majority of developers would be contributing in an attempt to achieve their own objectives.

In the article titled Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development, Shah attempts to address these conflicting opinions by conducting interviews with participants from the open source community [14]. Virtually all interviewees reported that their initial contribution was motivated by a need brought on by a work related purpose. Open source software was used to fulfill their work related purpose due to the relative ease of licensing as opposed to contract negotiated closed source software. Their reason for contribution was usually caused by the absence of their required feature. Shah states, Because a need exists, participants generally do not wait for others to solve the problem. Developers cooperate with other individuals in order to avoid conflicts within projects, but tended to focus on their own needs. Pertaining to the survey evidence from previous research, Shah found that a small subset of original contributors continue contributing long after their initial needs are met. These long term contributors tended to be highly skilled software developers who held managerial positions in their companies of employment. Open source software development provided them with a venue to solve challenging problems while taking advantage of their creativity. Interviewees indicated that these long term contributors are the ones who commit code, rewrite sections, fix bugs, and design software releases. These contributors themselves reported that they monitor mailing lists, bug reports, and code contributions in order to receive feedback and improve the project's code.

The items that long term contributors are tasked with completing are the only visible signs that the open source community exists: committing code, rewriting sections, fixing bugs, designing software releases, maintaining mailing lists, reading bug reports and improving existing code. If only a small portion of the open source community consists of long term contributors and long term contributors produce the visible signs that a community exists then the majority of developers are short term contributors and do not partake in the communities objective.

3 Experiment

Our aim is to show that the open source community consists of individuals carrying out their own agendas. If shown, it would follow that another model, other than a common ideology, would best represent the motives and behavior of developers within the community. We highlight details that reveal most developers contribute to a single project, and only a small portion of a project is impacted by any developers contributions. The methods and results are outlined below.

4 Data and Methods

Our data collection focuses around SourceForge, the largest open source development website that hosts a large variety and number of open source projects. Our aim is to demonstrate that open source developers contributions are limited to a narrow scope within the open source community. We will address this in two stages. In the first stage we will use an external database, containing information collected from SourceForge over the last six years, to analyze developer participation at a community level. The second stage will encompass the use of SourceForge's Concurrent Versions System (CVS). Using this record we will examine developers contributions at a project level. Both stages of observation are discussed in more detail below.

The University of Notre Dame hosts an open source database (`http://www.nd.edu/~oss/Data/data.html`) which maintains a vast amount of data collected from SourceForge. The data collection started with a dump of SourceForge's content in January of 2003. Starting in February of 2005, this dump became monthly and consists of endless resources for research on the open source community. We will use this wealth of knowledge to demonstrate that most participants, within the open source community, only participate in a single open source project. We believe this would be a fundamental blow to the idea of a well connected social community and strengthen the proposal that participants our fulfilling their own independent needs. The databases information however is limited, and it cannot be used to accurately determine a single developers contributions to a given project. We use Concurrent Versions System (CVS) data to bridge this gap and determine the knowledge developers have of a project.

The Concurrent Versions System, also referred to as CVS, maintains the history of all changes within a project. This system, an open source project itself,

is used widely throughout the SourceForge community. Its purpose is to track changes within source code and record appropriate information that can used to revert back to previous versions. This information includes details on the change along with the author who committed it. By using the CVS historical record, we are able to get each individual developers contribution to a project at any given time by counting the lines of codes he/she had modified or developed at that time. We use these details to show that the knowledge individual developers have of a project is greatly limited. If this can be shown then it can also be assumed that developers lack the necessary knowledge to contribute towards a unified goal.

In our project, the CVS data was collected directly from SourceForge through the servers that they provide. Using their system and a standard CVS client, the entire source code repository for a number of selected projects was downloaded. Through further use of the CVS client, the projects history was extracted using a standard command named log. This history was used to create our results.

5 Results

5.1 Particpant's Community Contributions

Our first hypothesis is: most participants within the open source community contribute to a single project. In order to demonstrate this hypothesis, we collected data that reflected participation within open source projects. For each month that data was available, we calculated the percentage of developers assigned to one project, two projects, and so on. The data for each month was an accumulation from all previous months. We have presented these findings in Fig. 1.

An almost unchanging trend that persists from 2003 to present can be observed. Data points that reflect participation during the months of July to September 2007 are corrupted due to a privacy issue SourceForge had with the database that was used (`http://zerlot.cse.nd.edu/mediawiki/index.php?title=User_group`). The data demonstrates that over 75% of all developers in the history of SourceForge have taken part in no more than one project. It shows that out of the 246,114 developers reported to be participating in the month of July 2009, only 55,873 are participating in more than one project. However these results do include projects that have been inactive for some time. To receive results that more accurately describe the month being analyzed, the data was filtered each month to include only active projects. Active projects were defined as projects that had issued a release of their software within two years prior of the month being evaluated.

Fig. 2 contains the filtered project participation results for active projects. Similarly to the unfiltered data, the months of July to September 2007 are corrupted, but differences are clearly shown. Approximately 85% of developers on active projects contribute to one project with less than 4% of developers contributing to more than three. A linear trend can be extracted to show that the percentage of developers working on a single project is rising. Likewise, linear trends can show that participation in multiple projects is declining.

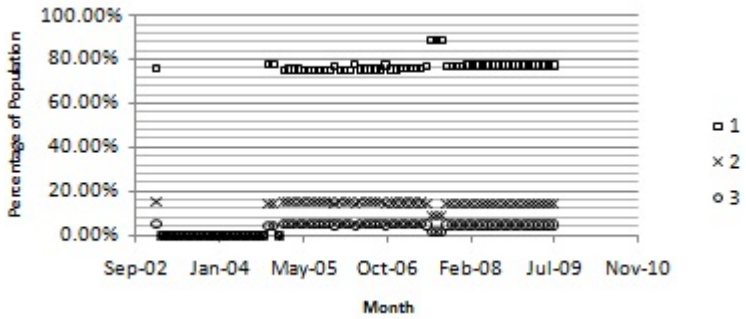

Fig. 1. Developer Participation on SourceForge

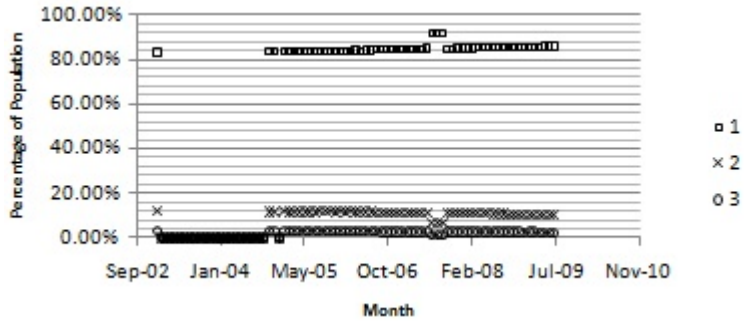

Fig. 2. Developer Participation within Active Projects on SourceForge

5.2 Particpant's Project Contributions

Our second hypothesis is: virtually all developers do not have knowledge of the entire project. This hypothesis can be proofed by examining each developers participation levels within the projects. To observe these levels, we measured the amount of knowledge participants had pertaining to projects in which they participated. SourceForge's Concurrent Versions System contained a repository of information that was able to perform this measurement. We analyzed projects with the highest amounts of participation during the month of July 2009. Out of the top 20 projects identified, 10 of these projects, including the top 4, were available to be analyzed using CVS repository. We measured the percentage of files that developers had made contributions to within a single project. A developer had to edit at least one line of a file to be counted as a contributor. Measurements are listed in Table 1. The number of participants for each project and the percentage of files developers contributed to are indicated.

Table 1 indicates that developers have a narrow scope of their projects. In some of these projects, the majority of developers have never contributed. Since there are relatively no project members participating in all coding efforts within a project, we can conclude that virtually all developers do not have knowledge of the entire project.

Table 1. Developer's Scope within Projects

Number of Participants	0%	0%-1%	1%-10%	10%-20%	20%-50%	50%-100%
84	61	16	4	2	0	1
93	0	70	20	0	3	0
94	56	22	10	3	1	0
103	34	40	20	3	6	0
112	28	57	20	7	0	0
129	12	83	28	4	2	0
151	77	53	20	1	0	0
162	73	64	24	1	0	0
249	68	136	44	1	0	0
428	204	184	32	4	3	1

6 Discussion

Past research [14] revealed virtually all participants in the open source community started contributing due to a work related need. This combined with the observation that nearly all participants discontinue their contributions within one year [14], we can assume that the majority of participants are fueled to contribute based on those needs. Our research results demonstrated that contributions from each developer were very narrow. Contributions from a single developer were mainly funneled into a single project and only impacted a small percentage of that project. This reinforces the notion that developers are driven to achieve their own objectives while contributing to the open source community.

Long term contributors, who likely started contributing through their own needs, continue to exercise authority over open source projects. Through code revisions and software releases, they guide the progress and future of open source projects. In our findings, we discovered some developers within the open source community that worked on a great number of projects and others that had contributed to a large portion of the project in which they were participating in. Determining if these ambitious developers are the same as the long term contributors, that maintain the community, should be an objective of future research.

With most participants working to achieve their own goals and long term contributors maintaining the projects, functioning software is produced. This conclusion proofed our hypothesis that the open source community software development phenomenon can be explained by modeling the community as a collective intelligence. All developers goals within a single project can be assumed to be related in some form, since they are all tied to the same project. As developers goals are fulfilled, the project will start to reach a functioning state. The long term contributors are the only ones needed to maintain that functioning state as the project grows with expanding functionality.

7 Conclusion

Our research has reinforced the idea that developers within the open source community are driven to contribute based on their personal needs. In this paper, we developed a collective intelligence model in order to explain how useful open source software can be created by developers who only contribute to satisfy personal needs. While this view does not hold for every contributor, it represents a large portion of the open source community. Based on our research presented in this paper, we have proofed that most participants within the open source community contribute to a single project and virtually all developers do not have knowledge of the entire project. The developers make contributions based on their personal needs, while their contributions collect and emerge as a related collection of useful functionality.

The scope of our data was limited to SourceForge, and it is still unknown if this community is the best sample to represent the open source community. Our analysis of project contributions could only be done on projects that made use of SourceForges CVS. Our sample did include fifty percent of the top 20 projects participated in, but this may be significant. The research presented in this paper did not demonstrate or highlight any differences among short term and long term contributors. Our aim was to give insight into how the majority of the open source community operates. Future research will aim at addressing these differences. Research on the open source community is still in its prime. We feel there are still benefits to be had even though we have learned a great deal. A self governing body, that produces useful society constructs, should be researched until we can recreate what nature has already done for us.

Acknowledgements

This research was done at Oak Ridge National Laboratory as part of the Department of Energys Internship program. Oak Ridge National Laboratory is managed by UT-Battelle LLC for the US Department of Energy under contract number DEAC05-00OR22725. This work was supported in part by the Lockheed Martin Corporation Shared Vision fund and Oak Ridge National Laboratory Seed Money fund (3210-2276). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Oak Ridge National Laboratory, the Department of Energy or the United States government. This manuscript has been authored by UT-Battelle, LLC, under contract DEAC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

1. Notre Dame OSS Database (2009), http://www.nd.edu/~oss/Data/data.html
2. Bagozzi, R.P., Dholakia, U.M.: Open source software user communities: A study of participation in Linux user groups. Management Science 52(7), 1099–1115 (2006)
3. Franke, N., von Hippel, E.: Satisfying heterogeneous user needs via innovation tool kits: The case of apache security software. Res. Policy 32, 1199–1215 (2003)
4. Gelernter, D.: Machine Beauty. Basic Books, New York (1998)
5. Ghosh, R.A.: First Monday interview with Linus Torvalds: What motivates free-software developers. First Monday 3(3) (1998), http://www.rstmonday.org/issues/issue3_3/torvalds/
6. Ghosh, R.A., Glott, R., Krieger, B., Robles, G.: Free/libre and open source software: Survey and study. Report, International Institute of Infonomics, University of Maastricht, Maastricht, The Netherlands (2002)
7. Hann, I.H., Roberts, J., Slaughter, S., Fielding, R.: Delayed returns to open source participation: An empirical analysis of the Apache HTTP Server Project. Working paper. Carnegie Mellon University, Pittsburgh, PA (2002)
8. Hertel, G., Niedner, S., Hermann, S.: Motivation of software developers in open source projects: An Internet based survey of contributors to the linux kernel. Res. Policy 32, 1159–1177 (2003)
9. Kuan, J.: Open source software as consumer integration into production. Working paper, SSRN (2001), http://ssrn.com/
10. Lakhani, K., von Hippel, E.: How open source software works: Free user to user assistance. Res. Policy 32(6), 923–943 (2003)
11. Lerner, J., Tirole, J.: The simple economics of open source. J. Indust. Econom. 52(6), 197–234 (2002)
12. Madey, G., Freeh, V., Tynan, R.: The Open Source Software Development Phenomenon: An Analysis based on Social Network Theory. In: Proceedings of the Americas Conference on Information Systems (AMCIS 2002), Dallas, TX, pp. 1806–1813 (2002)
13. Raymond, E.: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. OReilly & Associates, Sebastopol (1999)
14. Shah, S.K.: Motivation, governance, and the viability of hybrid forms in open sources software development. Management Science 52, 1000–1014 (2006)
15. Stallman, R.: Philosophy of the GNU project (2001), http://www.gnu.org/philosophy/
16. Weizenbaum, J.: Computer Power and Human Reason: From Judgment to Calculation. W. H. Freeman, San Francisco (1976)