A Longitudinal Study on Collaboration Networks and Decision to Participate in a FLOSS Community

Guido Conaldi
Institute of Management,
University of Lugano,
CH-6904 Lugano, CH.
guido.conaldi@usi.ch

Marco Tonellato
Institute of Management,
University of Lugano,
CH-6904 Lugano, CH.
marco.tonellato@usi.ch

ABSTRACT

In this paper we conjecture that individual decisions of FLOSS (Free/Libre Open Source Software) developers to take on a task are influenced by network relations generated by collaboration among project members. In order to explore our conjecture we collected data on a FLOSS project team consisting of 227 developers committed since 2002 to the development of a web browser. We reconstructed 2-mode co-collaboration networks (software developer by bug) in which a tie represents an action taken by a developer in order to solve a specific bug. Co-collaboration networks were collected at five points in time during a six-month development cycle of the software. We report and discuss results of longitudinal actor-based modeling that we specify to test for the influence of local network structures on developer's decision to take action on a specific bug. The study controls for bug-specific and developer-specific characteristics that may also affect developers' decisions exogenously. We also control for priority and severity levels assigned by the team to bugs in an attempt to manage voluntary contribution.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*Empirical, Open Source*

General Terms

Free/Libre Open Source, Bug-fixing, Self-managing Teams, Collaboration Networks, Stochastic Actor Oriented Models (SAOM), SIENA Models, Network Motives

1. INTRODUCTION AND MOTIVATION

FLOSS projects can be described as self-managing, virtual teams [2], i.e., organizations 'whose members are bound by a long-term common interest or goal, and who communicate and coordinate their work through information technology' [1, p. 742]. [1, 2] individuate as a key feature of all virtual organizations the reduced importance played by formal rules. Among the characteristics resulting from the marginal role that formal organizational structures assume in FLOSS projects there is also the voluntary nature of the decision to contribute that software developers take before joining them.

Indeed the FLOSS phenomenon represents an apparent paradox specifically because it requires the decision of several developers to invest their private resources in the development of a public good that cannot, by definition, be protected from free riding [7, 19, 20]. Successful FLOSS projects often cited as example of FLOSS organizational success, like the Linux kernel, the Apache web server or the Mozilla Firefox browser, are the ones that have been able to attract a relevant number of developers and to keep them active in contributing to the projects over time.

The crucial importance of attracting volunteers for the survival and success of FLOSS projects also stirred considerable academic attention on the motivations leading to contribution to all aspects of FLOSS projects, like, among others, contribution to codebase, bug-fixing, design, documentation, and testing. A large body of contributions has investigated both extrinsic and intrinsic motivations as the main explanations to the willingness to contribute [13, 6, 8]. [9] first listed two extrinsic motivations: the immediate benefit of 'scratching an itch' and the incentives for developers deriving from gaining reputation and signaling their abilities to potential future employers. Subsequent empirical analyses and surveys partially reduced the role played by these extrinsic motivations [6, 8]. A more complex picture has emerged in which also intrinsic motivations like the pure pleasure of programming, the sense of identification with a community, or the ideal that free software should be promoted *per se*, play an important role [5].

Nonetheless, all motivations investigated in the literature are individual in nature and tend to ignore the characteristics of the specific FLOSS projects to which developers consider whether to start or continue to contribute. More specifically, in the FLOSS literature the social networks formed by developers interacting while participating to a FLOSS project have been extensively studied as a crucial dimension for the survival and thrive of communities [4, 3, 11, 10] and the FLOSS phenomenon at large [12]. However, the direct effect of the local structures assumed by these social networks on the decision of potential members to collaborate in a project, to the best of our knowledge, has not yet been investigated.

Our study constitutes a first attempt in filling this gap. We concentrate on the decision to take part in the bug-fixing process, a required task for all FLOSS projects. In a context in which actions taken by members are mostly voluntary and always publicly visible we test whether the social network structures built around bugs by the developers already contributing to its fixing influence the decision of other developers to also take part to the process.

2. RESEARCH DESIGN

2.1 Data

In order to test whether local network structures in a bug-fixing co-collaboration network matter for the decision of developers to undertake action in a specific task, we chose as a case of study Epiphany, which is the default web browser of the GNOME graphical desktop environment. Epiphany has been active since November 2002 and counts 227 registered developers that have been working on its code over eight years. Whereas some of Epiphany's developers are paid contributors, Epiphany remains controlled by the community which develops it and therefore can be considered a relevant example of the self-managing FLOSS projects we are investigating. This project has a publicly accessible mailing list with 857 registered users and a dedicated section in the GNOME bug-tracking web server used by developers and external users to report and track bugs concerning the software and its documentation. Like all GNOME projects, the Epiphany community issues two stable releases per year, in March and in September. To control for these cyclicalities, we decided in our study to take into account an entire release cycle. Therefore, we considered all co-collaboration actions undertaken by developers in bug-fixing activities during the GNOME release cycle started in March 2006 and ended in September 2006. We collected all relevant data by parsing all the bug reports in GNOME Bugzilla repository relative to Epiphany. The data collection and storing was done using Bicho (v. 0.4 rev. 7198)[1], a software part of the FLOSSMetric project [14].

2.2 Variables and Model specification

Our aim is to model the influence that local network structures in a bug-fixing co-collaboration network have on developer's decisions to take further action in bug-fixing activities, while controlling for developer-related and bug-related attributes that might influence this decision. Thus, our dependent variable is a bipartite network that links developers to bugs which they contributed to fix. The modeling approach that we adopt requires dycotomic networks, therefore a tie between developer i and bug j signifies that developer i performed one or more actions in order to fix bug j. If two developers work on the same bug, they are collaborating to its solution and therefore multiple shared bugs form a relational structure of co-collaboration. All possible signals of an action leading to the bug-fixing were coded: changing their status or the severity levels, or marking them as duplicates, or submitting a patch for review. Given that actions can only be started by developers, directionality of ties in the generated network is ignored and ties are assumed by definition to go from developers to bugs when interpreting our results.

Independent variables can be distinguished into structural covariates, which capture the influence of specific local network structures of co-collaboration on the probability of creating a new tie and actor-related covariates, which capture the influence of individual characteristics of both developers and bugs on such probability. Since our theoretical focus is on local network structures as determinants of the individual decision to collaborate, actor-related attributes act here as controls for individual characteristics that may also affect the pattern of collaboration among developers, as already in the literature here briefly summarized. *Developer tenure* and *Bug tenure* are dummies that, respectively, reflect whether a developer was already collaborating in bug-fixing in previous release cycles and whether a bug was already existing in previous release cycles. *Bug communication* is a dummy indicating whether a bug raised an exceptional amount of comments posted on a bug report during the release cycle. *Bug cc'ing* is a dummy indicating whether a bug gathered an exceptional amount of followers during the release cycle. Finally, *Bug severity* is a dummy indicating whether to a bug was assigned a level of severity higher then 'normal' in the GNOME Bugzilla seven-step scale for bug severity[2].

The statistics representing our independent variables are idiosyncratic to the class of stochastic models for network dynamics we adopted in order to investigate our conjecture. Therefore we firstly introduce the model and after describe the independent variables at the core of our empirical test.

What justifies the meaningful conceptualization of a set of social interactions as a 'social network' is the assumption of dependence among individual dyads, in our case formed by a developer and a bug to which he or she is linked, both across network space and over time. This micro-relational structure of linked local network interactions cannot be captured by statistical models that assume independence among dyads. This argument provides a compelling reason to adopt Stochastic Actor Oriented Models (SOAM) for Network Dynamics (also known as 'SIENA models' from the name of the software package used to estimate them) that have been conceptualized and developed for the analysis of longitudinal network data [16, 17, 15].

Based on Markov Chain Monte Carlo (MCMC) simulations, these models model the change of one tie variable by one actor at a time (a so-called network micro-step) by specifying a multinomial logit distribution that maximize a random utility function (the so-called evaluation function) that describes actors satisfaction with their local network neighborhood configurations. Through this statistical apparatus SAOM are therefore able, starting from the observed bug-fixing co-collaboration network at several points in time, to estimate the probability with which a developer will create a new tie by acting on a bug given the local network configurations in which both the developer and the bug are already embedded. To capture meaningful local network structures potentially influencing the decision to act on a bug, following [15], we define several endogenous 2-mode network effects that will serve as structural covariates in our model.

Specifically we include *Density*, which captures the effect of the overall network density on the probability of creating a new tie, i.e., acting on a bug. *Transitivity*, which captures the extent to which two developers sharing a tie towards the same bug are more likely to engage in collaborative activities towards other bugs. *Developer degree activity*, which captures the tendencies in dispersion of the level of activity across developers. *Bug degree popularity*, which captures the extent to which bugs on which already many developers are acting are more likely to attract further ties. *Bug degree popularity* is the main variable of interest for us, whose significance we aim to test in order to investigate the influence of the local network structure surrounding a bug on the developers' decisions to take action. Finally *Developer-Bug degree assortativity*, which captures the tendency whereby

[1] `http://tools.libresoft.es/bicho`

[2] All the generated matrices, variables, and code necessary to reproduce the analyses are available from the authors upon request.

highly active developers direct their actions relatively more towards bugs with higher in-degree.

In SAOM for bipartite network, independent actor covariates enter the model specification in two ways, strictly determined by which mode an actor belongs to. The first is as a sender (ego) effect: actors with a higher (or lower) level of a specific attribute may tend to express significantly more (or less) network relations. The second is as a receiver (alter) effect: actors with a higher (or lower) level of a specific attribute may be the target of significantly more (or less) network relations. Hence, in our model *Developer tenure* is specified as ego effect while *Bug tenure*, *Bug communication*, *Bug cc'ing* and *Bug severity* as alter effects.

SOAM use longitudinal network data and therefore the six-month time of the release cycle of interest was sliced up in five two-month networks with a moving window of one month. These networks, taken in couples, represent the endpoints of four transition periods in the estimation. (See Figures 1-2 as examples of the measured co-collaboration network respectively at end of the second and forth period. Developers are depicted in red and bugs in blue. A tie is drawn if a developer took action on a bug during that specific 2-month period). All estimations were conducted using unconditional method of moments which allows estimating not only the evaluation function parameters previously introduced, but also parameters of the rate function, which models the speed by which each developer gets an opportunity to decide whether to take action on a bug given the local network configuration in which he or she is immersed. The analysis was performed using the R package RSiena[3] (v. 1.0.10 rev. 80).

3. RESULTS AND DISCUSSION

The final results of estimation are presented in Table 1. β coefficients represent the estimated parameters associated to each structural and actor-related covariates. Standard errors are also given for all the parameters. The significance of all the parameters, also reported in Table 1, can be assessed through regular t-statistics, defined as estimate divided by its standard error (Not to be confounded with convergence t-ratios explained below). The rate parameters are all largely positive, indicating that actors on average underwent a large number of micro-steps to produce a global network configuration that resembles the observed network. Convergence statistics (convergence t-ratios) are all smaller than 0.1 in absolute value, indicating that the deviation between the simulated statistics and their observed values is significantly close to zero, thus the model can be safely considered convergent [15].

Regarding the effects of the evaluation function, the parameter for density is negative and significant, which is always the case with statistical modeling of social networks, since they tend to be sparse by nature [18]. To capture the large skewness in the degree distribution of the developers we inserted in the model the inverse of the *Developer degree activity* effect, which actually turned out to be strongly positive and significant. A further endogenous control aimed at capturing skewness in the degree distribution is the *Developer-Bug degree assortativity* effect: its parameter is positive and significant, however small, indicating a slight tendency for most active developers to contribute to most popular bugs.

The *Bug degree popularity* and the *Transitivity* parameters are the results of main interest here. The negative and statistically significant coefficient of *Bug degree popularity* implies that the probability of a bug to attract a new tie decreases as the number of ties already present increases. The *Transitivity* parameter, which is also negative and significant, indicates that two developers who share connections to

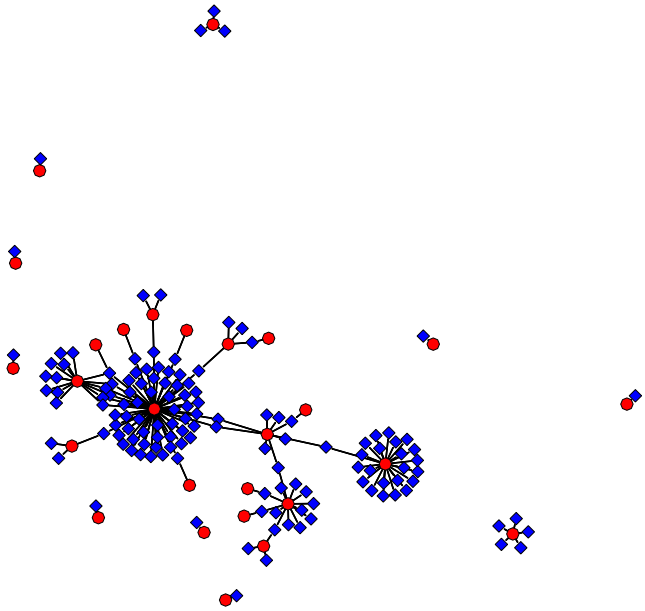

Figure 1: Co-collaboration network (end period 1)

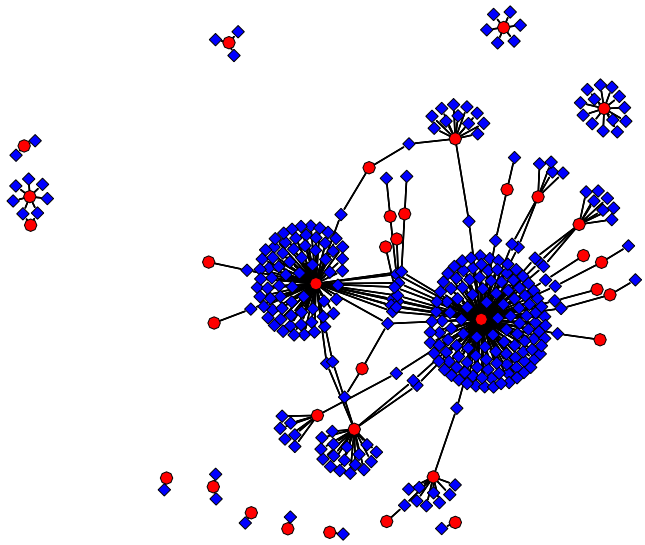

Figure 2: Co-collaboration network (end period 4)

[3] `http://r-forge.r-project.org/projects/rsiena`

Table 1: SAOM estimation results

	β	s.e.	conv. t-ratio
Rate parameter period 1	43.746	8.979	
Rate parameter period 2	54.277	10.859	
Rate parameter period 3	47.092	8.668	
Rate parameter period 4	78.235	13.665	
Structural covariates:			
Outdegree	-1.016***	0.332	-0.039
Transitivity	-0.355***	0.077	-0.008
Developer degree activity	-1.252***	0.303	0.061
Bug degree popularity	17.991***	1.786	0.001
Developer-Bug degree asrt.	0.018***	0.002	0.002
Actor-related covariates:			
Bug tenure	0.002	0.072	-0.029
Dev tenure	0.088	0.063	-0.001
Bug cc'ing	0.063	0.084	-0.004
Bug communication	1.139***	0.076	-0.014
Bug severity	0.038	0.062	0.014

$^{***}p < 0.001$.

the same bug tend not to engage in further co-collaboration patterns. This confirms that whether a developer decides to take action on a bug also depends directly on the local network structures surrounding such bug. In our context this translates into a scenario where developers tend to take action on bugs which are not already popular, possibly enacting an organizational coordination mechanism purely fueled by the configuration of the endogenous co-collaboration structure.

Finally, actor-related covariates entered the model to control for individual tendencies that are known also to affect the decision of developers to participate. Out of the five parameters, only *Bug communication* has a statistically significant impact: the more a bug has raised communication flows around it, the more likely developers will try to engage in fixing activities on it. Interestingly, and contrary to what we expected, high bug severity levels seem not to significantly increase the chances of developers to take action.

4. CONCLUSIONS

In this paper we aimed at studying the influence of local network structures on individual decisions to collaborate to a specific task. We collected data on the co-collaboration networks of developers involved in bug-fixing activities related to a FLOSS project during one of its release cycles. In addition, we gathered data on specific attributes related to both developers and bugs in order to control for individual motivations to engage in collaborative actions. We specified our model using the class of SAOM, specifically suited for estimating the effect of local network structures on the probability of creating new ties.

Our main result confirms our conjecture: in the context of FLOSS projects, whether a member decides to take action in order to fix a bug does also depend on the local structure of the co-collaboration network already surrounding that bug. In particular, in our context FLOSS developers tend to direct their activity relatively more towards bugs which have been received less activity by other developers. This leads in our case study to a virtuous circle of co-ordination over time generated purely by the co-collaboration structure of the FLOSS project.

Finally, we acknowledge several limitations in our study. Firstly, our conjecture has been tested only on one release cycle of one particular FLOSS project; secondly, it has not yet thoroughly tested to which extent the actor-related variables included in the model do control for the individual motivations already known in the literature to influence participation patterns of FLOSS developers. Therefore, further studies have to be carried out in order to investigate the scope conditions of our research and test the external validity of our control measures.

5. REFERENCES

[1] M. K. Ahuja and K. M. Carley. Network Structure in Virtual Organizations. *Organization Science*, 10(6):741–757, November 1999.

[2] M. K. Ahuja, D. F. Galletta, and K. M. Carley. Individual Centrality and Performance in Virtual R&D Groups: An Empirical Study. *Management Science*, 49(1):21–38, January 2003.

[3] C. Bird, A. Gourley, P. Devanbu, A. Swaminathan, and G. Hsu. Open Borders? Immigration in Open Source Projects. *Proceedings of the Fourth International Workshop on Mining Software Repositories*, 2007.

[4] K. Crowston and J. Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.

[5] E. Franck and C. Jungwirth. Reconciling Rent-Seekers and Donators–The Governance Structure of Open Source. *Journal of Management and Governance*, 7(4):401–421, 2003.

[6] A. Hars and S. Ou. Working for Free? Motivations for Participating in Open-Source Projects. *International Journal of Electronic Commerce*, 6(3):25–40, 2002.

[7] B. Kogut and A. Metiu. Open-Source Software Development and Distributed Innovation. *Oxford Review of Economic Policy*, 17(2):248–264, June 2001.

[8] K. Lakhani and R. Wolf. *Why hackers do what they do: Understanding Motivation Effort in Free/Open Source Projects*, pages 3–21. MIT Press, Cambridge, 2005.

[9] J. Lerner and J. Tirole. Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2):197 – 234, 2002.

[10] Y. Long and K. Siau. Social Network Structures in Open Source Software Development Teams. *Journal of Database Management*, 18(2):25–40, 2007.

[11] L. López-Fernández, G. Robles, J. M. González-Barahona, and I. Herraiz. Applying Social Network Analysis Techniques to Community-driven Libre Software Projects. *International Journal of Information Technology and Web Engineering*, 1(3):27–48, 2006.

[12] G. Madey, V. Freeh, and R. Tynan. *Modeling the free/open source software community: A quantitative investigation*, pages 203–220. Idea Group Publishing, 2005.

[13] M. Osterloh, S. Rota, and M. Von Wartburg. Open Source - New Rules in Software Development.

Working paper, pages 1–23, 2009.

[14] G. Robles, J. M. González-Barahona, D. Izquierdo-Cortazar, and I. Herraiz. Tools for the Study of the Usual Data Sources found in Libre Software Projects. *International Journal of Open Source Software & Processes*, 1(1):24–45, 2009.

[15] T. A. Snijders, G. G. van De Bunt, and C. E. Steglich. Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60, 2010.

[16] T. A. B. Snijders. Stochastic actor-oriented dynamic network analysis. *Journal of Mathematical Sociology*, 21:149–172, 1996.

[17] T. A. B. Snijders. The Statistical Evaluation of Social Network Dynamics. *Sociological Methodology*, 31:361–395, 2001.

[18] C. Steglich, T. A. B. Snijders, and P. West. Applying SIENA - An Illustrative Analysis of the Coevolution of Adolescents Friendship Networks, Taste in Music, and Alcohol Consumption. *Methodology*, 2(1):48–56, 2006.

[19] E. von Hippel and G. von Krogh. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, 14(2):209–223, 2003.

[20] G. von Krogh and E. von Hippel. The Promise of Research on Open Source Software. *Management Science*, 52(7):975–983, 2006.