A Quantitative Study of Social Organisation in Open Source Software Communities*

Marcelo Serrano Zanetti, Emre Sarigöl, Ingo Scholtes, Claudio Juan Tessone, and Frank Schweitzer

Chair of Systems Design, ETH Zurich, Switzerland
{mzanetti,semre,ischoltes,tessonecfschweitzer}@ethz.ch

Abstract

The success of open source projects crucially depends on the voluntary contributions of a sufficiently large community of users. Apart from the mere size of the community, interesting questions arise when looking at the *evolution of structural features* of collaborations between community members. In this article, we discuss several network analytic proxies that can be used to quantify different aspects of the social organisation in social collaboration networks. We particularly focus on measures that can be related to the cohesiveness of the communities, the distribution of responsibilities and the resilience against turnover of community members. We present a comparative analysis on a large-scale dataset that covers the full history of collaborations between users of 14 major open source software communities. Our analysis covers both aggregate and time-evolving measures and highlights differences in the social organisation across communities. We argue that our results are a promising step towards the definition of suitable, potentially multi-dimensional, resilience and risk indicators for open source software communities.

1998 ACM Subject Classification D.2.8 Metrics, K.4.3 Organisational Impacts

Keywords and phrases open source software, mining software repositories, social networks

Digital Object Identifier 10.4230/OASIcs.ICCSW.2012.116

1 Introduction

What are the most important social factors that lead to successful and sustainable open source software projects? According to *Linus' Law* - which states that "given enough eyeballs, all bugs are shallow" [7] - the quality and success of open source software (OSS) critically depends on the existence of a sufficiently large community of developers who review, modify and improve the publicly available source code. Apart from development efforts, another important success factor is the existence of a stable community of users who report software defects, request and inspire new features, reproduce bugs or comment on issues reported by other users. By employing the collective knowledge and diverse experiences of many contributors, most OSS communities manage to provide technical assistance to less experienced users, often on a time scale that is competitive to commercial software support.

Depending on the distribution of competencies and responsibilities of contributors, largely different patterns of collaborations may arise. While it is generally difficult to assess these social factors of OSS projects, the availability of large scale data on community dynamics increasingly allows to study the *social dimension of OSS projects* from a quantitative perspective [8, 16]. Previous studies have mainly focused on rather simple proxies of social

* This work was supported by the SNF through grant CR12I1_125298. C.J Tessone acknowledges financial support from the SBF through grant C09.0055.

2012 Imperial College Computing Student Workshop (ICCSW'12).
Editor: Andrew V. Jones; pp. 116–122
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

dynamics like the evolution of the number of contributors and contributions or the time span of a user's activity and were mostly based on a rather limited set of snapshots of a single project. Using a large scale dataset of time-stamped social interactions that has been collected from the Bugzilla bug-tracker installations of 14 major OSS projects, in this paper we study the *fine-grained evolution of structural features of networks of user collaborations*. We thus take a *network perspective on OSS communities* and highlight differences in the social organisation of software projects that can be related to their activity, their cohesion as well as their resilience against fluctuations in the community members. By applying standard measures from social network analysis we particularly quantify how tightly community members collaborate, how equal responsibilities are distributed and how resilient collaboration topologies are against the loss of (central) community members. While similar tools have been applied to OSS projects before [3, 6], to the best of our knowledge, the present paper is the first to study these network analytic measures on a dataset that covers the full, fine-grained history of 14 well-established and successful OSS communities.

2 Social Organisation in OSS Communities: A Network Perspective

In order to make substantiated statements about the structure and dynamics of the social organisation of OSS communities, we recently completed collecting data on the history of user collaborations recorded by the Bugzilla installation of 14 well-established OSS projects. Bugzilla[9] is an open source bug tracking system which is utilised by users and developers alike to report bugs, keep track of open issues and feature requests and comment on issues reported by others. Since the Bugzilla installations of OSS projects are used to foster collaboration between community members, it constitutes a valuable source of data that allows us to track social interactions between developers and users.

2.1 Building Social Networks from Bug-Reports

Data in the Bugzilla database are arranged around the notion of *bug reports*. Each bug report has a set of fields describing aspects like the user who initially filed the bug report, its current status (e.g. *pending*, *reproduced*, *solved*, etc), to whom the responsibility to provide a fix has been assigned, attachments which may be used to reproduce or resolve the issue, comments and hints by other community members, or a list of community members which shall be informed about future updates. Apart from an initial bug report, Bugzilla additionally stores the full history of all updates to any of the fields of a bug report. Each of these change records includes a time stamp, the ID of the user performing the change as well as the new values of the changed fields. While our dataset comprises change records for all possible fields, in this article we focus on those that indicate changes in the users that are assigned responsibility to fix an issue (henceforth called the *ASSIGNEE* field) and changes to the list of users to whom future updates of the bug shall be sent via E-Mail (henceforth called the *CC* field). We consider any updates in the *CC* and *ASSIGNEE* field of a bug report as a time-stamped edge from the user who performed the update to the user(s) who were added to the *CC* field or the *ASSIGNEE* list of responsible developers respectively.

Based on the data extraction procedure described above, we obtain a large time-aggregated network of nodes representing community members and time-stamped edges representing a particular interaction between two users. For most of the projects considered, the Bugzilla history from which we extract the network is longer than ten years. The fact that - in social networks aggregated over such long periods of time - most of the users represented by nodes have never been active within the same time period limits the expressiveness of the network structure in terms of a project's "social organisation". In order to overcome this issue, we perform a *dynamic network analysis* by defining a sequence of *monthly collaboration networks* based on the time stamps of edges. In particular, we define a 30 day sliding time window and filter out those edges whose time stamps are outside the window and those nodes who did not have any interactions in the corresponding time period. By progressively advancing the start date of the sliding 30 day time window by one day increments we obtain a sequence of collaboration networks that allows us to study the structure of the community's social organisation as well as its evolution over time. Naturally, most of the monthly networks obtained in the way described above will not be fully connected. Since the network analytic measures we intend to apply assume connected topologies, we perform a component analysis on all snapshots and restrict our quantitative analysis to the largest connected component (LCC). In order to test the significance of our findings we further compute the fraction of those nodes who are part of the largest connected component. Table 1 shows the 14 OSS projects that are included in our dataset along with the time period and the total number of bug reports and updates that we included in our analysis. The column *LCC/TOTAL* furthermore indicates the fraction of users in the LCC, averaged over all monthly snapshots of the corresponding project. Here one observes that our data shows a rather large degree of variation with respect to this fraction, which may be seen as an argument that this measure is an interesting indicator for the *cohesiveness* of OSS communities by itself. Nevertheless, we argue that for all projects the fraction of users in the LCC is sufficiently large to make substantiated statements about the project's social organisation.

Table 1 Aggregated measures for the studied projects. From column *LCC/Total* to the last on the right, the numbers indicate the mean value ± standard deviation.

Project Name	Bugs	Updates	Period	LCC/Total	Nodes in LCC	Edges	Mean Degree	Assortativity	Closeness Central.	Clustering Coefficient
XAMARIN	4552	20721	2011-2012	0.93±0.05	46.76±8.12	98.15±22.70	2.07±0.29	-0.14±0.11	0.40±0.07	0.22±0.05
THUNDERBIRD	35388	313957	2000-2012	0.53±0.26	64.82±53.49	86.44±80.05	1.05±0.42	-0.23±0.17	0.40±0.27	0.04±0.05
LIBREOFFICE	8916	78341	2010-2012	0.78±0.11	73.83±32.06	114.41±49.10	1.56±0.26	-0.20±0.10	0.40±0.09	0.13±0.06
MAGEIA	6600	46921	2006-2012	0.93±0.07	77.54±21.80	156.00±59.24	1.95±0.30	-0.37±0.12	0.54±0.09	0.14±0.04
MANDRIVA	60546	368463	2002-2012	0.70±0.18	88.15±60.70	142.16±118.44	1.41±0.38	-0.29±0.15	0.40±0.14	0.07±0.05
FIREFOX	112953	1067914	1999-2012	0.58±0.23	171.77±117.79	240.79±180.44	1.16±0.44	-0.15±0.11	0.32±0.23	0.04±0.04
SEAMONKEY	90040	993392	1998-2012	0.67±0.15	210.39±251.43	364.42±482.54	1.48±0.48	-0.19±0.13	0.34±0.11	0.08±0.06
NETBEANS	210921	1875878	2000-2012	0.96±0.05	269.71±292.07	1069.72±1509.12	3.39±1.13	-0.12±0.08	0.37±0.05	0.23±0.08
OPENOFFICE	118135	915749	2000-2012	0.88±0.19	319.01±169.88	931.35±591.80	2.52±0.84	-0.12±0.10	0.34±0.15	0.12±0.06
GENTOO	140216	661783	2002-2012	0.80±0.07	338.97±110.86	617.73±211.92	1.82±0.27	-0.29±0.10	0.49±0.13	0.04±0.03
KDE	179470	648331	2002-2012	0.75±0.12	361.16±246.16	424.61±301.20	1.15±0.07	-0.16±0.07	0.32±0.07	0.01±0.01
ECLIPSE	356415	2594385	2001-2012	0.78±0.08	472.58±180.71	964.47±411.94	2.06±0.38	0.05±0.08	0.25±0.05	0.13±0.03
GNOME	550722	2751441	2000-2012	0.67±0.12	523.76±585.26	610.16±616.81	1.25±0.22	-0.17±0.09	0.25±0.08	0.03±0.04
REDHAT	414163	3777634	2006-2012	0.45±0.26	658.06±865.97	983.58±1297.18	1.19±0.35	-0.12±0.20	0.30±0.23	0.00±0.01

2.2 Network Measures

While the literature is rich in terms of measures able to quantify structural features of networks [11, 5], due to space limitations here we focus on three measures which are able to capture basic network qualities that relate to the *cohesiveness* of a community, the distribution of responsibilities among its members and its resilience against fluctuations in the user base. The first network measure is based on the *closeness centrality* of a node, which is defined as the inverse of the sum of the shortest path length to all other nodes in the network.

$$Cc(n_i) = \sum_{j=1,j\neq i}^{N} \frac{N-1}{d(n_i,n_j)} \in [0,1] \tag{1}$$

where $Cc(n_i)$ corresponds to the *closeness centrality* score of node n_i, $d(n_i, n_j)$ is the length of the shortest path between nodes n_i and n_j, while N corresponds to the total number of nodes in a given network. Finally, the factor $N-1$ is a normalisation constant [2]. Based on this, the *closeness centralisation* of a network (Cc_{global}) can be calculated by taking the sum of the differences between the node with the highest value of closeness centrality (n^*) and the closeness centrality scores of all other nodes. This quantity is then normalised to the range of 0 to 1 using the theoretical value that results from a (maximally centralised) star network. Equation (2) presents the formal definition, while more details can be found in [2, 11]. In the context of OSS collaboration networks, closeness centralisation captures to what degree responsibilities, collaboration and communication are distributed equally across community members.

$$Cc_{global} = \sum_{i=1}^{N} \frac{Cc(n^*) - Cc(n_i)}{\frac{(N-2)(N-1)}{2N-3}} \in [0,1] \tag{2}$$

The second measure, the *clustering coefficient* of a network (C), measures how closely community members interact with each other in the sense that an interaction between a user X and Y, as well as an interaction between user Y and Z will also entail a direct interaction between the users Y and Z. The formal definition is presented in equations (3) and (4).

$$C(n_i) = \frac{2L_{D_{n_i}}}{D_{n_i}(D_{n_i}-1)} \in [0,1] \tag{3}$$

$$C = \frac{1}{N}\sum_{i=1}^{N} C(n_i) \in [0,1] \tag{4}$$

where D_{n_i} is the number of nodes directly connected to the node n_i, while $L_{D_{n_i}}$ is the number of edges between them. Therefore, the clustering coefficient $C(n_i)$ of node n_i expresses the fraction of edges that were realised from the possible $\frac{D_{n_i}(D_{n_i}-1)}{2}$ edges which are expected in a fully connected network with D_{n_i} nodes. We obtain the clustering coefficient of a network by averaging the clustering coefficient scores of all existing nodes (see equation (4)). This procedure can be seen as measuring how *cohesive* the community is in terms of nodes being embedded in collaborating clusters [11].

Finally, the *assortativity* (r) measures an individual's preference to connect to other individuals that have a similar or different degree of connectivity (the degree being a node's number of connections to different nodes). Networks in which nodes are preferentially connected to nodes with similar degree are called assortative. In this case a positive degree assortativity ($0 \ll r \leq 1$) indicates a positive correlation between the degrees of neighbouring nodes. Networks in which nodes are preferentially connected to nodes with different degree are called disassortative and in this case degree assortativity is negative ($0 \gg r \geq -1$). In networks with zero degree assortativity, there is no correlation between the degrees of connected nodes, i.e. nodes do not exhibit a preference for one or the other. Formally,

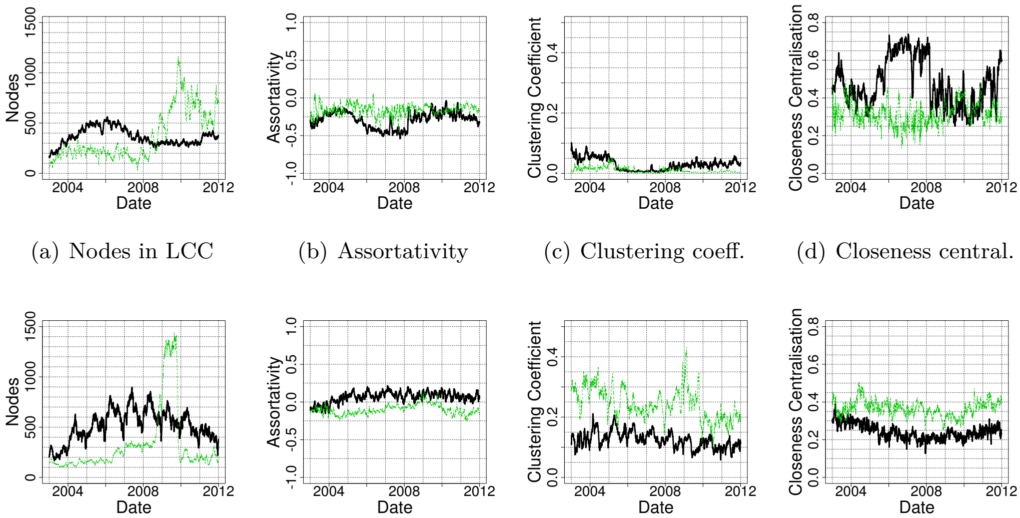
$$r = \frac{\sum_{ij} ij(e_{i,j} - q(i)q(j))}{\sigma(q)^2} \in [-1,1] \tag{5}$$

where e_{ij} is the fraction of all links in the network that join together nodes with degrees i and j, $q(i) = \sum_j e_{i,j}$, $q(j) = \sum_i e_{i,j}$ and $\sigma(q)$ is the standard deviation of the distribution of q. The term $q(i)q(j)$ is the equivalent to the expected value of $e_{i,j}$ inferred from a random network. Therefore, if $r = 0$ the pattern of interconnection between nodes is also random [4].

3 Comparative Analysis of OSS Communities

As described above, the preliminary results presented here have been obtained for the LCC of the network of monthly collaborations in terms of *CC* and *ASSIGNEE* interactions. While Table 1 shows the aggregate measures averaged over all time windows for every project in our database, due to space constraints we limit the presentation of the dynamics of the social organisation to the projects Gentoo and KDE (both Gnu/Linux related projects) as well as Eclipse and NetBeans (both Java IDEs). These have been chosen because a) their communities are of comparable size and age, b) the respective pairs of projects address similar problem domains and c) they represent contrasting examples with respect to the measures studied in this paper.

Figure 1 shows the evolution of the number of nodes in the LCC, its assortativity, clustering coefficient and closeness centralisation for these four projects. For all projects, the fraction of nodes in the LCC is rather stable with values between 0.7 and 1 consistent with the aggregate values given in Table 1. The same is true for the evolution of the mean degree. We thus omit these plots. The four projects show significant differences in the evolution of the clustering coefficient that cannot be explained by mere size effects. In the particular time frame between 2006 and 2008, the clustering coefficient of the Eclipse community (≈ 0.15) was roughly ten times higher than that of the Gentoo community (≈ 0.01), although the LCCs of both communities were of comparable size (≈ 500 nodes). In addition, the clustering coefficient of the Gentoo community shows an interesting dynamics, dropping to a very small value between 2006 and 2008 and increasing thereafter.

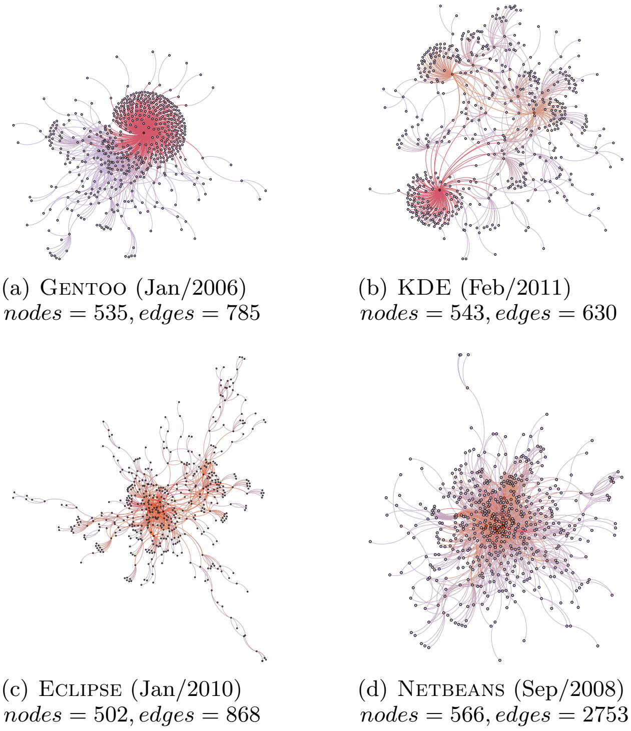


(a) Nodes in LCC (b) Assortativity (c) Clustering coeff. (d) Closeness central.

(e) Nodes in LCC (f) Assortativity (g) Clustering coeff. (h) Closeness central.

Figure 1 Evolution of structural measures of the LCC in the monthly Bugzilla collaboration networks. (a-d): Gnu/Linux related projects Gentoo (black) and KDE (green), (e-h): IDEs Eclipse (black) and NetBeans (green).

A different perspective of the structural change the Gentoo community was undergoing is given in Figure 1(d) which displays a visible plateau in the closeness centralisation of the network within the same period. In fact, as can be seen in the network depicted in Figure 2(a), in the period between 2006 and 2008 most of the collaborations were mediated by a



(a) Gentoo (Jan/2006)
$nodes = 535, edges = 785$

(b) KDE (Feb/2011)
$nodes = 543, edges = 630$

(c) Eclipse (Jan/2010)
$nodes = 502, edges = 868$

(d) Netbeans (Sep/2008)
$nodes = 566, edges = 2753$

Figure 2 Four monthly collaboration networks with comparable size showing largely different social organisation (the network visualisation was generated by Gephi [1]).

single central community member, while the social organisation of the Eclipse community depicted in 2(c) was structured in a much more homogeneous way. The evolution of degree assortativity is captured in Figures 1(b) and 1(f). Both the level of degree assortativity as well as its dynamics differ across the projects. The collaboration network of Eclipse exhibits a tendency towards assortative structures (meaning that high degree nodes are preferentially connected to high degree nodes). The opposite is true for the KDE and the Gentoo communities which show a tendency towards disassortativity. We thus argue that assortativity is suitable to further differentiate the social organisation of OSS communities.

4 Conclusions and Future Work

We have studied measures that capture different structural dimensions in the social organisation of OSS projects. Our analysis is based on a comprehensive dataset collected from the bug tracking communities of 14 major OSS projects. We view the social organisation from the perspective of time-evolving networks and highlight how projects, although similar in terms of size, problem domain and age, a) largely differ in terms of clustering coefficient, assortativity and closeness centralisation and b) that some projects show interesting dynamics with respect to these measures that cannot be explained by mere size effects. We argue that the phase of high closeness centralisation and low clustering coefficient observed in the Gentoo community between 2006 and 2008 may be interpreted as a lack of *social cohesion* which can possibly pose a risk for the project.

While our results are necessarily preliminary, we currently extend our work by adding spectral measures like algebraic connectivity and inequality measures like the Gini index that can highlight further differences in the social organisation [13]. A detailed case study is under preparation [14] and further includes community performance indicators (e.g. response times, bug fixing times and fraction of open issues) that can be mined from our dataset. The eventual goal of our project is the provision of multi-dimensional indicators for the social and technical organisation of OSS projects that are correlated with performance and that can be considered in the management and evaluation of OSS projects [12, 15, 10]. Such indicators can be useful when taking informed decisions about which OSS project to invest in or rely on. Furthermore, due to the distributed nature of collaborations, individuals often lack a global perspective on evolving communication and coordination structures, even though these can influence long-term success. An inclusion of suitable indicators in community platforms like e.g. Bugzilla can assist in determining risks and allow project managers to timely react by shifting responsibilities, fostering information flow or changing organisational procedures.

References

1 M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the ICWSM '09*. AAAI, 2009.

2 Linton C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979.

3 James Howison, Keisuke Inoue, and Kevin Crowston. Social dynamics of free and open source team communications. *Open Source Systems*, pages 319–330, 2006.

4 Mark E. J. Newman. Mixing patterns in networks. *Phy. Review E*, 67:026126, 2003.

5 Mark E. J. Newman. *Networks: an introduction.* Oxford Univ Press, 2010.

6 Roozbeh Nia, Christian Bird, Premkumar Devanbu, and Vladimir Filkov. Validity of network analyses in open source projects. In *proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR)*, pages 201–209. IEEE, 2010.

7 Eric S. Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

8 Gregorio Robles and Jesus Gonzalez-Barahona. Contributor turnover in libre software projects. In Ernesto Damiani and et al, editors, *Open Source Systems*, volume 203, pages 273–286. Springer Boston, 2006.

9 Nicolas. Serrano and Ismael Ciordia. Bugzilla, itracker, and other bug trackers. *Software, IEEE*, 22(2):11–13, 2005.

10 Claudio Juan Tessone, Markus Michael Geipel, and Frank Schweitzer. Sustainable growth in complex networks. *Europhysics Letters*, 96:58005, 2011.

11 Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

12 Marcelo Serrano Zanetti. The co-evolution of socio-technical structures in sustainable software development: Lessons from the open source software communities. In *proceedings of the 34th ICSE, doctoral symposium*, pages 1587–1590, 2012.

13 Marcelo Serrano Zanetti, Ingo Scholtes, Claudio Juan Tessone, and Frank Schweitzer. The evolution of social organisation and risk in open source software projects. In preparation.

14 Marcelo Serrano Zanetti, Ingo Scholtes, Claudio Juan Tessone, and Frank Schweitzer. A quantitative study of social organisation in the gentoo community. In preparation.

15 Marcelo Serrano Zanetti and Frank Schweitzer. A network perspective on software modularity. In *GI-Edition - Lecture Notes in Informatics (LNI), Proceedings P-200, ARCS 2012 Workshops*, pages 175–186, 2012.

16 Minghui Zhou and Audris Mockus. What make long term contributors: Willingness and opportunity in oss community. In *proceedings of the 34th ICSE*, pages 518–528, 2012.