The Rise and Fall of a Central Contributor: Dynamics of Social Organization and Performance in the GENTOO Community

Marcelo Serrano Zanetti, Ingo Scholtes, Claudio Juan Tessone and Frank Schweitzer
Chair of Systems Design, ETH Zurich, CH-8092 Zurich, Switzerland
http://www.sg.ethz.ch/

Abstract—Social organization and division of labor crucially influence the performance of collaborative software engineering efforts. In this paper, we provide a quantitative analysis of the relation between social organization and performance in GENTOO, an Open Source community developing a LINUX distribution. We study the structure and dynamics of collaborations as recorded in the project's bug tracking system over a period of ten years. We identify a period of increasing centralization after which most interactions in the community were mediated by a single central contributor. In this period of maximum centralization, the central contributor unexpectedly left the project, thus posing a significant challenge for the community. We quantify how the rise, the activity as well as the subsequent sudden dropout of this central contributor affected both the social organization and the bug handling performance of the GENTOO community. We analyze social organization from the perspective of network theory and augment our quantitative findings by interviews with prominent members of the GENTOO community which shared their personal insights.

I. INTRODUCTION

An important prerequisite for the success of Open Source Software (OSS) projects is the ability to build a sufficiently large and stable community of users and contributors. While actual source code is typically contributed by a rather small and stable set of core developers, the wider - and possibly more diverse - community plays an important role in processes related to software quality management. Here, most OSS projects rely on a large number of part-time *contributors* who report bugs, triage pending bug reports or provide support and solutions for issues reported by others. This community effort in handling bug reports does not only unburden developers; it also significantly increases software quality, thus bearing the potential to attract more users. Furthermore, in [1] it was argued that - despite their volatility - bug handling communities are a common entry point for long-term contributors who - after getting insight into a project's organizational and technical structures - may eventually become members of the core developer team. As such, the structure and dynamics of bug handling communities is of particular importance for the success of OSS projects. In order to ensure a timely response to bug reports, the management of the project has to find efficient organizational structures and a reasonable division of labor, despite the fact that these communities are typically highly heterogeneous in terms of dedication and skills.

In this paper, we present a quantitative analysis of the structure and dynamics of the bug handling community of GENTOO, an OSS project developing a LINUX distribution. Our study is based on a data set covering more than $150,000$ collaboration events recorded by the project's BUGZILLA installation over a period of more than ten years. The contributions of our study are as follows:

- We study collaboration structures of the GENTOO bug handling community by applying quantitative measures that capture cohesion, centralization, clustering and communication efficiency. Our analysis reveals a period of increasing centralization and decreasing cohesion that resulted in a situation where most interactions in the community were mediated by a single *central contributor*.
- In the period of maximum centralization the central contributor unexpectedly left the project. We analyze the implications for the project's social organization, which include a temporary loss of cohesion as well as subsequent efforts to reorganize the community.
- We complement our study by an analysis of the community's performance in terms of bug handling efficiency and response time. Our findings suggest that the performance improved during the active period of the *central contributor*, while her retirement had a lasting negative effect on bug handling efficiency and response time.
- We substantiate our quantitative findings by personal insights into the social dynamics of the GENTOO community provided by three long-term contributors. These insights support our findings and highlight potential applications of our quantitative measures in the monitoring of collaboration structures in OSS projects.

The remainder of this paper is organized as follows. In section II we summarize relevant related work studying the structure of OSS communities and its impact on performance. In section III we introduce data collection and network analysis methods that form the basis of our case study. In section IV we present quantitative results on the evolution of the social organization, as well as bug handling performance in the GENTOO community. We further interpret our findings, align them with personal insights shared by prominent community members and discuss threats to validity. Finally, in section V we summarize our contributions and highlight future research on the application of network-based analysis methods in the management of software development communities.

II. STUDIES OF SOCIAL ORGANIZATION IN COLLABORATIVE SOFTWARE ENGINEERING

The question how the structure and dynamics of social organization influences the performance and success of collaborative software development efforts has been studied by researchers from different fields using a variety of methods. Due to the availability of data, many of these studies address

OSS communities, which consist of *users*, *developers* and other *contributors*, who contribute to the project in terms of documentation, maintenance of web sites or the submission and handling of bug reports. Members of such communities typically need to self-organize in a way that guarantees information flow as well as a coordinated allocation of tasks and responsibilities. The processes and structures of this self-organization process have been studied in a number of works.

Since it plays a central role in software quality assurance, *bug handling communities* have been the subject of many studies. Compared to the development of source code, in [2] it was found that the bug handling process is based on the contributions of a much wider community. In a recent work presented in [1], this community has further been shown to be an important entry point for long-term contributors and developers. As an important finding, lack of attention paid to bug reporters and fast negative feedback by the community decreases the likelihood for such users to contribute to the project for a long period. This is partly in line with arguments about the negative impact of a too strict duplicate bug policy in bug handling communities put forth in [3].

The collaboration structures emerging in bug handling communities can be extracted by different means. Communication topologies of the bug handling communities of OSS projects hosted on SOURCEFORGE have been analyzed in [4]. Here it was shown that large projects - measured in terms of the number of contributors - tend to have lower degrees of centralization in communication. The authors further call for a detailed longitudinal analysis of changes in the social organization of OSS projects during periods of growth. Our work complements this study in the sense that we a) analyze the dynamics of centralization during a phase of growth in the GENTOO community and b) show the impact of increasing centralization on community performance and cohesion.

In addition to studies at the level of the community, the relationship between the network position of contributors and their individual success (like e.g. the number of bug reports leading to bug fixes) has been studied in [5]. The authors find that both the centrality of contributors, as well as their embedding in cohesive clusters of communication has beneficial effects on the bug fixing performance. A similar finding has been presented in [6], which studies the impact of social aspects on individual performance in bug handling communities. Our paper complements this work in the sense that we study network-wide measures of communication efficiency and centralization, their dynamics during community growth, as well as their relation to the bug handling performance of the community. We further highlight potential risks associated with the presence of central contributors in situations when these contributors leave the community unexpectedly.

The relationship between communication structures and success at the level of teams was studied in [7]. Here it was shown that positive team performance is related to communication structures that facilitate information dissemination. However, no clear relation between differences in the coordination practices and the project success could be identified. In [8], the dynamics of collaboration structures of 14 OSS communities has been studied. Similarly, in [9], co-ordination practices of the bug handling process have been studied for four OSS communities. The authors found that contributions are not distributed equally and that the community is organized in a core-periphery structure. Unequal division of labor and an increasing degree of centralization are compatible with findings about the rise of a leader are presented in [10]. Here, a leader is defined as a contributor who consistently provides high quality contributions, co-ordinates efforts [11] and around whom the community is centered [12]. Usually, leadership in OSS projects is shared between several contributors. The analysis performed in [13] shows that overdependence on a leader results in an unstable situation where the project may accelerate - initially - its development, but which may end up saturating the leader.

The present paper extends these previous works in the following way. First, we study the dynamics of a more comprehensive set of network measures that can be interpreted in terms of *cohesion*, *centralization* and *communication efficiency*. We particularly study how the social organization of the GENTOO community evolves during an initial phase of growth and a subsequent phase of increasing centralization that is due to the presence of a central contributor. We then relate our results with proxies for community performance and study how both performance and social organization are impacted by the loss of a central contributor. Finally, we interpret and substantiate our findings by means of insights from actual contributors to the GENTOO community.

III. METHODOLOGY

In our study of the dynamics of social organization in the bug handling community of GENTOO, we use the project's installation of the BUGZILLA bug tracker as data source. We first describe our process of retrieving data and extracting evolving collaboration networks. We then introduce the quantitative measures applied in our analysis of collaboration networks and briefly comment on their interpretation in the context of OSS projects. Furthermore, we summarize how we selected three community members in order to substantiate our findings by means of personal insights from former and active contributors to the GENTOO project.

A. Data Collection

In January 2002, the GENTOO community started to use the BUGZILLA bug tracking system. The full history of all bug reports submitted since then are recorded in the database of the project's BUGZILLA installation. Data available for each of these bug reports include the history of all updates to any field along with time stamps and the ID of the user who applied the update. In the context of our analysis, we particular extract the ID of the user who initially submitted a bug report, as well as the time of the submission and the status of a bug report, like e.g. *unconfirmed*, *pending*, *reproduced* or *resolved*. For those bugs whose final status was set to *resolved*, we additionally collected the *resolution* field of the report, which can take one of the values *fixed*, *duplicate*, *invalid*, *needinfo* or *wontfix*. An entry *fixed* refers to those bugs for which the community eventually provided a fix. Bug reports whose *resolution* field was set to *duplicate* were identified to be duplicates of an existing bug report that refers to the same issue. Bugs with the final resolution *invalid* are those that do not refer to actual software issues, instead referring for instance to a misunderstanding on the user's side. If a bug report is incomplete in the sense that it lacks important information that would allow to reproduce or fix the underlying issue and if the reporting user fails to provide the necessary information within a certain time, the *resolution* field of a

bug report is set to *needinfo*. Finally, the resolution of those bug reports that are valid and complete, but that nevertheless cannot be fixed either due to a lack of resources or the fact that the issue is due to a external dependency are marked as *wontfix*. The fact that all changes to the *resolution* field of a bug report as well as the submission of the bug report itself are associated with a precise time stamp, further allows us to compute the number of bugs that were submitted or resolved with a given status within a given period of time. In addition to all updates that relate to the resolution status of a bug, we also extracted the full history of the *assignee* and the *cc* fields of each bug report. The *assignee* field contains the ID of the user who was made responsible for providing a solution for a particular bug report, while the *cc* field contains a list of user IDs that are being notified about any future updates on a particular bug.

All of the data were collected via the public *API* of the GENTOO project's BUGZILLA installation. In total, we retrieved data on $140,216$ bug reports and $661,783$ change events recorded between January 1st 2002 and April 26th 2012. Some statistics of the data set, including the fraction of resolved bugs falling in each of the aforementioned resolution categories are shown in Table I.

B. Network Construction

A core aspect of our study is the quantitative analysis of the collaboration structure of the GENTOO community during particular periods of time. Even though our data set contains the full record of updates to bug reports, for the construction of collaboration networks, we limit our study to those update events that unambiguously capture dyadic *social* interactions between two contributors. In particular, for each addition of a user ID to the *cc* and *assignee* field of a bug report, we infer a dyadic interaction between the contributor performing the change and the ID of the user that was added to the field. We further associate this interaction with the time stamp of the associated update of the bug report. Focusing on updates to the *cc* and *assignee* fields of bug reports necessarily provides a limited perspective on the social organization of a community. Nevertheless we decided to neglect additional data like e.g. the sequence of comments on bugs for which an inference of directed interaction networks is more difficult and error-prone. We rather argue that the collaboration networks resulting from our construction procedure are nevertheless insightful. The fact that a contributor A adds contributor B to the *cc* field of a bug indicates that A is aware of B and that A knows about the interests or competencies of B. Furthermore, the fact that contributor X adds Y to the *assignee* field of a bug report highlights that these contributors have different roles in the community, like e.g. X identifying the cause of an issue and assigning it to Y.

Excluding those change events where contributors added themselves to the *cc* or *assignee* field, we infer more than $150,000$ directed interactions between different members of the GENTOO community. The structure and dynamics of these interactions can be studied in terms of a *collaboration network* in which nodes represent contributors and directed edges represent interactions between them. A quantitative analysis of such network structures can reveal interesting insights into the community's organization. Rather than aggregating all interactions occurring over a period of ten years, we further utilize the fact that all interactions inferred from our data set

TABLE I
BASIC STATISTICS OF THE BUGZILLA DATA SET USED FOR THIS STUDY.

Statistic	GENTOO
Observation period	01/04/2002 to 04/26/2012
Bug reports	140,216
Change events	661,783
Users	36,555
Collaboration events	153,610
Change events / Bug reports	4.72
Resolved (Resolved / Bug reports)	86,352 (0.61)
FIXED (FIXED / Resolved)	39,858 (0.46)
DUPLICATE (DUPLICATE / Resolved)	20,529 (0.24)
INVALID (INVALID / Resolved)	14,923 (0.17)
WONTFIX (WONTFIX / Resolved)	7,959 (0.09)
NEEDINFO (NEEDINFO / Resolved)	3,083 (0.04)

are *time-stamped*. In particular, we define a time window of 30 days, filter out all interactions whose time stamps are outside this time window and construct a network from all remaining interactions (see an illustration of this procedure in Figure 1). Starting on the first day of the observation period, we then progressively slide the start date of this time window by one day increments. This sliding window approach yields a sequence of $3,765$ networks, each of them representing the collaboration structures of the community within a 30 day period starting at a particular day. By analyzing this sequence of networks, we obtain a time series of network measures that capture the dynamics of social organization. It is important to note that the collaboration networks obtained in the way described above are not necessarily connected, i.e. they may consist of different disconnected components. In order to still provide a single measure that can be compared to previous and subsequent snapshots, we limit our analysis to the network's *largest connected component* (LCC). We additionally measure the size of the LCC and indicate its relative size in terms of the fraction of all nodes that are connected to the LCC.
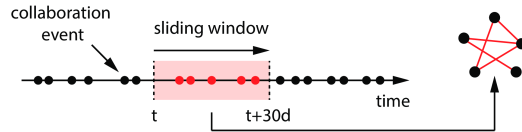

Fig. 1. Sliding window procedure for the construction of evolving networks.

C. Network Measures

In the following, we briefly introduce a number of quantitative, network-theoretic measures that we found to capture interesting aspects of the dynamics of social organization in the GENTOO community. For the sake of brevity, we omit the formal definition of these measures and rather introduce the motivation and interpretation in the context of our study. For the formal definition of the measures mentioned in this section, we refer the interested reader to [14], [15]. For our analysis we use their respective implementations in the network analysis package IGRAPH [16].

a) Closeness centralization: The first measure that we applied in our analysis is *closeness centrality*. The normalized closeness centrality of a node can be defined based on the inverse of the sum of the shortest path lengths to all other nodes in the network. As such, it captures the centrality of a node in terms of how close it is to all other nodes in the network. Based on the distribution of *closeness centralities* of all nodes, one can furthermore define the so-called *closeness centralization* of a network. This network-wide measure

captures the degree to which the topology is *centralized*. In a (maximally centralized) star network it takes a maximum value of 1 while it is 0 for networks in which all shortest paths between all pairs of nodes have the same length (like e.g. a fully connected topology). In the context of our analysis, the closeness centralization of a collaboration network captures to what degree contributors have the same importance for indirect information exchange. Precisely, in a network with maximum closeness centralization all collaborations are mediated by a single individual, while in networks with smaller closeness centralization community members have more equal roles.

b) Clustering coefficient: The *clustering coefficient* of a network measures how closely community members interact with each other in the sense that interactions between users A and B, as well as between B and C will also entail a direct interaction between the users A and C. This property of a network can be quantified at the level of nodes by computing the fraction of those pairs of a node's neighbors u and v that are connected by a direct link (u, v). By averaging the clustering coefficient scores of all nodes it is possible to measure the global clustering coefficient of a network. In the context of our analysis, the (mean) clustering coefficient of a monthly collaboration network captures how *cohesive* the community is in terms of contributors being embedded in collaborating clusters. In other words, this measure captures to what extent two collaborators also collaborate with other collaborators of their peers.

c) Degree Assortativity: The *degree assortativity* of a node measures an individual's preference to connect to peers that have a similar or different number of connections (degree). Networks in which nodes are preferentially connected to nodes with similar degree are called assortative. A positive degree assortativity indicates a positive correlation between the degrees of neighboring nodes. Networks in which nodes are preferentially connected to nodes with different (i.e. smaller or higher) degree are called disassortative, in which case degree assortativity is negative. Zero *degree assortativity* means that there is no correlation between the degrees of connected nodes, i.e. nodes do not exhibit a preference for one or the other. In our analysis, we use degree assortativity to capture the contributors' preference to collaborate with other contributors that are - from the perspective of the number of collaborations - of similar or different importance than themselves.

d) Algebraic Connectivity: An interesting aspect of network analysis is that the influence of a network topology on *dynamical processes* like e.g. information flow, cascading failures or synchronization phenomena can be captured by means of so-called *spectral* properties. One important measure in this line is the so-called *algebraic connectivity* of a network. This scalar property particularly captures whether the topology contains *small cuts*, i.e. whether all shortest paths between different parts of the network pass through a small number of edges. The existence of such small cuts is known to hinder information spreading and synchronization [17]. At the same time, it can be seen as an indicator for robustness since it captures the effect of a failure of a small number of nodes and associated links. The algebraic connectivity is defined as being the second smallest eigenvalue of the so-called Laplacian matrix, which is defined as the difference $D - A$ between a diagonal matrix D in which the diagonal elements represent the degrees of nodes and the adjacency matrix A of the network topology. The algebraic connectivity

of a network is greater than 0 *iff* the network topology is connected, i.e. iff a path exists between all pairs of nodes. This is a corollary to the fact that the number of times 0 appears as an eigenvalue of the Laplacian matrix is equal to the number of the network's connected components. In the context of this paper we use algebraic connectivity it to measure the communication efficiency and robustness of the the community's collaboration structure.

D. Interviews with community members

In order to substantiate our quantitative findings with insights into the community, we contacted a number of long-term contributors to the GENTOO bug handling community. We received three very insightful and detailed replies, which contain many details and serve as an external validation for our quantitative findings. We omit the real names of the contributors and refer to them as *Alice*, *Bob* and *Chris* instead. *Alice* was the - by far - most central contributor to the GENTOO bug handling community in the period between October 2004 and March 2008. She was effectively handling most of the bug reports, until she left the project suddenly in March 2008. *Bob* was involved in a community initiative to establish formal procedures regarding the submission and handling of bug reports that were - in part - necessitated by the departure of *Alice*. *Chris* is another long-term contributor to the project, second only to *Alice* in terms of cumulative contributions to the bug handling process. In our questionnaire, we asked for personal insights regarding the following questions:

- What was the impact of the central contributor *Alice* on the involvement of other contributors and project performance?
- What were the reasons for *Alice* leaving the project?
- What was the motivation for the establishment of formal procedures for the bug handling process?
- Was this initiative successful in terms of improving the performance of the community?
- What implications did the establishment of formal procedures have for the social organization of the community?

IV. DYNAMICS OF SOCIAL ORGANIZATION AND PERFORMANCE

In the following, we study the dynamics of GENTOO's bug handling community during the time between 2002 and 2012. Our methodology is based on a network-theoretic analysis of collaboration networks by means of the measures discussed in section III.

Based on the activity of the central contributor *Alice*, we divide the observation period into three periods $P1$, $P2$, $P3$. In period $P1$, between January 2002 and October 27, 2004, *Alice* was not yet active and the community was growing. During the second period $P2$ starting on October 28 2004, *Alice* gradually became the most central contributor. She unexpectedly left the community after her last contribution on March 29 2008, which marks the start of the third period $P3$ in which $Alice$ was not active anymore. In the following sections, we show how community cohesion, centralization and performance evolved in these three periods.

A. Community Cohesion

We first focus on the size of the largest connected component (LCC) of the respective monthly collaboration networks. The relative size of the LCC (i.e. the fraction of all nodes

belonging to the LCC) is shown in Figure 2(a). Since it captures how many of the contributors were disconnected from the rest of the community, this measure can be seen as a proxy for the *cohesion* of the community. In Figure 2(a), period $P2$ is highlighted in green. As one can see, there is no significant difference between the periods $P1$ and $P2$ in terms of the relative size of the LCC; it rather remains stable around a value of 75%. However, a remarkable dynamics can be seen in period $P3$ after *Alice* had left the community: After a small drop, one observes a steady increase in the relative size of the LCC starting around the end of 2008. The relative size of the LCC eventually reaches 95% around the end of 2011.

Another *cohesion*-related measure is the average node degree in the monthly collaboration networks, i.e. the average number of different community members, a contributor was collaborating with during one month. In Figure 2(f), one observes a fast decrease of this measure during period $P2$, when the central contributor *Alice* was active. Remarkably, it was *increasing* during the periods $P1$ and $P3$, when *Alice* was not active.

Apart from the relative *size* of the LCC, a further interesting question is how efficient and robust the collaboration structures are *within* the largest connected component. For this, we compute the *algebraic connectivity* of the LCC, a measure from spectral graph theory that captures how *well-connected* a topology is. As argued in section III, networks with larger algebraic connectivity a) facilitate information flow and synchronization processes and b) are more robust against the loss of nodes and links. The dynamics of algebraic connectivity is shown in Figure 2(c). Comparing period $P2$ to $P1$ and $P3$, one observes that the presence of *Alice* decreased both the variance and the mean of the algebraic connectivity. A straight-forward interpretation of this finding is that - as *Alice* was involved in many of the collaborations - the collaboration network's robustness decreased. Furthermore, as most collaborations were mediated through her, the potential of congestion in this particular node increased, thus effectively decreasing communication efficiency of the topology.

Another interesting question from the perspective of social organization is to what degree two contributors that collaborated with a third contributor also collaborated with each other. This is captured by the clustering coefficient of a network, whose dynamics is shown in Figure 2(g). The dramatic decrease of the clustering coefficient during period $P2$ and the gradual increase in period $P3$ highlights the mediator role played by *Alice*. As *Alice* was involved in most of the collaborations, direct connections between users collaborating with her seemingly became unnecessary. Another signature of the community's tendency to preferentially collaborate with the most central collaborator can be seen in 2(h). As described in section III, the assortativity captures the preference of contributors to collaborate with other contributors that are more or less important than themselves. A significant decrease of assortavitity from about -0.15 to -0.45 can be seen in period $P2$ when *Alice* was active. This substantiates the assumption that most community members primarily collaborated with the most central collaborator while collaborations between contributors with similar importance decreased.

A particular concern one may have in the analysis presented above is that it is unclear to what extent it is the presence of *Alice* that affects the dynamics of network measures. One may suspect that it is the mere number of collaborations

(a) relative size (b) relative size (without *Alice*)

(c) algebraic connectivity (d) algebraic con. (without *Alice*)

(e) network size (f) mean degree

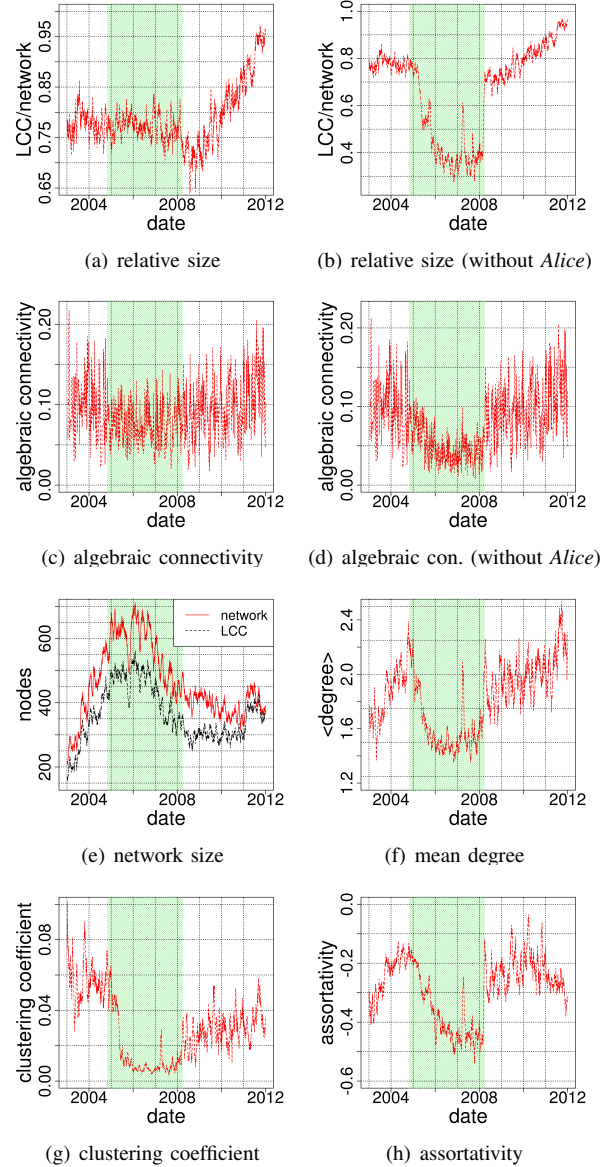(g) clustering coefficient (h) assortativity

Fig. 2. Dynamics of size and cohesion of the GENTOO bug handling community. Period $P2$ during which the central contributor *Alice* was active is highlighted in green.

involving her that increasingly dominate the community, while the existing collaboration structures are left more or less untouched. In order to avoid this pitfall, we additionally ran our analysis on all monthly collaboration networks, however removing *Alice* as well as all interactions in which she was involved. We then computed the relative size of the LCC and algebraic connectivity to the residual networks. Compared to Figures 2(a) and 2(c), a clear difference can only show up during period $P2$, if *Alice*'s presence indeed impacted the residual collaboration structures. The plots of the relative size of the LCC (Figure 2(b)) and algebraic connectivity (Figure 2(d)) of the residual collaboration networks highlight that the activity of *Alice* during period $P2$ significantly changed the organization of the community. We particularly observe that - for the residual network - the fraction of users connected to the LCC dropped significantly from about 75% to about 30% over a period of two years. Furthermore, algebraic

connectivity of the residual network experienced a significant drop, thus highlighting that during *Alice*'s presence the residual collaboration topology became less well-connected.

To visually illustrate the quantitative findings about the evolution of collaboration structures provided above, in Figure 3 we additionally show four representative examples for the monthly collaboration networks during the periods $P1$ (Figure 3(a)), $P2$ (Figure 3(b)) and $P3$ (Figure 3(c)). In addition, 3(d) depicts an example for a residual network constructed by removing all interactions of *Alice* from the network depicted in Figure 3(b).

From our quantitative study of the evolution of collaboration structures in the *Gentoo* community, we can draw the following observation:

Observation: *During the presence of the central contributor* Alice*, cohesion in the* GENTOO *bug handling community decreased.*

(a) October 2004

(b) July 2006

(c) May 2008
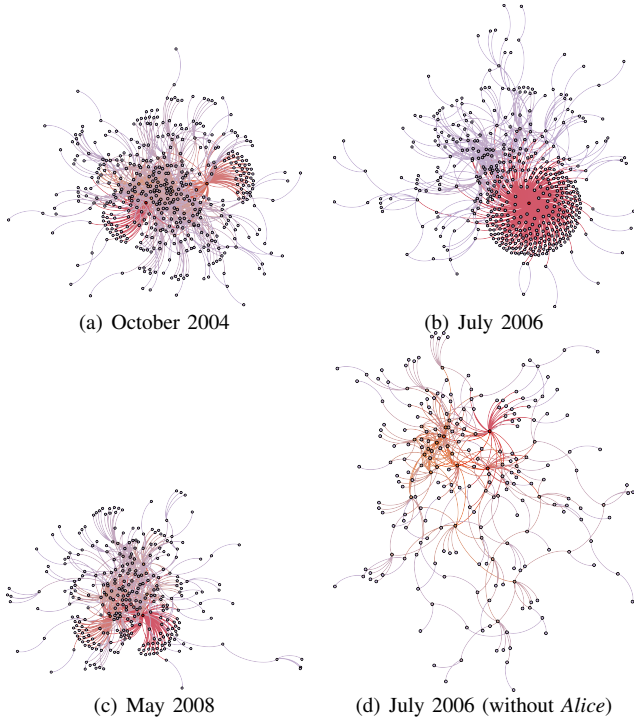
(d) July 2006 (without *Alice*)

Fig. 3. Illustration of representative monthly collaboration networks

B. Centralization

A particularly important mechanism that could explain the loss of cohesion in the community is an increasing centralization of communication. In this section we analyze the changes in centralization in the GENTOO community. We not only study centralization from a network perspective, i.e. the increase of the *topological* centrality of *one* particular node. We also consider the effects on the number of contributors that were involved in the bug handling process in terms of assigning bug reports or forwarding information.

We first analyze the degree of centralization in the GENTOO community from the perspective of *closeness centralization*. As argued in section III, this measure captures to what extent the roles of contributors differ in terms of having short paths to all other community members. The dynamics of closeness centralization shown in Figure 4(a) exhibits a decreasing tendency during the period $P1$. A comparison to

the dynamics of community size during $P1$ (see Figure 2(e)) highlights that the growth of the community coincided with a decrease in centralization, which is in line with the findings of [4]. However, the decrease in closeness centralization in period $P1$ was followed by a significant increase during period $P2$ when *Alice* became active. From the start of period $P2$ in October 2004 until the end in March 2008 closeness centralization increased from about 0.3 to 0.7. When *Alice* left the community, closeness centralization experienced a sudden drop, fluctuating around a value of 0.4 during the period $P3$.

The finding that during period $P2$ the collaboration structures became more centralized is complemented by Figure 4(b), which shows the number of different contributors assigning a bug report to another contributor within a given 30 day period. This number is of particular interest, as it captures how many contributors were actually involved in the bug triaging process by *assigning* work to others. Again mirroring the increasing size of the community, in period $P1$ one observes an increase in the number different contributors assigning bug reports. At the end of period $P1$ in October 2004, about 170 different contributors were assigning bug reports. This increase is followed by a *decrease* during period $P2$, again coinciding with the activity of *Alice*. This development was only stopped in March 2008, after *Alice* had left the project. After a sudden increase at the beginning of $P3$, the number of different contributors assigning bug reports remained rather stable until 2011, when it experienced another increase.

From the above analysis, we draw the following observation:

Observation: *In the period where* Alice *was active, centralization in the* GENTOO *community increased steadily.*

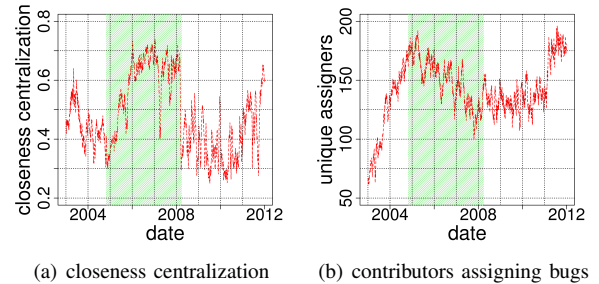(a) closeness centralization

(b) contributors assigning bugs

Fig. 4. Centralization in the GENTOO community. Period $P2$ during which the central contributor *Alice* was active is highlighted in green.

C. Bug Handling Performance

Apart from studying the evolution of collaboration structures, our data set further allows to study the *bug handling* performance of the GENTOO community. As the simplest proxy for performance, we measure the rate at which bugs were reported and resolved. We further study the responsiveness of the community in terms of the *median time to resolve a bug*, i.e. the median time elapsed from the submission of a bug report to the point where it was finally resolved. Similarly, we measure the *median time to the first response* in terms of any update to the submitted bug report, like e.g. the bug being forwarded or assigned, commented on, or its status being changed to reproduced. Figure 5(a) shows the dynamics of the median number of bugs that were reported and resolved per day. During period $P1$ one observes a continuous increase both in the number of reported and resolved bugs which coincides with the growth of the GENTOO community shown in 2(e).

During period $P2$, both the number of reported and resolved bugs decreased, which can again be understood based on the decrease in the number of active contributors shown in Figure 2(e). In both periods $P1$ and $P2$, the rate of reporting and resolving bugs closely match each other, thus indicating that - on average - the number of bugs resolved per day matched the number of newly reported bugs. This lastingly changed after *Alice* had left the project. In period $P3$ one can observe an increasing discrepancy between the rate at which bugs were reported and resolved, hence indicating a growing number of unresolved, pending bug reports. Furthermore, a remarkable increase in both the number of reported and resolved bug reports can be seen around March 2011, although the discrepancy between both remains. This coincides with an increase in the number of active community members (see Figure 2(e)). One possible explanation is that it coincides with the GENTOO community having a regular *LiveDVD* release. As it lowers the threshold of using the *Gentoo Linux* distribution, this can explain an increasing number of contributors submitting bug reports, as well as the increase in the number of different contributors assigning bug reports shown in Figure 4(b).

Apart from the mere number of reported and resolved bugs, an important measure of performance of bug handling communities is the time they take to provide a first response as well as a resolution for a reported bug. This *responsiveness* is of particular importance, as potential users frequently use this as an indicator when making an informed decision about which software to adopt. Figure 5(b) shows the median time resolve and respond to a newly reported bug in days and hours respectively. Both numbers show a remarkable dynamics which coincides with the activity of the central contributor *Alice*. During period $P2$, the median time to resolve and respond to a newly submitted bug report was more than one order of magnitude smaller than in the periods $P1$ and $P3$.

From our analysis of bug handling performance, we thus draw the following observation:

Observation: *During the presence of the central contributor* Alice, *the bug handling performance of the* GENTOO *community increased significantly, while her retirement had a lasting negative impact.*

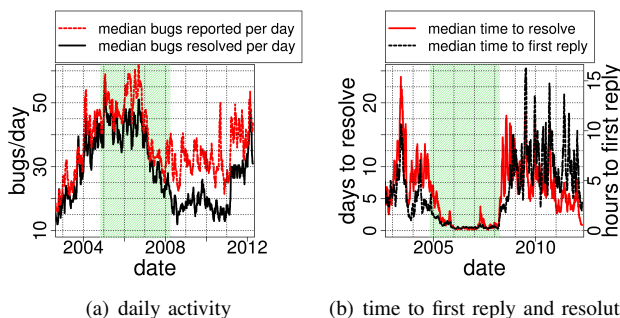(a) daily activity   (b) time to first reply and resolution

Fig. 5. Bug handling performance in the GENTOO community. Period $P2$ during which the central contributor *Alice* was active is highlighted in green.

D. Discussion

We close this section by combining our quantitative results with personal insights shared by three long-term contributors: *Alice*, *Bob* and *Chris*. By this, we substantiate our interpretation of *Alice*'s role during period $P2$ and the consequences of her presence for the cohesion and performance of the GENTOO community.

In sections IV-A, IV-B, and IV-C, we point out that during period $P2$ the community experienced a significant loss of cohesion, as well as an increase of centralization and performance. We further argued that during period $P2$, most of the collaboration was mediated by a small subset of contributors, *Alice* herself being at the core of this group. Indeed, in response to our questionnaire, *Alice* describes that she "was practically the only person involved in bug wrangling". *Bob* confirms that *Alice* "had been doing our *bug wrangling* more or less alone for a few years". *Alice* complements this picture by saying that the "workload at that time - [if I recall correctly] - was about 4 hours a day, probably more in case I did not have time to do the bug wrangling for a day or two". As a consequence of this centralization of bug handling tasks, during period $P2$ our analysis shows a significant increase in performance, measured in terms of response time and bug resolution rate. This finding is confirmed by the community and *Bob* attributes it to the fact that "having a single person on the task greatly helps in finding duplicate bug reports". Furthermore, he argues that having "more [contributors] would water down the quality".

A further observation of our study is that the cohesion of the community (measured e.g. in terms of mean degree, clustering coefficient or algebraic connectivity) decreased significantly during the presence of *Alice*. This is an interesting observation as it highlights secondary effects of the presence of a central contributor on the evolution of collaboration structures within the remaining community. Although it is necessarily difficult to make any substantiated claims about causality, one may conjecture that it is the mere presence and dedication of a central contributor that drives this loss of cohesion. *Bob* indirectly confirms this by arguing that apparently "our bug tracker's users had come to rely on a single person to "assist" them in finding and fixing bugs".

For the community, the retirement of *Alice* was perceived as an unexpected event. According to *Bob*, in 2008 *Alice* "suddenly left the project". He further confirms that she "stopped unexpectedly". Clearly, one of the most interesting questions that cannot be answered by a quantitative study alone is why *Alice* decided to leave the community. She answered our question for the underlying reasons as follows: "I would mostly attribute that to a serious loss of motivation caused by disruptive social environment in the project as a whole". Moreover, she highlights her dissatisfaction with "more and more time being spent on bureaucracy, "paperwork", and creating of useless structures within the project". On the contrary, *Chris* - another prominent contributor - remarks that "some people find formalization to be an unnecessary bureaucratic barrier, but when you get to be as big as Gentoo, it's pretty much inevitable".

Independently of the reasons for *Alice*'s retirement, the risk of relying too much on a central contributor became obvious in a remarkable event during period $P2$, when *Alice* was still active. In early 2007, according to her own account, *Alice* was "repeatedly subject to [...] disciplinary proceedings and [she] was suspended from the project for a couple of weeks" due to a verbal conflict with another contributor. Around this time, a sudden and short increase in the response time (see Figure 5(b)) as well as a decrease in closeness centralization (see Figure 4(a) can be observed, thus serving as an early warning sign of the problems to come when *Alice* would leave.

Despite this early indicator, it was only after *Alice* had

left that the community took measures to reorganize the community. In particular, *Bob* initiated the BUG WRANGLERS project, which a) called for more contributors in bug handling and b) established formal procedures regarding the tasks and goals of bug triaging[1]. In response to our questions, *Bob* describes the project as a success arguing that "the targets that relate to the content of bug reports are now usually met when serious bug wranglers review them". However, despite this initiative, our finding of a lasting negative impact on bug handling performance after the resignation of *Alice* is confirmed by *Bob*, saying that the "goal of responding to bugs within a day is still something to work on".

E. Threats to Validity

We now discuss limitations of our analysis and highlight possible threats to validity. Since our paper is a case study focused on the GENTOO community, we cannot make any claims about the general applicability of our results. Even though our study as well as the feedback by the community provide some interesting hints, we would further like to emphasize that we cannot make conclusive statements regarding the causal relation between increasing centralization, performance and cohesion. In particular, we cannot rule out external reasons driving *both* the increase of centralization and the loss of cohesion in the community. Despite this disadvantage, we argue that our case study is interesting by itself, being a valuable addition to the literature on benefits and risks of centralization in collaboration topologies. In order to validate our findings, we thus call for similar studies on OSS communities and other collaborative software engineering projects.

Another possible concern is the choice of our network construction procedure as well as the choice of length of the sliding window in our dynamic analysis. In order to only extract *meaningful* collaboration events and facilitated by the size of our data set, we only considered *cc* and *assign* collaborations. Nevertheless, it is clear that taking into account further relations, like e.g. comments, could possibly augment our perspective of collaboration topologies. At the same time, we argue that - even though we have explored different sizes for the sliding window - we did not see any qualitative change of our results. Eventually, we decided to include the results of a 30 day window size, since this period is long enough to include collaborations of more occasional collaborators. At the same time, a one month period is short enough to not aggregate collaborations occurring far apart in time. As such, our methodology of performing a *dynamic network analysis* can be seen as a strength compared to the simpler approach of considered a single time-aggregated network.

V. CONCLUSION

The main contributions of our paper are the following:

- We study the dynamics of social organization and performance in the bug handling community of GENTOO.
- We find a period in which the activity of a single contributor resulted in a significant increase of centralization and performance.
- Our analysis further shows that the period when the central contributor was active coincided with a significant decrease of cohesion.

---

[1] See the website of the BUG WRANGLERS project available online at http://www.gentoo.org/proj/en/qa/bug-wranglers/index.xml

- We further find that the loss of the central contributor had a lasting negative impact on the bug handling performance of the community.

To the best of our knowledge, our paper is the first to quantitatively study how the rise and fall of a central contributor impact the social organization and performance of an OSS community. Even though the general statements that can be drawn from a case study are necessarily limited, we argue that our work highlights interesting directions for future research. We would like to emphasize that the quantitative measures used in our study allow to clearly identify shifts in the social organization that are confirmed by insights by actual contributors. As such, we argue that these measures can potentially be used in monitoring tools suitable to augment the social awareness of community managers.

REFERENCES

[1] M. Zhou and A. Mockus, "What make long term contributors: Willingness and opportunity in OSS community," *2012 34th International Conference on Software Engineering (ICSE)*, pp. 518–528, Jun. 2012.

[2] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology*, vol. 11, no. 3, pp. 309–346, 2002.

[3] N. Bettenburg, R. Premraj, and T. Zimmermann, "Duplicate bug reports considered harmful really?" in *2008 IEEE International Conference on Software Maintenance*. Ieee, 2008, pp. 337–345.

[4] K. Crowston and J. Howison, "The social structure of Free and Open Source Software development," *First Monday*, vol. 10, 2005.

[5] K. Ehrlich and M. Cataldo, "All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, ser. CSCW '12. New York, NY, USA: ACM, 2012, pp. 945–954.

[6] M. S. Zanetti, I. Scholtes, C. J. Tessone, and F. Schweitzer, "Categorizing bugs with social networks: A case study on four open source software communities," 2013, accepted for the 35th ICSE - Software Engineering in Practice track.

[7] T. Wolf, A. Schroter, and D. Damian, "Mining task-based social networks to explore collaboration in software teams," *IEEE Software*, pp. 58–66, 2009.

[8] M. S. Zanetti, E. Sarigöl, I. Scholtes, C. J. Tessone, and F. Schweitzer, "A quantitative study of social organisation in open source software communities," in *ICCSW*, 2012, pp. 116–122.

[9] K. Crowston and B. Scozzi, "Coordination practices within FLOSS development teams: The bug fixing process," *Computer Supported Acitivity Coordination*, 2004.

[10] P. Giuri, F. Rullani, and S. Torrisi, "Explaining leadership in virtual teams: the case of open source software," *Inform. Econ. Policy*, vol. 20, pp. 305–315, 2008.

[11] B. Scozzi, K. Crowston, U. Yeliz Eseryel, and Q. Li, "Shared mental models among open source software developers," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. IEEE, 2008, pp. 306–306.

[12] P. Evans and B. Wolf, "Collaboration rules," *Harvard Bus. Rev.*, vol. 7, pp. 96–103, 2005.

[13] B. Sadowski, G. Sadowski-Rasters, and G. Duysters, "Transition of governance in a mature open software source community: evidence from the debian case," *Inf. Econ. Policy*, vol. 20, pp. 323–332, 2008.

[14] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[15] M. E. J. Newman, *Networks: an introduction*. Oxford Univ Press, 2010.

[16] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006.

[17] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2001.