**Temporal Analysis of Dynamic Collaboration Graphs of**

**Open Source Software Development: Forking**

Amir Azarbakht

School of Electrical Engineering and Computer Science

Oregon State University

Corvallis, Oregon

[azarbaam@eecs.oregonstate.edu](mailto:azarbaam@eecs.oregonstate.edu)

ABSTRACT

How can we understand FOSS collaboration better? Can so- cial issues that emerge be identified and addressed as they happen? Can the community heal itself, become more trans- parent and inclusive, and promote diversity? We propose a technique to address these issues by quantitative analy- sis and temporal visualization of social dynamics in FOSS communities. We propose using social network analysis to identify unhealthy dynamics; this will help predict forma- tion of unhealthy dynamics and which gives the community a heads-up when they can still take action to ensure the sustainability of the project.

Categories and Subject Descriptors

H.5.m [Information interfaces and presentation (e.g., HCI)]: [Miscellaneous]

General Terms

Measurement, Reliabality, Human Factors

Keywords

Free/Open Source Software, Social Dynamics, Temporal Anal- ysis, Forking, Visualization, Temporal Visualization, Social Network Analysis, FOSS, FLOSS.

1. INTRODUCTION

Social networks are a ubiquitous part of our social lives, and the creation of online social communities has been a natural extension of this phenomena. Software media plays an im- portant role in software engineering, as software developers use them to communicate, learn, collaboborate and coordi- nate with others.[34] Free and Open Source Software (FOSS) development efforts are prime examples of how community can be leveraged in software development, groups are formed around communities of interest, and depend on continued interest and involvement in order to stay alive.[25]

Though the bulk of collaboration and communication in FOSS communities occurs online and is publicly accessi- ble, there are still many open questions about the social dynamics in FOSS communities. Projects might go through a metamorphosis when faced with an influx of new develop- ers or the involvement of an outside organization. Conflicts between developers’ divergent visions about the future of the project might lead to forking of the project and dilution of the community. Forking, either as a violent split when there is a conflict or as a friendly divide when new features are experimentally added both affect the community [4].

Research on forking exists. It ranges from the studies of Robles et al. [30] on forking that identified all significant FOSS forks since 1990, to the works of Baishakhi et al. [2] on post-forking porting of new features or bug fixes from peer projects. It encompasses works of Nyman on develop- ers’ opinions about forking [27], developers motivations for performing forks [23], the necessity of code forking as tool for sustainability [26], and Syeed’s work on socio-technical dependencies in the BSD projects family [35].

Most research on forking, however, is post-hoc. It looks at the forking events in restrospect and tries to find the outcome of the fork; what happened after the fork happened; what was the cause of forking, and such. The run-up to the forking events are seldom looked at. This leaves a number of questions unanswered: Was it a long-run trend? Was the community polarized, before forking happened? Was there a shift of influence? Did the center of gravity of the community change? What was the tipping point? Was it predictable? Is it ever predictable? We are missing that context.

Additionally, studies of FOSS communities tend to suffer from an important limitation. They treat community as a static structure rather than a dynamic process.

In this study, we propose to use temporal social network analysis to study the evolution and social dynamics of FOSS communities. With these techniques we aim to identify bet- ter measures for influence, and the shift of influence, mea- sures associated with unhealthy group dynamics, e.g. a sim- mering conflict, as well as early indicators of major events in the lifespan of a community. One set of dynamics we are especially interested in, are those that lead FOSS projects to fork.

This paper is organized as follows: We present related lit- erature on online social communities. We then present the gap in the literature, and discuss why the issue needs to be addressed. After that, in methodology, we describe research goals and research questions, how data gathering, analysis, and visualization of the findings is proposed to be done. At the end, we present preliminary results, discussion and threats to validity.

2. RELATED WORK

The social structures of free and open source software de- velopment communities have been studied extensively. Re-

searchers have studied the social structure and dynamics of team communications [5][16][17][13][22], identifying knowl- edge brokers and associated activities [33], project sustain- ability [26][22], forking [25], requirement satisfation [9], their topology [5], their demographic diversity [19], gender differ- ences in the process of joining them [18] and the role of the core team in their communities [37], etc. Most of these stud- ies have tended to look at community as a static structure rather than a dynamic process [8]. This makes it hard to determine cause and effect, or the exact impact of social changes.

Table 1: The main reasons for forking as classified by Robles and Gonzalez-Barahona [30]

Reason for forking Example forks

of lines in BSD release patches consist of ported edits, and on

|  |  |  |
| --- | --- | --- |
|  | Technical (Addition of func- | Amarok & Clementine |
| Post-forking porting of new features or bug fixes from peer | tionality) | Player |
| projects happens among forked projects [2]. A case study of | More community-driven de- | Asterisk & Callweaver |
| the BSD family, i.e. FreeBSD, OpenBSD, and NetBSD, all | velopment |  |
| of which evolved from the same codebase found that 10-15% | Differences among developer | Kamailio & OpenSIPS |

team

average 26-58% of active developers participate in porting

Discontinuation of the origi-

Apache web server

per release. They also found that over 50% of ported changes

propagate to other projects within three releases. [2]

Visual exploration of the collaboration networks in the We-

nal project Commercial strategy forks LibreOffice & OpenOf-

fice.org

Legal issues X.Org & XFree

bKit project was the focus of a study that aimed to observe

how key events in the mobile-device industry affected the WebKit collaboration network over its lifetime. [36] They found that coopetition (both competition and collaboration) exists in the open source community; morover, they observed that the “firms that played a more central role in the We- bKit project such as Google, Apple and Samsung were by

2013 the leaders of the mobile-devices industry. While more peripheral firms such as RIM and Nokia lost market-share [36]”.

The study of communities has grown in popularity in part thanks to advances in social network analysis. From the earliest works by Zachary [38] to the more recent works of Leskovec et al. [20][21], there is a growing body of quanti- tative research on online communities. The earliest works on communities was done with a focus on information dif- fusion in a community [38]. Zachary investigated the fission of a community, the process of communities splitting into two or more parts. He found that fission could be predicted by applying the Ford-Fulkerson min-cut algorithm [10] on the group’s communication graph; “the unequal flow of sen- timents across the ties” and discriminatory sharing of infor- mation lead to “subcommunities with more internal stability than the community as a whole [38].”

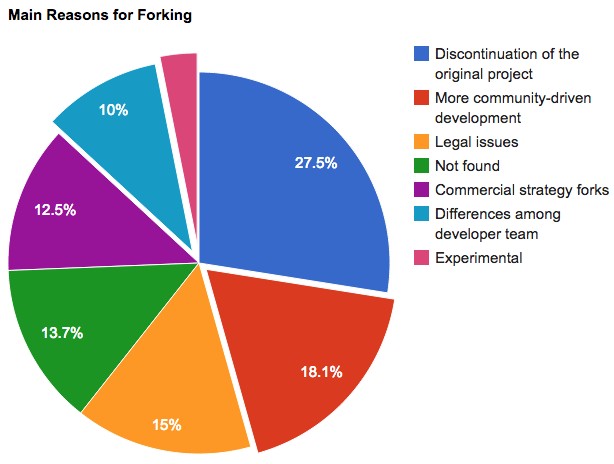
Community splits in FOSS are referred to as forks, and are relatively common. Forking is defined as “when a part of a development community (or a third party not related to the project) starts a completely independent line of develop- ment based on the source code basis of the project.” Robles and Gonzalez-Barahona [30] identified 220 significant FOSS projects that have forked over the past 30 years, and com- piled a comprehensive list of the dates and reasons for fork- ing. They classified these into six main categories. (Table

1.) which we build on extensively. They identified a gap in the literature in case of “how the community moves when a fork occurs [30].”

The dynamic behavior of a network and identifying key events was the aim of a study by Asur et al [1]. They studied

Figure 1: The reasons for forking as classified by Robles and

Gonzalez-Barahona [30]



three DBLP co-authorship networks and defined the evolu- tion of these networks as following one of these paths: a) Continue, b) k-Merge, c) k-Split, d) Form, or e) Dissolve. They also defined four possible transformation events for individual members: 1) Appear, 2) Disappear, 3) Join, and

4) Leave. They compared groups extracted from consecu- tive snapshots, based on the size and overlap of every pair of groups. Then, they labeled groups with events, and used these identified events [1].

Table 2: The measures of diversity used by Asur et al. [1]

Metrics Meaning

Stability Tendency of a node to have interactions

with the same nodes over time Sociability Tendency of a node to have different in-

teractions

Influence Number of followers a node has on a network and how its actions are copied and/or followed by other nodes. (e.g. when it joins/leaves a conversation, many other nodes join/leave the conversation,

too) Popularity Number of nodes in a cluster (how

crowded a sub-community is)

The communication patterns of FOSS developers in a bug repository were examined by Howison et al. [16]. They cal- culated out-degree centrality as their metric. Out-degree centrality measures the proportion of the number of times a node contacted other nodes (outgoing) over how many times it was contacted by other nodes (incoming). They calculated this centrality over time “in 90-day windows, moving the window forward 30 days at a time.” They found that “while change at the center of FOSS projects is relatively uncom- mon,” participation across the community is highly skewed, following a power-law distribution, where many participants appear for a short period of time, and a very small number of participants are at the center for long periods. Our approach is similar to theirs in how we form collaboration graphs and perform our temporal analysis. Our approach is different in terms of our project selection criteria, the metrics we exam- ine, and our research questions.

The tension between diversity and homogeneity in a commu- nity was studied by Kunegis et al. [19]. They defined five network statistics used to examine the evolution of large- scale networks over time. They found that except for the diameter, all other measures of diversity shrunk as the net- works matured over their lifespan. Kunegis et al. [19] ar- gued that one possible reason could be that the community structure consolidates as projects mature.

Community dynamics was the focus of a recent study by Hannemann and Klamma [14] on three open source bioin- formatics communities. They measured ”age” of users, as starting from their first activity and found survival rates and two indicators for significant changes in the core of the community. They identified a survival rate pattern of 20-40-

90%, meaning that only 20% of the newcomers survived after their first year, 40% of the survivors survived through the second year, and 90% of the remaining ones, survived over

the next years. As for the change in the core, they suggested that a falling maximal betweenness in combination with an increasing network diameter as an indicator for a significant change in the core, e.g. retirement of a central person in the community. Our approach builds on top of their findings, and the evolution of betweenness centralities and network diameters for the projects in our study are depicted in the following sections.

3. MOTIVATION

The run-up to the forking events are seldom studied. There are unanswered questions about the run-up to a fork: Was it a sudden change, or a long-run trend? Was the com- munity polarized or united, before forking happened? Was there a shift of influence? Did the center of gravity of the community change? What was the tipping point? Was it predictable? Is it predictable? We are missing that context.

To better understand and measure the evolution, social dy- namics of forked FOSS projects, and integral components to understanding their evolution and direction, we need new and better tools. With this knowledge and these tools, we could help projects reflect on their actions, and help commu- nity leaders make informed decisions about possible changes or interventions. It will also help potential sponsors make informed decisions when investing in a project, and through- out their involvement to ensure a sustainable engagement.

Identification is the first step to rectify an undesired dy- namic before the damage is done. We want to map the dynamics of communities to real world phenomena. A com- munity that does not manage growing pains may end up stagnating or dissolving. Managing growing pains is espe- cially important in case of FOSS, where near half the project contributors are volunteers [11]. Oh et al. [28] have argued that openness in FOSS is “[...] generally perceived as having a positive connotation, however, the term can also be inter- preted as referring to some unconstructive characteristics, such as unobstructed exit, susceptible, vulnerable, fragile, lacking effective regulation, and so on. The unobstructed exit and lack of regulatory force inherent in the OSS com- munity can result in a community’s susceptibility and vul- nerability to herded exits by its participants. Commercial vendor intervention, an alternative project becoming avail- able, and licensing issues can result in some original core members ceasing to provide their loyal service for the com- munity, which can prompt their coworkers to leave as well” [28].

Recipes for success or stagnation, sustainability or fragmen- tation could be identifiable, leading to a set of best practices and pitfalls.

4. RESEARCH GOALS

Social interactions reflects the changes the community goes through, we argue. And so, it can be used to describe the context surrounding a forking event. Three of the six main reasons for forking [30], as listed in Table 1 are so- cially related, and so should be reflected in the social in- teraction data. For example, if a fork occured because of a desire for “more community-driven development”, we expect to see interaction patterns in the collaboration data show- ing a strongly-connected core that is hard to penetrate for

Table 3: The measures of diversity used by Kunegis et al. [19]

|  |  |  |
| --- | --- | --- |
| Network property | Network is diverse when | Diversity Measures |
| Paths between nodes | Paths are long | Effective diameter |
| Degrees of nodes | Degrees are equal | Gini coefficient of the degree distribu-  tion |
| Communities | Communities have similar sizes | Fractional rank of the adjacency ma-  trix |
| Random walks | Random walks have high probability  of return | Weighted spectral distribution |
| Control of nodes | Nodes are hard to control | Number of driver nodes |

the rest of the community. In other words, in this case, the power stays in the hands of the same people throughout, as new people come and go.

We aim to 1) Analyze, quantify and visualize how the com- munity is structured, how it evolves, and the degree to which community involvement changes over time; 2) Test whether the results of the analysis match what developers in the project’s community remember? And whether showing the analysis results to a FOSS developer would result in he/she drawing the same conclusion as the analysis results; and 3) Test if this method helps current FOSS community mem- bers, and whether it is useful to know how and why projects fork. Specifically, our research objectives and research ques- tions are listed in the following.

4.1 Research Objective #1:

What are the social patterns associated with dif- ferent types of forking?

R.Q. 1.1 Is there a prototypical fork for “personal differ- ences” reason for forking?

R.Q. 1.2 Is there a prototypical fork for “more community- driven development” reason for forking?

R.Q. 1.3 Is a labeled project as a “technical differences” fork really only a “technical differences” fork?

Are there patterns that exemplify these categories? What establishes an inflection point (fork)? Which metrics are indicative of inflection?

R.Q. 1.4 What are the determining factors?

R.Q. 1.5 Where are the determining factors? In Mailing

List data? In Code?

R.Q. 1.6 What do the determining factor look like?

4.2 Research Objective #2:

Match between our analysis and the real world? Or, does our analysis reflect what happened? If yes, how well?

R.Q. 2.1 Does my analysis of the situation match what peo- ple in that community remember?

R.Q. 2.2 If I show the analysis results to an open source de- veloper, would they draw the same conclusion as our analysis results?

4.3 Research Objective #3:

Can we use this method to know how and why projects fork?

R.Q. 3.1 Current community folks, have them reflect on it, and estimate the correct future

R.Q. 3.2 Can we use these to know how and why projects fork?

5. METHODOLOGY

5.1 Phase 1: Data Collection

The study of forks by Robles and Gonzalez-Barahona [30] lists 220 significant forks since 1990, and the forking reason associated with each project. We applied three selection criteria to those projects. A project was short-listed if it was recent, i.e. the fork had happened after the year 2000; its data was available; and they had a community of more than a handful of contibutors. This three-stage filtering process resulted in the projects listed in Table 6.

Data collection involved analyzing mailing list archives. We collected data for the year in which the fork happened, as well as for three month before and three months after that year in order to capture the social context context at the time of the fork.

5.2 Phase 2: Creating Communication Graphs Many social structures can be represented as graphs. The nodes represent actors/players and the edges represent the interaction between them. Such graphs can be a snapshot of a network – a static graph – or a changing network, also

Table 4: Projects forked because of “personal differences among the developer team” [30] sorted in chronological order, and their data availability status

Original Forked Date Data available? Collected?

GNU Emacs X Emacs 1991, ? Only after 2000 N/A NetBSD OpenBSD 1995, Oct Yes, but scarce N/A xMule aMule 2003, Aug Only 2006-2007 N/A lMule xMule 2003, Jun - N/A Sodipodi Inkscape 2003, Nov Yes Req.

Nucleus CMS Blog:CMS 2004, May Only after Sept

2004

N/A

BMP Audacious 2005, Oct Yes Req. ntfsprogs NTFS-3G 2006, Jul - N/A

OpenWRT FreeWRT 2006, May Only after Oct

2006

N/A

QtiPlot SciDavis 2007, Aug - N/A Kamailio OpenSIPS 2008, Aug Yes Yes Blastwave.org OpenCSW 2008, Aug - N/A jMonkeyEngine Ardor3D 2008, Sept Yes, but scarce N/A Frog CMS Wolf CMS 2009, Jul - N/A Aldrin Neil 2009, ? - N/A Ffmpeg libav 2011, Mar Yes Yes

called a dynamic graph. In this phase, we processed inter- actions data to form a communication graph of the commu- nity. We were looking for how people interacted with each other. We decided to treat the general mailing list as a per- son, because the bulk of the communication was targeted at it, and most newcomers start by sending their questions to the general mailing list email address for maximum ex- posure. Each communication effort was captured with a time-stamp. This allowed us to form a dynamic graph, in which the nodes would exist if and only if they had an in- teraction with another node during the time period we were interested in.

5.3 Phase 3: Temporal Visualization and Tem- poral Social Network Analysis

In this phase, we wanted to analyze the changes that hap-

pen to the community over a given period of time, e.g. three months before and three months after the year in which the forking event happened. For our first study, we measured betweenness centrality [6] of the most significant nodes in the graph, and the graph diameter over time. Figures [6][7][8] show the betweenness centralities over the 1.5 year period for the Kamailio, Amarok and Asterisk projects respectively. To do temporal analysis, we had two options; 1) look at snapshots of the network state over time, (e.g. to look at the network snapshots in every week, the same way that a video is composed of many consecutive frames), and 2) look at a period through a time window. We preferred the sec- ond approach, and looked through a time window of three months wide with 1.5 month overlaps. To create the visu- alizations, we used a 3 months time frame that progressed six days a frame. In this way, we had a relatively smooth transition.

There are many ways of looking at an individual’s impor- tance. One is called closeness centrality. The farness of a node is defined as the sum of its distances to all other nodes.

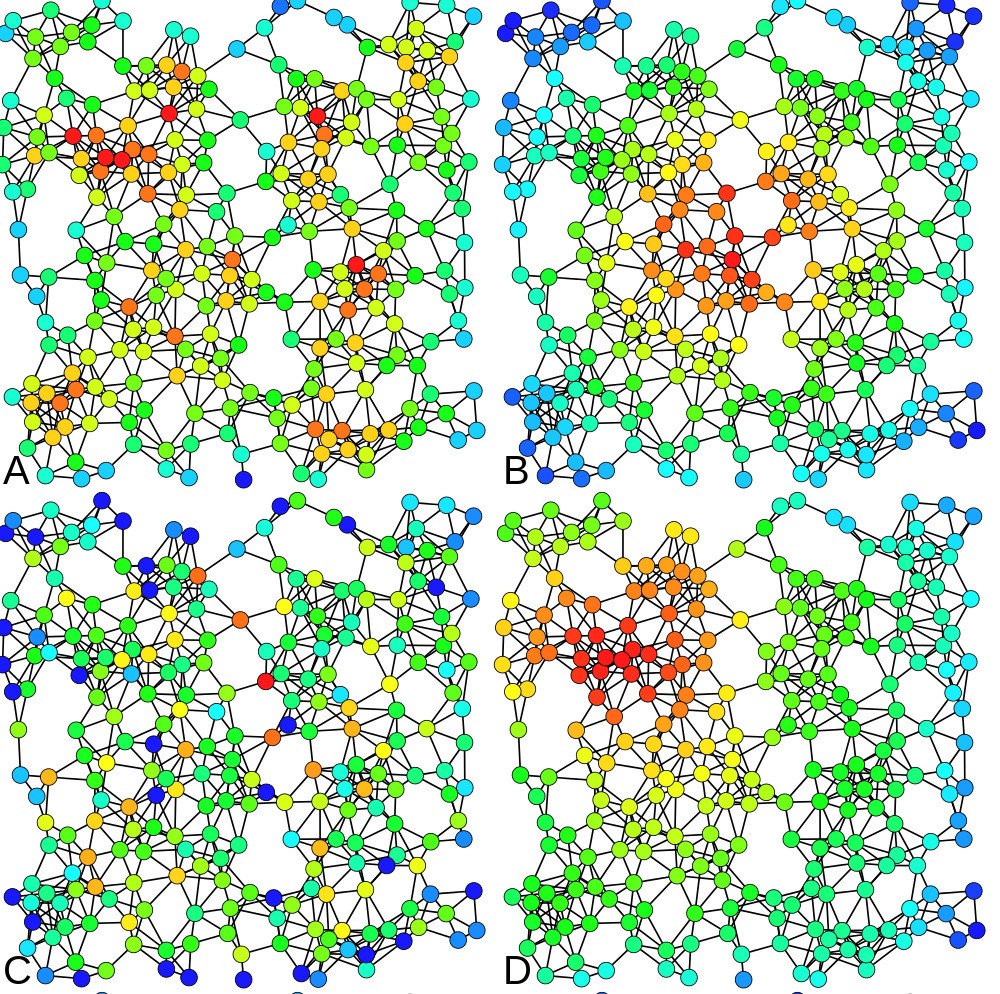


Figure 2: Heat-map color-coded examples of nodes with high centrality metric are shown above. The same network is analysed four times with the following centrality measures: A) Degree centrality, B) Closeness centrality, C) Between- ness centrality and D) Eigenvector centrality [31]

Table 5: Projects forked because of the need for more community-driven development by Robles and Gonzalez-Barahona [30]

sorted in chronological order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Original | Forked | Date | Data available? | Collected? |
| Nethack | Slash’EM | 1996, ? | - | N/A |
| GCC | EGCS | 1997, ? | - | N/A |
| SourceForge | Savane | 2001, Oct | - | N/A |
| PHPNuke | PostNuke | 2001, Sum | Not found | N/A |
| QTExtended | OPIE | 2002, May | - | N/A |
| GraphicsMagick | Graphics | 2002, Nov | Only after 2003 | N/A |
| freeglut | OpenGLUT | 2004, Mar | Yes | Yes |
| Mambo | Joomla! | 2005, Aug | - | N/A |
| SER | Kamailio | 2005, Jun | Only after 2006 | N/A |
| PHPNuke | RavenNuke | 2005, Nov | Not found | N/A |
| Hula | Bongo | 2006, Dec | - | N/A |
| Compiere | A Dempiere | 2006, Sept | No Dev. mailing  list | N/A |
| Compiz | Beryl | 2006, Sept | Only after Jun  2007 | N/A |
| SQL-Ledger | LedgerSMB | 2006, Sept | No Dev. mailing  list | N/A |
| Asterisk | Callweaver | 2007, Jun | Yes | Yes |
| CodeIgniter | KohanaPHP | 2007, May | Not found | N/A |
| OpenOffice.org | Go-oo.org | 2007, Oct | Only after Jun  2011 | N/A |
| Mambo | MiaCMS | 2008, May | - | N/A |
| TORCS | Speed Dreams | 2008, Nov | Yes, but scarce | N/A |
| MySQL | MariaDB | 2009, Jan | Yes | No |
| Nagios | Icinga | 2009, May | Yes | Req |
| Project Darkstar | RedDwarf | 2010, Feb | Yes, but scarce | N/A |
| SysCP | Froxlor | 2010, Feb | - | N/A |
| Dokeos | Chamilo | 2010, Jan | Not found | N/A |
| GNU Zebra | Quagga | 2010, Jul | - | N/A |
| rdesktop | FreeRDP | 2010, Mar | Yes | Yes |
| OpenOffice.org | LibreOffice | 2010, Sept | Only after Jun  2011 | N/A |

Redmine ChiliProject 2011, Feb Yes, but scarce N/A

Table 6: Forked projects for which collaboration data was collected

|  |  |  |
| --- | --- | --- |
| Pro jects | Reason for forking | Year |
| Kamailio & OpenSIPS | Differences among developer team | 2008 |
| ffmpeg & libav | Differences among developer team | 2011 |
| Asterisk & Callweaver | More community-driven development | 2007 |
| rdesktop & FreeRDP | More community-driven development | 2010 |
| freeglut & OpenGLUT | More community-driven development | 2004 |
| Amarok & Clementine Player | Technical (Addition of functionality) | 2010 |
| ApacheCouchDB & BigCouch | Technical (Addition of functionality) | 2010 |
| Pidgin & Carrier | Technical (Addition of functionality) | 2008 |

The closeness of a node is defined as the inverse of the far- ness. More informally, the more central a node is the lower its total distance to all other nodes. Closeness centrality can be used as a measure of how fast information will spread through the network [7]. Secondly, if we are looking for peo- ple who can serve as bridges between two distinct communi- ties, we could measure the node’s betweenness centrality. Be- tweenness centralities for mediators who act as intermediate entities between other nodes are higher [7]. Third, if cross- community collaboration is the focus, we can measure edge betweenness centrality. Edges connecting nodes from differ-

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Amarok | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ApacheCouchDB | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Nodes & Edges vs. Time

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pidgin | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

ent communities have higher edge centrality values. In the

Sep Nov Jan Mar Apr Jun Aug Oct Dec Feb Apr Jun

community collaboration graph, edge betweenness or stress of an edge is the number of these shortest paths that the edge belongs to, considering all shortest paths between all pairs of nodes in the graph. Fourth, one can claim that certain peo- ple in the community are more important than others, and

1.00

0.75

0.50

Nodes and Edges

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Asterisk | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Freeglut | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rdesktop | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

whoever is close to them, is relatively more important than

others. In graph terms, this is measured by eigenvector cen- trality, which is based on the assumption that connections to high-profile nodes contribute more to the importance of

1.00

0.75

Feb Apr Jun Aug Nov Jan Mar May Nov Jan Mar May

1.00

0.75

a node. Google’s PageRank link-analysis algorithm [29] is a variant of the eigenvector centrality measure. In short, cen- trality measures have been used in several studies to identify key player in a community.

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kamailio | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ffmpeg | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Mar May Jul Sep Nov Oct Dec Feb Apr Jun

Time

**variable**

N E

In addition to the centrality measures, we planned to look into the resilience of the community as well. By resilience, we mean how well the network holds its structure and form when some parts of it are deleted, added, or changed. For a graph, the resilience of a graph is a measure of its robustness to node or edge failures. This could occur for instance when an influential member of the community leaves. Many real- world graphs are resilient to random failures but vulnerable to targeted attacks. Resilience can be related to the graph diameter : a graph whose diameter does not increase much on node or edge removal has higher resilience [7].

1.00

Figure 3: Nodes and Edges over Time

Diameter vs Time

0.75

5.4 Phase 4: Temporal Visualization

Several visualization techniques and tools are used in the field of social network analysis, for instance, Gephi [3], which is a FLOSS tool for exploring and manipulating networks. It is capable of handling large networks with more than 20,000 nodes and features several SNA algorithms. It is customiz- able with plugins and we used it for dynamic network visu- alization. We visualized the dynamic network changes us- ing Gephi [3]. The videos show how the community graph is structured, using a continuous force-directed linear-linear model, in which the nodes are positioned near or far from each other proportional to the graph distance between them.

This results in a graph shape between between Fru¨chterman

& Rheingold’s [12] layout and Noack’s LinLog [24].

6. RESULTS AND DISCUSSION

6.1 Kamailio Project

Figure 5 shows four key frames from the Kamailio project’s social graph around the time of their fork (the events de- scribed here are easier to fully grasp by watching the video. A node’s size in a proportional to the number of interactions

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Amarok | | | | | | | | | | ApacheCouchDB | | | | | | | | | | Pidgin | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1.00

0.75

0.50

Diameter

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Asterisk | | | | | | | | | | Freeglut | | | | | | | | | | Rdesktop | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1.00

0.75

0.50

0.25

0.00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kamailio | | | | | | | | | | | Ffmpeg | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Sep Nov Jan Mar Apr Jun Aug Oct Dec Feb Apr Jun

Feb Apr Jun Aug Nov Jan Mar May Nov Jan Mar May

**Fork** Normal Fork

Mar May Jul Sep Nov Oct Dec Feb Apr Jun

Time

Figure 4: Diameter changes over Time

the node (contributor) has had within the study period and the position and edges of the nodes change if they had inter- actions within the time window shown, with six day steps per frame. The 1 minute and 37 seconds video shows the life of the Kamailio project between October 2007, and March

2009. Nodes are colored based on the modularity of the net- work.

The community starts with the GeneralList as the the biggest node, and four larger core contributors and three lesser size core contributors. The big red-colored node’s transitions are hard to miss, as this major contributor departs from the core to the periphery of the network (Video minute 1:02) and then leaves the community (Video minute 1:24) captur- ing either a conflict or retirement. This corresponds to the personal difference category of forking reasons.

Figure 6 shows the betweenness centrality of the major con- tributors of Kamailio project over the same time period. The horizontal axis marks the dates, (each mark represents a 3-month time window with 1.5 months overlap). The ver- tical axis shows the percentage of the top betweenness cen- tralities for each node. The saliency of the GeneralList – colored as light blue – is apparent due to to its continuous and dominant presence in the stacked area chart. The chart legend lists the contributors based on the color and in the same order of appearance on the chart starting from the bottom. One can easily see that around the ”Aug. 15, 2008

- Nov. 15, 2008” tick mark on the horizontal axis, several contributors’ betweenness centralities shrink to almost zero and disappear. This helps identify the date of fork with a month accuracy. The network diameter of the Kamailio project over the same time period is also shown in Figure

6. The increase in the network diameter during this period

The video for the Asterisk project is also available online, and the results from our quantitative analysis of the be- tweenness centralities and the network diameters are shown in Figure 8. The results show that the network diameter remained steady at 6 throughout the period. The Aster- isk community was by far the most crowded project, with

932 nodes and 4282 edges. The stacked area chart shows the distribution of centralities, where we see an 80%-20% distribution (i.e. 80% or more of the activity is attributed to six major players, with the rest of the community ac- counting for only 20%). This is evident in the video repre- sentation as well, as the top-level structure of the network holds throughout the time period. The results from the vi- sual and quantitative analysis links the Asterisk fork to the more community-driven category of forking reasons.

7. TIMELINE

Table 7 outlines my research timeline. It also shows the mapping between research questions and the phases of the study.

Table 7 Timeline

Spring 2012 • Literature review

Fall 2012 • Literature review & data collection

Fall 2013 • Data cleaning and wrangling

Winter 2014 • Creating communication graphs

Spring 2014 • Temporal visualization and temporal SNA

Fall 2014 • Statistical analysis RQ1-5

confirms the findings of Hannemann and Klamma [14].

This technique can be used to identify the people involved in conflict and the date the fork happened with a months

Spring 2015 • Statistical analysis and

planning

Summer 2015 • Interview and user study design

RQ1-5

accuracy, even if the rival project does not emerge immedi-

Fall 2015 • Interviews (including pilots) RQ2

ately.

6.2 Amarok Project

The video for the Amarok project fork is available online1 , and the results from our quantitative analysis of the be- tweenness centralities and the network diameters are shown

Winter 2016 • User studies (including pilots)

Spring 2016 • Writing

June 2016 • Defense

8. CONCLUSION

RQ3

in Figure 7. The results show that the network diameter

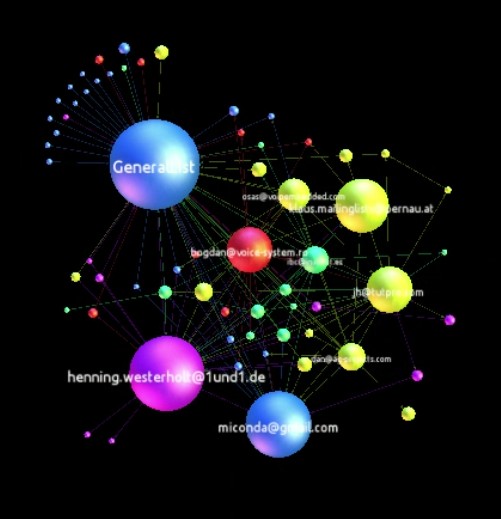
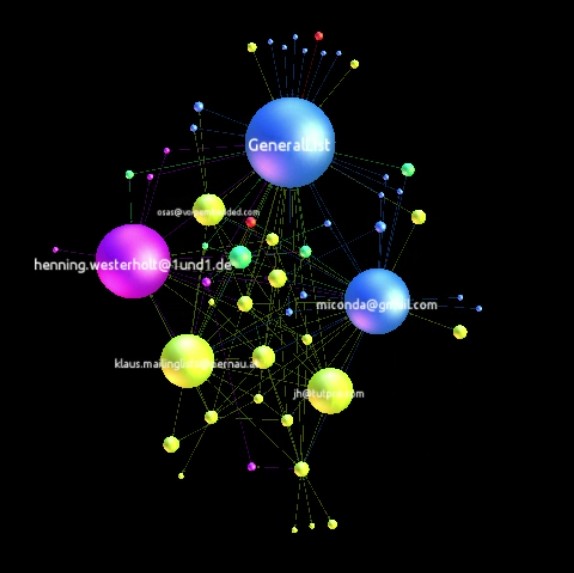
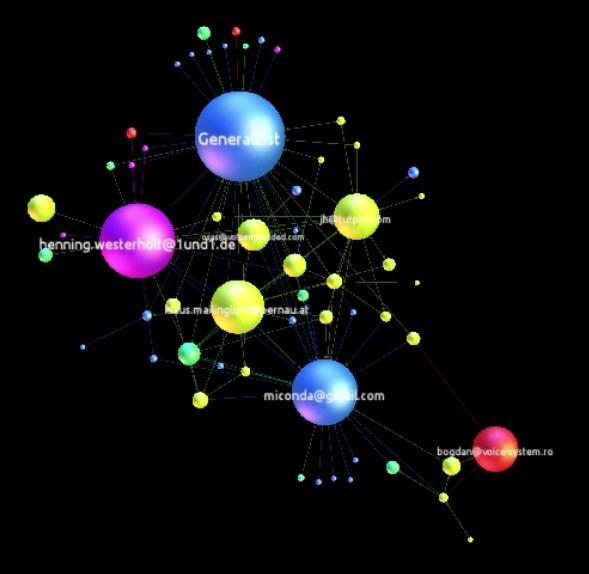
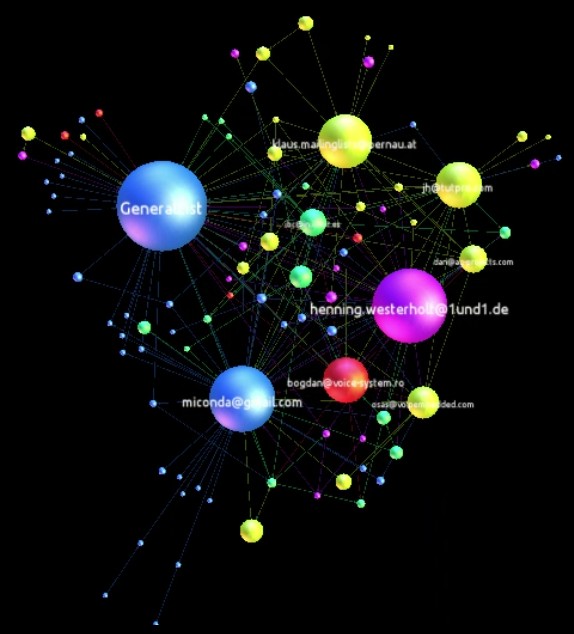
has not increased over the period of the fork, which shows a resilient network. The video shows the dynamic changes in the network structure, again typical of a healthy network, rather than of simmering conflict. These indicators show that Amarok fork in 2010 arguably belongs to the “addition of technical functionality” rationale for forking, as there are no visible social conflict.

6.3 Asterisk Project

1 Video visualizations available at [http://eecs.oregonstate.edu/˜azarbaam/OSS2014/](http://eecs.oregonstate.edu/~azarbaam/OSS2014/)

We studied the collaboration networks of three FOSS projects using a combination of temporal visualization and quantita- tive analysis. We based our study on two papers by Robles and Gonzalez-Barahona [30] and Hannemann and Klamma [14], and identified three projects that had forked in the recent past. We mined the collaboration data, formed dy- namic collaboration graphs, and measured social network analysis metrics over an 18-month period time window.

We also visualized the dynamic graph (available online) and as stacked area charts over time. The visualizations and the quantitative results showed the differences among the projects in the three forking reasons of personal differences



(a) (b) (c) (d)

Figure 5: Snapshots from video visualization of Kamailio’s graph (Oct. 2007 - Mar. 2009) in which a core contributor (colored red) moves to the periphery and eventually departs the community.

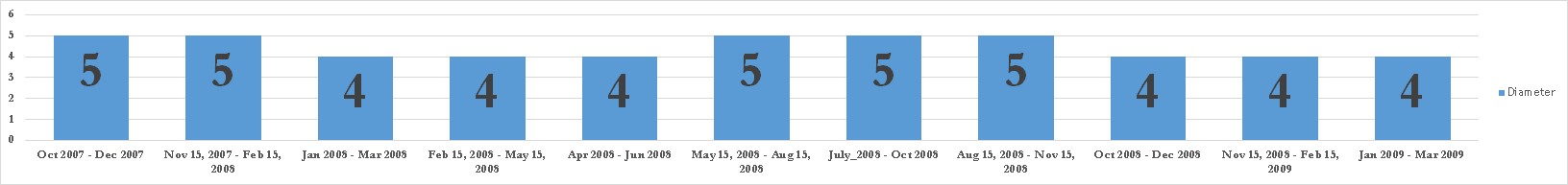
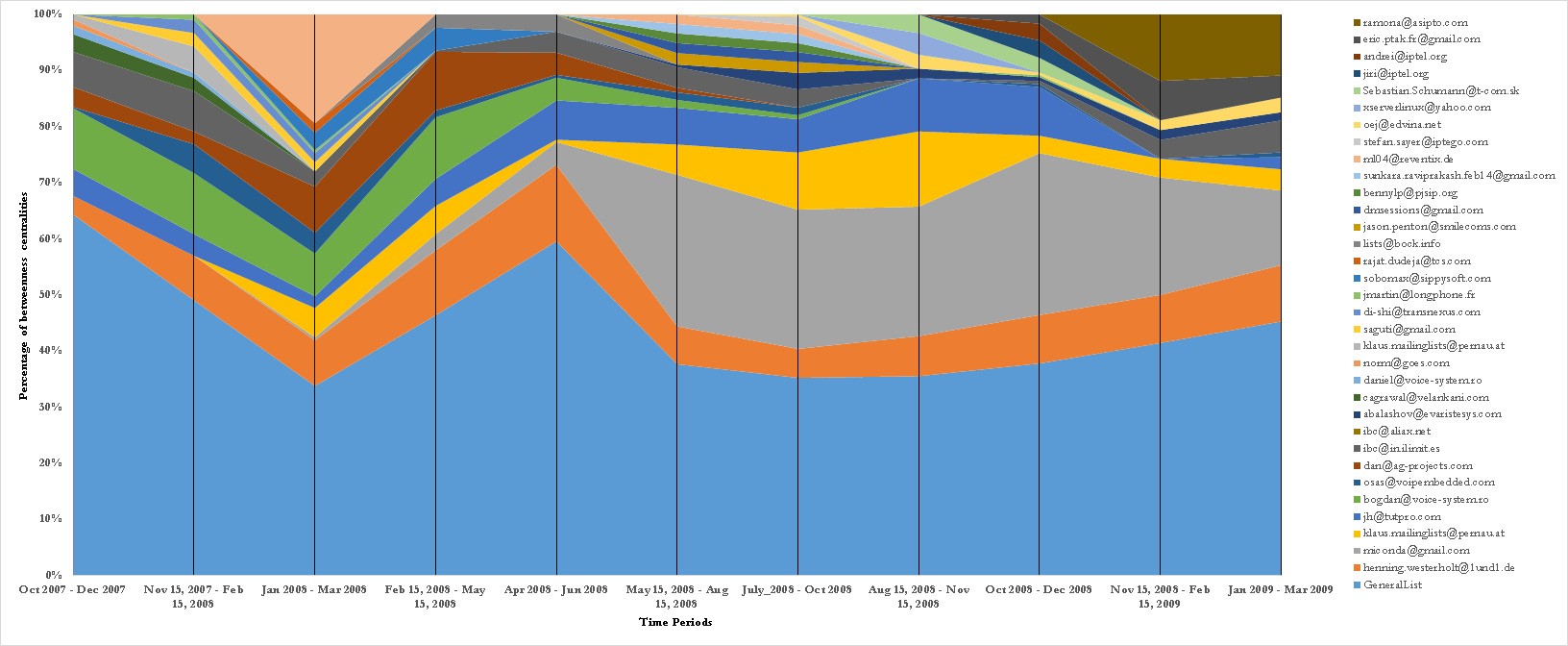


Figure 6: Kamailio top contributors’ betweenness centralities and network diameter over time (Oct. 2007 to Mar. 2009) in

3-month time windows with 1.5-month overlaps

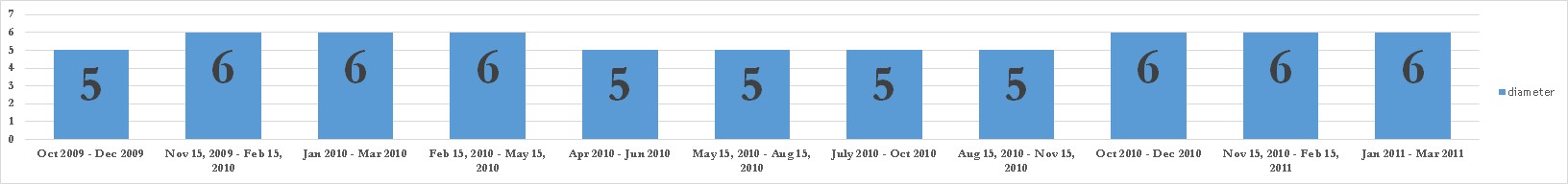
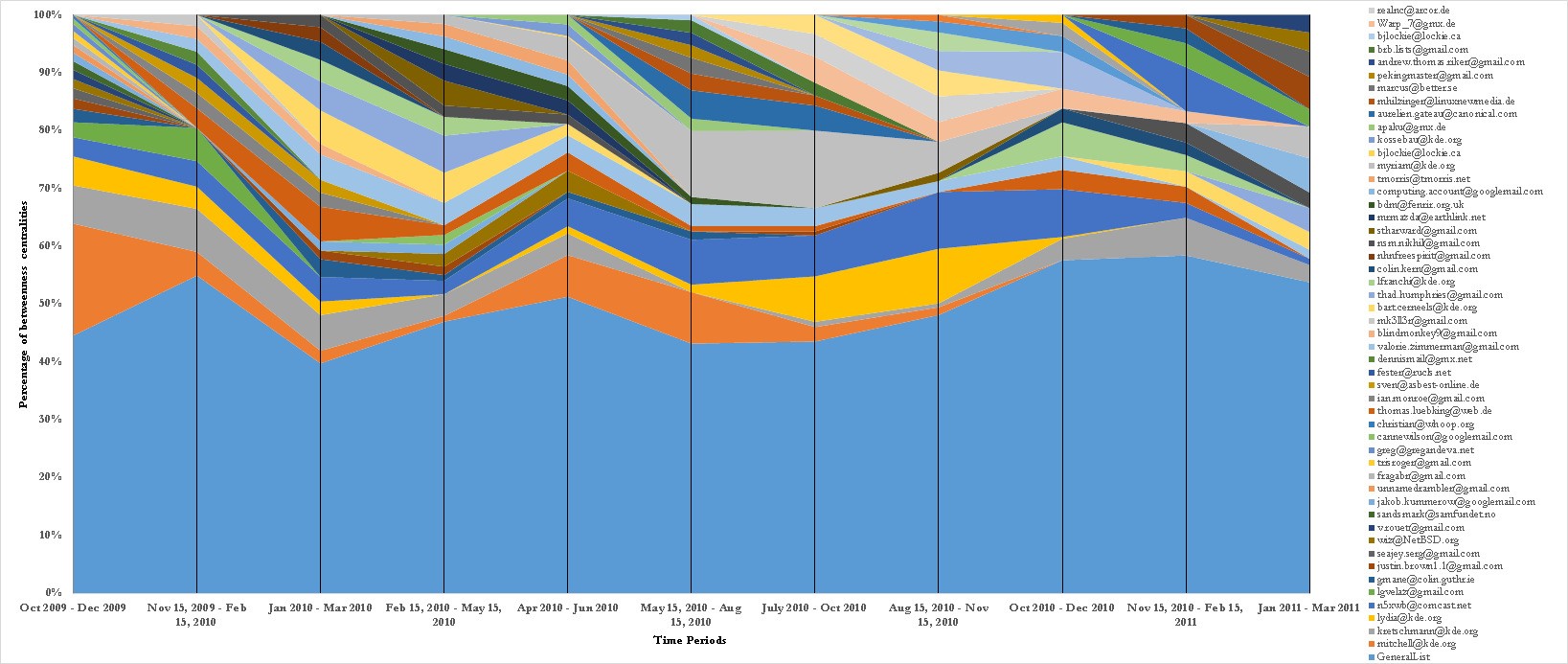


Figure 7: Amarok project’s top contributors’ betweenness centralities and network diameter over time between Oct. 2009 to

Mar. 2011 in 3-months time windows with 1.5 months overlaps

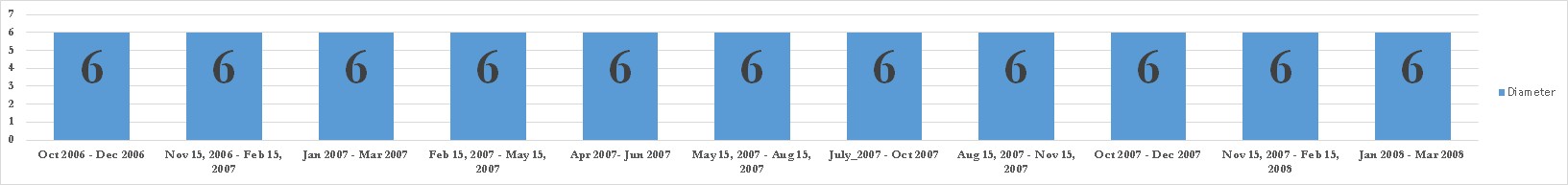
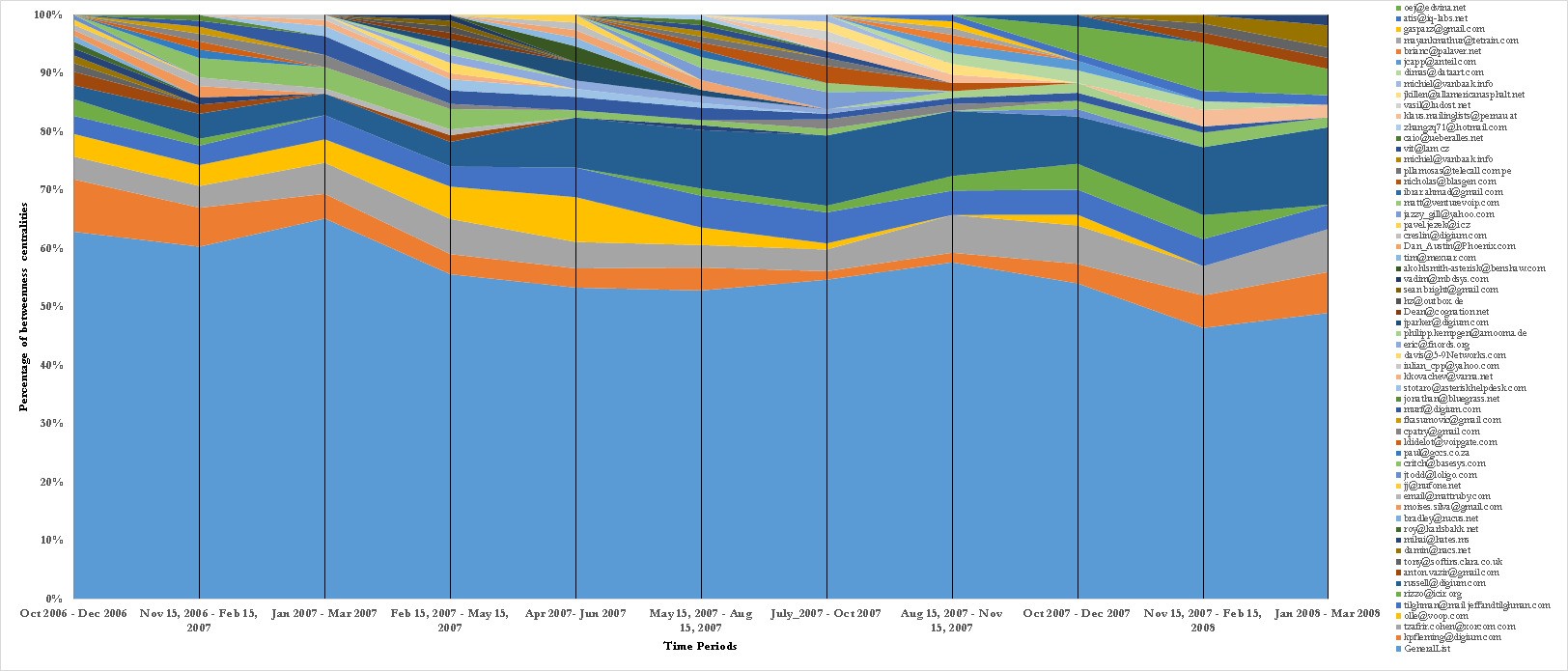


Figure 8: Asterisk project’s top contributors’ betweenness centralities and network diameter over time between Oct. 2009 to

Mar. 2011 in 3-months time windows with 1.5 months overlaps

and reveal information that cannot be seen otherwise.

|  |  |  |
| --- | --- | --- |
| among the developer teams, technical differences (addition of new functionality) and more community-driven develop- ment. The personal differences representative project was | [6] | Symposium on Foundations of software engineering, New York, NY, USA: ACM, pp. 24-35, 2008. Brandes, U. “A Faster Algorithm for Betweenness |
| identifiable, and so was the date it forked, with a month |  | Centrality ”, in Journal of Mathematical Sociology |
| accuracy. The novelty of the approach was in applying the |  | 25(2):163-177, 2001. |
| temporal analysis rather than static analysis, and in the tem- | [7] | Chakrabarti, D., and C. Faloutsos. “Graph mining: |
| poral visualization of community structure. We showed that |  | Laws, generators, and algorithms,” ACM Computing |
| this approach shed light on the structure of these projects |  | Surveys, 38, 1, Article 2, 2006. |

9. THREATS TO VALIDITY

The presented findings may not be generalized to all OSS projects. The projects studies in this paper were selected from a pool of candidate projects, partly because data about them was available. Given access, a better sampling ap- proach has to be adopted, which could result in a more robust investigation. Furthermore, the proposed technique uses the data from online communications. The assump- tion that all the communication can be captured by mining repositories is intuitively imperfect, but inevitable. Hence, to minimize the effect of this assumption, we plan to comple- ment the quantitative approach with a qualitative approach of interviewing key individuals from the community as fu- ture work.

Acknowledgement

The author would like to thank his academic adviser, Prof. Carlos Jensen, and his committee members, Prof. Margaret Burnett, Prof. Ronald Metoyer, and Prof. Christopher Scaf- fidi for their help and guidance throughout the author’s PhD program. The author would also like to thank the open source developers of the projects studied for making their data available, without which this study would not have been possible.

10. REFERENCES

[1] Asur, S., S. Parthasarathy, and D. Ucar, (2009), “An

event-based framework for characterizing the evolutionary behavior of interaction graphs,” in ACM Trans. Knowledge Discovery Data. 3, 4, Article 16, (November 2009), 36 pages. 2009.

[2] Baishakhi R., C. Wiley, and M. Kim, “REPERTOIRE: a cross-system porting analysis tool for forked software projects,” in Proc. of the ACM SIGSOFT 20th Int. Sym. on the Foundations of Software Engineering (FSE ’12). ACM, New York, NY, USA, Article 8 , 2012.

[3] Bastian, M., S. Heymann, and M. Jacomy, “Gephi: an open source software for exploring and manipulating networks,” presented at the Int. AAAI Conf. on Weblogs and Social Media, 2009.

[8] Crowston, K., K. Wei, J. Howison, and A. Wiggins. “Free/Libre open-source software development: What we know and what we do not know,” ACM Comput. Surv. 44, 2, Article 7, 2012.

[9] Ernst, N., S. Easterbrook, and J. Mylopoulos, “Code forking in open-source software: a requirements perspective,” arXiv preprint arXiv:1004.2889, 2010.

[10] Ford, L. R. and D. R. Folkerson, “A simple algorithm for finding maximal network flows and an application to the Hitchcock problem,” Canadian Journal of Mathematics, vol. 9, pp. 210-218, 1957.

[11] Forrest, D., C. Jensen, N. Mohan, and J. Davidson, “Exploring the Role of Outside Organizations in Free/ Open Source Software Projects,” in Proceedings of the

8th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2012, Hammamet, Tunisia, September 10-13, 2012.

[12] Fruchterman, T. M. J. and E. M. Reingold, “Graph drawing by force-directed placement,” Softw: Pract. Exper., vol. 21, no. 11, pp. 1129-1164, 1991.

[13] Guzzi, A., A. Bacchelli, M. Lanza, M. Pinzger, and A. van Deursen. “Communication in open source

software development mailing lists,” in Proc. of the

10th Working Conference on Mining Software Repositories (MSR ’13). IEEE Press, Piscataway, NJ, USA, 277-286. 2013.

[14] Hannemann, A and , R. Klamma “Community Dynamics in Open Source Software Projects: Aging and Social Reshaping,” in Proc. of the IFIP Second Int. Conf. on Open Source Systems, pp. 80-96, 2013.

[15] Howison, J. and K. Crowston. “The perils and pitfalls of mining SourceForge,” In Proceedings of the Int. Workshop on Mining Software Repositories (MSR

2004), pp. 7-11. 2004.

[16] Howison, J., K. Inoue, and K. Crowston, “Social dynamics of free and open source team communications,” in Proc. of the IFIP Second Int. Conf. on Open Source Systems, 319-330, 2006.

[17] Howison, J., M. Conklin, and K. Crowston, “FLOSSmole: A collaborative repository for FLOSS research data and analyses,” Int. Journal of Information Technology and Web Engineering, 1(3),

17-26. 2006.

[18] Kuechler, V., C. Gilbertson, and C. Jensen, “Gender Differences in Early Free and Open Source Software Joining Process,” Open Source Systems: Long-Term Sustainability, 2012.

[4] Bezrukova, K

C. S. Spell, J. L. Perry, “Violent Splits

[19] Kunegis, J., S. Sizov, F. Schwagereit, and D. Fay,

Or Healthy

Div

” ides? Coping With Injustice Through

“Diversity dynamics in online networks,” in Proc. of

Faultlines,” Personnel Psychology, Vol 63, Issue 3.

2010.

[5] Bird, C., D. Pattison, R. D’Souza, V. Filkov, and P.

Devanbu, “Latent social structure in open source projects,” in Proc. of the 16th ACM SIGSOFT Int.

the 23rd ACM Conf. on Hypertext and Social Media,

USA, 2012.

[20] Leskovec, J., Kleinberg, J., and Faloutsos, C.: “Graphs over time: densification laws, shrinking diameters and possible explanations,” in Proc. of the

SIGKDD Int. Conf. on Knowledge Discovery and data Mining, 2005.

[21] Leskovec, J., K. J. Lang, A. Dasgupta, and M. W.

Mahoney, “Statistical properties of community structure in large social and information networks,” in Proc. of the 17th Int. Conf. on World Wide Web (WWW ’08), ACM, 2008.

[22] Nakakoji, K., Y. Yamamoto, Y. Nishinaka, K.

Kishida, and Y. Ye. “Evolution patterns of

open-source software systems and communities,” in Proc. of the International Workshop on Principles of Software Evolution (IWPSE ’02). ACM, New York, NY, USA, 76-85. 2002.

[23] Mikkonen, T., L. Nyman, “To Fork or Not to Fork: Fork Motivations in SourceForge Projects,” in Int. J. Open Source Softw. Process. 3, 3. July, 2011.

[24] Noack, A., “Energy models for graph clustering,” J.

Graph Algorithms Appl., vol. 11, no. 2, pp. 453-480,

2007.

[25] Nyman, L. , “Understanding code forking in open source software,” in Proc. of the 7th Int. Conf. on Open Source Systems Doctoral Consortium, Salvador, Brazil, 2011.

[26] Nyman, L., T. Mikkonen, J. Lindman, and M.

Foug`ere, “Forking: the invisible hand of sustainability in open source software,” in Proc. of SOS 2011: Towards Sustainable Open Source, 2011.

[27] Nyman, L., “Hackers on Forking,” in Proc. of Int.

Sym. on Open Collaboration, OpenSym, 2014. [28] Oh, W., Jeon, S., “Membership Dynamics and

Network Stability in the Open-Source Community:

The Ising Perspective ” in Proceedings of the Twenty-Fifth International Conference on Information Systems. 2004.

[29] Page, B, B. Sergey, R. Motwani and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Technical Report, Stanford InfoLab, 1999.

[30] Robles, G. and J. M. Gonzalez-Barahona, “A comprehensive study of software forks: Dates, reasons and outcomes,” in Proc. of the 8th Int. Conf. on Open Source Systems, Hammamet, Tunisia, 2012.

[31] Rocchini, C. (Nov. 27 2012), Wikimedia Commons, Available: <http://en.wikipedia.org/wiki/File:Centrality.svg>,

2012.

[32] Singer, L., F. Figueira Filho, B. Cleary, C. Treude, M.

Storey, and K. Schneider. “Mutual assessment in the social programmer ecosystem: an empirical investigation of developer profile aggregators,” in Proc. of the 2013 conference on Computer supported

cooperative work (CSCW ’13). ACM, New York, NY, USA, 103-116. 2013.

[33] Sowe, S., L. Stamelos, and L. Angelis, “Identifying knowledge brokers that yield software engineering knowledge in OSS projects,” Information and Software Technology, vol. 48, pp. 1025-1033, Nov 2006.

[34] Storey, M., L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky, “The (R) Evolution of social media in software engineering,” in Proc. of the on Future of Software Engineering (FOSE 2014). ACM, New York, NY, USA, 100-116. 2014.

[35] Syeed, M. M., “Socio-Technical Dependencies in

Forked OSS Projects: Evidence from the BSD Family,” Journal of Software 9.11 (2014): 2895-2909.

2014.

[36] Teixeira, J., and T. Lin, “Collaboration in the

open-source arena: the webkit case,” in Proc. of the

52nd ACM conference on Computers and people research (SIGSIM-CPR ’14). ACM, New York, NY, USA, 2014.

[37] Torres, M. R. M., S. L. Toral, M. Perales, and F.

Barrero, “Analysis of the Core Team Role in Open Source Communities,” in Complex, Intelligent and Software Intensive Systems (CISIS), 2011 Int. Conf. on, pp. 109-114. IEEE, 2011.

[38] Zachary, W., “An information flow model for conflict and fission in small groups,” Journal of Anthropological Research, vol. 33, no. 4, pp. 452-473,

1977.