
Koper-Capodistria, Slovenia, 25 June 2013
Proceedings of the Doctoral Consortium at the 9th
International Conference on Open Source Systems,
Koper-Capodistria, Slovenia, 2013

Edited by:

Klaas-Jan Stol
Lero—The Irish Software Engineering Research Centre
University of Limerick, Ireland

Björn Lundell
University of Skövde, Sweden

Gregory R. Madey
University of Notre Dame, IN, USA
Preface

Open Source Software (OSS) remains to be an active research area and the OSS conference as its premier publication venue has reached its ninth edition this year.

To facilitate new researchers with an arena to present and receive feedback on their research, the OSS conference has had a Doctoral Consortium for several years. The principle objective of the consortium is to provide doctoral students the opportunity to present their research at various stages of production – from early drafts of their research design to near completion of their dissertation – in a forum where they can get critical and helpful feedback from a community of interested scholars and other students as they work to finish their degree.

This volume contains the five papers, each of which was reviewed by members of the program committee. After the reviews, authors were given the opportunity to revise their paper based on the input they received from the reviewers. This volume contains the revised versions of the papers, which were presented at the Doctoral Consortium at the Ninth International Conference on Open Source Systems, in Koper-Capodistria, Slovenia in June 2013. The papers provide a look into the research-in-progress by Ph.D. students who study a variety of aspects of OSS.

We wish to thank the members of the Program Committee of the Doctoral Consortium who have provided valuable feedback on the papers. We also thank all Ph.D. students for their participation.

Klaas-Jan Stol
Björn Lundell
Gregory Madey
Program Committee

Cornelia Boldyreff
Andrea Capiluppi
U. Yeliz Eseryel
Joseph Feller
Patrick Finnegan
Jesús M. Gonzalez-Barahona
Imed Hammouda
Chris Jensen
Stefan Koch
Lorraine Morgan
Gregorio Robles
Barbara Russo
Walt Scacchi
Maha Shaikh
Giancarlo Succi
Tony Wasserman

University of East London, UK
Brunel University, UK
University of Groningen, Netherlands
University College Cork, Ireland
University of New South Wales, Australia
Universidad Rey Juan Carlos, Spain
Tampere University of Technology, Finland
Google, USA
Bogazici University, Turkey
Lero, NUI Galway, Ireland
Universidad Rey Juan Carlos, Spain
Free University of Bolzano-Bozen, Italy
University of California, Irvine, USA
University of Warwick, UK
Free University of Bolzano-Bozen, Italy
Carnegie Mellon Silicon Valley, USA
# Table of Contents

Analyzing FOSS Collaboration and Social Dynamics with Temporal Social Networks ................................................................. 1  
*Amir Azarbakht and Carlos Jensen*

Professionalization Dynamics in a Network of an Open Source Software Project ................................................................. 15  
*Clément Bert-Erboul*

Open-Source Platform-based Strategies: Multiple Case Studies in the Mobile Devices Industry ................................................. 27  
*Jose Teixeira*

QA Practices and FLOSS Communities ................................................................................................................................. 37  
*Adina Barham*

The role of Free/Libre Open Source Software User’s Face-to-Face Meeting: A Case Study of Drupal User’s Local Meetup ................. 57  
*Eunyoung Moon*
Analyzing FOSS Collaboration and Social Dynamics with Temporal Social Networks

Amir Azarbakht, Carlos Jensen

Oregon State University, School of Electrical Engineering & Computer Science
1148 Kelley Engineering Center, Corvallis OR 97331, USA
{azarbaam,cjensen}@eecs.oregonstate.edu
http://eecs.oregonstate.edu/~azarbaam
http://eecs.oregonstate.edu/people/jensen

Abstract. How can we understand FOSS collaboration better? Can social issues that emerge be identified and addressed before it is too late? Can the community heal itself, become more transparent and inclusive, and promote diversity? We propose a technique to address these issues by quantitative analysis of social dynamics in FOSS communities. We propose using social network analysis metrics to identify growth patterns and unhealthy dynamics; giving the community a heads-up when they can still take action to ensure the sustainability of the project.

Keywords. FOSS, Social Dynamics, Temporal Analysis, Free/Open Source Software, FLOSS, Forking, Social Network Analysis

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. Free/Open Source Software (FOSS) development efforts are prime examples of how community can be leveraged in software development, as efforts are formed around communities of interest, and depend on continued interest and involvement in order to stay alive [Nyman 2011].

Though the bulk of collaboration and communication in FOSS communities occurs online and is publicly accessible, there are many open questions about the social dynamics in FOSS communities. Projects might go through a metamorphosis when faced with an influx of new developers or the involvement of an outside organization. Conflicts between developers raised as the result of divergent opinions about the future of the project might lead to an ensuing fork of project and the 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 [Bezrukova et al. 2010].

Most recent studies of FOSS communities have tended to suffer from two important limitations. First, they treat community as a static structure rather than a dynamic
process. Second, many social dynamics in FOSS have been studied using a case-study methodology, focusing on a selected subset of the available data.

In this paper, we propose to use social network analysis to study the evolution and social dynamics of FOSS communities. With these techniques we hope to identify measures associated with unhealthy group dynamics, e.g. a simmering conflict, as well as early indicators of major events in the lifespan of an online community. One dynamic we are especially interested in are those of forked FOSS projects. We will seek to validate this technique by comparing the results of our analysis to the results of a study of forked FOSS projects by [Robles and Gonzalez-Barahona 2012]. The goal is to demonstrate that this quantitative approach can be applied to commonly available FOSS archives to get a better understanding of the evolution of these communities.

This paper is organized as follows: We present related literature on online social literature, and what further study needs to be done. Next, we discuss why the issue needs to be addressed and who benefits from it, in the motivation section. Following that, we present three research questions framed as hypotheses that we are going to test. After that, we propose a methodology as to how to test the validity of the hypotheses, which includes gathering data, doing the analysis, and the visualization of the findings. At the end, we present future work and challenges.

2 Related Work

The social structures of FOSS communities have been studied extensively over the past decade. Researchers have studied the social structure and dynamics of team communications [Howison et al. 2006, Bird et al. 2008], identifying knowledge brokers and their associated activities in FOSS projects [Sowe et al. 2006], their sustainability [Nyman 2011], FOSS forking [Nyman 2011, Robles and Gonzalez-Barahona 2012], their topology [Bird et al. 2008], their demographic diversity [Kunegis et al. 2012], gender differences in the process of joining them [Kuechler et al. 2012] and the role of the core team in their communities [Torres et al. 2011]. They have tended to look at community as a static structure rather than a dynamic process. This makes it hard to determine cause and effect, or the exact impact of social changes.

The study of communities has grown in popularity in part thanks to advances in social network analysis. From the earliest works on studying information flow and predicting conflict and fission in groups by Zachary [Zachary 1977], to the more recent works of [Leskovec et al.] on the statistical properties of community structure in social networks, there is a growing body of quantitative research on online communities.

The earliest works on communities was done with a focus on information diffusion in a community [Zachary 1977]. 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 [Ford and Fulkerson 1957]
on the group’s communication graph; “the unequal flow of sentiments across the ties” and discriminatory sharing of information lead to “subcommunities with more internal stability than the community as a whole.”

Community splits in FOSS projects 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 development based on the source code basis of the project.” Robles and Gonzalez-Barahona [Robles and Gonzalez-Barahona 2012] identified 220 significant FOSS projects that have forked over the past 30 years, and compiled a comprehensive list of the dates and reasons for forking. They classified these into six main categories. Table 3. shows their results, discussed further in the next sections, and which we build on extensively in this work. They identified a gap in the literature in case of “how the community moves when a fork occurs”.

The dynamic behavior of a network and identifying key events was the aim of a study by [Asur et al. 2009]. They studied three DBLP co-authorship networks and defined the evolution 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 consecutive snapshots, based on the size and overlap of every pair of groups. Then, they labeled groups with events, and used these identified events to calculate the metrics in Table 1 for the nodes and the network.

### Table 1. The measures of diversity used by [Asur et al. 2009]

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Tendency of a node to have interactions with the same nodes over time</td>
</tr>
<tr>
<td>Sociability</td>
<td>Tendency of a node to have different interactions</td>
</tr>
<tr>
<td>Influence</td>
<td>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)</td>
</tr>
<tr>
<td>Popularity</td>
<td>Number of nodes in a cluster (how crowded a sub-community is)</td>
</tr>
</tbody>
</table>

The communication patterns of FOSS developers in a bug repository were examined by [Howison et al. 2006]. They calculated 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 uncommon,” 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 examine, and our research questions.

The tension between diversity and homogeneity in a community was studied by Kunegis et al. [Kunegis et al. 2012]. Table 2 lists five network statistics they 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 networks matured over their lifespan. Kunegis et al. [Kunegis et al. 2012] argued that one possible reason could be that the community structure consolidates as projects mature.

<table>
<thead>
<tr>
<th>Network property</th>
<th>Network is diverse when</th>
<th>A network is diverse when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paths between nodes</td>
<td>Paths are long</td>
<td>Effective diameter</td>
</tr>
<tr>
<td>Degrees of nodes</td>
<td>Degrees are equal</td>
<td>Gini coefficient of the degree distribution</td>
</tr>
<tr>
<td>Communities</td>
<td>Communities have similar sizes</td>
<td>Fractional rank of the adjacency matrix</td>
</tr>
<tr>
<td>Random walks</td>
<td>Random walks have high probability of return</td>
<td>Weighted spectral distribution</td>
</tr>
<tr>
<td>Control of nodes</td>
<td>Nodes are hard to control</td>
<td>Number of driver nodes</td>
</tr>
</tbody>
</table>

To date, most studies on FOSS have only been carried out on a small number of projects, and using snapshots in time. To our knowledge, no study has been done of project forking that has taken into account the temporal dimension.

3 Motivation

To better understand and measure the evolution and social dynamics of FOSS projects, 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 community leaders make informed decisions about possible changes or interventions. We want to map the dynamics of communities to real world phenomena. Identification is the first step to rectify an undesired dynamic before the damage is done. A community that does not manage growing pains may end up stagnating or dissolving.

Managing growing pains is especially important in case of FOSS, where near half the project contributors are volunteers [Forrest et al. 2012]. [Oh et al. 2004] have argued that openness in FOSS is “[…] generally perceived as having a positive connotation, however, the term can also be interpreted 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 community can result in a community’s susceptibility and vulnerability to herded exits by its participants. Commercial vendor intervention, an alternative project becoming available, and licensing issues can result in some original core members ceasing to provide their loyal service for the community, which can prompt their coworkers to leave as well” [Oh et al. 2004].

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

4 Research Questions

We argue that the social interactions data reflects the changes the community goes through, and will be able to describe the context surrounding a forking event. Robles and Gonzalez-Barahona [Robles and Gonzalez-Barahona 2012] classify the main reasons for forking into six classes, listed in Table 3.

Table 3. The main reasons for forking as classified by [Robles and Gonzalez-Barahona 2012]

<table>
<thead>
<tr>
<th>Reason for forking</th>
<th>Example forks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical (Addition of functionality)</td>
<td>Xpdf &amp; Poppler</td>
</tr>
<tr>
<td>More community-driven development</td>
<td>EGCS &amp; GCC</td>
</tr>
<tr>
<td>Discontinuation of the original project</td>
<td>Apache web server</td>
</tr>
<tr>
<td>Commercial strategy forks</td>
<td>LibreOffice &amp; OpenOffice.org</td>
</tr>
<tr>
<td>Legal issues</td>
<td>X.Org &amp; XFree</td>
</tr>
<tr>
<td>Differences among developer team</td>
<td>OpenBSD &amp; NetBSD</td>
</tr>
</tbody>
</table>

Three of the six listed reasons are socially related, and so should arguably be reflected somehow in the social interaction data. As an example, if a fork occurs because of a desire for “more community-driven development”, we expect to see an interaction patterns in the collaboration data showing a strongly-connected core community getting more isolated from the rest of the community, and the formation of another very active and well-connected core. More specifically, our research hypotheses are:

Hypothesis #1: Social interactions data can be used to predict an imminent fork.

Hypothesis #2: “Personal differences” and “technical differences” are associated with distinctly different interaction patterns. It is possible to tell the difference between when the reason for forking was personal vs. technical.

Hypothesis #3: It is possible to tell through collaboration patterns between when the reason for forking is “more community-driven development” and “differences among the developer team”.
5 Proposed Methodology

Determining how FLOSS networks evolve over time requires us to adopt a case study methodology. To this end, we propose to examine several FLOSS projects. This will involve obtaining mailing list archives, bug repositories, Git logs, creating the collaboration graphs, applying social network analysis technique and measuring SNA metrics, and visualizing the evolving graphs. We plan to have five phases as described in the following:

5.1 Phase 1: Data Mining

We plan to study six projects that have forked and six projects that have not forked as a control group. Data collection involves mining mailing list archives, bug repositories, and Git logs. [Howison and Crowston 2004] described their experiences of “spidering, parsing and analysis of SourceForge data.” Building on their research, we will carefully avoid the pitfalls prone to skew the study. We plan to contribute to and use FLOSSmole [Howison et al. 2006] resources, which provides scripts for FLOSS research data and analyses.

5.2 Phase 2: Creating Communication Graphs

Many social structures can be represented as graphs. The nodes represent actors/players and the edges between them represent the interaction among the actors. Such graphs can be a snapshot of the network, which forms a static graph, or a longitudinally changing network, also called a dynamic graph.

In this phase, we process the data to form a communication graph representing the community. This is fraught with complications. Collaboration graph for a large community, e.g. Linux Kernel with 9515 contributors will be a complex network of 9515 nodes and millions of edges. Special graph sampling algorithms are needed to use clustering techniques and social network analysis tools on such a large and complex network. These graph-sampling techniques have been studied and compared by [Wang et al. 2011] and [Lu et al. 2011]. [Chi et al. 2007] e.g. applied a temporal smoothness to their spectral clustering algorithm, and argued that incorporating this method results in a more robust clustering results, while it provides less sensitivity to short-term noise and is “adaptive to long-term clustering drift.”

5.3 Phase 3: Temporal Evolution Analysis

In this phase, we want to analyze the changes that happen to the community over a given period of time, e.g. two years before and two years after the forking event. To this end, we are interested in seeing the changes in the roles individuals have, any shifts in the center of gravity of the project, unusual dilution or concentration of the part of
the community and how information can diffuse along the community. These changes can translate into questions, e.g. is everyone still connected to everyone like before, or there are bridges burnt down. The roles individuals take in the community are in par with the individual’s importance or centrality in the network. We can measure each individual’s importance with a social network analysis measure called centrality, i.e. who is more important/central than the others? e.g. who has the most number of friends in a community? (degree centrality)

There are many ways of looking at an individual’s importance. This is measured by a centrality called closeness centrality. The farness of a node is defined as the sum of its distances to all other nodes. The closeness of a node is defined as the inverse of the farness. 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 [Chakrabarty and Faloutsos 2006]. Secondly, if we are looking for people who can serve as bridges between two distinct communities, we could measure the node’s betweenness centrality. Betweenness centralities for mediators who act as intermediate entities between other nodes are higher [Chakrabarty and Faloutsos 2006]. Third, if cross-community collaboration is the focus, we can measure edge betweenness centrality. Edges connecting nodes from different communities have higher edge centrality values. In the 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 people in the community are more important than others, and whoever is close to them, is relatively more important than others. In graph terms, this is measured by eigenvector centrality, which is based on the assumption that connections to high-profile nodes contribute more to the importance of a node. Google’s PageRank link-analysis algorithm by [Brin and Page 1998] is a variant of the eigenvector centrality measure. In short, centrality measures have been used in several studies to identify key player in a community.

In addition to the centrality measures, we plan 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 [Chakrabarty and Faloutsos 2006].

We plan to measure these metrics defined on the communication graph, as well as clustering coefficient of node, degree distribution of the entire network, and the graph’s diameter, whose definitions are out of the scope of this paper.

The significance of SNA measures on a single collaboration graph is defined as compared to a random graph, e.g. an Erdős-Rényi random graph of the same size [Erdős and Rényi 1959], to examine their relative significance. We plan to do this analysis over
We are aware that while datasets consisting of huge real-world graphs are obtainable, their sizes may range from thousands nodes and several million edges. While conventional algorithms to compute graph metrics (shortest paths, centrality, betweenness, etc.) exist, many of these algorithms become impractical for large graphs. Leskovec [Leskovec et al. 2008] argues that one can use sampling algorithms – but that is a non-trivial problem since the goal is to maintain structural properties of the network.

### 5.4 Phase 4: Temporal Visualization

Several visualization techniques and tools are used in the field of social network analysis, for instance Gephi [Bastian et al. 2009], NetLogo [Wilensky 1999], igraph [Csardi and Nepusz 2006], NetworkX [Hagberg et al. 2008], SoNIA [Bender-deMoll and McFarland 2006], NodeXL [Smith et al. 2009]. Gephi, 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 customizable with plugins and can be used for dynamic network visualization.
5.5 Phase 5: Interviewing & Triangulation

To complement the quantitative techniques, we plan to interview key players identified in the community and the situation studied to gain their perspective on the community as well. This will be useful in cross-validating the results, as the part of communication that was not logged online could be taken into account in this way. Furthermore, we will compare our findings to those of [Robles and Gonzalez-Barahona 2012]. This should allow us to determine how well the SNA techniques work to identify the underlying reasons for a fork.

5.6 Challenges

The proposed technique uses the data from online communications. The assumption that all the communication can be captured by mining repositories is intuitively imperfect, but inevitable. Hence, to minimize the effect of this assumption, we complement the quantitative approach with a qualitative approach of interviewing key individuals from the community. This will help triangulate the results.
6 Preliminary Results

Some initial work is done, namely scraping mailing list data off several FOSS projects, as well as their bug repositories. We have also done a literature review of the existing work on social network analysis on FOSS communities, and have experienced working with large graphs. There are a myriad of possible applications for the possible outcomes of this line of research, and the gap in the existing body of research shows a need for further study of FOSS social network analysis.

7 Acknowledgments

We would like to thank the peers at the Human-Computer Interaction Lab at Oregon State University, with special thanks to Victor Kuechler and Jennifer L. Davidson for their feedback.

References

Analyzing FOSS Collaboration and Social Dynamics with Temporal Social Networks


(This page intentionally left blank)
From Community to Collective: the History of an Open Source Software Project

Clément Bert-Erboul
PhD student in Sociology at Clersé Laboratory CNRS UMR 8019 Lille University, France clement.bert-erboul@ed.univ-lille1.fr

Abstract. In our work, we focus on the correlation between some changes in contribution rules in a free software project and changes in the structure of the contributors’ network. By studying the archives of a mailing list in the Videolan project, we show how a community of engineering students has turned into a digital collective of voluntary contributors working in IT companies. The software development by students is characterized by a high turnover rate of contributors, and a rampant development in a small clique living in the same place. After, the network structure has changed with the arrival of worldwide IT professionals and more stable core contributors.

Keyword: social network, open source community, diffusion list, digital collective.

1 Introduction

We analyze the emancipation of an open source developers group', in relation to its birth context¹. At the beginning the organization was led by members of personal networks and looked like a community. During this period some students of Ecole Centrale de Paris (ECP) developed and wrote pieces of the personal and experimental source code. Thereafter the group was composed by people working in the IT industry and are now linked through a more formal, stable and hierarchic organization. The rules of contributions are passed from the commitment of engineers student to the participation of professional developers based on professional standards. These transformations exhibit that the relations between various actors are recomposed during key moments in the life of the digital collective (Kilamo 2011, Bert-Erboul, 2012). During these periods, some norms and motivations of commitment (Lakhani and Wolf, 2003) are redefined, and determine whether the activity of the group is extended or the contributors stop it.

By taking as an example the case of the nonprofit association VideoLAN, we study the changes in a group of contributors producing, via the Internet, an open software able to broadcast and play multimedia content (containing video and / or

¹ I thank Catherine Darville for her review.

sound). After the opening of the source code in 2001 the Videolan software becomes popular as it has been downloaded for free more than a billion times. Such popularity reveals the quasi-industrial distribution activity conducted by the free software movement (Weber, 2004).

Sharing the source code has been a practice in existence since the beginning of the computer industry (Armer, 1980). In the 1950s, the writing and the use of software programs required varied technical skills often regrouped within a team of developers. Under these circumstances the flow of source code allows the coordination of programmers working on similar software. The exchange of the source code between software developers was formalized in the late 1980s by free licenses (Stallman et al., 2011). The use of free licenses ensures that developers are allowed to copy, study, modify and redistribute the software and forbids the individual ownership of the work collectively done. These trading rules are largely maintained through the hobbyist activities outside the economic system mainly based on the closing software source code. Computer clubs (Levy, 1984) have played an important role in the software economy by institutionalizing and legitimizing the open source code production. VideoLAN developers follow this tradition by producing an open source software massively used by individuals and businesses on the Internet.

By a historical study of the nonprofit organization and an analysis of the relationships generated by the exchange of messages between 2001 and 2011 on a mailing list, we show some changes in the network linking the contributors. At first we present elements to define the term “community” and “social network” in Open Source Project. Afterwards we present the methodology to analyze the set of emails exchanged on a mailing list. In a third step, we show that the network of VLC developers was firstly a set of personal networks of students. Later this group became a bigger social network with multimedia professional engineers linked in a hierarchical structure.

2 Communities and Social Network in an open source project

Some social changes in the Videolan contributors’ network between 2001 and 2011 show the commitments (Becker, 1960) and the withdrawals (Fillieule, 2005) of the open source project developers. These back and forth successions of contributors are studied by some sociologists of the social movements. To answer Olson’s paradox (Olson, 1965), the researchers analyze the influence of the context to understand why people participate to a collective action (Tilly, 1978) without a direct reward. In the VLC case, series of test/ errors between participants are close to the thesis of the actor-network theory. Michel Callon in his research on the scientific and economic systems reveals how profit sharing, the mechanisms of enlistment and betrayal, leads the actors to enter or leave the collective action (Callon, 1986). With the ethnographic approach
of the actor-network theory we study the changes in the social-technical groups, and we analyze if some heterogeneous contributors define a coherent set of behaviors.

2.1 The Videolan community

The unit, the longevity and growth of VideoLAN are symptomatic of this coherence. Indeed, the Videolan collective had never forked. To explain the success and continuation of these remote organizations, social sciences are observing the governance arrangements of these organizations (O'mahony and Ferraro, 2007). The strict technical conventions governing the writing of the source code imply the existence of a strong social control among the programmers. Sociological and psychological literacy uses the term “community” (Proulx and Latzko-Toth, 2005) to describe the remote groups organizing on the Internet. In the context of the Ecole Centrale de Paris, the term "community" (Rheingold, 1993) describes quite well the VideoLan activities. The students were a small population composed of individuals, attached to each other by strong institutional and emotional links. The system of solidarity in the building of the ECP imposes on individuals a daily social control. For these students, to quit the Videolan collective and the ECP would be catastrophic for their personal identity. However, the relevance of the community term is limited to the ECP.

2.2 The Videolan collective

The social network analysis on the Internet showed transformation in the relationship in a group are built by different type of ties (Wellman, 1979) such as friendship, family, neighborhood or work for example. Our study shows that the group of developers was more or less at the beginning a set of personal network of students in an engineering school. These developments were crowned by institutional prize2. After, the digital collective became a more structured and hierarchical social network with a large and fickleness set of contributors from different backgrounds. Hence the term "virtual community" seems wrong to account for the type of relationship caused by the participation in software development. Indeed, within the sociological sense of community (Tönnies, 1957), leaving the group is socially hard, because the outgoing people are losing a part of their identities.

The core contributors of VideoLAN have already been studied. Thomas Basset (Basset, 2005) produced a monograph on this group of contributors to the ECP between 2001 and 2003. The research highlights the paradox existing between the use of free licenses and the difficulty for information to flow in a group confined in a boarding school. In this study, the author presents as a collegial social structure the

---

2 The IBM Linux Challenge in 2001, the Apple Design Award in 2003.
VideoLAN organization (Waters, 1989). In this network a strong technical component centralizes the expertise around a small clique of participants. The class mates turnover tends to reinforce the specialization of some contributors, who each year develop new features. Our study completes this previous research. Since 2006, the VideoLAN project has no longer been hosted by the Ecole Centrale de Paris3, and contributors from all backgrounds have contributed in the project. The maintenance of the activity is accompanied by a partial rewriting of the source code in accordance with professional technical conventions. The annual renewal of contributors within the ECP had created inconsistencies in the source code. Indeed the successive experiments had rarely been updated by the following generations of students. The redesign of the code by professionals follows the technical and legal agreements allowing contributors from the multimedia professions to make comments or contributions without additional cognitive costs. Finally the Videolan collective has gone through various steps in its organization. At the beginning the developers were student and they constructed a sort of community but it was recomposed each year. The activity was popular all around the world. Then the group became an international digital collective of people who work in the IT industry. With a social network analysis we will show that the rules of participation are not the same in both organizations.

3 Methodology

To analyze the commitment in the collective, we decided to study the network of exchanged messages in a mailing list and interpret these results. To check the quantitative data we use the qualitative knowledge (Seaman 1999) of the Videolan collective, built on interviews and observations during development days between 2010 and 2012. Having presented the method of data extraction, we will clarify the meaning given to the degree of centrality obtained by our protocol.

3.1 Data extraction

We study the exchange of messages between February 2001 and December 2011 in cohorts of 12 months4. These messages were exchanged by the participants of the list "VlcDevel" dedicated to the development of the media player software VLC, the most famous project of the Videolan collective. The messages on the list concern the use of the software, source code changes, or more rarely the distribution of rights between contributors. The archive retrieved is public5, and represents 90,860 messages.

3 The growth of VLC user's number increases the technical bandwidth costs.
4 Except for the first year which count 11 months.
5 http://mailman.videolan.org/pipermail/vlc-devel/
exchanged between 2,944 different email addresses. The subscription to the list is open and free. To compose the corpus each year, we take into account all the individuals who have posted at least one message on the list. Every January 1st we restart our counting.

The use of mailing lists creates a relationship between the subscribers, based on semi-private asynchronous exchanges. By sharing the contents the contributors provide a reusable cohort of information (Gensollen, 2004) and enable the improvement of information. The subscribers to the list do not exclude each other in a discussion. The presence of a contributor does not reduce the possibility to another contributor to read the message at the same time. On the list, messages written in English are unmoderated and the objects of threads concern few subscribers, although all subscribers can read and participate in the discussion.

3.2 Study of the centrality of degree

After downloading the online archive we converting it to a matrix format to reconstruct the networks of the participants in the annual mailing list6. To study the interactions in the discussions we established the relationships between the participants with different topics of discussion (threads). From this bimodal network (a two mode network), contributor / thread / contributor, we decided to connect with each other all participants involved in the same thread in a unimodal undirected (a one mode network) network contributor/contributor. The centrality of degree (Freeman, 1979) obtained represents the number of contacts an individual contributor is connected with. A contact is labeled by an email address and represents a human being or bots. If no one responds to the message of an individual, the centrality of the participant will be 0. If a contributor is connected to another contributor, its centrality increases by 1 for each new contributor contacted. If the contributor responds to its own message (loop), its centrality increases once by 2 contacts.

This quantification of centrality in a sociotechnical network enables to taking into account the relationships which are often ignored between machines and human beings. Indeed the mailing list contains a lot of computer-generated emails. By sending a large number of messages, the bots participate in the complexity of the network. However, the bots depend on the human beings activity. So we did not want to give them a disproportionate importance. Despite the large amount of messages they distribute, the bots have a degree of centrality of 2. The human beings do not return directly to these subjects which are automatically generated. These threads are updates that follow other updates or series of error messages. Thus, the network nodes represent human actors (n = 2941) and bots (n = 3).

6 I thank Daniel Caillibaud to have written the program of data preparation
To study and quantify the commitment in a mailing list, we decided that the messages had equivalent values. We did not detail the multiplexity of relationships. Thus, a “thank you” weight as much as a longer message, a bug report has the same weight as a bug fix. If all the messages are equal, all the contributors have also the same weight. A member of the association is as important as a non official member. To preserve the centrality and hierarchy of the contributors we refined email addresses (Bird et al., 2006) as much as could. We merged the various email addresses of the same contributor when they were readable. Thus, counting all the contacts we preserve the qualitative nature of relationship of a contributor. A contributor will be more central if he is in contact with different people with whom he will discuss various topics.

Our analysis focuses on the degree of centrality distribution in the group, that is to say the number of different contacts between peers which exists in the collective. This diversity reflects a part of the collective’s activity. The changes in the number of contacts between the contributors have concomitant variations with changes in the organization of the collective. We chose as changing period, the departure of the Videolan developers from the Ecole Centrale de Paris in 2006. After this period the technical rules governing the digital collective are renegotiated by the contributors. To observe the changes, we analyze the levels of commitment in the mailing list participants. By classifying the contributors' degree of centrality each year we study how the most active contributors are organizing.

4 An OSS social network

By studying the distribution of centrality in the group of participants to the list of VLC developers, we find three groups of contributors often mentioned in the literature (Crowston and Howison, 2005) and by the developers themselves. In the center, there are the most active contributors. Around this core there are less regular contributors. In the periphery we find the sophisticated users of the software. Our longitudinal study shows that the distribution of the contributors within these three groups varies in parallel with the changes in the collective. The rewriting of some part of the source code (since the departure of the group of ECP) has seemed to have encouraged the dialogue between the contributors by increasing the number of contacts within the active contributors. In addition, the transition from a student activity to an activity of IT professionals tends to increase the difference between the most central actors in the network and the least connected one. During all the observed period the less linked periphery represents the largest share of the population present on the list.

7 Some contributors encrypt their address, others use symbols not supported by the archiving system.
4.1 The center of the network

In the center of the group, we distinguish a core of contributors with a large number of contacts. In our study we put in this class the contributors having between 7 and 90 contacts. The whole gap of this interval is due to the growth of the developers group. This evolution shows that contributes in an emerging open source project is different from contributes in an international one. This large scale of magnitude indicates the steps of the project at different maturity stages. This core of contributors represents between 8% and 13% of the workforce and centralizes 40 to 50% of existing relationships in the network each year. Over time the core's members have changed. However, the staff turnover in this category slows sharply after the departure from ECP. Between 2007 and 2011, 15 contributors remained active in the community, while during the first five years (2001-2005) only six contributors were present for several years on the list, and among them, only three were active each year. The regular change of contributors during the ECP in the VideoLAN collective is related to the successive class mates. This generational rotation underscores the student dimension of the software related to the training of young engineers.

The rotation of participants is not related to a mathematical phenomenon of exclusion of one group by another. The number of subscribers is theoretically unlimited. The increase in the number of contacts does not question the centrality of others. Instead, the arrival of new contributors tends to increase the centrality of the remaining individuals. The commitment and withdrawal of the contributors in the collective indicates that they are not driven by a common purpose or by a systematic search of prestige. When the VideoLAN project was hosted by the Ecole Centrale Paris (2001 to 2006) there was no significant variation between the participants with little or no contact and the contributors with a broad range of contacts. Between 2001 and 2006, the contacts standard deviation was stable at 3.5. However, the differences in levels of commitment of the most central contributors has increased from 2007, and the deviation from the average has fluctuated between 4.5 and 7.5 after the departure of the ECP group.

4.2 Active contributors

Around the core's contributors, we defined a group of participants whose members had between 3 and 6 contacts. The group was stable over time and had about 10% of the workforce during the ECP period and 12% in the next period. Between 2001 and 2006, this intermediate category represented 20% of the relationships of the network. From 2007, the proportion of existing relationships in the network held by the middle circle has been between 25 and 30%. The increased involvement of the members corresponds to the period of the source code formatting with professional standards (Fig. 1 and 2). The technical standardization is linked with the increase of contributors'
discussions built on common conventions (Favereau, Biencourt, et al. 2002) of computer programming. Thus, the transformations of the commitment device give rise to remarks from diverse stakeholders.

**Figure 1.** Network 2006 VLCdevel contributors to the list with a score of degree centrality of at least 3 (N = 57 or 13% of the population in 2006).

**Figure 2.** 2009 Network contributors VLCdevel the list with a score of degree centrality of at least 3 (N = 95 or 17% of the population in 2009)
4.3 A wide fringe

In the network, the majority of contributors in the mailing list are not highly interconnected. Each year, more than 80% of the workforce in the list has had between 0 and 2 contacts. When the software was hosted at ECP this population represented about 35% of network relations. This proportion decreased due to the arrival of IT workers in the digital collective. We see that users have has few or no contact but they have concentrated 25% of network relations.

The low integration in the network is partly due to the fickleness of the contributors. Between 2001 and 2011 about 50% of the email addresses indexed had no contact with other participants of the list. In other words, each year half of the participants list received no response to their message, and did not participate to another subject. This lack of relationship has different explanations. The original messages may call no response. Other threads can be opened in parallel on the same issue. Sometimes, the contributor doesn't know any answer on the proposed topic.

This part of the network with a high proportion of solitary individuals, kept away open source developers from a "community" definition. In fact, a community in the sociological sense is not "disconnected" from half of its members. This device with solitary people is important for the collective insofar as they relay bugs, ask questions on the use of the software and materialize the success of the software on the Internet. This continuous flow of information about the VLC software produced by heterogeneous actors gives a panoptic vision about the software, promoting the establishment of a shared and coherent representation of the contribution activity. In the digital collective the independence of the individuals is also expressed by the large number of egocentric messages (loops discussion).

Contributors update threads wherein they have already spoken. Each year, more than 90% of messages link the author with himself. This emphasizes both the involvement of contributors and the strong specialization in the software project. The original message can be left to the review of others subscribers of the list. The contribution is updated thereafter by successive improved versions. Loops show that the mediated and asynchronous public dialogue is a paradoxical way to talk with itself. This allows a glimpse of the relationship between individualistic practices and collective action on the Internet.

5 Conclusion and Discussion

The network of open source software developers is built by three groups well identified by the scientific literacy: core, active, fringe. The fringe contributors’ area seems to be attracted by the advanced functionalities of the software without a dialogue with the developers to improve the source code. In the two other groups, the
contributors have established conventions marked by a technical language included in the source code. In these three groups we analyze the changes of the organization of the social network. Firstly the group of developers was composed by some personal networks of students living in the same boarding school. This group composed a kind of community renewed each year. Afterwards developers were regrouped in an international social network of IT workers. During this second period, the developers formed a digital collective with a hierarchical structure and shared standards.

Our study shows that the mode of organization of a group depends on the context. The identification of changes in the collective's network by stages of maturity highlights a methodological point. Indeed, if groups of open source developers groups change how can they be compared? We believe it is necessary to make a story of each group to evaluate the heuristic scope of comparison. Our text shows that a network of community and a network of digital collective have not the same structural network characteristics.

Bibliography

(This page intentionally left blank)
Open-platforms: Multiple case-studies in the mobile devices industry

Jose Teixeira
University of Turku
Turku Centre for Computer Science
EIT ICT labs
Jose.Teixeira@utu.fi
WWW home page: http://www.jteixeira.eu/

Abstract. This doctoral student research addresses the emergence of open-source based platforms currently attaining notorious relevance within the mobile devices industry. It aims at providing a better understanding on “how and why corporations follow open-source based platform strategies?” within high-networked markets under constant technological evolution. Drawn over the constructs of multi-sided platforms (Rochet & Tirole, 2003; Hagiu & Wright, 2011) and plenty of studies on open-source software adoption such as Lerner and Tirole (2002), Feller and Fitzgerald (2002) and Weber (2004), this research also bridges business strategy in network economics (Gawer & Henderson, 2007; Gawer & Cusumano, 2008) with the psychological and anthropological concepts of sense of community (SoC) and communities of practice (CoP). The researcher follows a theory testing approach, where established theories developed on previous decades over the Personal Computer and Operating Systems are replicated in a completely new and emergent context. By taking a pure qualitative case-study approach on the molds of Eisenhardt (1989) and Yin (2008), the researcher developed a set of descriptive and in-depth case-studies within the mobile devices industry, answering the lack of empirical research on strategies used by open-source platform providers. Natural occurring data was collected from Internet sites and semi-structured interviews were conducted within an industrial key-player with the goal of providing a framework and a theory that would extend what is known on open-source products to open-source platforms. Practitioners are then provided with better guidance for designing and implementing their networked technological strategies, to let their computer-based platforms succeed in high competitive under network effects.

Keywords. OSS, FLOSS, Open-source, Platforms, Technological-strategy

1 Introduction

The research author seeks to complement the existing body of theoretical knowledge on open-source software, drawn mostly from research on open-source software products, by pointing his lenses to software platforms.

With a purposive-focus on the telecommunications industry, the researcher studies the emergence of mobile open-source platforms such as Maemo, Meego, Android and Bada empowering big corporations such as Google, Nokia, Intel and Samsung within the high-competitive mobile devices market.

Computer-based platforms combine core components with complementary products and services habitually made by a variety of external entities (Gawer & Cusumano 2008). In the case of high-technology and high-competitive markets, many firms follow a platform-based strategies due to the impossibility of satisfying by themselves an exceedingly complex consumer group (Hagiu 2004).

2 Research questions

The aim of this research study is to investigate the strategies used by open-source platform providers. Its core objective is to provide a better understanding on how companies like Apple, Google and Nokia integrate open-source software technological components under public domain into their platforms empowering proprietary mobile devices. A set of research questions include:

- RQ1: **What** are the open-source software technological **components** integrated by the mobile devices vendors?
- RQ2: **How** open-source software **affects** the competitive dynamics of the mobile devices industry?
- RQ3: **How** rival platform-vendors **collaborate** in the open-source arena?
- RQ4: **Why** companies follow open-source platform-based strategies?

The research questions follow and What, How and Why logic. The researcher strategically addressed first the What and How questions, without rushing to answer to the final Why question (RQ4).

3 Literature review

This research reviews multi-disciplinary literature, mostly within the information systems, economics and strategy research streams. The research author pursued to
cover the current body of theoretical knowledge addressing the open-source software (OSS) phenomena, as addressed by Lerner and Tirole (2002), Feller and Fitzgerald (2002) and Weber (2004); research addressing computer-based platforms and multi-sided platforms (Rochet & Tirole, 2003; Hagiu, 2004; Hagiu & Wright, 2011); and research bridging business strategy and network economics (Gawer & Henderson, 2007; Gawer & Cusumano, 2008).

Regarding the review of OSS literature, a crucial part of this research, the author identified several recent and comprehensive literature reviews addressing the OSS phenomena (Stol & Babar 2009; Aksulu & Wade 2010; Hauge et al. 2010; Lindman 2011).

A notable systematic literature review by Stol and Babar (2009), covering specialized conference proceedings, pointed out the heterogeneity and lack of empirical studies dealing with OSS in organizations. Another systematic literature review from Hauge et al. (2010), confirmed the heterogeneity in which companies approach OSS prompting lack of empirical research on OSS adoption in organizations.

Lindman (2011) extended the two previous mentioned reviews, in yet another systematic literature review covering both open-source journals (Hauge et al. 2010) and top Information Systems journals (Rainer & Miller 2005). He argues that current body of knowledge emphasis on OSS community-driven development with limited interest on OSS within organizations.

Last but not the least, yet another literature review by Aksulu and Wade (2010) paused and reflected on the state of open-source research; analyzed and categorized a wide-set of open-source research; and proposed a framework to situate OSS research within a wider nomological network while proposing future directions for open-source research.

Besides the existence of the previously reported recent, comprehensive and systematic literature reviews on OSS, and in order to review relevant theoretical knowledge on other relevant constructs such as computer-based platforms and multi-sided platforms (Rochet & Tirole, 2003; Hagiu, 2004; Hagiu & Wright, 2011), business strategy and network economics (Gawer & Henderson, 2007; Gawer & Cusumano, 2008), the author was forced to perform his own systematic literature review.

From the beginning of the doctoral project, the author kept reviewing by systematic manners a literature review bridging OSS with platforms. This in-progress literature review, based on search keywords and research databases, was initially defended at the Information Systems Research Seminar in Scandinavia (IRIS) seminar in 2011, and submitted as a working paper to the European Conference on Information Systems (ECIS) in 2012. The research author identified a set of emergent research, with stronger focus on platforms over products, that highlighting the limited empirical research on strategies used by open-source platform providers on industry.

For concluding this chapter, from reviewing a very wide body of theoretical knowledge, the author identified a limited but emergent research stream addressing
both OSS and platforms. If the scholars Stol and Babar (2009), Hauge et al. (2010) and Lindman (2011) argue, on a common voice, that current research lacks empirical studies bridging OSS with organizations, the author argues that **current research lacks** even more **empirical studies bridging OSS platforms with organizations**.

4  Theoretical background

This study lays down on previous research built around the concept of multi-sided platforms as introduced by Rochet and Tirole (2003) and further developed by Hagiu (2004), Rochet and Tirole (2006) and Boudreau (2008), the concept of computer-based platforms first coined by Pohl (1998); and key studies on OSS adoption (Wang & Wang 2001; Feller & Fitzgerald 2002; West 2003; Dedrick & West 2004).

With a strong multidisciplinary research character, constructs from outside the perceived boundaries of the information system research were included; Such as the Gawer and Cusumano (2008) strategy tool kit for firms pursuing platform based operational models and the work from Katz and Shapiro (1994) who looks at the role of network effects in the telecommunication industry in a pure network economical context. From philosophy, the theoretical propositions of sense of community (SoC) from McMillan and Chavis's (1986) were adopted, as well as the concept of community of practice (CoP) as proposed by the cognitive anthropologists Lave and Wenger (1991).

5  Methodology and scope

5.1  Epistemological Ontologies and Method

The research author, committed to an article-collection dissertation strategy, did not follow a single epistemological and ontological trajectory. Rather than choosing a particular research approach *apriori* of the field work, the researcher adopted different approaches for each paper while pursuing epistemological consistency on each individual research project.

The author took the stance of a pluralist (Mingers, 2004) employing multi-dimensional research strategies (Mason, 2006) that is, accepting and celebrating a wide variety of paradigms and research approaches on the same topic (Mingers, 2001), in this case the integration of OSS by platform-vendors in the emergent mobile devices industry.
Driven by What, How and Why research questions, the approach of this research is mostly qualitative and most original articles are interpretative (Klein & Myers 1999; Walsham 1995). The research was conducted according to ethnographic principles that empower data collection with limited influence from the established body of theoretical knowledge (Myers 1999; Atkinson 2006). The author was more interested by grounding theories rather than building variance based models (Benbasat et al. 1987; Myers 1997; Markus & Robey 1988; Urquhart & Fernández 2013). By taking and ethnographic approach, the theoretical integration was performed *a posteriori* of the field work and data collection.

The researcher built its methodological awareness both by attending doctoral-level courses and seminars and by reviewing three books in the following order Silverman (2009), Eriksson & Kovalainen (2008) and Myers (2002), all wide-recognized methodology book on qualitative research applied on social sciences, business studies and Information Systems.

The following Table 1 summarizes the research questions, methods and methodological sources driving each individual research project, as reported in more detail on the author's article collection bundled in doctoral dissertation.

**Table 1. Methods employed in each article/research-question**

<table>
<thead>
<tr>
<th>Article</th>
<th>Question</th>
<th>Method</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What do we know about open-source and platforms?</td>
<td>Literature review</td>
<td>Webster &amp; Watson 2002 Järvinen 2008</td>
</tr>
<tr>
<td>2</td>
<td>Measuring the openness of computer-based multi-sided platforms</td>
<td>Literature review</td>
<td>Webster &amp; Watson 2002 Järvinen 2008</td>
</tr>
<tr>
<td>3</td>
<td>RQ1</td>
<td>Descriptive case-study</td>
<td>Yin 1989</td>
</tr>
<tr>
<td>4</td>
<td>RQ2</td>
<td>Descriptive case-study</td>
<td>Yin 1989 Eisenhardt 1989 Kozinets 2002</td>
</tr>
<tr>
<td>5</td>
<td>RQ3</td>
<td>Ethnographic case-study</td>
<td>Eisenhardt 1989 Myers 1999</td>
</tr>
<tr>
<td>6</td>
<td>RQ4</td>
<td>Interpretative case-study</td>
<td>Eisenhardt 1989 Darke et al. 1998</td>
</tr>
<tr>
<td>Article</td>
<td>Question</td>
<td>Unit of Analysis</td>
<td>Collected data</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Article 1</td>
<td>What do we know about open-source and platforms?</td>
<td>Body of theoretical knowledge</td>
<td>Research papers retrieved from research databases indexing books, journals and conference proceedings.</td>
</tr>
<tr>
<td>Article 2</td>
<td>Measuring the openness of computer-based multi-sided platforms</td>
<td>Body of theoretical knowledge</td>
<td>Research papers retrieved from research databases indexing books, journals and conference proceedings.</td>
</tr>
<tr>
<td>Article 3</td>
<td>RQ1 Mobile devices industry players controlling the iOS, Android and Maemo platforms.</td>
<td>Mobile devices industry players controlling the iOS, Android and Maemo platforms.</td>
<td>Open-source software projects source-code and documentation</td>
</tr>
<tr>
<td>Article 4</td>
<td>RQ2 Mobile devices industry players controlling the iOS, Android and Maemo platforms.</td>
<td>Mobile devices industry players controlling the iOS, Android and Maemo platforms.</td>
<td>Companies public financial information, companies-press releases, generalist press and specialized press on technology and mobile devices</td>
</tr>
<tr>
<td>Article 5</td>
<td>RQ3 The WebKit open-source software project</td>
<td>The WebKit open-source software project</td>
<td>Open-source software projects bug-repositories and the change-log from an open-source software project version-control system.</td>
</tr>
<tr>
<td>Article 6</td>
<td>RQ4 Major firm within industry steering an open-source based mobile platform</td>
<td>Major firm within industry steering an open-source based mobile platform</td>
<td>8 Semi-structured interviews: Software developers, Testers, Integrators, Marketing and Strategy</td>
</tr>
<tr>
<td>Article</td>
<td>Collected data</td>
<td>Approach</td>
<td>Methodological references</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>----------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
| 1       | Research papers retrieved from research databases indexing books, journals and conference proceedings. | Analysis of literature review | Webster & Watson 2002  
Järvinen 2008  
Vom Brocke et al. 2009  
Okoli, K Schabram 2010 |
| 2       | Research papers retrieved from research databases indexing books, journals and conference proceedings. | Analysis of literature review | Webster & Watson 2002  
Järvinen 2008  
Vom Brocke et al. 2009  
Okoli, K Schabram 2010 |
| 3       | Open-source software projects source-code and documentation | Multiple-case study Description of OSS components integrated by Apple, Nokia, Google | Yin 1989  
Eisenhardt 1989 |
| 4       | Companies public financial information, companies-press releases, generalist press and specialized press on technology and mobile devices | Multiple-case study 3 rich description of the OSS strategies employed by Apple, Nokia, Google Cross-case analysis | Yin 1989 |
| 5       | Open-source software projects bug-repositories and the change-log from an open-source software project version-control-system. | Network Visualization Figuralisation Heterogeneous Network Analysis Actor-Network Theory | Elias 1978  
Callon 1986  
Walsham 1997  
Cambrosio et al. 2004 |
Eisenhardt 1989  
Darke et al. 1998  
Urquhart & Fernández 2013 |
5.2 Data collection

Data was collected mostly from public-available and naturally-occurring qualitative data on the Internet such as: listed-companies public financial information, companies-press releases, generalist press, specialized press on technology and mobile devices, open-source software projects source-code and documentation, open-source software projects bug-repositories and the change-log from an open-source software project version-control-system. In addition, and for addressing the Why research question (RQ4), semi-structured interviews were conducted in a major firm within the Mobile devices industry.

The included Table 2 sensitizes the research data-collection procedures employed at each individual research project. More details are provided on each article bundled with author's doctoral dissertation.

5.3 Data analysis and theory-building

Reflecting the pluralism employed on this research (Mingers 2001; Mingers 2004; Mason 2006) a multitude of qualitative data-analysis and theory-building approaches were employed. The Table 3 provides more details on the employed data-analysis and theory-building approaches and corresponding methodological sources that guided the research design. More detail is provided in each dissertation article.

References


Lindman, Juho. Not Accidental Revolutionaries: Essays on Open Source Software Production and Organizational Change. Aalto University, School of Economics, Department of Information and Service Economy, 2011.


QA Practices and FLOSS Communities

Adina Barham
Hitotsubashi University, Graduate School of Social Sciences, 2-1 Naka,
Kunitachi, Tokyo, 186-8601, Japan
adina.barham@yahoo.com

Abstract. As FLOSS software is gaining more popularity, communities are realizing that a more rigorously controlled development process is necessary in order to produce high quality products. Such a process includes verification steps and implementing formal quality assurance (QA) procedures that are increasingly present in FLOSS development. These changes in development affect the community structure as new groups of contributors emerge. Although FLOSS communities have been the main focus of many studies, little research has been done on how this emerging ‘layer’ is integrated in the community structure and how it affects the overall dynamics. This study aims to start filling this gap by analyzing available communication data from communities that have formally adopted QA and develop a set of hypotheses. The conclusions reached so far, although provisory, suggest that quality assurance activity may not be increasing steadily over time but dependent on other factors. In addition, data collected points to a highly connected network that contains a large group of members communicating over various channels.

Keywords: quality assurance, test, social network analysis, information flow

1 Introduction

The development process behind FLOSS products has stirred a lot of interest because it seemed to contradict traditional software development by employing a more “bazaar” like approach to the development process. However, despite the fact that many still believe a somewhat mythical description of FLOSS [1], communities started adopting methodologies that were traditionally employed only in proprietary software development. The result is a new hybrid form that is impacting the software market offering a whole new array of possibilities [2-6].

As the user base of FLOSS products increased so did the demand for higher quality from FLOSS products. One of the consequences is a higher need for more complex development process that includes rigorous verification. For example, formal quality assurance (QA) used to characterize proprietary software development but it has been increasingly present within FLOSS. Due to the fact that communities themselves represent the most important resource in deciding the success or failure of a FLOSS
project [7], it is important to study how implementing such a new phase affects their structure and dynamics.

2 Background and Motivation

One of the first steps of this research consisted in assessing the presence of QA procedures within popular FLOSS projects [35]. For this reason, the top 50 FLOSS software products ranked by number of downloads on www.ohloh.net were accessed and it has been discovered that almost one third have QA procedures present. It has been considered that QA is present if there are clear references to its existence (for example, mailing lists, IRC channels, webpages, wikis and so on) and they are easily identifiable. For the next phase of this preliminary assessment, the top 100 software projects ranked by number of users were analyzed. The conclusion remained almost the same as 27 projects had some form of QA present in the development process. In addition it has been noted that almost half of these projects had dedicated communication channels for members performing QA tasks. The remaining projects either had channels for contributors to post test results or other resources dedicated to describe QA within the project such as wikis or websites. The main conclusion that can be drawn from this preliminary research is that QA teams exist in communities where there are dedicated communication channels for members performing QA tasks. However, whether the members of these teams are exclusively performing QA tasks represents an issue that needs to be further studied.

Verification, and more extensively quality assurance procedures under the FLOSS development model is a topic that attracts a lot of interest within the academic community and is reflected in both papers [8-15] and conferences. In addition, this trend is also reflected by the interest that various organizations and governments have shown by starting up initiatives related to quality within FLOSS such as the Qualipso¹ or Qualoss² projects.

Due to the fact that communities represent the driving force behind FLOSS projects, researchers have also focused on aspects such as structure and dynamics [16-18], communication patterns between core and periphery [19-21], or migration within the hierarchy of FLOSS projects [22]. Despite extensive research on both QA and communities little research has sought to link QA with the rest of the community. This research aims to start filling that gap and create a set of hypotheses that can be confirmed in later study.

¹ Qualipso (Trust and Quality in Open Source Systems), http://www.qualipso.org/
3 Research Questions

Q1: What QA activities are present in FLOSS development?

Software quality assurance represents a set of complex activities that are applied throughout the development process. According to Pressman [23], quality assurance includes activities such as competent analysis, design, coding, and testing, as well as formal technical reviews, a multitiered testing strategy, better control of software work products and the changes made to them, and the application of accepted software engineering standards. The question becomes whether within FLOSS development processes one can find the same types of QA activities as in proprietary software.

Q2: What are the characteristics of QA activities and who is performing them?

QA activities can vary based on technical difficulty, for example users may provide automated test tools which might mean that QA might be divided into two subgroups based on activity type. In the case of Firefox, peripheral members (i.e. members of the periphery as defined in the onion model of FLOSS communities) perform some QA tasks such as posting bugs on the issue tracker and the percentage is 20 to 25% [20]. It would be interesting to compare peripheral involvement in other case studies or with other QA related activities.

Q3: How does a QA contributor fit into the community?

Despite the fact that research has shown that the onion model is more complex than previously thought [22], [24] it is generally accepted that FLOSS communities can be split into four layers: passive users, active users, co-developers and core developers. (Passive and active users are together included in the periphery). This research aims to investigate how the emerging QA teams fit into FLOSS communities. More specifically, it asks whether QA is established as a separate category or whether members that belong to other layers perform QA related tasks. Previous research has shown that participants ascend and descend the organizational hierarchy as well as move laterally to other tracks but it has also been suggested that many QA tasks are assimilated by other roles [22]. It would be interesting to note, if all projects taken into consideration share the same structure with respect to the QA contributors.

Q4: What are the communication patterns between QA members as well as with other project participants?
The goal of analyzing communication patterns between QA members is to find the central figures, or in other words members with high activity within the community and observe their evolution over time within the project. Another issue would be comparing central figures based on different data sets, for example central figures based on e-mailing activity and central figures based on issue tracker activity. Social networks are not static and the structure is continuously changing over time [25], which means that participant position within the network is dynamic. For this reason it would be useful to learn the path, or paths, which the central members took to get into that position. Furthermore, previous research has shown that information access correlates with productivity and participants who have better access to information are able to contribute more efficiently [26]. Hence it would be useful to know whether there are participants who control the information flow; such dependence on a few participants may be risky for the project’s long-term sustainability.

4 Data and Research Method

“The problem of quality management is not what people don't know about it. The problem is what they think they know” [27]. In other words, quality assurance (QA) is often misunderstood and is actually a process that requires a complex set of methodologies and procedures [23]. Considering the complexity of QA, an important step in this study was to define QA within the scope of this research. As a result, the following tentative QA definition was devised:

**Testing, contributing code to automated testing tools or any test related activity, triaging bugs or any activity performed on the projects issue tracker, participating on the QA dedicated communication channels represent QA related tasks.**

However, defining QA raises a circularity issue due to the fact that these activities emerged from analyzing publicly available data. It may be possible that some project participants perform other QA activities than the ones observed but are not active on communication channels. The first solution to this problem would be looking at the definition of QA within proprietary (traditional) development and comparing it with activities discovered within FLOSS communities. Another solution for addressing this issue would be method triangulation by confirming the definition through interviews with key members of the community at a later stage in this research. In addition, as one of the premises of this research relies on the correct identification of participants performing QA tasks, it is very important to use additional methods to confirm QA participants. Therefore interviews will also be used to clarify if there are private communication channels between community members.
This research uses five FLOSS projects as case studies in order to generate hypotheses about QA within FLOSS development. The question whether case studies provide sufficient evidence in order to generalize is still a controversial topic. However, it has been agreed that case studies, beside being used for validating existing theory can also provide enough insight in order to produce hypotheses that can be validated through other research methods [28]. A widely cited paper in FLOSS research that uses the case study approach is the study of Apache by Mockus et. al [29]. A set of hypotheses resulted after examining the development process within the Apache server community. Similarly, this research will follow the observational path as a research framework which means that the “goal is to collect a set of observations and to explain them in terms of a set of meaningful concepts” [30]. This goal will be achieved by starting with the substantive domain or area of interest, which is QA within FLOSS, and a technique, which is the case study methodology. For this purpose, the following communities were chosen as case studies:

- Mozilla -- The community behind Mozilla produces successful FLOSS products such as Firefox and Thunderbird. It is a mature community that is well known for stable releases and a high quality standard.
- Ubuntu Community -- The community behind Ubuntu produces an operation system (a Linux distribution). Similar to Mozilla is an established and mature community that is well known for stable releases and a high quality standard.
- Plone -- Plone is a content management system. As opposed to the Mozilla and Ubuntu communities, the Plone community is rather small but it produces a very popular content management system.
- KDE - the Plasma Desktop environment and many other products
- LibreOffice - office software

Communication data was downloaded and stored into local PostgreSQL databases using Python scripts that perform “screen scraping” (web crawlers) or open source tools (for mailing lists exchanged between Plone community members)3. In addition, www.ohloh.net4 was used for data cleaning and further clarifying contributor roles as it contains code contributors’ usernames and nicknames. For example, a member might have more than one username/nickname and www.ohloh.net was used to merge these usernames. However, at this point data cleaning was conducted superficially as data download is not yet complete. An additional clean-up will be performed after all data sets have been downloaded into local databases.

Social network analysis techniques are employed because they allow easy discovery of communication patterns, community evolution and user group affiliation and migration paths. In order to apply social network analysis techniques,

3 https://github.com/MetricsGrimoire/MailingListStats
4 https://www.ohloh.net/
communities will be represented as graphs. Project participants will be represented as nodes (vertices) while interactions will be represented as edges (arcs). The number of interactions between participants will be represented as the weight (value) of the graphs’ arcs.

For the purpose of this research interactions are defined as e-mails exchanged between community members, comments on the issue tracker or any kind of information exchange on a public communication channel. These interactions are defined using either quotation-based analysis [31] either thread-based analysis [16], [24]. Thread based analysis assumes that relations are created between members that are exchanging messages under the same thread. Quotation-based analysis assumes that relations are created between two members if one has quoted another member’s message in his own message. Quotation based analysis was used for mailing lists (except for Mozilla mailing lists) while thread based analysis was used for Mozilla mailing lists and issue tracker communication.

Social networks are dynamic, with a structure that is continuously changing and for this reason, as previous research has shown, it is important to consider time frames [24], [32-33].

4.1 Mozilla

The Mozilla community was chosen as a first case study in order to help draw preliminary conclusions and create a framework for further research. Mailing list data was downloaded in July 2011. At that time, according to the Mozilla Quality Assurance (QMO) website\(^5\), there were 5 sub teams: Web QA, Desktop Firefox, Browser Technologies, Automation and Services\(^6\). Web QA, Desktop Firefox, Browser Technologies, Services teams used the mozilla.dev-quality mailing list while the Automation team used the Mozmill developer mailing list.

Bugzilla data was acquired in February – March after which data cleaning was performed on both data sets by removing double posts, removing spam or normalising name formats. As expected, the QA mailing list was created much later than the issue tracker. For the purpose of analysing activity data should be normalised however, if activity pertaining to a certain period of time will not be taken into account then it might be possible to omit migration of certain members between layers of the community.

The Mozilla.dev-quality mailing list data contains 2535 e-mails exchanged between February 2006 and June 2011 while the Mozmill developer mailing list data contains 1155 e-mails exchanged between October 2008 and July 2011. The traffic and number

\(^5\) https://quality.mozilla.org/
\(^6\) The structure of the Mozilla Quality Assurance teams is dynamic considering the fact that the Services team was later dropped and that since data was collected, other changes have also taken place.
of users is higher on the Mozilla.dev-quality mailing list than the Mozmill developer mailing list (see Table 1). This is to be expected as the Mozmill developer list is addressed to more technically aware users since it is dedicated to an automated testing tool produced by the Mozilla community primarily to test their own products.

Table 1. Mozilla.dev-quality: 2006/17/2-2011/6/30, Mozmill developer 2008/10/1-2011/7/21

<table>
<thead>
<tr>
<th></th>
<th>Mozilla.dev-quality</th>
<th>Mozmill developer</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic</td>
<td>1042</td>
<td>313</td>
<td>1299</td>
</tr>
<tr>
<td>Messages</td>
<td>2535</td>
<td>1155</td>
<td>3689</td>
</tr>
<tr>
<td>Thread initiators</td>
<td>199</td>
<td>47</td>
<td>233</td>
</tr>
<tr>
<td>Distinct author</td>
<td>293</td>
<td>61</td>
<td>332</td>
</tr>
</tbody>
</table>

The issue tracker (Bugzilla) data set covers all Mozilla products from 1998 to 2012 containing 687,221 bugs with 5,834,507 associated comments. Bug ids range from 0 to 724,339; the collected bugs thus represent 94.87% of the id range. The remaining 5.13% were not collected because they were not publicly available or due to bad html that could not be parsed.

Approximately 4,400 distinct project members were identified as assigned to fix bugs. Without getting the data associated with code commits it is not safe to assume that these members were also the members that posted the bug fix. However, they can be considered as known code contributors within the community. These users are also active when it comes to posting bug comments as well as sending e-mails on the QA mailing lists. After cross-referencing members active on the mailing lists and code committers, 883 bugs were found, most of which pertain to Firefox.

From all the e-mails exchanged 152 (4.12%) were sent by authors who had sent only one e-mail throughout the period taken into consideration for this research. On the other hand 135,466 bugs (19.70%) were posted by members who had posted only one bug throughout the period taken into consideration. Most activity has steadily increased over time; however activity on the QA mailing lists do not seem to be linked to time progression. As a result it would be useful to find out if the activity peak in 2009 is linked to interior events or external factors. Developer activity includes bugs to which they are assigned, comments posted on the issue tracker as well as e-mails send on the QA mailing lists.

If we consider that 11 e-mails (average number of e-mails sent) is the lower limit for active users then one can draw the conclusion that only a small percentage of the community can be considered active as only 16.8% of the users sent more than 11 e-mails and 17.69% of users received more than 11 replies. Following the same

---

7 The difference in the total is due to cross posting and users belonging to both lists
principle, only 4.39% of users show a higher than average activity posting more than 39 comments and 9.25% more than 6 bugs.

Table 2. Member activity on a yearly basis

<table>
<thead>
<tr>
<th></th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments</td>
<td>328846</td>
<td>335323</td>
<td>467087</td>
<td>528199</td>
<td>658030</td>
<td>703857</td>
</tr>
<tr>
<td>Bugs</td>
<td>42015</td>
<td>41995</td>
<td>56785</td>
<td>60880</td>
<td>78089</td>
<td>78896</td>
</tr>
<tr>
<td>E-mails</td>
<td>343</td>
<td>361</td>
<td>556</td>
<td>1307</td>
<td>739</td>
<td>384</td>
</tr>
<tr>
<td>Dev bugs</td>
<td>119571</td>
<td>123234</td>
<td>174742</td>
<td>177776</td>
<td>227123</td>
<td>226555</td>
</tr>
<tr>
<td>Dev Comments</td>
<td>258458</td>
<td>271679</td>
<td>375729</td>
<td>449539</td>
<td>541707</td>
<td>561853</td>
</tr>
<tr>
<td>Dev e-mails</td>
<td>196</td>
<td>286</td>
<td>343</td>
<td>953</td>
<td>500</td>
<td>264</td>
</tr>
</tbody>
</table>

In this phase of the research, due to the fact that data collection and cleaning took longer than anticipated, social network analysis techniques could not be applied to the whole data set. Instead interaction was analyzed between active members on the mailing list (more than 10 e-mails sent – 55 users) and 10 members with an average activity on the issue tracker (not among the top/bottom commentators). The resulting network does not depict relations between all QA members and its role is only to offer a sample of the interaction patterns within the community. After eliminating loops (replies to themselves) this sub-network had a number of 1433 participants with 2593 connections; 933 of these connections were formed by more than one interaction. The average degree is 3.16, which means that the average number of connections a member has is approximately 3.

Fig.1. Social Network - Mailing list and 10 active members on the issue tracker
4.2 Ubuntu

Ubuntu was chosen as the next case study for this research as it is yet another mature and successful FLOSS project. As opposed to the Mozilla projects, Ubuntu has a dedicated QA team\(^8\) as well as a Bug Squad\(^9\). The Bug Squad’s tasks include assigning bugs to packages, ensuring that bug reports are adequate, finding duplicate bugs, replicating bugs and so on.

The QA team dataset contains e-mails from October 2007 to August 2012, the Bug Squad team dataset contains e-mails spanning from July 2006 to August 2012 while the Laptop testing team contains e-mails from August 2005 to September 2012 was downloaded. In total 7595 e-mails were exchanged on all three mailing lists.

At this point of the research mailing list data was not cleaned thus the results presented in this paper are only provisionary.

**Table 3.** Member activity on a yearly basis

<table>
<thead>
<tr>
<th></th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA mailing list</td>
<td>-</td>
<td>-</td>
<td>30</td>
<td>324</td>
<td>321</td>
<td>694</td>
<td>486</td>
<td>591</td>
<td>2446</td>
</tr>
<tr>
<td>Bug Squad mailing list</td>
<td>-</td>
<td>206</td>
<td>515</td>
<td>536</td>
<td>475</td>
<td>1232</td>
<td>633</td>
<td>280</td>
<td>3877</td>
</tr>
<tr>
<td>Laptop testing team</td>
<td>338</td>
<td>729</td>
<td>117</td>
<td>0</td>
<td>35</td>
<td>20</td>
<td>12</td>
<td>21</td>
<td>1272</td>
</tr>
</tbody>
</table>

If we consider that 5 e-mails (the average number of e-mails sent per user) is the lower limit for active users, then we can conclude that a small percentage of the community has a higher than average activity as only 14.51% of participants had sent more than 5 e-mails.

In this phase of the research, due to the fact that data was not cleaned thoroughly, final conclusions regarding communication patterns between members can’t be drawn. However a preliminary analysis was conducted using Pajek by integrating the three mailing lists. After eliminating loops (replies to themselves) this sub-network had a number of 1086 participants with 2642 connections; 472 of these connections were formed by more than one interaction. The average degree is 4.8, which means that the average number of connections a member has is approximately 5. The network contains 30 components from which 29 contain less than 3 members. In other words, there is a large connected group of 1019 members.

\(^8\) [http://qa.ubuntu.com/](http://qa.ubuntu.com/)

\(^9\) [https://wiki.ubuntu.com/BugSquad](https://wiki.ubuntu.com/BugSquad)
4.3 **Plone**

Plone was also chosen as its development process includes a QA step and is yet another successful project\(^\text{10}\). As opposed to Ubuntu and Mozilla, Plone has only one QA dedicated mailing list. The QA team has a webpage\(^\text{11}\) where one can find basic information such as activity description, communication channels and team leaders. In the case of Plone, QA activities include triaging new bugs, validating submitted patches, ensuring that new releases are usable and generally help in the release process.

In order to measure QA activity levels on more than one channel, issue tracker data as well as mailing list data were taken into account. Data was retrieved in December 2012 – January 2013. The issue tracker data contains a total of 13026 bugs with 55883 associated comments while the mailing lists contain a total of 29525 e-mails which were downloaded using MailingListStats\(^\text{12}\), an open source tool. In addition, a list of contributors containing names and names used in code repositories was downloaded from Ohloh.net. This list was used to perform data cleaning as follows: names contained in the issue tracker and the mailing list were compared with names from code repositories and merged.

A detail that can be observed from Table 4 is that there is no correlation between time progression and activity levels. In addition, there seems to be no correlation between the number of bugs opened and the comment activity level, which leads to question what causes these spikes in activity levels.

\(^{10}\) [http://plone.org/](http://plone.org/)

\(^{11}\) [http://plone.org/community/teams/qa-team](http://plone.org/community/teams/qa-team)

\(^{12}\) [https://github.com/MetricsGrimoire/MailingListStats](https://github.com/MetricsGrimoire/MailingListStats)
The QA mailing list activity started in 2011 and includes 41 members from whom approximately 70% had sent only one e-mail. As can be seen from Table 5, most users are active on other mailing lists and even contribute code, which means that in the case of Plone, QA can’t be considered as a separate layer in the community.

Table 4. Member activity on a yearly basis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugs</td>
<td>748</td>
<td>1503</td>
<td>1461</td>
<td>1349</td>
<td>944</td>
<td>1482</td>
<td>1336</td>
<td>1391</td>
<td>1060</td>
<td>620</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>1102</td>
<td>3106</td>
<td>3322</td>
<td>4057</td>
<td>3636</td>
<td>5193</td>
<td>5592</td>
<td>11251</td>
<td>6297</td>
<td>4015</td>
<td>8312</td>
</tr>
<tr>
<td>E-mails</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>556</td>
<td>2792</td>
<td>4661</td>
<td>7615</td>
<td>6578</td>
<td>1241</td>
<td>3375</td>
<td>2707</td>
</tr>
<tr>
<td>QA e-mails</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>47</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. QA mailing activity – users that sent more than one e-mail

<table>
<thead>
<tr>
<th>Author</th>
<th>QA e-mails</th>
<th>Other e-mails</th>
<th>Lists</th>
<th>Bugs</th>
<th>Comments</th>
<th>Code Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eric S</td>
<td>43</td>
<td>137</td>
<td>5</td>
<td>76</td>
<td>886</td>
<td>Yes</td>
</tr>
<tr>
<td>alan r</td>
<td>8</td>
<td>152</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>David G</td>
<td>6</td>
<td>190</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>Mindsways</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Dylan J</td>
<td>4</td>
<td>264</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>Unrecognizable</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Laurence R</td>
<td>3</td>
<td>132</td>
<td>7</td>
<td>40</td>
<td>170</td>
<td>Yes</td>
</tr>
<tr>
<td>American PM</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Gil F</td>
<td>2</td>
<td>26</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>robert r</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Christian L</td>
<td>2</td>
<td>37</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Anthony E</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Elizabeth L</td>
<td>1</td>
<td>23</td>
<td>3</td>
<td>0</td>
<td>865</td>
<td>Yes</td>
</tr>
</tbody>
</table>

13 For layout purposes some e-mail author names were shortened. For the full table containing full names please contact the paper author.
An interesting finding after analysing activity levels of QA list members is that the person listed as QA lead on the Plone webpage has only sent one e-mail using her name on the mailing list.

To create the network graph only members that created at least one relation were taken into consideration. The next step consisted of eliminating loops or arcs starting and pointing to the same vertex. An additional reduction was performed in order to remove vertices that had no connections with other vertices. The resulted network contains 3414 vertices connected by a total of 16042 arcs from which 5093 have a value that is different than 1. This means that 10949 connections (68.25%) are created by only one interaction. One could draw the conclusion that these members are occasional contributors or part of the periphery. From the remaining arcs approximately 30.67% have values between 2 and 79, which means that arcs with values between 1 and 79 account to almost 99% of all arcs. The “heaviest” 10 arcs in the network have values between 1243 and 7781.

The average degree is 9.39, which means that in average a person interacts with approximately 9 other people. The density is 0.0013 where the density represents the number of lines expressed as a proportion of the maximum number of lines (that the greater the density the tighter the network structure is).

To get an idea of how users that perform different tasks are distributed in the community structure, contributors were divided into the following four groups (clusters):

1. Code contributors and members of the QA mailing list
2. Code contributors but not members of the QA mailing list
3. Not code contributors but members of the QA mailing list
4. Not code contributors and not members of the QA mailing list

Table 6. Cluster member frequency

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Frequency</th>
<th>Frequency%</th>
<th>Representative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>0.60</td>
<td>Hanno</td>
</tr>
<tr>
<td>2</td>
<td>344</td>
<td>9.94</td>
<td>Khink</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>0.46</td>
<td>Steve</td>
</tr>
<tr>
<td>4</td>
<td>3079</td>
<td>88.98</td>
<td>jonstahl</td>
</tr>
</tbody>
</table>

Table 6 presents the size of each cluster and how much percent does that cluster represent from the total of the community. It can be easily observed that the cluster containing members who are neither code contributors neither QA contributors is the largest representing almost 89% of the whole community. On the other hand people performing QA task appear to add up to only 1.06% of the community. At this phase of the research it is still unclear whether members of cluster 4 are periphery members that occasionally contribute or if they have different roles in the community.
The following image represents an aggregated graph of the Plone network where each cluster is represented by a different color.

![Graph of Plone network](image)

**Fig.3. Social Network – Plone**

To analyze the communication patterns only between users that are active on the QA mailing list a sub-network containing only clusters 1 and 3 was extracted and isolate vertices were removed. The resulted network had 31 vertices linked by 133 arcs of which 58 had a value greater than 1. The density has a value of 0.14 while the average degree is 8.58. The “heaviest” 10 arcs have values between 49 and 125.

In order to portray an accurate image of the network evolution after dedicating a communication channel to the QA team, social network analysis methods were applied using 6 months time frames. The size of the network varied between 226 and 746 vertices, which is to be expected considering the fact that many arcs were created after only one interaction. In addition the highest degree was 7.52 while the lowest was 4.29. The next step in this research is to apply smaller time frames and then compare central figures in different time frames as to define activity change in time or in other words how the contribution level (interest) of contributors develops over time.
4.4 Other Projects

In order to compare previous results with other communities, other two projects were chosen, namely KDE\textsuperscript{14} and LibreOffice\textsuperscript{15}. The mailing lists associated to the QA team of both projects were downloaded in January 2013.

<table>
<thead>
<tr>
<th>Table 7. Activity levels on a yearly basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Name</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>KDE</td>
</tr>
<tr>
<td>LibreOffice</td>
</tr>
</tbody>
</table>

Graphs associated to the social networks of both projects were created using methods similar to the ones used for previously described communities. The KDE graph contains 48 vertices connected by 117 arcs of which 35 have a value greater than 1. The density is 0.05 while the average degree is approximately 5. The LibreOffice graph contains 168 vertices connected by 868 arcs of which 345 have a value greater than 1. The density is 0.03 while the average degree is approximately 10.

![KDE Graph](image1.png) ![LibreOffice Graph](image2.png)

**Fig.4.** Social Network – KDE and LibreOffice QA mailing lists

At this phase of the research no conclusion can be drawn regarding these two projects as the data gathered represents only mailing list communication between QA members. As a next step, the list of contributors will be downloaded to the local database from ohloh.net and clusters will be created as in the Plone study case. In addition, other mailing lists and bug tracker data will be added.

\textsuperscript{14} http://community.kde.org/Getinvolved/Quality
\textsuperscript{15} http://www.libreoffice.org/get-involved/qa-testers/
5 Conclusions

Q1: What QA activities are present in FLOSS development?

A clear definition of QA activities has not been made, however evidence such as the existence of a QA mailing list oriented to more technically aware users, in the case of Mozilla, might suggest that there is more than one type of QA task. In addition, tasks performed by Ubuntu's Bug Squad may also be considered as QA activities thus pointing out that a detailed classification is necessary. In the case of Plone there does not seem to be a clear separation in QA activity types.

Q2: What are the characteristics of QA activities and who is performing them?

Members of the QA team are active on other mailing lists and also contribute code. This aspect needs to be further studied in order to determine the percentage in which QA contributors perform non-QA activities.

The activity of community members that “hit and run” is higher on the Issue Tracker than on the QA mailing list for Mozilla projects. This may suggest that QA contributors have a more sustained activity in the Mozilla community. Another difference is that issue tracker activity has shown an increase over time while mailing list data showed a peak level. In addition, Ubuntu mailing list activity and Plone mailing list and issue tracker activity showed similar spikes. This might suggest that activity levels may not be related to time progression but to other variables that need to be identified.

Q3: How does a QA contributor fit into the community?

In the case of Plone, QA is not a separate layer within the community considering the fact that the most active members have also contributed code to the project and were active on other mailing lists. However, it may be possible that non-QA activities were performed in different time frames than the ones in which the contributors were part of the QA team. For further clarifying this aspect it is necessary to compare these time frames in another study. Approximately 70% of QA mailing list members have sent only one e-mail and 30% have not been active on other channels. Due to the fact that data cleaning was performed superficially at this phase, a later reassessment is necessary. Furthermore, members who are active on the QA mailing list account to only 1.06% of the whole community.

In the case of Mozilla, Ubuntu, KDE and LibreOffice it is not clear yet whether QA is established as a separate layer in the community as more data needs to be collected. However, these teams represent a layer in the community considering that they have dedicated communication channels.
**Q4: What are the communication patterns between QA members as well as with other project participants?**

The data associated with Ubuntu, Mozilla, KDE and LibreOffice is incomplete and thus general conclusions cannot yet be drawn regarding communication patterns. However, based on the preliminary social network analysis, the conclusion is that there is a core of people that are highly active.

In the case of Mozilla, the data collected so far points to the fact that there is a large team spanning both mailing lists and issue tracker. In addition, judging from the activity of QA members and code committers on the issue tracker it is safe to say that interaction with other community members has been increasing. In the case of Plone, the community as a whole seems to form a large component that spans both issue tracker and mailing list with the exception of some smaller sub-networks. This means that for both projects there is a lower risk of a few members controlling the information flow and jeopardizing communication.

### 6 Limitations and Further research

The purpose of this research is not to create a recipe for success that can be applied to all FLOSS projects as every project has its unique characteristics. The goal is to start clarifying how and by whom is QA performed in FLOSS projects by creating a set of hypotheses that may be confirmed in future studies. However, assessing whether the quality of FLOSS projects has improved or not after implementing QA is difficult even though previous studies have attempted to create a methodology for measuring FLOSS quality [34] and thus is beyond the scope of this study.

One of the limitations of this study is that the community members might also use other communication channels that are not publicly available. In addition, at this point of the research data cleaning was performed superficially thus the results presented are only provisory.

The next step of this research is integrating additional data such Ubuntu contributor list and issue tracker data (download is still in progress). Also, a thorough data cleaning is necessary before applying any additional social network analysis techniques.

Follow-up interviews are necessary in order to correlate data peaks and anomalies, clarify the private communication aspect as well as define QA activities and to confirm the findings of this study. In addition, community analysis should use time frames for all the projects taken into consideration. Furthermore, community evolution should be analyzed with time frames, before adopting formal QA as well in order to track down potential migration of members from one team to the other.
7 References


Free/Libre Open Source Software users’
Local Face-to-Face Meetings in
Career-Related Activities

Eunyoung Moon
School of Information, University of Texas at Austin, Austin, Texas
eymoon@utexas.edu,
WWW home page: http://www.ischool.utexas.edu/~eymoon/

Abstract. While prior studies on Free/Libre Open Source Software (FLOSS) users have mainly looked at FLOSS users in the context of FLOSS communities, little is known about whether and how FLOSS communities play a role in the context of its users’ careers. This study takes an inductive approach to investigate the phenomena that are emergent and poorly understood. As promising themes emerge from a pilot study on Drupal users’ local face-to-face meetings, the author proposes research question and research design. The proposed study plans to investigate local LAMP (Linux, Apache web server, MySQL, PHP/Perl/Python) users who attend local face-to-face meetings. The proposed study pays attention to: what role do FLOSS users’ local face-to-face meetings play in the context of their career-related activities. It is expected that the proposed study will contribute to FLOSS research by giving a new insight into FLOSS users in their careers, how FLOSS projects fit into local FLOSS users’ careers, and how to manage offline activities for FLOSS users.

1 Introduction

A majority of studies on Free/Libre Open Source Software (FLOSS) have looked at FLOSS users in the context of FLOSS projects. From this perspective, researchers studied what motivates FLOSS users to seek out communities [20] and to participate in FLOSS user group activities [2,14]. In that study, researchers found that new FLOSS users seek out communities due to a need to solve problems [20] as well as specific personal motives [2,14]. Study on Apache web server user groups [14] found that participants who posted answers to questions in Apache online help system gained direct learning benefits. Researchers on Linux user group [2] suggested that FLOSS users are motivated by a combination of social and psychological factors such as identification with the open source movement. In a similar vein, researchers have looked at how FLOSS users join FLOSS projects [13,22,25], how they are socialized in FLOSS communities [7], and why they stay or leave FLOSS projects [8].

While prior studies have mainly looked at FLOSS users in the context FLOSS projects, few studies [5,15] investigated FLOSS communities from the perspective on its users’ lives. Crowston et al. (2007) investigated the role of face-to-face meetings in
the rest of FLOSS developers’ lives by observing and interviewing FLOSS developers at the several formal annual events such as Apachecon 2003 and 2004, as well as, OSCON 2004. The authors in that study suggested that the value of face-to-face meetings within FLOSS context lies in face-to-face socialization which facilitates developers’ interactions through developing and maintaining social ties, and helping build shared mental models. Furthermore, the authors found that FLOSS conferences helped FLOSS developers set aside time for the FLOSS projects. Marlow and Dabbish (2013) in a recent study [15] investigated how potential employers utilize activity traces on GitHub for hiring software developers and how job seekers attempt to manage the impressions in GitHub as prospective employees. The authors in that study suggested that GitHub provides employers with the detailed information for recruitment, for example, activity level in FLOSS projects, code styles, and project management abilities. Furthermore, it was found that job seekers attempted to promote their presence in order to supplement their job applications by providing a link to their GitHub account. Those findings indicate that GitHub is employed by FLOSS users in the context of their careers.

In sum, a majority of studies on FLOSS investigated FLOSS users mainly from the perspectives on FLOSS projects rather than on their lives [5] or careers [15]. In the pilot study, however, the author identified FLOSS users’ career-related activities in FLOSS users’ local face-to-face meetings, based on the qualitative data collected from multiple sources. While face-to-face meetings of FLOSS users appear to occur more today than in the past, there is little research that sheds light on them. The following section starts with a pilot study in which the promising themes emerge.

2. A pilot study

As a pilot study, the author has studied three types of local Drupal\(^1\) users meetings, two regular meetups and the annual Local Drupal Day. The period of the pilot was from September to December in 2012. The Drupal project was chosen because 1) Drupal project is mature, 2) local Drupal meetups have two types of meetup: Newbie meetup for beginners and Drupal user group (DUG) meetup for advanced users. It was expected that the researcher could meet a diverse range of local Drupal users.

A local Drupal user group (DUG) meetup was created by two big Drupal-based companies in a local city in 2009 since two companies wanted to get together and to exchange ideas face-to-face. One of the two companies has supported local weekly DUG meetups as well as local monthly Drupal events such as the local meeting on Local Drupal Day, which is an annual event where Drupal users gather in cities across

\(^1\) Drupal [6] is a free and open-source content management system written in PHP. Drupal was initially released in 2001 and current version is 7.22. Its uses range from personal blogs to corporations, the government sites, and universities.
the world. The weekly DUG meetups have been held at the local pizza store in the open space. In this weekly meetup, DUG meetup attendees sat around the table with meals and shared the individual Drupal issues with other attendees.

The monthly Drupal events have been held at the local co-work space in which presentations or workshops are available. The monthly events had the presentations on the latest issues or special issues in Drupal and presenters stood in the front stage. In the middle of the presentation or after the presentation, attendees asked questions to the presenter and discussed the issues.

A local Drupal newbie meetup was created by the individual Drupal developer working in the local college. This meetup aims to help Drupal beginners learn the basics of Drupal from the installation to customization. The newbie meetup has been held in the seminar room of the local college. This meetup had the presentation on the basics of Drupal as well as the question and answer sections.

In the observation, the author participated in local meetups and a local event as a Drupal beginner and aDrupal meetup member, spending a total of 10 hours. Approximately 60 members attended local Drupal Day, 12 members attended local Drupal newbie meetup, 5 members attended local weekly DUG meetup at the first observation, 4 members attended local weekly DUG meetup at the second observation.

In the course of observation, the author joined in conversations among Local Drupal Day attendees regarding the local event itself. Also, in the local meetups, the author asked questions regarding terminologies used in Drupal, third-party tools to help start learning Drupal, as well as, questions about the projects that meeting attendees were developing and how meetup plays a role in their projects based on Drupal. By engaging in observation, the author wrote detailed field notes and audio-taped conversations when meeting attendees speak too technically. In addition, the author took photos of what local DUG meetup attendees worked on.

In addition, the author conducted two semi-structured interviews with the organizer of Drupal newbie meetup and the organizer of DUG meetup. The interview protocol was developed in advance, however, the question was modified and added, depending on the interviewee’s interests. The interview questions mainly addressed: the motivation to use/develop Drupal, the overall information about local meetings, the motivation to attend other local meetings for Drupal users, and whether the interviewee has been involved in other FLOSS projects.

The author also collected a total of 273 meetup members’ online introductions from local Drupal newbie meetup website and local DUG meetup website. DUG meetup members’ introductions included answers to the questions: 1) what would you like to learn from and/or contribute to the local Drupal community?, 2) have you done a project in Drupal, and 3) if so, describe the project. In addition, the author paid attention to the local Drupal meetup’s message boards, including a total of 52 messages posted by local commercial companies and organizations looking for Drupal
developers. Several business cards the author received in Local Drupal Day were also collected as artifacts for the study.

To analyze the qualitative data collected, the author used ATLAS.ti [1], a qualitative data analysis software, in the course of coding. In the 1st round of coding, the author identified and developed concepts by a line or a paragraph of field notes, interview transcripts, and artifacts. This stage of coding resulted in naming and categorizing phenomena. Then, in the 2nd round of coding, the author focused on the context of local face-to-face meeting attendees’ career-related activities. In this stage, the author made connections between a category and its subcategories. For example, the code, *Career-related activities* were developed in the second stage of coding by making connections among sub-codes developed in the 1st round of coding: *job requirement*, *looking for a new job*, *the use of Drupal at workplace*, and *job advertisement*. As promising theme such as *Career-related activities* emerges in the course of coding, the author plans to focus data collection on it.

### 3. The Research Question emerges

In the field notes, interviews, and local Drupal meetup members’ online introduction, the author identified FLOSS users’ local meetings in terms of *career-related activities*. For instance, local meetup attendees brought to the meeting individual specific problems encountered in the course of developing the content management systems for their own customers. In the meetup, attendees discussed their technical problems to look for the solutions. For example, one attendee learned that the part he wanted to modify was hardcoded and he should submit a patch or create a new module from scratch for completing his commercial consulting services. Such an episode accompanied that attendees mentioned that they couldn’t find answers by googling or by posting the problem in online community. These episodes reveal that Drupal users’ local meetups seem to be the place in which meetup attendees share or solve the technical issues of their *job requirement*. This theme also appeared from the analysis of local Drupal meetup members’ introductions. Furthermore, in the preliminary interviews, the organizer of local Drupal newbie meetup noted that he uses Drupal since there are local communities in which somebody can help him. The organizer of local DUG meetup also mentioned:

> I have talked about some problems, like I have to do this for client. Have you been using this module before? The same questions of other people come here to ask and I have the same question as well.

64 Drupal meetup members specified their careers in online introductions. Also, 5 business cards the author received at local face-to-face meetings helped the author identify their careers. The analysis of those data showed that many of the attendees are
employed locally. The background of Drupal meetup members as employees appeared as a developer, a system administrator, a web designer, a company owner.

From the analysis of online introductions, themes looking for a new job and the use of Drupal at workplace emerged. The theme looking for a new job was defined as the cases when the member introduced oneself as prospective employee who was seeking for a new job with Drupal. For example, one member introduced oneself:

(I’m a) Web designer with Drupal experience looking for work! Enjoyed being a part of the Neighboring town Drupal group from 2008-2012, but looking forward to getting to know the local Drupal users.

Meetup members also introduced themselves as workers who use Drupal at workplace at Drupal meetup website as well as at local DUG meetups. Meetup members mentioned what they’ve been working on Drupal in the workplace. In local DUG meetups, each meetup attendee introduced that the individual has worked on the development websites for non-profit organizations, for online gamers, for one’s own customers, and has worked as an administrator with Drupal. The following online introductions are the examples of this case:

New to the local city, have been doing some heavy duty Drupal implementation for the company. I work for since late last year. We implemented the Drupal 7(version) site.

I'm a Linux systems administrator who's been developing Drupal sites for about a year.

The analysis of online introductions showed that attendees included local employers. A total of 9 meetup members’ introductions were identified as employers who were looking for Drupal developers. These introductions specified the technical skills needed for the job, information about the companies and where to contact. The example of introduction is as follows:

We are ‘local Marketing’, a marketing and development firm based in a beautiful local city. We're looking for LAMP/Drupal developers...please contact us!

Local Drupal meetup members who work for local companies (e.g., two big local Drupal-based companies), local non-profit organizations, local universities, local state agencies, or local staffing firms posted a total of 52 job advertisements in the message boards of local Drupal meetup websites. These job advertisements described the work modes as either in-house or telecommunication and the positions as either temporary contract work or regular full-time work. For example, one job advertisement described that local Drupal developers were immediately needed to complete the development of two custom modules in the next 2 weeks. In contrast to this case to look for temporary workers, one of the two big Drupal-based local companies posted a total of 7 job
advertisements to look for local full-time workers such as Drupal developers, Drupal web designers, and architect.

Two local DUG meetups were financially supported by the commercial company that was looking for senior software developers who know Drupal. In these meetups, meetup attendees repeated the job description again and again and exchanged information so that potentially interested developers can contact the company. In the middle of conversation, meetup attendees noted that the company’s main base is not the US and started to talk about the Drupal-based local companies in terms of their technical interests such as programming languages and operating systems. Furthermore, meetup attendees shared the perspectives on the local context including weather, the city as a tech city, and its cultural features.

Those emerging themes job requirement, looking for a new job, the use of Drupal at workplace and job advertisement emerged in the 1st round of coding, and the author made a connection among them by creating a theme career-related activities in the 2nd round of coding. However, those sub-codes of career-related activities might vary according to FLOSS project.

To focus on the promising themes career-related activities, the author proposes the research question: What role do FLOSS users’ local face-to-face meetings play in the context of their career-related activities?

4. Research Design

4.1 Research Setting

To further investigate FLOSS users’ local face-to-face meetings in their career-related activities, the author proposes to collect data from FLOSS user groups in a local city of the US. In this study, local face-to-face meetings include active local FLOSS users’ meetups held on a regular basis, local FLOSS conferences, and local FLOSS events etc.

Based on a preliminary study on Drupal users, the author proposes to investigate other FLOSS user groups with the following criteria: 1) FLOSS project is mature, 2) the use of FLOSS is common at commercial software companies, 3) local FLOSS user groups have had local face-to-face meetings on a regular basis. The user groups that best match these criteria are local meetings focused on LAMP (Linux, Apache web server, MySQL, and PHP, Python or Perl). Choosing multiple, though similar, cases should provide a basis for comparison and give more insight to the proposed study. The researcher will seek recommendations for participants for other relevant meetings.
4.2 Data collection

4.2.1 Qualitative interviews
The author proposes to conduct qualitative interviews with a diverse range of FLOSS users who attend local face-to-face meetings. To recruit respondents from a variety of cases, the author uses two methods. Firstly, the author will start with respondents from members of local FLOSS meetups held on a regular basis and then, ask each respondent to recommend others who could be interviewed and who attended the local face-to-face meetings, known as snowball sampling [23]. However, using only snowball sampling creates a risk that only participants from the same networks will be sampled. Accordingly, the author will use theoretical sampling in order to mitigate a biased sampling, depending on a set of the respondents. Theoretical basis will be developed, based on the FLOSS survey [9]. In that survey, the authors reported occupational background of FLOSS developers as software engineers, programmers, consultants, executives/marketing/product sales professionals, university staffs, and students. The feature of employment status will also be useful to construct the sample. Following these two features of occupational backgrounds and the employment status, the author will ask the respondents to recommend the individuals in this set of theoretical sampling.

In the interview, the author plans to develop the interview protocol with the following main questions: 1) what was the first/recent local meeting the interviewee attended and what motivated him or her to attend the meeting, 2) whether the interviewee have exchanged business cards with other attendees and if so, what prompted this exchange and whether the interpersonal relationship evolves or not. 3) the degree of working with FLOSS at workplace and how it relates to attending to local FLOSS users’ face-to-face meetings, compared to online venues. In addition, interviews will include questions for FLOSS users as employers about the ways of looking for FLOSS developers for recruitment and whether and how the hiring process relates to local FLOSS users’ meetings such as local FLOSS conferences and local FLOSS users’ meetups.

4.2.2 Observation
The author will collect data by observing local face-to-face meetings, taking notes that will describe the settings, the activities that took place, the people who attended, and the meanings of what was observed [18]. The data collected from observation will enable the author to comprehend new dimensions of meeting attendees’ actions [11] that might not be captured in interviews since the respondents could be unaware of their behaviors. Observing local FLOSS users’ face-to-face meetings will enable the author to understand the nature of each local FLOSS communities as well as individual meeting attendees. In these observations, the author will take field notes in
detail and record conversations with permission, when meeting attendees talk too technically for the observer’s immediate comprehension.

### 4.2.3 Public Documentation

The author will collect artifacts from FLOSS user groups’ websites as well as local face-to-face meetings. In the local meetup website, job advertisement from commercial companies and meetup members’ introduction are the examples of artifact for analysis. In addition, public messages among local FLOSS users’ meetup members will be analyzed.

By collecting multiple data of multiple sources, this study increases validity through triangulation, enabling the researcher to both confirm understandings from individual sources of data and to discover aspects perhaps not included in one source of data.

### 4.3 Data analysis and procedures

In the proposed study, the purpose of data analysis lies in discovering and understanding the relationships among FLOSS users who attend local FLOSS users’ face-to-face meetings and how FLOSS projects fit into careers [10]. That is to say, the author moves towards explanation [19] by engaging in several coding processes [21]. In the 1st round of coding, the author labels phenomena and discover themes from labeled phenomena. In the 2nd round of coding, the author makes a connection between a theme and its sub-theme regarding a set of relationships indicating context, actions, and interactional strategies etc. In this stage, the data collected both in the pilot study and in the future study put back together in new ways by making connections. Accordingly, it is expected that new themes will occur and existing themes will be likely to be modified, as the author collects the qualitative data from multiple FLOSS users’ local meetings. The number of processes to make connections between a theme and its sub-themes cannot be fixed since it depends on the properties and dimensions of contexts from the data collected. After this process, finally a core theme will be formulated to explain the relationships among conceptual constructs or stories.

### 4.4 Future Plan for Validity

The author plans to discuss about the findings with the key informants for the validity of this study, known as “member checking” [11]. To be specific, the researcher plans to check the interpretation in the course of coding the qualitative data. In addition, the author plans to finally send my final writing to the key informants.
The researcher will re-visit pertinent literature [21] on networking events for careers outside FLOSS [3,4,16,17,24]. Returning to existing literature, the author will investigate how findings on networking for careers [12] outside FLOSS appear in similar ways and in different ways in the context of FLOSS by pointing out a set of conditions or concepts in FLOSS communities.

5. Expected Contribution

The proposed study is expected to contribute to FLOSS research theoretically and practically. Theoretically, it is expected to develop a way of thinking about FLOSS users in the context of their careers. A recent study on employer’s use of github for recruitment and hiring for software developers [15] suggested a new perspective on the use of GitHub, a collaboration tools for FLOSS projects, in the context of its users’ careers. The proposed study will provide the new perspective on 1) FLOSS projects in the context of FLOSS users’ careers and 2) how FLOSS projects fit into local FLOSS users’ careers. The proposed study investigating multiple FLOSS projects in local FLOSS events will give researchers an understanding of how experiences in FLOSS projects relate to FLOSS users’ job seeking, satisfying job description, and hiring processes.

Practically, the proposed study is expected to help further understandings of FLOSS users’ offline activities. While few studies [2,5] have looked at what FLOSS participants do in their face-to-face meetings, the proposed study investigates how local FLOSS users’ face-to-face meetings play a role in their activities beyond socialization. The author expects the proposed study would provide FLOSS communities with insight that will be useful in managing offline activities in their communities.

Acknowledgment
The author is grateful for the helpful comments of her doctoral advisor, James Howison and for the encouragement of Joan Hughes.

References


