

AN ABSTRACT OF THE DISSERTATION OF

Amirhosein “Emerson” Azarbakht for the degree of Doctor of Philosophy in
Computer Science presented on May 18, 2017.

Title: Longitudinal Analysis of Collaboration in Forked Open Source Software
Development Projects

Abstract approved: _____

Carlos Jensen

Social interactions are a ubiquitous part of our lives, and the creation of online social communities has been a natural extension of this phenomena. Free and Open Source Software (FOSS) development efforts are prime examples of how communities can be leveraged in software development, where groups are formed around communities of interest, and depend on continued interest and involvement.

Not everything works smoothly all the time in open source projects. Problems arise for a variety of reasons, including collaboration and communication problems, which results in uncertainty about the operational health and survivability of the projects. Many stake-holders are affected by this uncertainty, including industry sponsors, individual contributors, corporate developers, and users, who all have decided to invest time and effort in the project, and will be affected if a project suffers from troubles.

Forking in FOSS, either as a non-friendly split or a friendly divide, affects the community. Such effects have been studied, shedding light on how forking

happens. However, most existing research on forking is post-hoc. In this study, we focus on the seldom-studied run-up to forking events.

We used the following two approaches to study the evolution and social dynamics of FOSS communities; 1) Time series analysis of the contents of the messages sent and received on the projects developers mailing list, for the time period of 10-month run-up to the fork was analyzed for anomalies, indicative of simmering conflicts. 2) Social network analysis using a developer-oriented approach to statistically model the changes a community goes through in the run-up to a fork, in which the model represents tie formation, tie breakage, and tie maintenance between developers. We estimated several model parameters that capture the variance in the changes the community goes through. We found that conflict-driven forks exhibited anomalies; time series analysis of sentiments showed the anomalies occurred before and close to the fork event. Whereas non-conflict-driven forks did not suffer from such pre-fork anomalies. The objective was to be able to evaluate the operational health of the project community, and intervene if need be. We suggest anomaly detection of the time series analysis may be used by the project stakeholders/investors as key indicators left in the record, that can be used to identify problems among developers, and intervene if need be.

We also found that in conflict-driven forks, (1) the developers maintained a preference for interacting with developers who had similar out-degrees, in contrast to the non-conflict-driven forks, where the developers did not require similar out-degrees. The interpretation may be that a project with non-conflict-driven forks had a more inclusive and classless core developer team. (2) The interactions were reciprocal, in contrast to the non-conflict-driven forks, where the interactions did not need to be reciprocal to happen. The interpretation

may be that the projects with non-conflict-driven forks were more open to interactions whether or not they would get something back in return from the other developer. (3) In conflict-driven forks, the more senior developers preferred to interact with other more senior developers, in contrast to the non-conflict-driven forks. The interpretation may be that the senior developers in projects with conflict-driven forks were less involved with junior developers than in projects with non-conflict-driven forks. (4) In non-conflict-driven forks, the developers with high source code contribution levels interacted more with other high source code contributors. (5) In non-conflict-driven forks, high levels of contribution to the source code brings you connections more rapidly, while high levels of contributions to the mailing list is not suggestive of this. This can be interpreted as a sign of meritocracy based on code, rather than talk, which captures a healthy dynamic in these projects.

©Copyright by Amirhosein “Emerson” Azarbakht

May 18, 2017

All Rights Reserved

Longitudinal Analysis of Collaboration in Forked Open Source Software
Development Projects

by
Amirhosein “Emerson” Azarbakht

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented May 18, 2017
Commencement June 2017

Doctor of Philosophy dissertation of Amirhosein “Emerson” Azarbakht presented
on May 18, 2017

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Amirhosein “Emerson” Azarbakht, Author

ACKNOWLEDGEMENTS

I would like to express sincere appreciation to family, friends, colleagues, and committee members.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1

Longitudinal Analysis of Collaboration in Forked Open Source Software Development Projects

0.1 Introduction

Lorem ipsum.