

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Incentives, Mechanisms and Platform Design for IT-enabled Open Innovation: Lessons
from Online Communities and Open-Source Software

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor
of Philosophy

in

Management

by

Wei Chen

Committee in charge:

Professor Kevin Xiaoguo Zhu, Chair
Professor Terrence William August
Professor Joshua S. Graff Zivin
Professor Karsten Theil Hansen
Professor Viswanathan Krishnan

2016

Copyright

Wei Chen, 2016

All rights reserved.

The Dissertation of Wei Chen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016

DEDICATION

I dedicate this dissertation to the memory of my father, for the seed he planted to pursue my passions. I also dedicate this dissertation to my mother for her constant, unconditional love and support.

TABLE OF CONTENTS

Signature Page.....	iii
Dedication.....	iv
Table of Contents.....	v
List of Figures.....	vii
List of Tables.....	viii
Acknowledgments.....	ix
Vita.....	xii
Abstract of the Dissertation.....	xiii
Dissertation Overview.....	1
Chapter Overviews.....	5
References.....	8
Chapter 1 Engaging the Wisdom of Crowds: Structural Analysis of Dynamic User Contributions in Online Communities.....	10
1.1 Introduction.....	11
1.2 Literature Review.....	14
1.3 Research Context.....	19
1.4 Structural Modelling of User Behavior.....	23
1.5 Analysis and Discussion.....	29
1.6 Closing Remarks.....	40
Appendix 1.1.....	50
Appendix 1.2.....	54
References.....	56
Chapter 2 Competition among Proprietary and Open-Source Software Firms: The Role of Licensing on Strategic Contribution.....	61
2.1 Introduction.....	62
2.2 Literature Review.....	68
2.3 Model.....	73
2.4 Consumer Market Equilibrium and Pricing.....	78
2.5 Oligopolistic Competition in Enterprise Software Markets.....	84
2.6 Concluding Remarks.....	107
Appendix 2.1.....	116
References.....	147
Chapter 3 “Release Early, Release Often”? The Impact of Release Frequency in Open-Source Software Co-Creation.....	151
3.1 Introduction.....	152
3.2 Literature Review.....	156
3.3 Theoretical Analysis.....	158
3.4 Data.....	168
3.5 Empirical Analysis and Results.....	172

3.6 Discussion.....	178
3.7 Concluding Remarks.....	182
References.....	189

LIST OF FIGURES

Figure 1.1: Hidden Markov Model of User Contributions..... 44
Figure 1.2: Trends of Key Variables..... 44
Figure 1.3: Average Answers over Time and Teneure Distribution..... 45
Figure 1.4: Fluctuation of User Contribution over Time..... 45
Figure 1.5: Design Simulations: What if reputation is easier to earn?..... 46
Figure 1.6: Out-of-sample Forecasting and Simulations..... 47
Figure 2.1: Sequence of Events..... 111
Figure 2.2: Equilibrium Effort Choices and Resulting Quality Levels under Region H as Affeted by License Restrictiveness..... 111
Figure 2.3: Consumer Surplus Associated with the Providers’ Offerings under Region H..... 112
Figure 2.4: Equilibrium Prices and Demand under Region H..... 112
Figure 2.5: Equilibrium Effort Choices and Resulting Quality Levels under Region L as Affeted by License Restrictiveness..... 113
Figure 2.6: Equilibrium Prices and Demand under Region L..... 114
Figure 2.7: Consumer Surplus Associated with the Providers’ Offerings under Region L..... 115

LIST OF TABLES

Table 1.1: Variables and Descriptive Statistics.....	48
Table 1.2: Results of HMM Bayesian Estimation.....	49
Table 1.3: Mean Posterior Transition Matrices.....	49
Table A1.1: Estimation Results from Simulation Data.....	55
Table 3.1: Variables and Descriptive Statistics.....	185
Table 3.2: Impact of Release Frequency on Download.....	186
Table 3.3: Impact of Release Frequency on Community Contribution.....	187
Table 3.4: Impact of Release Frequency on Anonymous and Non-Resolved Patches.....	188

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Kevin Zhu, my primary advisor and committee chair. I am eternally grateful for his excellent guidance and patience throughout these years. Thanks to him, I have learned not only to be thoughtful in research, but also to be deliberate in life. I have also learned how to be an effective instructor through observing his teaching and working with him. Looking back to the last several years, I realize how much I have grown through numerous emails, conversations, and meetings with him. I am grateful that Kevin has been a patient and supportive advisor, and has mentored me through my graduate years.

I am also very grateful to Terrence August, my other important source of learning throughout my graduate years. Terry directed me in my analytical training and guided me through the steep learning curve in mathematical modeling. I still remember how he quickly led me out of the obstacles that seem undefeatable to me, for many time. He has pushed me to understand and practice rigorous thinking. He has also been a great teacher in choosing research topics and communicating research findings.

The rest of my committee has been invaluable through their support and feedback. Vish Krishnan, in particular, has offered as much feedback and advice as we worked on our paper. I also thank him for his tolerance of my slow progress. My two other committee members, Karsten Hansen and Joshua Graff Zivin, have patiently endured my talks and offered whatever feedback was possible. I also thank Karsten for his valuable advice during my job market preparation.

I would like to thank Hyoduk Shin, whose advice for job market and research was invaluable. I also appreciate the time he took to mentor me. It was the conversations with

him that made me realize how much better I could do in everything. I also owe many thanks to Qiang Ye, who has been my master advisor and led my way to the research journey. His passion and creativity have inspired and encouraged me during my Ph.D. I am also thankful to Xiahua Wei, for our numerous conversations on research and life.

I would also like to thank many Rady faculty members for helping me to develop my research and teaching. These include Sanjiv Era, Vincent Nijs, David Schkade, Uri Gneezy, and many others. I thank my fellow students and friends, Riccardo, Duy, Yinchu, Junghee, Coby, Lakshmi, Kev, Kristen, Matt, Min, Dani and others, for listening to me in my practice talks, for giving me suggestions on research, and for being part of my Rady community. I also owe a debt of gratitude to Marisol Nierva-Magnano, Laurel Nelson, and Taylor Robinette for their support along the way.

Life has also blessed me with the most amazing friends. The graduate years were made so much better as a result of them. Riccardo, I will forever cherish the late nights we spent together dealing with econ courses, as well as numerous conversations over lattes and espressos. I am in debt to him for his help and suggestions in my job market preparation and job talks. I also appreciate that I got to know Yanjin and Yicong, with whom I have experienced many adventures together. I am also grateful for the companion of my other friends and the joy we shared together: Zhoushen, Tiger, Dora, Tony, Yue, Jihang, Hao, Qian, and Guangming. I will always remember the fun music bands, full house dinner parties, late night games, and memorable skip trips with them.

Finally, I owe a huge debt of gratitude to my mother. She has always been supporting me unconditionally and encouraging me with her best wishes. And last but most definitely not least, I would like to thank my girlfriend, Karen. Her companion and

support in the past year were invaluable and will be forever remembered.

Chapter 1, in full, has been submitted for publication of the material. Chen, Wei, Xiaohua (Anny) Wei, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.

Chapter 2, in full, has been submitted for publication of the material. August, Terrence, Wei Chen, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is currently being prepared for submission for publication of the material. Chen, Wei, Vish Krishnan, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.

VITA

- 2006 Bachelor of Science, Management Information Systems, Harbin Institute of Technology, China
- 2008 Master of Science, Management Science and Engineering, Harbin Institute of Technology, China
- 2007-2009 Research Assistant, School of Hotel and Tourism Management, Hong Kong Polytechnic University, Hong Kong
- 2009-2015 Teaching Assistant, University of California, San Diego
- 2013 Summer Teaching Scholar, University of California, San Diego
- 2016 Doctor of Philosophy, Management, University of California, San Diego

PUBLICATIONS

Ye, Q., Law, R., Gu, B., and Chen, W. 2011. The Influence of User-Generated Content on Traveler Behavior: An Empirical Investigation on the Effects of E-Word-of-Mouth to Hotel Online Bookings. *Computers in Human Behavior* **27**(2) 634–639.

Law, R., Ye, Q., Chen, W., and Leung, R. 2009. An Analysis of the Most Influential Articles Published in Tourism Journals from 2000 to 2007: A Google Scholar Approach. *Journal of Travel & Tourism Marketing* **26**(7) 735–746.

ABSTRACT OF THE DISSERTATION

Incentives, Mechanisms and Platform Design for IT-enabled Open Innovation: Lessons
from Online Communities and Open-Source Software

by

Wei Chen

Doctor of Philosophy in Management

University of California, San Diego, 2016

Professor Kevin Zhu, Chair

In the past decade, information technology (IT) has changed how we produce knowledge and goods towards an open and collaborative way. At the same time, new issues challenge how we manage this open model. We need a better understanding of the incentives of the participants, as well as the mechanisms that impact these incentives, to design better such IT systems that enable the open production model. This dissertation is a first step to address these issues by learning from open-source software and online communities, focusing on the incentives, mechanisms, and design of IT systems.

Chapter 1 examines how motivating mechanisms, such as badges, votes, and status systems, *dynamically* engage the users and impact their contributions. We find that the effect of motivating mechanisms on user motivation is not uniform across users over time. For example, we find that badges could suffer a *negative* effect when a user is already engaged. Chapter 2 focuses on how open-source licensing affects competition among open-source and proprietary software. OSS license schemes serve as a means to govern the intellectual property. Distinct licenses each have different restrictions, which in turn affect the incentives of contributions. We build a game-theoretical model to understand how license restrictiveness impacts investments and the quality of services brought to market. Chapter 3 studies release frequency as a coordinating device in the adoption and evolution of OSS projects. We find that release frequency has a curvilinear relationship with both download and community contribution. Furthermore, when the costs are high, fast release frequency may decrease the consumers' incentive to adopt and even exhaust the community's incentive to contribute.

This body of work helps understand the incentives of individuals and firms who participate in the open production model. Our results offer managerial implications to firms that are formulating strategies to participate in such open production models, and provide insights for the design of IT platforms that enable the open models.

Dissertation Overview

Information technology (IT) has changed how we produce knowledge and goods in the past decade. Knowledge production has become more open and collaborative. For instance, after about 15 years of production, Wikipedia now includes about 4.8 million articles in English and well over 35 million in all languages¹, which put it one of the six most visited site in the world. Scientific projects have started to involve the general public (the “crowd”) in research. The crowd has helped discover new classes of galaxies and relevant protein structure for HIV transmission (Sauer mann and Franzoni 2015). Companies are also exploiting new technologies to support user-to-user assistance (Lakhani and von Hippel 2003) and generate new ideas for products (Bayus 2013).

Beyond generating knowledge and ideas, organizations are also growingly produce products and services in an open and collaborative way. It begins with open-source software (OSS), where thousands of developers all over the globe collaborate online to produce quality software. Now these OSS systems support a big portion of web servers (e.g., Linux, Apache, and MySQL) and mobile devices (Android) in the world. And many new software products/services contain more or less some open-source components. This open model of production has also spread to other types of products and services. For example, Lego now produces several lines of toys with lead users

¹ http://en.wikipedia.org/wiki/History_of_Wikipedia, accessed May 2015.

(Antorini et al. 2012). Quirky.com relies on the community to develop consumer products (Kornish and Ulrich 2014). Recently, companies have begun to crowdsource solutions through online platforms such as InnoCentive.com and Kaggle.com (Lakhani 2008), to crowdfund new products (Burtch et al. 2013), and to utilize offline resources to disrupt industries in the sharing economy (The Economist 2013).

At the same time, new issues emerge as the open model becomes more popular. First, the open model usually involves several parties with different incentives. Consider the case of multisided platforms. Research shows that it is challenging to bring multiple sides onboard and satisfy their diverse, sometimes conflicting, interests (Hagiu 2014). For open-source software, it is now well understood that developers contribute for diverse motivations (Roberts et al. 2006). In general, participants in such open production may have different incentives (Ariely et al. 2009; Bénabou and Tirole 2006). However, how to effectively motivate them accordingly is still not clear. Therefore, to effectively utilizing the outside efforts, companies must develop a better understanding of the decision making process of different participants, as well as mechanisms that would appropriately influence the decisions towards common interests.

Second, due to the diverse motivations, the classic price mechanism may not be readily applied. Instead, we often have to consider or rely on other mechanisms to manage this process. When the participants are not paid, what mechanisms could engage the users more? When an entrepreneur releases an OSS product to compete with a dominant proprietary software vendor, how would the licensing affect the incentives of contributors and the quality of the product? When an OSS project innovates with an outside community, how should the project pace the speed so that it does not deplete the

community? Understanding the impact of these mechanisms could inform us in better managing the novel production models.

Third, since most of these new productions are enabled by IT platforms, the design of incentives and mechanisms in those platforms becomes important to facilitate successful outcomes. Designing such platforms is a socio-technical problem rather than simply a technical one, greatly influenced by the incentives of the participants. To design better such systems, we must develop a deeper understanding of the incentives of the participants, as well as the mechanisms that could affect those incentives.

In this dissertation, I attempt to address these issues by learning from two important contexts where the open production model originates: open-source software and online communities. And I focus on the three core aspects in my exploration: the incentives of different participants, the mechanisms that impact these incentives, and the design and use of the IT platforms that enable the open production model. I provide an overview of my research questions and goals below, and then expand on each topic individually in the chapter overviews section.

In the context of an online knowledge-sharing community, Chapter 1 examines how motivating mechanisms, such as badges, votes, and status systems, *dynamically* engage the users and impact their contributions. We find that the effect of motivating mechanisms on user motivation is not uniform across users over time. The same motivating mechanism could work differently or even oppositely when the same user is in different states. For example, we find that badges are effective to engage a low-motivation user, but may suffer a *negative* effect when a user is already engaged. Our

results offer implications for the design of online communities that rely on the crowd to produce knowledge.

Chapter 2 focuses on how open-source licensing affects competition among open-source and proprietary software. Open-source software is actually distributed under certain OSS license schemes as a means to govern the intellectual property. Distinct licenses each have different restrictions on the use and modification of the software as well as its derivatives (Laurent 2004). And these restrictions in turn affects the incentives of strategic contributions. In this chapter, we build a game-theoretical model to understand how license restrictiveness impacts equilibrium investments and the quality of services brought to market. These results offer managerial implications to software firms that are formulating strategies to participate in OSS and provide insights to policy makers in the design of appropriate policies that govern intellectual property rights associated with OSS.

While Chapter 2 focuses on the impact of OSS licensing, Chapter 3 examines how the innovation of an OSS project would affect the users and the community in the co-creation of OSS. Specifically, we study release frequency as a coordinating device in the adoption and evolution of OSS projects. We find that release frequency has a curvilinear relationship with both download and community contribution. Furthermore, when the costs are high, fast release frequency may decrease the consumers' incentive to adopt and even exhaust the community's incentive to contribute. Software teams must understand that they need to manage the open-source community carefully: while sharing of information does not necessarily deplete the OSS resource, providing feedback by testing and bug-fixing requires effort, and can deplete the OSS community. These findings have

implications for managing the co-creation process of OSS and offer insights for the emerging open model of production.

Chapter Overviews

In the following, we expand the discussion of the three chapters that comprise this dissertation.

Chapter 1 studies the design of motivating mechanisms under dynamic user contributions in online communities. We characterize individual-level dynamics in a hidden Markov model with two latent motivation states (high vs. low), and examine the influence of different mechanisms with a structural model. From Bayesian estimation on user-level panel data, our results show that reputational motivations are important to transfer users to the high motivation state. For example, if a user were to earn one more accepted answer, the probability that the user stays in the high motivation state would increase by 4.9%. Further, we find that the same motivating mechanisms may work differently in the two motivation states. Surprisingly, badges may suffer the “moral licensing” effect when a user is already highly motivated, even though they help transfer low-motivation users to the high-motivation state. In addition, highly motivated users are more responsive to community size and the demand for knowledge. Design simulations on our structural model provide insights into the consequences of changing specific motivating mechanisms. Our findings offer guidance to platform designers on how to motivate community contributions and build sustainable online communities.

Chapter 2 studies the role of open-source licensing on the strategic contributions in the competition among open-source and proprietary software. Open-source software

(OSS) firms are increasingly using service-based business models to compete with established proprietary software firms. Because other members of the open-source community can strategically contribute to OSS and compete in the services market, the nature of competition between OSS and proprietary software firms is becoming more complex. Further, their incentives are strongly influenced by the licensing schemes that govern OSS. We study a 3-way game with strategic contribution from the community and focus on how open-source licensing affects competition among an open-source originator, open-source contributor, and a proprietor competing in the same software market. In this regard, we examine: (i) how quality investments and service prices are endogenously determined in equilibrium, (ii) how license restrictiveness impacts equilibrium investments and the quality of services brought to market, and (iii) how license restrictiveness affects consumer surplus and social welfare. Although some in the open-source community often advocate restrictive licenses such as GPL, because it is not always in the best interest of the originator for the contributor to invest greater development effort, such licensing can actually be detrimental to both consumer surplus and social welfare when it exacerbates this incentive conflict. We find such an outcome to be the case in markets characterized by software providers with similar development capabilities. In contrast, when their capabilities are more dispersed, a more restrictive license can instead encourage greater effort from the OSS contributor, lead to higher OSS quality, and provide a larger societal benefit. These results offer managerial implications to software firms that are formulating strategies to participate in OSS and provide insights to policy makers in the design of appropriate policies that govern intellectual property rights associated with OSS.

Chapter 3 investigates how release frequency of OSS would affect the community contribution and adoption. A central virtue of open-source software (OSS) is the contribution from the communities, yet our knowledge of how to coordinate and maximize the benefit of such contributions is limited. In this paper, we study the impact of release frequency as a coordinating device in the adoption and evolution of open-source projects. We first build a stylized model to characterize an OSS project as a two-sided market, with the two sides of consumers and community developers. Our model predicts that release frequency should have a curvilinear (inverse-U) relationship with both adoption and community contribution. Our empirical analysis support the hypotheses. Releasing too often seems to backfire due to the subtle effects on the supply side: it may exhaust the community contribution. High adoption cost and development cost may attenuate the effectiveness of frequent release. Furthermore, if the consumers can benefit more from the community contribution, higher release frequency might be helpful. Meanwhile, our results also show that high release frequency may decrease the absorption of contribution by the OSS team. These results bring implications for managing technology-enabled collaboration in open-source communities and research on open-source software, open innovation, and software adoption.

References

- Antorini, Y. M., Muñiz, Jr., A. M., and Askildsen, T. 2012, March 20. Collaborating With Customer Communities: Lessons From the Lego Group. *MIT Sloan Management Review* (Spring 2012)
- Ariely, D., Bracha, A., and Meier, S. 2009. Doing Good or Doing Well? Image Motivation and Monetary Incentives in Behaving Prosocially. *American Economic Review* **99**(1) 544–555.
- August, T., Shin, H., and Tunca, T. I. 2013. Licensing and Competition for Services in Open Source Software. *Information Systems Research* **24**(4) 1068–1086.
- Bayus, B. L. 2013. Crowdsourcing New Product Ideas over Time: An Analysis of the Dell IdeaStorm Community. *Management Science* **59**(1) 226–244.
- Bénabou, R., and Tirole, J. 2006. Incentives and Prosocial Behavior. *American Economic Review* **96**(5) 1652–1678.
- Burch, G., Ghose, A., and Wattal, S. 2013. An Empirical Examination of the Antecedents and Consequences of Contribution Patterns in Crowd-Funded Markets. *Information Systems Research* **24**(3) 499–519.
- Hagiu, A. 2014, Winter. Strategic Decisions for Multisided Platforms. *MIT Sloan Management Review* **55**(2) 71–80.
- Kornish, L. J., and Ulrich, K. T. 2014. The Importance of the Raw Idea in Innovation: Testing the Sow’s Ear Hypothesis. *Journal of Marketing Research* **51**(1) 14–26.
- Lakhani, K. R. 2008. InnoCentive. com (A). *Harvard Business School Case* (608-170)
- Lakhani, K. R., and von Hippel, E. 2003. How Open Source Software Works: “Free” User-To-User Assistance. *Research Policy* **32**(6) 923–943.
- Laurent, A. M. S. 2004. *Understanding Open Source and Free Software Licensing*, (First Edition) O’Reilly Media.

Roberts, J. A., Hann, I. H., and Slaughter, S. A. 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science* **52**(7) 984–999.

Sauermann, H., and Franzoni, C. 2015. Crowd Science User Contribution Patterns and Their Implications. *Proceedings of the National Academy of Sciences* **112**(3) 679–684.

The Economist. 2013, March 9. The Rise of the Sharing Economy. *The Economist*

Chapter 1

Engaging the Wisdom of Crowds: Structural Analysis of Dynamic User Contributions in Online Communities

This chapter studies the design of motivating mechanisms under dynamic user contributions in online communities. We characterize individual-level dynamics in a hidden Markov model with two latent motivation states (high vs. low), and examine the influence of different mechanisms with a structural model. From Bayesian estimation on user-level panel data, our results show that reputational motivations are important to transfer users to the high motivation state. For example, if a user were to earn one more accepted answer, the probability that the user stays in the high motivation state would increase by 4.9%. Further, we find that the same motivating mechanisms may work differently in the two motivation states. Surprisingly, badges may suffer the “moral licensing” effect when a user is already highly motivated, even though they help transfer low-motivation users to the high-motivation state. In addition, highly motivated users are more responsive to community size and the demand for knowledge. Design simulations

on our structural model provide insights into the consequences of changing specific motivating mechanisms. Our findings offer guidance to platform designers on how to motivate community contributions and build sustainable online communities.

1.1 Introduction

In recent years, more and more organizations have begun to leverage the “wisdom of crowds” to facilitate collaborative innovation (Malone et al. 2010). In the digital age of knowledge economy, online communities have become an important way to organize such collective innovation (Boudreau and Lakhani 2009). Built on online platforms over Web 2.0 or social media, online communities go beyond the conventional closed R&D, and are often more efficient in bringing together large numbers of geographically dispersed individuals to spark novel ideas, collaborate on inventions, and accumulate knowledge in support of a common interest (von Hippel 2005). Even traditional established companies have started to reach out to user innovation through online communities (Bayus 2013).

Despite the high expectation of online communities, many failed to achieve critical mass because they lacked a sustained participation of users (Ransbotham and Kane 2011). Recent evidence shows that online communities face a common challenge of sustainability: user participation tends to decline over time. For instance, the number of contributors in Wikipedia has been decreasing (Simonite 2013); only 33,276 users contributed in March 2013, a decline of more than a third from 56,400 during the peak in 2007 (Halfaker et al. 2013). Further, contributors commonly switch between the state of intensive contribution and the idle state (Sauer mann and Franzoni 2013). Motivating

users with the wrong mechanism at the wrong time may actually drive them away. Therefore, it is important to understand the dynamics of user contributions in order to design sustainable online communities.

Existing literature that examines why users contribute voluntarily in online communities mostly focuses on the static case, which seems inadequate to capture the dynamic user contributions (Kane et al. 2014). Recently, a new line of literature emerges to address the dynamics on the community level (Faraj et al. 2011). Yet, it is not clear how the individual-level dynamics aggregate up to the community level. Further, prior research studies the effects of user commitment and different motivating mechanisms on user contributions (Kraut et al. 2012). But it remains unclear whether such mechanisms are effective under dynamic user contributions.

Given the state of the literature, we seek to study three research questions: (1) How do we model individual-level dynamics in user contributions? (2) What is the impact of different motivating mechanisms on user contributions while users exhibit different states in the dynamics? (3) How can we better design these mechanisms so as to motivate user contributions effectively?

To address these questions, we propose a structural econometric model that integrates a *hidden Markov model* (HMM) into the public goods framework. This structural approach characterizes user contributions under different motivation states (i.e., *high* and *low*) and the transition between the states. With this model, we examine the effect of specific motivating mechanisms on user contributions under different states. We then empirically evaluate our structural model by Bayesian estimation with data collected from the knowledge-sharing community StackExchange. Although our empirical setting

is unique to StackExchange, our goal is to generalize the findings beyond this context to capture the dynamics of user contributions in online communities in general.

We find that (1) community size and the demand of knowledge stimulates user contributions, especially when the user is in high motivation state; (2) the same motivating mechanisms work differently in high vs. low motivation states; (3) community interactions, especially reputational motivations, are important for users to transition into to the high motivation state; (4) badges are effective to transfer low-motivation users to the high motivation state, but badges may suffer the “moral licensing” effect when a user is already in the high motivation state.

Our paper has the following features. First, our methodology advances the modelling approach by providing a structural model of individual-level user dynamics in online communities. We use the public goods model to formalize the effect of different motivating mechanisms under different states of motivation. By distinguishing motivation states and estimating state-dependent contributions, we are able to explicitly characterize the dynamics of user contributions at the individual level. Further, our approach does not rely on the types of motivating mechanisms in the specific empirical setting. It is applicable to a wide range of online communities.

Second, we provide insights into an increasingly important mode of organizing innovation based on collective intelligence. Our results offer implications for the design of online communities that rely on the crowd to produce knowledge. We find that different motivating mechanisms drive user dynamics (state transitions) differently, which helps us identify mechanisms that are effective. The hidden states also allow us to classify users in real time and help community managers target the right kinds of users

with observational data. Simulations on our structural model also enable community managers to design effective mechanisms.

1.2 Literature Review

Many online communities rely on voluntary contributions from members to produce content and accumulate knowledge (Malone et al. 2010). Because the end product is often open and free but only the contributors incur the production costs, these online communities can suffer from free-riding and thus under-provision problems, i.e., the “tragedy of the commons” (Hardin 1968). In general, voluntary cooperation is inherently fragile, even if most people are conditional cooperators (Fischbacher and Gächter 2010). Indeed, we observe the decline of user contributions in online communities (e.g. Bayus 2013, Simonite 2013). Hence, sustaining voluntary contributions poses a key challenge for online communities. It requires effective institutional designs to enhance continued contributions over time.

1.2.1 Dynamics of User Contribution

The dynamics of user contributions can be characterized at either the community level or the individual user level. On the community level, the dynamics may come from membership turnover (Butler 2001). Recently, Ransbotham and Kane (2011) find that more turnover may be better for the community at the knowledge-retention stage of the life cycle. However, it remains unclear how dynamics at the aggregated level may come from individual-level behaviors, and what mechanisms that community designers can use to promote the desired outcome.

To narrow this gap, we focus on the dynamics of user contributions at the individual-level. Acknowledging that a contributor may eventually leave the community, we seek to understand how to motivate users to contribute more when they are still in the community. Further, while the change-retain tension and membership turnover are more relevant to changing and defending *existing* knowledge, we focus on user contributions to *new* content in our context (Kane et al. 2014). Our perspective of individual-level dynamics is similar to that of Sauermann and Franzoni (2013), who study how the interest-based motivation affects the participation dynamics in crowd-based knowledge production. They characterize interest as a psychological state. We extend their work by modeling motivation states in a formal HMM model, which enables us to infer the motivation state of a user. Our model also incorporates the user interactions with the community to explain the dynamics of user contributions.

1.2.2 Engaging User Contributions

Commitment and Engagement

User commitment is an important construct in the studies of user contributions in online communities (Kraut et al. 2012). User commitment characterizes a psychological bond between an individual and the community (Ren et al. 2012). Different forms of commitments are found to associate with different types of behaviors (Bateman et al. 2011). Another similar construct is user engagement, which is defined as a proactive psychological state geared toward contributing (Ray et al. 2014). The findings on these constructs suggest a direction to model user motivation states in our HMM model.

Given the psychological nature of commitment and engagement, relevant studies measure them with survey data (e.g., Bateman et al. 2011, Ray et al. 2014). However, these constructs are hard to quantify with consensus from the community manager's standpoint. Further, commitment/engagement may also be subject to individual dynamics. The level of commitment of a user may fluctuate, depending on his interaction with the community. It is costly to survey a large number of users over time to reveal this dynamics. Instead, we use observational data to infer motivation states and characterize individual dynamics. This approach extends the literature and enables community managers to estimate the dynamic motivation states of all users.

Motivating Mechanisms

Online communities employ various mechanisms, such as points, badges, levels, and status that appeal to users to enhance engagement and commitment (Burke 2011). It is then useful to understand how users respond to these mechanisms. Studies have distinguished the role of intrinsic motivations and extrinsic motivations (see von Krogh and von Hippel 2006 for a review), and reveal the effectiveness of various motivations, for example, the enjoyment of social image (Ren and Kraut 2011), self-efficacy (Kankanhalli et al. 2005, Ray et al. 2014), and entertainment through social interactions (Ren and Kraut 2011). Among them, we focus on reciprocity and reputational motivation in our research context.

First, reciprocity means making valuable contributions to the community for mutual benefit (e.g., Chiu et al. 2006, Faraj and Johnson 2011). The literature suggests that users who have received others' help tend to return the favor as they have benefited

from the experience and knowledge. Reciprocity is shown to motivate developers to perform mundane tasks in open source software development (Lakhani and von Hippel 2003). Likewise, StackExchange users whose questions have been answered by others may be more likely to answer others' questions in return.

Second, reputational motivation is another critical factor driving users to contribute. People care about their self-image, and the way others perceive them (Bénabou and Tirole 2006). For example, studies show that “ego-boo” is important to drive participation in social media (Toubia and Stephen 2013) and open source software development (Raymond 1999). Social comparison has been shown to improve the contributions of users whose contributions are below the median (Chen et al. 2010). Similarly, users on StackExchange may care about the evaluation of their peers.

Although reputational motivations can enhance contribution, they might also render outcomes in the opposite direction. While Chen et al. (2010) show that social comparison encourages users below the median to contribute more, they also find that users above the median could decrease their contributions to conform to the social norm. Faraj and Johnson (2011) suggest that social identity in online communities can come with either a negative or positive consequence. The behavioral economics literature documents that costly prosocial behavior can lead to the “moral licensing” effect, which states that people may feel justified to behave non-prosocially when they have done something pro-social (Gneezy et al. 2012). Having contributed to the online community and been endorsed by reputation points and badges, users may feel licensed not to contribute subsequently.

Despite the substantial literature on motivating mechanisms, little is known about whether and how their effectiveness changes with the fluctuations of commitment or engagement. Answers to these questions would inform platform designers of implementing mechanisms that better induce desirable motivation states and contributions. To narrow this gap, we use a public goods model to formalize the effect of various motivating mechanisms under different states, so as to characterize individual-level dynamics.

1.2.3 Model the Dynamics of User Contribution

One challenge of capturing individual-level dynamics is that the structure of such dynamics is usually unobserved. To capture this latent structure of dynamics, discrete state space structure is a useful approach in the choice modelling literature (e.g., Heckman 1981). For example, an individual's present decision depends on his previous decision. In most of these models, the states are observable (e.g., brand switching of customers). A limitation of the observed state models is that they tend to ignore other dynamics that could contribute to the change of states. In many other scenarios, however, we cannot observe the underlying states that drive the individual-level dynamics, e.g., motivation states in our research context. In this case, the hidden Markov model (HMM) can be useful.

An HMM is a stochastic process that consists of three elements: a finite set of hidden states, observed outcomes conditional on the hidden state, and the probabilities of transitioning from one state to another. It has wide applications in modelling signal processing, speech recognition, biology, business cycles, and stock market volatility (e.g.,

Hamilton 1989). Recently, HMM is also adopted to study promotion dynamics (e.g., Moon et al. 2007) and customer relationship management (Netzer et al. 2008) in marketing, and the learning of developers in open source software (Singh et al. 2010). As far as we are aware, it has not yet been applied to modelling user contributions in online communities.

Figure 1.1 summarizes our HMM-based structural model. It illustrates how a user could switch between motivation states through various motivation schemes, and how his contribution probability depends on his state. Specifically, our HMM model has three elements, as shown in Figure 1.1:

(1) We model users with two hidden *motivation states*: high and low. The states capture the strength of motivation to contribute. At any time t , a user is in only one state.

(2) From time $t-1$ to t , the user could switch between the two latent states with certain probabilities, which are affected by the user's interaction with the community, such as how his contributions are evaluated by the peers.

(3) Conditional on his state in t , a user may respond differently to community and individual characteristics (e.g., size of the community and the demand for knowledge). We can observe this state-dependent response as his amount of contributions in t .

1.3 Research Context

We study our research questions in a representative online community StackExchange (stackexchange.com), which is a large network of knowledge-sharing platforms based on Wikipedia-style voluntary contributions. It started in 2008 with StackOverflow, a question-and-answer website on programming. Now it has expanded to

more than 100 sub-sites covering widespread technical (e.g. math, Tex) and non-technical (e.g. cooking, bicycle) topics. On each sub-site, users ask topic related questions and provide answers. Users can also vote, comment, revise, or even remove questions and answers as they do in Wikipedia, which allows the community to improve the content collectively.

StackExchange employs various mechanisms to encourage contribution and to maintain the high quality of questions and answers. For example, when a user receives 10 up-votes on one of his answers, he earns a “Nice Answer” badge. If the answer receives more than 40 up-votes and is accepted by the question poster, the answer provider will be rewarded a “Guru” badge. In our sample, StackExchange has 158 types of badges and has awarded them 414,761 times. With up-votes, a user also earns reputation points, which grant him new privileges when his points reach certain thresholds. The badges and points display right below the user name on his profile page. These mechanisms serve as important channels for the users’ identity verification (Ma and Agarwal 2007).

The features of StackExchange help us understand user behaviors when innovation is organized in a voluntary online community. First, StackExchange provides detailed data about user interactions. For example, we can observe when a user receives an up-vote on his answer, and whether his answer has been accepted. The fine-grained user-level data help us identify the effect of different interactions on users’ transition probabilities. Second, although our context is a knowledge-sharing community, our analysis could be generalized in a broader sense, as many other online communities are using similar motivating mechanisms. For example, peer voting is used in crowdsourcing

ideation initiatives (Huang et al. 2014), and the badge system is one important device in many online communities (Piskorski et al. 2010).

1.3.1 Data

Our data comes from SuperUser (superuser.com), a sub-site of StackExchange, for computer enthusiasts and power users. We employ SuperUser because of its data quality and user incentive concerns. First, SuperUser is the third largest sub-site on StackExchange by the number of contributions. Hence the site has rich information on user interactions that can help us study our research questions. Second, contributors to programming related sites such as StackOverflow may contribute owing to career concerns. That is, high reputation on StackOverflow could signal their technical competency, which may make it easier for future job-hunting. This signaling motivation may crowd out other incentives of prosocial behaviors (Ariely et al. 2009).

SuperUser was launched in July 2009, and has accumulated about 214,000 questions and over 351,000 answers by April 2014. We collected detailed data about daily activities of each user from July 12th 2009 to March 1st 2012 (964 days). We only include users who contributed at least 10 answers during the sample period. Because these users make a majority of the contributions (over 80%), it is critical to understand their motivations and dynamics (Sauermaun and Franzoni 2013). Our full sample contains 2,147 users who have contributed 127,360 out of the 157,375 answers in our data.¹

¹ This amount of answers could require more than 26,200 hours of work, assuming each answer takes 10 minutes on average. This estimate is conservative since many users would need to do some coding in order to provide an answer, which may take more time.

1.3.2 Community and User Level Trends

We first demonstrate the general trends of the data in Figure 1.2. Except the surge around the launch of the website and a slight decline after 500 days, the numbers of new questions (panel-a) and answers (panel-b) are relatively stable over time. Similarly, the trends are stable for the numbers of badges and up-votes, as shown in panels (d) and (e), respectively. Panel (c) plots the number of accepted answers each day. The stable trend suggests that question posters deem the quality of answers being consistent over time. We also plot the average up-votes per answer in Panel (f). The trend is stable except for a decline at the initial stage. Overall, SuperUser is a relatively healthy community with steady contributions in our sample.

The contributions at the individual level, however, show a different pattern. Panel (a) in Figure 1.3 presents the average number of answers contributed by each user over time. The contribution decreases exponentially regardless of the length of time the user stays. The decreasing trend persists in all other sub-samples. Panel (b) in Figure 1.3 shows the histogram of contribution tenure, which is defined as the days between the first and last answers of each users. We can see significant heterogeneity in the time span during which users contribute. The average contributions decrease over time, but we still observe that some users stay for a long time in the community. Are these users naturally more likely to contribute, or do their interactions with the peers make them stay? Even though users may eventually leave the community, can we learn from their behaviors so that we might be able to mitigate the declining trend?

Figure 1.4 plots the contributions of five random users from our sample (user IDs anonymized). Each row shows the answers of a user over 200 days. Each point represents

the number of answers contributed by the user. The point is missing if the user does not contribute. We observe that even relatively active users exhibit substantial fluctuation of contributions during their tenure. They actively contribute for some time periods, while being idle for other periods. Our goal is to model the fluctuation of user contributions, and study the influence of different motivating mechanisms under different states of users.

1.4 Structural Modelling of User Behavior

In this section, we describe the details of our structural model with HMM, where a user interacts with the community and decides his level of contributions.

1.4.1 A Static Model of User Contribution as Public Goods

We use a public goods framework to model user contributions to an online community. In our research context, user contributions are public goods in nature, because they are voluntary, and are free and open. The key issue about public goods is free-riding, which means that everyone can share the benefits, but only the contributors incur the production cost. Naturally, under-provision is a common equilibrium in many pure altruism models (Andreoni 1988), where individual's utility comes only from the cumulative provision of public goods. But these models are not very helpful to explain why large groups, such as online communities, are able to attract substantial user contributions. Such discrepancy between theoretical models and empirical phenomena may be reconciled by *impure altruism models* (e.g. Andreoni 1990). In such models, individuals contribute because they may obtain utilities not only from pure altruism, but

also from their own private benefits, such as signalling personal skills or the fulfilment of helping others.

In our static public goods model, each self-interested user chooses how much to contribute. A user's net utility consists of three parts: (1) his valuation of the accumulated contribution (e.g., knowledge) in the community, (2) his valuation of his own contribution, and (3) his cost of contribution. The first part captures the benefit the user could obtain from the community, as suggested by the pure altruism literature. The second part intends to capture the impure altruism. The third part suggests that making contribution is costly in terms of time and effort.

Assuming additive separability of the above three parts, we specify the utility function of user i at time t as:

$$U_{it}(Y_{it}, X_{it}, W_{i,t-1}, \sum_{\tau=1}^t \sum_{j=1}^{N_t} Y_{j\tau}) = \gamma_i \sum_{\tau=1}^t \sum_{j=1}^{N_t} \delta^{t-\tau} Y_{j\tau} + f(X_{it}, W_{i,t-1}) Y_{it} - \frac{1}{2} c_i Y_{it}^2, \quad (1.1)$$

where Y_{it} is the contribution of user i at time t . Intuitively, a user gains utility from the accumulative knowledge in the community, and his own incremental contribution at present, net of his cost.

We choose functional forms following Chen et al. (2010). In the first term on the right hand side, γ_i is user i 's marginal benefit from the accumulated contribution of the community, N_t is the number of users in the community at time t , and δ is a discount factor of the old contribution. In the second term, $f(X_{it}, W_{i,t-1})$ captures user i 's valuation of his own contribution at time t . This could be viewed as a parsimonious version of the "image rewards" as in the prosocial behavior model of Bénabou and Tirole (2006), which will "depend on the informational and economic context, including what others are

doing.” Therefore, this valuation could change over time with X_{it} , which is a vector of community and individual characteristics. Essentially, in our model a user’s valuation of his own contribution could fluctuate because of his changing characteristics and interactions with peers in the community. The third term is the cost function. We use a quadratic cost function to capture the convex cost of contributions (Gu et al. 2007).

In equilibrium, the contribution of user i at time t is:

$$Y_{it}^* = \frac{\gamma_i + f(X_{it}, W_{i,t-1})}{c_i}. \quad (1.2)$$

For analytical tractability, we assume that the valuation $f(X_{it}, W_{i,t-1})$ follows a linear form. We then obtain:

$$Y_{it}^* = X_{it}' \beta_{s_{it}} + \varepsilon_{it}, \quad (\varepsilon_{it} | X_{it}, s_{it}) \sim N(0, \sigma^2), \quad (1.3)$$

where X_{it} is a vector of community and individual characteristics. We assume that the error term ε_{it} follows a normal distribution with mean zero and variance σ^2 . Our goal is to estimate the coefficient vector $\beta_{s_{it}}$, which captures the influence of vector X_{it} on the user’s contribution Y_{it} . Note that vector $\beta_{s_{it}}$ depends on user i ’s motivation state s_{it} , which is a feature of our model.

1.4.2 Motivation States in HMM

Our proposed HMM characterizes the fluctuation of a user’s contribution as two stochastic processes: a process of observed contributions, and an underlying unobserved process of the user’s motivation states. A user could have two hidden motivation states: *high* or *low*. We denote s_{it} the state of user i at time t :

$$s_{it} = \begin{cases} 0 & \text{if in low motivation state} \\ 1 & \text{if in high motivation state} \end{cases}$$

The hidden state captures the time-dependent feature of a user's valuation of his own contribution, i.e., the strength of his motivation to contribute. For example, if a user has high valuation of the contributions he provided to the community at time $t-1$ (i.e., in the high motivation state), he may also highly value his contributions at time t . Based on his state, a user responds differently to the community and individual characteristics (i.e., vector X_{it}). For example, if a user is in a high motivation state, he may be more likely to respond to the new questions in the community. The observed contributions could be regarded as a noisy signal of the hidden state process. The pair of processes – hidden state and observed contributions – together form a hidden Markov chain (Rabiner 1989).

From time t to $t+1$, a user may stay in one state, or switch to the other. In our HMM, the state process $(s_{it})_{i \geq 0}$ is characterized as a first-order Markov chain with state space $S = \{0, 1\}$. Together with Y_{it} , the observed contributions of user i at time t , we can model the vector-valued stochastic process (Y_{it}, s_{it}) as a hidden Markov chain. The probability of transition from one period to the next is then defined as:

$$P((Y_{it}, s_{it}) | (Y_{i,t-1}, s_{i,t-1})) = P(Y_{it} | s_{it}) p(s_{i,t-1}, s_{it}),$$

where $p(s_{i,t-1}, s_{it})$ is the transition probability from state $s_{i,t-1}$ to state s_{it} , and $P(Y_{it} | s_{it})$ is the conditional probability describing the state-dependent contributions. Given their importance in our model, we elaborate the transition probability and state-dependent contribution probability in the next two sub-sections, respectively.

1.4.3 Transition Probabilities of States

The unobserved states could switch between high and low motivation states. The transition matrix $P(s_{i,t-1}, s_{it})$ below characterizes the probability of such transitions:

$$P(s_{i,t-1}, s_{it}) = \begin{bmatrix} p(0, 0) & p(0, 1) \\ p(1, 0) & p(1, 1) \end{bmatrix},$$

where $p(j, 1)$ is the transition probability from state j ($j \in \{0, 1\}$) to the high motivation state ($s_{it} = 1$), and $p(j, 0) = 1 - p(j, 1)$. We assume that the transition probability $p(j, 1)$ is influenced by a user's interactions with the community, which may create certain social or personal norms for the user (Bénabou and Tirole 2006). The user may then evaluate his own contributions differently based on the norms. For instance, if all of his past contributions were voted up and appreciated, the user would be more likely to value his own contribution more and remain highly motivated. Otherwise, he may switch to the low motivation state.

We model the transition probabilities with a *probit* model (Wooldridge 2010). We assume that the *states* are determined by a latent propensity of transition L_{it} :

$$L_{it} = W'_{i,t-1} \xi_{s_{i,t-1}} + u_{it}, \quad (u_{it} | W_{i,t-1}, s_{i,t-1}) \sim N(0, \sigma_u^2) \quad (1.4)$$

such that

$$s_{it} = \begin{cases} 1 & \text{if } L_{it} > 0 \\ 0 & \text{if } L_{it} \leq 0 \end{cases}$$

where $W_{i,t-1}$ is a vector of lagged variables related to the user's previous interactions with the community, $\xi_{s_{i,t-1}}$ is a vector of the corresponding coefficients, and u_{it} is a normal error term from the probit model. Note that $\xi_{s_{i,t-1}}$ is state-specific,

capturing different effects of $W_{i,t-1}$ under different states. Then we obtain the transition probability as follows:

$$\begin{aligned}
p(j, 1) &= P(s_{it} = 1 | s_{i,t-1} = j, W_{i,t-1}) \\
&= P(L_{it} > 0 | s_{i,t-1} = j, W_{i,t-1}) \\
&= P(W'_{i,t-1} \xi_j + u_{it} > 0 | s_{i,t-1} = j, W_{i,t-1}) \\
&= \Phi\left(\frac{W'_{i,t-1} \xi_j}{\sigma_u}\right),
\end{aligned} \tag{1.5}$$

where Φ is the standard normal distribution function. When a user first joins the community, we assume he has an initial probability p_0 to be in high motivation state and a probability $1 - p_0$ to be in low motivation state.

1.4.4 State-Dependent Contributions

Given the states above, we now derive the conditional probability $P(Y_{it} | s_{it})$. Since the observed user contributions are non-negative, we adopt the standard *Tobit* model (Wooldridge 2010):

$$Y_{it}^* = X'_{it} \beta_{s_{it}} + \varepsilon_{it}, \quad (\varepsilon_{it} | X_{it}, s_{it}) \sim N(0, \sigma^2),$$

$$\text{and } Y_{it} = \max(0, Y_{it}^*),$$

where Y_{it} stands for the observed contributions. Then the state-dependent contributions would follow the distribution below. The probability of making no contribution is

$$\begin{aligned}
P(Y_{it} = 0 | X_{it}, s_{it}) &= P(Y_{it}^* < 0 | X_{it}, s_{it}) \\
&= \Phi\left(-\frac{X'_{it} \beta_{s_{it}}}{\sigma}\right) \\
&= 1 - \Phi\left(\frac{X'_{it} \beta_{s_{it}}}{\sigma}\right).
\end{aligned}$$

For $Y_{it} > 0$, the probability density function is $\frac{1}{\sigma} \phi\left(\frac{Y_{it} - X'_{it}\beta_{s_{it}}}{\sigma}\right)$, where ϕ is the standard normal density function.

1.5 Analysis and Discussion

1.5.1 The Estimation and Identification

We estimate the state-dependent contribution parameters $\beta_{s_{it}}$ in equation (1.3), and the transition matrix parameters $\zeta_{s_{it}}$ in equation (1.4). Since $s_{it} \in \{0,1\}$, we essentially estimate the parameter vectors $\beta = (\beta_0, \beta_1)$ and $\zeta = (\zeta_0, \zeta_1)$, where β captures the effect of community and individual characteristics on the contribution behavior, and ζ captures the influence of community interactions on the user's state transition probability. To estimate these key parameters, we also need to estimate the standard deviations σ and σ_u , as well as the state process $S = \{s_{it}\}_{t=1,\dots,T; i=1,\dots,N_t}$. For ease of reference, we write the parameter space as $\theta = \{\beta, \zeta, \sigma, \sigma_u\}$ and S . Note that β and ζ are state-dependent, while σ and σ_u are not.

We estimate our HMM using a Bayesian procedure developed by Kim and Nelson (1999). The Bayesian estimation algorithm treats θ and S as random variables with prior distributions. The algorithm then updates their joint distributions $\pi(\theta, S | Y, X, W)$ using Gibbs sampling (Albert and Chib 1993). This updates the posterior distribution by incorporating the observed information from data (Technical details of the sampling algorithm are in the appendix).

Bayesian estimations of HMM models may encounter the “label switching” problem (Jasra et al. 2005), which means our posterior distribution of θ and S may be invariant if we switch the label 0 and 1. Since the *high* and *low motivation* states in our context have self-evident economic interpretation, we adopt a normalization requirement, i.e., the constant term in β_0 is smaller than that in β_1 in each draw of our Gibbs samplers. This requirement means that without any stimulus, a user in the high motivation state on average contributes more than if he were in the low motivation state. This technique helps us identify the two states in our model.

1.5.2 Samples and Variables

To test our structural model, we construct a user-date panel of the 2,147 users in 964 days in our sample from SuperUser. Because of computational burden (over 2 million data points), we divide the sample into sub-samples that each contains 100 days. Our estimation focuses on a subsample from the first 100 days with 561 unique users, and 44,271 user-date observations. Table 1.1 presents the definitions of our variables and summary statistics. We use other sub-samples for robustness checks.

Among various ways to contribute on the community, providing answers may be the most crucial because of the knowledge-sharing nature of the site. It is also the most challenging activity as it takes time and effort and requires certain domain knowledge. Therefore, our dependent variable is $Answers_{it}$, which is the number of answers provided by user i at time t .

We categorize two sets of explanatory variables that may affect users’ transition probabilities (W) and conditional contributions (X), respectively. The variables in vector

We capture the social interactions that could have an enduring effect on a user's motivation state. First, if a user's questions are answered by others, he may be more likely to return to the site and may have higher chance to contribute just because he is on the site. Moreover, he may be more likely to answer others' questions out of reciprocity. We use the number of answers a user receives on his past questions ($Answers_received_{i,t-1}$) to capture such reciprocity. Second, reputational motivations can play a role in state transitions. When more answers provided by a user are voted up or accepted as the best answer, one may value his own contribution higher because the contribution is appreciated by the community. This may transfer the user to the high motivation state so that he contributes even more. We measure these effects by the number of up-votes that a user receives on his previous answers ($Upvotes_answer_{i,t-1}$) and by the number of accepted answers of a user ($Accepted_answers_{i,t-1}$). To examine the effect of the badge system, we include $Badges_{i,t-1}$ as another explanatory variable, which represents the incremental number of badges earned by user i for his answers at time $t-1$.

The variables in vector X capture community characteristics that may have a direct effect on a user's valuation of his own contribution. For example, $New_questions_t$ is the total number of new questions at time t , which can signal the demand for knowledge. $Group_size_t$ is the number of users who participate in any activities at time t , which we use as a proxy for community size. Classic public goods models show that the average level of contribution *decreases* with group size, while in impure altruism models, the private benefits can increase with group size, as the enjoyment of contributing is enhanced by the number of recipients. We call this the *social effect*. As a group becomes larger, the motivation of pure altruism can decrease, while the social effect can increase.

The social effect is used to explain why individual contributions in large groups can be sustained (Zhang and Zhu 2011). Given the importance of group size, we include it as a contextual factor in our analysis.

The vector X also contains variables of individual characteristics. These time-variant variables control for individual heterogeneity. For instance, the days since user i registered ($Tenure_{it}$) account for the declining trend of contributions. We also include the total number of answers that have been provided by the user ($Total_answers_{i,t-1}$). The rationale is that if a user has provided more answers in the past, he may be more inclined to provide new answers in the current period.

1.5.3 Estimation Results

Table 1.2 reports the posterior means and standard deviations of the structural model based on Bayesian estimation. The coefficients in vector β and ζ vary across states (the two columns), indicating that a change in states could lead to a change in the contribution behavior. The initial probabilities of being in low vs. high motivation states are 0.856 and 0.144, respectively (bottom row). Hence, a user is more likely to be in the low motivation state when he joins the site as a new member. This confirms the importance of study how to energize and motivate community members. For ease of discussion, we refer high and low motivation states as H and L , respectively.

State-dependent Contributions (β)

We first examine the state-dependent contributions (i.e., X_{it} , top panel). The interpretation of the two states is mainly determined by the state-specific intrinsic

propensity to contribute (the parameter c_1), as we discussed in the estimation section. The estimates of c_1 are -1.235 and 2.221 for the L and H motivation states, respectively (both significant at 1% level). This shows that on average a user is much less likely to contribute when he is in the L state.

The coefficients of $New_questions_t$ indicate the response of users to the demand of knowledge. We can see that users respond differently under different states. A highly motivated user may be more responsive to new questions (0.007, significant at 1% level), while a less motivated user may be not (0.0002, insignificant). Hence, users in state H are more responsive to the demand of knowledge, and they supply more knowledge when the need arises.

Regarding $Group_size_t$, the coefficient is positive and significant (both at 1% level). A user may contribute more when the community is larger, which confirms the “social effect” discussed above. Meanwhile, this effect is much larger when a user is in state H than in L (coefficients 0.025 vs. 0.004). This means that when a user is in state L , the social effect is weaker. Compared with him being in state H , his valuation of his own contribution would increase less with the group size.

For individual characteristics, we find a negative relationship between *Tenure* length and user contributions. The coefficient is -0.006 and -0.013 for H and L motivation states, respectively (both significant at 1% level). Users who have been a member for a longer period of time on average contribute less. A possible explanation is that the longer a user has been associated with the community, the more inertia (or lower incentives) he has in terms of contribution, while rookies or fresh members may tend to be more

engaged and contribute more. Such a *stalling effect* poses a significant challenge to online communities.

We also confirm the positive relationship between past answers and the probability to contribute under both motivation states (both coefficients of $Total_answers_{i,t-1}$ are significant at 1% level).

State Transition Probabilities (ξ)

We now turn to the effects of different motivating mechanisms on state transition probabilities (i.e., $W_{i,t-1}$, bottom panel in Table 1.2). In the absence of motivating mechanisms, all users are likely to transition into the state L (c_2 is negative in both states, significant at 1% level). However, less motivated users are more likely to remain in L : the constant term has a more negative coefficient (-2.555 vs. -0.248).

Reciprocity seems to have different effect across the two states. For users in state L , receiving more answers on their previous questions tends to transfer them into state H (coefficient 0.040, significant at 1% level, row 2). This is good news. In contrast, the coefficient is insignificant for highly motivated users. These users may tend to contribute anyway, not because they want to return the favour of their peers. Hence, the *reciprocity* of providing answers can be more useful to stimulate users in the L state. This differential effect seems a useful finding for the online community organizers.

The coefficients on $Upvotes_answer_{i,t-1}$ and $Accepted_answers_{i,t-1}$ are both positive and highly significant (except $Upvotes_answer_{i,t-1}$ in the high motivation state). We interpret these as the result of the reputational motivation, or “*image rewards*.” When a user receives more up-votes or has more accepted answers, his reputational motivation

is satisfied and his inference about the value of his own contribution thus increases. Further, this “image reward” is more prominent in state L . For instance, $Upvotes_answer_{i,t-1}$ significantly increases the likelihood of transitioning to state H for a user in state L . The pattern is similar but even more evident for $Accepted_answers_{i,t-1}$. Together, these results highlight the effectiveness of *image reward* as a motivating scheme, especially for users in state L , which is an issue we really care about in this research.

Likewise, earning more badges on answers seems to transition a user in state L into state H (0.169, significant at 1% level, row 5). Surprisingly, its effect becomes negative for users in H (-0.076, significant at 1% level). This seems to suggest that earned badges may not help to keep users engaged in the high motivation state. This may be due to the “moral licensing” effect of pro-social behavior. If so, using badge system to motivate user contributions should be gauged carefully, especially considering the fact that badges are widely used in many online communities. This could be an interesting area for future research.

1.5.4 Transition Matrices and Marginal Effects

To better understand the effect of reputational motivations on users’ transition probabilities, we substitute the estimates into equation (1.5) to calculate the probabilities. Transition matrix (a) in Table 1.3 presents the transition probabilities evaluated at the mean level of community interactions (from column “Mean” in Table 1.1). The transition probabilities are substantially different when a user is in low vs. high motivation states. This confirms that modelling the stochastic process with the two hidden states is

reasonable. Further, the matrix indicates the stickiness of state L . Once a user is in this state, he is most likely to be trapped, and even if a user starts off in state H , he also tends to slip down to state L . This implies the challenge of inherent deteriorating participation as we posed earlier, and the importance of stimulating users to become more motivated.

To quantify the marginal effect of each reputational motivation on transition probability, we first calculate the transition probabilities when the mean value of a variable increases by one unit, while holding other variables constant. The matrices (b) – (d) in Table 1.3 show the transition probabilities caused by such a change for *up-votes*, *accepts* and *badges*, respectively. We focus on up-votes, accepts and badges, because they are the mechanisms that platform designers could manage. For example, if the community decreases the cost of up-votes or even enhances the incentives of up-votes, the number of up-votes is likely to increase. If the platform designer changes the setup such that each question could accept multiple answers, then the mean of accepted answers is likely to increase. Further, because online communities provide various kinds of badges to users, a more careful design of the badge system may help elevate the user contributions.

We then take the difference between respective cells of (a) and (b) – (d) to get the marginal effect on transition probability. For example, in matrix (b), receiving one additional *up-vote* on average increases the probability of transitioning to state H for a user in state L from 0.7% to 0.9%, while a user in state H would increase his likelihood of staying in the state from 41% to 43%. Similarly, in matrix (c), each one additional *accepted answer* could lift the transition probability to state H by 2.3% and 4.9% for users in low and high motivation states, respectively. These changes are non-trivial

because the low motivation state tends to be sticky. With more than 200 up-votes and 30 accepted answers each day on StackExchange, the effects of these mechanisms significantly enhance the contributions at the community level.

1.5.5 Design Simulations

We turn to the normative perspective using design simulations.² We do three simulation experiments to see if platform designers can encourage more contributions by strengthening users' reputational motivations: up-votes received for answers, accepted answers, and badges. If it becomes easier to improve self-image and earn reputation through each of these channels, are users going to provide more answers? We hypothetically double the value of the variables $Upvotes_answer_{i,t-1}$, $Accepted_answers_{i,t-1}$, and $Badges_{i,t-1}$, making it twice as easy to enhance reputational motivation in each case. We then simulate, under each scenario, the evolution of the total number of answers in the community over time.

Figure 1.5 presents the simulation results, which are the average of 100 simulation iterations for each user on each date. Graphs (a) – (c) show the simulated total number of answers (dash lines) versus the actual total number of answers (solid line). In graphs (d) – (f), we plot the number of users who are in state H according to our algorithm. At any time, a user is classified as being in either the high or low motivation state (unobservable), which can be recovered from the posterior probability distribution. The

² These simulation experiments correspond to “counterfactual experiments” in the empirical industrial organization literature (Reiss and Wolak 2007). In a structural model, if we specify a counterfactual antecedent (an event/parameter different from the real observations), then we can evaluate the counterfactual consequent (a result that is expected to hold if the antecedent were true). This analysis is used for policy evaluation.

solid line shows the users in state H under the current design, and the dashed lines show the simulated users in state H if we were to change the corresponding motivation mechanism.

We discover three patterns in the simulation results in Figure 1.5. First, the simulated number of answers is greater than the actual data in all three cases. This means that when it becomes easier to receive reputational rewards (through up-votes, accepted answers or badges), users will contribute more. Second, up-votes and accepted answers seem to be more effective than badges, which may be due to the “moral licensing” effect of badges in high motivation state. However, as a design mechanism, badges are much easier to change than up-votes and accepted answers. To test the effectiveness of different badges, a platform designer could potentially examine the simulated experiment on many specific badges. Third, interestingly, the effect of each reputational motivation is much greater in later time periods ($t > 40$) than the initial or early periods. This poses a vital issue for the sustainability of online communities.

Our results show that while at the startup stage, strengthening reputational motivation may not promote contributions significantly. And yet it becomes more important to offset the adverse time trend to a large extent as the community grows. Together, our experiments imply that it is important for platform designers to manage reputational motivations, so as to encourage users to contribute, especially when the community passes the startup stage. Note that we are not suggesting a constant effect of these mechanisms; as we change the design of the community, the perception of the users may change accordingly. Rather, our design simulation serves as a direction for further explorations.

To further check whether the above patterns are consistent over time in different sub-samples, we conduct out-of-sample forecasting and simulation analysis. We examine the time period $t = 100 - 200$, and repeat the above simulations on the number of answers. In Figure 1.6, the solid lines plot the actual observation of answers in the time period. The dashed lines depict the forecasted answers given the different design of the mechanisms. Graph (a) in Figure 1.6 shows the simulated total answers under current design, which can be regarded as an out-of-sample forecasting. Again, our model captures the contributions relatively accurately. The results of the analysis in graphs (b) – (d) in Figure 1.6 are consistent with Figure 1.5.

1.5.6 Robustness Checks

We conduct several robustness checks.³ First, to ensure that our results are not biased by the sample period chosen (or the life-stage of the community), we estimate the model on several alternative sample periods. Other than the initial probabilities, the results from those estimations are qualitatively consistent with the results reported above. The different initial probabilities may come from the fact that we treat some users as new users although they have been in the community for some time. We also notice some magnitude change of the effects of up-votes and badges. For example, in our estimation results on the sample period $t = 300 - 400$, we notice that both coefficients of up-votes and badges increase even though the signs are the same. This is reasonable because the number of up-votes and badges are lower in the later stage, which amplifies the marginal

³ Results are available upon request.

effects. This also means a careful design is more important in the later stage of the community.

Second, we vary the users included in our sample to ensure that the results are not driven by specific samples of users. Our original sample contains all users that have contributed more than 10 answers on the site. We then estimate the model separately on users who have contributed more than one answer, and those with more than 50 answers, respectively. The results are consistent.

Third, to assess the external validity, we estimate the model on data from two other sub-sites of StackExchange (cooking and bicycles). The results are similar to those in Table 1.2, except that the state-specific intrinsic propensity to contribute c_1 varies across sites. Because c_1 captures the average intrinsic propensity to contribute without stimulus, it may come from site-specific characteristics.

1.6 Closing Remarks

Online communities represent a paradigm of unconventional knowledge collaboration in the sense that they are open, voluntary, and collaborative. They can also tap into talent pools beyond organizational as well as geographical boundaries. While they are effective to spark innovation, online communities face a fundamental challenge: the declining trend of user contributions over time. Consequently, motivating users to contribute becomes a key issue for the sustainability of online communities. The objective of this chapter is to characterize the dynamics of online user behavior, and identify the mechanisms that would induce users to contribute more, so as to mitigate the typical declining trend of participation over time. To achieve this objective, we propose a

structural model, and estimate it with Bayesian method using data from a representative online community.

By doing so, this chapter makes several contributions to the literature. First, our structural approach advances the literature on the dynamics of user contributions in online communities. We incorporate HMM into the theory of public goods, and apply the model to the context of online communities. By distinguishing users' latent motivation states, the model captures the dynamics of voluntary user contributions, which is a crucial issue under-addressed in the literature. While we use StackExchange as a testing field, we hope our framework is applicable to other online communities.

Second, we show that motivating mechanisms work differently depending on which state the user is in. The empirical difficulty of capturing dynamics is that the dynamics structure is unobservable. To handle this challenge, we use latent states of motivations in our structure model, and allow the impact of motivating mechanisms to vary across states. We find that conditional on the motivation state, reciprocity is more effective to induce the contributions from users in the low motivation state. In addition, we find that reputational motivation has a positive effect on user contributions, but may suffer the “moral licensing” effect for users in high motivation state. Also, users in a highly motivated state are more responsive to the demand for knowledge and the size of the community, and are more likely to remain highly motivated. These findings further highlight the importance of distinguishing user latent states in order to explain the dynamics of user behavior. These results have shed light into a key question in the research on collective innovation through online communities, i.e., how to *design* mechanisms that will kindle user contributions effectively.

Third, our structural model allows us to perform interesting design simulations. We calibrate parameters that can be managed by online communities, i.e., parameters related to reputational motivations (*up-votes*, *accepted answers*, and *badges*). We find that these motivational devices are useful to elevate contributions. Although badges are popularly used in many online communities, our results show that using badges can be counterproductive in certain situations. Hence, the effect of badges deserves careful consideration. In contrast, *up-votes* and *accepted answers* are shown to be much more effective, especially for users in the *low* motivation state. Together, these results highlight the effectiveness of *image reward* as a motivating scheme, especially in switching users from low state to high state, which is an issue we really care about in this research. Furthermore, the boosting effect of such reputational motivation is substantially greater in later time periods than at the startup stage of the online communities.

We hope these results provide managerial implications for designing various mechanisms, and evaluating their effectiveness on encouraging user contribution. For example, a platform designer could do experiments on a specific badge and decide how to adjust it. Along this line, for organizations that leverage the collective intelligence of online communities to accumulate knowledge, our results provide hints to design effective mechanisms. First, managers need to be mindful that users have a different propensity to contribute, and it is important to design instruments to motivate contributions. As our design simulations suggest, reputational motivations are effective in this aspect. If the design of the community makes it easier for users to gain *upvotes*, for instance, users are more likely to become highly motivated. Another mechanism could be to facilitate reciprocity by making more visible the number of a user's questions

addressed by the communities, and inviting him to give back. Second, managers should also consider how to foster the community. This includes attracting new users so as to make the “social effect” more prominent and encouraging users to ask questions so as to raise the “demand of knowledge.”

In summary, this chapter uses a structural approach to model the dynamics of user contributions in online communities. The findings provide new evidence about the effect of various motivating mechanisms, and offer managerial insights into the design of online communities. While many open questions remain, we hope these initial results will help stimulate more research in this growing area.

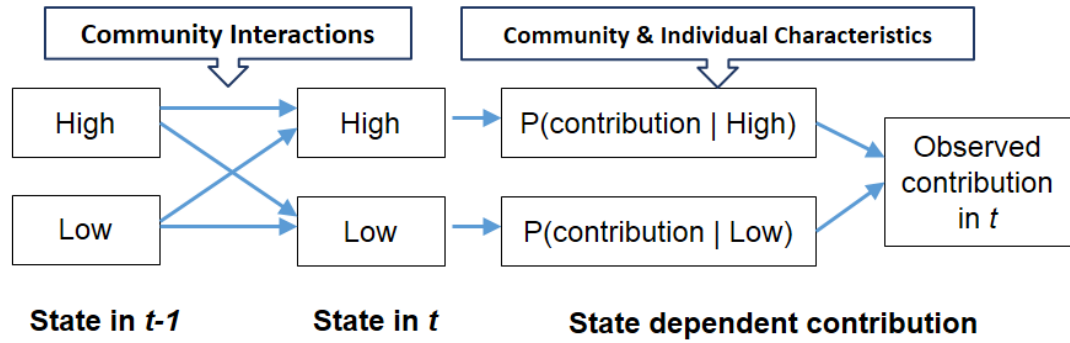


Figure 1.1: Hidden Markov Model of User Contributions

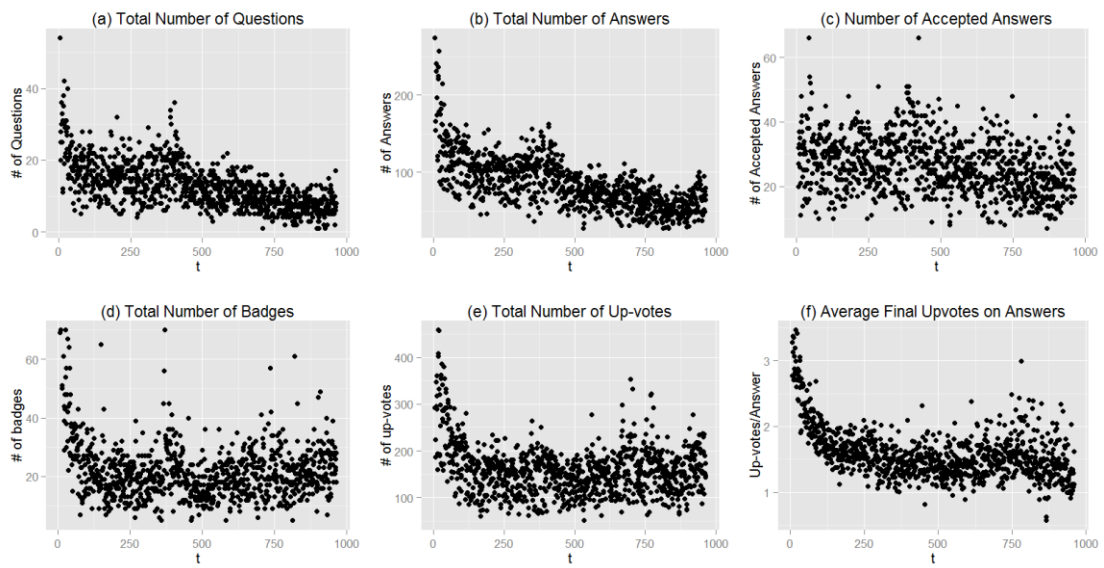


Figure 1.2: Trends of Key Variables

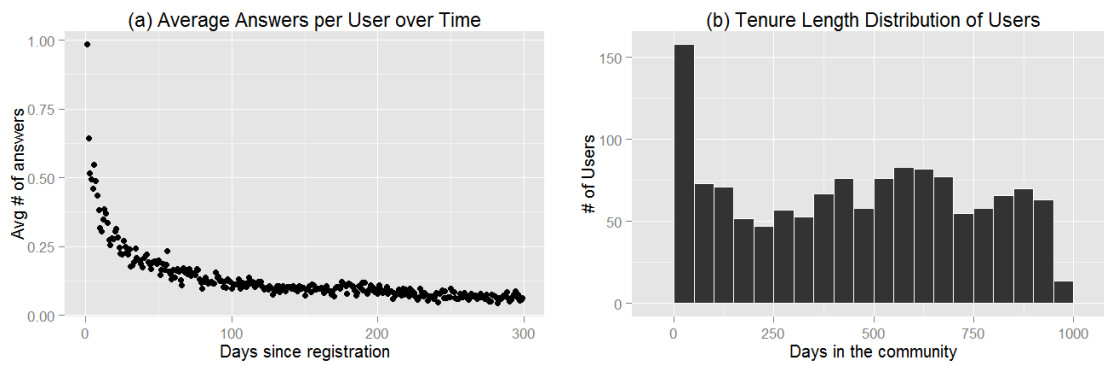


Figure 1.3: Average Answers over Time and Tenure Distribution

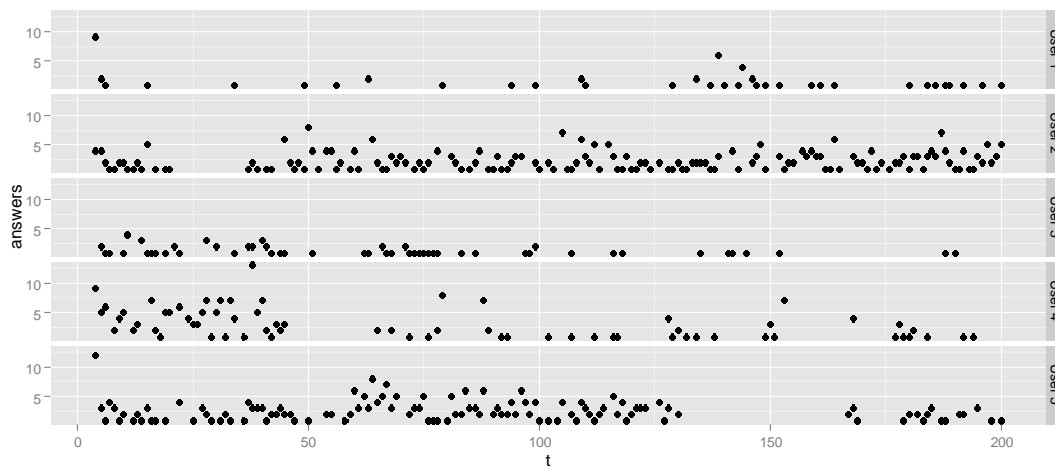
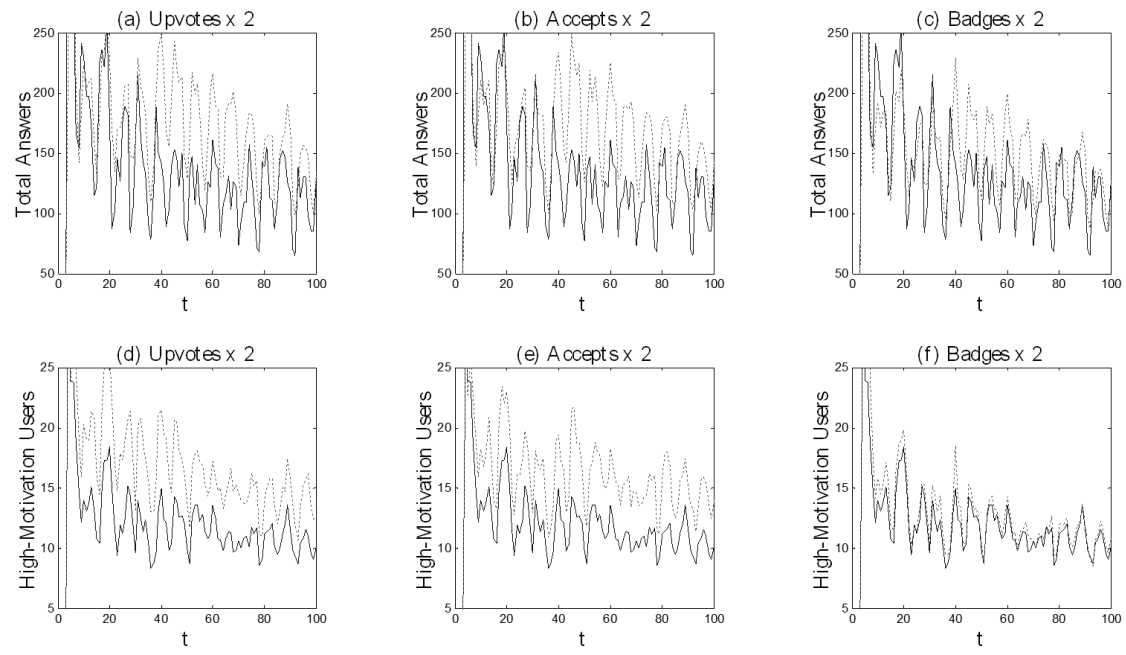


Figure 1.4: Fluctuation of User Contribution over Time



Panel	(a) – (c)	(d) – (f)
————	Observed # of answers	Classified high-motivation users under current design
-----	Simulated # of answers	Classified high-motivation users in simulation

Figure 1.5: Design Simulations: What if reputation is easier to earn?

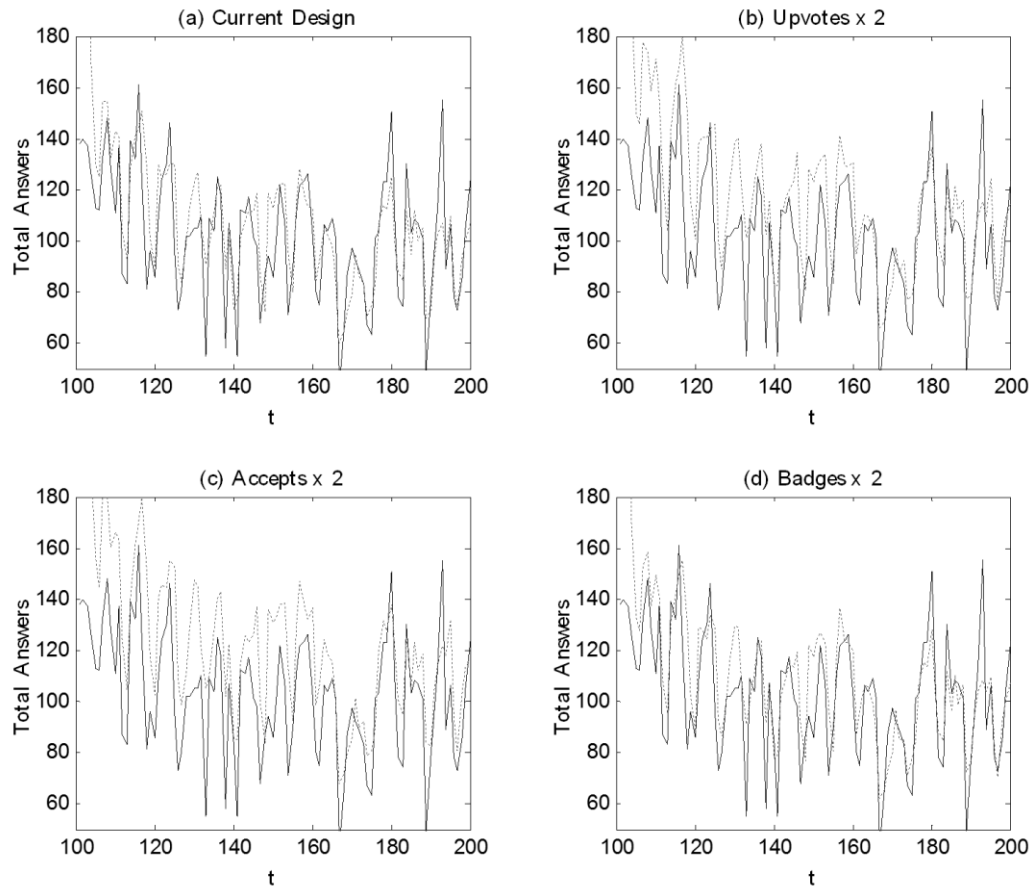


Figure 1.6: Out-of-sample Forecasting and Simulations

Table 1.1: Variables and Descriptive Statistics

Variable	Description	Mean	Sd	Min	Max
Dependent Variable (Y_{it})					
$Answers_{it}$	Number of answers	0.31	1.20	0	31
Community and Individual Characteristics (X_{it})					
$New_questions_t$	Number of new questions	103.50	27.36	0	251
$Group_size_t$	Number of participated users	139.27	30.99	0	222
$Tenure_{it}$	Number of days since the user registered	43.66	27.27	0	99
$Total_answers_{i,t-1}$	Total number of past answers by the user	17.47	39.94	0	679
Community Interactions ($W_{i,t-1}$)					
$Answers_received_{i,t-1}$	Number of comments to past answers of the user	0.12	0.71	0	39
$Upvotes_answer_{i,t-1}$	Number of up-votes to past answers of the user	0.52	2.16	0	44
$Accepted_answers_{i,t-1}$	Number of accepted answers of the user	0.06	0.33	0	8
$Badges_{i,t-1}$	Number of badges earned by the user	0.12	0.48	0	11

Table 1.2: Results of HMM Bayesian Estimation

Variable Name	State 0 (Low Motivation)	State 1 (High Motivation)
X_{it}	β – Posterior Mean (Standard Deviation)	
<i>Const_X</i>	-1.235*** (0.038)	2.221*** (0.270)
<i>New_questions_t</i>	0.0002 (0.0005)	0.007*** (0.002)
<i>Group_size_t</i>	0.004*** (0.000)	0.025*** (0.002)
<i>Tenure_{it}</i>	-0.006*** (0.000)	-0.013*** (0.002)
<i>Total_answers_{s_{i,t-1}}</i>	0.010***(0.000)	0.028***(0.001)
σ^2	1.392*** (0.024)	
$W_{i,t-1}$	ξ – Posterior Mean (Standard Deviation)	
<i>Const_W</i>	-2.555*** (0.026)	-0.248** (0.098)
<i>Answers_received_{i,t-1}</i>	0.040** (0.018)	0.003 (0.028)
<i>Upvotes_answer_{i,t-1}</i>	0.095*** (0.018)	0.048 (0.030)
<i>Accepted_answers_{i,t-1}</i>	0.574*** (0.055)	0.125*** (0.035)
<i>Badges_{i,t-1}</i>	0.169*** (0.033)	-0.076*** (0.030)
σ_u^2	1.000*** (0.010)	
Initial Probability	0.856*** (0.016)	0.144*** (0.016)

* The 90% confidence interval does not include zero; ** The 95% confidence interval does not include zero; *** The 99% confidence interval does not include zero. For brevity, we use “significant” and “insignificant” in the results discussion.

Table 1.3: Mean Posterior Transition Matrices

	(a) Mean Interactions		(b) Up-votes		(c) Accepts		(d) Badges	
	t		t		t		t	
	Low	High	Low	High	Low	High	Low	High
t-1								
Low Motivation	99.3%	0.7%	99.1%	0.9%	97.0%	3.0%	98.9%	1.1%
High Motivation	58.9%	41.1%	57.0%	43.0%	54.0%	46.0%	61.9%	38.1%

Appendix 1.1

The MCMC Estimation of the HMM

We estimate the parameters vector $\{\theta, S\}$ with Gibbs sampling (Albert and Chib 1993). We generate the joint posterior distribution by sampling from each conditional distribution of the following parameter blocks:

$$\begin{aligned}\boldsymbol{\theta} &= (\theta_1, \theta_2', \theta_3', \theta_4', \theta_5')' \\ \theta_1 &= \sigma^{-2} \\ \theta_2 &= (\beta_0', \beta_1')' \\ \theta_3 &= (\xi_0', \xi_1')' \\ \theta_4 &= (s_{i1}, s_{i2}, \dots, s_{iT})', i = 1, \dots, n \\ \theta_5 &= (L_{i1}, L_{i2}, \dots, L_{iT})', i = 1, \dots, n\end{aligned}$$

We denote $\theta_{-i} = (\theta_j)', \forall j \neq i$.

(1) Generate $\theta_1 = \sigma^{-2}$ from $p(\theta_1 | \theta_{-1}, Y, X, W)$.

Prior: $\sigma^{-2} \sim \Gamma(\alpha, \delta)$. Conditional on θ_{-1}, Y, X, W is equivalent to observing

$\{\varepsilon_{it}\}$ for $\varepsilon_{it} = Y_{it} - X_{it}\beta_{s_{it}}$

Posterior: $\sigma^{-2} | \theta_{-1}, Y, X, W \sim \Gamma(\alpha + \frac{1}{2}nT, \delta + \frac{1}{2}SSR)$, where $SSR = \sum_{i=1}^n \sum_{t=1}^T \varepsilon_{it}^2$.

(2) Generate $\theta_2 = (\beta_0', \beta_1')'$ from $p(\theta_2 | \theta_{-2}, Y, X, W)$.

Prior: $\beta_j | \sigma^{-2} \sim N(m_j, \sigma^2 M_j)$ $j = 0, 1$ (independent of each other)

Posterior: Conditional on $\{s_{it}\}$, only those observations for which $s_{it} = j$ are relevant to posterior distribution of β_j .

$$\beta_j | \theta_{-2}, Y, X, W \sim N(m_j^*, \sigma^2 M_j^*)$$

$$M_j^* = (M_j^{-1} + \sigma^{-2} \sum_{i=1}^n \sum_{t=1}^T X_{it} X_{it}' 1_{\{s_{it}=j\}})^{-1}$$

$$m_j^* = M_j^* (M_j^{-1} m_j + \sigma^{-2} \sum_{i=1}^n \sum_{t=1}^T X_{it} Y_{it} 1_{\{s_{it}=j\}})$$

(3) Generate $\theta_3 = (\xi_0', \xi_1')$ from $p(\theta_3 | \theta_{-3}, Y, X, W)$.

Prior: $\xi_j | \sigma_u^{-2} \sim N(mw_j, \sigma_u^2 Mw_j)$ $j = 0, 1$

Posterior: $\xi_j | \theta_{-3}, Y, X, W \sim N(mw_j^*, Mw_j^*)$

where $Mw_j^* = (Mw_j^{-1} + \sum_{i=1}^n \sum_{t=1}^T W_{i,t-1} W_{i,t-1}' 1_{\{s_{i,t-1}=j\}})^{-1}$

and $mw_j^* = Mw_j^* (Mw_j^{-1} mw_j + \sum_{i=1}^n \sum_{t=1}^T W_{i,t-1} L_{it}' 1_{\{s_{i,t-1}=j\}})$

Note that since σ_u is not identifiable, we normalize it to 1 in the estimation.

(4) Generate $\theta_4 = (s_{i1}, s_{i2}, \dots, s_{iT})', i = 1, \dots, n$ from $p(\theta_4 | \theta_{-4}, Y, X, W)$.

We generate the states using the single-move Gibbs-sampling algorithm in Kim and Nelson (1999), which is also the well-known Forward-Backward algorithm.

Denoting Ψ_{it} as information for user i up to time t , and Ψ_{iT} as information from the whole sample, we follow the forward-backward algorithm as below to get

$P(s_{it} | S_{i,-t}, \Psi_{iT})$:

Forward: Calculate $P(s_{it} | \Psi_{it})$

Step 1: Given $P(s_{i,t-1} = k | \Psi_{i,t-1})$, $k = 0, 1$ at the beginning of period t , calculate

$$P(s_{it} = j, s_{i,t-1} = k | \Psi_{i,t-1}) = P(s_{it} = j | s_{i,t-1} = k, \Psi_{i,t-1})P(s_{i,t-1} = k | \Psi_{i,t-1})$$

$$\text{where } P(s_{it} = j | s_{i,t-1} = k, \Psi_{i,t-1}) = \begin{cases} \Phi(W_{i,t-1}\xi_k), & \text{if } j = 1 \\ 1 - \Phi(W_{i,t-1}\xi_k), & \text{if } j = 0 \end{cases}$$

For the first period, we use the initial probability $P(s_{i,1} = 1) = p_0$ and

$$P(s_{i,1} = 0) = 1 - p_0.$$

Step 2: Once X_{it} and Y_{it} are observed in period t , we update the probability term

by calculating $P(s_{it} = j | \Psi_{it}) = \sum_{k=0}^1 P(s_{it} = j, s_{i,t-1} = k | \Psi_{it})$ where

$$\begin{aligned} & P(s_{it} = j, s_{i,t-1} = k | \Psi_{it}) \\ &= P(s_{it} = j, s_{i,t-1} = k | \Psi_{i,t-1}, X_{it}, Y_{it}) \\ &= \frac{f(Y_{it} | s_{it} = j, s_{i,t-1} = k, \Psi_{i,t-1}, X_{it})P(s_{it} = j, s_{i,t-1} = k | \Psi_{i,t-1})}{f(Y_{it} | \Psi_{i,t-1}, X_{it})} \\ &\propto f(Y_{it} | s_{it} = j, X_{it})P(s_{it} = j, s_{i,t-1} = k | \Psi_{i,t-1}) \end{aligned}$$

Backward: In the backward process, we generate s_{it} conditioning on Ψ_{it} and $s_{i,t+1}$,

($t = T-1, T-2, \dots, 1$) using $g(s_{it} | \Psi_{it}, s_{i,t+1}) \propto g(s_{i,t+1} | s_{it}, \Psi_{it})g(s_{it} | \Psi_{it})$. We then can

calculate

$$Pr(s_{it} = k | s_{i,t+1}, \Psi_{it}) = \frac{g(s_{i,t+1} | s_{it} = k, \Psi_{it})g(s_{it} = k | \Psi_{it})}{\sum_{k=0}^1 g(s_{i,t+1} | s_{it} = k, \Psi_{it})g(s_{it} = k | \Psi_{it})}$$

Then we can use a random number drawn from a uniform distribution to generate

s_{it} .

(5) Generate $\theta_5 = (L_{i1}, L_{i2}, \dots, L_{iT})'$, $i = 1, \dots, n$ from $p(\theta_5 | \theta_{-5}, Y, X, W)$.

L_{it} determines s_{it} according to the following formula:

$$s_{it} = \begin{cases} 1 & \text{if } L_{it} > 0 \\ 0 & \text{if } L_{it} \leq 0 \end{cases}$$

Conditional on θ_{-5} , we can generate L_{it} as below:

If $s_{it} = 0$, draw L_{it} from a truncated normal distribution $TN_{(-\infty, 0]}(W_{i,t-1}\xi_{s_{i,t-1}}, 1)$.

If $s_{it} = 1$, draw L_{it} from $TN_{[0, \infty)}(W_{i,t-1}\xi_{s_{i,t-1}}, 1)$.

Repeating this for $t = 1, \dots, T$ and $i = 1, \dots, n$ gives a draw from $p(\theta_5 | \theta_{-5}, Y, X, W)$.

Appendix 1.2

Testing the Estimation on Simulated Data

Because our model has a non-linear feature by incorporating the *Tobit* and *probit* models, we could not use standard statistical software to estimate it. We have to write our own estimation algorithm instead. Hence we did, but we need to ensure that it is correct before applying the algorithm to the actual data. We run the algorithm on simulated data based on known parameters, and test whether it could recover the “true” parameters.

We first generate the “true” parameters θ , the community and individual characteristics variables $X = \{X_{it}\}_{t=1,\dots,T; i=1,\dots,Nt}$, and the community interaction variables $W = \{W_{it}\}_{t=1,\dots,T; i=1,\dots,Nt}$. Since we assume that a user has an initial probability p_0 to be motivated, at $t = 1$ we draw the initial state s_{i1} of user i from a Bernoulli distribution using the initial probability p_0 for each user i that enters the community. Conditional on s_{i1} , we then draw the contribution $Y_{i1} = \max(0, Y_{i1}^*)$, where $Y_{i1}^* = X_{i1}\beta_{s_{i1}} + \varepsilon_{i1}$ and ε_{i1} is generated from a normal distribution with mean 0 and variance σ^2 . For any $t > 1$, we first draw $L_{it} = W_{i,t-1}\xi_{s_{i,t-1}} + u_{it}$, where u_{it} is drawn from $N(0, \sigma_u^2)$, then we generate the state s_{it} according to L_{it} . Repeating the same process, we generate $Y = \{Y_{it}\}_{t=1,\dots,T; i=1,\dots,Nt}$ for all $t=1,2,\dots,T$.

With the simulation data $\{X, W, Y\}$, we estimate the model with our procedure and present the results in Table A1.1. Our simulation data contains 1,537 individuals and 10 periods of time. The community and individual characteristics vector X contains four variables, and the community interaction vector W contains three variables. In Table 1,

the “True Parameters” column on the left displays the original parameters $\theta = \{\beta, \zeta, \sigma, \sigma_u\}$ that we employ to generate the simulation data. The “Estimation” column on the right displays the estimated parameters.⁴ Our estimation recovers the “true” parameters accurately. This confirms the reliability of the estimation algorithm, and gives us confidence in its empirical application to the actual data.

Table A1.1: Estimation Results from Simulation Data

Variables	True Parameters		Estimation	
	State 0	State 1	State 0	State 1
β			Mean (Standard Deviation)	
x1	3	5	3.00 (0.01)	4.99 (0.01)
x2	4	6	4.00 (0.01)	5.99 (0.01)
x3	5	7	5.01 (0.01)	7.02 (0.01)
x4	6	8	6.00 (0.01)	7.99 (0.01)
σ^2		2	1.98 (0.03)	
ξ				
w1	4	7	0.42 (0.04)	0.74 (0.04)
w2	5	8	0.46 (0.04)	0.82 (0.04)
w3	6	9	0.62 (0.04)	0.88 (0.04)
σ_u^2		100	1.00	
$(p_0, 1 - p_0)$	0.91	0.09	0.89 (0.01)	0.11 (0.01)
$T = 10$	$N = 1,537$	Draws = 5,000		

⁴ Note that σ_u is not identifiable in equation (5). Therefore, the estimates of ζ are actually ζ/σ_u from the true parameters. We normalize $\sigma_u^2=1$ in all estimations.

References

- Albert, J. H., & Chib, S. (1993). Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association*, **88**(422), 669-679.
- Andreoni, J. (1988). Privately Provided Public Goods in A Large Economy: The Limits of Altruism. *Journal of Public Economics*, **35**(1), 57-73.
- Andreoni, J. (1990). Impure Altruism and Donations to Public Goods: A Theory of Warm-Glow Giving. *The Economic Journal*, **100**(401), 464-477.
- Antorini, Y. M., Muniz, A. M., & Askildsen, T. (2012). Collaborating with Customer Communities: Lessons from the Lego Group. *MIT Sloan Management Review*, **53**(3), 73-79.
- Ariely, D., Bracha, A., and Meier, S. (2009). Doing Good or Doing Well? Image Motivation and Monetary Incentives in Behaving Prosocially. *American Economic Review* **99**(1) 544–555.
- Bateman, P. J., Gray, P. H., and Butler, B. S. (2011). The Impact of Community Commitment on Participation in Online Communities. *Information Systems Research* **22**(4) 841 –854.
- Bayus, B. L. (2013). Crowdsourcing New Product Ideas over Time: An Analysis of the Dell Ideastorm Community. *Management Science*, **59**(1), 226-244.
- Bénabou, R., and Tirole, J. (2006). Incentives and Prosocial Behavior. *American Economic Review*, **96**(5), 1652–1678.
- Boudreau, K. J., & Lakhani, K. R. (2009). How to Manage Outside Innovation. *MIT Sloan Management Review*, **50**(4) 69–73.
- Butler, B. S. (2001). Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures. *Information Systems Research* **12**(4) 346 –362.

- Burke, B. (2011). *Maverick Research: Motivation, Momentum and Meaning: How Gamification Can Inspire Engagement*. Gartner Inc. (October)
- Chen, Y., Harper, F Maxwell, Konstan, J., & Li, S. X. (2010). Social Comparisons and Contributions to Online Communities: A Field Experiment on MoviLens. *American Economic Review*, **100**(4), 1358-1398.
- Chiu, C., Hsu, M., & Wang, E. T. (2006). Understanding Knowledge Sharing in Virtual Communities: An Integration of Social Capital and Social Cognitive Theories. *Decision Support Systems*, **42**(3), 1872-1888.
- Faraj, S., Jarvenpaa, S. L., and Majchrzak, A. (2011). Knowledge Collaboration in Online Communities. *Organization Science* **22**(5) 1224–1239.
- Faraj, S., & Johnson, S. L. (2011). Network Exchange Patterns in Online Communities. *Organization Science*, **22**(6), 1464-1480.
- Fischbacher, U., & Gächter, S. (2010). Social Preferences, Beliefs, and the Dynamics of Free Riding in Public Goods Experiments. *American Economic Review*, **100**(1), 541-556.
- Gneezy, A., Imas, A., Brown, A., Nelson, L. D., & Norton, M. I. (2012). Paying To Be Nice: Consistency and Costly Prosocial Behavior. *Management Science*, **58**(1), 179-187.
- Gu, B., Konana, P., Rajagopalan, B., & Chen, H. W. M. (2007). Competition among Virtual Communities and User Valuation: The Case of Investing-Related Communities. *Information Systems Research*, **18**(1), 68-85.
- Halfaker, A., Geiger, R. S., Morgan, J. T., & Riedl, J. (2013). The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline. *American Behavioral Scientist*, **57**(5), 664-688.
- Hamilton, J. D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, **57**(2), 357-384.
- Hardin, G. (1968). The Tragedy of the Commons. *Science*, **162**(3859), 1243-1248.
- Huang, Y., Vir Singh, P., and Srinivasan, K. (2014). Crowdsourcing New Product Ideas under Consumer Learning. *Management Science*, **60**(9), 2138–2159.
- Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science*, **20**(1), 50–67.
- Kane, G., and Ransbotham, S. (2013). Which Came First? Contribution Dynamics in Online Production Communities. *ICIS 2013 Proceedings*.

- Kankanhalli, A., Tan, B. C., & Wei, K. (2005). Contributing Knowledge to Electronic Knowledge Repositories: An Empirical Investigation. *MIS Quarterly*, **29**(1), 113-143.
- Kim, C.-J., and Nelson, C. R. (1999). *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*, The MIT Press.
- Kraut, R. E., Resnick, P., Kiesler, S., Burke, M., Chen, Y., Kittur, N., Konstan, J., Ren, Y., and Riedl, J. (2012). *Building Successful Online Communities: Evidence-Based Social Design*, MIT Press.
- Lakhani, K. R., & von Hippel, E. (2003). How Open Source Software Works: “free” User-to-User Assistance. *Research Policy*, **32**(6), 923-943.
- Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, **50**(2), 197-234.
- Levine, S. S., and Prietula, M. J. (2013). Open Collaboration for Innovation: Principles and Performance. *Organization Science*, **25**(5), 1414-1433.
- Ma, M., and Agarwal, R. (2007). Through a Glass Darkly: Information Technology Design, Identity Verification, and Knowledge Contribution in Online Communities. *Information Systems Research*, **18**(1), 42 –67.
- Malone, T., Laubacher, R., and Dellarocas, C. (2010). The Collective Intelligence Genome. *MIT Sloan Management Review*, **51**(3) 21–31.
- Moon, S., Kamakura, W. A., & Ledolter, J. (2007). Estimating Promotion Response When Competitive Promotions Are Unobservable. *Journal of Marketing Research*, **44**(3), 503-515.
- Netzer, O., Lattin, J. M., & Srinivasan, V. (2008). A Hidden Markov Model of Customer Relationship Dynamics. *Marketing Science*, **27**(2), 185-204.
- Piskorski, M. J., Eisenmann, T. R., Bussgang, J. J., and Chen, D. (2010). Foursquare. Harvard Business School Case 711-418.
- Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, **77**(2), 257-286.
- Ransbotham, S., & Kane, G. C. (2011). Membership Turnover and Collaboration Success in Online Communities: Explaining Rises and Falls from Grace in Wikipedia. *MIS Quarterly*, **35**(3), 613-627.
- Ray, S., Kim, S. S., & Morris, J. G. (2014). The Central Role of Engagement in Online Communities. *Information Systems Research*, **25**(3), 528-546.

- Ren, Y.Q., Harper, F.M., Drenner, S., Terveen, L., Kiesler, S., Riedl, J., and Kraut, R.E. (2012). Building Member Attachment in Online Communities: Applying Theories of Group Identity and Interpersonal Bonds, *MIS Quarterly*, **36**(3), Sep, pp. 841-864.
- Ren, Y., & Kraut, R. E. (2011). A Simulation for Designing Online Community: Member Motivation, Contribution, and Discussion Moderation. *Working paper*.
- Raymond, E. S. (1999). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media.
- Reiss, P. C., and Wolak, F. A. (2007). Chapter 64 Structural Econometric Modeling: Rationales and Examples from Industrial Organization. *Handbook of Econometrics*, J. Heckman and E. Leamer (ed.), (Vol. 6, Part A) 4277–4415.
- Roels, G., & Su, X. (2013). Optimal Design of Social Comparison Effects: Setting Reference Groups And Reference Points. *Management Science*, **60**(3), 606-627.
- Sauermann, H., and Cohen, W. M. (2010). What Makes Them Tick? Employee Motives and Firm Innovation. *Management Science*, **56**(12) 2134–2153.
- Sauermann, H. & Franzoni, C. (2013). Participation Dynamics in Crowd-Based Knowledge Production: The Scope and Sustainability of Interest-Based Motivation. *Working paper*.
- Simonite, T. (2013). The Decline of Wikipedia. *MIT Technology Review*, October 22, 2013.
- Singh, P. V., Tan, Y., & Youn, N. (2010). A Hidden Markov Model of Developer Learning Dynamics in Open Source Software Projects. *Information Systems Research*, **22**(4), 790-807.
- Toubia, O., & Stephen, A. T. (2013). Intrinsic vs. Image-Related Utility in Social Media: Why Do People Contribute Content to Twitter? *Marketing Science*, **32**(3), 368-392.
- von Hippel, E. (2005). *Democratizing Innovation*. MIT press.
- von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. *Management Science*, **52**(7), 975-983.
- Wooldridge, J. M. (2010). *Econometric Analysis of Cross Section and Panel Data*. MIT press.
- Zhang, X., & Zhu, F. (2011). Group Size and Incentives to Contribute: A Natural Experiment at Chinese Wikipedia. *American Economic Review*, **101**(4), 1601-1615.

Chapter 1 is currently under review for publication of the material. Chen, Wei, Xiahua (Anny) Wei, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.

Chapter 2

Competition among Proprietary and Open-Source Software Firms: The Role of Licensing on Strategic Contribution

Open-source software (OSS) firms are increasingly using service-based business models to compete with established proprietary software firms. Because other members of the open-source community can strategically contribute to OSS and compete in the services market, the nature of competition between OSS and proprietary software firms is becoming more complex. Further, their incentives are strongly influenced by the licensing schemes that govern OSS. We study a 3-way game with strategic contribution from the community and focus on how open-source licensing affects competition among an open-source originator, open-source contributor, and a proprietor competing in the same software market. In this regard, we examine: (i) how quality investments and service prices are endogenously determined in equilibrium, (ii) how license restrictiveness im-

pacts equilibrium investments and the quality of services brought to market, and (iii) how license restrictiveness affects consumer surplus and social welfare. Although some in the open-source community often advocate restrictive licenses such as GPL, because it is not always in the best interest of the originator for the contributor to invest greater development effort, such licensing can actually be detrimental to both consumer surplus and social welfare when it exacerbates this incentive conflict. We find such an outcome to be the case in markets characterized by software providers with similar development capabilities. In contrast, when their capabilities are more dispersed, a more restrictive license can instead encourage greater effort from the OSS contributor, lead to higher OSS quality, and provide a larger societal benefit. These results offer managerial implications to software firms that are formulating strategies to participate in OSS and provide insights to policy makers in the design of appropriate policies that govern intellectual property rights associated with OSS.

2.1 Introduction

In most software markets, open-source software (OSS) has increasingly competed with proprietary software over time (e.g., Linux vs. Windows, Firefox vs. Internet Explorer, Open ERP vs. SAP, Hadoop vs IBM General Parallel File System (GPFS), and Red Hat JBoss vs. Oracle WebLogic). More recently, Google's Android, the most popular open-source mobile operating system, has changed the competitive landscape in the mobile devices market (Butler 2011). Other Internet companies, such as Facebook,

LinkedIn and Twitter, are also strategically contributing heavily to open-source projects. As a result, it is important to examine the impact open-source products have on competition in software markets and add to our broader understanding of the open-source movement that is being developed in the academic literature (von Krogh and von Hippel 2006).

Because its source code is open and available, open-source software is typically offered for free. Previous studies have examined the impact of being free (or nearly free) on competition (Casadesus-Masanell and Ghemawat 2006, Lee and Mendelson 2008). When one reflects on the contributors to OSS, one might visualize a group of individual developers working in their spare time and contributing for fun, being intrinsically motivated; in fact, the literature shows that the motivations underlying developer contributions are indeed quite broad (Roberts et al. 2006). However, as more and more *companies* adopt and rely upon open source as a fundamental component of their business strategy, these companies, in contrast to typical individuals in the past, have very clear economic incentives to invest in the development and maintenance of OSS products that are critical to their businesses. These extrinsic motivations have important implications on market competition.

Historically open-source software originated from individual programmers sharing development efforts toward a common interest. However, in current times, more and more contributors are now being paid by their employers to work on particular open-source software projects and be active members of the respective communities. For example, more than 80% of the contributions to the Linux kernel now stem from paid

developers (Corbet et al. 2013). Moreover, an interesting observation from the sponsors' list of many open-source projects is that rival firms often work side-by-side toward the development of the same project. For example, Google and Samsung are both on the top 10 list of sponsors of the Linux kernel (Corbet et al. 2013); similarly, RedHat and Rackspace are both important contributors to the open cloud computing project, OpenStack (Gonzalez-Barahona 2013). Because of these observed behaviors, the academic community has worked to better understand the incentives of these competing and collaborating firms. Recently, scholars began modeling the strategic decisions underlying open-source development driven by complementary products (Haruvy et al. 2008) or services (August et al. 2012, 2013). This stream of literature generates insights relevant to the decision to "go open" and its implications on competition.

Open-source software often enters markets that are dominated by powerful proprietary software firms. How this aspect of the market structure relates to the aforementioned economic incentives of firms to contribute to open source has yet to be studied in the literature but has important and wide-ranging implications on the quality and competitiveness of the products that emerge in such software markets. If a firm chooses to offer its product as open source, it has to battle the incumbent from day one. The good news is that the open-source firm can leverage development efforts from the community, which may include third-party firms that are competing and collaborating with it. This structure makes competition different from traditional settings where the competitors are clearly marked; in this case, open-source contributors can be both helpful and hurtful to each other's profitability. Instead of a typical duopoly, here we have three strategic players in

the game: the proprietary firm, the open-source originating firm, and a strategic, third-party contributor to the open-source product. Our first research question then is: how does consideration of a strategic contributor affect competition between open-source and proprietary software?

In contrast to the popular belief that open source means free of restrictions and liberal in copyright, intellectual property (IP) concerns do not disappear in the open-source world. In fact, contributions from the open-source community may be substantially affected by IP rights (Wen et al. 2013). In the open-source ecosystem, OSS projects are actually distributed under various license schemes as a means to govern intellectual property. Distinct licenses each have different restrictions on the use and modification of the software as well as its derivatives. The GNU General Public License (GPL) is a widely-employed open-source license that is considered to be quite restrictive with regard to what a contributor to the project can do with the software. In particular, this license has “copyleft” requirements which forces any contributor who modifies and redistributes the GPL licensed software to make his or her derivative work also licensed according to GPL, making it available back to the community (Laurent 2004). From a social perspective, GPL licensing advocates claim this form of licensing guarantees the rights granted to the software cannot be taken away. However, contributing organizations with commercial interests often find such restrictions to be detrimental (Stewart et al. 2006). On the other end of the spectrum lies the Berkeley Software Distribution (BSD) license which imposes minimal restrictions on anyone that uses or develops a software offering on top of a BSD licensed software product, granting *permission* to

freely use and modify the software (Laurent 2004).

Extant research has begun to characterize the determinants of open-source license choice (see, e.g., Lerner and Tirole 2005b, Singh and Phelps 2013), such as the preferences of developers and the community as well as social influence. It is equally important to consider the impact of licensing on the economic incentives of firms to participate in OSS development, particularly when the contributors shift from individuals to commercial firms. Because the strategic development efforts made by these firms largely determine the qualities of the products that emerge, licensing can significantly impact software markets. Moreover, because OSS competes with proprietary counterparts, licensing can also influence the quality of proprietary offerings brought to market as well. Our second research question then is: how does the degree of OSS license restrictiveness affect the incentives of the open-source originator, the open-source contributor, and the proprietary firm?

When one reflects on the open-source movement and the ideology behind it, an open question is what types of licensing genuinely lead to better outcomes in software markets. For example, do certain licenses lead to higher quality products/services in the market? To that end, our third research question then is: under what market conditions do permissive and restrictive licenses each respectively help improve consumer surplus associated with the software offerings?

To answer this set of research questions, we develop a game-theoretic model in which we capture the economic incentives of a competing and collaborating OSS originator and contributor, as well as a proprietary software vendor. We first characterize the

equilibrium effort investments and prices for all three entities. Second, we study how license restrictiveness affects the strategic interactions among the three players and characterize its effect on product qualities. Third, we explore the implications of different licensing schemes on consumer surplus and social welfare.

We find that players' strategic behaviors change as the competitiveness of the market changes, and the implications of license restrictiveness vary accordingly. Using this model, we explore two regions relating to the development cost efficiency of these three strategic players: (i) high cost dispersion and (ii) low cost dispersion. In the former, the players differ in their capabilities to a greater degree, whereas the latter region has the potential to become more competitive. First, we examine the case when the cost efficiency differential between the open-source originator and the proprietary firm is high, i.e., competition between the two tends to be relatively mild. As OSS licensing becomes more restrictive, even though the originator can leverage more of the contributor's effort, the contributor in fact also prefers to exert greater effort. As a result, the originator's quality increases even though the originator itself may exert more or less effort. Because of these competitive effects, we find that restrictive licenses lead to higher quality OSS offerings but a lower quality proprietary offering. Both consumer surplus and social welfare generally increase because the increased quality OSS offerings cause the proprietary firm to price more competitively.

Second, when the cost efficiency differential is low and competition can potentially be more intense, both the open-source originator and contributor exert less effort to limit the quality of their respective offerings to avoid excessive competition with the

proprietary firm. Their strategic behavior is quite different than in the previous case. Because restrictive licenses cause all three providers' qualities to decrease, consumer surplus and social welfare also suffer. This suggests that GPL-style licenses may be inappropriate for software markets with low cost dispersion; more permissive BSD-style licenses can lead to better outcomes.

In this paper, our unique contribution is that we examine how competition unfolds between proprietary and open-source software when the competing open-source offerings stem from strategic, profit-motivated entities who contribute to OSS and leverage their expertise to provide value-added services. In this sense, a firm that originates OSS must compete against both strategic contributors to the OSS who compete in the same services market as well as proprietary counterparts. Because providers in enterprise software markets often fit this characterization, we aim to provide insights into such markets by studying the economic incentives for firms to invest in software development and compete on both quality and pricing of their services. OSS licensing can strongly influence the providers' incentives, hence our analysis focuses on the impact of license restrictiveness and particularly highlights how it affects the quality of software brought to market.

2.2 Literature Review

Our study is related to the body of literature that examines how firms compete with OSS (Lerner and Tirole 2005a). As a prerequisite to analyzing competition, re-

searchers have thoroughly studied the motivations of open-source developers which led to a stream of literature that connects contributions that are being made to various extrinsic and intrinsic motivations (see, e.g., Hars and Ou 2002, von Krogh and von Hippel 2006, Roberts et al. 2006, Iansiti and Richards 2006, August et al. 2013). Several papers consider these diverse motivations to examine how a commercial firm competes with an open-source product. This literature can be classified into two groups dependent on whether the open-source investments are driven by commercial or non-commercial interests.

First, we discuss the non-commercial case where the OSS product is generally available at zero price. Casadesus-Masanell and Ghemawat (2006) formulate a model motivated by competition between Windows and Linux, where Linux benefits from superior demand-side learning and lower (zero) prices that boost market share. Even under these conditions, they show that Windows can persist in the market. Lee and Mendelson (2008) examine a market characterized by network effects and compatibility issues where open-source developers maximize a weighted sum of consumer surplus and intrinsic benefit. They find that in some cases, a commercial firm has incentives to make its product incompatible with OSS and of higher quality. Casadesus-Masanell and Llanes (2011) also study compatibility issues when the commercial firm can open part of its codebase and benefit from additional effort stemming from the OSS community. Athey and Ellison (2014) model the evolution of OSS when developers contribute due to reciprocal altruism and study how a proprietary firm dynamically prices its product to compete. Prior work has also examined the impact of product heterogeneity (Bitzer

2004), network effects (Cheng et al. 2011), and lock-in strategies (Zhu and Zhou 2012) on competition between OSS and proprietary software.

Our work benefits from these studies but also differs from them by focusing on profit-maximizing, commercial open-source firms, and thus is closer to the second group of papers that examine commercial interests. In particular, we model open-source firms who invest in OSS and compete in the market for value-added services. In this portion of the literature, several papers study complementary products and services. Haruvy et al. (2008) examine a monopolist's decision on whether to open the source code when it can profit from a complementary product. Mustonen (2005) and Asundi et al. (2012) study a proprietary firm's incentives to support OSS when the firm can benefit from either network effects or through users' preferences on customization. In the context of platform competition between OSS and proprietary software with two-sided network effects, Economides and Katsamakos (2006) show that a proprietary system dominates a system based on an OSS platform when the demand potential for the OSS system is relatively limited. Kim et al. (2006) examine competition between proprietary software and OSS, where OSS is offered under either a dual licensing scheme or a support model. When the OSS firm uses a support model, they find that a proprietary firm squeezes the OSS firm out of the market by pricing at the marginal cost of support.

Sen (2007) examines usability differences among proprietary software, OSS with support services, and OSS without support services (free). Sen (2007) studies competition among three vertically differentiated offerings, where the qualities of the proprietary and base OSS product without services is fixed, and the focus lies on how the

OSS provider of support services invests in usability. We complement this paper in several ways. First, in our model, we have three providers investing strategically such that the software qualities are endogenously determined in equilibrium. Second, similar to August et al. (2013), we include a strategic contributor who helps to improve the OSS product (which also can benefit the OSS originator) but acts as a competitor in the market for services. Different from August et al. (2013), our work includes a proprietary firm in order to study 3-way competition which can significantly impact the OSS originator's investment incentives. Further, while they study the decision whether to pursue a proprietary or OSS path, we focus on how *licensing* (especially the restrictiveness of licensing) affects competition due to its impact on strategic contributions from the third party.

In this respect, our work is related to the literature that explores OSS licensing. Lerner and Tirole (2005b) explore the choice of open-source licenses as it relates to an OSS originator's ability to induce contributions from the community and generate returns from commercial clients that may prefer more permissive licenses. Singh and Phelps (2013) examine the relationship between OSS license choice and social influence. It is noteworthy that oftentimes the license type can already be pre-determined to some extent by industry. For example, Polanski (2007) shows that restrictive licenses such as GPL may be a rational choice for the first innovator in a sequential innovation setting. Firms often have no choice but to adhere to certain license restrictions in order to make their products compatible with other software they intend to leverage. August et al. (2012) study the policy implications of OSS licensing as it swings a software originator's

decision to go either proprietary or open source.

There are several empirical papers that explore the effects of different licenses, including how license restrictiveness impacts the incentives of community participants. Fershtman and Gandal (2007) report that output per contributor is much higher in projects with *less* restrictive licenses. In contrast, Colazo and Fang (2009) find that a *more* restrictive license will induce greater contribution and faster development of the product. Belenzon and Schankerman (2015) find that developers strongly sort by license type, i.e., developers who belong to a project with a restrictive license almost exclusively contribute to other projects with restrictive licenses. Similarly, developers who belong to a project with a permissive license primarily contribute to other projects with permissive licenses.

These empirical findings, albeit inconclusive, further motivate our research questions defined earlier. An open question is how does license restrictiveness alter the economic incentives of profit-driven OSS firms when they are actively competing against proprietary firms in the same market. To explore this issue, our paper focuses on the effect of license restrictiveness in a market where a proprietary firm competes against both an open-source originator and an open-source contributor who make investments to jointly improve the OSS product and separately develop expertise that yield value in the service market. Our model enables us to better understand how licensing affects the quality of software being produced in these competitive environments. Along the way, we also generate insights into the impact of OSS on quality, consumer surplus and social welfare by examining the role of OSS licensing.

2.3 Model

Three players, a proprietary software vendor, i.e., a *proprietor*, an open-source software *originator*, and an open-source software *contributor* compete in an enterprise software market. We denote the proprietor with subscript p , the originator with subscript o , and the contributor with subscript c . In this market, in order to derive value from the enterprise software, a consumer must install it, integrate it with existing business systems and processes, and acquire support for the software going forward; i.e., the consumer needs to obtain *services* from a service provider of the software who has expertise in order to access this value. The value derived by the consumer also clearly depends on the service provider chosen. For example, a service provider with greater expertise and secondary utility tools can offer higher quality services.

In our model, the proprietor is the main service provider for its proprietary, closed-source enterprise software. However, for the open-source enterprise software alternative, both of the two open-source providers (the originator and contributor) are capable of providing these services. Further, we narrow our focus to a profit-motivated originator and a profit-motivated contributor who strictly generate revenues by offering value-added services.¹ Even though the open-source software itself is a common foundation for both the originator and contributor's offerings, the *total quality* of the complete solution offered by each firm can differ because they vary in their expertise and corresponding service

¹There are typically other contributors to open-source software who have non-strategic motivations including hobbyism and altruism. Incorporating such non-strategic contributions will not change the essence of our results, hence we employ a simplified model that highlights how strategic motivations affect the quality of software solutions brought to the market.

qualities. We denote the total quality for the proprietor, originator, and contributor with Q_p , Q_o , and Q_c , respectively.

There is a continuum of consumers who have heterogeneous preferences on the total quality of the complete software solutions which we model as a uniformly distributed type characteristic $\theta \in \Theta = [0, 1]$. Thus a consumer with type θ derives value θQ_p , θQ_o , or θQ_c , depending on whether she contracts with the proprietor, originator, or contributor, respectively. Their total qualities are determined by their effort investments which we next describe. For the proprietary closed-source case, the proprietor incurs all development costs alone. In particular, the proprietor chooses a development effort $e_p \in \mathbb{R}_+$ and correspondingly incurs a quadratic, convex cost of effort $\beta_p e_p^2/2$, where $\beta_p > 0$ is a measure of cost efficiency. Based on this effort investment, the proprietor's total quality is given by $Q_p = s_p e_p$, where $s_p > 0$ is a measure of how effectively it translates effort into total quality.

Having observed the proprietor's effort investment and quality, the originator of open-source software can invest in the design and development of a competitive, open-source solution. Analogous to the proprietor, the originator chooses a development effort $e_o \in \mathbb{R}_+$ and similarly incurs a cost of effort $\beta_o e_o^2/2$, where $\beta_o > 0$. Finally, having observed both the proprietor and originator's investments, the contributor may put forth effort to improve the open-source software and benefit its own service offering. The contributor chooses a development effort $e_c \in \mathbb{R}_+$ and incurs a cost of effort $\beta_c e_c^2/2$, where $\beta_c > 0$.

Effort in both proprietary and open-source software development generally in-

volves labor and resources, and, in the software industry, firms exhibit significant variation in size and worker abilities. Further, for any given project, there can be significant differences in resource availability and their shadow prices. Therefore, β_p , β_o , and β_c are generally different due to firms heterogeneity in this dimension (see, e.g., Oi 1983).

Unlike the proprietor, the originator and contributor's total qualities are affected by each other's investments which leads to cross effects. In particular, the contributor benefits from the effort invested by the originator toward development of the open-source product. Similarly, the contributor's subsequent developments are open source and available to the public, hence the originator also benefits from the contributor's efforts. Therefore, in this enterprise software and related services marketplace, the originator and the contributor are complementary to each other in improving the total quality of the software and services; however, they also compete against each other, as well as the proprietor, in providing these services to the marketplace. Capturing these effort interactions, we model the originator's total quality as

$$Q_o = s_o e_o + s_{oc} e_c, \quad (2.1)$$

where $s_{oc} > 0$ indicates the cross effect of the contributor's effort on the originator's total quality, while $s_o > 0$ is the direct effect of the originator's own effort on its quality.

Similarly, the contributor's total quality also depends on the originator's effort through

an analogous parameter, s_{co} , and is given by

$$Q_c = s_c e_c + s_{co} e_o, \quad (2.2)$$

where $s_{co} > 0$ represents the cross effect of the originator's effort on the contributor's total quality and $s_c > 0$ represents the direct effect. The magnitudes of the coefficients s_p , s_o , s_c , s_{oc} , and s_{co} depend critically on characteristics of the particular enterprise software market being studied as well as both the nature of and relationship between the service-providing firms. In our competitive framework, these coefficients carry substantial economic and strategic significance. In particular, they strongly influence a firm's incentive to invest because having a competing service provider also derive a large benefit can actually amplify or reduce the firm's willingness to make these investments. Modeling such strategic interactions is essential for developing an understanding how open-source originators and contributors compete with proprietary offerings.

After the proprietor, originator, and contributor effort investments are made sequentially in that order, all three service providers simultaneously set the prices of their offerings: p_p , p_o , and p_c , respectively. These prices represent the total price a consumer must pay for the software and services offered by each provider. For simplicity, we assume that the unit cost of providing these services is the same for each provider and denoted $c > 0$.

After these prices are set, consumers' usage decisions are made in the last stage. Denoting the net utility to consumer θ with $V(\theta)$, she obtains the following payoffs

depending on her provider choice:

$$V(\theta) = \begin{cases} \theta Q_p - p_p & \text{if contracted with the proprietor;} \\ \theta Q_o - p_o & \text{if contracted with the originator;} \\ \theta Q_c - p_c & \text{if contracted with the contributor;} \\ 0 & \text{if not contracted.} \end{cases} \quad (2.3)$$

In summary, the timeline for our model is described in the following and depicted in Figure 2.1.

- 1- The proprietor invests effort e_p in developing a proprietary, closed-source software product.
- 2- Having observed the proprietor's effort, e_p , and the quality of its offering, the open-source software originator decides whether and how much to invest in developing its open-source software, e_o .
- 3- Having observed both the proprietor and originator's efforts, the open-source contributor decides whether to also provide services for the open-source software. If so, the contributor also invests in the open-source development effort, e_c .
- 4- All firms simultaneously price their offerings, p_p , p_o , and p_c .
- 5- Consumers decide whether to stay out of the market or purchase from one of the three service providers.

2.4 Consumer Market Equilibrium and Pricing

2.4.1 Consumer Market Equilibrium

We begin by examining the final stage of the game at which point each consumer either selects a service provider or chooses to remain out of the market. At this point, all providers' effort levels (e_p , e_o , and e_c) and prices (p_p , p_o , and p_c) have already been set. Hence, the consumers observe total qualities (Q_p , Q_o , and Q_c) and prices and make consumption decisions. Each consumer $\theta \in \Theta$ can select a strategy from one of four options: P (contract with the proprietor), O (contract with the originator), C (contract with the contributor), or N (not contract with any provider). The resulting consumer market equilibrium critically depends on the ordering of total qualities in magnitude and is indifferent with regard to who provides it. Thus, without loss of generality, we characterize this equilibrium when $Q_i > Q_j > Q_k$ where $i, j, k \in \{p, o, c\}$ and $i \neq j \neq k$.²

Eight outcomes are possible in equilibrium, which account for all combinations of having each firm service a positive mass or zero mass of consumers. In the Appendix, we present a complete characterization of the consumer market equilibrium along with its proof.³ However, in light of our research objectives, one outcome where all three service providers exist in the marketplace and service positive masses of consumers is particularly relevant to our research questions. Because we examine parameter regimes where endogenous qualities and prices lead to this consumer market outcome, we for-

²Because qualities are endogenous to the model, including consumer market equilibria for the cases where either $Q_i = Q_j$ or $Q_j = Q_k$ is not essential. We formally establish such outcomes do not arise in subsequent analysis of the effort game's equilibrium.

³Please see Lemma A.1 in the Appendix for complete details on the consumer market equilibrium.

malize it in the following lemma.⁴

Lemma 1 *For fixed prices p_i , p_j , and p_k , and qualities $Q_i > Q_j > Q_k$, if $p_i < Q_i$, $p_i - Q_i + Q_j \leq p_j < p_i Q_j / Q_i$, and $\frac{p_j(Q_k - Q_k) - p_i(Q_j - Q_k)}{Q_i - Q_j} \leq p_k < p_j Q_k / Q_j$, are satisfied, then the consumer market has the following threshold characterization:*

- (a) *consumers with $\theta \in (\theta_{ij}, 1]$ contract with firm i ,*
- (b) *consumers with $\theta \in (\theta_{jk}, \theta_{ij}]$ contract with firm j ,*
- (c) *consumers with $\theta \in (\theta_k, \theta_{jk}]$ contract with firm k ,*
- (d) *consumers with $\theta \in [0, \theta_k]$ do not use the software,*

where $\theta_{ij} = \frac{p_i - p_j}{Q_i - Q_j}$, $\theta_{jk} = \frac{p_j - p_k}{Q_j - Q_k}$, $\theta_k = \frac{p_k}{Q_k}$.

Intuitively, this lemma formalizes that the highest quality firm services the top tier of the consumer market. The second quality firm services the middle tier, and the third quality firm services the lower tier. Consumers with the lowest types find it preferable to remain out of the market considering the prices of the software solutions and the relatively low utility these consumers derive. To benefit the mathematical exposition, it is useful to denote the equilibrium strategy profile in the consumer market as $\sigma^*(\theta | \mathbf{Q}, \mathbf{p})$ which takes vectors (boldface) of qualities and prices as given and maps each consumer type to her equilibrium strategy as prescribed by Lemma 1; specifically, $\mathbf{Q} = (Q_p, Q_o, Q_c)$ and $\mathbf{p} = (p_p, p_o, p_c)$.

⁴For all formal statements of lemmas and propositions, the technical proofs are provided in the Appendix.

As established in Lemma 1, consumers are essentially agnostic as to who provides the higher quality offering. If the contributor chooses to pour in effort to the extent that its offering carries the highest quality, then consumers with the highest types would have strong incentives to contract with the contributor in equilibrium. However, as we turn our attention in the next section to the effort choice problems faced by the three providers, we establish that for fairly broad parameter regimes, an outcome where $Q_p > Q_o > Q_c$ prevails in equilibrium. Because this quality ordering is so commonly observed in enterprise software markets, we carefully focus attention on such regimes with a goal of better understanding how efforts are leveraged in the open-source domain to compete on quality.

2.4.2 Strategic Pricing of Software Solutions

Given total qualities and an understanding of how prices influence the consumer market equilibrium, in the second-to-last stage, the three providers compete in the software market on prices. In particular, at this stage, the service providers consider their initial investments as sunk. Thus, the relevant profit functions for this stage can be defined in a straightforward manner by

$$\tilde{\Pi}_p(p_p | p_o, p_c, \mathbf{Q}) = (p_p - c) \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}, \mathbf{p}) = p\}} d\theta, \quad (2.4)$$

$$\tilde{\Pi}_o(p_o | p_p, p_c, \mathbf{Q}) = (p_o - c) \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}, \mathbf{p}) = o\}} d\theta, \quad (2.5)$$

and

$$\tilde{\Pi}_c(p_c | p_p, p_o, \mathbf{Q}) = (p_c - c) \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}, \mathbf{p}) = c\}} d\theta, \quad (2.6)$$

respectively. Each provider maximizes its current-stage profit function taking the other two providers' prices as given. Intersecting these best response price functions gives rise to the simultaneously set Nash equilibrium prices which we denote by $p_p^*(\mathbf{Q})$, $p_o^*(\mathbf{Q})$, and $p_c^*(\mathbf{Q})$.

Focusing on the case where qualities satisfy $Q_p > Q_o > Q_c > c$, the Nash equilibrium prices fall into one of three regions depending on the level of service costs. If the service cost is low and satisfies $c < \tau_A$ where $\tau_A = \frac{Q_c(Q_p - Q_o)}{4Q_p - Q_o - 3Q_c}$, then under equilibrium pricing, all three service providers are contracted with by consumers. As the service cost rises and satisfies $\tau_A \leq c < \tau_B$, where the upper threshold is defined by

$$\tau_B = \frac{Q_c Q_o (Q_p - Q_o)}{2Q_p(Q_o - Q_c) + Q_o(Q_p - Q_c) + Q_o(Q_p - Q_o)},$$

in equilibrium, the proprietor and originator strategically set prices to squeeze the contributor out of the market. As the service cost rises even higher and satisfies $\tau_B \leq c < \tau_C$ where $\tau_C = Q_o/2$, then the contributor is more naturally out of the market under equilibrium pricing; in this case, the prices are interior points of the proprietor and originator profit functions rather than chosen on a boundary.

Similarly, when $\tau_C \leq c < \tau_D$ where $\tau_D = \frac{Q_o Q_p}{2Q_p - Q_o}$, the proprietor also strategically prices the originator out of the market. Finally, provided $\tau_D \leq c$, even by setting an interior profit-maximizing price, the proprietor is the only provider servicing consumers in equilibrium.

In the following, we will focus on wide parameter regions that yield equilibrium

outcomes where the proprietor, originator, and contributor are all in the market servicing a positive share of the consumer population. In these regions, $c < \tau_A$ is satisfied, and the Nash equilibrium prices are given in the following lemma.

Lemma 2 *If $Q_p > Q_o > Q_c > c$ and $c < \tau_A$ are satisfied, then the Nash equilibrium prices to the simultaneous price-setting subgame are characterized by:*

$$p_p^* = \frac{(Q_p - Q_o)(Q_p(4Q_o - Q_c) - 3Q_oQ_c) + c(Q_p(7Q_o - Q_c) - Q_o(Q_o + 5Q_c))}{2(Q_p(4Q_o - Q_c) - Q_o(Q_o + 2Q_c))} \quad (2.7)$$

$$p_o^* = \frac{(Q_p - Q_o)Q_o(Q_o - Q_c) + 3cQ_o(Q_p - Q_c)}{Q_p(4Q_o - Q_c) - Q_o(Q_o + 2Q_c)} \quad (2.8)$$

and

$$p_c^* = \frac{(Q_p - Q_o)(Q_o - Q_c)Q_c + c(4Q_pQ_o - Q_o^2 + 2Q_pQ_c - 2Q_oQ_c - 3Q_c^2)}{2(Q_p(4Q_o - Q_c) - Q_o(Q_o + 2Q_c))}. \quad (2.9)$$

Lemma 2 establishes the relationship between the equilibrium prices and the qualities brought to the market by these providers. We can now turn our attention to the sequential effort selection investment problems that determine these qualities.

2.4.3 Investment and Quality Contribution

In this section, we build a mathematical formulation of the decision problems faced by the three types of software providers in an enterprise market. Typically, a proprietor innovates, invests in software and service development, and begins providing its

offering in the market. Subsequently, in the open-source domain, an originator introduces an open-source alternative software offering after incurring its own investments. Third, a contributor to the open-source product may also invest and compete in the services market. Thus, given e_p and e_o , we first describe the contributor's effort investment decision problem. Its profit function at this stage can be written as

$$\Pi_c(e_c | e_p, e_o) = (p_c^*(\mathbf{Q}(e_c)) - c) \left(\int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}(e_c), \mathbf{p}^*(\mathbf{Q}(e_c))) = C\}} d\theta \right) - \beta_c e_c^2 / 2. \quad (2.10)$$

By maximizing (2.10) over $e_c \in \mathbb{R}_+$, we denote the contributor's best response to each possible set of proprietor and originator efforts with $e_c^*(e_p, e_o)$.

Rolling back to the originator's decision problem, its profit function can be written as

$$\begin{aligned} \Pi_o(e_o | e_p) &= (p_o^*(\mathbf{Q}(e_c^*(e_o), e_o)) - c) \left(\int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}(e_c^*(e_o), e_o), \mathbf{p}^*(\mathbf{Q}(e_c^*(e_o), e_o))) = O\}} d\theta \right) \\ &\quad - \beta_o e_o^2 / 2, \end{aligned} \quad (2.11)$$

and similarly, by maximizing (2.11) over $e_o \in \mathbb{R}_+$, we denote the originator's best response to each possible proprietor's effort level with $e_o^*(e_p)$.

Finally, we examine the initial effort investment decision faced by the proprietor. Its profit function can be written as

$$\Pi_p(e_p) = (p_p^*(\mathbf{Q}(e_p)) - c) \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}(e_p), \mathbf{p}^*(\mathbf{Q}(e_p))) = P\}} d\theta - \beta_p e_p^2 / 2, \quad (2.12)$$

where we use the shorthand notation $\mathbf{Q}(e_p) = \mathbf{Q}(e_c^*(e_p, e_o^*(e_p)), e_o^*(e_p), e_p)$. Similar to the other providers, the proprietor maximizes (2.12) over $e_p \in \mathbb{R}_+$. We denote the equilibrium to this sequential effort-selection game with $\mathbf{e}^* = (e_p^*, e_o^*, e_c^*)$.

Because prices are easily adapted without timing concerns, all three providers set prices simultaneously in the market. As formulated above, this seemingly simple setting is mathematically complex, involving six optimization problems with four levels of nesting. Given its complexity, it is not analytically possible to characterize the equilibrium solution over the entire parameter space. For the benefit of the paper's exposition, we focus on two parameter regions that are both relevant to the software industry and through which we are able to provide insights into our central research questions in this study.

In the next section, we elaborate on these parameter regions while connecting them to observations in enterprise software markets. For each region, we then characterize the equilibrium solution which permits a discussion of how quality competition unfolds in these markets. Importantly, we generate insights into how licensing affects the qualities of software offerings that are provided in these competitive markets.

2.5 Oligopolistic Competition in Enterprise Software Markets

As we laid out in Section 2.3, firms are generally heterogeneous with regard to their respective development costs. Recall the cost efficiency parameter is given by β_p ,

β_o , and β_c for the proprietor, originator, and contributor, respectively. In this section, we focus on two regions which differ in the degree of dispersion in the cost efficiencies of these three providers.

2.5.1 High Cost Dispersion

Efforts and Qualities

The first region of focus, henceforth referred to as Region H, captures a scenario where the cost efficiencies of the three providers are fairly dispersed. In particular, the proprietor is more cost efficient in development than the originator who is in turn more cost efficient than the contributor. Analytically, we define $\beta_p = k_p b^2$ and $\beta_o = k_o b$ and examine the region where $b < \bar{b}$, $s_{co} < s_o \lambda_H$, and $s_{oc} < s_c / \lambda_H$ for constants $k_p, k_o, \bar{b}, \lambda_H > 0$ such that $\beta_p < \beta_o < \beta_c$ and the open-source contributor can only benefit from the originator's efforts to a limited extent.⁵ Taken together, these conditions on parameters comprise the mathematical representation of Region H.

Intuitively, market structures vary considerably over different classes of software products, and there are many examples where the proprietor is a large organization with extensive resources enabling it to develop products in a cost-efficient manner, whereas an open-source originator competing in the same market is a smaller organization with relatively more constrained development resources. Nevertheless, the open-source originating firm has still sufficiently organized to economize on development efforts to some extent. At the most constrained end of the spectrum, a contributor to the open-source

⁵All of these parameters are described in detail in Section 2.3.

project may invest in development and service at a small scale in order to provide basic consulting and implementation services to the market, leveraging expertise derived from its investments.

As an example, for operating systems, Microsoft has the engineering resources and the scale to invest heavily and efficiently in development of its Windows server products. Red Hat competes in the same market by offering Red Hat Enterprise Linux which is an enterprise offering of its open-source software. Red Hat is a much smaller organization facing more pressing development constraints, i.e., its investment cost function is more convex. Contributors to the Fedora community project which feeds future Red Hat Enterprise Linux releases also provide value-added services in the market, but these contributors are the most constrained in terms of development resources. A similar structure was observed with the large proprietor Oracle in the database management systems market, MySQL being a leading open-source originator, and a contributor such as Percona who leverages its own development expertise to contract with consumers in the same market.⁶ With regard to enterprise resource planning (ERP) software and customer relationship management (CRM) software, large proprietors such as SAP and SalesForce establish their respective markets, and smaller originators such as OpenERP and SugarCRM bring open-source alternative offerings to the market subsequently. These smaller players clearly have cost efficiencies that are different and lower (modeled as greater convexity in their development cost functions).

⁶Later on, eventually after being acquired by Sun Microsystems in 2008, MySQL became the property of Oracle Corporation in a 2009 acquisition of Sun Microsystems.

In Region H where the providers are characterized by these differences in their abilities to invest efficiently in the development of their software offerings, we can characterize the equilibrium effort investment levels that arise in the sequential investment game that is typically found in these markets. The following lemma establishes the equilibrium effort levels.

Lemma 3 *There exists a constant $\bar{K} > 0$ such that, under the conditions of Region H, the equilibrium efforts of the proprietor, originator, and contributor satisfy*

$$\left| e_p^* - \left(\frac{s_p}{4k_p b^2} - \frac{4k_p K_1^3 s_o^5 (2s_{co}^2 + 13s_{co}s_o + 3s_o^2)}{k_o^2 K_2^6 s_p^3} - \frac{2k_p K_1 s_o^2 K_3 b}{\beta_c k_o^3 K_2^9 s_p^5} \right) \right| < \bar{K} b^2,$$

$$\left| e_o^* - \left(\frac{s_o^2 K_1}{k_o K_2^2 b} - \frac{8k_p K_1^3 s_o^4 (s_{co} + 2s_o)}{k_o^2 K_2^5 s_p^2} - \frac{K_4 b}{16\beta_c^2 k_o^3 K_2^7 K_1^2 s_o^4 s_p^4} \right) \right| < \bar{K} b^2,$$

and

$$\left| e_c^* - \left(\frac{s_c s_o^2 (4s_o - 7s_{co}) + s_{co}^2 (s_{co} + 2s_o) s_{oc}}{4\beta_c K_2^3} - \frac{K_5 b}{8k_o K_1 K_2^6 s_o^2 s_p^2 \beta_c^2} \right) \right| < \bar{K} b^2,$$

respectively, where $K_1 = s_o - s_{co}$ and $K_2 = 4s_o - s_{co}$.⁷

Consistent with their heterogeneity in cost efficiencies, the three providers invest to varying extent. In particular, their overall investments in equilibrium \mathbf{e}^* determine the quality of offerings in the market. It directly follows from Lemma 3 that the equilibrium qualities of the offerings satisfy the following ordering: $Q_p(\mathbf{e}^*) \gg Q_o(\mathbf{e}^*) > Q_c(\mathbf{e}^*)$. The proprietor is the clear quality leader in the market while, because of cross effects, the

⁷The other constants, K_3 , K_4 , and K_5 , are fully characterized in the appendix.

originator and contributor compete slightly more aggressively although the originator's offering bears a higher quality in equilibrium. Because with open-source software there exist cross-effects of effort investments on the qualities of the open-source offerings, we aim to explore how the strength of these effort interactions affects quality competition among the three providers.

A parameter of primary interest is s_{oc} which reflects the ability of the open-source originator to benefit from the effort investments of the contributor toward the common open-source product, which is used as the basis for both of their offerings in the subsequent services market. There is a close connection between s_{oc} and the various open-source licensing models that are currently employed. In the open-source community, there are many different licenses that are used, some more commonly than others. For example, the Open Source Initiative who maintains the definition of what it means to be "open source" lists around 70 different open-source licenses that satisfy their definition (OSI 2014) and provides detailed information on each license. One significant dimension along which open-source licenses vary is the degree of restrictiveness with regards to rights granted to the community working with or using the open-source software in question, ranging from permissive BSD-style licenses to restrictive GPL-style licenses.

It is quite typical that an enterprise information system is composed of several, distinct software products working together to achieve a business objective. A simple e-commerce site for a small business would require one or more servers running an operating system (e.g., Windows, Red Hat Enterprise Linux, and Oracle Solaris), webserver software (e.g., Microsoft IIS, Apache HTTP Server, and IBM HTTP Server), applica-

tion server software (e.g., IBM Websphere, Oracle WebLogic, and Red Hat JBoss), and a database management system (e.g., Microsoft SQL Server, Oracle, MySQL, and PostgreSQL). For an open-source originator intending to compete in a particular software market, the open-source license of its product may often be governed by the open-source licenses of existing software that its product requires as dependencies, particularly if the originator's intent is to distribute everything together as part of a complete solution to its enterprise customers.

Notably, the restrictiveness of a license alters the incentives of economically-driven contributors to invest in a given open-source software product. In our model, the parameter s_{oc} essentially captures this feature. The more restrictive the license (i.e., s_{oc} increases), the originator is able to benefit to a greater extent from subsequent contributor efforts. In the following, we explore what impact the prevalence of open-source software with restrictive licenses and the constraints they impose has on competition among the strategic players in an enterprise software market.

Proposition 1 *Under the conditions of Region H, if the licensing of the open-source software becomes more restrictive license, i.e., if s_{oc} increases:*

- (i) *The proprietor's invested effort e_p^* decreases, whereas the contributor's invested effort e_c^* increases ;*
- (ii) *The originator's invested effort e_o^* decreases if the contributor's cost parameter is high. However if the contributor's cost parameter is low, then the originator's effort increases in s_{oc} up to a threshold value, \bar{s}_{oc} , and then decreases in s_{oc} beyond \bar{s}_{oc} .*

Technically, there exists a $\bar{\beta}_c > 0$ such that $\frac{de_o^*}{ds_{oc}} < 0$ if $\beta_c \geq \bar{\beta}_c$.⁸ However, if $\beta_c < \bar{\beta}_c$ then there exists $\bar{s}_{oc} \in (0, s_c s_o / s_{co})$ such that $\frac{de_o^*}{ds_{oc}} > 0$ for $s_{oc} \in (0, \bar{s}_{oc})$ and $\frac{de_o^*}{ds_{oc}} < 0$ for $s_{oc} \in (\bar{s}_{oc}, s_c s_o / s_{co})$;

(iii) The proprietor's quality $Q_p(\mathbf{e}^*)$ decreases, whereas both the originator and contributor's qualities, $Q_o(\mathbf{e}^*)$ and $Q_c(\mathbf{e}^*)$, respectively, increase.

Proposition 1 establishes that a restrictive license will intensify competition in markets exhibiting properties consistent with Region H. Specifically, part (iii) of Proposition 1 demonstrates that the proprietor's quality decreases whereas both the originator and contributor's qualities increase in equilibrium. In that these qualities are determined by the equilibrium effort levels that arise in the sequential effort investment game, we next discuss how strategic interactions among them are affected by licensing.

One particularly noteworthy interaction highlighted here is that more restrictive licenses can actually increase the economic incentive for contributors to invest effort into open-source software. Part (i) of Proposition 1 formally demonstrates that this positive relationship between e_c^* and s_{oc} occurs in Region H, i.e., the contributor's equilibrium effort level is strictly increasing in license restrictiveness. Recall that $Q_o = s_o e_o^* + s_{oc} e_c^*$ which is to say that an increase in the parameter s_{oc} serves to help the originator better leverage the contributor's efforts toward the originator's own service offerings. Then, the originator can free ride on the contributor's efforts which by common sense can *reduce* the contributor's incentives to incur these investments. Yet, our model establishes the

⁸An analytical expression for $\bar{\beta}_c$ is provided in the appendix.

opposite behavior can arise instead.

Despite the originator's free riding, the contributor can have a significant incentive to *help* the originator become a stronger competitor against the proprietor. If the total quality of the originator's offering becomes higher, then the originator also becomes closer in the quality space to the proprietor, creating space at the low end of the consumer market. Thus, the contributor can offer a higher quality offering at the low end where it can charge a higher price. However, if the extent to which the originator free rides on the contributor's effort is too substantial, i.e., markedly scaling back its own effort investment as the contributor increases its investment, then the originator's resultant quality would not increase sufficiently enough to enable the contributor to generate returns at the low end of the market.

In part (ii) of Proposition 1, we answer this question by formally studying how the originator behaves, in equilibrium, considering changes in the restrictiveness of the license and the strategic responses of the proprietor and contributor. When the ability of a contributor to increase the quality of its own offering is limited because its investment cost function is highly convex, i.e., $\beta_c \geq \bar{\beta}_c$, then the originator has an incentive to scale back its own effort to induce the contributor to invest more. As s_{oc} increases which is to say licensing becomes more restrictive, the originator reduces its own effort and free rides on the contributor whose increased effort, combined with a more restrictive license, effectively pushes the quality of the originator higher and out of the contributor's desired market at the low end. A similar incentive is found when the contributor has the ability to cost-efficiently increase its own quality, i.e., $\beta_c < \bar{\beta}_c$, but the originator can strongly

leverage that effort toward its own quality offering, i.e., s_{oc} is at the high end of the range. Again, in this case, the originator scales back effort as s_{oc} increases within this range to induce the contributor to invest more to boost its own quality as well as the originator's quality.

In Figure 2.2, this behavior is illustrated in the right-hand portion of panels (b), (c), (e) and (f). Specifically, panel (b) shows how the originator decreases its effort as s_{oc} increases, which places the contributor in a position where it must increase its investment as seen in panel (c). The contributor's increase in effort leads to a net positive effect on both the originator's quality and the contributor's quality which is shown in right-hand side of panels (e) and (f), respectively. Closely examining the slope of the curves in panels (e) and (f), it becomes apparent that at the higher end of the range of s_{oc} , the primary focus of the contributor's increased effort is to heavily boost the originator's quality; its own quality increase is marginal in comparison. While the contributor's behavior is counter intuitive on the surface, inducing a larger quality gap between the offerings is actually profitable to the contributor, and it also helps the originator compete more strongly with the proprietor.

On the other hand, when the contributor is cost efficient ($\beta_c < \bar{\beta}_c$) and the originator cannot strongly leverage the contributor's effort toward its own quality offering, i.e., s_{oc} is at the low end of its range, the originator's investment incentive is significantly altered. In particular, the originator is wary of the contributor whose investment benefits its own quality significantly more than it benefits the originator's quality. Thus, as s_{oc} increases, the contributor has an incentive to significantly increase its own quality

and compete more strongly against the originator whose quality is marginally higher as a result of this higher investment. Because of the contributor's strategic behavior, the originator finds it preferable to also make a larger effort investment and increase its own quality to maintain a sufficient quality gap and avoid excessive price competition. The left-hand side of panel (b) illustrates how the originator sharply increases effort as s_{oc} increases, which leads to a sharp increase in the originator's quality as seen in the same portion of panel (e).

When the cost functions of the three providers exhibit high dispersion (Region H), the proprietor is significantly more cost efficient in the development of its software offering. However, the open-source mode of production enables the originator and contributor to still achieve an increased quality in their offerings by leveraging the joint investment from both organizations which has an additive effect on quality (i.e., $Q_o = s_o e_o^* + s_{oc} e_c^*$ and $Q_c = s_c e_c^* + s_{co} e_o^*$). In fact, the nature of open source helps to distribute development costs and enable both contributing providers to operate on the less convex part of their cost functions while still achieving higher quality levels through their separate, additive contributions. Recognizing this open-source production advantage, the proprietor leverages its own cost efficiency and first mover advantage to invest heavily and bring a high quality offering to the market. This ensures that even though open-source offerings benefit from these cross-effort effects, there will still be sufficient distance between the three quality offerings in equilibrium.

Nevertheless, when competing against open-source production, a proprietor does not generate as significant of a return on its effort investments as it would as a monopolist

or another proprietor who incurs all development costs internally and cannot leverage an open-source community. Because of these reasons, a proprietor's net incentive is to throttle its own investment if the open-source providers are able to generate higher quality in more restrictive licensing regimes. Proposition 1 formalizes that the proprietor reduces e_p^* and furthermore $Q_p(\mathbf{e}^*)$ decreases as s_{oc} increases throughout its domain. Panels (a) and (d) of Figure 2.2 illustrate the impact of license restrictiveness on the proprietor's equilibrium effort level and quality.

Consumer Surplus and Social Welfare

Because license restrictiveness greatly impacts the incentives of all providers to incur investments and compete in the services market, we next turn attention to examining the aggregate effect of restrictiveness on the market. In particular, we are concerned with how licensing affects the market overall due to the qualities of the offerings brought to market and the respective price competition that results. For our model, consumer surplus is defined as

$$CS = \sum_{k \in \{p,o,c\}} \int_{\Theta} (\mathbf{Q}_k(\mathbf{e}^*)\theta - p_k^*) \cdot \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}(\mathbf{e}^*), \mathbf{p}^*) = k\}} d\theta, \quad (2.13)$$

and, similarly, social welfare can be measured as

$$SW = \sum_{k \in \{p,o,c\}} \Pi(e_k^*) + CS. \quad (2.14)$$

As seen in Proposition 1 and Figure 2.2, an increase in license restrictiveness greatly affects the equilibrium efforts exerted by the proprietor, originator, and contributor, and thus the equilibrium quality levels each of these firms brings to the market. Further, because of changes in the quality of offerings, price competition is also significantly affected. Because qualities and prices determine the extent to which consumers benefit in these markets, we next examine how consumer surplus and social welfare are affected by license restrictiveness. The following proposition establishes its impact on these measures in markets where the providers exhibit high cost dispersion.

Proposition 2 *Under the conditions of Region H, both consumer surplus and social welfare increase as the license governing the open-source software has a greater degree of restrictiveness.*

Above in Figure 2.2, we illustrated how the equilibrium quality level of the proprietor decreases whereas the equilibrium quality levels of the originator and contributor both increase in the license restrictiveness parameter s_{oc} , which is also formalized in part (iii) of Proposition 1. Because the providers strategically interact in a manner where the quality of the solutions they provide are responding in mixed directions to changes in license restrictiveness, how the consumer surplus associated with the offerings is affected is unclear. Moreover, when the providers engage in pricing competition subsequently after qualities are determined, the fact that the highest quality provider, the proprietor, is decreasing its quality and the two lower quality providers, the originator and contributor, in equilibrium are increasing their qualities may suggest that pricing competition is

stiffening which could be beneficial to consumer surplus.

In order to better ascertain how prices and consumer choice are affected, we illustrate in Figure 2.4 how the equilibrium prices are affected by s_{oc} and, in turn, how equilibrium consumer choices respond to equilibrium qualities and prices. For convenience, we denote the demand generated in equilibrium by each provider with

$$D_p = \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}^*, \mathbf{p}^*) = P\}} d\theta, \quad (2.15)$$

$$D_o = \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}^*, \mathbf{p}^*) = O\}} d\theta, \quad (2.16)$$

and

$$D_c = \int_{\Theta} \mathbf{1}_{\{\sigma^*(\theta | \mathbf{Q}^*, \mathbf{p}^*) = C\}} d\theta, \quad (2.17)$$

and use this notation in panels (d), (e), and (f) in Figure 2.4.

Examining panel (d) of Figure 2.2 and panel (a) of Figure 2.4, although the proprietor decreases its quality, it decreases its price even more sharply to compensate its reduced effort. In fact, panel (d) of Figure 2.4 illustrates that the net effect of how it adapts its price and quality in response to s_{oc} becoming more restrictive leads to a greater demand. Part of the reason the proprietor must drop its price steeply relative to its quality decrease is due to the increased qualities brought to the market jointly by the open-source originator and contributor. Panels (b) and (c) of Figure 2.4 demonstrate how both the originator and contributor raise prices in equilibrium substantially to recoup their greater investments in quality associated with increased license restrictiveness. For both of these

providers, their higher quality / higher price offerings dampen their respective demand which is illustrated in panels (e) and (f) of Figure 2.4.

Aggregating the effects of quality and price competition among the three providers, in Figure 2.3, we examine how consumer surplus relates to license restrictiveness. Panel (a) clearly shows how the net effect of the proprietor's lower pricing coupled with lower quality in equilibrium is beneficial to consumers. Similarly, the originator's higher quality and higher priced offering still results in an aggregate positive contribution to consumer surplus. In fact, the only equilibrium behavior detrimental to consumer surplus is due to the contributor at the high end of the s_{oc} range. Recall from panel (b) of Figure 2.2 that it is actually the *originator* who has incentives to scale back effort with increases in s_{oc} at the high end of license restrictiveness. Because of the originator's decrease in equilibrium effort, the contributor's quality increase (as seen in panel (f) of Figure 2.2) is marginal in comparison to how the contributor raises price (as seen in panel (c) of Figure 2.4). Therefore, in this range, the consumer surplus associated with the contributor decreases with increases in s_{oc} .

Taking all three providers' equilibrium behavior discussed above into account, Proposition 2 formally establishes that in regions of high cost dispersion, consumer surplus will tend to increase with increasingly restrictive licenses that govern OSS. In this region, the restrictiveness of the license provides incentives for the originator and contributor to increase the quality of their products. Although the proprietor decreases quality, the market becomes much more competitive and the proprietor decreases its price to the extent that even consumer surplus associated with its own offering increases in

equilibrium. Although consumer surplus associated with the contributor's offering can be lower at the high end of the s_{oc} range and some consumers at the low end of the type space are driven out of the market due to increased prices, which is illustrated by panels (c) and (f) of Figure 2.4, the aggregate impact on consumer surplus is positive. This is primarily driven by the competitive pricing by the proprietor as it is forced to compete closer in the quality space against the other two providers; *the market has become more competitive.*

The effects stemming from the boost to consumer surplus extend to social welfare which also increases in license restrictiveness as is stated in Proposition 2. Because of the stiffer competition on quality and price, the proprietor's profits decrease in equilibrium. In contrast, the open-source providers leverage license restrictiveness to generate higher quality offerings and tend to increase their respective profits. Because the proprietor's profit loss is mostly a contribution to consumer surplus, the net effect on social welfare stemming from the increased competitiveness in the market is still positive.

2.5.2 Low Cost Dispersion

Efforts and Qualities

The second region of focus, henceforth referred to as Region L, captures a scenario in which the cost efficiencies of the three providers are much less dispersed in the parameter space (in contrast to Region H). This is a more competitive setting where the proprietor, originator, and contributor have similar abilities to invest in development

and compete aggressively in the market for services. Analytically, we define $\beta_p = \hat{k}_p b$, $\beta_o = \hat{k}_o b$, $\beta_c = \hat{k}_c b$, $c = \kappa b$, $s_c = \kappa_c b$, $s_{co} = \kappa_{co} b$, and then examine the region where $b < \bar{b}$, $s_p^2 / \hat{k}_p \geq s_o^2 / \hat{k}_o$, and $s_{oc} < \lambda_L s_o \kappa_c / \kappa_{co}$, for constants $\hat{k}_p, \hat{k}_o, \hat{k}_c, \kappa, \kappa_c, \kappa_{co}, \bar{b}, \lambda_L > 0$ such that the open-source contributor can only benefit from the originator's efforts to a limited extent (relative to the originator's ability to leverage the contributor). Taken together, these conditions on parameters comprise the mathematical representation of Region L.

In contrast to Region H, both proprietary and open-source firms have similar capabilities in some software markets. For example, in the late 1990s and early 2000s, BEA Systems and JBoss were competing in the enterprise Java application server market. BEA System's flagship product WebLogic was proprietary, whereas the JBoss Application Server was an open-source product with a revenue model primarily based on services. Both companies were fairly new and of similar capabilities at the time.⁹

Another example where the landscape is more competitive due to the nature of the market relates to software development frameworks. In many cases, firms create and maintain software development frameworks in order to derive service revenues from other organizations who build upon these frameworks and require services. These markets tend to have a wide variety of providers, both open source and proprietary and dependent on the programming language on which the framework is based. For example, Ruby on Rails is a popular open-source development framework that has gained traction over the last decade for expediting web application development. On the other hand, the

⁹Both JBoss and BEA Systems eventually were acquired by Red Hat in 2006 and Oracle in 2008, respectively.

Base One Foundation Component Library (BFC) is a proprietary development framework for ASP.NET. In that software development frameworks tend to be much more lightweight in comparison to the software behind operating systems, database management systems, and application servers, even if larger firms such as Microsoft, IBM, and Apple compete against smaller developer firms in these markets, their respective development capabilities will be more similar.

In Region L where the providers have similar cost efficiencies, we can characterize the equilibrium effort investment levels that arise in the sequential investment game that is typically found in these markets. The following lemma characterizes these equilibrium effort levels.

Lemma 4 *There exists a constant $\bar{K} > 0$ such that, under the conditions of Region L, the equilibrium efforts of the proprietor, originator, and contributor satisfy*

$$\left| e_p^* - \left(\frac{M_1 s_o^2}{r k_o M_2^3 s_p b} - \frac{M_6}{k_c M_2^3 s_o (k_p M_4^3 s_o^2 + 24 r^3 k_o M_5 s_p^2)} \right) \right| < \bar{K} b,$$

$$\left| e_o^* - \left(\frac{M_1 s_o}{k_o M_2^3 b} + \frac{M_7}{2 k_c k_o M_2^3 s_o^2 (k_p M_4^3 s_o^2 + 24 r^3 k_o M_5 s_p^2)} \right) \right| < \bar{K} b,$$

and

$$\left| e_c^* - \frac{M_3 (\kappa_c M_2 M_3 s_o - 6 r \kappa_{co} s_{oc})}{4 k_c M_2^3 s_o} \right| < \bar{K} b,$$

respectively, where there exists a unique $r \in (0, 4/7)$ satisfying

$$-64 + 176r - 196r^2 + 147r^3 + 4r(64 - 112r + 132r^2 - 139r^3 + 28r^4)t = 0$$

$$\text{and } t = \frac{\hat{k}_o s_p^2}{\hat{k}_p s_o^2}.^{10}$$

Lemma 4 establishes that even if the providers' cost efficiencies are more homogeneous, the sequential timing of their effort choices leads to an equilibrium outcome where the qualities of their offerings satisfy $Q_p(\mathbf{e}^*) > Q_o(\mathbf{e}^*) > Q_c(\mathbf{e}^*)$.¹¹ Even though the equilibrium ordering of qualities matches what is found in Region H, the providers' more similar development abilities in Region L will lead to significantly different strategic interactions. This fact becomes clear as we examine the impact of license restrictiveness on the equilibrium effort investments and resulting qualities.

Proposition 3 *Under the conditions of Region L, if the licensing of the open-source software becomes more restrictive, i.e., if s_{oc} increases:*

(i) *The invested effort levels of the proprietor e_p^* , originator e_o^* , and contributor e_c^* all decrease in equilibrium ;*

(ii) *The proprietor's quality $Q_p(\mathbf{e}^*)$ and contributor's quality $Q_c(\mathbf{e}^*)$ both decrease.*

However, the originator's quality $Q_o(\mathbf{e}^)$ increases and then decreases in s_{oc} . Technically, there exists $\bar{s}_{oc} \in (0, \lambda_L s_o \kappa_c / \kappa_{co})$ such that $\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} > 0$ for $s_{oc} \in (0, \bar{s}_{oc})$ and*

$\frac{dQ_o(\mathbf{e}^)}{ds_{oc}} < 0$ for $s_{oc} \in (\bar{s}_{oc}, \lambda_L s_o \kappa_c / \kappa_{co})$.*

Previously in Section 2.5.2, we learned that when firm capabilities are characterized by a high degree of cost dispersion (Region H), more restrictive licenses can

¹⁰The other constants, M_i for $i = 1, 2, \dots, 7$ are fully characterized in the Appendix.

¹¹This ordering follows from substituting the expressions for the equilibrium effort levels given in Lemma 4 into the quality expressions introduced in Section 2.3.

actually increase the economic incentives for originators and contributors to invest effort into OSS. In contrast, part (i) of Proposition 3 establishes that as firms become more similar in development capability, their incentives to invest in development are substantially and negatively altered; in particular, the strategic interaction between the originator and contributor on the open-source side of the market results in reduced investments in equilibrium as restrictiveness increases. As before, because $Q_o = s_o e_o^* + s_{oc} e_c^*$, the originator has an incentive to free ride on the contributor's efforts, but the contributor also has an incentive to drive the originator's quality closer in the quality space to the proprietor, creating space at the low end of the consumer market such that the contributor can offer a higher quality offering. However, because of the greater similarity in development capabilities in Region L, the proprietor strategically limits the ability of the open-source firms from leveraging the complementarities that stem from restrictive licenses.

In particular, because the open-source firms are similarly cost efficient in development, the proprietor is more wary of investing heavily in quality. If it does so, the open-source originator can also invest efficiently and leverage contributor efforts to compete intensely and limit the proprietor's return on its development investment. Because the proprietor can foresee this problem, under the conditions of Region L, it invests to a lesser degree and offers a lower quality product. Given the proprietor's quality, the originator can cost efficiently increase the quality of its offering in the market but, in this case, it becomes more concerned with and essentially more constrained by the prior quality choice commitment of the proprietor than by its own development costs. Hence, when facing a more restrictive license, the originator simply has heightened incentives

to scale back its own investment and substitute it with the contributor's investment to a greater degree. In a sense, the originator does not want its quality offering to increase too much because it only leads to more severe price competition with the proprietor who has chosen a lower position in the quality space already. In this situation, the contributor also now no longer has incentives (as in Region H) to contribute heavily in order to push the originator further up in the quality space; indeed, the contributor realizes in equilibrium the originator will strategically not allow it.

In Figure 2.5, we show how the equilibrium efforts are impacted by license restrictiveness and the consequence on the total quality brought to the market. Consistent with part (i) of Proposition 3, all three providers reduce their equilibrium effort investments as s_{oc} increases which is illustrated in panels (a)-(c). The originator takes a balanced approach toward managing the total quality of its joint offering. As s_{oc} initially increases, it free rides to a greater degree on the contributor's effort while scaling back its own effort which is shown in panel (b) of Figure 2.5. Even though the contributor reduces its effort in response, the total contribution to the originator's quality from the contributor (i.e., $s_{oc}e_C^*$) still leads to a net increase in the quality of the originator's offering which is illustrated in the left-hand side of panel (e). Said differently, the originator adjusts its own effort while better leveraging the contributor's effort to improve the quality of the originator's offering, which is analytically established in part (ii) of Proposition 3. However, as s_{oc} increases to the more restrictive end of the spectrum, the proprietor is more concerned with the ability of the originator to leverage the contributor. As a result, the proprietor moves down sharply in the quality space, which is reflected in panels

(a) and (d). Given the proprietor's behavior, the originator also reduces its effort more sharply to drive the quality of its offering lower to avoid head-to-head competition with the proprietor by creating distance in the quality space from above. In turn, the contributor's quality is also more sharply affected negatively as seen in panel (f) of Figure 2.5 which also serves to alleviate price competition in the market for services.

How the equilibrium pricing and consumption decisions adapt in equilibrium is qualitatively different from the results under Region H (seen in Figure 2.4). In Region L, the contributor's effort and corresponding quality must drop substantially (in contrast to its increasing nature in Region H which is illustrated in panels (c) and (f) of Figure 2.2), as the contributor becomes increasingly constrained by the positions occupied by the proprietor and originator in the quality space. In fact, as illustrated in panel (c) of Figure 2.5, the contributor drops its effort substantially more (proportionally) than the decreases in effort made by either the proprietor or originator. Correspondingly, the contributor also sharply decreases its price in equilibrium which is shown in panel (c) of Figure 2.6. Notably, the contributor accelerates its price drop as s_{oc} increases to the most restrictive end of the licensing continuum in order to gain market share as seen in panel (f). On the other hand, the originator's equilibrium price mirrors its inverse U-shape quality response to license restrictiveness, which is shown in panel (b). Similarly, the proprietor scales back its price in conjunction with its quality although it is not as affected by increases in s_{oc} as might be expected. In particular, an increase in s_{oc} could permit the originator to better leverage the contributor's effort toward bringing a higher quality offering to the market. However, because both the originator and contrib-

utor instead scale back effort substantially as s_{oc} increases due to the proprietor's initial quality choice, the license restrictiveness does not hurt the proprietor's profitability as is illustrated by its limited decrease in both quality and price.

Consumer Surplus and Social Welfare

Turning our attention to the aggregate impact of the above strategic behavior on the surplus consumers derive in the market, we obtain the following characterization in Region L.

Proposition 4 *Under the conditions of Region L, both consumer surplus and social welfare decrease as the license governing the open-source software becomes more restrictive, after an initial increase at the permissive end of the licensing spectrum.*

Proposition 4 highlights an important insight into the role open-source licensing should play in software markets. In particular, although some in the open-source community often advocate restrictive licenses such as GPL which have *copyleft* characteristics, such licensing can actually be detrimental to both consumer surplus and social welfare when considering the service-based business models that govern the incentives of the providers of the software offerings in the market.

To illustrate this point, panels (a), (b) and (c) of Figure 2.7 demonstrate that beyond a certain point of license restrictiveness, the consumer surplus associated with all three providers decreases as s_{oc} becomes larger (i.e., more restrictive). In markets characterized by software providers with similar development capabilities, a restrictive

license amplifies the incentive conflicts. Specifically, such a license makes the open-source community a greater threat to the proprietor due to how the collaborative nature of open-source development can potentially benefit from such licensing. Because of these concerns, the proprietor strategically brings to market a purposeful lower quality offering to force the open-source originator into a position where (i) it prefers to limit the synergies stemming from the open-source cross effort effects, and (ii) it cannot cost efficiently invest to pursue a strategy where it brings the highest quality offering to the market.

Therefore, under conditions where the providers in the market are more competitive and there is potential to increase consumer surplus, it is important for less restrictive licenses to be encouraged and supported by open-source communities. Contrasting this implication with Proposition 2, we see that more restrictive licenses seem to benefit society the most in markets where the characteristics among the providers exhibit significant dispersion. For example, this might occur when the proprietary firm is a clear and dominant leader such as in the case of Windows versus Linux where the open-source firms face greater resource constraints and lower cost efficiencies. GPL licenses can increase consumer surplus and social welfare under these market conditions, but they are not beneficial when the provider firms are closer in capability and more competitive; this is the essence of Proposition 4.

2.6 Concluding Remarks

In this paper, we study competition among open-source and proprietary software firms while considering the strategic interplay of open-source contributors who also invest in the development of OSS and compete in the services market. Although prior work has mostly focused on a mixed-model of duopolistic competition, the existence of strategic contributors who vie for service revenues can significantly alter competitive forces. We add to the literature by analyzing a model that includes all three strategic players competing in the market and then focus on the important role OSS licensing plays in moderating strategic contributions. In particular, because the open-source originator and contributor both collaborate toward developing the OSS product and compete against each other and the proprietary firm, the economic incentives associated with investment in OSS vary substantially depending on the license governing it as well as the market conditions. For example, the originator may prefer to leverage the contributor to compete more fiercely with the proprietor, while the contributor faces its own dilemma whether to exert more or less investment in developing the joint OSS product (i.e., more effort will contribute to quality yet allow the originator to free ride on this effort). The degree of license restrictiveness is a critical attribute that can either amplify or diminish such incentives.

We focus our modeling efforts on two types of market conditions. In the first set, the three firms are relatively diverse in terms of development costs, which is to say their capabilities are fairly disparate. In this case, we find that more restrictive licenses can

be quite beneficial. More restrictive licenses (such as GPL), by common sense, could reduce the strategic contributor's incentives to incur investments due to free riding by the originator. However, our model establishes the opposite behavior can arise instead, provided the proprietor has sufficient incentive to compete at a high enough quality level. In this case, the contributor is actually motivated to help the originator move closer to the proprietor in the quality space. This is largely a consequence of the nature of OSS co-production - the originator and contributor experience "cross effects" due to the collaborative OSS development environment. Because of this nature, when facing a more restrictive license, the contributor is willing to exert greater effort so as to help boost the quality of the originator's offering which, in turn, helps create space for the contributor to serve the lower end of the consumer market.

Anticipating this behavior, the proprietor also has incentives to strategically decrease quality and sharply reduce price in order to capture more market share and compete aggressively with the OSS offerings. Thus, if the contributor increases effort, the originator is squeezed in the middle and becomes more sensitive in deciding whether to increase or scale back its own investment. Because of the cross effects, the originator's quality increases regardless, but in contrast to the proprietor, both OSS firms increase prices in equilibrium. We find that the combined effect still results in an aggregate positive contribution to consumer surplus, primarily driven by the proprietor being forced to become more competitive on pricing. Ultimately, this intensified competition benefits consumers.

In the second set of market conditions, we study a setting in which the devel-

opment capabilities of the firms are more homogeneous, which is seemingly a more competitive scenario. We find that all three firms scale back their effort investments in equilibrium when facing a more restrictive license. Further, these reduced investments are reflected by lower qualities for both the proprietor and contributor. However, the originator can strategically manipulate its own investment and again harness cross effects from the contributor to possibly increase the quality of its offering. Because of the greater similarity in development capabilities, the proprietor sees the OSS offerings as a greater threat and strategically limits the originator's ability to leverage the stronger complementarities that stem from restrictive licenses. Specifically, the proprietor brings to market a purposeful, lower quality offering to force the originator into a position where it prefers to limit such complementarities; in fact, the originator instead generally prefers to lower its quality so as to avoid head-to-head price competition with the proprietor. Anticipating the originator's incentive to throttle effort, the contributor also lacks incentives to contribute heavily as in the previous case.

Taken in aggregate, the reduced qualities tend to reduce consumer surplus and social welfare. Thus, in this case, more permissive licenses (such as BSD) would be beneficial. Further, it is important for such licenses to be encouraged by policy makers as well as open-source communities. This highlights an important insight into the role open-source licensing should play in software markets. In particular, although some in the open-source community often advocate restrictive licenses such as GPL which have copyleft characteristics, such licensing can actually amplify the incentive conflicts and become detrimental to both consumer surplus and social welfare in competitive markets

where OSS firms are strategically vying for service revenue.

Open-source software has become a mainstay for businesses as they compete in dynamic environments that reward flexibility and agility. Provision of value-added services is critical in this context and is an essential aspect of the commercial OSS business model. By developing a better understanding of competition among open-source contributors and proprietary firms, we aim to provide both organizations and policy makers with insights into how licensing can affect market outcomes. These insights can help guide software firms as they determine the appropriate strategy to participate in OSS and influence policy makers as they examine regulation that governs the intellectual property rights associated with OSS. We hope the work reported in this paper will help stimulate more research efforts in this growing area.

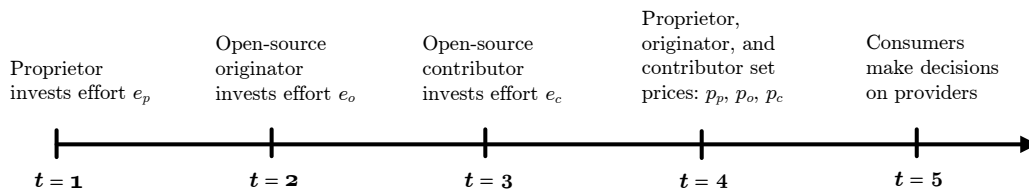


Figure 2.1: Sequence of events

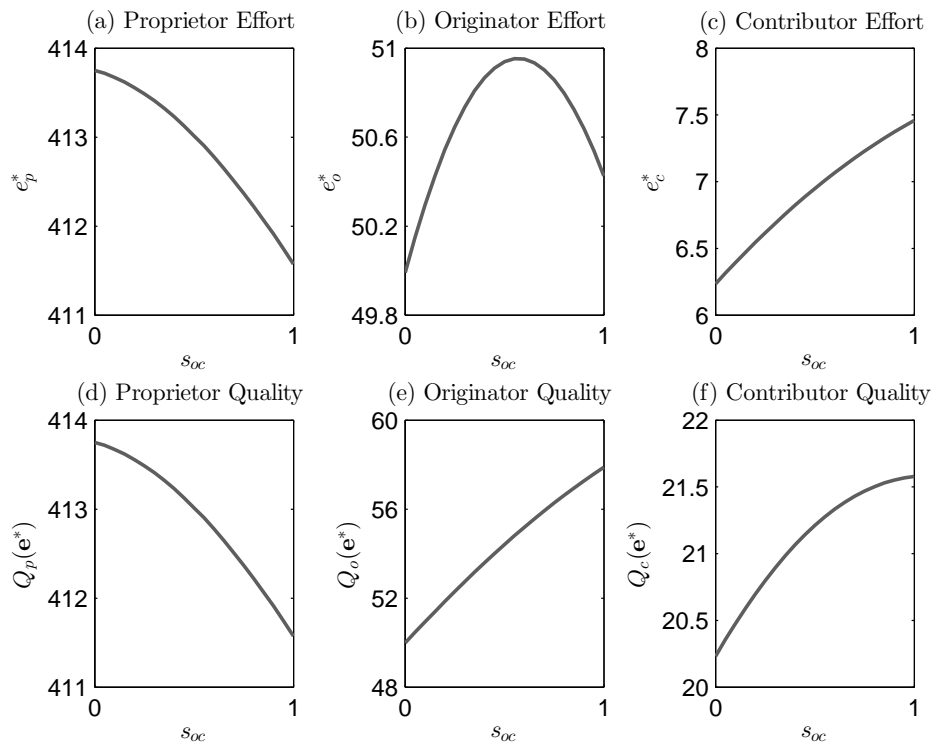


Figure 2.2: Equilibrium effort choices and resulting quality levels under Region H as affected by license restrictiveness. The parameter values are: $b = 0.1, s_p = 1, s_o = 1, s_c = 1, s_{co} = 0.28, c = 0.001, k_p = 0.06, k_o = 0.009,$ and $\beta_c = 0.001$.

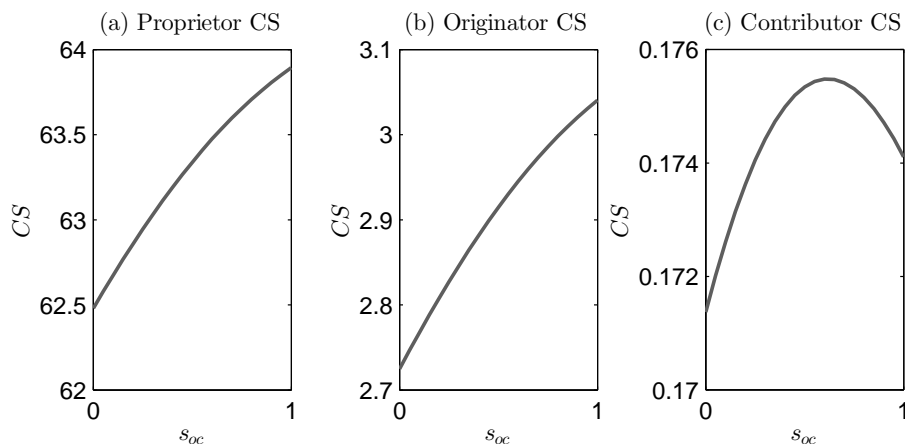


Figure 2.3: Consumer surplus associated with the providers' offerings under Region H. The parameter values are: $b = 0.1, s_p = 1, s_o = 1, s_c = 1, s_{co} = 0.28, c = 0.001, k_p = 0.06, k_o = 0.009,$ and $\beta_c = 0.001$.

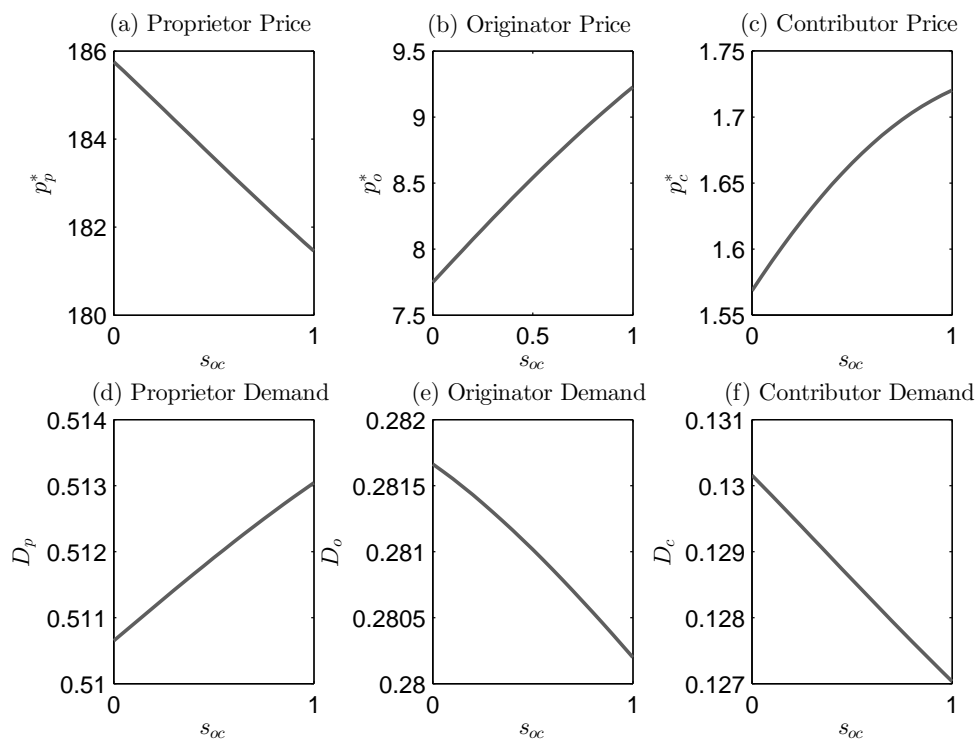


Figure 2.4: How the equilibrium prices and demand for each of the providers' offerings are affected by the extent of license restrictiveness under Region H. The parameter values are: $b = 0.1, s_p = 1, s_o = 1, s_c = 1, s_{co} = 0.28, c = 0.001, k_p = 0.06, k_o = 0.009,$ and $\beta_c = 0.001$.

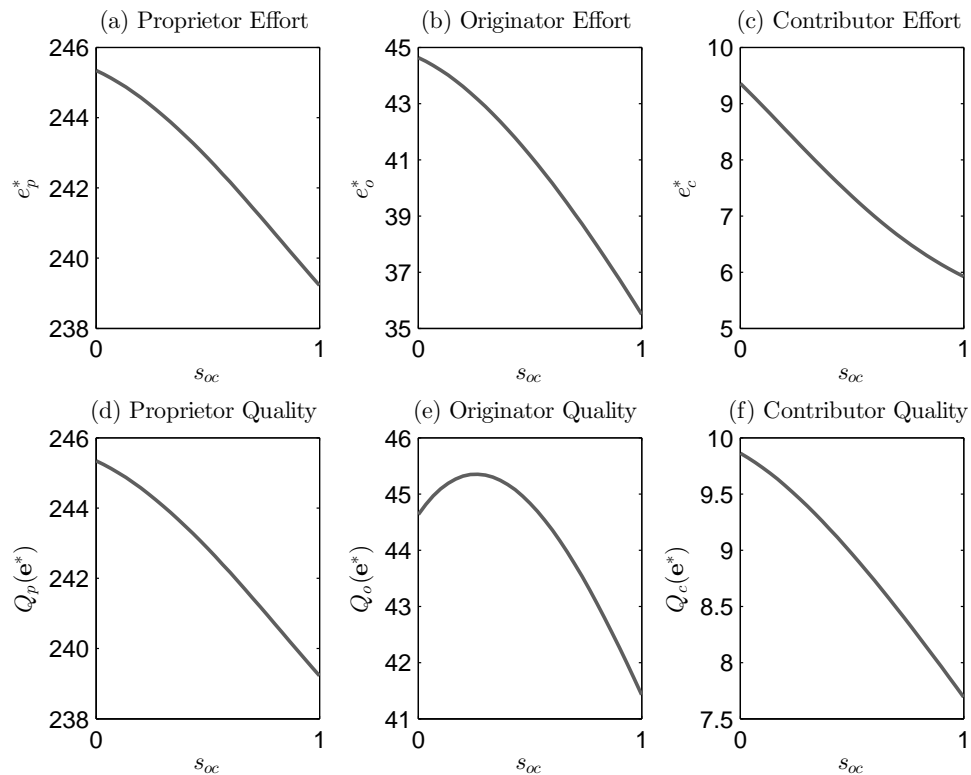


Figure 2.5: Equilibrium effort choices and resulting quality levels under Region L as affected by license restrictiveness. The parameter values are: $b = 0.1, s_p = 1, s_o = 1, \kappa_c = 1, \kappa_{co} = 2, \kappa = 0.01, \hat{k}_p = 0.01, \hat{k}_o = 0.01,$ and $\hat{k}_c = 0.001$.

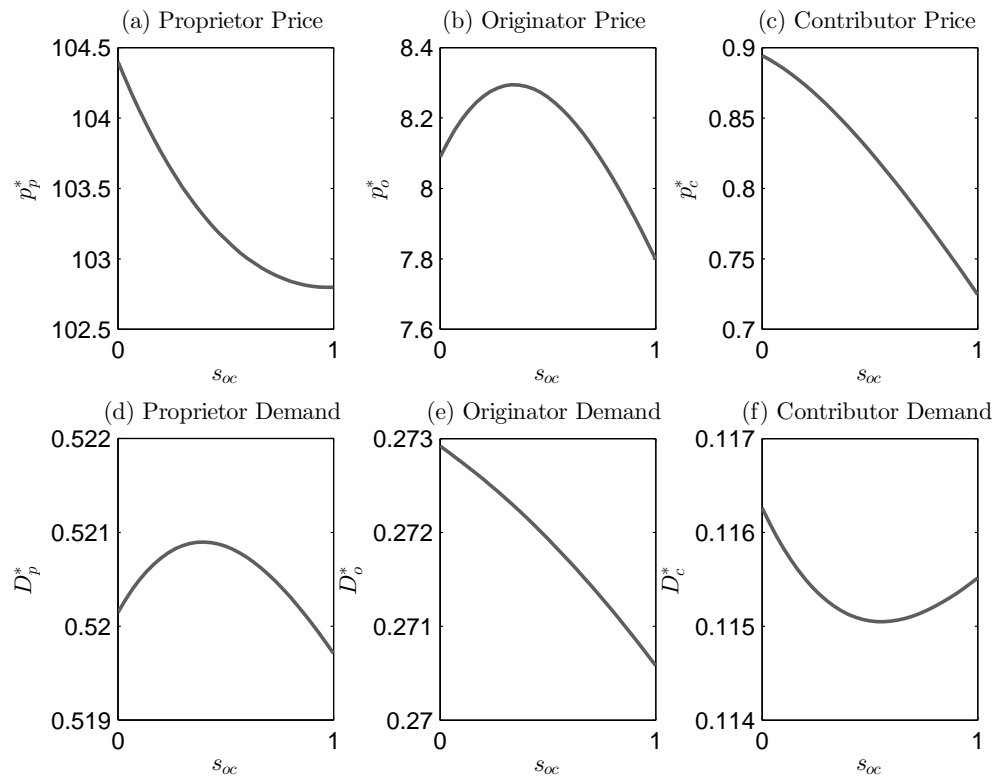


Figure 2.6: How the equilibrium prices and demand for each of the providers' offerings are affected by the extent of license restrictiveness under Region L. The parameter values are: $b = 0.1$, $s_p = 1$, $s_o = 1$, $\kappa_c = 1$, $\kappa_{co} = 2$, $\kappa = 0.01$, $\hat{k}_p = 0.01$, $\hat{k}_o = 0.01$, and $\hat{k}_c = 0.001$.

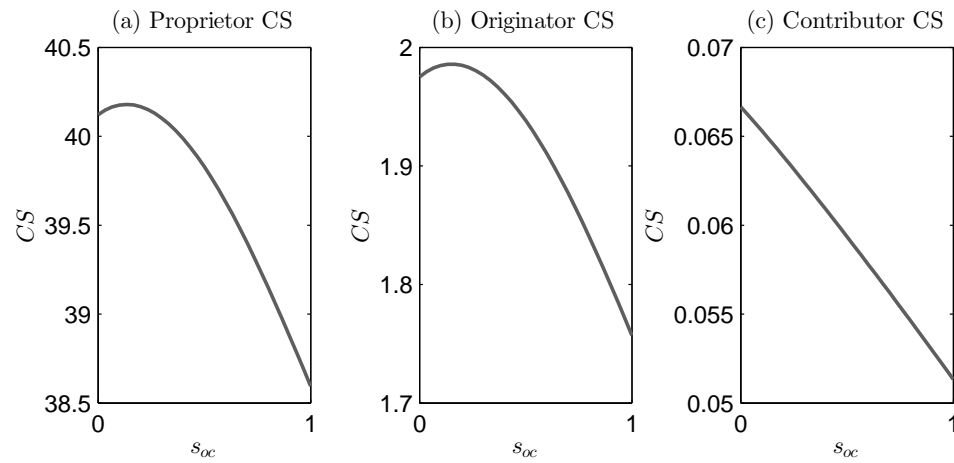


Figure 2.7: Consumer surplus associated with the providers' offerings under Region L. The parameter values are: $b = 0.1, s_p = 1, s_o = 1, \kappa_c = 1, \kappa_{co} = 2, \kappa = 0.01, \hat{k}_p = 0.01, \hat{k}_o = 0.01, \hat{k}_c = 0.001$.

Appendix 2.1

Supplementary Materials

Lemma A.1 *For fixed prices p_i , p_j , p_k , and qualities $Q_i > Q_j > Q_k$, the consumer market has the following characterization of regions:*

Region I: If $p_i \geq Q_i$, $p_j \geq Q_j$, and $p_k \geq Q_k$, then no consumer uses the software;

Region II: If $p_i < Q_i$, $p_j \geq \frac{p_i Q_j}{Q_i}$, and $p_k \geq \frac{p_i Q_k}{Q_i}$, then only firm i is active in the market, and

(a) consumers with $\theta \in [0, \theta_i)$ do not use the software, and

(b) consumers with $\theta \in [\theta_i, 1]$ purchase from firm i ;

Region III: If $p_i < Q_i$, $p_i - Q_i + Q_j < p_j < \frac{p_i Q_j}{Q_i}$, and $p_k \geq \frac{p_j Q_k}{Q_j}$, then only firms i and j are active in the market, and

(a) consumers with $\theta \in [0, \theta_j)$ do not use the software,

(b) consumers with $\theta \in [\theta_j, \theta_{ij})$ purchase from firm j , and

(c) consumers with $\theta \in [\theta_{ij}, 1]$ purchase from firm i ;

Region IV: If either (i) $p_i < Q_i$, $p_j \geq \frac{p_i Q_j}{Q_i}$, and $p_i - Q_i + Q_k < p_k < \frac{p_i Q_k}{Q_i}$, or (ii) $p_i < Q_i$, $p_i - Q_i + Q_j < p_j < \frac{p_i Q_j}{Q_i}$, and $p_i - Q_i + Q_k < p_k \leq p_j - \frac{(p_i - p_j)(Q_j - Q_k)}{Q_i - Q_j}$, then only firms i and k are active in the market, and

(a) consumers with $\theta \in [0, \theta_k)$ do not use the software,

(b) consumers with $\theta \in [\theta_k, \theta_{ik})$ purchase from firm k , and

(c) consumers with $\theta \in [\theta_{ik}, 1]$ purchase from firm i ;

Region V: If $p_i < Q_i$, $p_i - Q_i + Q_j < p_j < \frac{p_i Q_j}{Q_i}$, and $p_j - \frac{(p_i - p_j)(Q_j - Q_k)}{Q_i - Q_j} < p_k <$

$\frac{p_j Q_k}{Q_j}$, then all three firms are active in the market, and

- (a) consumers with $\theta \in [0, \theta_k)$ do not use the software,
- (b) consumers with $\theta \in [\theta_k, \theta_{jk})$ purchase from firm k ,
- (c) consumers with $\theta \in [\theta_{jk}, \theta_{ij})$ purchase from firm j , and
- (d) consumers with $\theta \in [\theta_{ij}, 1]$ purchase from firm i ;

Region VI: If either (i) $p_i \geq Q_i$, $p_j < Q_j$, and $p_k \geq \frac{p_j Q_k}{Q_j}$, or (ii) $p_i < Q_i$, $p_j \leq p_i - Q_i + Q_j$, and $p_k \geq \frac{p_j Q_k}{Q_j}$, then only firm j is active in the market, and

- (a) consumers with $\theta \in [0, \theta_j)$ do not use the software, and
- (b) consumers with $\theta \in [\theta_j, 1]$ purchase from firm j ;

Region VII: If one of the following holds: (i) $p_i \geq Q_i$, $p_j \geq Q_j$, and $p_k < Q_k$; (ii) $p_i \geq Q_i$, $p_j < Q_j$, and $p_k \leq p_j - Q_j + Q_k$; (iii) $p_i < Q_i$, $p_j \geq \frac{p_i Q_j}{Q_i}$, and $p_k \leq p_i - Q_i + Q_k$; (iv) $p_i < Q_i$, $p_i - Q_i + Q_j < p_j < \frac{p_i Q_j}{Q_i}$, and $p_k \leq p_i - Q_i + Q_k$; (v) $p_i < Q_i$, $p_j \leq p_i - Q_i + Q_j$, and $p_k \leq p_j - Q_j + Q_k$, then only firm k is active in the market, and

- (a) consumers with $\theta \in [0, \theta_k)$ do not use the software, and
- (b) consumers with $\theta \in [\theta_k, 1]$ purchase from firm k ;

Region VIII: If either (i) $p_i \geq Q_i$, $p_j < Q_j$, and $p_j - Q_j + Q_i < p_k < \frac{p_j Q_k}{Q_j}$, or (ii) $p_i < Q_i$, $p_j \leq p_i - Q_i + Q_j$, and $p_j - Q_j + Q_k < p_k < \frac{p_j Q_k}{Q_j}$, then only firms j and k are active in the market, and

- (a) consumers with $\theta \in [0, \theta_k)$ do not use the software,
- (b) consumers with $\theta \in [\theta_k, \theta_{jk})$ purchase from firm k , and
- (c) consumers with $\theta \in [\theta_{jk}, 1]$ purchase from firm j ;

where $\theta_i = p_i / Q_i$, $\theta_j = p_j / Q_j$, $\theta_k = p_k / Q_k$, $\theta_{ij} = (p_i - p_j) / (Q_i - Q_j)$, $\theta_{ik} = (p_i -$

$p_k)/(Q_i - Q_k)$, and $\theta_{jk} = (p_j - p_k)/(Q_j - Q_k)$.

Proof. A consumer with type θ : (a) prefers to purchase from firm i rather than to not use the software if and only if $\theta Q_i - p_i \geq 0$, i.e., $\theta \geq \theta_i$; (b) prefers to purchase from firm j rather than to not use the software if and only if $\theta Q_j - p_j \geq 0$, i.e., $\theta \geq \theta_j$; (c) prefers to purchase from firm k rather than to not use the software if and only if $\theta Q_k - p_k \geq 0$, i.e., $\theta \geq \theta_k$; (d) prefers to purchase from firm i rather than firm j if and only if $\theta Q_i - p_i \geq \theta Q_j - p_j$, i.e., $\theta \geq \theta_{ij}$; (e) prefers to purchase from firm i rather than firm k if and only if $\theta Q_i - p_i \geq \theta Q_k - p_k$, i.e., $\theta \geq \theta_{ik}$; (f) prefers to purchase from firm j rather than firm k if and only if $\theta Q_j - p_j \geq \theta Q_k - p_k$, i.e., $\theta \geq \theta_{jk}$.

By (a), (d), and (e) above and the definition of Θ , $\sigma(\theta) = i$ if and only if

$$\theta \geq t_A \triangleq \min(\max(\theta_i, \theta_{ij}, \theta_{ik}), 1). \quad (\text{A.1})$$

Similarly, $\sigma(\theta) = j$ if and only if

$$t_B \triangleq \max(\theta_j, \theta_{jk}) \leq \theta < t_C \triangleq \min(\theta_{ij}, 1), \quad (\text{A.2})$$

and $\sigma(\theta) = k$ if and only if

$$\theta_k \leq \theta < t_D \triangleq \min(\theta_{ik}, \theta_{jk}, 1). \quad (\text{A.3})$$

Finally, $\sigma(\theta) = \emptyset$ if and only if

$$0 \leq \theta < t_E \triangleq \min(\theta_i, \theta_j, \theta_k, 1). \quad (\text{A.4})$$

To see Region V, first define

$$t_V \triangleq p_j - \frac{(p_i - p_j)(Q_j - Q_k)}{Q_i - Q_j}. \quad (\text{A.5})$$

By (A.1), $t_A = \theta_{ij} < 1$ because $t_V < p_k$ implies $\theta_{ik} < \theta_{ij}$, $p_i - Q_i + Q_j < p_j < p_i Q_j / Q_i$ implies $\theta_i < \theta_{ij}$, and $p_i - Q_i + Q_j < p_j$ implies $\theta_{ij} < 1$. Hence, $\sigma = i$ for $\theta \geq \theta_{ij}$. Further, because $p_k < p_j Q_k / Q_j$ implies $\theta_{jk} > \theta_j$, we obtain that $t_B = \theta_{jk} < t_C = t_A = \theta_{ij}$. Thus, $\sigma = j$ for $\theta \in [\theta_{jk}, \theta_{ij})$. Because $t_V < p_k$ implies $\theta_{ik} > \theta_{jk}$, we obtain that $t_D = t_B = \theta_{jk}$, and $\sigma = k$ for $\theta \in [\theta_k, \theta_{jk})$.

Finally, $t_E = \theta_k$ because $p_i < Q_i$, $p_j < p_i Q_j / Q_i$, and $p_k < p_j Q_k / Q_j$. Therefore, $\sigma = \emptyset$ for $t \in [0, \theta_k)$, which finishes the characterization presented in Region V. The proofs of the remaining regions follow closely with that of Region V. ■

Lemma 2 (Generalized Statement) *If $Q_i > Q_j > Q_k > c$, there exist threshold values*

$0 < \tau_A < \tau_B < \tau_C < \tau_D$ such that

Region (i): If $c < \tau_A \triangleq \frac{Q_k(Q_i - Q_j)}{4Q_i - Q_j - 3Q_k}$, then

$$p_i^* = \frac{(Q_i - Q_j)(Q_i(4Q_j - Q_k) - 3Q_j Q_k) + c(Q_i(7Q_j - Q_k) - Q_j(Q_j + 5Q_k))}{2(Q_i(4Q_j - Q_k) - Q_j(Q_j + 2Q_k))}, \quad (\text{A.6})$$

$$p_j^* = \frac{(Q_i - Q_j) Q_j (Q_j - Q_k) + 3c Q_j (Q_i - Q_k)}{Q_i (4Q_j - Q_k) - Q_j (Q_j + 2Q_k)}, \quad (\text{A.7})$$

and

$$p_k^* = \frac{(Q_i - Q_j) (Q_j - Q_k) Q_k + c (4Q_i Q_j - Q_j^2 + 2Q_i Q_k - 2Q_j Q_k - 3Q_k^2)}{2(Q_i (4Q_j - Q_k) - Q_j (Q_j + 2Q_k))}. \quad (\text{A.8})$$

Region (ii): If $\tau_A \leq c < \tau_B \triangleq \frac{Q_k Q_j Q_i - Q_k Q_j^2}{4Q_j Q_i - (Q_k Q_j + Q_j^2 + 2Q_k Q_i)}$, then

$$p_i^* = \frac{cQ_k + cQ_j - Q_k Q_j + Q_k Q_i}{2Q_k}, \quad p_j^* = \frac{Q_j c}{Q_k}, \quad p_k^* = c. \quad (\text{A.9})$$

Region (iii): If $\tau_B \leq c < \tau_C \triangleq \frac{Q_j}{2}$, then

$$p_i^* = \frac{(2Q_i + 3c - 2Q_j)Q_i}{4Q_i - Q_j}, \quad p_j^* = \frac{Q_j(Q_i - Q_j) + c(Q_j + 2Q_i)}{4Q_i - Q_j}, \quad p_k^* = c. \quad (\text{A.10})$$

Region (iv): If $\tau_C \leq c < \tau_D \triangleq \frac{Q_j Q_i}{2Q_i - Q_j}$, then

$$p_i^* = \frac{Q_i}{Q_j} c, \quad p_j^* = c, \quad p_k^* = c. \quad (\text{A.11})$$

Region (v): If $c \geq \tau_D$, then

$$p_i^* = \frac{Q_i + c}{2}, \quad p_j^* = c, \quad p_k^* = c. \quad (\text{A.12})$$

Proof of Lemma 2: Given $Q_i > Q_j > Q_c > c$, it is easy to verify that τ_A, τ_B, τ_C , and τ_D satisfy $0 < \tau_A < \tau_B < \tau_C < \tau_D$. First, we examine Region (i) where $c < \tau_A$ is satisfied. Further, suppose Region I of Lemma A.1 is satisfied in equilibrium. Then, $\tilde{\Pi}_i = 0$. However, firm i can deviate to Region II by setting a price $c < p < Q_i$ and obtain $\tilde{\Pi}_i(p | p_j, p_k) > 0$. Hence, Region I cannot occur in equilibrium. Next, suppose Region II of Lemma A.1 is satisfied in equilibrium. Then, by (2.4), (2.5), and (2.6), $\tilde{\Pi}_j = 0$. However, firm j can deviate to Region III by setting a price $c < p < p_i Q_j / Q_i$, which is possible because $c < \tau_A$ implies $p_i Q_j / Q_i > c$. In that case, $\tilde{\Pi}_j(p | p_i, p_k) > 0$. Thus, Region II cannot occur in equilibrium. Following similar steps, we can rule out Region III, IV, VI, VII, and VIII in Lemma A.1. Therefore, we can focus on Region V for candidate equilibria. By (2.4), (2.5), (2.6), and Lemma A.1, we obtain

$$\tilde{\Pi}_i(p_i | p_j, p_k) = (p_i - c) \left(1 - \frac{p_i - p_j}{Q_i - Q_j} \right), \quad (\text{A.13})$$

$$\tilde{\Pi}_j(p_j | p_i, p_k) = (p_j - c) \left(\frac{p_i - p_j}{Q_i - Q_j} - \frac{p_j - p_k}{Q_j - Q_k} \right), \quad (\text{A.14})$$

and

$$\tilde{\Pi}_k(p_k | p_i, p_j) = (p_k - c) \left(\frac{p_j - p_k}{Q_j - Q_k} - \frac{p_k}{Q_k} \right). \quad (\text{A.15})$$

Because $Q_i > Q_j > Q_k$, by (A.13), (A.14), and (A.15), all three residual profit functions are strictly concave, with unconstrained maximizers characterized by

$$p_i = \frac{p_j + c + Q_i - Q_j}{2}, \quad (\text{A.16})$$

$$p_j = \frac{(c + p_k)Q_i + (p_i - p_k)Q_j - (c + p_i)Q_k}{2(Q_i - Q_k)}, \quad (\text{A.17})$$

and

$$p_k = \frac{cQ_j + p_jQ_k}{2Q_j}. \quad (\text{A.18})$$

Simultaneously solving (A.16), (A.17), and (A.18), we obtain the equilibrium prices in (A.6), (A.7), and (A.8). By (A.6), (A.7), (A.8), and $c < \tau_A$, it is straightforward to verify the conditions of Region V in Lemma A.1 are satisfied. Therefore, p_i^* , p_j^* , and p_k^* given in (A.6), (A.7), and (A.8) are the unique candidate equilibrium prices in Region V of Lemma A.1.

To ensure that no firm would deviate to another region, we fix p_j^* and p_k^* and consider the pricing of firm i . Suppose it sets $p_i \leq p_k^*Q_i/Q_k$, then Region II applies. Suppose firm i sets $p_k^*Q_i/Q_k < p_i \leq (p_k^*(-Q_i + Q_j) + p_j^*(Q_i - Q_k))/(Q_j - Q_k)$, then Region IV of Lemma A.1 applies. Suppose it sets $(p_k^*(-Q_i + Q_j) + p_j^*(Q_i - Q_k))/(Q_j - Q_k) < p_i \leq p_j^* - Q_j + Q_i$, then Region V applies. Finally, suppose $p_i > p_j^* - Q_j + Q_i$, then

Region VIII applies. In summary, the profit function of firm i is given by

$$\tilde{\Pi}_i(p_i | p_j^*, p_k^*) = \begin{cases} (p_i - c)(1 - \frac{p_i}{Q_i}) & \text{if } p_i \leq \frac{p_k^* Q_i}{Q_k} \text{ (Region II);} \\ (p_i - c)(1 - \frac{p_i - p_k^*}{Q_i - Q_k}) & \text{if } \frac{p_k^* Q_i}{Q_k} < p_i \leq \frac{p_k^* (-Q_i + Q_j) + p_j^* (Q_i - Q_k)}{Q_j - Q_k} \\ & \text{(Region IV);} \\ (p_i - c)(1 - \frac{p_i - p_j^*}{Q_i - Q_j}) & \text{if } \frac{p_k^* (-Q_i + Q_j) + p_j^* (Q_i - Q_k)}{Q_j - Q_k} < p_i \leq p_j^* - Q_j + Q_i \\ & \text{(Region V);} \\ 0 & \text{if } p_i > p_j^* - Q_j + Q_i \text{ (Region VIII).} \end{cases}$$

(A.19)

By (A.19), $\tilde{\Pi}_i(\cdot | p_j^*, p_k^*)$ is continuous. Further, because $(Q_i + c)/2 \geq p_k^* Q_i / Q_k$ under $c < \tau_A$, $\tilde{\Pi}_i(\cdot | p_j^*, p_k^*)$ increases in Region II. Also, $\tilde{\Pi}_i(\cdot | p_j^*, p_k^*)$ is increasing in Region IV if and only if $p_i \leq (c + p_k^* + Q_i - Q_k)/2$, which is satisfied because $Q_i > Q_j > Q_k > c$ and $c < \tau_A$ imply $(p_k^* (-Q_i + Q_j) + p_j^* (Q_i - Q_k)) / (Q_j - Q_k) \leq (c + p_k^* + Q_i - Q_k) / 2$. Therefore, p_i^* given in (A.6) is the unique price that maximizes (A.19).

Similarly, fixing prices of firms i and k , we examine the price setting problems

of firm j . Its profit function is given by

$$\tilde{\Pi}_j(p_j | p_i^*, p_k^*) = \begin{cases} (p_j - c)(1 - \frac{p_j}{Q_j}) & \text{if } p_j \leq p_i^* - Q_i + Q_j \text{ (Region VI);} \\ (p_j - c)(\frac{p_i^* - p_j}{Q_i - Q_j} - \frac{p_j}{Q_j}) & \text{if } p_i^* - Q_i + Q_j < p_j \leq \frac{p_k^* Q_j}{Q_k} \\ & \text{(Region III);} \\ (p_j - c)(\frac{p_i^* - p_j}{Q_i - Q_j} - \frac{p_j - p_k^*}{Q_j - Q_k}) & \frac{p_k^* Q_j}{Q_k} < p_j \leq \frac{p_k^*(Q_i - Q_j) + p_i^*(Q_j - Q_k)}{Q_i - Q_k} \\ & \text{(Region V);} \\ 0 & \text{if } p_j > \frac{p_k^*(Q_i - Q_j) + p_i^*(Q_j - Q_k)}{Q_i - Q_k} \\ & \text{(Region IV).} \end{cases}$$

(A.20)

By (A.20), $\tilde{\Pi}_j(\cdot | p_i^*, p_k^*)$ is continuous. Further, it is increasing on $[0, p_i^* - Q_i + Q_j]$ if and only if $p_j \leq (c + Q_j)/2$, which is satisfied because $Q_i > Q_j > Q_k > c$ implies $p_i^* - Q_i + Q_j < (c + Q_j)/2$. Also, $\tilde{\Pi}_j(\cdot | p_i^*, p_k^*)$ is increasing on $(p_i^* - Q_i + Q_j, p_k^* Q_j / Q_k]$ if and only if $p_j \leq (c + p_i^* Q_j / Q_i) / 2$, which is satisfied since $p_k^* Q_j / Q_k \leq (c + p_i^* Q_j / Q_i) / 2$. Therefore, p_j given in (A.7) maximizes (A.20).

Finally, fixing p_i^* and p_j^* , we examine the price setting problem of firm k , whose profit function is given by

$$(A.21) \quad \tilde{\Pi}_k(p_k | p_i^*, p_j^*) = \begin{cases} (p_k - c)(1 - \frac{p_k}{Q_k}) & \text{if } p_k \leq p_i^* - Q_i + Q_k \text{ (Region VII);} \\ (p_k - c)(\frac{p_i^* - p_k}{Q_i - Q_k} - \frac{p_k}{Q_k}) & \text{if } p_i^* - Q_i + Q_k < p_k \leq \\ & \frac{p_j^*(Q_i - Q_k) + p_i^*(-Q_j + Q_k)}{Q_i - Q_j} \text{ (Region IV);} \\ (p_k - c)(\frac{p_j^* - p_k}{Q_i - Q_k} - \frac{p_k}{Q_k}) & \text{if } \frac{p_j^*(Q_i - Q_k) + p_i^*(-Q_j + Q_k)}{Q_i - Q_j} < p_k \leq \frac{p_j^* Q_k}{Q_j} \\ & \text{(Region V);} \\ 0 & \text{if } p_k > \frac{p_j^* Q_k}{Q_j} \text{ (Region III).} \end{cases}$$

By (A.21), $\tilde{\Pi}_k(\cdot | p_i^*, p_j^*)$ is continuous. Further, it is increasing on $[0, p_i^* - Q_i + Q_k]$ if and only if $p_k \leq (c + Q_k)/2$, which is satisfied because $Q_i > Q_j > Q_k > c$ implies $p_i^* - Q_i + Q_k < (c + Q_k)/2$. Also, $\tilde{\Pi}_k(\cdot | p_i^*, p_j^*)$ is increasing on $(p_i^* - Q_i + Q_k, (p_j^*(Q_i - Q_k) + p_i^*(-Q_j + Q_k))/(Q_i - Q_j)]$ if and only if $p_k \leq (c + p_i^* Q_k / Q_i)/2$, which is satisfied since $(p_j^*(Q_i - Q_k) + p_i^*(-Q_j + Q_k))/(Q_i - Q_j) \leq (c + p_i^* Q_k / Q_i)/2$. Therefore, p_k given in (A.8) maximizes (A.21). This completes the proof of Region (i). Regions (ii) through (v) follow a similar train of logic and are omitted for brevity. \square

Proof of Lemma 3: We define $z = 1/b$ and then examine e_p^* , e_o^* , and e_c^* as b becomes small. We can express $e_p^* \sim z^m$, $e_o^* \sim z^n$, and $e_c^* \sim z^q$ for some constants $m, n, q \in \mathbb{R}$. In other words, e_p^* is in the order of z^m , e_o^* is in the order of z^n , and e_c^* is in the order of z^q . Our goal is to determine the value of m, n , and q in equilibrium.

Suppose $m \leq \max(n, q)$ in equilibrium. By (2.1) and (2.2), $Q_o \sim z^{\max(n, q)}$ and

$Q_c \sim z^{\max(n,q)}$. First, suppose $n > q$. By Lemma 2, p_o is $O(z^n)$. Because $|\Theta| = 1$, $\tilde{\Pi}_o$ is $O(z^n)$. The costs are $O(z^{2n-1})$ by (2.11). To generate non-negative profit for the originator, $z^n \geq z^{2n-1}$ should be satisfied, which gives $n \leq 1$. On the other hand, suppose $n \leq q$. Similarly, $\tilde{\Pi}_c$ is $O(z^q)$ and costs are $O(z^{2q})$, which requires $q \geq 2q$, i.e., $q \leq 0$. Hence, $\max(n,q) \leq 1$. Because $m \leq \max(n,q)$, we obtain $m \leq 1$. Thus, Π_p is $O(z^1)$. However, the proprietor could instead set $m = \frac{3}{2}$ such that $\tilde{\Pi}_p \sim z^{\frac{3}{2}}$ and costs are $O(z^{2(\frac{3}{2})-2}) = O(z)$, and hence $\Pi_p \sim z^{\frac{3}{2}}$. Therefore, the proprietor will deviate, which means $m \leq \max(n,q)$ cannot happen in equilibrium. Therefore, $m > \max(n,q)$.

Suppose $\max(n,q) \leq 0$. Then, for the originator, Π_o is $O(1)$. However, suppose the originator instead selects $n = \frac{1}{2}$. Then $Q_o \sim z^{\frac{1}{2}}$ and $Q_c \sim z^{\frac{1}{2}}$. Because from the previous argument, $m \geq \frac{3}{2}$ and we obtain $\tau_A \sim z^{\frac{1}{2}}$, which implies Region (i) of Lemma 2 is satisfied. Thus, $p_o \sim p_c \sim z^{\frac{1}{2}}$, which implies $\tilde{\Pi}_o \sim z^{\frac{1}{2}}$, whereas costs are $O(z^{2(\frac{1}{2})-1}) = O(1)$, and hence $\Pi_o \sim z^{\frac{1}{2}}$ such that the originator will deviate. Therefore $\max(n,q) > 0$.

Suppose $q \geq n$. Because $m > \max(n,q) > 0$, $\tau_A > c$ as $b \rightarrow 0$. Thus, by Region (i) of Lemma 2, $\tilde{\Pi}_c \sim z^q$ and costs are $O(z^{2q})$, which requires $q \geq 2q$, i.e., $q \leq 0$. However, we have shown that $\max(n,q) > 0$, which is a contradiction. Therefore, $q < n$. Further, because $s_o > s_{c_o}$ in Region H, we have $Q_p > Q_o > Q_c$ in Region (i).

Substituting the equilibrium prices in Lemma 2 into (2.10), we obtain the contributor's profit as

$$\Pi_c(e_c|e_p, e_o) = \frac{(Q_o - Q_c) Q_o (c(Q_o - 4Q_p) + Q_c(3c - Q_o + Q_p))^2}{4Q_c(Q_o(Q_o - 4Q_p) + Q_c(2Q_o + Q_p))^2} - \frac{1}{2}\beta_c e_c^2. \quad (\text{A.22})$$

Substituting $Q_p = s_p e_p$, (2.1), and (2.2) into (A.22) and differentiating twice with respect to e_c , and then plugging in $e_p \sim z^m$ and $e_o \sim z^n$, we find that the second order condition is satisfied, i.e., $d^2\Pi_c/de_c^2 < 0$ for all e_c . Analyzing the first order condition and collecting terms in powers of z , we find that it can be written as

$$E_1 z^q + E_2 + Y_1(z) = 0, \quad (\text{A.23})$$

where $E_1, E_2 \in \mathbb{R}$ and $Y_1(z)$ is a polynomial in z with terms that will be dominated by E_2 in absolute value. As $z \rightarrow \infty$, equation (A.23) has to hold for all z values. This is not possible if $q > 0$ because in that case $E_1 z^q$ dominates all other terms in absolute value and explodes. Therefore, in equilibrium, $E_1 z^q$ and E_2 have to cancel each other, which implies $q=0$.

Next, for the originator's effort problem, we obtain the profit function of the originator by substituting the equilibrium prices in Region (i) of Lemma 2 into (2.11):

$$\Pi_o(e_o|e_p) = \frac{(Q_o - c)^2 (Q_o - Q_c) (Q_p - Q_c) (Q_p - Q_o)}{(Q_o (Q_o - 4Q_p) + Q_c (2Q_o + Q_p))^2} - \frac{1}{2} \beta_o e_o^2. \quad (\text{A.24})$$

Taking a total derivative of (A.24) with respect to e_o (and another one to show concavity as before), and substituting $e_p \sim z^m$, $e_o \sim z^n$ and $e_c \sim z^0$ into the first order condition, we obtain

$$E_3 z^{n-1} + E_4 + Y_2(z) = 0, \quad (\text{A.25})$$

where $E_3, E_4 \in \mathbb{R}$ and $Y_2(z)$ is a polynomial in z with terms that will be dominated by

E_4 in absolute value. From a similar argument as in the contributor's problem, we can obtain $n = 1$.

For the proprietor's effort problem, the profit function of the proprietor is

$$\Pi_p(e_p) = \frac{(Q_p - Q_o)(Q_o(c - 4Q_p) + Q_k(-c + 3Q_o + Q_p))^2}{4(Q_o(Q_o - 4Q_p) + Q_k(2Q_o + Q_p))^2} - \frac{1}{2}\beta_p e_p^2. \quad (\text{A.26})$$

Similarly, we can obtain the first order condition for the proprietor:

$$E_5 z^{m-2} + E_6 + Y_3(z) = 0, \quad (\text{A.27})$$

where $E_5, E_6 \in \mathbb{R}$ and $Y_3(z)$ is a polynomial in z with terms that will be dominated by E_6 in absolute value. Similarly, we establish concavity and demonstrate that, by (A.27), $m = 2$.

Based on the values $m = 2$, $n = 1$, and $q = 0$, substituting $e_p = P_1 z^2 + O(z)$, $e_o = O_1 z + O(1)$, and $e_c = C_1 + O(1/z)$ into the three first order conditions and equating the lead coefficients of the highest order terms with respect to z to zero, we obtain P_1 , O_1 , and C_1 from the following equations:

$$s_c(7s_{co} - 4s_o)s_o^2 - s_{co}^2(s_{co} + 2s_o)s_{oc} - 4C_1(s_{co} - 4s_o)^3\beta_c = 0, \quad (\text{A.28})$$

$$k_o O_1(s_{co} - 4s_o)^2 + (s_{co} - s_o)s_o^2 = 0, \quad (\text{A.29})$$

and

$$4k_p P_1 - s_p = 0. \quad (\text{A.30})$$

Repeating similar steps as above, we obtain the optimal effort investment levels to the order of b^2 as below:

$$e_p = \frac{s_p}{4k_p b^2} - \frac{4k_p K_1^3 s_o^5 (2s_{co}^2 + 13s_{co}s_o + 3s_o^2)}{k_o^2 K_2^6 s_p^3} - \frac{2k_p K_1 s_o^2 K_3 b}{\beta_c k_o^3 K_2^9 s_p^5} + O(b^2), \quad (\text{A.31})$$

$$e_o = \frac{s_o^2 K_1}{k_o K_2^2 b} - \frac{8k_p K_1^3 s_o^4 (s_{co} + 2s_o)}{k_o^2 K_2^5 s_p^2} - \frac{K_4 b}{16\beta_c^2 k_o^3 K_2^7 K_1^2 s_o^4 s_p^4} + O(b^2), \quad (\text{A.32})$$

and

$$e_c = \frac{s_c s_o^2 (4s_o - 7s_{co}) + s_{co}^2 (s_{co} + 2s_o) s_{oc}}{4\beta_c K_2^3} - \frac{K_5 b}{8k_o K_1 K_2^6 s_o^2 s_p^2 \beta_c^2} + O(b^2), \quad (\text{A.33})$$

where $K_1 = s_o - s_{co}$, $K_2 = 4s_o - s_{co}$, $K_3 = 16k_p K_1^3 s_o^5 (4s_{co}^4 + 55s_{co}^3 s_o + 58s_{co}^2 s_o^2 - 66s_{co}s_o^3 + 30s_o^4) \beta_c + k_o^2 K_2 s_p^2 (3K_2 s_c^2 s_o^4 (6s_{co}^2 + 41s_{co}s_o - 20s_o^2) + s_{oc} (s_c s_o^2 (-19s_{co}^5 - 258s_{co}^4 s_o - 11s_{co}^3 s_o^2 + 94s_{co}^2 s_o^3 - 468s_{co}s_o^4 + 176s_o^5) + s_{co} (s_{co}^6 + 30s_{co}^5 s_o + 32s_{co}^4 s_o^2 - 40s_{co}^3 s_o^3 + 204s_{co}^2 s_o^4 - 176s_{co}s_o^5 + 192s_o^6) s_{oc}) - 4c K_1 K_2^4 (s_{co}^2 + 6s_{co}s_o + 2s_o^2) \beta_c)$,

$K_4 = 256k_p^2 K_1^6 s_o^{10} (s_{co}^2 + 7s_{co}s_o + s_o^2) \beta_c^2 - 32k_o^2 k_p K_1^3 s_o^5 s_p^2 \beta_c (9s_c^2 s_o^4 (-8s_{co}^2 - 5s_{co}s_o + 4s_o^2) + s_c s_o^2 (19s_{co}^4 + 104s_{co}^3 s_o - 129s_{co}^2 s_o^2 + 248s_{co}s_o^3 - 80s_o^4) s_{oc} - s_{co} (s_{co}^5 + 17s_{co}^4 s_o - 15s_{co}^3 s_o^2 + 50s_{co}^2 s_o^3 - 20s_{co}s_o^4 + 48s_o^5) s_{oc}^2 + 4c K_1 K_2^4 (s_{co} + 2s_o) \beta_c) + k_o^4 K_2 s_p^4 (-3s_{co}^4 (7s_{co} - 4s_o) s_o^6 (7s_{co}^2 + 25s_{co}s_o + 4s_o^2) + 4s_c^3 s_o^4 (s_{co} + 2s_o) (7s_{co}^4 + 149s_{co}^3 s_o - 72s_{co}^2 s_o^2 + 56s_{co}s_o^3 - 32s_o^4) s_{oc} - s_{co}^4 (s_{co} + 2s_o) (s_{co}^4 + 35s_{co}^3 s_o + 8s_{co}s_o^3 + 64s_o^4) s_{oc}^4 - 8c K_2^4 s_{co} (s_{co} + 2s_o) (s_{co}^3 +$

$$\begin{aligned}
& 10s_{co}^2s_o - 12s_{co}s_o^2 + 16s_o^3)s_{oc}^2\beta_c + 16c^2K_1K_2^8\beta_c^2 + s_c^2s_o^2(-s_{co}(s_{co}^6 + 93s_{co}^5s_o + 933s_{co}^4s_o^2 + \\
& 493s_{co}^3s_o^3 + 24s_{co}^2s_o^4 + 912s_{co}s_o^5 - 512s_o^6)s_{oc}^2 - 8cK_2^4s_o(s_{co} + 2s_o)(11s_{co} + 4s_o)\beta_c) \\
& + 2s_{co}s_o s_{oc}(s_{co}^2(s_{co}^6 + 51s_{co}^5s_o + 249s_{co}^4s_o^2 + 19s_{co}^3s_o^3 + 120s_{co}^2s_o^4 + 336s_{co}s_o^5 - 128s_o^6)s_{oc}^2 + \\
& 4cK_2^4(s_{co}^4 + 23s_{co}^3s_o + 34s_{co}^2s_o^2 + 32s_o^4)\beta_c), \text{ and}
\end{aligned}$$

$$\begin{aligned}
& K_5 = 96k_pK_1^3s_o^5(s_{co}s_o^2(-5s_{co} + 2s_o) + s_{co}(s_{co}^2 + 2s_o^2)s_{oc})\beta_c + k_o^2K_2(s_{co}s_o - s_{co}s_{oc})s_p^2 \\
& ((7s_{co} + 8s_o)(s_{co}s_o - s_{co}s_{oc})(s_{co}s_o^2(-7s_{co} + 4s_o) + s_{co}^2(s_{co} + 2s_o)s_{oc}) - 16cK_2^4(s_{co} + 2s_o) \\
& \beta_c). \square
\end{aligned}$$

Proof of Proposition 1: Technically, we show that under the conditions of Region H

$$(i) \frac{de_p^*}{ds_{oc}} < 0, \text{ and } \frac{de_c^*}{ds_{oc}} > 0;$$

(ii) There exists $\bar{\beta}_c > 0$ such that if $\beta_c \geq \bar{\beta}_c$, then $\frac{de_o^*}{ds_{oc}} < 0$ for all $s_{oc} \in [0, \frac{s_c s_o}{s_{co}}]$; if $\beta_c < \bar{\beta}_c$, then there exists $\bar{s}_{oc} \in (0, \frac{s_c s_o}{s_{co}})$ such $\frac{de_o^*}{ds_{oc}} \geq 0$ for $s_{oc} \in [0, \bar{s}_{oc}]$, and $\frac{de_o^*}{ds_{oc}} < 0$ for $s_{oc} \in (\bar{s}_{oc}, \frac{s_c s_o}{s_{co}}]$;

$$(iii) \frac{dQ_p(\mathbf{e}^*)}{ds_{oc}} < 0, \frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} > 0, \text{ and } \frac{dQ_c(\mathbf{e}^*)}{ds_{oc}} > 0,$$

$$\text{where } \bar{\beta}_c = \frac{k_o^2 K_2 s_c^2 (s_{co} + 2s_o) (-7s_{co}^4 - 149s_{co}^3 s_o + 72s_{co}^2 s_o^2 - 56s_{co} s_o^3 + 32s_o^4) s_p^2}{8k_p K_1^3 s_o^3 (-19s_{co}^4 - 104s_{co}^3 s_o + 129s_{co}^2 s_o^2 - 248s_{co} s_o^3 + 80s_o^4)}.$$

First, differentiating (A.33) with respect to s_{oc} , we obtain

$$\frac{de_c^*}{ds_{oc}} = \frac{s_{co}^2 (s_{co} + 2s_o)}{4K_2^3 \beta_c} + O(b), \quad (A.34)$$

which is positive because $s_o > s_{co}$ in Region H implies $K_2 = 4s_o - s_{co} > 0$.

By differentiating (A.31) with respect to s_{oc} ,

$$\frac{de_p^*}{ds_{oc}} = -\frac{2k_p K_1 s_o^2 (s_c s_o^2 A_1 + 2s_{co} A_2) b}{k_o K_2^8 s_p^3 \beta_c} + O(b^2), \quad (\text{A.35})$$

where $A_1 = -19s_{co}^5 - 258s_{co}^4 s_o - 11s_{co}^3 s_o^2 + 94s_{co}^2 s_o^3 - 468s_{co} s_o^4 + 176s_o^5$ and $A_2 = s_{co}^6 + 30s_{co}^5 s_o + 32s_{co}^4 s_o^2 - 40s_{co}^3 s_o^3 + 204s_{co}^2 s_o^4 - 176s_{co} s_o^5 + 192s_o^6$. Because $s_o > s_{co}$ in Region H, it is easy to see that $K_1 = s_o - s_{co} > 0$ and $A_2 > 0$. Let $r = s_{co}/s_o < 1$, we can rewrite $A_1 = f(r)s_o^5$, where $f(r) = 176 - 468r + 94r^2 - 11r^3 - 258r^4 - 19r^5$. By Sturm's Theorem, there can only be one root for $f(r) = 0$ for $r \in (0, 1]$. Because $f(0) = 176 > 0$, $f(1) = -486$, and $f(r)$ is continuous, there exists a unique root $\bar{r} \in (0, 1]$ such that $f(r) > 0$ for $r \in (0, \bar{r})$. Hence, $A_1 > 0$ for $r \in (0, \bar{r})$. Therefore, $de_p^*/ds_{oc} < 0$. This completes the proof of part (i). Because s_p is positive, this also implies that $dQ_p(\mathbf{e}^*)/ds_{oc} < 0$.

Differentiating (A.32) with respect to s_{oc} , we obtain

$$\frac{de_o^*}{ds_{oc}} = \frac{(A_3 s_{oc}^3 + A_4 s_{oc}^2 + A_5 s_{oc} + A_6) b}{8k_o K_1^2 K_2^7 s_o^4 s_p^2 \beta_c^2} + O(b^2), \quad (\text{A.36})$$

where $A_3 = 2k_o^2 K_2 s_{co}^4 (s_{co} + 2s_o) (s_{co}^4 + 35s_{co}^3 s_o + 8s_{co} s_o^3 + 64s_o^4) s_p^2$,

$$A_4 = -3k_o^2 K_2 s_c s_{co}^2 s_o (s_{co}^6 + 51s_{co}^5 s_o + 249s_{co}^4 s_o^2 + 19s_{co}^3 s_o^3 + 120s_{co}^2 s_o^4 + 336s_{co} s_o^5 - 128s_o^6) s_p^2,$$

$$A_5 = s_{co} s_o^2 (k_o^2 K_2 s_c^2 (s_{co}^6 + 93s_{co}^5 s_o + 933s_{co}^4 s_o^2 + 493s_{co}^3 s_o^3 + 24s_{co}^2 s_o^4 + 912s_{co} s_o^5 - 512s_o^6) s_p^2 - 32k_p K_1^3 s_o^3 (s_{co}^5 + 17s_{co}^4 s_o - 15s_{co}^3 s_o^2 + 50s_{co}^2 s_o^3 - 20s_{co} s_o^4 + 48s_o^5) \beta_c), \text{ and}$$

$$A_6 = 2k_o^2 K_2 s_c^3 s_o^4 (s_{co} + 2s_o) (-7s_{co}^4 - 149s_{co}^3 s_o + 72s_{co}^2 s_o^2 - 56s_{co} s_o^3 + 32s_o^4) s_p^2$$

$$+ 16k_p K_1^3 s_c s_o^7 (19s_{co}^4 + 104s_{co}^3 s_o - 129s_{co}^2 s_o^2 + 248s_{co} s_o^3 - 80s_o^4) \beta_c.$$

Because the denominator of the first term is positive, we only need to examine the sign of the numerator. Let $g(s_{oc}) = A_3 s_{oc}^3 + A_4 s_{oc}^2 + A_5 s_{oc} + A_6$. Then, $g(s_{oc})$ is a third degree polynomial of s_{oc} and has at most three real roots. It is easy to see that $A_3 > 0$. We can obtain that $g(s_c s_o / s_{co}) = -16k_p K_1^4 K_2 s_c s_o^6 (2s_{co}^3 + 25s_{co}^2 s_o - 17s_{co} s_o^2 + 44s_o^3) \beta_c < 0$.

Next, let

$$\underline{s}_{oc} = -\frac{s_c s_o^2 (-19s_{co}^4 - 104s_{co}^3 s_o + 129s_{co}^2 s_o^2 - 248s_{co} s_o^3 + 80s_o^4)}{2s_{co} (s_{co}^5 + 17s_{co}^4 s_o - 15s_{co}^3 s_o^2 + 50s_{co}^2 s_o^3 - 20s_{co} s_o^4 + 48s_o^5)}.$$

We can obtain that $\underline{s}_{oc} < 0$ and $g(\underline{s}_{oc}) > 0$. Therefore, $g = 0$ has one root in $(\frac{s_c s_o}{s_{co}}, \infty)$, another in $(-\infty, \underline{s}_{oc})$, and the third in $(\underline{s}_{oc}, \frac{s_c s_o}{s_{co}})$. We are interested in the third one which falls into our definition of Region H. Let \bar{s}_{oc} be the root in $(\underline{s}_{oc}, \frac{s_c s_o}{s_{co}})$. If $g(0) > 0$, then $\bar{s}_{oc} \in (0, \frac{s_c s_o}{s_{co}})$. Otherwise, $\bar{s}_{oc} \in (\underline{s}_{oc}, 0)$. Solving for $g(0) = 0$, we obtain

$$\bar{\beta}_c = \frac{k_o^2 K_2 s_c^2 (s_{co} + 2s_o) (-7s_{co}^4 - 149s_{co}^3 s_o + 72s_{co}^2 s_o^2 - 56s_{co} s_o^3 + 32s_o^4) s_p^2}{8k_p K_1^3 s_o^3 (-19s_{co}^4 - 104s_{co}^3 s_o + 129s_{co}^2 s_o^2 - 248s_{co} s_o^3 + 80s_o^4)}. \quad (\text{A.37})$$

Therefore, if $\beta_c > \bar{\beta}_c$, then $g(0) < 0$ for $s_{oc} \in (0, \frac{s_c s_o}{s_{co}})$, which implies $\frac{de_o^*}{ds_{oc}} < 0$. However, if $\beta_c \leq \bar{\beta}_c$, then $g(0) \geq 0$, which means for $s_{oc} \in [0, \bar{s}_{oc})$, $g(s_{oc}) > 0$, and hence, $de_o^*/ds_{oc} > 0$; for $s_{oc} \in (\bar{s}_{oc}, s_c s_o / s_{co})$, $g(s_{oc}) < 0$, and thus, $de_o^*/ds_{oc} < 0$. This completes the proof of part (ii).

We can now turn to the impact of s_{oc} on qualities of the originator and the con-

tributor:

$$\frac{dQ_c(\mathbf{e}^*)}{ds_{oc}} = \frac{s_c s_{co}^2 (s_{co} + 2s_o)}{4K_2^3 \beta_c} + O(b) \quad (\text{A.38})$$

and

$$\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} = \frac{s_c s_o^2 (4s_o - 7s_{co}) + 2s_{co}^2 (s_{co} + 2s_o) s_{oc}}{4K_2^3 \beta_c} + O(b). \quad (\text{A.39})$$

Since $K_2 > 0$, $dQ_c(\mathbf{e}^*)/ds_{oc} > 0$. Define $\lambda_H = \min(\bar{r}, 4/7)$. Then, $dQ_o(\mathbf{e}^*)/ds_{oc} > 0$ for $s_{co}/s_o < \lambda_H$, which is from the definition of Region H. This completes the proof. ■

Proof of Proposition 2: Technically, we show that $\frac{dSW}{ds_{oc}} > 0$ and $\frac{dCS}{ds_{oc}} > 0$ under the conditions of Region H.

Substituting (A.31), (A.32), (A.33) into (2.13), and differentiating with respect to s_{oc} , we obtain

$$\frac{dCS}{ds_{oc}} = \frac{s_o (A_1 s_{oc} + A_2)}{32 (4s_o - s_{co})^6 \beta_c} + O(b), \quad (\text{A.40})$$

where $A_1 = 4s_{co}^2 (s_{co} + 2s_o) (s_{co}^2 - 42s_{co}s_o + 56s_o^2)$ and $A_2 = s_c s_o (-s_{co}^4 + 36s_{co}^3 s_o + 700s_{co}^2 s_o^2 - 1120s_{co}s_o^3 + 448s_o^4)$. Because $s_{co}/s_o < \lambda_H$, $A_1 > 0$ and $A_2 > 0$, which implies $\frac{dCS}{ds_{oc}} > 0$.

Similarly, substituting (A.31), (A.32), (A.33) into (2.14), and differentiating with respect to s_{oc} , we obtain

$$\frac{dSW}{ds_{oc}} = \frac{A_3 s_{oc} + A_4}{32 (4s_o - s_{co})^6 \beta_c} + O(b), \quad (\text{A.41})$$

where $A_3 = 2s_{co}^2 (s_{co} + 2s_o) (-s_{co}^3 + 20s_{co}^2 s_o - 60s_{co}s_o^2 + 80s_o^3)$ and $A_4 = s_c s_o^2 (3s_{co}^4$

$-160s_{co}^3s_o + 484s_{co}^2s_o^2 - 800s_{co}s_o^3 + 320s_o^4$). Because $s_{co}/s_o < \lambda_H$ implies $A_3 > 0$ and $A_4 > 0$, we obtain $dSW/ds_{oc} > 0$. This completes the proof. ■

Proof of Lemma 4: We define $z = 1/b$ and examine e_p^* , e_o^* , and e_c^* as b becomes small. Further, we can express $e_p^* \sim z^m$, $e_o^* \sim z^n$, and $e_c^* \sim z^q$ for some $m, n, q \in \mathbb{R}$.

Suppose $m \leq \max(n, q) - 1$. By (2.1), (2.2), and the definition of Region L, $Q_o \sim z^{\max(n, q)}$ and $Q_c \sim z^{\max(n, q) - 1}$. By Lemma 2, p_o is $O(z^{\max(n, q)})$. Because $|\Theta| = 1$, $\tilde{\Pi}_o$ is $O(z^{\max(n, q)})$. The effort of the originator costs $O(z^{2n-1})$. To generate non-negative profit, $z^{\max(n, q)} \geq z^{2n-1}$ should be satisfied, which gives us $n \leq 1$ if $n \geq q$ and $q \geq 2n - 1$ if $n < q$. Similarly, $\tilde{\Pi}_c$ is $O(z^{\max(n, q) - 1})$ and costs are $O(z^{2q-1})$, which requires $n \geq 2q$ if $n \geq q$ and $q \leq 0$ if $n < q$. Hence, $\max(n, q) \leq 1$. Because $m \leq \max(n, q) - 1$, we obtain $m \leq 0$. Thus, Π_p is $O(1)$. However, the proprietor could instead set $m = 1/2$ such that $\tilde{\Pi}_p \sim z^{1/2}$ and costs are $O(z^{2(1/2)-1}) = O(1)$, and hence $\Pi_p \sim z^{1/2}$. Therefore, the proprietor will deviate and we conclude that $m > \max(n, q) - 1$.

Suppose $\max(n, q) \leq 0$. Then, for the originator, Π_o is $O(1)$. However, suppose the originator instead selects $n = \frac{1}{2}$. Then $Q_o \sim z^{\frac{1}{2}}$ and $Q_c \sim z^{-\frac{1}{2}}$. Because from the previous argument, $m \geq 1/2$ and we obtain $\tau_A \sim z^{-\frac{1}{2}} > c \sim z^{-1}$, which implies Region (i) of Lemma 2 is satisfied. Thus, $p_o \sim z^{\frac{1}{2}}$, which implies $\tilde{\Pi}_o \sim z^{\frac{1}{2}}$, whereas costs are $O(z^{2(\frac{1}{2})-1}) = O(1)$, and hence $\Pi_o \sim z^{\frac{1}{2}}$ such that the originator will deviate. Therefore, $\max(n, q) > 0$.

Suppose $q \geq n$. As noted above, it follows that $q \leq 0$. However, we have shown that $\max(n, q) > 0$, which is a contradiction. Hence, $q < n$.

Suppose $m > n$, we have $Q_p > Q_o > Q_c$. The contributor's profit is given by (A.22). Substituting $Q_p = s_p e_p$, (2.1), and (2.2) into (A.22), differentiating twice with respect to e_c , and then plugging in $e_p \sim z^m$ and $e_o \sim z^n$, we find that the second order condition is satisfied, i.e., $d^2\Pi_c/de_c^2 < 0$ for all e_c . Analyzing the first order condition and collecting terms in powers of z , we obtain

$$E_1 z^{-1} + E_2 z^{q-1} + Y_1(z) = 0, \quad (\text{A.42})$$

where $E_1, E_2 \in \mathbb{R}$ and $Y_1(z)$ is a polynomial in z with terms that will be dominated by $E_1 z^{-1}$ in absolute value. As $z \rightarrow \infty$, equation (A.42) has to hold for all z values. This is not possible if $q > 0$ because in that case $E_1 z^{q-1}$ dominates all other terms in absolute value. Therefore, in equilibrium $E_1 z^{-1}$ and $E_2 z^{q-1}$ have to cancel each other, which implies $q = 0$.

Taking a total derivative of (A.24) with respect to e_o (and another one to show concavity as before), and substituting $e_p \sim z^m$, $e_o \sim z^n$ and $e_c \sim z^0$ into the first order condition, we obtain

$$E_3 z^{n-1} + E_4 + Y_2(z) = 0, \quad (\text{A.43})$$

where $E_3, E_4 \in \mathbb{R}$ and $Y_2(z)$ is a polynomial in z with terms that will be dominated by E_4 in absolute value. Similar to previous argument, we can obtain $n = 1$.

Substituting $e_p \sim z^m$, $e_o \sim z^1$ and $e_c \sim z^0$ into the first order condition of the proprietor, we obtain

$$E_5 z^{m-1} + E_6 + Y_3(z) = 0, \quad (\text{A.44})$$

where $E_5, E_6 \in \mathbb{R}$ and $Y_3(z)$ is a polynomial in z with terms that will be dominated by E_6 in absolute value. Similarly, we establish concavity and obtain $m = 1$. However, because $m = 1$ and $n = 1$, it contradicts $m > n$. Hence, $n - 1 < m \leq n$. Following the same process as above, we can rule out $m < n$ and obtain that $m = n = 1$ and $q = 0$.

Therefore,

$$e_p = P_1 z + O(1), e_o = O_1 z + O(1), \text{ and } e_c = C_1 + O(1/z). \quad (\text{A.45})$$

We can still have $Q_o > Q_p > Q_c$ if $s_p P_1 < s_o O_1$. We show below that the proprietor will always prefer P_1 such that $s_p P_1 > s_o O_1$ under the condition of Region L. Substitute (A.45) into (A.42) and solving $E_1 + E_2 = 0$, we obtain that C_1 has to satisfy the following:

$$C_1 = \begin{cases} \frac{(P_1 s_p - O_1 s_o)(O_1^2 s_o^2 \kappa_c - 5O_1 P_1 s_o s_p \kappa_c + 4P_1^2 s_p^2 \kappa_c - 6O_1 P_1 s_o c s_p \kappa_{co})}{4(4P_1 s_p - O_1 s_o)^3 \hat{\kappa}_c} & \text{if } s_p P_1 > s_o O_1; \\ \frac{(O_1 s_o - P_1 s_p)(4O_1^2 s_o^2 \kappa_c - 5O_1 P_1 s_o s_p \kappa_c + P_1^2 s_p^2 \kappa_c + 6O_1 P_1 s_o c s_p \kappa_{co})}{4(4O_1 s_o - P_1 s_p)^3 \hat{\kappa}_c} & \text{if } s_p P_1 \leq s_o O_1. \end{cases} \quad (\text{A.46})$$

Suppose that the proprietor is the quality leader. We can substitute (A.45) into (A.43) and obtain the first order condition :

$$\frac{P_1^2 s_o s_p^2 (7O_1 s_o - 4P_1 s_p)}{(O_1 s_o - 4P_1 s_p)^3} - O_1 \hat{\kappa}_o = 0. \quad (\text{A.47})$$

Given P_1 , if the originator is forced to optimize as a quality follower, we have the optimal

\hat{O}_1 solves

$$f_o(O_1) = P_1^2 s_o s_p^2 (-7O_1 s_o + 4P_1 s_p) + O_1 (O_1 s_o - 4P_1 s_p)^3 \hat{k}_o = 0. \quad (\text{A.48})$$

Because $f_o(0) = 4P_1^3 s_o s_p^3 > 0$, $f_o(4P_1 s_p / 7s_o) = -55296k_o P_1^4 s_p^4 / 2401s_o < 0$, and

$$\frac{df_o(O_1)}{dO_1} = -7P_1^2 s_o^2 s_p^2 + 4(O_1 s_o - 4P_1 s_p)^2 (O_1 s_o - P_1 s_p) \hat{k}_o < 0 \quad (\text{A.49})$$

for all $O_1 \leq P_1 s_p / s_o$, we obtain the unique root $\hat{O}_1 \in (0, \frac{4P_1 s_p}{7s_o})$ by the intermediate value theorem. Substitute \hat{O}_1 into the profit, we obtain

$$\hat{\Pi}_o = \left(-\frac{1}{2} \hat{k}_o \hat{O}_1^2 + \frac{P_1 s_o s_p \hat{O}_1 (P_1 s_p - s_o \hat{O}_1)}{(-4P_1 s_p + s_o \hat{O}_1)^2} \right) \cdot z + O(1). \quad (\text{A.50})$$

By the envelope theorem and $P_1 s_p > s_o \hat{O}_1$, we obtain

$$\frac{d\hat{\Pi}_o}{dP_1} = \frac{\partial \hat{\Pi}_o}{\partial P_1} = \frac{s_o^2 s_p \hat{O}_1^2 (2P_1 s_p + s_o \hat{O}_1)}{(4P_1 s_p - s_o \hat{O}_1)^3} \cdot z + O(1) > 0. \quad (\text{A.51})$$

Now suppose that the originator is the quality leader. Substituting (A.45) into (A.43), we can obtain the originator's first order condition:

$$\frac{O_1 (4s_o^2 (4O_1^2 s_o^2 - 3O_1 P_1 s_o s_p + 2P_1^2 s_p^2) - (4O_1 s_o - P_1 s_p)^3 \hat{k}_o)}{(4O_1 s_o - P_1 s_p)^3} = 0. \quad (\text{A.52})$$

Given P_1 , if the originator is forced to optimize as the quality leader, we have the optimal

\tilde{O}_1 solves

$$g_o(O_1) = 4s_o^2(4O_1^2s_o^2 - 3O_1P_1s_osp + 2P_1^2s_p^2) - (4O_1s_o - P_1s_p)^3\hat{k}_o = 0. \quad (\text{A.53})$$

We can see that g_o is a third order polynomial of O_1 , and

$$\frac{dg_o}{dO_1} = -192O_1^2s_o^3\hat{k}_o + O_1(32s_o^4 + 96P_1s_o^2s_p\hat{k}_o) - 12P_1s_o^3s_p - 12P_1^2s_osp^2\hat{k}_o, \quad (\text{A.54})$$

which is a quadratic function of O_1 . If $P_1 > \frac{s_o^2}{3s_p\hat{k}_o}$, $\frac{dg_o}{dO_1} < 0$, and (A.53) has only one real root. If $P_1 \leq \frac{s_o^2}{3k_osp}$, the polynomial (A.54) has two roots:

$$r_1 = \frac{s_o^3 + 3P_1s_osp\hat{k}_o - \sqrt{s_o^6 - 3P_1s_o^4s_p\hat{k}_o}}{12s_o^2\hat{k}_o} \text{ and } r_2 = \frac{s_o^3 + 3P_1s_osp\hat{k}_o + \sqrt{s_o^6 - 3P_1s_o^4s_p\hat{k}_o}}{12s_o^2\hat{k}_o}. \quad (\text{A.55})$$

Since $g_o(r_1) > 0$ and $g_o(r_2) > 0$, the only root of (A.53) satisfies $\tilde{O}_1 > r_2$. Therefore, (A.53) has only one real root \tilde{O}_1 given any value of P_1 . We obtain the profit of the originator in this case:

$$\tilde{\Pi}_o = \left(-\frac{1}{2}\hat{k}_o\tilde{O}_1^2 + \frac{4s_o^2\tilde{O}_1^2(\tilde{O}_1s_o - P_1s_p)}{(4\tilde{O}_1s_o - P_1s_p)^2} \right) \cdot z + O(1). \quad (\text{A.56})$$

By envelope theorem and $P_1s_p \leq s_o\tilde{O}_1$, we have

$$\frac{d\tilde{\Pi}_o}{dP_1} = \frac{\partial\tilde{\Pi}_o}{\partial P_1} = \frac{4s_o^2s_p\tilde{O}_1^2(P_1s_p + 2s_o\tilde{O}_1)}{(P_1s_p - 4s_o\tilde{O}_1)^3} \cdot z + O(1) < 0. \quad (\text{A.57})$$

By the monotone properties established in (A.51) and (A.57), $\hat{\Pi}_o < \tilde{\Pi}_o$ if $P_1 = 0$, and $\hat{\Pi}_o > 0 > \tilde{\Pi}_o$ if $P_1 = 4s_o^2/9\hat{k}_o$, hence there exists a unique $\bar{P}_1 \in (0, 4s_o^2/9\hat{k}_o)$ such that if $P_1 < \bar{P}_1$, $\hat{\Pi}_o < \tilde{\Pi}_o$, which means the originator will prefer to be the quality leader (choosing \tilde{O}_1); if $P_1 > \bar{P}_1$, $\hat{\Pi}_o > \tilde{\Pi}_o$, which implies that the originator will prefer to be the quality follower (choosing \hat{O}_1). We next turn to the proprietor's effort decision.

Suppose $P_1 > \bar{P}_1$, then the proprietor is the quality leader. Solving $E_5 + E_6 = 0$, we can obtain the first order condition of the proprietor as

$$\begin{aligned}
f_p(P_1) &= 8P_1^2 s_o^2 s_p^4 (-\hat{O}_1 s_o + P_1 s_p) (7\hat{O}_1 s_o + 8P_1 s_p) \\
&\quad + 4s_p^2 (-\hat{O}_1 s_o + 4P_1 s_p)^3 (2\hat{O}_1^2 s_o^2 - 3\hat{O}_1 P_1 s_o s_p + 4P_1^2 s_p^2) \hat{k}_o \\
&\quad - (\hat{O}_1 s_o - 4P_1 s_p)^2 (2P_1^2 s_o^2 s_p^2 (7\hat{O}_1 s_o + 8P_1 s_p) + (\hat{O}_1 s_o - 4P_1 s_p)^4 \hat{k}_o) \hat{k}_p \\
&= 0.
\end{aligned} \tag{A.58}$$

We have shown that (A.48) has only one unique root $\hat{O}_1 \in (0, 4P_1 s_p/7s_o)$. Let $r = \frac{s_o \hat{O}_1}{s_p P_1}$, then there exists a unique $r \in (0, \frac{4}{7})$. Plugging r into (A.48), we can obtain

$$P_1 = \frac{(-4 + 7r)s_o^2}{(-4 + r)^3 r s_p \hat{k}_o}, \quad O_1 = \frac{(-4 + 7r)s_o}{(-4 + r)^3 \hat{k}_o}. \tag{A.59}$$

Substituting (A.59) into (A.58), we obtain

$$4r(64 - 112r + 132r^2 - 139r^3 + 28r^4)t + (-64 + 176r - 196r^2 + 147r^3) = 0, \tag{A.60}$$

where $t = \frac{\hat{k}_o s_p^2}{\hat{k}_p s_o^2} \geq 1$ is defined in the conditions of Region L. Given t , we can solve (A.60) for the unique solution r . Substituting (A.59) into (A.46), we can obtain P_1 , O_1 , C_1 , and the proprietor's profit $\hat{\Pi}_p$ when it is the quality leader as

$$\hat{\Pi}_p = \frac{M_1 (8rtM_2M_3 - M_1) s_o^2 z}{2r^2 t M_2^6 \hat{k}_o} + O(1), \quad (\text{A.61})$$

where $M_1 = 4 - 7r$, $M_2 = 4 - r$, and $M_3 = 1 - r$.

Now suppose $P_1 < \bar{P}_1$, we can also obtain the first order condition as

$$\begin{aligned} g_p(P_1) = & \tilde{O}_1 s_o^2 s_p (8P_1^2 s_o s_p^2 (\tilde{O}_1 s_o - P_1 s_p) (5\tilde{O}_1 s_o + P_1 s_p) + \tilde{O}_1 (4\tilde{O}_1 s_o - 7P_1 s_p) \\ & (4\tilde{O}_1 s_o - P_1 s_p)^3 \hat{k}_o) - P_1 (-4\tilde{O}_1 s_o + P_1 s_p)^2 (8P_1^2 s_o^2 s_p^2 (5\tilde{O}_1 s_o + P_1 s_p) \\ & + (-4\tilde{O}_1 s_o + P_1 s_p)^4 \hat{k}_o) \hat{k}_p = 0. \end{aligned} \quad (\text{A.62})$$

We have shown that (A.53) has only one unique root $\tilde{O}_1 \in (P_1 s_p / s_o, \infty)$. Let $r = \frac{s_o \tilde{O}_1}{s_p P_1}$, then there exists a unique $\tilde{r} \in (1, \infty)$. Combining (A.62) and (A.53), we can obtain

$$P_1 = \frac{4(2 - 3\tilde{r} + 4\tilde{r}^2) s_o^2}{(-1 + 4\tilde{r})^3 s_p \hat{k}_o}, \quad O_1 = \frac{4\tilde{r}(2 - 3\tilde{r} + 4\tilde{r}^2) s_o}{(-1 + 4\tilde{r})^3 \hat{k}_o}, \quad (\text{A.63})$$

where $\tilde{r} = \frac{s_o \tilde{O}_1}{s_p P_1} > 1$ solves

$$\begin{aligned} & (2 + 14\tilde{r} - 127\tilde{r}^2 + 196\tilde{r}^3 - 176\tilde{r}^4 + 64\tilde{r}^5) t - 4(42 - 95\tilde{r} + 164\tilde{r}^2 - 112\tilde{r}^3 + 64\tilde{r}^4) = 0. \\ & (\text{A.64}) \end{aligned}$$

Given t , we can solve for \tilde{r} and obtain the proprietor's profit $\tilde{\Pi}_p$ in this case:

$$\tilde{\Pi}_p = \frac{4(2 - 3\tilde{r} + 4\tilde{r}^2)(-4 + (6+t)\tilde{r} - (8+5t)\tilde{r}^2 + 4t\tilde{r}^3)s_o^2z}{t(1-4\tilde{r})^6\hat{k}_o} + O(1). \quad (\text{A.65})$$

Comparing the profits in (A.61) and (A.65), we can obtain that $\hat{\Pi}_p > \tilde{\Pi}_p$ if $t = 1$.

Further, we can obtain

$$\frac{d\hat{\Pi}_p}{dt} = \frac{(4-7r)^2s_o^2z}{2(-4+r)^6r^2t^2\hat{k}_o} + O(1) \text{ and } \frac{d\tilde{\Pi}_p}{dt} = \frac{8(2+\tilde{r}(-3+4\tilde{r}))^2s_o^2z}{t^2(1-4\tilde{r})^6\hat{k}_o} + O(1). \quad (\text{A.66})$$

Because r is determined by (A.60) and we have shown that there is one unique root r in $(0, \frac{4}{7})$ for any t . We obtain

$$t = \frac{64 - 176r + 196r^2 - 147r^3}{4r(64 - 112r + 132r^2 - 139r^3 + 28r^4)}.$$

By the inverse function theorem, $r'(t) = \frac{1}{t'(r)}$. We then can obtain the sign of $r'(t)$ by looking at $t'(r)$'s sign:

$$t'(r) = \frac{-4096 + 14336r - 32512r^2 + 63232r^3 - 91760r^4 + 74200r^5 - 36897r^6 + 8232r^7}{4r^2(64 - 112r + 132r^2 - 139r^3 + 28r^4)^2}.$$

Because the numerator is negative and the denominator is positive for $r \in (0, \frac{4}{7})$, it follows that $t'(r) < 0$. Hence, we obtain that as t increases, r decreases. Because $\bar{r} \approx 0.20$ when $t = 1$. Therefore $r \in (0, \bar{r})$ for $t \geq 1$. Similarly, we note that in (A.64), $\tilde{r} > \underline{r} \approx 1.53$. By investigating (A.66) within these ranges, we can obtain that $\frac{d\hat{\Pi}_p}{dt} > \frac{d\tilde{\Pi}_p}{dt} \forall t \geq 1$.

Therefore, the proprietor would prefer $P_1 > \bar{P}_1$. According to the analysis above, we can conclude that

$$P_1 = \frac{M_1 s_o^2}{r \hat{k}_o M_2^3 s_p b}, \quad O_1 = \frac{M_1 s_o}{\hat{k}_o M_2^3 b}, \quad \text{and} \quad C_1 = \frac{M_3 (s_o \kappa_c M_2 M_3 - 6r s_{oc} \kappa_{co})}{4s_o M_2^3 \hat{k}_c}, \quad (\text{A.67})$$

where $M_1 = 4 - 7r$, $M_2 = 4 - r$, and $M_3 = 1 - r$. Note for $C_1 > 0$, we need

$$s_{oc} < \frac{(4-5r+r^2)s_o \kappa_c}{6r \kappa_{co}}. \quad \text{Define } \lambda_L = (4 - 5r + r^2) / 6r. \quad \text{We have } C_1 > 0 \text{ for } s_{oc} < \lambda_L s_o \kappa_c / \kappa_{co},$$

which is given in the definition of Region L.

Substituting P_1 , O_1 , and C_1 into the first order conditions, following similar process as above, we can find the second terms in the proprietor and the originator's equilibrium effort levels, which gives us

$$e_p = \frac{M_1 s_o^2}{r k_o M_2^3 s_p b} - \frac{M_6}{k_c M_2^3 s_o (k_p M_4^3 s_o^2 + 24r^3 k_o M_5 s_p^2)} + O(b), \quad (\text{A.68})$$

$$e_o = \frac{M_1 s_o}{k_o M_2^3 b} + \frac{M_7}{2k_c k_o M_2^3 s_o^2 (k_p M_4^3 s_o^2 + 24r^3 k_o M_5 s_p^2)} + O(b), \quad (\text{A.69})$$

and

$$e_c = \frac{M_3 (\kappa_c M_2 M_3 s_o - 6r \kappa_{co} s_{oc})}{4k_c M_2^3 s_o} + O(b), \quad (\text{A.70})$$

where $M_4 = 16 - 16r + 21r^2$, $M_5 = -256 - 1920r + 1888r^2 - 1160r^3 - 189r^4 + 98r^5$,

$$\begin{aligned} M_6 = & 2r^2 s_p ((17408 - 151808r + 437760r^2 - 479680r^3 + 211804r^4 + 24621r^5 - \\ & 100516r^6 + 19411r^7 + 588r^8) s_o^2 \kappa_{co} \hat{k}_c + (11264 - 21248r - 41472r^2 + 113216r^3 \\ & - 100076r^4 + 46239r^5 - 874r^6 - 2093r^7 + 147r^8) s_o s_{oc} \kappa_c \hat{k}_o + 6(2048 - 8192r - 576r^2 + \end{aligned}$$

$19120r^3 - 16204r^4 + 7455r^5 + 3234r^6) s_{oc}^2 \kappa_{co} \hat{k}_o$), and

$$\begin{aligned} M_7 = & 8r^4(-1792 + 23680r - 35040r^2 + 16712r^3 - 6611r^4 - 951r^5 + 308r^6 + \\ & 49r^7) s_o s_{oc} s_p^2 \kappa_c \hat{k}_o^2 - 24r^4(1280 - 7616r + 11184r^2 - 3164r^3 + 3506r^4 + 2373r^5 + 294r^6) \\ & s_{oc}^2 s_p^2 \kappa_{co} \hat{k}_o^2 - 2(32 - 136r + 128r^2 + 85r^3 - 116r^4 + 7r^5) M_4^2 s_o^4 \kappa_{co} \hat{k}_c \hat{k}_p + r(-80 + 188r - \\ & 102r^2 - 13r^3 + 7r^4) M_4^2 s_o^3 s_{oc} \kappa_c \hat{k}_o \hat{k}_p - r s_o^2 \kappa_{co} \hat{k}_o (8r^2(-3072 - 4352r + 39680r^2 - 12672r^3 \\ & - 15620r^4 - 4783r^5 - 17520r^6 + 3367r^7 + 392r^8) s_p^2 \hat{k}_c + 3(16 - 68r + 40r^2 + 21r^3) M_4^2 s_{oc}^2 \\ & \hat{k}_p). \quad \square \end{aligned}$$

Proof of Proposition 3: Technically, we show that

$$(i) \frac{de_p^*}{ds_{oc}} < 0, \frac{de_o^*}{ds_{oc}} < 0, \text{ and } \frac{de_c^*}{ds_{oc}} < 0;$$

$$(ii) \frac{dQ_p(\mathbf{e}^*)}{ds_{oc}} < 0, \text{ and } \frac{dQ_c(\mathbf{e}^*)}{ds_{oc}} < 0;$$

$$(iii) \text{ There exists } \bar{s}_{oc} \in (0, \lambda_{Ls_o} \kappa_c / \kappa_{co}), \text{ such that } \frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} > 0 \text{ for } s_{oc} \in [0, \bar{s}_{oc}),$$

and $\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} < 0$ for $s_{oc} \in (\bar{s}_{oc}, \lambda_{Ls_o} \kappa_c / \kappa_{co})$.

First, differentiating (A.70) with respect to s_{oc} , we obtain

$$\frac{de_c^*}{ds_{oc}} = -\frac{3(r\kappa_{co}M_3)}{2(\hat{k}_c M_2^3 s_o)} + O(b). \quad (\text{A.71})$$

By our proof in Lemma 4, $r \in (0, \bar{r})$, where $\bar{r} < 1/5$. Hence, $M_3 = 1 - r > 0$ and $M_2 = 4 - r > 0$. Therefore, $\frac{de_c^*}{ds_{oc}} < 0$. Next, differentiating (A.69) with respect to s_{oc} , we obtain

$$\frac{de_o^*}{ds_{oc}} = -\frac{r(M_2(8r^3 t A_3 + A_1 M_4^2) s_o \kappa_c + 6(8r^3 t A_4 + A_2 M_4^2) s_{oc} \kappa_{co})}{2(M_4^3 + 24r^3 t M_5) s_o^2 M_2^3 \hat{k}_c} + O(b), \quad (\text{A.72})$$

where $A_1 = 20 - 42r + 15r^2 + 7r^3$, $A_2 = 16 - 68r + 40r^2 + 21r^3$, $A_3 = 448 - 5808r +$

$7308r^2 - 2351r^3 + 1065r^4 + 504r^5 + 49r^6$, and $A_4 = 1280 - 7616r + 11184r^2 - 3164r^3 + 3506r^4 + 2373r^5 + 294r^6$. Given $r \in (0, \bar{r})$, we obtain that $M_4^3 + 24r^3tM_5 > 0$, $8r^3tA_3 + A_1M_4^2 > 0$, and $8r^3tA_4 + A_2M_4^2 > 0$. Therefore, $\frac{de_o^*}{ds_{oc}} < 0$. Further, because $\frac{de_o^*}{ds_{oc}} < 0$ and $\frac{de_c^*}{ds_{oc}} < 0$, we have $\frac{dQ_c(\mathbf{e}^*)}{ds_{oc}} = s_c \frac{de_c^*}{ds_{oc}} + s_{co} \frac{de_o^*}{ds_{oc}} < 0$. For the proprietor,

$$\frac{de_p^*}{ds_{oc}} = -\frac{2r^2t(A_5s_o\kappa_c + 12A_6s_{oc}\kappa_{co})}{\hat{k}_cM_2^3(M_4^3 + 24r^3tM_5)s_osp} + O(b), \quad (\text{A.73})$$

where $A_5 = 11264 - 21248r - 41472r^2 + 113216r^3 - 100076r^4 + 46239r^5 - 874r^6 - 2093r^7 + 147r^8$, and $A_6 = 2048 - 8192r - 576r^2 + 19120r^3 - 16204r^4 + 7455r^5 + 3234r^6$.

We can obtain that $A_5 > 0$ and $A_6 > 0 \forall r \in (0, \bar{r})$. Therefore, we have $\frac{de_p^*}{ds_{oc}} < 0$. Then $\frac{dQ_p(\mathbf{e}^*)}{ds_{oc}} = s_p \frac{de_p^*}{ds_{oc}} < 0$. This completes the proof of part (i) and (ii). We now turn to part (iii) in the proposition:

$$\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} = \frac{A_7\kappa_cs_o - 12A_8r\kappa_{co}s_{oc}}{4\hat{k}_cM_2^3(M_4^3 + 24r^3tM_5)s_o} + O(b), \quad (\text{A.74})$$

where $A_7 = -M_2(16r^4tA_3 + 2rA_1M_4^2 - (M_4^3 + 24r^3tM_5)M_3^2)$, and $A_8 = A_2M_4^2 + M_4^3M_3 + 8r^3t(A_4 + 3M_5M_3)$. We obtained that $M_4^3 + 24r^3tM_5 > 0$. We can further obtain that $r \in (0, \bar{r})$ implies $A_7 > 0$ and $A_8 > 0$, which means there exists $\bar{s}_{oc} = A_7\kappa_cs_o/12r\kappa_{co}A_8 \in (0, \lambda_L\kappa_cs_o/\kappa_{co})$, such that when $s_{oc} \leq \bar{s}_{oc}$, $\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} \geq 0$, and when $s_{oc} > \bar{s}_{oc}$, $\frac{dQ_o(\mathbf{e}^*)}{ds_{oc}} < 0$.

This completes the proof. ■

Proof of Proposition 4: Technically, we show that

- (i) There exists $\bar{s}_{oc} \in (0, \lambda_Ls_o\kappa_c/\kappa_{co})$, such that $\frac{dCS}{ds_{oc}} \geq 0$ for $s_{oc} \leq \bar{s}_{oc}$, and $\frac{dCS}{ds_{oc}} < 0$

for $s_{oc} > \bar{s}_{oc}$;

(ii) There exists $\tilde{s}_{oc} \in (0, \lambda_L s_o \kappa_c / \kappa_{co})$, such that $\frac{dSW}{ds_{oc}} \geq 0$ for $s_{oc} \leq \tilde{s}_{oc}$, and $\frac{dSW}{ds_{oc}} < 0$ for $s_{oc} > \tilde{s}_{oc}$.

Substituting (A.68), (A.69), (A.70) into (2.13), and differentiating with respect to s_{oc} , we obtain

$$\frac{dCS}{ds_{oc}} = \frac{A_1 - A_2 s_{oc}}{8M_2^6 (M_4^3 + 24r^3 t M_5) s_o \hat{k}_c} + O(b), \quad (\text{A.75})$$

where $A_1 = 8r^2(-11264 + 18688r + 7168r^2 + 43840r^3 - 78164r^4 + 33787r^5 - 31874r^6 - 5999r^7 + 490r^8)tM_2^2 s_o \kappa_c + (112 - 176r - 35r^2)(16 - 64r + 63r^2 - 16r^3 + r^4)M_4^2 s_o \kappa_c$ and $A_2 = -96r^2(-32768 + 155648r - 82944r^2 - 44288r^3 - 87040r^4 + 109056r^5 - 242516r^6 - 9289r^7 + 16170r^8)t\kappa_{co} - 12r(-8 + 11r)(112 - 176r - 35r^2)M_4^2 \kappa_{co}$.

Given $r \in (0, \bar{r})$, we can obtain that $A_1 > 0$ and $A_2 > 0$. Because we have already shown that $M_4^3 + 24r^3 t M_5 > 0$, we obtain $\bar{s}_{oc} = A_1/A_2 \in (0, \lambda_L s_o \kappa_c / \kappa_{co})$ such that if $s_{oc} \leq \bar{s}_{oc}$, $dCS/ds_{oc} \geq 0$; if $s_{oc} > \bar{s}_{oc}$, $dCS/ds_{oc} < 0$. Substituting (A.68), (A.69), (A.70) into (2.14), and differentiating with respect to s_{oc} , we obtain

$$\frac{dSW}{ds_{oc}} = \frac{A_3 - A_4 s_{oc}}{8M_2^6 (M_4^3 + 24r^3 t M_5) s_o \hat{k}_c} + O(b), \quad (\text{A.76})$$

where $A_3 = (81920 - 421888r - 54136r^{10}t - 49952r^{11}t + 2352r^{12}t + 357r^9(-343 + 800t) - 1024r^2(-1381 + 1056t) + 256r^3(-17503 + 9600t) + 8r^8(77420 + 26051t) + 64r^4(142835 + 34592t) - 16r^5(657761 + 526208t) - r^7(3159289 + 4343296t) + r^6(7370060 + 8278656t))M_2 s_o \kappa_c$,

$$A_4 = 12r^2(r^7(129654 - 709000t) + 185136r^8t + 84672r^9t + 12288(3 + 64t) -$$

$$24576r(-41 + 152t) + 256r^2(-17667 + 8992t) + 512r^3(13611 + 10604t) \\ - 112r^4(48207 + 72704t) + 32r^5(74529 + 170656t) + r^6(-239757 + 226592t)) \kappa_{co}.$$

Similarly, we obtain $A_3 > 0$ and $A_4 > 0$ for $r \in (0, \bar{r})$. Therefore, we obtain $\tilde{s}_{oc} = A_3/A_4 \in (0, \lambda_L s_o \kappa_c / \kappa_{co})$, such that $dSW/ds_{oc} \geq 0$ for $s_{oc} \leq \tilde{s}_{oc}$, and $dSW/ds_{oc} < 0$ for $s_{oc} > \tilde{s}_{oc}$. ■

References

- Asundi, J., O. Carare, K. Dogan. 2012. Competitive implications of software open-sourcing. *Decision Support Systems* **54**(1) 153–163.
- Athey, S., G. Ellison. 2014. Dynamics of open source movements. *Journal of Economics & Management Strategy* **23**(2) 294–316.
- August, T., H. Shin, T. I. Tunca. 2012. Generating value through open source: Software service market regulation and licensing policy. Working paper, University of California, San Diego, University of Maryland.
- August, T, H. Shin, T. I. Tunca. 2013. Licensing and competition for services in open source software. *Information Systems Research* **24**(4) 1068–1086.
- Belenzon, S., M. Schankerman. 2015. Motivation and sorting of human capital in open innovation. *Strategic Management Journal* **36**(6) 795–820.
- Bitzer, J. 2004. Commercial versus open source software: the role of product heterogeneity in competition. *Economic Systems* **28**(4) 369–381.
- Butler, M. 2011. Android: Changing the mobile landscape. *IEEE Pervasive Computing* **10**(1) 4–7.
- Casadesus-Masanell, R., P. Ghemawat. 2006. Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Science* **52**(7) 1072–1084.
- Casadesus-Masanell, R., G. Llanes. 2011. Mixed source. *Management Science* **57**(7) 1212–1230.
- Cheng, H. K., Y. Liu, Q. C. Tang. 2011. The impact of network externalities on the competition between open source and proprietary software. *Journal of Management Information Systems* **27**(4) 201–230.
- Colazo, J., Y. Fang. 2009. Impact of license choice on open source software development activity. *Journal of the American Society for Information Science and Technology* **60**(5) 997–1011.

- Corbet, J., G. Kroah-Hartman, A. McPherson. 2013. Linux kernel development: How fast it is going, who is doing it, what they are doing, and who is sponsoring it (2013 edition). Tech. rep., The Linux Foundation.
- Economides, N., E. Katsamakos. 2006. Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science* **52**(7) 1057–1071.
- Fershtman, C., N. Gandal. 2007. Open source software: Motivation and restrictive licensing. *International Economics and Economic Policy* **4**(2) 209–225.
- Gonzalez-Barahona, J. M. 2013. Companies contributing to Open-Stack (Apr 2014). <http://blog.bitergia.com/2013/04/04/companies-contributing-to-openstack-grizzly-analysis/>.
- Hars, A., S. Ou. 2002. Working for free? motivations for participating in open-source projects. *Int. J. Electron. Commerce* **6**(3) 25–39.
- Haruvy, E., S. P. Sethi, J. Zhou. 2008. Open source development with a commercial complementary product or service. *Production and Operations Management* **17**(1) 29–43.
- Iansiti, M., G. L. Richards. 2006. The business of free software: Enterprise, incentives, investment, and motivation in the open source community. Working Paper, Harvard Business School, Boston, MA.
- Kim, B. C., P. Chen, T. Mukhopadhyay. 2006. Pricing open source software. *ICIS 2006 Proceedings*. Milwaukee, WI.
- Laurent, A. M. St. 2004. *Understanding Open Source and Free Software Licensing*. O'Reilly Media, Inc.
- Lee, D., H. Mendelson. 2008. Divide and conquer: Competing with free technology under network effects. *Production and Operations Management* **17**(1) 12–28.
- Lerner, J., J. Tirole. 2005a. The economics of technology sharing: Open source and beyond. *The Journal of Economic Perspectives* **19**(2) 99–120.
- Lerner, J., J. Tirole. 2005b. The scope of open source licensing. *J Law Econ Organ* **21**(1) 20–56.
- Mustonen, M. 2005. When does a firm support substitute open source programming? *Journal of Economics & Management Strategy* **14**(1) 121–139.
- Oi, W. Y. 1983. Heterogeneous firms and the organization of production. *Economic*

Inquiry **21**(2) 147–171.

OSI. 2014. Open source licenses (May 2014). <http://opensource.org/licenses>.

Polanski, A. 2007. Is the general public licence a rational choice? *Journal of Industrial Economics* **55**(4) 691–714.

Roberts, J. A., I. H. Hann, S. A. Slaughter. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. *Management Science* **52**(7) 984–999.

Sen, R. 2007. A strategic analysis of competition between open source and proprietary software. *Journal of Management Information Systems* **24**(1) 233–257.

Singh, P. V., C. Phelps. 2013. Networks, social influence, and the choice among competing innovations: Insights from open source software licenses. *Information Systems Research* **24**(3) 539–560.

Stewart, K. J., A. P. Ammeter, L. M. Maruping. 2006. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research* **17**(2) 126–144.

von Krogh, G., E. von Hippel. 2006. The promise of research on open source software. *Management Science* **52**(7) 975–983.

Wen, W., C. Forman, S. J. H. Graham. 2013. The impact of intellectual property rights enforcement on open source software project success. *Information Systems Research* **24**(4) 1131–1146.

Zhu, K. X., Z. Z. Zhou. 2012. Lock-in strategy in software competition: Open-source software vs. proprietary software. *Information Systems Research* **23**(2) 536–545.

Chapter 2 is currently under review for publication of the material. August, Terrence, Wei Chen, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.

Chapter 3

“Release Early, Release Often”?

The Impact of Release Frequency in Open-Source Software Co-Creation

A central virtue of open-source software (OSS) is the contribution from the communities, yet our knowledge of how to coordinate and maximize the benefit of such contributions is limited. In this paper, we study the impact of release frequency as a coordinating device in the adoption and evolution of open source projects. We first build a stylized model to characterize an OSS project as a two-sided market, with the two sides of consumers and community developers. According to the model, release frequency should have a curvilinear (inverse-U) relationship with both adoption and community contribution. Our empirical evidences support this hypotheses. Releasing too often seems to backfire due to the subtle effects on the supply side: it may exhaust the community contribution. High adoption cost and development cost may attenuate the effectiveness of frequent release. Furthermore, if the consumers can benefit more from the community contribution, higher release frequency might be helpful. Meanwhile, our results also show

that high release frequency may decrease the absorption of contribution by the OSS team. These results bring implications for managing technology-enabled collaboration in open-source communities and research on open-source software, open innovation, and software adoption.

3.1 Introduction

Open source software (OSS) has become an important example of collaborative production and a new “private-collective” model of innovation (von Hippel and von Krogh 2003). Extending beyond successful software products such as Linux, Apache, and MySQL, even non-software products are now embracing the open-source approach, as seen by Tesla’s opening up of its patent portfolio to its rivals. However, innovation in OSS does not always unfold smoothly, with many projects failing to reach critical mass of download by users or attract enough contribution by developers (Chengalur-Smith and Sidorova 2003; Fogel 2005). To better understand and manage the success of OSS, the governance of OSS stands out as an important research topic (von Krogh and von Hippel 2006).

The central virtue of the open-source approach is the availability of a large pool of skilled labor who can contribute to the refinement of the product – specifically, product testing, bug fixing, and quality enhancing. This pool for open-source community usually is limited though it may be larger than proprietary closed-source communities. Specifically, the increasing number of OSS products and their frequent releases compete

for the time of the community and may deplete the community and make it a capacitated resource.

One specific aspect of the governance of OSS that has not been addressed to date is the management of the finite common pool of the OSS community capacity. An OSS project usually starts with a small core team, and then grows as it attracts community contribution. We define *community contribution* as the work emerged from those users and developers outside the core team. Since most core teams are small and have limited resources, community contributions are important to the quality and diffusion of the OSS products (Setia et al. 2012). The accumulation of knowledge and coordination between the core team and the community is critical to the success and sustainability of the projects. Like other common community resources (Ostrom 1990), the OSS community must also be managed.

While previous studies identify important project characteristics affecting OSS success (e.g. Grewal et al. 2006; Singh et al. 2011; Stewart et al. 2006), how to manage the co-creation process has not been well studied, which leaves a gap in our understanding of the nature of OSS co-creation. Seeking to narrow this gap, we examine how to coordinate the co-creation process by focusing on the impact of release frequency in the development and adoption process of an OSS project. Release frequency refers to *the number of releases in a defined time period*. It is a deliberate decision made by the core team, and as we will see in this paper, has an important effect on community contribution and co-creation success.

In the influential essay “The Cathedral and the Bazaar”, Raymond (1999) advocates that an OSS project should “release early, release often” to receive more

community contribution and hence produce a higher quality product, which implies that frequent releases could help OSS projects succeed in competition. The reason is what Raymond attributes as Linus's Law - "given enough eyeballs, all bugs are shallow". Over the years, this has been accepted as a "common wisdom" in OSS development (Crowston et al. 2006; Michlmayr 2007). A large community could help detect the bugs and enhance the quality, thus being the most valuable resource to an OSS project. But the community should be big "enough" and the community has to be willing to contribute. To the best of our knowledge, no prior research has empirically evaluated this strategy, that is, the effect of release frequency in the context of the OSS co-creation process.

Motivated by the issues identified above, we seek to study the following research questions: (1) How is the release frequency of an OSS project associated with its adoption as measured by download? (2) What is the effect of release frequency on community contribution? (3) What is the influence of various environmental factors such as adoption cost, development cost, etc., in this process?

We begin by constructing a two-sided market model to explain the consumers' adoption decisions and the community developers' contribution decisions. Our model features the cross externalities the consumers and developers impose on each other (Rochet and Tirole 2006). The change of release frequency would affect both sides due to these externalities. Our model predicts a curvilinear relationship between release frequency and download. A similar relationship also exists between release frequency and community contribution.

We then formulate hypotheses and empirically test them with a panel data set assembled from SourceForge.net. Our empirical evidence is consistent with our

theoretical predictions. While releasing frequently is initially positively associated with download, releasing too often may hurt the adoption of the project. In fact, release frequency has a curvilinear relationship with download. This relationship may emanate from the supply side of the OSS co-creation process, where we also find a curvilinear relationship between release frequency and community contribution. Further analysis shows that both adoption and development cost may attenuate the effectiveness of high release frequency. Apparently fast release frequency when the costs are high may decrease the consumers' incentive to adopt and even exhaust the community's incentive to contribute. Moreover, "release early, release often" may work better when the consumers could benefit more from the community contributions, which highlights the cross externalities in the two-sided market. We also find that releasing too often may decrease the core team's absorption of community contribution. These findings have implications for managing the co-creation process of OSS and offer insights for the emerging open model of production.

Our key theoretical view point is that the open-source community must be carefully managed like a common pool of exhaustible shared resources. In this sense, the management of the community contribution is like tackling the common-pool resources problem in the economics literature (Ostrom 1990), although a key difference is that natural resources cannot actively/endogenously resist depletion. With OSS, the community can choose which OSS products to participate based on the alignment of values and incentives, indirectly excluding and limiting some software developers. In this sense, the OSS community resource is somewhere between a private good and common good (O'Mahony 2007). Software teams must understand that they need to manage the

open source community contribution carefully: while sharing of information does not necessarily deplete the OSS resource, providing feedback by testing and bug-fixing requires effort, and can deplete the OSS community. Specifically, we propose that the developer teams risk exhausting the community, and not receiving adequate contributions if they overextend the release frequency of their products.

3.2 Literature Review

Our study is related to the literature on software release policies. Several recent studies consider how release timing relates to market demand. Arora et al. (2006) argue that a software vendor may want to release a product early when the market is larger, because a larger market decrease the average cost to patch after the release. Studies also point out that the ability to harness user contributions after release creates incentives for the software vendor to release early (Jiang et al. 2011; August and Niculescu 2013). Even though these studies treat software release (upgrade) as a one-time event, their insights could be extended to sequential releases of software products in our research.

Empirical research on software release policy is rare. The following studies are noteworthy. Banker and Slaughter (1997) find that managers would rather release small modifications even though combining small releases could potentially reduce maintenance cost. They attribute opportunity costs of delaying projects as the main explanation. Later studies in this stream examine different aspect of software maintenance (Banker et al. 1998; Harter et al. 2000; Harter and Slaughter 2003). The main focus, though, is the internal perspective of software development. In this paper, we

focus on the effect of release frequency on the external market demand and community contribution, therefore providing empirical evidence on software release policy in an open innovation setting.

Our research is also related to the new product development (NPD) literature. A central question in this literature is the trade-off between time-to-market and product performance (e.g., Cohen et al. 1996; Agarwal and Bayus 2002). Similar to studies on software release policy, this literature also treats product release as a one-time event. Nonetheless these studies provide good insights for decisions on product release time. Our context enables us to test the effect of release frequency by controlling the quality improvement, which provides an opportunity to separate the time-to-market effect from the quality improvement.

Beyond the trade-offs of product release, the OSS development falls more into the scenario of rapid sequential innovation (Kornish 2001; Ramachandran and Krishnan 2008), because a project may undergo significant changes after the initial release. Our context is quite different from the physical goods setting in the literature. First, the marginal cost of improving the software after initial release is minimal comparing to physical goods. Besides, since OSS products are offered for free in our context, we examine the effect of release frequency independent of the pricing issues. Furthermore, while the sequential innovation literature focuses on the pricing and product design issues, our focus is on the collaboration between the core team and the community.

Our paper also extends the aforementioned research into the new realm of innovation with user communities (von Hippel 2001). User contribution started with open source software, but now has expanded to open innovation in many other areas

(Boudreau and Lakhani 2009), such as crowdsourcing ideas (Bayus 2013), co-creating products (Kornish and Ulrich 2014), and the mobile app economy (Ghose and Han 2014). However, the product release problem discussed in earlier literature is nearly untapped in this area. Our paper treats the OSS project as a two-sided market (Parker and Van Alstyne 2005; Economides and Katsamakas 2006) and incorporates the decisions of both the consumers and community developers. We then examine the effect of release frequency in both the adoption and development processes, and hopefully provide insights into how to coordinate the effort of the user/developer community in this new approach.

In terms of OSS, since the seminal work of Lerner and Tirole (2002), the literature on the motivations of OSS contributors has revealed that developers contribute for a mix of intrinsic and extrinsic motivations (Roberts et al. 2006). While prior research explains why programmers voluntarily contribute to OSS, few studies have focused on how to manage OSS projects, especially on how to coordinate these diverse motivations to achieve project success. Our analysis also extend the OSS literature to release policy decisions, which the OSS team can manage in the continuous improvement process.

3.3 Theoretical Analysis

To address our research questions, we study the effect of release frequency in both the adoption and development processes of OSS. In terms of adoption, we focus on the success of an OSS project in garnering user interest (Stewart et al. 2006), which is measured by the download of a project. In terms of development, we examine the impact

of release frequency on the community contribution. We also analyze the moderating effects of costs and externalities from both sides. These analyses allow us to delve into the subtle effects of environment factors in OSS co-production. This is geared to understand release frequency as a coordinating device between the consumers and the community developers.

3.3.1 The Impact of Release Frequency: A Two-sided Model

We begin with a highly stylized model of OSS co-creation, which we hope will capture the key interactions behind the impact of release frequency. While deciding the release frequency, an OSS project faces two interacting sides: one side for consumers and another for community developers. The two sides have cross externalities. Specifically, during a given time period T , an OSS project has a fixed quality enhancement Q and number of releases $n \in [0, \bar{N}]$, where \bar{N} is a finite positive number. We assume that each release delivers an equal share of the quality enhancement to consumers. The release frequency decision n will affect the decisions of both sides.

On the consumer side, consumers benefit from the quality enhancements in the new version of the software. In OSS development, quality enhancements come from two sources: the core team and the community. On one hand, the OSS team is dedicated to the development. But most core teams are small and have limited resources (Raymond 1999). On the other hand, a much larger community works voluntarily for the project (Setia et al. 2012; Belenzo and Schankerman 2014). Therefore, we model that consumers derive utility from the quality improvement, the community contribution, and incur an adoption cost from updating the software:

$$U = \alpha_u + \theta \frac{Q}{n} + \gamma_u N_d - c_u,$$

where α_u is a consumer's valuation of the software, θ is the valuation of the new quality improvement in each release, N_d stands for the community contribution, γ_u is the valuation of community contribution, and c_u is the adoption cost of a consumer. As the release frequency increases, the quality enhancements from the core team becomes smaller in each release. One exception is when the project does not release any new version. In this case, we assume that $U = \alpha_u + \gamma_u N_d - c_u$.

Even though most of the OSS products are distributed for free, costs exist in adoption (Ellison and Fudenberg 2000). No matter whether a user is installing or upgrading, it takes time and effort to implement and learn the new version, especially for software that affects large organizations. If a new release is of limited scope and contains few quality enhancements, the adoption cost may outweigh the benefit.

We assume that α_u is uniformly distributed over $[0, 1]$ and $0 \leq \theta Q + \gamma_u \leq c_u < 1$. The second assumption states two constraints on our model. First, the cost of adoption is smaller than the highest valuation of the software from users. This constraint guarantees that even if there is neither new release nor community contribution, some consumers with high valuation of the software will still adopt. Second, the sum of quality improvement and community contribution is smaller than the adoption cost. This means if a consumer does not value the software at all, the new release and community contribution cannot make him/her adopt.

A consumer chooses to adopt the new version if $U > 0$. Then we can find the portion of consumers who adopt the new version: $N_u = 1 + \theta \frac{Q}{n} + \gamma_u N_d - c_u$. When $n = 0$, we would have $N_u = 1 + \gamma_u N_d - c_u$.

On the developer side, a community developer values the software itself, appreciates the users who adopt the new version, and incurs a development cost:

$$U_d = \alpha_d + \gamma_d N_u - n c_d,$$

where α_d captures the developer's inherent valuation of the OSS project, γ_d captures the developer's valuation of the user adoption N_u , and $n c_d$ captures the cost of development. When the project does not have any new release, we assume that $U_d = \alpha_d + \beta N_u - c_d$.

We also assume that α_d is uniformly distributed over $[0, 1]$, $0 < \gamma_d < c_d < 1$, and $\bar{N} c_d < 1$. Then we can obtain the portion of developers who decide to contribute by letting $U_d > 0$: $N_d = 1 + \gamma_d N_u - n c_d$. If $n = 0$, we can also obtain $N_d = 1 + \gamma_d N_u - c_d$.

Combining the two sides, we can obtain the equilibrium consumer adoption and community contribution: If $n = 0$, $N_u = \frac{1 - c_u + \gamma_u - c_d \gamma_u}{1 - \gamma_d \gamma_u}$ and $N_d = \frac{1 - c_d + \gamma_d - \gamma_d c_u}{1 - \gamma_d \gamma_u}$. If $n \in [1, \bar{N}]$, then

$$N_u = \frac{1 + \gamma_u - c_u}{1 - \gamma_d \gamma_u} - \frac{c_d \gamma_u}{1 - \gamma_d \gamma_u} \cdot n + \frac{\theta Q}{1 - \gamma_d \gamma_u} \cdot \frac{1}{n};$$

$$N_d = \frac{1 + \gamma_d - \gamma_d c_u}{1 - \gamma_d \gamma_u} - \frac{c_d}{1 - \gamma_d \gamma_u} \cdot n + \frac{\gamma_d \theta Q}{1 - \gamma_d \gamma_u} \cdot \frac{1}{n}.$$

Insofar we have analyzed the download and community contribution for each release. Since we assume that each release is the same, we can obtain the total download and community contribution by timing them together:

$$\begin{aligned}
\text{Download} &= n \cdot N_u = -\frac{c_d \gamma_u}{1 - \gamma_d \gamma_u} \cdot n^2 + \frac{1 + \gamma_u - c_u}{1 - \gamma_d \gamma_u} \cdot n + \frac{\theta Q}{1 - \gamma_d \gamma_u}; \\
\text{Contribution} &= n \cdot N_d = -\frac{c_d}{1 - \gamma_d \gamma_u} \cdot n^2 + \frac{1 + \gamma_d - \gamma_d c_u}{1 - \gamma_d \gamma_u} \cdot n + \frac{\gamma_d \theta Q}{1 - \gamma_d \gamma_u}.
\end{aligned}$$

We can see that both the download and community contribution have a curvilinear (inverse-U) relationship with the release frequency n . That is, there exist a positive value of n where the effect of release frequency is zero; before this point, release frequency has a positive effect on the dependent variables; after this point, release frequency has a negative effect. We also notice that this turning point is different for download and community contribution:

$$n_u^* = \frac{1 + \gamma_u - c_u}{2c_d \gamma_u} \quad \text{and} \quad n_d^* = \frac{1 + \gamma_d - \gamma_d c_u}{2c_d}.$$

Notice here we are not assuming the choice of release frequency from the project's perspective. Because there are many diverse motivations in OSS projects, it is not certain that the OSS project would want to maximize the download or the community contribution. Instead, our model and empirical results serve as a first exploration of the release frequency as a coordinating device between the two sides.

3.3.2 Hypotheses

Impact of Release Frequency on the Two Sides

We first explore how the download and community contribution are associated with release frequency. Taking derivatives of *Download* and *Contribution* with regard to n , we can see that higher release frequency could have two opposite effects for both the download and the community contribution. On one hand, it increases the

opportunities to interact with the consumers and the community. Each new version is an opportunity to gain consumer adoption and community contribution. Therefore, higher release frequency can potentially benefit the project. On the other hand, high release frequency could also hurt the project two ways. First, the community may need time to test the software, discover the bugs, and write the patches. Releasing too often may exhaust the community due to the cost of these activities. Second, given the total quality improvement, higher release frequency will lead to lower quality improvement per release, which will decrease the incentive of user adoption. Therefore, “release early, release often” may only work up to a certain threshold, and then begin to hurt the OSS project. We expect that a curvilinear (inverse-U) relationship exists between release frequency and download/community contribution, and that a moderate release frequency may benefit the OSS project the most.

Hypothesis 1A (H1A). *Release frequency has a curvilinear (inverse-U) relationship with download.*

Hypothesis 1B (H1B). *Release frequency has a curvilinear (inverse-U) relationship with community contribution.*

Besides the direct effect of release frequency on download and community contribution, we are also interested in the moderate effect of other parameters in the model, namely the adoption cost c_u , the development cost c_d , consumers’ valuation of community contribution γ_u , and the community developers’ valuation of adopting users γ_d . We call these four variables the environmental factors because they characterizes the environment that the OSS project operates in. Understanding the

moderate effects could provide implications for OSS projects in specific situations. We investigate their moderate effect in the rest of the section.

First, we can obtain that $\frac{\partial^2 \text{Download}}{\partial n \partial c_u} = -\frac{1}{1-\gamma_d \gamma_u} < 0$ and $\frac{\partial^2 \text{Contribution}}{\partial n \partial c_u} = -\frac{\gamma_d}{1-\gamma_d \gamma_u} < 0$, which means that higher adoption cost would decrease the effect of release

frequency on download and community contribution. This leads to our next hypotheses:

Hypothesis 2A (H2A). *Higher adoption cost will negatively moderate the effect of release frequency on download.*

Hypothesis 2B (H2B). *Higher adoption cost will negatively moderate the effect of release frequency on community contribution.*

The negative moderate effect of adoption cost on the consumer side is easy to understand. The consumers may weigh the benefit they gain from quality enhancements against the high cost. When the adoption cost is high, high release frequency may not work since it divides the quality improvement to tiny chunks. The negative moderate effect on the developer side comes from the cross externality. We use the audience of the project to test this hypothesis. If an OSS project is mainly for end users, the adoption cost may be low. Otherwise, if the software is for developers or enterprise, the cost of testing and upgrade the software is going to be much higher.

Regarding development cost c_d , we can obtain that $\frac{\partial^2 \text{Download}}{\partial n \partial c_d} = -\frac{2n\gamma_u}{1-\gamma_d \gamma_u} < 0$ and $\frac{\partial^2 \text{Contribution}}{\partial n \partial c_d} = -\frac{2n}{1-\gamma_d \gamma_u} < 0$, which means that higher development cost will negatively moderate the effect of release frequency. The negative moderate effect on the developer side comes directly from the developer's evaluation of benefit and cost of

contribution. While the negative moderate effect on the consumer side comes from the cross externality. We use the cumulative code committed to the repository to measure the development cost. The intuition is that the more code a project has, the higher cost it takes for a developer to contribute to the project.

Hypothesis 3A (H3A). *Higher development cost will negatively moderate the effect of release frequency on download.*

Hypothesis 3B (H3B). *Higher development cost will negatively moderate the effect of release frequency on community contribution.*

We now turn to the moderate effect of the cross externalities (γ_u and γ_d). We first evaluate the consumers' marginal valuation of the community contribution γ_u . As we can derive the partial derivatives: $\frac{\partial^2 \text{Download}}{\partial n \partial \gamma_u} = \frac{1 + \gamma_d - \gamma_d c_u - 2 n c_d}{(1 - \gamma_d \gamma_u)^2}$ and $\frac{\partial^2 \text{Contribution}}{\partial n \partial \gamma_u} = \frac{\beta(1 + \gamma_d - \gamma_d c_u - 2 n c_d)}{(1 - \gamma_d \gamma_u)^2}$. We can see that the moderate effect of γ_u is positive when $n < n_d^*$ and negative when $n > n_d^*$. Therefore, we propose:

Hypothesis 4A (H4A). *Higher consumer marginal benefit from community contribution will positively moderate the effect of release frequency on download when release frequency is relatively low and negatively moderate it when release frequency is relative high.*

Hypothesis 4B (H4B). *Higher consumer marginal benefit from community contribution will positively moderate the effect of community contribution on community contribution when release frequency is relatively low and negatively moderate it when release frequency is relative high.*

Since the value of the turning point n_d^* is an empirical question. It is possible that most observations in our sample is smaller than this turning point. Then we would observe a monotone moderate effect of consumer valuation of community. We use the number of fixed bugs to measure this marginal benefit from community contribution. The rationale is that the more bugs the project fixes, the more the consumers could benefit from the community contribution.

Regarding the externality of consumer adoption on the developers (γ_d), we can also obtain the partial derivatives: $\frac{\partial^2 \text{Download}}{\partial n \partial \gamma_d} = \frac{\gamma_u(1-c_u+\gamma_u-2n\gamma_u c_d)}{(1-\gamma_d\gamma_u)^2}$ and $\frac{\partial^2 \text{Contribution}}{\partial n \partial \gamma_d} = \frac{1-c_u+\gamma_u-2n\gamma_u c_d}{(1-\gamma_d\gamma_u)^2}$. We can see that the moderate effect of γ is positive when $n < n_u^*$ and negative when $n > n_u^*$. Therefore, we propose:

Hypothesis 5A (H5A). *Higher developer marginal benefit from consumer adoption will positively moderate the effect of release frequency on download when release frequency is relatively low and negatively moderate it when release frequency is relative high.*

Hypothesis 5B (H5B). *Higher developer marginal benefit from consumer adoption will positively moderate the effect of release frequency on community contribution when release frequency is relatively low and negatively moderate it when release frequency is relative high.*

Similar to the moderate effect of the consumer valuation of community contribution, the externality of consumer adoption on community contribution may also show as monotone empirically. We test H5A and H5B with the license restrictiveness of an OSS project. OSS projects are typically distributed under various types of OSS

licenses, which have different restrictions on the use and modification of the software. A highly *restrictive license* such as GNU General Public License (GPL) requires all derived works (even if they just use the project instead of modifying it) to follow the same license (i.e., contribute back to the community). A *permissive license* (such as the Berkeley Software Distribution, BSD) has fewer restrictions on the community to use or modify the work (Laurent 2004). Colazo and Fang (2009) find that restrictive projects attract more voluntary contributors. Stewart et al. (2006) find that license restrictiveness matters for organizational sponsorship on developer activities. Recently, Belenzon and Shankerman (2014) find that developers sort strongly by license types, and suggest that the developers in projects with restrictive licenses are more likely motivated by intrinsic motivations. We build on this literature to examine the subtle effect of incentives in managing OSS release policy. Since developers in restrictive projects mainly contribute for intrinsic motivations (Belenzon and Schankerman 2014), the developers in projects with restrictive licenses would be less likely to respond to release frequency changes.

Motivation and Assimilation of Contribution

We further examine the effect of community motivations with the contributions from anonymous users. Anonymous contributors are likely to be intrinsically motivated, because they do not provide their identity to receive the extrinsic rewards (Belenzon and Schankerman 2014). Hence, anonymous contributions are less likely to associate with fast release frequency. In this case, the portion of anonymous contributions may not have a significant association with release frequency. Another possibility could be that the community is suspicious about frequent release because the contributors fear that the

team is exploiting their free contributions. If so, the community contribution may have a negative correlation with release frequency. We further test these two possibilities using anonymous contributions in our data.

Our model assumes away the cost of incorporating community contribution and preparing for new releases. For any community contribution such as bug reports and patches, the team has to spend time and effort to test and incorporate them. Since the core team usually has limited resources, as the release frequency increases, the core team would be stretched too thin and have inadequate time to absorb the community contributions into the project (Cohen and Levinthal 1990). Further, August and Niculescu (2013) show that the firm may benefit from releasing later if the cost of processing user error reports is too large, which is the case in our context of a small team facing a large amount of community contributions. Given the number of contributions, the core team may only have the time and effort to process a portion of them.

Hypothesis 6 (H6). The portion of community contribution that is processed by the team would decrease as release frequency increases.

3.4 Data

To test the above hypotheses, we collected data from the open source repository SourceForge.net (SourceForge). Since its inception in 1999, SourceForge has been a major platform for OSS development and distribution, as well as an important data source for OSS studies (see e.g., Lerner and Tirole 2005; Singh et al. 2011; Belenzon and Schankerman 2014). We collected the data on release history and other characteristics of

157,720 projects available on SourceForge from 1999 to 2010. The release information is a unique feature of our dataset, which enables us to study the impact of release frequency on adoption and community contributions in the co-production process of OSS development.

A large percentage of projects on SourceForge were created but had no activities (Chengalur-Smith and Sidorova 2003). In our data set, 83,171 projects (out of 157,720 projects) have at least one download at the time of data collection. This is still a large set of projects. Among them, we further narrow down to popular projects with more cumulative downloads than the average (52,704) of all projects, which gives us 3,874 projects in the sample. Because projects enter SourceForge at different times, we constructed an unbalanced quarterly panel for all the projects. Table 3.1 provides variable definitions and descriptive statistics. We transform all count variables in our data using natural logarithm to reduce the impact of extreme observations¹.

3.4.1 Key Variables

Our main interest is the effect of release frequency, which is represented by $releases_{it}$ in our quarterly panel. Higher values of release frequency indicate that the project is more likely adopting the “release early, release often” strategy. We also include the quadratic term of release frequency as $releases_{it}^2$ to test the possible curvilinear relationship.

On the consumer side, we use $download_{it}$ to measure the adoption, which is the log-transformed number of download for project i in quarter t . Since many companies

¹ Since count variables in our data could be zero, we apply $\log(x+1)$ to the variable x .

profit from OSS with complementary products and services (Watson et al. 2008), larger download may be related to higher potential profits.

On the developer side, we use the patches contributed to project j in quarter t as the measure for community contribution. More than 95% of the community interactions in SourceForge fall into one of the four types: patches, bug reports, feature requests, and support requests. Among them, bug reports and patches are argued to be highly efficient because the community has access to the source code and can help locate a bug to specific code lines (Raymond 1999). We choose patches as the measure for two reasons. First, new bugs may be introduced when a project speeds up release frequency (Ji et al. 2005). It is not clear whether the contribution comes from the incentive of the community or the decrease of quality. Second, the cost of writing a patch is much higher than reporting a bug. Therefore patches may be a more meaningful measure for community contribution.

After a patch is submitted, the core team can either accept the patch and incorporate it into the code or mark it as some other status, such as “Fixed” or “Invalid”. Meanwhile, some patches do not have any responses and the resolutions are marked as “None” or nonresponse. We construct a variable $nores_ratio_{jt}$, which is the ratio of nonresponse patches over the total number of patches, designed to measure the absorption of community contributions by the core team. When this variable increases, it means that the project absorbs fewer of the patches from the community. SourceForge also reveals the identity information of the submitter of the patches. Usually the patch contributor would be a registered user on SourceForge. However, sometimes an anonymous user submits a patch to a project. This turns out to be a useful feature of the

data. Those developers cannot be driven by career concerns because they do not reveal their identity and hence cannot benefit from signaling their ability (Belenzon and Schankerman 2014). This separates them apart from those who may gain benefit by contributing. Such anonymous contributors are marked as “Nobody” on SourceForge. We construct a variable $nobody_ratio_{it}$ to capture the portion of patches from anonymous users.

3.4.2 Moderate and Control Variables

Quality Enhancement: We need to control for quality enhancement by the core team in our analysis (Q in our model). We use the number of code modification ($code_{it}$, i.e., commits; the number of times the source code was modified) to operationalize quality enhancement (Colazo and Fang 2009).

Adoption Cost: Our data contains a project’s intended audience, which could be end users, developers, system administrators, or others (Lerner and Tirole 2005b). We use the audience as an indicator of adoption cost ($user_i = 1$ if the audience of project i is end users, and 0 otherwise). End-user oriented programs usually have lower adoption cost than projects for developers and system administrators because the latter two may have more interdependencies with other software. Hence, the adoption cost (c_u in our model) is lower when a project is end-user oriented.

Development Cost: We use the cumulative code commits of project i up to quarter t as a measure for the development cost ($cumcode_{it}$; c_d in our model). As the cumulated code of a project increases, the cost for a developer to find and fix a bug should increase because of the complexity brought by the amount of source code.

License Restrictiveness: We use the license restrictiveness to measure the developer's benefit from user adoption (γ_d in our model). The rationale is that the ability to profit from an OSS project should decrease if the license is more restrictive (Lerner and Tirole 2005). Following the literature (Lerner and Tirole 2005), we construct a binary variable hr_i to indicate whether project i has a highly restrictive license such as GPL.

Bugs Fixed: We use the number of bugs fixed to measure a user's marginal benefit from community contribution ($bugsfixed_{it}$; γ_u in our model). Bugs reported by the community means nothing to the consumer unless it is fixed by the team or other community developers. Thus, higher number of bugs fixed indicates that the consumers could benefit more from the community contribution.

Product Diffusion and Network Effects: Besides the effect of release frequency, we expect that product diffusion and network effects may also influence the market share of an OSS project. The product diffusion literature finds that adoption is influenced by the current user base and the number of potential users (e.g. Bass 1969). Network effects indicate that the value of a product increases with the size of the network. Following Duan et al. (2009), we control the network effect and product diffusion by the cumulative download ($cum_{i,t-1}$) and days since a project registered on SourceForge ($age_{i,t-1}$) before quarter t .

3.5 Empirical Analysis and Results

We use the following specifications to test our hypotheses on the effect of release frequency on download and community contribution:

$$Y_{it} = \alpha_i + \beta_1 \text{releases}_{it} + \beta_2 \text{releases}_{it}^2 + \beta_3 \text{releases}_{it} X_{it} + Z'_{it} \xi + \varepsilon_{it} . \quad (3.1)$$

The dependent variable Y_{it} is download_{it} and patches_{it} in two regressions, respectively. Then the coefficients β_1 and β_2 could capture the curvilinear relationship if $\beta_1 > 0$ and $\beta_2 < 0$. X_{it} is one of the moderator variables (user_i , cumcode_{it} , hr_i , and bugsfixed_{it}). β_3 captures the moderate effect of X_{it} . Z_{it} is a vector of control variables that contains $\text{cum}_{i,t-1}$, $\text{age}_{i,t-1}$, $\text{age}_{i,t-1}^2$, and code_{it} . And ε_{it} is a random error.

We then estimate the model with fixed-effect (FE) panel data specification. We use FE rather than random effect (RE) because FE allows arbitrary correlation between α_i and the explanation variables, while RE does not (Wooldridge 2010, p. 252). Hausman-Wu test (see Wooldridge 2010) shows that FE models are indeed preferred for the current data set. Note that because α_i absorbs any time-invariant characteristics of a project such as license type, intended audience, etc., their individual direct effect on the download and community contribution could not be identified. However, we examine their interactions with release frequency to identify their moderate effects.

3.5.1 Effect of Release Frequency in Adoption

We first examine how the release frequency of an OSS project is associated with its download. Table 3.2 presents the estimated parameters from the adoption regression. The dependent variable is download_{it} . The regression also includes a set of quarter dummy variables, whose coefficients are omitted for brevity. The model-fit indices are shown in the bottom rows. We focus on the coefficients of releases (subscripts are omitted for brevity in discussion) and its interaction terms to test each hypothesis.

In Model (1), we test the effect of release frequency on download by just controlling the network effect and product diffusion. The significant positive coefficient ($p < 0.01$) of *releases* confirms the wisdom of “release early, release often”. However, it is not that the more often the better. The negative coefficient ($p < 0.01$) of the quadratic term *releases*² supports our H1A. That is, there exists a curvilinear relationship between release frequency and download. With the estimation in Model (1), we can calculate the threshold $n_u^* = \exp(\text{releases}^*) \approx \exp(4.5) \approx 90$. This number is large in that 99% of the release frequencies fall below it. Therefore, the core team probably should push the release to the limit if they care only about download.

We also find significant coefficients of the control variables (*code*, *cum* and *age*). The signs of the coefficients are as expected. Higher quality enhancement from the team increases the download. Download also increases with the install base and decrease as the project becomes older.

We then analyze the moderate effects with interaction terms. In Model (2), the moderate effect of *user* is positive and significant ($p < 0.01$), which suggests that faster release may work better when the adoption cost is low. In other words, when the adoption cost is high, the effect of release frequency on download is lower. This supports our H2A. Model (3) tests the moderate effect of development cost. The coefficient is negative and significant ($p < 0.01$), which means that the higher the development cost, the lower the effect of release frequency on download. Therefore, our H3A is supported as well.

We find a significant positive moderate effect of *bugsfixed* in Model (4). This moderate effect suggests that when consumers can gain more value out of community contribution, the positive effect of release frequency is higher. H4A is supported. This

means that in project with higher capacity to convert community contribution to quality enhancement, release early and often may work better.

Model (5) tests the moderate effect of license restrictiveness. We do not find a statistically significant effect. Therefore, H5A is not supported. This could be due to several reasons. First, it could be that since γ_d is very small, the effect may not be statistically detectable. Second, it could be because that developers do not value the new adoption that much. Third, it could be that license restrictiveness is not a good measure for the developer's valuation of consumer adoption. This leaves interesting directions for future exploration.

3.5.2 Effect of Release Frequency on Community Contribution

We now examine the coordinating role of release frequency in the OSS co-development process. The results are presented in Table 3.3. In all models, *patches* is the dependent variable. Our H1B about the curvilinear relationship between release frequency and community contribution is supported in Model (1) with significant ($p < 0.01$) positive coefficient of *releases* and negative coefficient of *releases*². As release frequency increases, the community may have contributed more because their efforts are visible to the users in a shorter time. However, as release frequency becomes too high, the community may have difficulties to keep up, and therefore the contribution begin to decrease. We can calculate the threshold $n_d^* = \exp(\text{releases}^*) \approx \exp(2.2) \approx 9$, which is much smaller than the threshold n_u^* on the consumer side. Therefore, if an OSS project

increases the release frequency, the community contribution may have been first exhausted.

Column (2) in Table 3.3 presents the results related to the moderate effect of adoption cost. We do not find a significant effect of the adoption cost. Therefore, H2B is not supported. The reason can be that the adoption cost affects the community contribution through consumer adoption. If the developers do not value the consumer adoption much, then this may not be statistically detectable. In contrast, we find a negative moderate effect ($p < 0.05$) of development cost in Model (3), which supports our H3B. That is, as the development cost increases, the effect of release frequency on community contribution would decrease.

We find a significant positive moderate effect for *bugsfixed* in Model (5). Hence, given the relatively high threshold, we do see that for major part of the release frequency spectrum, the moderate effect of *bugsfixed* is positive. Similar to the download results, we do not find a significant moderate effect of license restrictiveness in Model (5). Therefore, H5B is not supported.

3.5.3 Motivation and Absorption of Contribution

To double check the effect of contributor incentives, we examine the number and ratio of patches from anonymous users as the dependent variables in column (1) and (2) of Table 3.4. Because neither coefficients of *releases* and *releases*² are significant, we cannot reject the hypothesis that the effect of releases on anonymous contributions is zero. This could mean that releases may only work when the contributors of an OSS

project have certain extrinsic motivations.² This may also explain why the moderate effect of license restrictiveness is not significant. Therefore, for users who contribute purely out of intrinsic motivations, release frequency may not be a good lever.

Column (3) and (4) of Table 3.4 present the results when we use the number and ratio of none-resolved patches as the dependent variables. It seems that as the release frequency increases, the ratio of unresolved patches also increases. Our H6 is supported. This means even though the number of patches increase, the portion that the core team can manage actually decreases. Our interpretation is that this provides evidence about the absorptive capacity of the team. Even though the number of patches increases as release frequency increases, the team may not have enough capacity to absorb all patches into the source code and incorporate the quality improvement. The implication is that for teams with limited resources, releasing too fast may not be a good idea.

3.5.4 Robustness Checks

We performed several tests on our measures and specifications to check the robustness of our results. First, we checked the robustness of our model using alternative measures by replacing download numbers with download market share. We also used bug reports as the measure of community contributions. The results are qualitatively consistent with those reported in Tables 3.2 and 3.3.

Next, we varied the time unit t of our panel from a quarter to a month and a year and reconstructed the whole sample. We estimated our models on the new monthly and yearly panel data and found that most of our results remained unchanged qualitatively.

² Notice that only a small portion of the projects have anonymous contributions. Therefore we should be cautious about the interpretation of this result.

But as the length of our time period increases, the release frequency that benefits the download the most also increases.

Finally, we re-ran our analysis on several subsamples and with different measures to ensure that our results are not driven by several projects with extreme release patterns. We also classified the releases by their version number into big and small releases, and then constructed our release frequency measure using only big releases. The results hold qualitatively and the effect of release frequency is stronger when we use number of big releases as the measure of release frequency. In some specifications the negative coefficient of *releases*² becomes insignificant. This is reasonable because a major release may contain many quality enhancements. Therefore, the curvilinear relationship may not appear with major releases.

3.6 Discussion

3.6.1 Major Findings

Despite growing interests in community-based open innovation, the existing research focuses mainly on the “why” of open source software, and generally ignored the “how” of OSS management, especially on how to manage the dynamic collaboration between the OSS team and the community. Motivated to narrow down this gap in the literature, we test the effect of release frequency on download and community contribution of OSS projects with a longitudinal data set, and investigated the moderate effects of environmental factors in this process. Our empirical results show the nonlinear

role of release frequency in shaping both the development and adoption processes in such open models of co-production.

Existing literature and common wisdom in the OSS community both indicate that frequent releases would benefit an OSS project. While we find that release frequency is positively associated with download and community contribution, taking it too far may backfire and hurt the project. We analyze the co-creation process of OSS product to underscore the reasons from the supply side. Three reasons may be behind this result. First, as release frequency increases, the quality enhancements within each new version may decrease. This may decrease consumers' incentive to adopt. And this may be passed to the developer side because of the cross externality. Second, as the core team speeds up the release frequency, the community might not be able to keep up with the pace. We indeed find that release frequency has a curvilinear relationship with community contribution. That is, releasing too fast may actually exhaust the community. Third, as the release frequency increases, the project team may have less time to incorporate the community contributions. Hence, the quality that the project absorbs from the community contributions actually decreases, which leads to lower market share. Therefore, releasing too fast may backfire.

Several environmental factors in the co-creation process may moderate the influence of release frequency. First, higher adoption cost may attenuate the effect of release frequency. Our results show that higher release frequency works better for end-user oriented projects where the adoption cost is assumed to be lower than developer or system administrator oriented projects. This attenuation effect mainly happens in the adoption process. Second, higher development cost may decrease the effect of release

frequency. We find this negative moderate effect in both download and community contribution. The major driver may be the cost for developers to examine the code and write the patch. But the consumers are also impacted due to the cross externality.

Third, we find that the more the consumers could benefit from community contribution, the better higher release frequency works. The community could contribute by reporting many bugs. But if none of them are fixed, the consumers may not benefit much. Our results show that when there are more bugs fixed, faster release could work better.

Fourth, we examine the role of OSS licenses type to examine the motivations in the co-development process. We do not find different effect of release frequency for projects with different license schemes. When the license is highly restrictive (e.g. GPL), community contributions may mainly come from developers who are intrinsically motivated. Since releasing frequently mainly satisfies the extrinsic motivation of signaling by delivering the work quickly to the relevant audience, fast release works less effectively in restrictive projects. We further confirm our results using ratio of patches from anonymous users, who may only contribute for intrinsic motivations. We find that release frequency may not work very effectively for these users.

Lastly, we also find that high release frequency could decrease the core team's absorption of community contribution. Our results show that as release frequency increases, the portion of un-resolved patches also increases, indicating that even the team itself may not be able to keep up with the fast release frequency.

3.6.2 Managerial Implications

Our study provides several useful implications for OSS project managers. First to ensure smooth development and successful adoption, the project manager should pay attention to the decisions on release frequency. It is wise to follow the motto “release early, release often” in OSS when the project team co-creates the products with the community. Releasing frequently could help deliver quality enhancements to consumers and obtain the feedback and contribution from the community fast, which demonstrates the Linus’s Law - “given enough eyeballs, all bugs are shallow”. However, taking it too far by releasing too frequently might backfire. This message is clear from our results.

With regard to the co-creation process, the project manager should decide the proper release frequency according to the goal of the core team, the type and size of the project, and the capability of the team and the community. First, the curvilinear relationships on two sides of the OSS project are different. Namely, the turning point where release frequency begins to have a negative effect is much lower on the developer side than that on the consumer side. This means that the community contribution may have been exhausted before the release frequency reaches the threshold on the consumer side. This may have important implications for the core team. If the goal is mainly to grow the install base, then releasing fast may be the choice. But if the goal is to obtain quality contribution from the community, then slower release frequency may work better with the community.

Second, the type and size of a project may influence the effect of release frequency. In projects that have high adoption cost, “release early, release often” may work less effectively. For example, for enterprise software that involves many other

systems, slower but steadier release strategy may work better. The OSS manager also needs to consider the complexity of the project. For a relatively mature project which has a large code base, it might be better to slow down a little bit even though the team still has many new quality enhancements. As the code becomes more complex, the cost for the community to detect and fix a bug also increases. This development cost makes fast release less effective.

Third, when a project has a small team or a small community, releasing too fast may not benefit the project. Misalignment of release frequency and community contributions might then affect the adoption of the project. The core team has to attract enough eyeballs from the community to shoot all the bugs. The team also needs to have enough resources to incorporate all the community contribution. Otherwise, the effort of releasing early and often may be wasted or even have a negative effect.

3.7 Concluding Remarks

As open source software gradually moves into the mainstream in software development, it becomes increasingly important to understand how to manage “open” models of product co-creation. Drawing upon theoretical perspectives of software releasing, two-sided markets, and open community, we develop an integrated model to examine the influence of OSS release frequency on download and community contribution. Our empirical results identify the curvilinear relationship between release frequency and download, and reveal effects of various environmental factors in shaping this relationship.

Our study examines the sequential product release problem in the OSS context and extends the software release literature and the NPD literature into the open model of co-creation. In our setting, the quality enhancements are controlled and the price is fixed. We add to the literature by empirically showing that the curvilinear relationship between release frequency and download due to the adoption cost. More importantly, we extend the literature to a collaboration environment where the team develops the product together with the community. Our results show that in such settings managers should also take into account the capacity of the community when they make the product release decisions.

We also add to the emerging literature on open-source innovation by analyzing how release frequency, a factor that the project managers can control, influences its download and community contribution. We show how the capability and motivations of the community matter. Our study considers the dynamic OSS co-production process, and provides insights on how to use release frequency as a coordination device in the co-creation process. Releasing too fast may weaken the positive impact of community contributions due to exhausted community contribution and absorptive capacity of the core team.

More broadly, this research is related to open models of co-production and its content, coordination, and adoption in user communities over time. In such an environment, technology-enabled platforms facilitate open collaboration and co-creation among distributed users and producers, without relying on traditional long-term employment or formal organizational affiliations. Meanwhile, new challenges such as product releases, community incentives, and intellectual property need to be managed so that the open community of co-production can be sustainable in the marketplace.

Our work reported in this paper, though it is examined in the OSS context, may help to understand these larger issues. If so, we hope that our work will stimulate more research in this emerging area.

Table 3.1: Variables and Descriptive Statistics

Variable	Description and Measure	Mean	Std. Dev	Min	Max
$download_{it}$	log-transformed download	7.674	2.139	0	17.102
$patches_{it}$	log-transformed patches	0.117	0.390	0	4.615
$releases_{it}$	log-transformed number of releases	0.289	0.574	0	5.112
$releases_{it}^2$	squared term of r_{it}	0.413	1.163	0	26.132
$code_{it}$	log-transformed number of code commits	0.898	1.977	0	12.134
$user_i$	=1 if project i is user oriented	0.585	0.493	0	1.000
$cumcode_{it}$	log-transformed cumulative commits	0.898	1.977	0	12.134
hr_i	=1 if project i has a highly restrictive license	0.676	0.468	0	1.000
$bugsfixed_{it}$	log-transformed number of bug fixes	0.143	0.433	0	4.060
cum_{it}	log-transformed cumulative download	10.298	2.554	0	19.308
age_{it}	log-transformed days since the registration	6.999	1.028	0	8.299
age_{it}^2	squared term of age_{it}	50.047	12.693	0	68.866
$patches_nobody_{it}$	log-transformed number of patches by anonymous users	0.006	0.076	0	4.543
$nobody_ratio_{it}$	ratio of patches by anonymous users	0.004	0.059	0	1.000
$patches_nores_{it}$	log-transformed number of patches that are not resolved	0.019	0.148	0	4.543
$nores_ratio_{it}$	ratio of patches with no resolutions	0.016	0.122	0	1.000

Subscripts i stands for project i , t stands for quarter t . $N = 3,874$. $Obs = 89,372$.

Table 3.2: Impact of Release Frequency on Download

Dependent Variable	<i>download</i>				
	(1)	(2)	(3)	(4)	(5)
<i>releases</i>	.573***	.523***	.604***	.562***	.562***
<i>releases</i> ²	-.063***	-.065***	-.053***	-.065***	-.063***
<i>code</i>	.075***	.074***	.091***	.074***	.075***
<i>L.cum</i>	.733***	.733***	.732***	.733***	.733***
<i>L.age</i>	-1.058***	-1.057***	-1.064***	-1.058***	-1.058***
<i>L.age</i> ²	.040***	.040***	.041***	.040***	.040***
<i>releases x user</i>		.090***			
<i>releases x cumcode</i>			-.030***		
<i>releases x bugsfixed</i>				.039***	
<i>releases x hr</i>					.016
Constant	3.703***	3.700***	3.686***	3.705***	3.703***
Quarter dummies			Included		
Project fixed effect			Included		
Adjusted R ²	0.620	0.620	0.620	0.619	0.619
Projects	3874	3874	3874	3874	3874
Observations	89372	89372	89372	89372	89372

* p<0.10 ** p<0.05 *** p<0.01

Table 3.3: Impact of Release Frequency on Community Contribution

Dependent Variable	<i>patches</i>				
	(1)	(2)	(3)	(4)	(5)
<i>releases</i>	.070***	.070***	.074***	.065***	.053***
<i>releases</i> ²	-.016***	-.016***	-.015***	-.016***	-.018***
<i>code</i>	.022***	.022***	.024***	.022***	.021***
<i>L.cum</i>	.015***	.015***	.014***	.015***	.014***
<i>L.age</i>	-.006	-.006	-.007	-.006	-.006
<i>L.age</i> ²	-5.1e-04	-5.1e-04	-3.6e-04	-5.0e-04	-4.8e-04
<i>releases x user</i>		-6.6e-04			
<i>releases x cumcode</i>			-.004**		
<i>releases x hr</i>				.006	
<i>releases x bugsfixed</i>					.055***
Constant	-.059**	-.059**	-.062**	-.060**	-.058*
Quarter dummies			Included		
Project fixed effect			Included		
Adjusted R ²	0.027	0.027	0.027	0.027	0.031
Projects	3874	3874	3874	3874	3874
Observations	89372	89372	89372	89372	89372

* p<0.10 ** p<0.05 *** p<0.01

Table 3.4: Impact of Release Frequency on Anonymous and Non-Resolved Patches

	(1)	(2)	(3)	(4)
Dependent Variable	patches_nobody	nobody_ratio	patches_nores	nores_ratio
<i>releases</i>	.016	.004	.052**	.032*
<i>releases</i> ²	.007	.006	-.008	-.005
<i>code</i>	.013***	.006***	.022***	.012***
<i>L.cum</i>	.010**	.007***	.021***	.018***
<i>L.age</i>	.058	.043*	.002	.02
<i>L.age</i> ²	-.008*	-.006**	-.003	-.004
Constant	-.160	-.120*	-.036	-.065
Quarter dummies		Included		
Project fixed effect		Included		
Adjusted R ²	0.030	0.017	0.029	0.021
Projects	299	299	515	515
Observations	7422	7422	12620	12620

* p<0.10 ** p<0.05 *** p<0.01

References

- Agarwal, R., and Bayus, B. L. 2002. The Market Evolution and Sales Takeoff of Product Innovations. *Management Science* **48**(8) 1024–1041.
- Arora, A., Caulkins, J. P., and Telang, R. 2006. Sell First, Fix Later: Impact of Patching on Software Quality. *Management Science* **52**(3) 465–471.
- August, T., and Niculescu, M. F. 2013. The Influence of Software Process Maturity and Customer Error Reporting on Software Release and Pricing. *Management Science* **59**(12) 2702–2726.
- Banker, R. D., Gordon B. Davis, and Slaughter, S. A. 1998. Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study. *Management Science* **44**(4) 433–450.
- Banker, R. D., and Slaughter, S. A. 1997. A Field Study of Scale Economies in Software Maintenance. *Management Science* **43**(12) 1709–1725.
- Bass, F. M. 1969. A New Product Growth for Model Consumer Durables. *Management Science* **15**(5) 215–227.
- Bayus, B. L. 2013. Crowdsourcing New Product Ideas over Time: An Analysis of the Dell IdeaStorm Community. *Management Science* **59**(1) 226–244.
- Belenzon, S., and Schankerman, M. 2014. Motivation and Sorting of Human Capital in Open Innovation. *Strategic Management Journal* to appear.
- Boudreau, K. J., and Lakhani, K. R. 2009, July 1. How to Manage Outside Innovation. *MIT Sloan Management Review* **50**(4) 69–73.
- Chengalur-Smith, S., and Sidorova, A. 2003. Survival of open-source projects: A population ecology perspective. *ICIS International Conference on Information Systems*.
- Cohen, M. A., Eliashberg, J., and Ho, T.-H. 1996. New Product Development: The Performance and Time-to-Market Tradeoff. *Management Science* **42**(2) 173–186.

- Cohen, W. M., and Levinthal, D. A. 1990. Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly* **35**(1) 128–152.
- Colazo, J., and Fang, Y. 2009. Impact of license choice on Open Source Software development activity. *Journal of the American Society for Information Science and Technology* **60**(5) 997–1011.
- Crowston, K., Howison, J., and Annabi, H. 2006. Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice* **11**(2) 123–148.
- Duan, W., Gu, B., and Whinston, A. B. 2009. Informational Cascades and Software Adoption on the Internet: An Empirical Investigation. *Management Information Systems Quarterly* **33**(1) 23–48.
- Economides, N., and Katsamakos, E. 2006. Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry. *MANAGEMENT SCIENCE* **52**(7) 1057.
- Ellison, G., and Fudenberg, D. 2000. The Neo-Luddite’s Lament: Excessive Upgrades in the Software Industry. *The RAND Journal of Economics* **31**(2) 253–272.
- Fogel, K. 2005. *Producing Open Source Software: How to Run a Successful Free Software Project*, O’Reilly Media.
- Ghose, A., and Han, S. P. 2014. Estimating Demand for Mobile Applications in the New Economy. *Management Science* **60**(6) 1470–1488.
- Grewal, R., Lilien, G. L., and Mallapragada, G. 2006. Location, location, location: How network embeddedness affects project success in open source systems. *Management Science* **52**(7) 1043–1056.
- Harter, D. E., Krishnan, M. S., and Slaughter, S. A. 2000. Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. *Management Science* **46**(4) 451–466.
- Harter, D. E., and Slaughter, S. A. 2003. Quality Improvement and Infrastructure Activity Costs in Software Development: A Longitudinal Analysis. *Management Science* **49**(6) 784–800.
- Von Hippel, E. 2001. Innovation by user communities: Learning from open-source software. *Mit Sloan Management Review* **42**(4) 82–86.
- Von Hippel, E., and von Krogh, G. 2003. Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. *Organization Science* **14**(2) 209–223.

- Jiang, Z., Sarkar, S., and Jacob, V. S. 2011. Postrelease Testing and Software Release Policy for Enterprise-Level Systems. *Information Systems Research* **23**(3) 635–657.
- Ji, Y., Mookerjee, V. S., and Sethi, S. P. 2005. Optimal Software Development: A Control Theoretic Approach. *Information Systems Research* **16**(3) 292–306.
- Kornish, L. J. 2001. Pricing for a Durable-Goods Monopolist Under Rapid Sequential Innovation. *Management Science* **47**(11) 1552–1561.
- Kornish, L. J., and Ulrich, K. T. 2014. The Importance of the Raw Idea in Innovation: Testing the Sow’s Ear Hypothesis. *Journal of Marketing Research* **51**(1) 14–26.
- Von Krogh, G., and von Hippel, E. 2006. The promise of research on open source software. *Management Science* **52**(7) 975–983.
- Laurent, A. M. S. 2004. *Understanding Open Source and Free Software Licensing*, O’Reilly Media, Inc.
- Lerner, J., and Tirole, J. 2002. Some Simple Economics of Open Source. *The Journal of Industrial Economics* **50**(2) 197–234.
- Lerner, J., and Tirole, J. 2005. The Scope of Open Source Licensing. *J Law Econ Organ* **21**(1) 20–56.
- Michlmayr, M. 2007. *Quality Improvement in Volunteer Free and Open Source Software Projects – Exploring the Impact of Release Management* PhD thesis, University of Cambridge.
- Parker, G. G., and Alstyne, M. W. V. 2005. Two-Sided Network Effects: A Theory of Information Product Design. *Management Science* **51**(10) 1494–1504.
- Ramachandran, K., and Krishnan, V. 2008. Design Architecture and Introduction Timing for Rapidly Improving Industrial Products. *Manufacturing & Service Operations Management* **10**(1) 149–171.
- Raymond, E. 1999. The cathedral and the bazaar. *Knowledge, Technology & Policy* **12** 23–49.
- Roberts, J. A., Hann, I. H., and Slaughter, S. A. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science* **52**(7) 984–999.
- Rochet, J.-C., and Tirole, J. 2006. Two-Sided Markets: A Progress Report. *The RAND Journal of Economics* **37**(3) 645–667.

- Singh, P. V., Tan, Y., and Mookerjee, V. 2011. Network Effects: The Influence of Structural Capital on Open Source Project Success. *Management Information Systems Quarterly* **35**(4) 813–829.
- Stewart, K. J., Ammeter, A. P., and Maruping, L. M. 2006. Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects. *Information Systems Research* **17**(2) 126–144.
- Watson, R. T., Boudreau, M.-C., York, P. T., Greiner, M. E., and Wynn, D., Jr. 2008. The Business of Open Source. *Commun. ACM* **51**(4) 41–46.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*, (second edition,) The MIT Press.

Chapter 3 is currently under review for publication of the material. Chen, Wei, Vish Krishnan, and Kevin Zhu. The dissertation author was the primary investigator and author of this paper.