[DISSERTATION TITLE]

by

[Full Name]

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
([MAJOR])

August 20XX

Dedication

To everyone.

Acknowledgements

Thank you to everyone.

Table of Contents

List of Tables

List of Figures

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Chapter 1

Introduction

ch:introduction

Chapter 2

Quid Pro Code

ch:pub-goods

2.1 Introduction

sec:intro

Open Source Software (OSS) projects are public information goods produced through incremental efforts of individual contributors.[*†] Interested parties can freely download software code for their own use and can also propose contributions to the original maintainer of the project[‡]. The very existence of OSS rebukes conventional wisdom on privately produced public goods[§] and various explanations have been offered to rationalize their provision, from signalling (Lerner and Tirole, 2002), (impure) altruism (Andreoni, 1990), need satiation (Athey and Ellison, 2014), and institutional structures imposed by self-organizing local communities (Ostrom, 1990; Benkler, 2002). In this study we examine an alternative channel through which widespread contribution to public OSS projects may be achieved: peer effects. Peer behavior can potentially affect the net returns to public

[*]Our use of the term "open source" requires some definition. In a general sense, OSS is a computer technology for which the underlying source code is made publicly available under a license permitting use, modification, and subsequent redistribution of derived products (Open Source Initiative, 2007). While there are many variations on the specifics of this definition, the most important feature of software projects considered in this study is that (1) they are distributed under some permissive OSS license (GitHub, Inc., 2022) and (2) they are *collaborative* projects that allow for modifications to be submitted from a contributor base wider than the original developer.

[†]Throughout this paper, we will use the terms "contributor", "developer", "individual", and "agent" interchangeably in reference to the population of study.

[‡]For example, a user may wish to propose a new feature or fix a software fault (i.e. a "bug").

[§]Since contribution is costly, agents choose their contribution levels both with respect to private benefits of contribution as well as the level of the OSS public good delivered by the efforts of their peers. If the net benefit of contribution is negative, an individual may simply opt to free-ride on the efforts of others, leading to misallocation of contribution away from an efficient equilibrium.

good contribution through various channels, improving returns and ameliorating contribution costs. Can peer influence drive heterogeneity in preferences and contribution costs, effectively subsidizing the private provision of public goods?

Consider the quandary faced by maintainers of OSS projects.[¶] Sindre Sorhus is a superstar OSS contributor. As of December 2021, he works on OSS full-time and is the author and primary maintainer of over 1,000 OSS projects (Sorhus, 2021). As the project maintainer of these many OSS projects, Sorhus interacts with the wider community of OSS contributors and has personally reviewed tens of thousands of proposed contributions to his projects. Sorhus once reflected that "$\sim$ 80% of contributors doesn't [*sic*] know how to resolve a merge conflict, almost no one writes a good pull request titles, $\sim$ 30% don't run tests locally before submitting a [pull request]", and "$\sim$ 40% don't include docs/tests" (Sorhus, 2019). In essence, Sorhus's concern centers around the lack of quality project contributions from his peers. Software development in general is a complex, ever-changing process and many potential contributors simply may lack the skills to contribute effectively. As opposed to shouldering the entire burden of OSS project development[‖], to what extent can the contributions efforts of skilled contributors like Sorhus actually improve the productivity of their peers?[**]

A key difference with OSS projects and other public goods is that production of OSS generates both a community of contributors and a set of auxiliary information goods around the project that can potentially reduce subsequent costs of contribution. For example, OSS project maintainers provide assistance and guidance to new contributors by responding to inquiries via mailing

[¶]In this paper, we will at times classify agents in the OSS public goods setting according to their level of participation in what is known as the "contributor funnel" (McQuaid, 2018). Users of an OSS project may utilize a software product but do not contribute to it. A subset of users are contributors and allocate some contribution effort to developing the project. A subset of contributors are maintainers, typically agents responsible for a large share of project contribution and may also have decision-making power over what proposed contributions are integrated into the project. We will also sometimes refer to these agents as developers.

[‖]While the use of OSS is itself non-rival, the contribution bandwidth of project maintainers is not (Brown, 2018).

[**]In other words, how do maintainers induce project users down the "contributor funnel" into becoming productive, recurring contributors?

lists, message boards, or real-time chat channels.[††] Moreover, OSS communities typically archive the history of such project-related interactions between contributors, creating a publicly accessible knowledge base for project development.[‡‡] OSS projects typically feature documentation[§§] that gives a broad overview of the project.[¶¶], provides detailed information on how the software operates at a technical level, and suggest how to properly propose new contributions.[***] Popular OSS projects can also generate a significant amount of buzz outside the contribution platform itself, from community-authored articles demonstrating usage to external forums[†††] where users can request help for various programming and software tasks. The combination of these features form the basis for peer effects on contributor productivity. Contribution activity itself can generate a form of "digital capital"[‡‡‡] for subsequent OSS production, working to both lower the initial fixed cost of contribution for potential contributors and to make current contributors more productive. Hence, in contrast with many conventional public goods settings, there is scope for individual and peer contribution to become strategic complements.

Salient examples of OSS begin to illustrate the scale at which developers have contributed labor towards the production of complex public information goods. As each OSS developers's "contribution bandwidth" is both scarce and costly, the significance of peer effects that drive contributor labor can be measured naturally in terms of the opportunity cost of a developer's time: what is the equivalent private market labor expenditure to finance the development of large OSS projects?

[††]Users who receive feedback on their contribution from project maintainers are far more likely to return to contribute in the future (Sholler et al., 2019).

[‡‡]In a similar fashion, OSS projects are overwhelmingly managed using a *version control system*, making the entire projects incremental development history public record.

[§§]Note that documentation is generated by developer labor and a contribution to the project itself.

[¶¶]Examples of high-level documentation include project `README` files bundled with the project source code, "wiki" pages, and long-form vignettes on project usage. For an example of best practices on how these are actually integrated into an OSS project, see Sections 8, 10, and 11 of Wickham, 2015.

[***]For example, a project maintainer may include a "contribution template" so that novice contributors avoid common pitfalls for new project contributions. Referring back to the example of Sindre Sorhus, this improves the quality of the proposed change and reduces the "back-and-forth" between maintainer and contributor.

[†††]A relevant example is the programming-focused question and answer website Stack Overflow which has been described as a sister community to OSS collaboration platforms such as GitHub (Eghbal, 2020).

[‡‡‡]Or more accurately, human capital that is recorded or codified as a public information good and then used as an input in the production of additional public goods.

Consider the case of the Linux Kernel. Regarded as the largest collaborative OSS project in history, the Linux Kernel was first released in 1991 by Linus Torvalds and has become the most widely used operating system basis for web servers, mobile devices, and high performance computing infrastructure. As of September 2021, the Linux Kernel has amassed over 31.3 million single lines of code from 23,927 distinct contributors over the past three decades. Using standard methods from software engineering cost estimation, it would take nearly 70 million person-hours to rewrite the entire kernel from scratch, which would cost over $1.05 billion today.[§§§][¶¶¶] While estimates for the use-valuation of OSS is an important ongoing area of research (Greenstein and Nagle, 2014; Nagle, 2019), in this study we seek to characterize the extent to which peer effects can mitigate production costs of OSS public goods.

We seek to empirically assess peer effects on public good production using the context of OSS. Our methodology is organized into two phases. In the first phase, we build intuition on the magnitude of net peer effects in OSS contribution using a reduced form approach. To address concerns over endogeneity, we develop an identification strategy to determine to what extent individual effort levels are influenced by the contribution levels of their peers. Specifically, we instrument the likely endogenous contribution effort of an agent's peers in a given project with the effort levels of the agent's "peers-of-peers" defined by common contribution in outside projects. The instrument operates by changing the relative incentives for peers to contribute to a given project by varying the incentives in external projects. This approach allows us to determine whether individual and peer contribution are strategic complements or substitutes on net, conditional on the set of developers that contribute at all. In the second phase, we develop a structural model of OSS contribution to pin down the microeconomic foundations for contributor behavior. We seek to place emphasis on

[§§§]Estimated (conservatively) using the COCOMO model of software development cost estimation developed by Boehm (1981) and the software utility `scc` (Source: https://github.com/boyter/scc).

[¶¶¶]The median annual salary for U.S. software developers in 2020 was $110,140 ($52.95 per hour) (U.S. Bureau of Labor Statistics, 2021).

Details for this identification strategy are given in Section ?? and Figure ??.

disentangling contribution decisions along the extensive versus intensive margin and integrate peer influence into both decisions. To this end, we embed a micro-founded model of private public good provision (Bergstrom, Blume, and Varian, 1986) into the selection model of (Heckman, 1979). The structural approach facilitates the recovery of individual productivity parameters, allowing us to characterize the welfare of particular contribution profiles and conduct counterfactual analysis. Our main counterfactuals of interest estimates the value of aggregate contribution added by peer effects.

We apply this methodological framework in an empirical analysis, focusing on the context of Open Source Software contribution. We use individual-level contribution data for a random sample of 2,287 highly collaborative OSS projects hosted on the GitHub collaboration platform. The remainder of this paper is organized as follows. We first provide additional background on OSS development in Section **??**. Next, we survey segments of related literature in Section **??**. We introduce the empirical setting in Section **??**, describing the OSS contribution on the GitHub platform and giving an overview of data included in the empirical sample. We then develop a reduced form strategy to estimate peer effects in Section **??**. With high-level insight on net peers effects in hand, we next develop a structural model of public good contribution with extensive and intensive margin peer effects in Section **??**. We outline an estimation strategy, present estimation results, and conduct counterfactual analysis to measure the value of contribution generated by distinct peer effects channels. Finally, we summarize and interpret our findings in Section **??** and discuss promising directions subsequent research.

Bibliography

Andreoni, James (1990). “Impure altruism and donations to public goods: A theory of warm-glow giving”. In: *The economic journal* 100.401, pp. 464–477.

Athey, Susan and Glenn Ellison (2014). “Dynamics of open source movements”. In: *Journal of Economics & Management Strategy* 23.2, pp. 294–316.

Benkler, Yochai (2002). “Coase’s Penguin, or, Linux and" The Nature of the Firm"”. In: *Yale law journal*, pp. 369–446.

Bergstrom, Theodore, Lawrence Blume, and Hal Varian (1986). “On the private provision of public goods”. In: *Journal of public economics* 29.1, pp. 25–49.

Boehm, Barry (1981). *Software Engineering Economics*. Vol. 197. Prentice-Hall, New York.

Brown, C. Titus (July 2018). *A framework for thinking about Open Source Sustainability?* http://ivory.idyll.org/blog/2018-oss-framework-cpr.html. Accessed: 2021-12-04.

Eghbal, Nadia (2020). *Working in public: the making and maintenance of open source software.* Stripe Press.

GitHub, Inc. (2022). *Choose an open source license: Licenses.* https://choosealicense.com/licenses/. Accessed: 2022-06-16.

Greenstein, Shane and Frank Nagle (2014). “Digital dark matter and the economic contribution of Apache”. In: *Research Policy* 43.4, pp. 623–631.

Heckman, James J (1979). “Sample selection bias as a specification error”. In: *Econometrica: Journal of the econometric society*, pp. 153–161.

Lerner, Josh and Jean Tirole (2002). “Some simple economics of open source”. In: *The journal of industrial economics* 50.2, pp. 197–234.

McQuaid, Mike (Aug. 2018). *iThe Open Source Contributor Funnel (or: Why People Don’t Contribute To Your Open Source Project).* https://mikemcquaid.com/2018/08/14/the-open-source-contributor-funnel-why-people-dont-contribute-to-your-open-source-project/. Accessed: 2021-12-04.

Nagle, Frank (2019). "Open source software and firm productivity". In: *Management Science* 65.3, pp. 1191–1215.

Open Source Initiative (Mar. 2007). *The Open Source Definition.* https://opensource.org/osd. Accessed: 2022-06-16.

Ostrom, Elinor (1990). *Governing the commons: The evolution of institutions for collective action.* Cambridge university press.

Sholler, Dan, Igor Steinmacher, Denae Ford, Mara Averick, Mike Hoye, and Greg Wilson (2019). "Ten simple rules for helping newcomers become contributors to open projects". In: *PLoS computational biology* 15.9, e1007296.

Sorhus, Sindre (May 21, 2019). *(@sindresorhus) Some observations from having merged thousands of pull requests in the past few years.* https://twitter.com/sindresorhus/status/1130791267393163267.

— (Dec. 2021). *Sindre Sorhus (personal web page).* https://sindresorhus.com/. Accessed: 2021-12-04.

U.S. Bureau of Labor Statistics (Sept. 2021). *Software Developers, Quality Assurance Analysts, and Testers : Occupational Outlook Handbook.* URL: %5Curl%7Bhttps://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm%7D.

Wickham, Hadley (2015). *R packages: organize, test, document, and share your code.* " O'Reilly Media, Inc."