Universidad
Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

Value Implications of Open Source Software:
An Empirical Outlook to the Open Innovation Model

Autor:

Dilan Aksoy-Yurdagül

Directores:

Andrea Fosfuri
Marco Giarratana

DEPARTAMENTO DE ECONOMIA DE LA EMPRESA
Universidad Carlos III de Madrid

Getafe, Noviembre 2013

Universidad Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

Value Implications of Open Source Software:

An Empirical Outlook to the Open Innovation Model

Autor: Dilan Aksoy-Yurdagül

Directores: Andrea Fosfuri, Marco Giarratana

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Secretario:

Calificación:

Getafe, de de

Acknowledgements

I am very grateful to my advisors, Andrea Fosfuri and Marco Giarratana, for having me guided through these years. It has been a privilege to have been their advisee and to have followed the path they have enlightened. I would like to give my deepest gratitude to two exemplar researchers, Sonali Shah and Francesco Rullani, to whom I look up to very much. I consider myself very lucky to have found the chance to work with them and to have been exposed to their work disciplines. I have learned a lot from them.

I would also like to thank to the Business Administration Department at Universidad Carlos III de Madrid. I am thankful to the valuable faculty members, Neus Palomeras, Eduardo Melero, Jaime Ortega, Josep Tribó, Alicia Barroso, Pablo Ruiz-Verdu, Samira Dias dos Reis, Fabrizio Cesaroni, Esther Roca Batllori, Ester Martínez-Ros, Emre Ekinci and Pedro Serrano for their insightful comments and suggestions on my dissertation and for having aided with great help through all these years I have been pursuing my PhD degree. I would like to thank to the members of our secretariat office, Marié, Juan Antonio, Begoña and Raquel, for making things easier when everything seemed so complicated. I was mainly funded by the Department of Business Administration at Carlos III de Madrid, the Spanish Ministry of Science and Innovation and the Spanish Ministry of Economy and Competitiveness for my research. I gratefully acknowledge the financial support from these sources, which made my research possible. The Department of Economics and Business at Luiss Guido Carli University in Rome and Foster School of Business at University of Washington in Seattle have hosted me as a visiting researcher for short terms during my doctoral studies. I am grateful to these institutions for having me there and for giving me the opportunity to be surrounded by many valuable scholars.

I warmly thank to my friends Juliana, Agata, Su-ping, Argyro, Gökhan and Aslı. I have endured many things thanks to them. I am particularly thankful to Zeynep and Inci for the sunny Madrid days we have wandered around together despite our very busy schedules. I am grateful to my very best friends, Mine and Duygu, for their positive energy at all times; to Merve, for her endless support across oceans; and to Magda, for being the best flat mate ever.

I would like to express how lucky I feel to have married to the most idealistic PhD student I have ever met. Emircan, thank you for being there when I needed the most and for giving me the energy and the will to accomplish my goals. I would not have made it through without your presence in my life.

Words alone are not enough to express my gratitude to my family. I thank my father, Kazım, for having supported me in all my pursuits and for believing in me for whatever it is. I am thankful to my younger sister Helin, for giving me a reason to be a hard worker; to my older sister, Hazal, with whom we have achieved to be friends during though years of our lives; to my grandparents, Şengül and Ali Hikmet, for making me feel at home even when I thought of them; to my aunt, Filiz, for cheering me up when I needed.

Special thanks go to my dearest Şebnem, my mother, who filled my heart with love, my soul with joy and my mind with endless dreams. I thank her for who I am and who I ever will be. It is all for you.

Abstract

This thesis consists of three essays on open source software (OSS) phenomenon, which constitutes a suitable setting for investigating the open innovation paradigm and the benefits of firm-community collaborations for a firm.

The first essay examines the impact of firms' stocks of intellectual property right endowments on the relationship between firms' OSS product portfolio and its firm value. The results suggest that firms taking more commercial actions in OSS paradigm enhance their firm value in the presence of large stocks of software patents. On the other hand, software trademark stocks have a negative effect on this relationship. Several potential explanations are proposed in the study, which emphasizes the importance of intellectual property right protection mechanisms for those firms that aim at building an OSS-based product portfolio.

The second essay investigates the benefits that might accrue to a firm from their code contributions as reflected in stock market prices. The study builds on recent work in the area of community-based innovation and tries to explain why a firm might benefit from contributing to the *commons*. The hypotheses are tested on a novel data set that captures the stock market responses to 149 press releases made over a thirteen-year period. The findings suggest that open source code contributions generate positive, abnormal returns for the contributing firm. In addition, we find that contributions of new code rather than existing code generate greater value for the firm—and that contributions of new code to software projects targeted at end-users rather than developers systematically generate even greater returns.

The third essay focuses on collaborations among firms and communities for obtaining better outcomes from open source projects. In specific, this study aims to disentangle how organizational design may affect the performance of an OSS project. Project's management model and employee involvement in the project are treated as potential mediators that may have an effect on the aforementioned relationship. The empirical analysis is undertaken on a sample of OSS projects hosted on the platform SourceForge.net from December 2006 to December 2008. The findings of this study are three fold. First, being directly involved in a project with a specific policy on OSS has a positive effect on project's performance. Second, coordination by firm has a negative effect on performance. Third, admin as an employee on main duty does not have a

direct positive effect on performance. However, it positively moderates the aforementioned relationship.

The contributions of these essays to the existing literature can be articulated as follows: The first essay contributes to the current debate on commercialization of OSS, mainly, by establishing the intermediary effect of firms' IPR holdings on the relationship between OSS commercialization and performance. IPR mechanisms in the form of software patents and software trademarks needed to be investigated as they are crucial firm resources in appropriating returns. This study sheds light on how much these mechanisms matter for performance consequences of commercializing OSS. Moreover, the study empirically shows how mismatching strategies may harm a firm's performance while the firm attempts to move towards a new business model for better outcomes.

The second essay contributes to the literature on community-based innovation by enhancing our understanding of the process by which firms can work within communities. Literature on open innovation encapsulates community-based innovation, as well as other mechanisms by which the firm searches its external environment for knowledge. Other mechanisms—alliances, joint ventures, university collaborations—have been the object of much empirical and theoretical attention in the strategy and technology management literatures. In contrast, community-based innovation has been the object of less scholarly work, but has attracted the attention of academics and practitioners alike, and firm engagement with communities appears to be on the rise. It also contributes to the literature on learning, where communities can provide valuable knowledge to firms, by providing systematic empirical documentation.

The third essay contributes to knowledge on benefits of collaborations between firms and OSS communities. Firm's role in the complex coordination mechanism of the project on project's success has been investigated. Along this line, the paper can also directly inform managers on the strategies they should apply to assure long term sustainability of their external knowledge sourcing activities through communities. Managing the boundaries of collaborations is essential. Written rules and guidelines lead to fruitful joint development of software.

Resumen

Esta tesis está compuesta por tres ensayos sobre el fenómeno de software de código abierto (SCA), lo que constituye un marco adecuado para la investigación del paradigma de la innovación abierta y los beneficios de las colaboraciones entre empresas y comunidades para una empresa.

El primer ensayo analiza el efecto de las reservas de derechos de propiedades intelectuales de una empresa sobre la relación entre el portafolio de productos SCA de empresas y el valor de la empresa. Los resultados sugieren que las empresas que han adoptado medidas comerciales en el paradigma de SCA aumentan su valor en la presencia de grandes cantidades de patentes de software. Por otra parte, la existencia de marcas de software tiene un efecto negativo en esta relación. Varias explicaciones se proponen en el estudio, que hace hincapié en la importancia de los mecanismos de protección de derechos de propiedad intelectual para las empresas que tienen como objetivo la construcción de una cartera de productos basado en SCA.

El segundo ensayo investiga los beneficios que una empresa podría obtener por sus contribuciones de código fuente como se refleja en los precios del mercado de valores. El estudio se centra en la línea de investigación de la innovación basada en la comunidad y trata de explicar cómo una empresa puede beneficiarse por contribuir al bien común. Las hipótesis se prueban con un dato nuevo que captura las reacciones del mercado de valores a 149 notas de prensa realizadas durante un período de trece años. Los resultados sugieren que las contribuciones de código abierto generan retornos anormales positivos para la empresa contribuyente. Además, descubrimos que las contribuciones de código nuevo respecto de aquellas de código existente generan mayor valor para la empresa. Y que las contribuciones de código nuevo a los proyectos de software dirigidos a los usuarios finales respecto de aquellos dirigidos a los desarrolladores, generan sistemáticamente aún mayores retornos.

El tercer ensayo se centra en la colaboración entre empresas y comunidades para la obtención de mejores resultados de los proyectos de código abierto. En concreto, este estudio tiene como objetivo desentrañar cómo el diseño organizacional puede afectar al rendimiento de un proyecto de software libre. El modelo de gestión del proyecto y la participación de los empleados en el mismo son tratados como mediadores potenciales que pueden tener un efecto en la relación mencionada. El análisis empírico se realiza

sobre una muestra de proyectos de software libre alojados en la plataforma SourceForge.net de diciembre 2006 a diciembre de 2008. Los hallazgos de este estudio son los siguientes. En primer lugar, participar directamente en un proyecto con una política específica en SCA tiene un efecto positivo en el rendimiento del proyecto. En segundo lugar, la empresa como la autoridad principal de coordinación tiene un efecto negativo en el rendimiento del proyecto. En tercer lugar, si el administrador es el empleado de la empresa no tiene un efecto directo sobre el rendimiento del proyecto. Sin embargo, modera positivamente la relación entre la empresa coordinadora y el rendimiento del proyecto.

Las contribuciones de estos ensayos a la literatura existente pueden articularse de la siguiente manera: El primer ensayo contribuye al debate actual sobre la comercialización de la SCA, sobre todo, estableciendo el efecto intermediario de derechos de propiedad intelectual sobre la relación entre la comercialización de software libre y el rendimiento. Los mecanismos de derechos de propiedad intelectual en forma de patentes de software y las marcas de software necesitan ser investigados, ya que son recursos cruciales de una empresa para poder capturar retornos. Este estudio arroja luz sobre cuánto estos mecanismos importan para el desempeño de comercialización de SCA. Por otra parte, el estudio muestra empíricamente cómo las estrategias incompatibles pueden dañar el rendimiento de la empresa, mientras la empresa pretende avanzar hacia un nuevo modelo de negocio para mejorar su rendimiento.

El segundo ensayo contribuye a la literatura de la innovación basada en la comunidad mediante la mejora de nuestra comprensión sobre el proceso por el que las empresas pueden trabajar con comunidades. La literatura sobre la innovación abierta encapsula la innovación basada en la comunidad, tanto como otros mecanismos por los que la empresa busca conocimiento en su entorno externo. Otros mecanismos- alianzas, empresa conjunta, colaboración con universidades- han sido objeto de atención empírica y teórica en la literatura de estrategia y de la gestión tecnológica. Por el contrario, la innovación basada en la comunidad ha sido objeto de trabajo escolar menos a menudo, pero ha atraído la atención de académicos y profesionales también, y la colaboración de las empresas con las comunidades parece estar en aumento. También contribuimos a la literatura del aprendizaje, donde las comunidades pueden proporcionar valiosos conocimientos a las empresas, aportando documentación empírica sistemática.

El tercer ensayo contribuye al conocimiento sobre los beneficios de la colaboración entre las empresas y las comunidades de software libre. El papel de la empresa en el mecanismo complejo de la coordinación sobre el éxito del proyecto ha sido investigado. En esta línea, este trabajo también puede informar directamente a los administradores sobre las estrategias, que se deben aplicar, para asegurar la sostenibilidad de la adquisición de conocimiento desde los recursos externos a largo plazo, a través de las actividades de las comunidades. La gestión de los límites de la colaboración es esencial. Las normas y directrices escritas conducen al proceso beneficioso de creación conjunto de software.

Contents

CHAPTER 1 - Introduction

1.1 Open Source Software

Academic research has dedicated a great deal of effort to reveal and explain the sources of knowledge flows that lead to innovation. Although established theories suggested a closed innovation model, whereby a company generates, develops and commercializes its own ideas, a fundamentally different organizational model for innovation and product development referred to as open innovation (Chesbrough, 2003) has recently come to the fore. Open innovation model focuses on identifying, exploiting and integrating external knowledge into their internal R&D activities (West and Gallagher, 2006; Laursen and Salter, 2006). Firms adopt this new open innovation model that exploits the knowledge of various actors (e.g. customers, suppliers, competitors or universities) to sustain the ability to introduce new products to the market successfully (Chesbrough, 2003). Open source software (OSS), which is freely available to all that accept the licensing terms of the software, is the outcome of such an open innovation model whereby companies collaborate with communities of volunteer developers with intentions to co-create software.

OSS has drawn attention from diverse academic fields throughout the last decade. This growing body of literature clearly emerged out of the great success achieved by well-known OSS projects such as Linux and Apache. The earliest studies centered mostly on motives that drove users and developers to contribute to OSS projects (Lakhani and Wolf, 2005; Lerner and Tirole, 2002), how innovation processes functioned (Lakhani and von Hippel, 2003), and governance issues in OSS communities (O'Mahony and Ferraro, 2007; Shah, 2006). As OSS became commercially viable (Fitzgerald, 2006), a new stream of research began to focus more on for-profit firms that collaborated with OSS communities (Bonaccorsi, Giannangeli and Rossi, 2006; Dahlander and Magnusson, 2005), the intellectual property right (IPR) mechanisms they used to ensure their returns from open business models (Henkel, 2006; Lerner and Tirole, 2005), and the competitive dynamics introduced by OSS (Bonaccorsi and Rossi, 2003; West, 2003).

Henkel (2006) identifies four groups of benefits of OSS engagement for a firm: standard setting and compatibility, increased demand for complementary goods and services, external development support, and signals of technical excellence or good citizenship. Furthermore, OSS products can enhance the value created by proprietary software because they expand the firm's product portfolio and help sustain its diversification. Specifically, they encourage quality improvements and innovation, threaten competitors, reduce costs, lead to dynamic capabilities, and ultimately can produce a competitive advantage (Bonaccorsi et al., 2006; Dahlander and Magnusson, 2005; Schmidth and Schnitzer, 2003; West 2003).

Despite the potential benefits articulated above, given the open nature of OSS that tends to make them public good (Lerner and Tirole, 2002), firms face the challenge of reaping these benefits out of it. Private-collective model of innovation enables us to understand why contributing to OSS communities may bring about private rewards for contributing parties that are not available to free-riders (von Hippel and von Krogh, 2003). These private benefits, which exceed the costs of contributing, might accrue to the firm by enhancing the value of complementary assets (Dahlander and Wallin, 2006) or by increasing network externalities and nurturing knowledge flows (West, 2003). Appropriability regimes that use legal mechanisms of protection help firms advance their capabilities to retain created value in this specific setting due to the modular architecture of OSS code that can be combined with proprietary developments (Teece, 1986, 1998).

Although previous studies have identified potential benefits of OSS for engaging parties at large, the research stream lacks empirical evidence on the benefits of OSS engagement for a for-profit firm. To our knowledge, few studies have sought to unpack if and how firms actually benefit from contributing to innovation communities. Stam (2009) explores how firms that commercialize OSS products improve their innovative and financial performance through community participation. Alexy and George (2013) report systematic links between open and distributed models of innovation and firm value conditional on the legitimacy of firms doing so, as perceived by the capital market once the firm chooses its business model. Fosfuri, Giarratana and Luzzi (2008), on the other hand, emphasize the role of IPR mechanisms in firms' decision to engage in OSS.

In this thesis, I aim to extend the recent line of research mentioned above by focusing on the effect of engaging in OSS communities and commercialization of OSS on firm performance. I attempt to shed some light on the open innovation paradigm by providing with empirical evidence on how, if any, firms may benefit from collaborations with communities within the context of OSS.

1.2 References

Alexy, O., George, G. 2013. Category Divergence, Straddling, and Currency: Open Innovation and the Legitimation of Illegitimate Categories, *Journal of Management Studies* 50(2), 173-203.

Bonaccorsi, A., Giannangeli, S., Rossi C. 2006. Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science* 52(7), 1085-1098.

Bonaccorsi, A., Rossi C. 2003. Why OSS can succeed. *Research Policy* 327, 1243-1258.

Chesbrough, H. W. 2003. Open Innovation: The New Imperative for Creating and Profiting from Technology. *Harvard Business School Press*, Boston, MA.

Dahlander, L., Magnusson, M. G. 2005. Relationships between open source software companies and communities: observations from Nordic firms. *Research Policy* 34, 481–493.

Dahlander, L., Wallin, M. W. 2006. A man on the inside: unlocking communities as complementary assets. *Research Policy* 35(8), 1243-1259.

Fitzgerald, B. 2006. The transformation of open source software. *MIS Quarterly* 30 (3), 587-598.

Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35(7), 953-969.

Fosfuri, A., Giarratana M., Luzzi A. 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science* 19, 292-305.

Lakhani, K. R. von Hippel, E. 2003. How open source software works: "free" user-to-user assistance. *Research Policy* 32, 923-943.

Lakhani, K. R., Wolf, R. G. 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects, in J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani (eds.) Perspectives on Free and Open Source Software. MIT Press, Cambridge, MA, USA.

Laursen, Keld, Salter, A. 2006. Open for innovation: the role of openness in explaining innovation performance among UK manufacturing firms. *Strategic Management Journal* 27, 131–150.

Lerner, J., Tirole J. 2002. Some simple economics of open source. *Journal of Industrial Economics* 50(2), 197-234.

Lerner, J., Tirole, J. 2005. The scope of open source licensing. *Journal of Law, Economics and Organization* 21, 20-56.

O'Mahony, S., Ferraro F. 2007. The emergence of governance in an open source community. A*cademy of Management Journal* 50(5), 1079–1106.

Schmidth, K. M., Schnitzer M. 2003. Public subsidies for open source? Some economic policy issues of the software market. *Harvard Journal of Law & Technology* 16(2), 474-502.

Shah, S. K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science* 52(7), 1000-1014.

Stam, W. 2009. When does community participation enhance the financial and innovative performance of open source companies? *Research Policy* 38, 1288-1299.

Teece, D. J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15, 285-305.

Teece, D. J. 1998. Capturing Value from Knowledge Assets: The New Economy, Markets for Know-How, and Intangible Assets. *California Management Review* 40(3), 55-79

von Hippel, E., von Krogh, G. 2003. Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science. *Organization Science* 14(2), 209-233.

West, J. 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32(7), 1259–1285.

West, J., Gallagher, S. 2006. Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management* 36(3), 319-331.

CHAPTER 2 - The Impact of Open Source Software Commercialization on Firm Value

2.1 Introduction

Open source software (OSS) has drawn attention from diverse academic fields following the great success achieved by well-known OSS projects such as Linux and Apache. Several studies centered mostly on motives that drove users and developers to contribute to OSS projects (Lerner and Tirole, 2002; Lakhani and Wolf, 2005), how innovation processes functioned (Lakhani and von Hippel, 2003), and governance issues in OSS communities (Shah, 2006; O'Mahony and Ferraro, 2007). As OSS became commercially viable (Fitzgerald, 2006), another line of research that focuses on for-profit firms collaborating with OSS communities (Dahlander and Magnusson, 2005; Bonaccorsi, Giannangeli and Rossi, 2006), the intellectual property right (IPR) protection mechanisms they used to ensure their returns from open business models (Lerner and Tirole, 2005; Henkel, 2006), and the competitive dynamics introduced by OSS (Bonaccorsi and Rossi, 2003; West, 2003) has emerged.

Although the widespread adoption and acceptance of OSS has been attributed to its complementarity to proprietary developments (Bonaccorsi and Rossi, 2003; Schmidth and Schnitzer, 2003; West 2003) and its potential benefits for the engaging parties have been identified at large (Henkel, 2006), the benefits that OSS commercialization might bring about to a for-profit firm still lack empirical evidence. Investigation on commercializing OSS in relation to its profitability and the heterogeneity in firms' assets in determining the level of profitability, if any, remains underexplored. To the best of my knowledge, there are only a few studies that try to link OSS commercialization and firm performance. Fosfuri, Giarratana and Luzzi (2008) focus on heterogeneity among firms' endowments of protection mechanisms for IPR in explaining their decisions on incorporating OSS into commercial product portfolios. Their results suggest that variations in pre-existing stocks of protection mechanisms for IPR, namely patents and trademarks, help to explain why some firms are taking more commercial actions within the OSS paradigm than others. Stam (2009) explores how firms that commercialize OSS products improve their innovative and financial performance through community participation. Alexy and George (2013) report systematic links between open and distributed models of innovation and firm value conditional on the legitimacy of firms doing so, as perceived by the capital market once the firm chooses its business model.

In this paper, I aim to extend the recent line of research mentioned above by focusing on the effect of commercializing OSS on firm performance. Specifically, I assess the moderating role of IPR protection mechanisms on the relationship between firm value and OSS commercialization. In order to do so, I study a panel data for 70 companies covering a seven year time span from 2003 to 2009. I argue that the potential for appropriating returns within the context of OSS commercialization depends critically on firms' possession of software patents and software trademarks. The findings are twofold. First, I show that OSS commercialization cannot benefit a firm without the right set of IPR stocks. Second, software patent and software trademark stocks interact with OSS portfolio, such that while excessive number of software trademarks may affect negatively the relationship between firm value and OSS commercialization, software patent stocks have a positive effect on this relationship.

In the next section, I present the theoretical background for the empirical analysis, before describing the data and identifying the variables in my estimations. In the empirical section, I present the results. Finally, I offer a brief discussion of the findings and conclude.

2.2 The Role of Firms' IPR Protection Mechanisms

The literature on protecting the valuable knowledge is extensive (Rumelt, 1984; Barney, 1986). Closed innovation model suggests that a company generates, develops and commercializes its own ideas and it is managers' key challenge to retain the knowledge within the firm in order to appropriate returns (Teece, 1986). On the other hand, open innovation, which has come to the fore rapidly as a new paradigm for corporate innovation, focuses on identifying, exploiting and integrating external knowledge into their internal R&D activities (West and Gallagher, 2006). Firms that adopt the open innovation model integrate these external sources (e.g., customers, suppliers, competitors, universities) into their internal innovation processes and competitive strategy to sustain the ability to introduce new products to the market successfully (Chesbrough, 2003). OSS, which is freely available to all that accept the licensing terms of the software, is the outcome of such an open innovation model whereby companies collaborate with communities of volunteer developers. Given the open nature of OSS that tends to make them public good (Lerner and Tirole, 2002), firms face the challenge of reaping the benefits out of it. Appropriability regimes that use legal mechanisms of protection help firms advance their capabilities to retain created value in this specific setting due to the modular architecture of OSS code that can be combined with proprietary developments (Teece, 1986, 1998).

How do firms' endowments of IPR affect the relationship between commercialization of OSS and performance? To address this question, I first investigate the IPR protection mechanisms that might be crucial in appropriating value for a firm that commercializes OSS. Studies of OSS and associated appropriation regimes suggest several protection methods, ranging from licensing (Behlendorf, 1999; Hecker, 1999; Raymond, 1999) to standard protections such as copyright, secrecy, lead time, and complementary assets (Feller and Fitzgerald, 2002; Dahlander, 2005; Fosfuri et al., 2008).

I focus specifically on software patents and software trademarks, which are important firm resources that explain the heterogeneity in firms' decision to engage in OSS commercialization (Fosfuri et al., 2008). Although copyrights provide an alternative IPR mechanism for software, my aim is to investigate whether or not firms' efforts to build an open source-based strategy would bring the necessity to build strategic IPR portfolios as opposed to questioning the power of copyrights, software patents or trademarks in protecting the underlying technology per se. Building strategic patent/trademark portfolios dependent upon a firm's short-term goals and long-term objectives is a debated issue in the domain of strategic management (Hall, 1992; Mendonça, Pereira and Godinho, 2004; Blind, Cremers and Mueller, 2009). Yet, within the context of OSS they have a critical role in defining the conditions in which its commercialization can be most beneficial.

2.2.1 Software Patents

As suggested by Hall and Ziedonis (2001), patents are important IPR mechanisms that a firm can possess not only because they operate as effective appropriation mechanisms but also because they confer negotiation power on technologies over external patent owners. Alexy and Reitzig (2013) argue that firms, which pursue private-collective model of innovation, continue acquiring patents mainly to gain *de facto* control over a technology among several proprietary competitors within the context of OSS. With a particular focus on appropriation mechanisms needed to profit from OSS commercialization, Mann (2006) argues that corporate members of OSS communities often continue to make heavy investments in software patents, primarily to protect themselves from the threat of litigation. The results suggest that without the intellectual property protection mechanisms, OSS cannot continue to grow in commercial importance.

Fosfuri et al. (2008) argue why having a large portfolio of software patents might be good for OSS commercialization in three ways: potential complementarity between the patented software and commercial OSS products; control that they exert on future improvements of an OSS project; protection they provide with against litigation. Shah (2006), on the other hand, suggests that tight control of IPR also might discourage participation in OSS projects that rely mainly on volunteer programmers. According to this point of view, the concept of extracting financial benefits from jointly developed software contradicts the core values of OSS movement and thus, harms community members' willingness to participate free of charge (Knyphausen-Aufseß and Schweizer, 2011).

In light of the previous related work, one may intuitively think that avoiding the extremes in building software patent portfolios when engaging in OSS communities for developing commercial OSS might be the optimal choice. However, it is important to make the distinction between creation of OSS and its commercialization packaged into proprietary products or services (Teece, 1998). Tight IPR might deter participation to an OSS project from volunteer developers. However, its benefit due to its use as an appropriation mechanism is essential. While implementation help by volunteer developers is important for the advancement of the project, firms that aim to develop a commercial product jointly with a community do so mainly in order to receive design help from their potential future customers (Goldman and Gabriel, 2005). Thus, firms are more concerned about attracting potential users, rather than programmers, to the communities that they collaborate with or they found with the aim of developing a commercial OSS product. Losing participation due to tight appropriability, thus, is not an issue for such communities whose principal contributors are future customers.

In turn, I suggest that the positive effect of software patent endowments will dominate and a large software patent stock will operate as a complementary asset for firms commercializing OSS. Even if the software patents in such a portfolio may not constitute a direct protection for what is being released per se, they may enable the firm to capture the value created through other channels. For instance, as noted by Lerner and Tirole (2002), by boosting profits on a complementary segment in proprietary form. A large portfolio of software patents will help firms expand their difficult-to-imitate complementary assets, which play key roles in firms' sustained competitive advantage equation, for commercial OSS products leading to superior performance as measured by Tobin's q.

Hypothesis 1: Large software patent stocks positively affect the relationship between a firm's open source software product portfolio and its firm value; such that as the number of software patents increases the relationship becomes stronger.

2.2.2 Software Trademarks

In contrast with the extensive debate on the value of patents in classical incentive theory (Hall and MacGarvie, 2010), trademarks have received attention only recently (Mendonça et al., 2004). A trademark refers to any word, symbol, or name used by a manufacturer to distinguish its goods from those sold by others. New trademarks provide important tools for positioning products in commercial settings. Mendonça et al. (2004) test whether trademarks might provide complementary indicators of innovation, which generally is represented instead by R&D expenditures and patents. They find a positive relationship between trademarking activity and product innovation. Trademarks serve for differentiation purposes, distinguishing the brand by leveraging its owner's reputation. Registering trademarks is considered as an investment tailored to proprietary domain that aims at increasing consumers' willingness to pay a premium for the quality of the product and the brand name (Fosfuri et al., 2008). OSS is a collaborative mode of software development that targets consumers, who seek cost-saving solutions. Hence, firms with large numbers of OSS releases may cannibalize their existing lines of businesses in proprietary domains that are strongly associated with their trademark investments.

Trademarks might as well be argued to improve the relationship between a firm's OSS product portfolio and its value as they help firms build a strong brand name and reputation. Linux, Apache, and Debian all have trademarked their names, both for differentiation purposes and to prevent proprietary appropriations of their OSS code

(O'Mahony, 2003). However, the use of such a brand investment is to emphasize that the new product is superior to its existing substitutes with similar performances (Fosfuri et al., 2008). Consequently, I suggest that for firms with large repository of proprietary developments, commercializing OSS might entail the risk of eliminating their primary sources of competitive advantage, in other words, their proprietary software brand names that would lead to a loss of firms' product-based competitive edge.

As suggested by Alexy and George (2013), investors favor firms' innovative activity when it is done legitimately. The authors stress the importance of category emergence and argue that once a category is chosen, the firm should act in a way that complies with its current innovation strategy to receive benefits. Building a large stock of software trademarks can be seen as part of a proprietary-based strategy. Commercializing open source software in such a case may send mixed signals to its customers and damage its overall position in the market place. Moreover, most of the commercial OSS products target home users, who are willing to pay more for ready-to-use packages. On the other hand, OSS, at times, necessitates customization and is preferred by more skilled users (e.g. developers, corporate users).

Thus, I suggest that building an OSS-based software portfolio when having registered for several software trademarks, in other words when having invested in proprietary domains, will have a negative effect on firm performance as measured by Tobin's q.

Hypothesis 2: Large software trademarks stocks negatively affect the relationship between a firm's open source software product portfolio and its firm value; such that as the level of software trademark stocks increases the relationship becomes weaker.

2.3 Data and Descriptive Statistics

I undertake an empirical analysis with a novel data set in which I intend to include all companies operating in 10 software-related industries[1] from the *Fortune Global 500* list. Although the initial data contains information on 83 companies from

[1] These industries are Internet Services & Retailing, Electronics & Electrical Equipment, Network & Other Communications Equipment, Computers & Office Equipment, Computer Software, Computer Peripherals, Telecommunications, Photo & Control Equipment, Information Technology Services, and Semiconductors.

1999 to 2009, I excluded those firms that lack financial information throughout the time span under study. Although I started collecting data from 1999, I ran the analysis using data starting from 2003. All the main explanatory variables are cumulative numbers, and four years of batching period should provide more reliable data. The final sample, which consists of 490 firm-year (70 firms × 7 years) observations, enables me to explore the performance consequences of OSS commercialization under heterogeneous regimes of IPR protection.

Table 1: Variable Definitions

Variable Name	Description	Source
TobinsQ (Dependent Variable)	(Market Value + Preferred Stock + Debt) / Total Assets (Log transformed)	Compustat
OssPortfolio	$\sum_{1999}^{t}$ OSS products calculated for year t at a 15% discount rate (Log transformed)	PROMT
SoftwarePatents	$\sum_{1999}^{t}$ software patents calculated for year t at a 15% discount rate (Log transformed)	USPTO
SoftwareTrademarks	$\sum_{1999}^{t}$ software trademarks calculated for year t at a 15% discount rate (Log transformed)	USPTO
FirmSize	Employees in year t (Log transformed)	Compustat
FirmAge	Age of the firm (Log transformed)	Compustat

From the Compustat database, I extract financial data, including market values, total assets, total debt, and preferred stock. I use Tobin's q as the dependent variable, which offers a forward-looking proxy for firm value. When the value of Tobin's q is greater than 1, the outlook for the firm's growth opportunities is considered to be positive. Although this measure generally serves as a proxy to value intangible assets such as patents or trademarks, it also can reflect market hype or speculation, as is common in technology markets. As has been discussed by Lerner and Tirole (2002), releasing open source can be considered as a strategic move (e.g., to weaken a competitor). OSS commercialization and the use of complementary assets for revenue strategies should be differentiated as an effort for building a new business model that would bring benefits in the long term as opposed to releasing a proprietary software product with expectations of direct monetization for what is being released per se. Releasing source code signals technical excellence and goodwill (Henkel, 2006) that originates from firms' intellectual capital, which can be captured by Tobin's q. I calculate a modified version of Tobin's q using the following formula (Chung and Pruitt, 1994):

$$\text{Approximate } q = (MVE + PS + DEBT) / TA,$$

where MVE is the product of common shares outstanding and the month-end price that corresponds to the period end date; PS is the liquidating value of preferred stock; DEBT is the sum of total long-term debt and debt in current liabilities; and TA is the book value of total assets. The approximate q explains 96.6% of the variability of Tobin's original formulation (Lindenberg and Ross, 1981). Firm size, as measured by the number of employees, comes from Compustat.

I next searched for press articles that reported a “product announcement,” “new software release,” or “software evaluation” in the software sector (standard industry classification code 7372) in the PROMT database. Reports of product introductions that included the phrases “open source” or “Linux” indicated possible OSS product introductions. I read the text of each article in the possible OSS product introductions set to distinguish articles that clearly referred to an OSS product introduction. Using the data from the product introduction announcements, I compute the cumulative number of OSS product announcements as a proxy for a firm’s OSS product portfolio. Specifically, I sum the number of OSS products introduced to the market starting from 1999 at a 15% annual discount rate,[2] such that I use the following formula:

$$OssPortfolio_t = (Oss_t + Oss_{t-1}\times(0.85) + Oss_{t-2}\times(0.85)^2 + \ldots + Oss_{1999}\times(0.85)^{t-1999}),$$

where Oss_t is the number of OSS introductions in year t.

Table 2: Distribution of Explanatory Variables by Sector

	All Firms				OSS Commercializing Firms			
	No. of Firms	Av. # Products	Av. Sw Patents	Av. Sw Trademarks	No. of Firms	Av. # OSS Products	Av. Sw Patents	Av. Sw Trademarks
Internet Services	1	1.28	3.80	1.09	1	0.21	3.80	1.09
Electronics	10	1.31	4.80	1.86	3	0.69	7.09	3.24
Network	9	2.21	5.54	2.01	5	1.04	6.72	2.54
Comp. & Office Eq.	12	3.22	6.71	2.42	8	2.15	6.63	2.61
Computer Software	6	4.28	4.00	2.52	3	2.91	6.39	3.60
Computer Peripherals	4	1.70	4.31	1.25	2	0.28	5.21	1.83
Telecommunications	10	1.14	4.22	1.14	1	0.40	6.71	2.19
Photo & Control Eq.	4	1.48	3.70	1.59	0	0	0	0
I.T. Services	2	2.14	2.89	0.60	1	0.34	5.78	0.67
Semiconductors	12	1.55	5.08	1.36	4	1.03	6.15	2.07
Total No. of Firms	70				28			

[2] The Bureau of Industry Economics (BIE, 1994) adopts a discount rate of 15% when assessing private returns to patent holders because the value of patents tends to fall over time. I follow a similar logic for assessing private returns that OSS commericalization may bring about since the value associated with older products will diminish over time as technology evolves and enhanced products enter the market.

The information on patents and trademarks are from the U.S. Patent and Trademark Office (USPTO) database. I looked for all patents granted to a given firm each year (1999–2009). After gathering data about all patents, I identified software patents using Graham and Mowery's (2003) algorithm, according to which certain international classification classes—such as "Electric Digital Data Processing" (G06F), "Recognition of Data; Presentation of Data; Record Carriers; Handling Record Carriers" (G06K), and "Electric Communication Technique" (H04L)—represent software patents.[3] These classes are selected after examining the patents of six major U.S. software producers (1995 revenues; Microsoft, Adobe, Novell, Autodesk, Intuit, and Symantec) between 1984 and 1995. Patents in the three selected classes account for 57% of the patents assigned to the 100 largest firms in the software industry (Hall and MacGarvie, 2010). Furthermore, trends in these classes represent overall software patenting activity, because they include areas in which patenting grew rapidly in the late twentieth century.

Hall and MacGarvie (2010) compare three different classification methods to identify software patents with a sample of over 1000 manually identified software patents by John Allison (Allison and Lemley, 2000; Allison and Tiller, 2003). These three approaches to determine software patents are the ones that are used by Graham and Mowery (2003), by Bessen and Hunt (2007) and by Hall and MacGarvie (2010). Hall and MacGarvie (2010) also provide a new approach that combines those three approaches and suggest the superiority of the proposed combined method. Graham and Mowery (2003) approach does well on Type II error (detecting a patent as software patent when it is not), which is approximately 5%. However, it is not as successful in the case of Type I error (missing a software patent when it should have been identified), which is 40%. The proposed combined method by Hall and MacGarvie (2010) can achieve somewhat better at Type I error (27%) and does almost the same at Type II error (5%). Thus, I believe that applying the proposed combined method would not have changed my results since the software patents I have captured are representative for the actual software patent stocks a firm is endowed with. Identifying non-software patents as software patents is more problematic than missing some of the software patents that I should have identified. The other method by Bessen and Hunt (2007), although very good at identifying the software patents, identifies a large portion of non-software

[3] The groups included are as follows: G06F 3,5,7,9,11,12,13,15; G06K 9,15; and H04L 9.

patents as software patents. Hence, I prefer to use Graham and Mowery's (2003) approach over the other two. It is a more straightforward approach and will give robust results.

Using the same source, I also extracted trademark data at the firm level on a yearly basis by searching for all trademarks filed by each firm between 1999 and 2009. To distinguish software trademarks, I applied the search algorithm suggested by (Fosfuri et al., 2008) and searched for strings of words[4] in the text of the trademark description. The strings of words used to distinguish a software trademark are as follows: "computer software" or "operating system" or "computer program" or "software algorithm" or "data processing" or "software application". The error percentages are 14.2 and 7.7 for non-software trademarks that were included as software (Type II) and software trademarks that were not detected by the search algorithm (Type I) respectively. I compute the cumulative number software patents and software trademarks at a 15% annual discount rate in the same way I computed OSS portfolio.

Table 3: Descriptive Statistics

	Obs	Mean	Std.Dev.	Min	Max	1	2	3	4	5	6
1 *TobinsQ*	490	0.397	0.639	-1.434	2.371	1.000					
2 *OssPortfolio*	490	0.588	1.050	0	4.208	-0.168	1.000				
3 *SoftwarePatents*	490	5.442	2.249	0	9.456	-0.014	0.499	1.000			
4 *SoftwareTrademarks*	490	1.955	1.287	0	5.014	0.117	0.592	0.600	1.000		
5 *FirmSize*	490	10.145	1.392	6.446	12.895	-0.379	0.275	0.446	0.292	1.000	
6 *FirmAge*	490	3.639	0.818	1.098	5.069	-0.298	0.040	0.136	0.112	0.434	1.000

Table 2 displays the distribution of average OSS product portfolio sizes, software patents, and software trademark stocks by industry. Out of the 70 firms in the sample, 28 introduced at least one OSS product to the market during the 7-year time span. The descriptive statistics and pairwise correlations for the main variables appear in Table 3.

2.4 Empirical Analysis and Results

After conducting a Hausman test, I run fixed effects model with Tobin's q as the dependent variable to proxy for firm value. I incorporate interaction terms between

[4] See Fosfuri et al. (2008) for more detail about the accuracy of the algorithm.

OssPortfolio and the two variables that correspond to firms' IPR protection mechanisms, *SoftwarePatents* and *SoftwareTrademarks*. The full model is as follows:

$$(1) \quad TobinsQ_{it} = \beta_0 + \beta_1\, OssPortfolio_{it} + \beta_2\, SoftwarePatents_{it} + \beta_3 SoftwareTrademarks_{it} + \beta_4 FirmSize_{it} + \beta_5 FirmAge_{it} + \beta_6 OssPortfolio_{it} * SoftwarePatents_{it} + \beta_7 OssPortfolio_{it} * SoftwareTrademarks_{it} + \sum_{y=2003}^{2009} \alpha_y d_y + \varepsilon_{it},$$

where the subscripts *i* and *t* denote firm and year-specific observations, respectively. The dummy variable d_y indicates year effects. Firm-level characteristics, such as size and age, are also included in the model. I used the natural logarithms of all the variables.

The results from the first model with mean-centered interactions in Table 4 reveal that although the coefficient for *OssPortfolio*SoftwareTrademarks* is significantly negative, that for *OssPortfolio*SoftwarePatents* is significantly positive. On the other hand, coefficients for the direct effects of *OssPortfolio*, *SoftwareTrademarks* or *SoftwarePatents* are not significant. Hence, the findings confirm both *H1* and *H2*. Despite the rising trend in OSS commercialization by for-profit firms, relying solely on an open business model remains too risky in competitive markets. Appropriability mechanisms, such as software patents and software trademarks, appear to be crucial for firms that are willing to commercialize OSS.

I next estimate a second model to clarify the impact of firms' IPR protection mechanisms on the relationship between OSS product portfolio and firm value. I split the data for firms that commercialize OSS into two, as *LowOssPortfolio* and *HighOssPortfolio*, depending on the number of OSS releases of firms. Then I create the mean-centered interaction terms with *SoftwarePatents* and *SoftwareTrademarks* variables as before. The second model is as follows:

$$(2) \quad TobinsQ_{it} = \beta_0 + \beta_1\, LowOssPortfolio_{it} + \beta_2\, HighOssPortfolio_{it} + \beta_3\, SoftwarePatents_{it} + \beta_4 SoftwareTrademarks_{it} + \beta_5 FirmSize_{it} + \beta_6 FirmAge_{it} + \beta_7 LowOssPortfolio_{it} * SoftwarePatents_{it} + \beta_8 LowOssPortfolio_{it} * SoftwareTrademarks_{it} + \beta_9 HighOssPortfolio_{it} * SoftwarePatents_{it} + \beta_{10} HighOssPortfolio_{it} * SoftwareTrademarks_{it} + \sum_{y=2003}^{2009} \alpha_y d_y + \varepsilon_{it},$$

where the coefficients β_7–β_{10} measure the impacts of interactions between firms' OSS portfolios at different levels and their IPR protection mechanisms.

The results of the second model indicate that the interaction term between the high levels of OSS portfolio and software patents *(HighOssPortfolio*SoftwarePatents)* is significantly positive, while that for the interaction between the low levels of OSS portfolio and software patents (*LowOssPortfolio*SoftwarePatents)* is insignificant. A similar result is observed for the interaction terms between low and high levels of OSS portfolios and software trademarks, suggesting that high levels of OSS portfolio are largely responsible for the findings. The inclusion of the interaction terms has improved the model's explanatory power both in Model 1 and Model 2 as can be observed by looking at the adjusted R-squared values in Table 4. One may infer that while the marginal OSS released in interaction with software patents/software trademarks is associated with significant increase/decrease in Tobin's q, the decision of whether to commercialize OSS or not does not seem to have a significant effect without the right set of IPR mechanisms.

Specifically, firms taking more commercial actions while operating according to OSS paradigm can achieve a higher firm value when they possess complementary IPR protection mechanisms, such as software patents. While commercial OSS releases ensure diffusion and create a marketplace for complementary proprietary products and services, software patents help firm reap benefits. On the other hand, firms with large software trademark stocks experience decrements in firm value when they seek to serve their customers with OSS-based solutions. Open source solutions, which require skills for usage and maintenance, may not meet the needs of their established customers, who are willing to pay a premium for a ready-to-use package. Hence, OSS commercialization may cannibalize such firms' existing proprietary lines of businesses that are highly linked to their brand investments.

Table 4: Fixed Effects Model

VARIABLES	Baseline	Model 1	Baseline	Model 2
OssPortfolio	0.003 (0.115)	-0.021 (0.122)		
LowOssPortfolio			-0.040 (0.147)	-0.045 (0.160)
HighOssPortfolio			0.003 (0.116)	-0.024 (0.131)
SoftwareTrademarks	0.027 (0.049)	-0.012 (0.054)	0.025 (0.049)	-0.008 (0.055)
SoftwarePatents	0.034 (0.049)	0.067 (0.054)	0.034 (0.049)	0.061 (0.055)
*Oss*Sw_Trademarks*		-0.151** (0.072)		
*Oss*Sw_Patents*		0.104** (0.046)		
*Low_Oss*Sw_Trademarks*				-0.197 (0.123)
*High_Oss*Sw_Trademarks*				-0.141* (0.076)
*Low_Oss*Sw_Patents*				0.118 (0.107)
*High_Oss*Sw_Patents*				0.094* (0.049)
FirmSize	-0.125* (0.070)	-0.132* (0.071)	-0.123* (0.071)	-0.131* (0.071)
FirmAge	-0.131 (0.226)	-0.180 (0.222)	-0.138 (0.224)	-0.195 (0.221)
Year dummies				
_It_2003	0.358*** (0.074)	0.341*** (0.074)	0.353*** (0.075)	0.335*** (0.074)
_It_2004	0.330*** (0.066)	0.323*** (0.066)	0.326*** (0.067)	0.318*** (0.066)
_It_2005	0.347*** (0.059)	0.345*** (0.059)	0.343*** (0.060)	0.341*** (0.059)
_It_2006	0.325*** (0.052)	0.328*** (0.052)	0.322*** (0.052)	0.325*** (0.052)
_It_2007	0.261*** (0.048)	0.264*** (0.048)	0.259*** (0.048)	0.261*** (0.048)
_It_2008	-0.126** (0.053)	-0.125** (0.052)	-0.127** (0.053)	-0.126** (0.052)
Constant	1.931* (1.113)	2.173* (1.130)	1.710 (1.082)	2.226** (1.131)
Observations	490	490	490	490
Adj R-squared	0.840	0.842	0.839	0.841
R-squared	0.402	0.413	0.402	0.413
Number of firms	70	70	70	70

Notes: Robust standard errors are in parentheses (*** $p<0.01$, ** $p<0.05$, * $p<0.1$).
Mean-centered values of the predictors have been used.

Graphical analysis offers a visual explanation of the interactive effect between IPR protection mechanisms and OSS releases. Three values of software trademark stocks, which correspond to the 25th, 50th and 75th percentiles respectively, were selected as representative of the range of software trademark stocks included in the sample. For OSS portfolio, on the other, the representative values correspond to 5th, 25th, 50th, 75th and 95th percentiles.[5] In Figure 1, there is one regression line for each of these software trademark portfolios. The smallest stocks of software trademarks (software trademarks = 25th percentile value) showed a very small decrease in Tobin's q as OSS portfolio value increases from its value at 5th to 95th percentiles. At medium level software trademark stocks (software trademarks = 50th percentile value) showed a rather clearer decrease in Tobin's q as OSS portfolio becomes larger. The largest stocks of software trademarks (software trademarks = 75th percentile value) showed a notable decrease in Tobin's q with increasing values of OSS portfolio. Increases in software trademark stocks and OSS portfolio predicted a significant decrease in Tobin's q. Furthermore, it is possible to observe that as the level of software trademark stocks increases the relationship becomes stronger. In Figure 2, each regression line can be interpreted as a representative for three different levels of software patent stocks that correspond to the values at 25th, 50th and 75th percentiles. The graph showed that increases in software patent stocks and OSS portfolio predicted a significant increase in Tobin's q. Moreover, the statistically significant interaction suggests that this relationship is stronger for firms with larger software patent stocks. These results reflect the importance of aligning OSS commercialization with IPR protection mechanisms. These results imply that an optimal portfolio of IPR protection mechanisms for a firm, who aim to reap the largest benefits from a large portion of OSS portfolio, would be the one with large stocks of software patents and relatively small stocks of software trademarks.

[5] In Figure 1 and Figure 2, the minimum value for OSS portfolio is below zero because the representative values were obtained based on the mean-centered values of the explanatory variables that I used to compute the interaction terms.

Figure 1

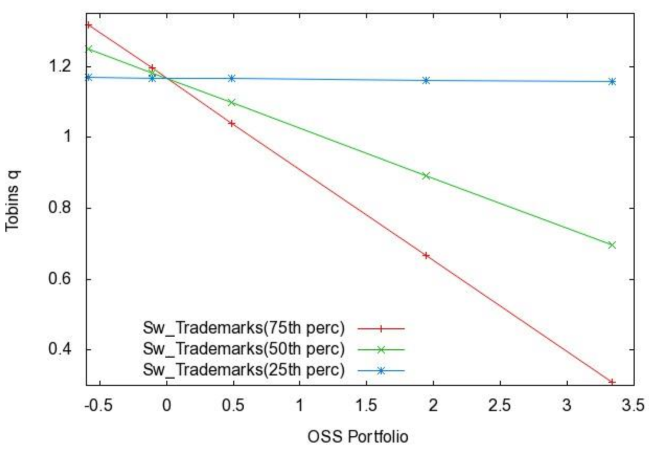
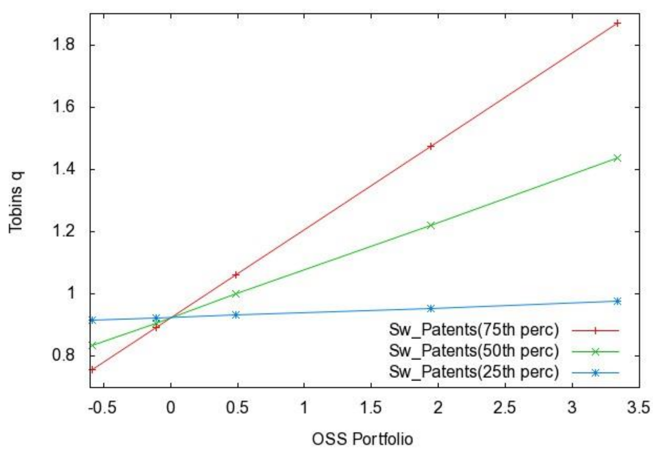


Figure 2



In order to alleviate the concerns that the results are being driven by the choice of Tobin's q as a measure of performance, I re-run the analysis using gross profit margin as an alternative dependent variable.[6] The results that I obtain from first and the second model using the alternative measure of performance remain unchanged in terms of directionality. However, there are some differences across coefficients on the direct and indirect effects in terms of significance. While, OSS portfolio, software trademarks and software patents seem to have a positive significant effect on performance measured through gross profit margin, the interaction terms created between OSS portfolio and software patents/trademarks are insignificant. As has been discussed earlier, the rationale for using Tobin's q is due to its ability to capture future performance potential. Bharadwaj, Bharadwaj and Konsynski (1999) argue that they use Tobin's q as a dependent variable in their study because IT investments contribute to long-run firm performance and to firm intangible value. In this study, I aim to investigate how firms reposition their IPR mechanisms with intentions to align better these assets with their business model. Since the focus of the study is on OSS commercialization, which can be seen as a long-term business model building activity of a firm, I believe using Tobin's q as a forward-looking measure of performance is well-suited to the aims of this study. Moreover, Tobin's q is risk-adjusted and less prone to manipulation compared to accounting measures of performance (Montgomery and Wernerfelt, 1988).

[6] Gross profit margin, also known as profit margin, is a commonly used accounting measure of performance that can be calculated by the following simple formula: Gross profit/Net sales.

2.5 Discussion and Conclusion

This study has two key findings. First, there is no direct positive effect of OSS portfolio on firm value. Thus, one can assert that the profitability of OSS commercialization is not as straightforward as that of proprietary software commercialization. In particular, appropriating returns from OSS commercialization cannot be achieved without complementary assets in proprietary segments since protecting the *commons* is the main challenge. Second, I assess the importance of firms' IPR protection mechanisms, and specifically software patents and software trademarks. OSS is a special case, which challenges classical appropriation mechanisms due to its open nature. However, I observe indirect positive/negative effects of software patents/software trademarks on the relationship between OSS and firm value. Specifically, I find that large software patent stocks help firms perform better when they commercialize open source products. On the other hand, large software trademarks have a negative effect on the relationship between OSS product portfolio and firm value. These results confirm the importance of matching a firm's appropriation mechanisms with its adopted business model. Software patenting works as a protection mechanism for complementary proprietary developments enabling firms appropriate returns from OSS. Software trademarks instead have a negative effect on the aforementioned relationship. Especially, firms with large stocks of trademarks may be at risk of jeopardizing their established proprietary developments as they introduce more OSS products to the market.

The findings contribute to the current debate on commercialization of OSS, mainly, by establishing the intermediary effect of firms' IPR holdings on the relationship between OSS commercialization and performance. Although there are studies that investigate the link between OSS and performance (Stam, 2009; Alexy and George, 2013), the moderating role of IPR mechanisms needed to be investigated as they are crucial firm resources in appropriating returns. Extending the work by Fosfuri et al. (2008), which emphasizes the role of these IPR mechanisms in firms' decision to engage in OSS, this study sheds light on how much these mechanisms matter for performance consequences of doing OSS. Moreover, the study empirically shows how mismatching strategies may harm a firm's performance while the firm attempts to move towards a new business model for better outcomes.

The transformation of OSS into a commercial strategy has obliged firms to reconsider their current business models. In doing so, managers must realize how much the fit of their IPR protection mechanism with their business model matters for value capture. Successful commercialization techniques combined with appropriate choices of intellectual property endowments can enable firms to capture the value created through open innovation models. Firms that aim to utilize advantageous attributes of OSS in order to create and appropriate value have to overcome the challenge of adjusting their respective resources in a way that is consistent with their current business model. Considering the findings of this study one might expect a firm, which commercializes OSS, to register for less software trademarks and to invest heavily in software patenting. My findings, which favor software patenting rather than trademark filing in an open source setting, further indicate that the adequacy of a firm's appropriation mechanism for its business model determines how much value it can capture. In this case, OSS commercialization interestingly may harm firms if they make wrong choices in their appropriation mechanisms.

Insufficient information about the development processes for the commercial OSS products and the licensing schemes under which they are released prevented me from undertaking a deeper analysis on the impact of OSS commercialization on firm value. It would have been helpful to know whether each product introduced to the market is a direct complement of a formerly introduced product of the same company (e.g., plug-in, extension). Another possibility would be examining the effect of copyright, as a potential isolating mechanism, on the relationship between new OSS product and firm performance. I believe the management model of the community and different firm participation strategies may also help us explain the success/failure of the outcome OSS product to be commercialized. The insights put forward in this study should encourage researchers, who are willing to investigate the profitability of OSS commercialization. I leave further investigation on potential explanatory variables, such as underlying licensing schemes, development processes or governing methods, which may explain the heterogeneities in returns generated by OSS commercialization, for future research.

2.6 References

Alexy, O., George, G. 2013. Category Divergence, Straddling, and Currency: Open Innovation and the Legitimation of Illegitimate Categories, *Journal of Management Studies* 50(2): 173–203.

Alexy, O., Reitzig, M. 2013. Private–collective innovation, competition, and firms' counterintuitive appropriation strategies. *Research Policy* 42(4): 895–913.

Allison, J. R., Lemley, M. A. 2000. Who's Patenting What? An Empirical Exploration of Patent Prosecution. *Vanderbilt Law Review* 58, 2099-2148.

Allison, J. R., Tiller, E. H. 2003. Internet Business Method Patents. In W. M. Cohen and S. A. Merrill (Eds.), *Patents in the Knowledge-Based Economy*, National Research Council, Washington: National Academies Press, 259-284.

Barney, J., 1986. Strategic Factor Markets: Expectations, Luck, and Business Strategy. *Management Science* 32(10), 1231-1241

Behlendorf, B., 1999. Open source as a business strategy. In Dibona, C., Ockman, S., Stone, M. (Eds.), *Open-Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly, 149-170.

Bessen J., Hunt, R. M. 2007. An Emprirical Look at Software Patents. *Journal of Economics & Management Strategy* 16(1), 157-189.

Bharadwaj, A. S., Bharadwaj, S. G., Konsynski, B. R. 1999. Information Technology Effects on Firm Performance as Measured by Tobin's q. *Management Science* 45(7), 1008-1024.

BIE (Bureau of Industry Economics), 1994. The economics of patents. Occasional Paper 18. AGPS, Canberra, January.

Blind, K., Cremers, K., Mueller, E. 2009. The influence of strategic patenting on companies' patent portfolios. *Research Policy* 38(2), 428-436.

Bonaccorsi, A., Giannangeli, S., Rossi C. 2006. Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science* 52(7), 1085-1098.

Bonaccorsi, A., Rossi C. 2003. Why OSS can succeed. *Research Policy* 327, 1243-1258..

Chung, K. H., Pruitt, S. W. 1994. A simple approximation of Tobin's q. *Financial Management* 23(3), 70-74.

Chesbrough, H. W. 2003. Open Innovation: The New Imperative for Creating and Profiting from Technology. *Harvard Business School Press*, Boston, MA.

Dahlander, L. 2005. Appropriation and appropriability in open source software. *International Journal of Innovation Management* 9(3), 259-285.

Dahlander, L., Magnusson, M. G. 2005. Relationships between open source software companies and communities: observations from Nordic firms. *Research Policy* 34, 481–493.

Feller, J., Fitzgerald, B. 2002. *Understanding Open Source Software Development*. Addison Wesley, Boston, Mass.

Fitzgerald, B. 2006. The transformation of open source software. *MIS Quarterly* 30 (3), 587-598.

Fosfuri, A., Giarratana M., Luzzi A. 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science* 19, 292-305.

Goldman, R., Gabriel, R. P. 2005. *Open Source as Business Strategy: Innovation Happens Elsewhere*. San Francisco, CA: Morgan Kaufmann.

Graham, S.J.H., Mowery, D.C., 2003. Intellectual property protection in the U.S. software industry. In W. M. Cohen and S. A. Merrill (Eds.), *Patents in the Knowledge-Based Economy*, National Research Council, Washington: National Academies Press, 219-58.

Hall, R. 1992. The strategic analysis of intangible resources. *Strategic Management Journal* 13(2), 135-144.

Hall, B., MacGarvie M. 2010. The private value of software patents. *Research Policy* 39(7), 994-1009.

Hall, B.H., Ziedonis, R.H., 2001. The patent paradox revisited: an empirical study of patenting in the U.S. semiconductor industry, 1979–1995. *RAND Journal of Economics* 32, 101-128.

Hecker, F., 1999. Setting up shop: The business of open-source software. *IEEE Software* 16(1), 45-51.

Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35 (7), 953–969.

Lakhani, K. R., von Hippel, E. 2003. How open source software works: “Free” user-to-user assistance. *Research Policy 32*(6), 923-943.

Lakhani, K. R., Wolf, R. G. 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S. Hissam, and K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. Cambridge, Mass: MIT Press, 3-22.

Lerner, J., Tirole J. 2002. Some simple economics of open source. *Journal of Industrial Economics* 50(2), 197-234.

Lerner, J., Tirole, J. 2005. The scope of open source licensing. *Journal of Law, Economics and Organization* 21, 20-56.

Lindenberg, E. B., Ross, S. A. 1981. Tobin’s q ratio and industrial organization. *Journal of Business* 54 (1), 1-32.

Mann, R. J. 2006. Commercializing open source software: Do property rights still matter? *Harvard J. Law Technology* 20, 1–46.

Mendonça, S., Pereira, T., Godinho, M. 2004. Trademarks as an indicator of innovation and industrial change. *Research Policy* 33, 1385-1404.

Montgomery, C., Wernerfelt, B. 1988. Diversification, Ricardian rents, and Tobin's q. *RAND Journal of Economics* 19, 623-632.

O’Mahony, S. 2003. Guarding the commons: How community managed software projects protect their work. R*esearch Policy* 32(7), 1179–1198.

O'Mahony, S., Ferraro F. 2007. The emergence of governance in an open source community. A*cademy of Management Journal* 50(5), 1079–1106.

Raymond, E. S., 1999. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly, 279.

Rumelt, R. P. 1984. Towards a strategic theory of the firm. In Competitive Strategic Management, Robert Boyden Lamb (Ed.), Englewood Cliffs, NJ: Prentice-Hall, Inc.

Schmidth, K. M., Schnitzer M. 2003. Public subsidies for open source? Some economic policy issues of the software market. *Harvard Journal of Law & Technology* 16(2), 474-502.

Shah, S. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science* 52(7), 1000-1014.

Stam, W. 2009. When does community participation enhance the financial and innovative performance of open source companies? *Research Policy* 38, 1288-1299.

Teece, D. J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15, 285-305.

Teece, D. J. 1998. Capturing value from knowledge assets: The new economy, markets for know-how, and intangible assets. *California Management Review* 40(3), 55-79

West, J. 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32(7), 1259–1285.

West, J., Gallagher, S. 2006. Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management* 36(3), 319-331.

Zu Knyphausen-Aufseß, D., Schweizer, L. 2011. Industry evolution and the interplay between extrinsic and intrinsic motivation. Software and Genomics from a Habermasian perspective. In P. Conrad und J. Sydow *(Eds.), Managementforschung 21.* Gabler-Verlag: Wiesbaden, S., 215-247.

Chapter 3 - Does it pay to be Open? Stock Market Responses to Contributions to the Commons

3.1 Introduction

> *"Of all the business reasons [for participating in open source software development]… design help may be the most important. Many software products fail because they do not meet the needs of their intended users." (Goldman and Gabriel, 2005, p. 80)*

Open source software projects such as Linux, Eclipse, and Mozilla have achieved remarkable success. These projects provide participants with the social context and resources to create useful and publicly available software that has, on occasion, displaced commercially produced software. The projects are distinctive from firms in that they allow others to access the knowledge they create rather than restricting access to key ideas and insights. These open source software projects are exemplars of a fundamentally different organizational model for innovation and product development—referred to as collective invention (Allen, 1983), private-collective innovation (von Hippel and von Krogh, 2003), and community-based innovation (Franke and Shah, 2003). This model extends well beyond the domain of software: innovative communities have been influential in fields as diverse as astronomy (Ferris, 2002), automobiles (Franz 2005, Kline and Pinch, 1996), sports equipment (Franke and Shah, 2003; Shah 2005; Baldwin, Heinerth and von Hippel, 2007), personal computers (Freiberger and Swaine, 2000), and video games (Jeppesen and Molin, 2003).

Innovation communities attract a variety of participants, ranging from hobbyists to firms. A number of firms have begun to engage actively and deeply with open source communities. For example, IBM currently employs hundreds of full-time employees to work within various open source development communities such as Linux and Apache; Netscape initially formed Mozilla project, which has then became very well known for producing Firefox web browser; Nokia and Oracle have contributed both software and developer time to the Eclipse Foundation. These companies are recognized not just for building products based on open source software and hardware, but also for contributing to the commons. This behavior is perplexing to a number of innovation scholars. In this paper, we seek to investigate if firms benefit from contributing to the commons *and* whether or not differences in *what* a firm contributes affects the value of the benefits that firms generate from their contributions.

Several practitioners and scholars have suggested that participating in an innovation community allows firms to access unique knowledge and insights that would be difficult or impossible to access through other means, arguing that insights provided by a wide range of others who possess heterogeneous knowledge bases and use experiences can provide immense benefits to corporate product development efforts (Winston Smith and Shah, 2013; Goldman and Gabriel, 2005; Chatterji and Fabrizio, 2012; Hargrave and Van de Ven, 2006). If this is indeed the case, we should observe value being generated for firms when they work collaboratively within communities. However, communities—like university labs or firm R&D departments—are a particular type of innovation engine and function in particular ways. Hence, engaging development assistance from the community-at-large may be more beneficial for some tasks than for others. We build on existing empirical findings to develop theory, which suggests that firms might most effectively leverage knowledge and gain insights by working with user communities and we provide empirical support for our arguments. Specifically, we empirically examine the following three questions. First, do firms benefit from contributing to innovation communities? Second, does code that contributes to the creation of novel software generate greater returns than contributions of existing software? And, third, do contributions of novel software to projects focused on building software for end-users, as opposed to developers, generate greater benefits?

To address these questions, this paper examines stock market responses to press releases announcing contributions of open source software code by publicly–listed, U.S. firms. Data from press releases are systematically coded and analyzed using an event study model (Austin, 1993). This approach provides a tangible measure of innovation-related benefits and has been used in the marketing and strategy literatures to estimate the value generated by a variety of innovation-related activities (Chaney, Devinney and Winer, 1991; Koku et al., 1997; Fosfuri and Giarratana, 2009; Sood and Tellis, 2009; Srinivasan, Pauwels, Silva-Risso and Hanssens, 2009).

Three key findings are uncovered. First, contributions to the commons provide tangible benefits to firms, as reflected in positive abnormal stock market returns to code contribution announcements. Second, contributions of more novel code generate greater positive abnormal stock market returns. Third, contributions of novel code targeted towards end-users appear to drive the positive stock market response to firms' contributions to the commons.

3.2 Do Open Source Code Contributions Generate Value for Firms? When?

Firms have contributed millions of lines of source code to open source software communities. For example, Sun Microsystems contributed over 9 million lines of Star Office software code (Sun Microsystems Inc., 2000), IBM contributed over half a million lines of relational database code to the Apache Software Foundation (IBM Corp., 2004), and Novell donated more than 200,000 lines of source code that formed the basis of Hula, a new community project (Novell Inc., 2005). These and similar actions have triggered curiosity within the academic community and the realization that participation in innovation communities may provide firms with tangible benefits.

The literature on firm-participation in community-based innovation is just emerging. To date, this literature has focused largely on understanding how firms *appropriate value* from their contributions to innovation communities. These studies suggest that value can indeed be appropriated, largely through the use of mechanisms outlined in Teece's seminal (1986) work: complementary assets (Lerner and Tirole, 2002), patents and trademarks (Fosfuri, Giarratana and Luzzi, 2008), and secrecy/selective revealing (Henkel, 2006).[7] Benefitting from innovation, however, requires both generating value and appropriating value (Teece, 1986). To our knowledge, few, if any, studies have sought to unpack if and how firms actually benefit from contributions to innovation communities. As a result, key strategic questions remain: *do firms benefit from contributions to user innovation communities?* and, *what*

[7] Although these findings are suggestive of the idea that firms might derive innovation-related benefits by working with communities to develop product features (innovations), empirical work examining value generation is needed. For example, Fosfuri et al. (2008) document that firms with greater stocks of intellectual property are more likely to commercialize open source software, arguing that firms that commercialize OSS might then subsequently rely on their stocks of software patents to profit from these innovative developments. However, it is also possible that firms with greater innovative ability are *both* more likely to engage in open source software development and have existing intellectual property stocks as a result of their proprietary innovative activities. Similarly, Henkel (2006) documents that firms are selective about what they reveal. However, it is not clear if the community subsequently play a role in further developing the information that the firm reveals—that is, in value generation.

types of contributions lead to the greatest benefits from firms? We seek to investigate these questions.

3.2.1 Insights from Innovative Users

Innovation is a time consuming, difficult, and challenging task. The success rate of corporate product development projects is low (Cooper and Kleinschmidt, 1995; Taylor, 2010). Empirical findings from across the innovation management literature suggest that accessing and integrating knowledge from product users is one way in which firms can improve their chances of product development success (Griffin and Hauser, 1993). The user innovation literature goes one step further, arguing that because the majority of potential customers do not necessarily recognize or have unmet product needs, market research that relies on information collected from the "average" potential customer will not necessarily uncover insights that lead to the development of revolutionary new products. Instead, this literature suggests focusing on a small fraction of innovative individuals—user innovators—who do experience needs unfulfilled by existing products and services and create innovative prototypes that fulfill those needs (von Hippel, 1988).

Knowledge of needs and features from the community of users is the key benefit that might accrue to a firm by contributing source code to the development of a public good. Most important benefit suggested by two senior research scientists at Sun Microsystems—an early and prolific experimenter with open source systems—is design benefits (Goldman and Gabriel, 2005) "Of all the business reasons given here, design help may be the most important. Many software products fail because they do not meet the needs of their intended users..." (Goldman and Gabriel, 2005, p. 80). Empirical studies find that the knowledge generated by innovative users is distinct from that generated by other sources of innovations and might provide firms with valuable innovative insights. In the context of scientific instruments, user innovations tend to embody altogether new product functionality, whereas producer innovations tend to improve existing product functions along dimensions known to be important to customers (Riggs and von Hippel, 1994). In the context of medical devices, firms engaging in corporate venture capital investment are more likely to incorporate innovative insights from user-founded startups than from employee- or academic-founded startups (Winston Smith and Shah, 2013). Learning from user communities support the product development process of a firm and make the growth trajectory of this process visible to its investors. Such collaborations will benefit a firm substantially by complementing its internal knowledge base with scientific, artistic, or technological knowledge (Hargrave and Van de Ven, 2006).

Collaborations with communities that provide knowledge, which firms cannot develop internally, are essential for technology development. Especially for small firms with limited resources, new product development may not be possible (Goldman and Gabriel, 2005, p.84-85). The knowledge and insights of a community of users and developers are critical resources especially for such firms. These firms, by opening up their innovation processes, may gain access to several resources outside the boundaries of the firm. Moreover, in community projects resource allocation is more efficient through self-assignment of tasks by community members since contributors know their skills and capabilities better than anyone else (Benkler, 2002). Hierarchy and politics, which tend to create a tension in an in-house project, are less of an issue for community projects (Goldman and Gabriel, 2005, p. 181-182).

In summary, engaging users may provide firms with several advantages that improve their ability to develop innovative new products and thereby generate value: innovative users provide insights into novel and needed functionality (problem identification),[8] possess knowledge that firms may not have regarding problem solutions, and may increase word of mouth knowledge of the product.

Hypothesis 1: Open source code contributions will generate value for firms.

As described above, the insights of external users can benefit firms' product development efforts. Open source software communities attract individuals interested in *using* a particular piece of software: some individuals may find that the software, as it stands, fulfills their needs, while others will need to modify the software to fulfill their needs. Individuals will therefore contribute to and modify a piece of software largely on an "as-needed" basis (Dahlander and Magnusson, 2005). However, we argue that the value that a firm derives from the insights of open source participants varies based on the strategic reasons for contributing code. For example, when code is contributed to gather insights regarding desired functionality, a firm might alter the features and functionality of a subsequent product offering such that the refined product is far

[8] Thereby both ensuring/vetting the usefulness of a product and potentially increasing the range of what the product can be used for.

superior to the product that might have been released without insights from the open source community (Goldman and Gabriel, 2005, p.86-87). When code is novel and has not yet been introduced on the commercial market, such insights are likely to be very valuable for firms as features have yet to be developed and refined. Garnering insights from the open source community at early stages of the development phase can help prevent product failure in commercial market (Raymond, 2001). This knowledge provides the firm with an advantage in the product market.

Contributing updates to code that has already been released in proprietary or open source form may provide the firm with some benefits, in the form of standardization of the code which other participants are downloading, using, and modifying and/or feedback on new features released as part of the update. As such, firms may gain some innovative insights through the contribution of updates. However, the innovative benefits gained from such releases are likely to be less than the innovative benefits garnered from the release of altogether novel code. Releasing novel code is more likely to result in a new product introduction, which will ensure a long-term stream of cash compared to updates, and, thus, will bring larger positive abnormal returns (Chaney, Deviney and Winer, 1991).

In contrast, firms also contribute code for a variety of reasons related to cost reduction, rather than innovation (Goldman and Gabriel, 2005; Bonaccorsi, Gianneli and Rossi, 2006; Dahlander and Magnusson, 2005). For example, firms might contribute code for products that they no longer wish to support (Goldman and Gabriel, 2005, p.90)[9] or as a means of offering a discontinued product development project a second life outside the firm as a means of motivating (or not demotivating, as the case may be) valued employees who invested significant effort and time into the project (Goldman and Gabriel, 2005, p.85). Such contributions signal the termination of a commercial product line or product development project, and hence reflect a decline in value for the firm.

In summary, contributions of novel code—code that has not been released—are more likely to be made by firms seeking innovative insights, whereas contributions of

[9] An example of this is Cisco System's decision to open source their Cisco Enterprise Print System software (CEPS) so as not to have to maintain it in-house. Once the software is open sourced, even if the original developers leave the company their participation is ensured for a long time (Goldman and Gabriel, 2005).

existing code are more likely to be made by firms seeking to release themselves from development, support, or maintenance costs. Hence, we expect that contributions of novel code will generate more value for firms.

Hypothesis 2: Contributing novel code to open source software communities will generate greater value for firms than contributing updates or existing code.

A particular piece of software may be targeted towards end users (e.g., applications or operating systems) or developers (e.g., development tools or programming languages). We suggest that a firm may generate more value by contributing novel open source code intended for end-users than for developers, due to the viability of a commercial market for the code coexisting with the community. Commercial and community-based diffusion of products have been observed to occur in parallel in the sports equipment, probe microscopy, and segments of the software industries (Shah and Mody, 2013; Shah, 2005; Fosfuri et al., 2008). For communities and commercial markets to coexist, two sets of individuals must be present: community participants engaged in building, modifying, and maintaining their own copies of the product and commercial (or "mass market") buyers choosing to purchase standardized and tested versions of a product (Shah and Mody, 2013; Shah, 2005).

In the case of development tools and programming languages, one would expect virtually all participants to access the code freely and openly through the community rather than purchase the code commercially, because individuals interested in using development tools and programming languages tend to be technology-savvy. In contrast, end-users interested using applications or operating systems are a much more heterogeneous set with respect to their knowledge of software and how to use it: software users span the range from expert software developers to individuals struggling to simply use a software program. As a result, we expect both types of participants to be present, thereby supporting both a community and a commercial market. A firm will generate value from a novel code contribution by applying insights gained from community interaction to products sold in the commercial marketplace.

Hypothesis 3: Open source contributions consisting of new code will generate greater value for firms when the code is intended for use by end-users rather than developers.

3.3 Data and Methodology

Our goal is to determine whether or not firms derive value from engaging in open innovation activities. In order to do so, we deploy an event study methodology. The abnormal stock market returns are a well-regarded means of assessing the value generated by innovation (Sood and Tellis, 2009). Event studies gauge the cumulative value of an action to the firm by assessing abnormal returns triggered by press releases. Abnormal returns reflect investors' expectation of the value of an action; because investors are the arbiters of the financial value of the firm and its activities (Aggarwal, Dai and Walden, 2006), abnormal returns provide a consensus estimate of the value of a particular action to a firm.

Event studies are widely used in finance, accounting, marketing and management research (MacKinlay, 1997; McWilliams and Siegel, 1997; Kothari and Warner, 2004). A number of studies have used event studies to assess the benefits that firms receive from engaging in innovation activities (Austin, 1993). For example, event studies have been used to evaluate product development announcements (Sood and Tellis, 2009) and new product introductions (Chaney et al., 1991; Koku et al., 1997; Fosfuri and Giarratana, 2009; Srinivasan et al., 2009). We use an event study methodology to estimate the abnormal returns to firms' announcements of open source code contributions.

3.3.1 Study Setting: Open Source Software Development

Open source software provides a particularly useful context for examining our questions of interest for several reasons. First, the event study methodology rests on the efficient markets hypothesis, which proposes that the stock price of a company fully reflects all publicly available information about a company: in the open source software context, not only do firms issue press announcements regarding their contributions to innovation communities, but their participation and contributions can be observed by those within the community and those outside the community. Second, the context allows us to access the necessary data, as it is likely that most firm contributions to open source software communities are documented by press releases. This is because open source community contributions tend to occur in open and publicly accessible online forums, where it would be difficult for firms to attempt to hide their participation or delay announcements of their contributions. Hence, a firm's efforts to co-create with the community are visible.[10] Third, of all the settings in which community-based innovation has been documented and studied, the most research has been conducted in the context of the software industry. This allows us rely on a variety of empirical studies to add insights and rigor to our theory building and improves our ability to interpret results. Fourth, the software industry is generally free of regulation, meaning that individuals and firms interested in contributing to and using the software generated by community-based innovation development are able to do so.

3.3.2 Sample

The event is defined as an announcement to release source code to the public under an OSI-approved open source software license.[11] Our study covers a thirteen-year time span from January 1st, 1999 through December 31st, 2011. We draw data from three main sources: Marketwire, PR Newswire and Business Wire. These sources are the leading press wire services[12] through which firms release information to the market (Solomon, 2012) and are widely used in event studies pertaining to business-related issues (e.g. Alexy and George, 2013). We accessed these data sources through the LexisNexis database.

Because our focus is on gauging the impact of open source software development activities on firms' innovation activities, we focus on press releases in which firms announce the decision to reveal software *code.*[13] This allows us to restrict our analysis to situations where firms are actively engaged in developing innovative code in a community-based context. We use a multi-step approach to identify relevant

---

[10] In contrast, in other product domains, firms may pursue collaboration with communities in less transparent settings (e.g., by working with communities or particularly innovative community members in offline meetings, workshops, and conferences). In such settings, a firm's contributions to communities may be difficult to observe and a firm may choose to not announce their contributions publicly, making it difficult to access the data needed to conduct an empirical study. The accessibility of data may be the reason why OSS is the most common setting for investigating community-based innovation.
[11] The Open Source Initiative (OSI) is a public benefit corporation and the community recognized body for reviewing and designating licenses as Open Source Definition (OSD) conformant. See http://opensource.org/about.
[12] "Press wire services act as intermediaries between firms and the media. When a firm seeks to distribute a press release, it sends the release to a press wire service, which then redistributes the release to media partners. Most publicly traded firms in the United States rely on the two leading press wire services- PR Newswire and Business Wire (a subsidiary of Berkshire Hathaway)." (Solomon and Soltes, 2013)
[13] Firms also contribute to open source communities through sponsorship, marketing, coordinating and distributing activities. Although these contributions might generate value for the firm, they do not reflect a firm's engagement in innovation-related activities.

press releases. First, all announcements that include the subject term[14] "open source software" and the search terms ("releas*" or "reveal*" or "contribut*") and ("open sourc*" or "source code") in the body are retrieved. Second, we identify announcements pertaining to firms listed on the NYSE, NASDAQ or AMEX. More than 7000 announcements over a period of 10 years were identified in this way. Third, we systematically assessed whether or not each announcement fit our event definition. We established a set of rules by which to identify instances when a firm released software code under an OSI-compliant license and applied this set of rules to each announcement. Each announcement was coded twice to ensure that it fit the event definition. This resulted in the identification of 231 events from 84 different firms. Fourth, it is necessary to exclude events that coincide with confounding events when conducting an event analysis (McWilliams and Siegel, 1997). Confounding events include instances where a single company makes another announcement on the same day or one day before or after our focal event occurs. This led us to exclude 76 events.

As a final step, we dropped those 3 events with the highest abnormal returns and 3 with the lowest abnormal returns in order to alleviate the concerns that the results may be influenced heavily on few observations. Interpretation of significance in abnormal returns is problematic, especially for small samples (McWilliams and Siegel, 1997). Although our sample size may not be considered as small compared to other event studies with a similar empirical setting (e.g. Oh, Gallivan and Kim, 2006; Alexy and George, 2013), we delete the outliers in our main regressions to avoid potential biases. These precautions reduce the sample size significantly. However this is not a limitation: our sample size is well within the acceptable range for event studies. The final sample consists of 149 events from 64 firms.

3.3.3 Variables

Dependent Variable

[14] LexisNexis® subject and industry terms cover a wide range of topics including Banking & Finance; Medicine & Health; Computing & Information Technology; Law & Legal Systems; Trends & Events; and much more. Subject experts and information professionals create index terms and rules governing their application. These experts then iteratively test and analyze results before a term is released to production. Controlled vocabularies are applied to new documents as they are added to the system. Periodically, all indexed documents are retrospectively updated with the latest vocabularies. New terms are added regularly based on the need of a particular taxonomy and customer feedback.

We use cumulative abnormal returns (*CAR*s) to measure the value generated by firms by engaging in community-based innovation developed. Cumulative abnormal returns are the sum of the daily, irregular stock market returns abnormal in the days surrounding an event of interest. CARs have been used to measure the value generated from innovative activities in a number of studies (Fosfuri and Giarratana, 2009; Sood and Tellis, 2009; Srinivasan et al., 2009). They provide a particularly useful measure of innovation for researchers, because they are not subject to manipulation by environmental factors and they reflect how much investors value the new information (McWilliams and Siegel, 1997). CARs are a well-regarded measure of the value generated by innovation (Sood and Tellis, 2009).

Focal Variables

Our focal variables are the characteristics of the code, which is made available to the general public.[15] Specifically, we are interested in whether or not the code is new and the audience for which it is intended. We use two alternative approaches to proxy for newness of the code. The first approach involved creating three mutually exclusive, binary variables: *New Code*, *Updated Code*, and *Existing Proprietary Code.* The *New Code* variable takes a value of 1 if the code is altogether new to the market (i.e., the code has not been diffused in the past, either as open source or proprietary source code). The *Updated Code* variable takes a value of 1 if the code is a new version (update) of existing open source or proprietary source code. The *Existing Proprietary Code* variable takes a value of 1 if the code has been sold in proprietary form and is being released under an open source license for the first time.[16] The alternative approach involves creating an ordinal variable, *Novelty*, which takes the value 0 if *Existing Proprietary Code* variable is equal to 1, 1 if *Updated Code* variable is equal to 1, and 2 if *New Code* variable is equal to 1. Figure 1 depicts how code novelty variable is constructed.

[15] Each press release in the sample provides extensive information on the characteristics of the code contribution and the community to which it is being contributed. Systematic common rules were devised to code each announcement in accordance with each variable definition. In a very small number of instances, insufficient data was available through the press release; in such cases, additional research was conducted using firm and software community web pages to collect the needed information.

[16] Note that source code categorized as *Existing Proprietary Code* includes no new components/features, hence the market has had access to the functionality of the code in the past. Cases where proprietary code was updated with new open source components/functionality are classified as *Updated Code* (5 such cases appear in the sample).

Figure 1: Categorization of Code Novelty

	Licensing Model	
	Open Source	**Proprietary**
New code (n=40)	New Code	-- [a]
Updates to existing code (n=27)	Updated Code	Updated Code
Existing code (n=82)	-- [b]	Existing Proprietary Code

[a] Code that was previously released as proprietary cannot be new by definition. Hence this category represents a null set.
[b] Existing code that was already released under an open source license would not be re-released. Hence this category represents a null set.

We also create codes to identify the target users of the code: developers or end users. Code contributions pertaining to development tools are classified as being primarily for use by developers (*Development Tools*), whereas code contributions for system software or application software are classified as being intended for use by end users (*System or Application SW*). There are 3 contributions that included code both for use by developers and by end users. *Development Tools* and *System or Application SW* variables are not mutually exclusive since these 3 observations are classified in both categories in the sample.

Controls

We include four sets of control variables in the empirical analysis. The first set controls for characteristics specific to an announcement. *Pre-announcement* is a binary variable that identifies press releases that discuss code contributions to be made in the future (Koku et al., 1997).[17] *Pre-release* is a binary variable that identifies announcements of alpha or beta releases of software that will be released in full form in the future. We control for these two binary variables to avoid our results being driven by announcements that include forward-looking statements.

The second set controls for firm specific characteristics. We control for *Software Patents,* the total number of software patents that firm *i* has been granted with during the five years prior to time t, in order to account for firms' endowments of

[17] We classified announcements that described code to be released in the future as pre-announcements (all preannouncements specify the expected time frame in which code will be contributed; this time frame varies from 1 month to 6 months in our sample). We classified announcements that provided a link to the released code or clearly stated that the code had already been released as actual code contributions.

intellectual property rights (IPRs).[18] In addition, we control for *OSS Announcement Ratio* with which we aim to capture a firm's interest and efforts of engaging in OSS movement (Hannan et al., 2007; Alexy and George, 2013). We calculate *OSS Announcement Ratio* by dividing the number of press releases that include "open source software" in the subject term to the total number of press releases by firm *i* over the 1-year prior to time *t*.

The third set of controls includes five sector dummy variables to control for the industries in which firm *i* operates: hardware, software, electronics, semiconductors and telecommunications. By doing so, we aim to control for firms' technical expertise in developing software (Aggarwal et al., 2006). We determine these industries using SIC code classifications (Fosfuri et al., 2008).[19] Finally, we introduce year dummies to control for time-variant effects (Shiller, 2005; Alexy and George, 2013).

3.3.4 Analytic Method

The abnormal return to an event (announcement) is obtained in two steps. First we estimate the normal return, which is the expected return that would have been observed if the event did not take place, for the time period preceding the event. We follow standard practice by using the time period 160 to 5 days prior to the event (Campbell, Lo and MacKinlay, 1997). Normal returns are estimated via Fama-French 3-factor model (Fama and French, 1992)[20].

$$(R_{it} - Rf_t) = \alpha_i + \beta_{1i} (Rmkt_t - Rf_t) + \beta_{2i}\, SMB_t + \beta_{3i}\, HML_t + \varepsilon_{it}$$

where R_{it} is the return on shares of firm *i* over the event window *t*, Rf_t is the risk-free rate of return at time *t*, $Rmkt_t$ is the return on all firms in NYSE, AMEX and NASDAQ at time *t*, SMB_t is the index of small versus big capitalization at time *t* and HML_t is the

[18] We initially controlled for firm size in our analysis following other studies that also use an event study methodology to measure returns to innovation (e.g. Aggarwal et al., 2006; Alexy and George, 2013). The intuition behind controlling for firm size in those studies is that since a large firm is likely to be engaged in various lines of business, an announcement might affect only a small portion of the firm's overall value. However, the *Software Patents* and *Firm Size* variables were highly correlated, thus, we dropped *Firm Size* from our list of controls. The results remained unchanged.

[19] The SIC codes for hardware, software, electronics, semiconductors and telecommunication sectors are 357, 737, 36 (except for 367 and 366), 367, and 366, respectively.

[20] Available at http://mba.tuck.dartmouth.edu/pages/faculty/ken french/Data_Library/f-f_factors.html

index of high versus low book/price ratio at time *t*. Daily financial data are obtained from Bloomberg/Datastream terminal.[21]

Next we compute the abnormal return (AR_{it}) by taking the difference between the observed return on the day of the event and the normal return estimated via the benchmark model described above. Thus,

$$AR_{it} = R_{it} - E\,[R_{it}]$$

where AR_{it} is the abnormal return, $R_{it,}$ is the observed return and $E\,[R_{it}]$ is the normal return for announcement *i* and event window *t*. We compute the cumulative abnormal return (CAR) for each event by aggregating abnormal returns through the event window for each security. We limit our event window *t* to 2 days that include the day of the event and the day before the event occurs.[22]

We also investigate the effect of code characteristics on the magnitude of abnormal returns using a cross-sectional regression model:

$$CAR_{it} = \beta_0 + \beta_1\, New\ Code_{it} + \beta_2\, Updated\ Code_{it} + B_3\, Software\ Patents_{it} +$$
$$\beta_4\, OSS\ Announcement\ Ratio_{it} + \beta_5\, Pre\text{-}release_{it} + \beta_6\, Pre\text{-}announcement_{it} +$$
$$\textstyle\sum_{s=1}^{5} \beta_s d_s + \sum_{y=1999}^{2011} \beta_y d_y + \varepsilon_{it}.$$

where CAR_{it} is the cumulative abnormal return across time *t* for security *i*. *New Code* and *Updated Code* are binary variables that are used for identifying newness of the code. We keep *Existing Proprietary Code* variable as the control group in this model. *Software Patents* and *OSS Announcement Ratio* are used as firm-specific controls. *Pre-release* and *Pre-announcement* are binary variables that account for forward-looking statements in the body of the announcement. Dummy variables d_s and d_y indicate sector and year effects, respectively.

3.4 Findings

Table 1 provides descriptive statistics. Correlation coefficients are presented in Table 2. In line with our expectations, the dependent variable (CAR) is positively correlated with contributions of new code, updated code, and system or application

[21] Available on Bloomberg/Datastream terminal at Foster School of Business Library.
[22] It is expected that the significance of CARs diminishes as the event window enlarges (Campbell et al., 1997), suggesting that the effect of the event is stronger around the day of the event.

software; and negatively correlated with contributions of existing proprietary code and development tools. In line with existing empirical studies, pre-announcements and pre-releases are inversely related to the dependent variable (Koku, 2009).[23]

Table 1: Descriptive Statistics

	Obs	Mean	Std.Dev.	Min	Max
1 CAR	149	.788	3.075	-8.255	12.666
2 Novelty	149	.718	.863	0	2
3 New Code	149	.268	.445	0 (n=109)	1 (n=40)
4 Updated Code	149	.181	.386	0 (n=122)	1 (n=27)
5 Existing Proprietary Code	149	.550	.499	0 (n=67)	1 (n=82)
6 System or Application SW	149	.483	.501	0 (n=77)	1 (n=72)
7 Development Tools	149	.523	.501	0 (n=69)	1 (n=80)
8 Software Patents	149	217.732	434.911	0	3027
9 OSS Announcement Ratio	149	.066	.071	.003	.377
10 Pre-announcement	149	.181	.386	0 (n=122)	1 (n=27)
11 Pre-release	149	.094	.293	0 (n=135)	1 (n=14)

Table 2: Correlation Matrix

	1	2	3	4	5	6	7	8	9	10	11
1 CAR	1.000										
2 Novelty	0.129	1.000									
3 New Code	0.109	0.903	1.000								
4 Updated Code	0.037	0.154	-0.285	1.000							
5 Existing Proprietary Code	-0.126	-0.924	-0.670	-0.520	1.000						
6 System or Application SW	0.046	0.036	0.020	0.033	-0.044	1.000					
7 Development Tools	-0.012	-0.023	-0.015	-0.017	0.026	-0.960	1.000				
8 Software Patents	0.055	0.010	-0.097	0.244	-0.103	0.003	0.049	1.000			
9 Oss Announcement Ratio	-0.042	0.083	0.081	-0.001	-0.072	0.094	-0.071	-0.092	1.000		
10 Pre-announcement	-0.118	-0.069	-0.010	-0.131	0.110	0.033	-0.052	-0.035	-0.122	1.000	
11 Pre-release	-0.006	-0.001	-0.039	0.087	-0.033	0.057	-0.024	0.025	0.150	-0.152	1.000

Figure 2 and Figure 3 depict the distribution of events and distribution of firms contributing code across years respectively. The trend in Figure 2 does not show a consistent decline or increase. However, there are two declines in the number of announcements worth commenting upon. The first begins in 2001, following the burst of dot-com bubble. The second begins in 2007 and continues through 2008, in parallel

[23] Koku (2009) find that announcements with less detailed information have a smaller positive effect on a firm's stock market value than announcements with detailed information about the new product introduced: the latter send a more credible signal to the market by enabling investors to differentiate real contributions from "vaporware". "Vaporware" refers to non-existent products that a firm claims are "forthcoming" (Koku, 2009). A similar pattern is expected for those announcements that include a link to the website through which the code contribution is available and that do not include such information.

with the start of the global financial crises. Figure 3 depicts a more consistent and steady upward trend in the overall number of firms making open source code contributions each year. A decrease in the number of firms does follow the burst of dot-com bubble.

Figure 2: Open Source Code Releases Across Years

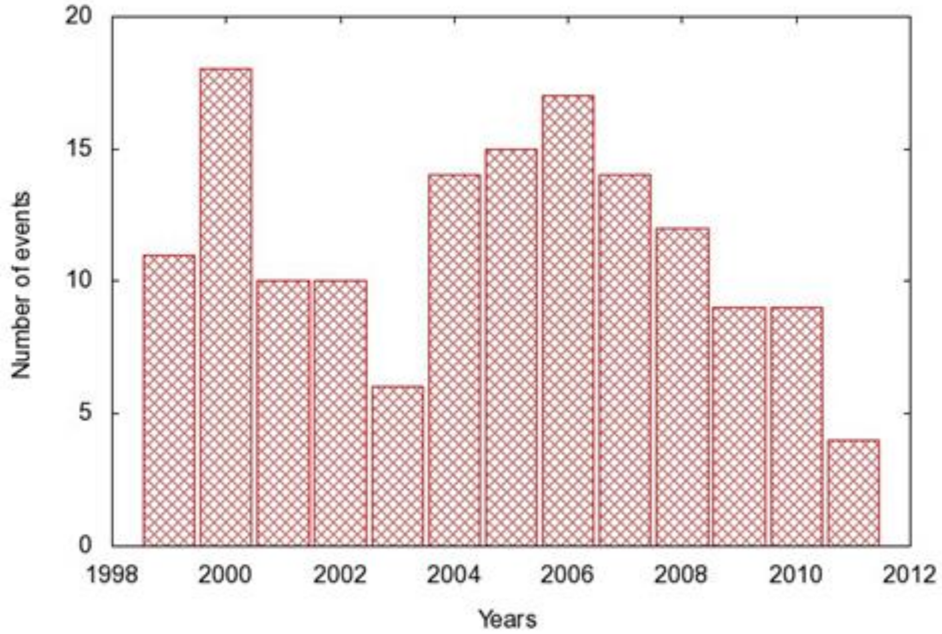


Figure 3: Number of Firms Releasing Open Source Code Across Years

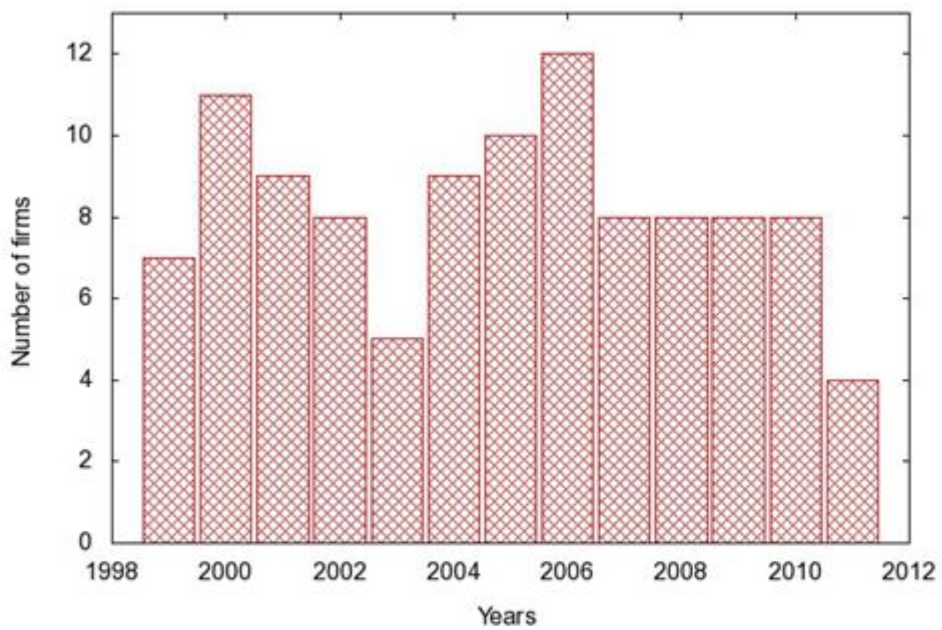


Table 3 presents the results of the cross section regression analysis in which the dependent variable is the CARs over a 2-day event window and where errors are clustered by firm.[24] The constant term in the first column (baseline model) reflects the CARs generated by code contribution events. The coefficient on the constant term remains positive and significant at the 0.1 level. In the second column we add two sets of control variables to account for announcement characteristics (*pre-release* and *pre-announcement)* and firm characteristics (*Software Patents* and *OSS Announcement Ratio)*. The coefficient on the constant term remains large and significant. These findings provide support for *Hypothesis 1:* on aggregate, investors' react positively to source code contributions. The results remain unchanged when we control for firm specific characteristics (*Software Patents*, *OSS Announcement Ratio*) or for the credibility of the announcement itself (*Pre-announcement*, *Pre-release*).

Finally, we introduce our focal variables as well as the two sets of controls that account for sector and time effects (Table 3, Column 3). In line with *Hypothesis 2*, which suggests that contributing novel code will generate greater value for firms than contributing updates or existing code, we observe that the coefficient on the *New Code* variable is significant at the 0.1 level and the coefficient on *Updated Code* variable is positive, but not significant. The results are in the same direction when we use the ordinary variable, *Novelty,* as an alternative measure to proxy for the newness of the code (Table 3, Column 6). The coefficient on *Novelty* is significant at the 0.1 level supporting *Hypothesis 2*.

In order to test for *Hypothesis 3*, we split the data into two sub-samples and use the same model to estimate coefficients on the variables that proxy for the newness of the code (Table 3, Columns 4 and 5). The first sub-sample includes those contributions that are intended for use by end-users and the second sub-sample includes those contributions that are intended for use by developers. While the coefficient on the *New Code* variable is large and significant at the 0.05 level for the sub-sample of end-user related contributions, its significance disappears for the sub-sample of code contributions that are intended for developers. It is possible to observe a similar trend when comparing the coefficient on *Updated Code* variable across two sub-samples.

[24] We undertake a test of significance for assessing the adequacy of using CARs as dependent variable in our linear regression model. We perform parametric t-test, which suggests that the average CAR value for the whole sample is statistically significantly different from zero at the 0.05 level.

While for the sub-sample of end-user related contributions it is positive, it becomes negative for the sub-sample of code contributions that are intended for developers. The findings are in support of *Hypothesis 3*, which suggests that, when the contribution is intended for use by end-users, the newer the code the greater benefits it will bring about. The results remain unchanged when we use the *Novelty* variable instead (Table 3, Columns 7 and 8). While the coefficient on the *Novelty* variable is large and significant at the 0.05 level in the column that represents the sub-sample of end-user related contributions, it is insignificant for the subsample of code contributions that are intended for developers. *Hypothesis 3* is supported in light of these findings.

Table 3: Results of OLS Regression

	Baseline Model		Main Results					
			Full sample	Sys/App SW	Dev. Tools	Full sample	Sys/App SW	Dev. Tools
VARIABLES	CAR	CAR	CAR	CAR	CAR	CAR	CAR	CAR
Novelty						0.617*	1.402**	-0.049
						(0.367)	(0.607)	(0.741)
New Code			1.290*	2.932**	0.119			
			(0.763)	(1.284)	(1.493)			
Updated Code			0.259	0.434	-1.381			
			(0.787)	(1.503)	(1.109)			
Software Patents		0.000	0.001	0.001	0.002	0.001	0.001	0.001
		(0.000)	(0.001)	(0.002)	(0.001)	(0.001)	(0.002)	(0.001)
OSS Announcement Ratio		-2.721	-4.219	-4.339	-6.613	-4.089	-3.829	-5.887
		(5.253)	(4.060)	(5.672)	(4.874)	(4.037)	(5.962)	(4.542)
Pre-announcement		-1.253*	-0.501	0.439	-0.919	-0.475	0.394	-0.630
		(0.676)	(0.720)	(1.527)	(1.189)	(0.733)	(1.584)	(1.246)
Pre-release		-0.251	0.576	0.858	0.324	0.556	0.613	0.398
		(1.515)	(1.931)	(3.710)	(1.025)	(1.925)	(3.650)	(1.094)
Sector Dummies			Yes	Yes	Yes	Yes	Yes	Yes
Year Dummies			Yes	Yes	Yes	Yes	Yes	Yes
Constant	0.696*	1.037**	2.495	2.861	2.091	2.523	3.089	1.992
	(0.383)	(0.506)	(2.040)	(2.685)	(2.641)	(2.024)	(2.606)	(2.620)
Observations	149	149	149	72	80	149	72	80
R-squared	0.000	0.020	0.132	0.234	0.284	0.130	0.227	0.270

Robust standard errors in parentheses are clustered by firms
*** p<0.01, ** p<0.05, * p<0.1

3.4.1 Robustness Checks

Although we account for the informational heterogeneity of announcements by controlling for pre-announcements in our main regression analysis, we re-run the analysis with a reduced sample in which we exclude pre-announcements. By doing so, we eliminate the implicit assumption of homogeneous information (Koku et al., 1997) and assess the rewards to the announcements of official code contributions. In such cases, open source code contributions are positively rewarded by the market with a 0.97% increase in valuation at a significance level of 0.05. Recalling that the CARs to the code contributions were about 0.66% for the full sample, it is possible to infer that investors react more favorably to actual source code contributions than to pre-announcements of source code contributions that will be made in the future. The result of a cross sectional regression analysis based on the reduced sample suggests that the findings remain unchanged (Appendix A).

We also conduct a robustness check for *Hypothesis 3* by introducing an interaction term between the categories that define newness of the code and a binary variable that identifies the intended audience of the code. The coefficient on the interaction terms between *New Code* and *System or Application Software* variables is positive and significant suggesting that benefits are larger for firms that contribute new code intended for end-users. These results are present in Appendix B.

3.5 Discussion

The core findings of this study are threefold. By participating in innovation communities, firms can improve their innovation outcomes and thereby generate value. Soliciting feedback from communities pertaining to novel products or innovations will generate greater benefits. And, finally, if these novel products or innovations are targeted towards end-users rather than developers, the benefits will be greater. These findings support the theory that users generate distinctive knowledge and that this knowledge is a valuable input into firms' product development efforts.

This study contributes to the user and community-based innovation literature by joining a small number of papers in documenting the value of user innovation to established firms. Product development is a herculean challenge (Cooper and Kleinschmidt, 1995; Prandelli, Sawhney and Verona, 2008). Although it has been suggested that engaging users is likely to provide firms with an improved way of managing this challenge (Urban and von Hippel, 1988; Lilien, 2002; von Hippel and Katz, 2002; Jeppesen and Molin, 2003; Jeppesen and Frederiksen, 2006; Shah, 2006), few studies have been able to provide data to support this presumption. In the medical device industry, firms draw on the innovative insights of users when developing their own innovations (Winston Smith and Shah, 2013; Chatterji and Fabrizio, 2012), however whether or not these innovations generate value for the firm has not been examined. Ours is the first study to provide systematic empirically documentation showing that engaging user communities leads to increased tangible, financial value for the firm.

This study makes a further contribution to this literature by beginning to identify *how* user communities might contribute most to firm's innovation efforts.[25] We examine just two aspects here—novelty and audience—future research might examine other characteristics of innovations (e.g. incremental vs. radical innovation) or the characteristics of the marketplace for the innovation (e.g. high-tech). A key goal for future work should be to continue to investigate "what activities is the firm best at and what activities are the communities best at?" Not all consumers help build a product, in fact, it appears that most do not. Both community and firm can work together to create better products for the mass market.

The findings add to our understanding on learning; where communities can provide valuable knowledge to firms. Firms become actively involved in the community-based innovation model at a gradual level through various means (Dahlander and Wallin, 2006). Active participation to communities, according to the private-collective model of innovation, will generate private benefits that will not be available to free-riders (von Hippel and von Krogh, 2003). However, this assumption lacks empirical evidence within the context of community participation (Lerner and Tirole, 2002). Learning from innovative community is an important way of gaining these private benefits, which will exclusively accrue to active contributors (Alexy and Reitzig, 2013). We look at a particular means of active firm participation in communities, code contribution, with intentions to foster co-creation of software. Although our study does not constitute a direct test for the private-collective innovation

[25] It is important to note that we look at the value generated by a particular set of interactions between firms and communities: the value triggered and developed when a firm contributes code to a community. Firms might also access user insights through other mechanisms. Whether or not the mechanism by which insights are accessed influences the type (or quality) of insight is an empirical question, however the consistency of the current empirical literature on user innovation suggests that users possess unique insights that are difficult for firms to independently generate.

model, we add to the literature by showing empirically that for-profit firms are to benefit from contributing to the commons. Our findings suggest that the contributions that generate the largest benefits are those that foster learning from communities by nurturing knowledge flows. Investors reward those contributions that are novel and thus, are more likely to create a long-term knowledge flow between the community and the firm. Firms, in order to sustain their ability to introduce new products to the market successfully, can and should search for ideas from their external environments (Chesbrough, 2003). And, communities represent a very robust alternative to firm-based knowledge creation due to the virtue of collective creativity (Lee and Cole, 2003).

Literature on open innovation encapsulates community-based innovation, as well as other mechanisms by which the firm searches its external environment for knowledge. Other mechanisms—alliances, joint ventures, university collaborations—have been the object of much empirical and theoretical attention in the strategy and technology management literatures. In contrast, community-based innovation has been the object of less scholarly work, yet recent empirical work suggests that community-based innovation may be a critical input to established firms' innovation processes and a source of startups (Shah and Tripsas, 2007; Shah, Winston Smith and Reedy, 2012). Firm engagement with communities appears to be on the rise within the context of OSS movement (Bonaccorsi et al., 2006). Gassmann, Enkel and Chesbrough (2009) suggest that open innovation approach within a community context seem to be beneficial for companies as well as users both practically and theoretically.

We contribute to the literature on community-based innovation by enhancing our understanding of the process by which firms can work within communities. We provide with empirical evidence on the benefits of co-creation of software in open source platforms for a firm with an established measure of perceived value of firms' innovative activity by investors. However, an appropriate metric system for innovation is needed to compare and contrast the impact of open versus closed innovation approaches for assessing the right balance in a firm's innovative agenda (Enkel and Lenz, 2009).

Our findings illustrate that firms can participate in community-based innovation and obtain private benefits from doing so. By engaging innovation communities, firms can nurture the co-creation of knowledge that can be fed into the development and design of commercial products. We provide firms with both a rationale for why community participation will benefit their innovation processes and guidance pertaining to how to derive particularly beneficial outcomes when sharing their knowledge with others in order to receive feedback.

As with most empirical research endeavors, there are several limitations to be considered. The general limitations that apply to all event studies are relevant to our study as well: the firms in our sample are publicly traded and therefore relatively large. As a result, the possibility of confounding events is higher. In order to prevent biased results, we carefully identified and eliminated those observations that coincided with a confounding event and limited the event window to a maximum of two days.

We examined a particular way in which firms can access ideas and feedback from innovation communities: code contributions. There are other means by which firms might actively engage with communities and/or community members, such as conference or meeting participation, observation, consulting, hiring, or buying or investing in startups that emerge from the community. Future research might investigate the viability of these mechanisms. Our key contribution is in showing that community participation benefits firms.

Finally, ours is a single industry study. Future research might seek to replicate these findings in other industrial contexts. Doing so will require adapting the methods and approaches used, as one reason that open source software provides a viable context for studying firm involvement in communities is the transparency of the setting.

3.6 Conclusion

Tapping into users' insights generate valuable innovative outcomes for firms. Benefits of engaging in user communities for established firms have long been discussed in management literature. Few, if any, studies have empirically shown that those benefits are tangible as reflected in stock market prices and that they may vary depending on the characteristics of the innovation that is going to be co-created within the community. We provide useful insights for firms interested in improving their product development processes. Communities of users may provide firms with valuable knowledge and extend their understanding on product development on many fronts. Soliciting feedback for creating a novel innovation as opposed to soliciting feedback for incremental developments generate greater benefits. End-users as opposed to corporate users will provide with more insightful ideas for a novel product development.

3.7 References

Alexy, O., George, G. 2013. Category Divergence, Straddling, and Currency: Open Innovation and the Legitimation of Illegitimate Categories, *Journal of Management Studies* 50(2), 173-203.

Alexy, O., Reitzig, M. 2013. Private–collective innovation, competition, and firms' counterintuitive appropriation strategies. *Research Policy* 42(4): 895-913.

Aggarwal, N., Dai, Q., Walden, E. A. 2006. Do Markets Prefer Open or Proprietary Standards for XML Standardization? An Event Study. *International Journal of Electronic Commerce* 11(1), 117-136.

Allen, R. C. 1983. Collective Invention. *Journal of Economic Behavior and Organization* 4, 1-24.

Austin, D. H. 1993. An event study approach to measuring innovative output: The case of biotechnology. *American Economic Review* 83(2), 253-258.

Baldwin, C., Hienerth, C., von Hippel, E. 2006. How user innovations become commercial products: a theoretical investigation and case study. *Research Policy* 35 (9): 1291-1313.

Benkler, Y. 2002. Coase's Penguin, or, Linux and the Nature of the Firm, *The Yale Law Journal* 112 (3), 369-446.

Bonaccorsi, A., Giannangeli, S., Rossi, C. 2006. Entry strategies under competing standards: Hybrid business models in the Open Source software industry. *Management Science* 52(7), 1085-1098.

Campbell, J.Y., Lo, A. W., MacKinlay, A. C. 1997. The Econometrics of Financial Markets. Princeton, NJ: Princeton University Press

Chaney, P. K., Devinney, T. M., Winer, R. S. 1991. The impact of new product introductions on the market value of firms. *Journal of Business* 64(4), 573-610.

Chatterji, A. and Fabrizio, K., 2012. How Do Product Users Influence Corporate Invention", *Organization Science* 23, 951-970.

Chesbrough, Henry W. 2003. Open Innovation: The New Imperative for Creating and Profiting from Technology. *Harvard Business School Press*, Boston, MA.

Cooper, R.G., Kleinschmidt, E.J. 1995. Benchmarking the Firm's Critical Success Factors in New Product Development. *Journal of Product Innovation Management* 12(5), 374-391.

Dahlander, L., Magnusson, M. G. 2005. Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34(4), 481-493.

Dahlander, L., Wallin, M. W. 2006. A man on the inside: unlocking communities as complementary assets. *Research Policy* 35(8), 1243-1259.

Enkel, E., Lenz, A. 2009. Open innovation metrics system. Proceedings of the R&D Management Conference, Vienna, Austria, June 21–24.

Fama, E. F., French, K. R. 1992. The cross-section of expected stock returns. *The Journal of Finance* 47(2), 427-465.

Ferris, T. 2002. Seeing in the Dark: How backyard stargazers are probing deep space and guarding earth from interplanetary peril. Simon and Schuster, New York.

Fosfuri, A., Giarratana, M. 2009. Masters of War: Rival's Product Innovation and New Advertising in Mature Product Markets. *Management Science* 55(2), 181-191.

Fosfuri, A., Giarratana, M., Luzzi, A. 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science* 19, 292-305.

Franke, N., Shah, S. 2003. How communities support innovative activities: An exploration of assistance and sharing among end-users. *Research Policy* 32(1), 157-178.

Franz, K. 2005. Tinkering*:* Consumers Reinvent the Early Automobile. University of Pennsylvania Press, Phildelphia, PA.

Freiberger, P., Swaine*,* M. 2000. *Fire in the Valley.* McGraw-Hill, New York.

Gassmann, O., Enkel, E., Chesbrough, H. 2009. Open R&D and open innovation: exploring the phenomenon, *R&D Management* 39(4), 311-316.

Goldman, R., Gabriel, R.P. 2005. *Open Source as Business Strategy: Innovation Happens Elsewhere*. Morgan Kaufmann, San Francisco, CA.

Griffin, A., Hauser, J.R., 1993. The Voice of the Customer. *Marketing Science* 12(1), 1-27.

Hannan, M. T., Pólos, L., Carroll, G. R. 2007. *Logics of organization theory: Audiences, codes, and ecologies*. Princeton, NJ: Princeton University Press.

Hargrave, T.J., Van de Ven, A. 2006. A collective action model of institutional innovation. *Academy of Management Review* 31(4), 864-888.

Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35(7), 953-969.

IBM, Corp., 2004, August 3. IBM Contributes Database Software to the Apache Software Foundation; In-creases Efforts to Drive Innovation Around Java Application Development. *Business Wire*. Retrieved from Lexis-Nexis database.

Jeppesen, L.B., Frederiksen, L. 2006. Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer Controlled Music Instruments. *Organization Science* 17, 22-44.

Jeppesen, L. B., Molin, M. J. *2003. Consumers as Co-developers:* Learning and Innovation outside the Firm. *Technology Analysis and Strategic Management* 15(3), 363-384.

Kline, R., *Pinch,* T. *1996*. *Users as* agents of technological change: the social construction of the automobile in the rural United Sates, *Technology and Culture* 37, 763-795.

Koku, P. S. 2009. The information content and the stock market's reaction to new product announcements in the USA computer industry. *Journal of Applied Management Accounting Research* (Winter), 21-36.

Koku, P. S., Jagbal, H. S., Viswanath, P. V. 1997. The effect of new product announcements and preannouncements on stockprice. *Journal of Market Focused Management* 2, 183–199.

Kothari, S. P., Warner, J. B. 2004. The Econometrics of Event Studies, Available at SSRN: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=608601

Lee, G. K., Cole, R. E. 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development, *Organization Science* 14(6), 633-649.

Lerner, J., Tirole J. 2002. Some simple economics of open source. *Journal of Industrial Economics* 50(2), 197-234.

Lilien, G.L., Pamela, D. Morrison, Kathleen Searls, Mary Sonnack, Eric von Hippel. 2002. Performance Assessment of the Lead User Idea Generation Process. *Management Science* 46 (12), 1513-1527.

MacKinlay, A. C. 1997. Event studies in economics and finance. *Journal of Economic Literature* 35, 13–39.

McWilliams, A., Siegel, D. 1997. Event studies in management research: Theoretical and empirical issues. *Academy of Management Journal* 40 (3), 626-657.

Novell, Inc., 2005, February 15. Novell Launches Open Source Collaboration Server Initiative; Novell Donates 200,000+ Lines of Source Code and Invites Community Development of the Next Generation of Innovative Collaboration Capabilities. *PR Newswire*. Retrieved from Lexis-Nexis database.

Oh, W., Gallivan, M. J., Kim, J. W. 2006. The market's perception of the transactional risks of information technology outsourcing announcements. *Journal of Management Information Systems* 22, 271-303.

Prandelli, E., Sawhney, M. S., Verona, G. 2008. *Collaborating with Customers to Innovate:Conceiving and Marketing Products in the Networking Age*. Edward Elgar Publishing: Northhampton, MA.

Raymond, E. 2001. The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly & Associates, Sebastopol, CA.

Riggs, W., von Hippel, E. 1994. Incentives to Innovate and the Sources of Innovation: The Case of Scientific Instruments. *Research Policy* 23, 459-469.

Rumelt, R. P. 1984. Towards a Strategic Theory of the Firm. In Competitive Strategic Management, Robert Boyden Lamb (Ed.), Englewood Cliffs, NJ: Prentice-Hall, Inc.

Shah, S. K. 2005. Open Beyond Software. *Open Sources 2*. Editors: Chris Dibona, Danese Cooper and Mark Stone. O'Reilly Media, Sebastopol, CA. 339-360.

Shah, S. K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science* 52(7), 1000-1014.

Shah, S. K., Mody, C.M. Creating a Context for Entrepreneurship: Examining How Users' Technological & Organizational Innovations Set the Stage for Entrepreneurial Activity. Cultural Commons. Editors: Brett Frischmann, Michael Madison and Katherine Strandburg. Oxford University Press. 2013

Shah, S. K., Tripsas, M. 2007. The Accidental Entrepreneur: The Emergent & Collective Process of User Entrepreneurship *Strategic Entrepreneurship Journal* 1(1): 123-140

Shah, S. K., Winston Smith, S., Reedy, E.J. 2012. Who Are User Entrepreneurs? Findings on Innovation, Founder Characteristics & Firm Characteristics. In *Kauffman Foundation Report.* Kauffman Foundation: Kansas City, MO.

Shiller, R. J. 2005. *Irrational Exuberance*, 2nd edition. Princeton, NJ: Princeton University Press.

Solomon, D. 2012. Selective Publicity and Stock Prices. *The Journal of Finance* 67(2), 599-637.

Solomon, D., Soltes, E. 2013. What are we meeting for? The consequences of private meetings with investors. Working Paper, USC and Harvard Business School.

Sood, A., Tellis, G. J. 2009. Do innovations really pay off? Total stock market returns to innovation. *Marketing Science* 28(3), 442-456.

Srinivasan, S., K. Pauwels, J. Silva-Risso, D. M. Hanssens. 2009. Product innovations, advertising, and stock returns. *Journal of Marketing* 73(1), 24-43.

Sun Microsystems, Inc., 2000, October 16. Sun Microsystems Announces Availability of StarOffice (TM) Source Code On OpenOffice.Org; Sun Works With CollabNet to

Deliver Largest Donation of Source Code to Open Source Community. *PR Newswire*. Retrieved from Lexis-Nexis database.

Taylor, A. 2010. The Next Generation: Technology Adoption and Integration through Internal Competition in New Product Development *Organization Science* 21(1), 3-41.

Teece, D. J. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15, 285-305.

Urban, G.L., von Hippel, E. 1988. Lead User Analyses for the Development of New Industrial Products. *Management Science* 34(5), 569-582.

von Hippel, E. 1988. *The Sources of Innovation*. Oxford University Press: New York.

von Hippel, E., Katz, R. 2002. Shifting Innovation to Users Via Toolkits. *Management Science* 48(7), 821-833.

von Hippel, E., von Krogh, G. 2003. Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science. *Organization Science* 14(2), 209-233.

Winston Smith, S., Shah, S. K. 2013. Do Innovative Users Generate More Useful Insights? An Analysis of CVC Investment in the Medical Device Industry. *Strategic Entrepreneurship Journal*.

3.8 Appendices

Appendix A: Robustness Check - CARs for Announcements Pertaining to Available Code

	Baseline Model		Main Results					
			Full sample	Sys/App SW	Dev. Tools	Full sample	Sys/App SW	Dev. Tools
VARIABLES	CAR	CAR	CAR	CAR	CAR	CAR	CAR	CAR
New Code			1.595*	3.798***	-0.244			
			(0.921)	(1.196)	(1.870)			
Updated Code			1.056	3.420*	-1.672			
			(0.841)	(1.976)	(1.191)			
Novelty						0.817*	1.889***	-0.280
						(0.431)	(0.580)	(0.874)
Software Patents		0.000	0.000	0.000	0.001	0.000	0.000	0.001
		(0.001)	(0.001)	(0.003)	(0.001)	(0.001)	(0.003)	(0.001)
OSS *Announcement Ratio*		-1.589	-1.271	-1.485	3.956	-1.503	-3.612	4.886
		(5.287)	(5.165)	(8.702)	(11.492)	(5.112)	(7.982)	(11.666)
Sector Dummies		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Year Dummies		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Constant	0.974**	2.508	2.027	4.045	1.430	2.028	4.096	1.361
	(0.452)	(2.361)	(2.495)	(3.136)	(2.729)	(2.476)	(2.983)	(2.783)
Observations	107	107	107	49	60	107	49	60
R-squared	0.000	0.174	0.204	0.474	0.365	0.204	0.461	0.350

Robust standard errors in parentheses are clustered by firms

*** p<0.01, ** p<0.05, * p<0.1

Appendix B: OLS Regression Results with Interaction Terms

VARIABLES	(1) CAR	(2) CAR	(1) CAR	(2) CAR
New Code	1.273*	-0.770		
	(0.760)	(1.123)		
*New Code * System or Application SW*		4.062**		
		(1.628)		
Updated Code	0.266	-0.347		
	(0.789)	(1.151)		
*Updated Code * System or Application SW*		1.149		
		(2.027)		
Novelty			0.610*	-0.374
			(0.366)	(0.571)
*Novelty * System or Application SW*				1.964**
				(0.803)
System or Application SW	0.266	-0.975	0.279	-1.076
	(0.603)	(0.941)	(0.596)	(0.857)
Software Patents	0.001	0.001	0.001	0.001
	(0.001)	(0.001)	(0.001)	(0.001)
OSS Announcement Ratio	-4.362	-5.435	-4.245	-5.224
	(4.121)	(3.639)	(4.093)	(3.671)
Pre-announcement	-0.540	-0.124	-0.517	-0.161
	(0.724)	(0.806)	(0.734)	(0.812)
Pre-release	0.540	0.603	0.520	0.497
	(1.906)	(1.872)	(1.900)	(1.849)
Sector Dummies	Yes	Yes	Yes	Yes
Year Dummies	Yes	Yes	Yes	Yes
Constant	2.397	2.655	2.419	2.772
	(2.072)	(1.928)	(2.056)	(1.855)
Observations	149	149	149	149
R-squared	0.133	0.174	0.131	0.171

Robust standard errors in parentheses are clustered by firms

*** p<0.01, ** p<0.05, * p<0.1

Chapter 4 – Firm-Community Collaborations: Policy and Formality Effects on Performance

4.1 Introduction

Recently, innovation projects have "opened-up". External linkages are now increasingly used by firms to tap on additional resources to develop their innovation and knowledge creation processes. Research has shown that the typology of external sources a firm can benefit from may be very heterogeneous (Laursen and Salter, 2006). Universities (Laursen and Salter, 2004) users (von Hippel, 1988) and suppliers (Jayarama, 2007) are only part of the possible subjects involved by firms in their innovation projects.

A lively stream of research on this theme has recently shown that firms' collaborate also with communities of users and developers (Jeppesen and Frederiksen, 2006; Dahlander Frederiksen and Rullani, 2007; Sproull, Dutton and Kiesler, 2008), i.e. firms could innovate by engaging in community-based innovation projects (CIP). Collaborations with communities engender substantial benefits to firms by augmenting and complementing the scientific, artistic, or technological knowledge that they produce internally (Hargrave and Van de Ven, 2006).

The literature on new product development (Trott, 1998) has shown that specific features of the projects and of the innovation to be produced (Brun, Saetre and Gjelsvik, 2009), as well as the environmental factors (Akgün, Byrne, Lynn and Keskin, 2007), and the firm specific characteristics (Swink and Song, 2007) affect considerably the success of the project itself and its sustainability. Among these variables, a particularly relevant construct deals with the characteristics of the participants in the project. The relative heterogeneity (Giuri, Ploner, Rullani and Torrisi, 2010) or familiarity (Akgün, Byrne, Keskin and Lynn, 2005) of the involved actors, their communication channels and the way in which interaction is organized (Schulze and Hoegl, 2006) are qualities and processes that clearly determine –to various extent- the project's capability to survive over time and the probability of a positive innovation outcome.

At present, communities producing Open Source software (OSS) within OSS projects figure prominently among users and developers communities. The OSS movement has recently experienced a radical metamorphosis: it has acquired commercially viable forms and has rapidly increased its economic importance (Fitzgerald, 2006; Dahlander, 2007). Software firms are getting increasingly involved in the OSS movement. Many firms, have adopted OSS-based business models (Dahlander and Magnusson, 2005; Gruber and Henkel, 2006) and have pursued to do business out of the OSS code that is freely downloadable from the Internet. Likewise, software firms' participation in OSS projects[26] is currently a widespread phenomenon. Firms participate in the projects of collective software development created by OSS communities through varied participation strategies. Firms sponsor and even coordinate OSS projects (O'Mahony and West, 2008), pay their employees to develop and debug open code (Hars and Ou, 2002), reveal their own proprietary software code to OSS developers (Henkel, 2009), and so on.

There is important heterogeneity on how firms deal with OSS, since, when involved, some firms may have specific policies on OSS in order to establish the guidelines of collaboration. Projects, on the other hand, may vary largely by their governing models. While some are coordinated by a non-profit organization, some are coordinated by for-profit firms. In some cases there might be a formal leader in the project or the project as well may be coordinated informally. Several studies have explored the benefits from OSS projects' participation (e.g. Henkel, 2009) in order to disentangle which incentives drive for-profit firms to engage in this form of collective action (see Capra, Francalanci, Merlo and Rossi-Lamastra, 2011 for a review). Another research stream has focused on the design of firms' collaborations within OSS projects. In this respect, it has been noted that mutual adaptations are needed. Specifically, contrary to what one can expect for innovation projects originating and developed in a more "usual" firm-controlled environment, for fruitfully participating in OSS projects firms must properly design their business models (e.g. Bonaccorsi, Giannangeli and Rossi, 2006; Dahlander and Magnusson, 2005) and modify their internal organization (Colombo, Piva and Rossi-Lamastra, 2010; Dahlander and Wallin, 2006) to be able to accommodate the specific peculiarities of a subject (the community) that is placed out of their control. At the same time, firms need to support the capabilities of OSS project

[26] An OSS project is defined as "any group of people developing software and providing their results to the public under an Open Source license" (Evers, 2000). OSS projects are generally made available online in so called development platform, which are also repositories for the code itself. These are dedicated web sites that provide a suitable environment for software development and interaction between programmers. The largest and most well-known repository is SourceForge (http://sourceforge net). At the beginning of January 2010, more than 380,000 OSS projects and 2 million users were registered at SourceForge.

communities to mobilize voluntary contributors (O'Mahony and Bechky, 2008; O'Mahony and West, 2008). In this line, a more recent research avenue investigates the impact of OSS projects' participation on firms' performance (Stam et al., 2009).

However, the lively debate on the theme, research on firms' involvement in OSS projects still leaves room for further investigations. Specifically, it is undoubtedly reasonable to expect that firm's role in project's coordination structure have diverse effects on projects' performance. This paper aims to fill this literature gap by answering the following research questions: 1) How does direct firm involvement with a specific policy affect the performance of a community project? 2) How does coordination by firm affect the performance of a community project? 3) Does the involvement of admin in the firm have a moderating role on the relationship between coordination by firm and project's performance?

The issue is of undoubted relevance. A crucial prerequisite for firms' external knowledge sourcing activities from OSS communities is the existence of a long-lasting knowledge flow from these peculiar external sources. The tension between the members of the community and the professional firm in governing the project may lead to inefficient coordination that eventually may dry up this flow and even challenge the very survival of the OSS project. Being inspired by profit-seeking motives, firms' role as the sole authority in governing the project run the risk of fatally undermine the core of OSS communities' activity, which is often based on values at odds with the firms' profit-oriented goals (O'Mahony and Bechky, 2008).

The findings of this study are three fold. First, being directly involved in a community with a specific policy has a positive effect on project's performance. Second, coordination by profit-oriented firm has a negative effect on performance. Third, admin as an employee on main duty does not have a direct positive effect on performance. However, it positively moderates the aforementioned relationship. The study contributes to our knowledge on benefits of collaborations between firms and communities. Firm's role in the complex coordination mechanism of a community-based project and its effect on project's success is studied in an empirical setting. To our knowledge, there are no studies in the management literature that assesses the effect of firm policy, which describes how to work within the community, on the community-based project's performance.

In the next section, we present the theoretical background for our empirical analysis. Then, we describe the data and methodology. After presenting the results, we offer a brief discussion of our findings and conclude.

4.2 Theoretical Background and Hypotheses

Firms are making significant efforts to engage in collaborations with communities with expectations to receive benefits from these communities in return (von Hippel and von Krogh, 2003; Henkel, 2006). Without establishing a sound relationship that will lead to fruitful outcomes in a joint project, it may not be possible to reap the benefits that will accrue to a firm. For that end, it is important to explain the success/failure of these collaborative projects in order to give meaning to the benefits that they may provide to a firm. Although potential benefits of engaging in communities of practice for a firm has been identified at large (West, 2003; Dahlander and Wallin, 2006), few, if any, articles have sought to explain the sources of heterogeneities in projects' success that might have been originated from firm involvement in these projects. In specific, we aim to link firm's role in the complex coordination mechanism of a project with the success of the project. Key drivers of sustaining a fruitful collaborative project are aimed to be investigated in depth.

While firms' efforts in allocating resources to a community project may seem most likely to enhance the outcome of a project, it brings together many potential conflicts. A critical error that a firm may commit when engaging in an OSS project is its failure in not clearly defining the role of core developers, who are assigned to work on the project (Goldman and Gabriel, 2005). In most OSS projects the members decide on what modules they are going to work on. For small scale OSS project, self-assignment of tasks by community members may turn out to be efficient since each of these participants would effectively allocate their own resources because they know their skills and experience better than anyone else (Benkler, 2002). However, for a joint project, which involves a firm, documenting the guidelines on how to develop the software within a community platform is very important. If there is firm involvement in the project, it is likely that the firm will market the project to attract as many volunteer developers as possible and assign its own full time employees to work on the project as well. Thus, firm involvement will enable both paid employees of the firm and volunteer developers from diverse technical backgrounds to work on the same project simultaneously. The heterogeneity in members' work disciplines may cause a "chaotic" style of development that would result in uncoordinated changes that interfere with one another's written code (Mockus, Fielding and Herbsleb, 2000). In order to avoid such a problem in a joint project, which includes various groups of participants, the core developers must have a detailed knowledge on which modules and files they are going to work on and also on what other core developers are doing (Mockus, Fielding and Herbsleb, 2000). Assigning work activities among team members will increase a community-based project's performance.

Another common mistake of firms relates to the undefined metrics to be used in measuring the success of the OSS project (Goldman and Gabriel, 2005). Firms, which engage in community-based projects, most likely, also undertake some in-house software development projects at the same time. Failure in defining the metrics for the success can put a community-based project under strain and will give rise to neglecting community activities by the firm. Employees of a firm should be well-informed about the potential benefits the community-based project will bring about to their firm in order to understand the importance of the OSS project. Defining the metrics used to measure the success of the project will ensure employee dedication to the project by simply stating that the project is not less important than any other development project that is done in-house (Goldman and Gabriel, 2005, p.266).

The biggest mistake, however, are those actions of a firm that treat a community-based project as if it were an internal proprietary project owned by the firm (Goldman and Gabriel, 2005). Those actions include neglecting the community participation and trying to schedule the entire development phase at an early stage instead of waiting for the community to grow organically or sneaking code from the joint-project to the codebase of an in-house project for instance (Goldman and Gabriel, 2005). Such disrespectful activities that ignore the community will destroy the community's trust in the company and discourage participation.

We suggest that if a firm's involvement in the project is guided by a policy, which is formally defined and made visible to the community members, the firm will be able to avoid the aforementioned mistakes that are most likely to make during the development of a community-based project. Having a specific policy, which clearly defines the role of the paid developers, discusses how to handle the volunteer contributions, outlines the goals of the project, and clarifies how the project will contribute to firm's business goals, will improve the performance of the community-based project.

Hypothesis 1: Direct involvement of profit-oriented firms with a specific policy will have a positive effect on OSS project's success.

Second hypothesis relates to the coordination style of the project. "A *coordination style* depends on a certain way how developers work on the source code, and more important, how they interact with each other" (Spaeth, 2005). Collaborations can be difficult when the interests, goals, and practices of participants differ (O'Mahony and Bechky, 2008). Extracting financial benefits from jointly developed software contradicts the core values of OSS movement (O'Mahony, 2003). Firm involvement in the project as the principal authority for coordination may create a tension between the community and the firm, which may deter volunteer participation. Moreover, endowments of intellectual property rights of the participating firm in the project may also discourage participation of volunteer programmers (Shah, 2006).

Even if the interests align, it is hard to manage the boundaries of a collaboration that consists of a professional organization and participants of a social movement (Timmermans and Leiter, 2000). Firm's hierarchical work discipline may not fit the free-will of the community members. Such a conflict will decrease efficient collaborative work among parties. Rucht (2004) suggests that in firm-community projects, firms tend to maintain their autonomy and in order to do so, they avoid becoming a single entity. It is hard to strike a balance between trust and control in the presence of firm involvement within a community-based project and failing to do so may deter participation from volunteers. Thus, we suggest that coordination by profit-oriented firm will have a negative effect on project's performance.

Hypothesis 2: Coordination by profit-oriented firms will have a negative effect on projects' performance.

Drawing on social network theory it is argued that there are favorable network positions that enable firms to manage key resources without ownership in the legal sense (Burt, 1992; Wasserman and Faust, 1994). Dahlander and Wallin (2006) suggest that firms act strategically by claiming these favorable network positions in an OSS community to be able to gain access to privileged information. Key person in a professional network may act as technological broker by combining the best of both worlds (Hargadon and Sutton, 1997) and build legitimacy, which can be achieved through proof of skillfulness and by providing help to other individuals in the community. Having a full time employee, whose main duty is to work on the project together with other participants, will help the community develop more organically instead of pushing a strong corporate agenda that will seem suspicious to the community.

We suggest that by assigning the administrator position to an employee as his/her main duty, a firm will have favorable position within the project. And, thus, negative effect of coordination by firm on performance will decrease. The admin of the project, who at the same time is an employee of the firm, will have advantage in terms of financial resources and will have access to the results and developments from in-house investments. A firm employee fully dedicated to work on the project as the admin may offset the tension between the community and the firm by building legitimacy and, thus, will positively moderate the relationship between coordination by firm and project's performance.

Hypothesis 3: If the administrator of the project is a full-time employee of the firm, whose main duty is to work on the project, the negative effect of coordination by profit-oriented firm will decrease.

4.3 Research Method

4.3.1 Sample and Data Collection

Our empirical analysis is undertaken on a sample of Java-based[27] OSS projects hosted on the platform SourceForge.net[28], which is the largest and most popular repository of community based OSS developments. Multiple sources of data have been used to gather information that enables us for grounded theory development and validation of theoretical constructs.

An online survey was run in late November 2007 to collect data on projects' coordination styles and firms' various involvement modes within these OSS projects (Capra, Francalanci, Merlo and Rossi-Lamastra, 2011). Specifically, the online survey was addressed to all the administrators of the 8,308 OSS projects written in Java and

[27] Java programming language is one of the most popular programming languages.
[28] cf. http://www.sourceforge.net

hosted on SourceForge.net as of November 2007 (see Capra et al., 2011 for further details). The survey includes questions that aim at identifying project characteristics, which are not possible to retrieve from the internet and also questions regarding the leading company, when there is firm involvement in the project. The questions related to firm involvement were more specifically about the role of the firm (e.g. sponsor, founder, directly involved with or without specific policy, simply acknowledges participation of its employees), the relation of the administrator of the project with the firm, and the management method of the project. We also gathered secondary data on each firm identified in this way using Internet searches and calling up directly the firms when needed.

The questionnaire was hosted on SurveyMonkey.com, a website specialized in online surveys. Each project's administrator was contacted via e-mail by sending him/her the URL of the questionnaire and a reference institutional home-page. In case of ambiguous or incoherent answers, she or he was re-contacted via e-mail and interviewed to clarify the goals of the questionnaire and obtain correct data. These interviews were also used to check whether the project information declared on SourceForge.net was correct. Before submitting the questionnaire to the selected administrators, a pilot test on a random sample of 195 administrators, stratified by project size, was run. This pilot test was carried out with several OSS developers and project administrators, who extensively discussed a preliminary version of the questionnaire. The final version accounts for the feedback and comments by the participants of this pilot stage. The survey was conducted starting from November, 27th 2007. After the first mailing, two follow-ups were performed, starting from December, 5th 2007 and December, 18th 2007, respectively. The answers obtained were relative to 1408 OSS projects.

We have then matched these data to those retrieved directly from the platform and collected by Notre Dame University (see Madey, 2009). Initially, we dropped 97 projects that were initiated after the first questionnaire in December 2006. Then, we dropped 212 observations, which either lack the necessary information related to the firm (e.g. size, age, and sector) or have missing items from the questionnaire[29]. The

[29] In order to alleviate the concerns that the results might have been affected from a non-response bias, we run Wilcoxon Mann-Whitney test to compare our variables of interests across sub-samples (sub-sample of observations before and after dropping the observations with missing items). Mean values of both of the dependent variables, *File_Size_t_3* and *File_Number_t_3*, seems statistically significantly different from

combined dataset includes information on 1099 OSS projects out of which 267 of them are being developed in collaboration with a firm. This final sample allows us to analyze the direct effects of firm involvement with specific policy and the coordination by firm as well as the indirect effect of admin's role in the firm as the employee on main duty through interaction with coordination by firm.

We have both projects with and without firms, and this allows us to isolate the effect of the firm presence, provided that we can control for the projects' characteristics both as such, and as a source of firms' selection of projects. To account for these problems and to decrease endogeneity, we thus decided to structure our equations as follows:

1) To assure exogeneity in the regression, we have created three time periods: t_1 spanning the months from August to December 2006, t_2 from August to December 2007, and t_3 from August to December 2008. Stock variables (such as the number of team members) are measured at the end of each period, while flow variables are measured over each whole period.

2) Our dependent variables are measured in period t_3, while our main variables of interest, i.e. policy, coordination styles, are relative to period t_2. We also included the main projects characteristics relative to period in t_2 in order to account for possible biases induced by the projects' intrinsic capabilities.

3) To rule out the selection bias described above we also included projects' characteristics in period t_1, so that we should obtain results net of the initial attractiveness projects had for firms and of the degree at which projects may have been open to firms' presence.

4.3.2 Measures and Validation

We aim to capture the performance of a project on the basis of its productivity. In specific, we use the sum of the total sizes of files (*File_Size_*t_3) and the number of files generated for each project (*File_Number_*t_3). These measures enable us to have an idea about the volume of the code developed in the project.

Three sets of items are used to proxy for our variables of interest in order to test the hypotheses put forward (see Appendix A). First set of items relate to firm's

---

each other for the two sub-samples. We also compare our main variables of interests across two sub-samples. The results remain unchanged meaning that our results are not caused due to a non-response bias.

direct/indirect involvement in the project and the presence of a policy on OSS. In order to test for *Hypothesis 1*, we create a binary variable, *Direct_Involvement_with_Policy_*t_2, which takes the value 1; if the firm is directly involved in the project with a specific policy and 0; otherwise. *Hypothesis 2* relates to the management model of the company and it postulates that coordination by profit-oriented firm will a have negative effect on project's performance. We create a binary variable, *Coordination_by_Firm_*t_2, which takes the value 1; if the firm is the main authority for projects' coordination activities and 0; otherwise. Finally, in order to test for *Hypothesis 3*, we create a binary variable, Admin/*Employee&Main_Duty_*t_2, which takes the value ;1 if the admin of the project is an employee of the firm, whose main duty is to work on the project and 0; otherwise.

We control for the remaining categories of the three items from which we created our variables of interest explained above. When testing *Hypothesis 1*, we control for the binary variables that represent the categories of direct involvement with no specific policy on the project (*Direct_Involvement_No_Policy_*t_2) and no direct involvement in the project (No*_Direct_Involvement_*t_2). The latter is the omitted category in the regression analysis. When testing *Hypothesis 2*, we control for the categories of coordination by formal leadership (*Coordination_Formal_Leadership_*t_2), coordination by key people through experience (*Coordination_Key_People_*t_2), informal coordination (*Coordination_Informal_*t_2), work alone on the project (*Coordination_Work_Alone_*t_2) and coordination by non-profit organization (*Coordination_by_Non_Profit_*t_2). We omit the last category in the regression. Finally, when testing *Hypothesis 3*, we control for the following categories that are derived from the third item; Admin is the owner of the firm (*Admin/Owner_*t_2), Admin is an employee, who has duties other than working on the project (*Admin/Employee&Not_Main_Duty_*t_2) and Admin is not an employee of the firm(*Admin/Not_Employee_*t_2). We designate the last category to be the baseline category in the regression.

We also introduce two sets of control variables, one of which controls for project characteristics and the other for firm specific characteristics in order to alleviate reverse causality and selection concerns[30]. Control variables related to projects'

[30] The control variables related to project characteristics that are used in the selection equation correspond to t_1 while in the outcome equation we use the same set of controls measured at t_2. There are only a few

characteristics that are relative to t_2 include status of the project as declared by the respondents (*Proj_Status_Active_*t_2, *Proj_Status_Respondent_Not_Active_*t_2, *Proj_Status_Not_Active_*t_2), the number of team members (*Team_Size_*t_2), the degree of the project in the network of projects on sourceforge.net (measured by the average number of other projects participated by the project's team members, *Proj_Degree_*t_2), the number of messages sent to the project's forums in the time period under analysis (*Forum_Msgs_*t_2), whether the projects has activated or not advanced tools for code production (i.e. Concurrent Versions System, *Coding_Tool_*t_2) or for communication (mailing lists and forums, *Communication_Tool_*t_2). Many more dummy controls at project level have been identified that correspond to the development status of the project (*Dev_Status_Mature/Stable_*t_2), the function of the software (*Topic_Software_Development_*t_2), the language spoken by the team members (*Language_English_*t_2), operating system the software was written for (*Operating_System_Linux_*t_2, *Operating_System_Group_Independent_*t_2)[31], programming language the software was written in (*Programming_Language_C_C++_Sharp_*t_2), the intended audience of the project (*Audience_Developers_*t_2) and the user interface type (*Environment_Api_*t_2). In addition, we introduced controls in the selection equation that were only relative to t_1 such as; whether or not the project has declared any trove categories[32] (*No_Trove_*t_1), project's registration time on Source Forge (*Registered_Time_*t_1), performance of the project in the same period by using the number of files created in t_1 (*File_Number_*t_1) and total sizes of the files created in t_1 (*File_Size_*t_1). By including these controls, we aim to avoid a potential selection bias that might originate from firms' decision to participate in a project or projects' willingness to receive firms' participation. Firm specific controls, which are relative to t_2, include size, age, sector and region in which the firm is headquartered. Table 1 provides descriptive statistics. Tetrachoric correlations of our main variables of interest are presented in Table 2. There seems to be no problem regarding multicollinearity among these variables. Table 3a and Table 3b present the pairwise correlations among all the variables.

---

exceptions that the variable was present for one of the two periods and, thus, was used in the corresponding equation (in selection equation if t_1 and in outcome equation if t_2). For instance, project's status variables correspond to t_2 and are not included in the selection equation. *No_Trove_*t_2 was dropped due to collinearity. Thus, we only included *No_Trove_*t_1 *in the selection equation.*

[31] OS Independent (written in an interpreted language).

[32] SourceForge net uses *Trove* system in order to classify projects. The categories include topic, license, intended audience, programming language, intended audience, development status etc. http://sourceforge.net/apps/trac/sourceforge/wiki/Software%20Map%20and%20Trove

Table 1: Descriptive Statistics

Variables	*No. Obs*	*Mean*	*Std. Dev.*	*Min*	*Max*
1 File_Size_t_3	267	5.208167	8.259461	0	23.25506
2 File_Number_t_3	267	.623222	1.134998	0	4.795791
3 Direct_Involvement_with_Policy_t_2	267	.3857678	.4876903	0	1
4 Direct_Involvement_No_Policy_t_2	267	.340824	.4748764	0	1
5 Coordination_by_Firm_t_2	1099	.1182894	.3230975	0	1
6 Coordination_Formal_Leadership_t_2	1099	.0500455	.2181381	0	1
7 Coordination_Key_People_t_2	1099	.2183803	.413335	0	1
8 Coordination_Informal_t_2	1099	.2829845	.4506541	0	1
9 Coordination_Work_Alone_t_2	1099	.3066424	.4613095	0	1
10 Admin/Owner_t_2	1099	.0746133	.262886	0	1
11 Admin/Employee&Main_Duty_t_2	1099	.0409463	.1982561	0	1
12 Admin/Employee&Not_Main_Duty_t_2	1099	.0709736	.2568977	0	1
13 Coord(firm)*Admin/Emp&Main_t_2	1099	.0209281	.1432089	0	1
14 Firm_Size_t_2	267	3.651685	2.389361	1	8
15 Firm_Age_t_2	267	30.31835	60.11486	4	790
16 Proj_Status_Active_t_2	1099	.055505	.2290676	0	1
17 Proj_Status_Respondent_Not_Active_t_2	1099	.0636943	.2443187	0	1
18 Proj_Status_Not_Active_t_2	1099	.3539581	.4784141	0	1
19 #Messages_on_Forum_t_2	1099	17.96087	180.6526	0	4156
20 Project_Degree_t_2	1099	2.271274	1.660326	1	23.66
21 Dev_Status_Mature/Stable_t_2	1099	.5641492	.4960936	0	1
22 Reg_Time_Every_6_Months_t_2	1099	9.028207	3.817576	1	15
23 Team_Size_t_2	1099	6.007279	9.715814	1	144
24 Topic_Software_Development_t_2	1099	.4131028	.4926152	0	1
25 Prog_Language_C_C++_Sharp_t_2	1099	.2565969	.4369538	0	1
26 Audience_Developers_t_2	1099	.7042766	.456575	0	1
27 Language_English_t_2	1099	.7015469	.4577877	0	1
28 Operating_System_Indep. _t_2	1099	.6797088	.4668008	0	1
29 Operating_System_Linux_t_2	1099	.1856233	.3889794	0	1
30 Environment_Api_t_2	1099	.1656051	.371895	0	1
31 Communication_Tool_t_2	1099	.9399454	.2376961	0	1
32 Coding_Tool_t_2	1099	.7825296	.412713	0	1

Table 2: Correlation Matrix (Tetrachoric Correlations)

	1	2	3	4	5	6	7	8	9	10	11
1 Direct_Involvement_with_Policy_t_2	1.000										
2 Direct_Involvement_No_Policy_t_2	-0.926	1.000									
3 Coordination_by_Firm_t_2	0.351	-0.015	1.000								
4 Coordination_Formal_Leadership_t_2	0.169	-0.155	-0.172	1.000							
5 Coordintion_Key_People_t_2	0.014	-0.208	-0.292	0.246	1.000						
6 Coordination_Informal_t_2	-0.222	0.112	-0.383	-0.456	0.119	1.000					
7 Coordination_Work_Alone_t_2	-0.201	0.164	-0.203	0.047	-0.442	-0.479	1.000				
8 Admin/Owner_of_Firm_t_2	0.194	-0.062	0.144	0.188	-0.014	0.151	-0.079	1.000			
9 Admin/Employee&Main_Duty_t_2	0.032	-0.093	0.085	0.098	-0.008	-0.549	0.005	-0.511	1.000		
10 Admin/Employee&Not_Main_Duty_t_2	-0.058	0.067	-0.082	0.002	-0.056	0.096	0.236	-0.389	-0.514	1.000	
11 Coord(firm)*Admin/Emp&Main_t_2	0.087	-0.005	0.571	-0.442	-0.162	-0.409	-0.084	-0.408	0.735	-0.406	1.000

4.3.3 Methodology

Heckman's two-step procedure, which combines a first stage probit and a second stage OLS model, is used in order to avoid a potential sample selection bias

(Heckman, 1979). While the first step corresponds to the estimation of the probit equation by MLE for the sample selection and to the computation of the inverse Mills ratio (Heckman's lambda), the second step corresponds to the estimation of outcome equation on the selected sample by OLS including lambda as an additional regressor. The procedure is also known as 'Heckit model' and it gives consistent estimators.

In our model, the selection equation includes project characteristics at t_1 that may have a critical role in firms' decision to participate actively in the project.[33] In the outcome equation we aim to explore the effect of our main variables of interest, which are relative to t_2, on project's performance measured by number of files created and total sizes of the files created in t_3. We control for firm level characteristics as well as project level characteristics in both equations. All the control variables in the outcome equation are relative to t_2. However, we used the same control variables in the selection equation measured in t_1 to be consistent.

We use the licensing scheme of the software project as the exclusion restriction in the selection equation. Firms may choose the project that they are going to participate based on the licensing terms of the project. Firms, often, try to bundle the open source software with complementary proprietary software from which they make money. Licensing terms of the OSS project and its compatibility with the licensing terms of the complementary proprietary product is an important reason why a firm would or would not want to make investment in an OSS project. The licensing term, however, will not have an effect on the amount of code produced. Most people contribute to open source projects with features and capabilities because they want to use the resulting software themselves (Goldman and Gabriel, 2005). Hence, individual users' initial purpose of contribution is not related to their aims of redistributing the software. However, the main difference among various open source software licenses originates from the conditions of redistribution and not from the terms of use. The individual users, who are interested in using the software and not in redistributing it, will continue contributing to the project no matter what the licensing term under which the initial code has been released. Thus, the licensing term of the initial code release will not affect how much code generated with the participation of the community.

[33] The reduced sample, on which we work on in the selection equation, includes only those projects in which there is firm involvement.

Table 3a: Correlation Matrix (Pearson Correlations)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 File_Size_t_3	1.00															
2 File_Number_t_3	0.91	1.00														
3 Direct_Involvement_with_Policy_t_2	0.17	0.20	1.00													
4 Direct_Involvement_No_Policy_t_2	-0.14	-0.13	-0.57	1.00												
5 Coordination_by_Firm_t_2	0.01	0.04	0.25	-0.00	1.00											
6 Coordintion_Formal_Leadership_t_2	0.16	0.23	0.09	-0.07	-0.11	1.00										
7 Coordination_Key_People_t_2	0.04	0.02	0.00	-0.13	-0.25	0.11	1.00									
8 Coordination_Informal_t_2	-0.14	-0.15	-0.15	0.09	-0.33	-0.16	0.02	1.00								
9 Coordination_Work_Alone_t_2	-0.08	-0.10	-0.11	0.09	-0.19	-0.05	-0.21	-0.16	1.00							
10 Admin/Owner _t_2	0.03	0.02	0.17	-0.03	0.23	0.06	-0.01	0.01	-0.06	1.00						
11 Admin/Employee&Main_Duty_t_2	0.23	0.29	0.03	-0.05	0.04	0.04	-0.03	-0.20	-0.03	-0.30	1.00					
12 Admin/Employee&Not_Main_Duty _t_2	-0.13	-0.14	-0.02	0.06	0.03	-0.06	-0.04	-0.04	0.14	-0.43	-0.29	1.00				
13 Coord(byfirm)*Adm/Emp&Main_t_2	0.21	0.25	0.06	0.01	0.33	-0.11	-0.08	-0.14	-0.07	-0.20	0.68	-0.20	1.00			
14 Firm_Size_t_2	-0.09	-0.09	-0.14	-0.08	-0.21	-0.10	0.12	0.04	0.01	-0.49	0.16	0.13	0.08	1.00		
15 Firm_Age_t_2	-0.09	-0.08	-0.11	-0.00	-0.11	-0.02	-0.03	0.12	-0.01	-0.20	0.06	-0.01	0.02	0.43	1.00	
16 Proj_Status_Active_t_2	-0.12	-0.10	0.04	-0.04	0.05	-0.03	0.14	0.01	-0.03	0.04	0.04	-0.07	0.00	-0.05	-0.07	1.00
17 Proj_Status_ Respondent_Not_Active_t_2	-0.09	-0.08	0.03	-0.07	-0.04	-0.10	0.05	0.08	-0.05	-0.11	0.08	-0.04	0.02	0.12	-0.01	-0.10
18 Proj_Status_Not_Active_t_2	-0.25	-0.23	-0.01	0.03	-0.11	-0.13	-0.13	0.07	0.03	-0.09	-0.15	0.01	-0.10	0.07	0.12	-0.17
19 # Messages_on_Forum	0.25	0.39	0.11	-0.04	0.05	0.11	0.02	-0.05	-0.05	-0.03	0.10	0.02	0.17	-0.05	-0.03	-0.05
20 Project_Degree_t_2	-0.06	-0.05	0.07	-0.11	-0.06	-0.03	-0.01	0.10	0.14	-0.04	-0.15	0.08	-0.13	-0.07	-0.08	0.01
21 Dev_Status_Mature/Stable	0.16	0.13	0.18	-0.09	0.00	0.00	0.08	-0.06	0.01	0.08	0.12	-0.05	0.11	-0.02	-0.06	0.09
22 Reg_Time_Every_6_Months_t_2	-0.04	-0.06	-0.03	0.12	0.13	0.05	-0.17	0.10	-0.07	0.07	-0.15	0.04	0.02	-0.10	-0.02	-0.19
23 Team_Size_t_2	0.43	0.54	0.09	-0.11	-0.04	0.23	0.22	-0.10	-0.16	-0.12	0.29	-0.07	0.18	-0.01	-0.02	-0.03
24 Topic_Software_Development_t_2	-0.03	-0.03	-0.02	0.05	-0.04	-0.01	0.16	0.08	-0.04	0.04	-0.02	-0.00	0.10	0.01	-0.09	0.00
25 Prog_Language_C_C++_Sharp_t_2	-0.01	0.01	-0.02	-0.03	-0.06	0.02	0.04	-0.06	0.07	-0.01	0.05	-0.08	-0.08	0.05	-0.02	0.08
26 Audience_Developers_t_2	-0.04	-0.01	-0.04	0.03	0.04	-0.01	0.14	0.02	0.06	0.02	0.07	0.05	0.08	0.01	-0.10	-0.01
27 Language_English_t_2	0.06	0.10	0.11	-0.09	0.01	0.03	0.06	-0.09	-0.04	-0.05	0.11	-0.04	0.04	-0.04	-0.12	0.09
28 Operating_System_Indep._t_2	0.00	0.00	-0.03	-0.03	0.08	-0.04	0.01	-0.03	0.02	0.00	-0.05	0.08	0.02	-0.09	-0.02	0.10
29 Operating_System_Linux_t_2	-0.03	-0.05	0.03	0.03	0.01	0.11	-0.09	-0.16	0.07	0.01	-0.04	0.02	-0.08	0.03	0.02	-0.01
30 Environment_Api_t_2	0.08	0.05	0.07	-0.01	0.19	0.06	-0.02	-0.13	-0.11	0.15	0.03	-0.12	0.02	-0.16	-0.09	-0.03
31 Communication_Tool_t_2	0.02	0.04	-0.04	0.12	-0.08	0.10	-0.06	-0.03	0.06	0.04	-0.06	-0.01	-0.07	-0.05	-0.01	-0.23
32 Coding_Tool_t_2	-0.30	-0.27	-0.06	0.05	-0.04	-0.00	-0.08	-0.09	0.11	-0.04	-0.04	0.03	-0.12	0.08	0.10	-0.03

Table 3b: Correlation Matrix (Pearson Correlations)

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
17 Proj_Status_ Respondent_Not_Active_t_2	1.00															
18 Proj_Status_Not_Active_t_2	-0.12	1.00														
19 # Messages_on_Forum	-0.03	-0.06	1.00													
20 Project_Degree_t_2	0.16	-0.09	-0.05	1.00												
21 Dev_Status_Mature/Stable	0.01	-0.15	0.09	-0.03	1.00											
22 Reg_Time_Every_6_Months_t_2	0.03	-0.08	0.09	-0.12	-0.23	1.00										
23 Team_Size_t_2	-0.06	-0.12	0.46	-0.08	0.17	-0.23	1.00									
24 Topic_Software_Development_t_2	0.11	-0.05	-0.00	-0.09	-0.03	-0.08	-0.03	1.00								
25 Prog_Language_C_C++_Sharp_t_2	-0.00	0.04	-0.06	0.16	-0.04	-0.39	0.05	-0.04	1.00							
26 Audience_Developers_t_2	0.09	-0.10	0.03	0.09	0.10	-0.14	0.08	0.41	0.10	1.00						
27 Language_English_t_2	-0.01	0.08	0.08	0.05	0.19	-0.44	0.11	-0.02	0.20	0.03	1.00					
28 Operating_System_Indep._t_2	-0.07	-0.06	-0.04	0.00	0.06	-0.09	0.01	0.10	-0.14	0.06	0.17	1.00				
29 Operating_System_Linux_t_2	-0.02	0.08	-0.02	0.06	0.06	-0.20	-0.05	-0.04	0.29	-0.07	0.11	-0.25	1.00			
30 Environment_Api_t_2	0.03	-0.09	-0.04	-0.09	0.06	0.24	-0.01	-0.08	-0.10	-0.04	0.08	0.03	0.04	1.00		
31 Communication_Tool_t_2	-0.04	0.06	0.04	-0.04	-0.06	0.11	0.09	-0.04	0.05	0.04	-0.01	-0.08	0.03	0.06	1.00	
32 Coding_Tool_t_2	0.10	0.20	-0.12	0.07	-0.11	-0.17	-0.13	-0.09	0.17	0.01	0.07	0.00	0.15	-0.03	0.22	1.00

4.4 Findings

The results of the Heckit model are presented in Table 4. It is possible to observe the direct effects of our main variables of interest- direct involvement with specific policy and coordination by firm- in the first model. In the second model, we introduce the interaction term between employee assignment as administrator and coordination by firm. For each model, the results are presented under two columns one of which corresponds to the outcome equation and the other to the selection equation. We find that firm's direct involvement in the project with specific policy has a positive effect on project's performance as measured by number of files and sum of sizes of files generated during the project. On the other hand, the coefficient on the binary variable that represents coordination by firm is negative and significant when we use file size as dependent variable. These results are in support of *Hypothesis 1* and *Hypothesis 2*. We observe that the direct effect of the binary variable that identifies the administrator of project as employee of the firm, whose main duty is to work on the project, do not have direct positive effect on project's performance as measured by file size. However, we observe a significant positive effect of $Employee/Main\_Duty\_t_2$ variable on performance as measured by number of files generated.

When we introduce the interaction term between this variable and the coordination by firm variable in the second model, we observe a significant coefficient on the aforementioned interaction term. This result suggests that the negative effect of coordination by firm on project's performance is positively moderated if the firm assigns an employee, as the administrator of the project, to work full time on it. Results of the second model highlight the positive effect of presence of a policy to work on the OSS projects. The coefficient on direct involvement (with specific policy) variable maintains its positive and significant effect also after having introduced the interaction term. One may infer that even if the firm is directly involved in the project, without a specific policy on OSS, it will not contribute much to the performance of the project. Looking at different management methods of projects, it is possible to observe that informal coordination has a negative significant effect on project's performance.

Table 4: Results of the Heckman Model

Variables	Model I		Model II		Model I		Model II	
	(1) File_Size_t_3	(2) Selection	(1) File_Size_t_3	(2) Selection	(1) File_No_t_3	(2) Selection	(1) File_No_t_3	(2) Selection
Firm involvement in the project								
H1 *Direct_Involvement_with_Policy_t_2*	2.271*		2.246*		0.507***		0.507***	
	(1.168)		(1.176)		(0.143)		(0.149)	
Direct_Involvement_No_Policy_t_2	-0.064		-0.104		0.163		0.165	
	(1.180)		(1.181)		(0.129)		(0.130)	
Coordination style								
H2 By_Firm_t_2	-2.002*		-3.323**		-0.210		-0.371**	
	(1.128)		(1.338)		(0.141)		(0.155)	
Formal_Leadership_t_2	1.279		1.893		0.484**		0.603**	
	(1.704)		(1.766)		(0.229)		(0.258)	
Key_People_Through_Experience_t_2	-0.961		-1.257		-0.277**		-0.316***	
	(1.013)		(1.022)		(0.114)		(0.114)	
Informal_Coordination_t_2	-2.447*		-2.998**		-0.222		-0.291*	
	(1.372)		(1.418)		(0.154)		(0.157)	
Work_Alone_t_2	-1.383		-1.817		-0.156		-0.223	
	(1.303)		(1.287)		(0.154)		(0.155)	
Admin involvement in the firm								
Admin/Employee&Main_Duty_t_2	1.516		-1.352		0.388**		0.030	
	(1.436)		(1.723)		(0.176)		(0.210)	
Admin/Employee&Not_Main_Duty_t_2	-2.010*		-1.562		-0.140		-0.075	
	(1.166)		(1.174)		(0.130)		(0.147)	
Admin/Owner_t_2	-0.216		0.297		0.142		0.206	
	(1.273)		(1.260)		(0.155)		(0.159)	
Interaction term								
H3 *Admin/Employee&Main_Duty_t_2 * Coord_By_Firm_t_2*			5.677**				0.739**	
			(2.388)				(0.365)	
Controls for firm-level characteristics								
Firm_Size_t_2	-0.106		-0.110		0.002		0.001	
	(0.204)		(0.203)		(0.026)		(0.030)	
Firm_Age_t_2	-0.007		-0.006		-0.001		-0.001	
	(0.006)		(0.006)		(0.001)		(0.001)	
Sector_Dummies_t_2	Included		Included		Included		Included	
Region_Dummies_t_2	Included		Included		Included		Included	
Controls for project characteristics								
Proj_Status_Active _t_2	-3.812***		-3.651***		-0.345**		-0.329**	
	(1.057)		(1.015)		(0.139)		(0.141)	
Proj_Status_Respondent_Not_Active_t_2	-3.070**		-2.678*		-0.380*		-0.331	
	(1.536)		(1.564)		(0.209)		(0.236)	
Proj_Status_Not_Active_t_2	-3.885***		-4.041***		-0.326***		-0.334***	
	(0.971)		(0.979)		(0.121)		(0.121)	
Communication_Tool_(t_2/t_1)	3.080*	-0.225*	2.793	-0.231*	0.516**	-0.206	0.477**	-0.218*
	(1.711)	(0.135)	(1.742)	(0.135)	(0.220)	(0.126)	(0.223)	(0.128)
Coding_Tool_(t_2/t_1)	-3.064***	-0.284***	-2.970**	-0.284***	-0.174	-0.225**	-0.159	-0.222**
	(1.187)	(0.085)	(1.178)	(0.085)	(0.173)	(0.088)	(0.195)	(0.089)
Team_Size_(t_2/t_1)	0.140*	0.011***	0.138*	0.011***	0.039***	0.009***	0.037***	0.009***
	(0.076)	(0.002)	(0.074)	(0.002)	(0.010)	(0.003)	(0.010)	(0.002)
Dev_Status_Mature/Stable_(t_2/t_1)	-0.650	0.154**	-0.877	0.151**	-0.192	0.130*	-0.216*	0.124*
	(0.963)	(0.075)	(0.950)	(0.074)	(0.127)	(0.067)	(0.131)	(0.066)
Topic_Software_Development_(t_2/t_1)	-1.390	0.213***	-1.844*	0.212***	-0.205	0.218***	-0.261*	0.220***
	(1.028)	(0.080)	(1.058)	(0.080)	(0.141)	(0.079)	(0.144)	(0.079)

Language_English_(t_2/t_1)	-0.519	0.043	-0.450	0.043	0.075	0.012	0.069	(0.098)
	(1.269)	(0.082)	(1.248)	(0.082)	(0.170)	(0.087)	(0.168)	0.008
Operating_System_Linux_(t_2/t_1)	-0.620	0.045	-0.690	0.042	-0.206	-0.009	-0.223	-0.023
	(1.406)	(0.110)	(1.385)	(0.110)	(0.177)	(0.134)	(0.178)	(0.166)
Operating_System_Group_Indep. _(t_2/t_1)	-0.172	0.106	-0.261	0.108	-0.038	0.044	-0.045	0.046
	(1.146)	(0.085)	(1.127)	(0.085)	(0.164)	(0.085)	(0.162)	(0.085)
Programming_Language_C_C++_Sharp_(t_2/t_1)	-0.352	0.226**	-0.194	0.215**	-0.021	0.226**	0.010	0.217**
	(1.103)	(0.097)	(1.089)	(0.097)	(0.142)	(0.088)	(0.142)	(0.088)
Environment_Api_(t_2/t_1)	0.069	0.237**	0.406	0.230**	-0.124	0.261***	-0.066	0.247**
	(1.208)	(0.100)	(1.226)	(0.100)	(0.153)	(0.094)	(0.153)	(0.108)
Audience_Developers_(t_2/t_1)	-2.982**	0.233**	-2.803*	0.228**	-0.383*	0.206**	-0.371*	0.190*
	(1.491)	(0.096)	(1.467)	(0.096)	(0.202)	(0.093)	(0.222)	(0.098)
Registered_Time	-0.430**	0.044***	-0.480***	0.044***	-0.049**	0.036***	-0.058**	0.036***
	(0.176)	(0.012)	(0.176)	(0.012)	(0.025)	(0.011)	(0.027)	(0.011)
File_Size_t_1		0.026***		0.027***		0.274***		0.279***
		(0.006)		(0.005)		(0.037)		(0.040)
No_Trove_t_1		1.907***		1.920***		1.599***		1.560***
		(0.452)		(0.440)		(0.328)		(0.354)
Exclusion Restrictions								
License_Apache_t_1		-0.153		-0.131		-0.049		-0.032
		(0.172)		(0.169)		(0.149)		(0.153)
License_Artistic_t_1		-0.311		-0.327		-0.240		-0.247
		(0.314)		(0.312)		(0.306)		(0.295)
License_BSD_t_1		0.194*		0.194*		0.216**		0.209*
		(0.118)		(0.114)		(0.106)		(0.115)
License_GPL_t_1		-0.318***		-0.305***		-0.265**		-0.252**
		(0.103)		(0.103)		(0.103)		(0.120)
License_LGPL_t_1		0.084		0.105		0.132		0.164
		(0.099)		(0.098)		(0.090)		(0.107)
License_MIT_t_1		-0.066		-0.030		0.036		0.079
		(0.241)		(0.235)		(0.255)		(0.260)
License_Other_t_1		-0.008		0.002		0.018		0.035
		(0.101)		(0.100)		(0.087)		(0.086)
License_Prop_Other_t_1		0.348**		0.372**		0.328*		0.369*
		(0.154)		(0.156)		(0.187)		(0.200)
License_Pub_Domain_t_1		-0.025		-0.010		-0.133		-0.111
		(0.299)		(0.295)		(0.346)		(0.373)
Constant	23.053***	-1.306***	24.774***	-1.311***	2.313***	-1.271***	2.580***	-1.263***
	(4.301)	(0.260)	(4.395)	(0.261)	(0.760)	(0.237)	(0.981)	(0.239)
Observations	1,099	1,099	1,099	1,099	1,099	1,099	1,099	1,099

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

4.5 Discussion and Conclusion

The findings of this study are three fold. First, being directly involved in a project with a specific policy on OSS has a positive effect on project's performance. Second, coordination by firm has a negative effect on performance. Third, admin as an employee on main duty does not have a direct positive effect on performance. However, it positively moderates the aforementioned relationship.

These findings contribute to knowledge on benefits of co-creation of innovation between firms and communities of users and developers. The role of the firm in a community based project's complex coordination mechanism and its effect on the success of the project is studied within the context of OSS. To our knowledge, there are no studies in the management literature that assesses the effect of having a specific policy, which describes how to work within a community, on the community-based project's performance. We also focus on the role of the firm as the main authority in coordinating the community project and try to assess the effect of it in project's success. As has been stated earlier, there are several means by which projects might be coordinated, such as formal leadership, informal coordination, key people through experience or work alone. Future research might investigate potential effects of these different coordination mechanisms on the benefits garnered from a collaborative mode of innovation.

It would be also interesting to investigate how different participation strategies may affect the success of such collaborative innovation projects. Firms' primary activities on projects vary largely. Some firms participate by writing code, by testing the early versions of the software or by reporting and fixing bugs; some others give logistic and financial support, take part in planning and designing; many others help in diffusing the product by marketing it. How each of these activities contributes to a successful innovation process through which both the firm and community may benefit is a key question to be resolved by future studies.

Along this line, the paper can also directly inform managers on the strategies they should apply to assure long term sustainability of their external knowledge sourcing activities through communities. Managing the boundaries of collaborations is essential. Written rules and guidelines lead to fruitful joint development of software. In light of these findings, one may intuitively think that firm's involvement in a community project seems to have a positive effect on the success of the project in the presence of formality and written rules. An explanation of such en effect could be that communities benefit from participation of firms due to their managerial capabilities. However, these capabilities should not exert too much control over so that it does not deter participation by volunteer developers. A balance should be struck between the community of developers and the professional firm for sustaining a fruitful collaboration that would bring benefits for both parties. The participative mode of managerial attitude of the administrator of the project, who is an employee of the firm, might be helpful in achieving this balance. In this way, firm's organization and management may actually fully develop and increase projects' performance.

4.6 References

Akgün, A. E., Byrne, J., Keskin, H., Lynn, G. S., Imamoglu, S. Z. 2005. Knowledge networks in new product development projects: A transactive memory perspective. *Information and Management* 42(8), 1105-1120.

Akgün, A. E., Byrne, J. C., Lynn, G. S., Keskin, H., 2007. New product development in turbulent environments: Impact of improvisation and unlearning on new product performance. *Journal of Engineering and Technology Management*, 24(3), 203-230.

Benkler, Y. 2002. Coase's penguin, or, Linux and the nature of the firm. *Yale Law Journal* 112, 369-446.

Bonaccorsi, A., Giannangeli, S., Rossi, C. 2006. Entry strategies under competing standards: Hybrid business models in the Open Source software industry. *Management Science* 52(7), 1085-1098.

Brun, E., Saetre, A. S., Gjelsvik, M. 2009. Classification of ambiguity in new product development projects. *European Journal of Innovation Management* 12(1), 62 – 85.

Burt, R. S. 1992. Structural Holes. Cambridge, MA: Harvard University Press.

Capra, E., Francalanci, C., Merlo, F., Rossi-Lamastra, C. 2011. Firms' involvement in Open Source projects: A trade-off between software structural quality and popularity. *Journal of Systems and Software* 84(1), 144-161.

Colombo, M.G., Piva, E., Rossi-Lamastra, C. 2010. L'allocazione del potere decisionale nelle imprese Open Source. Un'analisi empirica. *Economia e Politica Industriale* 37, 63-97.

Dahlander, L., Frederiksen, L., Rullani, F. 2007. Online communities and open innovation: governance and symbolic value creation. *Industry and Innovation* 15(2), 115-123.

Dahlander, L., Magnusson, M. G. 2005. Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34(4), 481-493.

Dahlander, L., Wallin, M. W. 2006. A man on the inside: unlocking communities as complementary assets. *Research Policy* 35(8), 1243-1259.

Dahlander, L. 2007. Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source communities. *Industrial and Corporate Change* 16(5), 913-943.

Evers, S., 2000. An Introduction to Open Source Software Development. Technische Universität Berlin, Fachbereich Informatik, Fachgebiet Formale Modelle, Logik und Programmierung (FLP).

Fitzgerald, B. 2006. The transformation of Open Source Software. *MIS Quarterly* 30(3), 587-598.

Giuri, P., Ploner, M., Rullani F., and Torrisi. 2010. Skills, Division of Labor and Performance in Collective Inventions: Evidence from Open Source Software. *International Journal of Industrial Organization* 28(1), 54-68

Goldman, R., Gabriel, R. P. 2005. *Open Source as Business Strategy: Innovation Happens Elsewhere*. San Francisco, CA: Morgan Kaufmann.

Gruber, M., Henkel, J. 2006. New ventures based on open innovation -an empirical analysis of start-ups firms in embedded Linux. *International Journal of Technology Management* 33(4), 256-372.

Hargadon, A. B., Sutton., R. I. 1997. Technology brokering and innovation in a product development firm. *Administrative Science Quarterly* 42(4), 716-749.

Hargrave, T.J., Van de Ven, A. 2006. A collective action model of institutional innovation. *Academy of Management Review* 31(4), 864-888.

Hars, A., Ou, S. 2002. Working for free? Motivations for participating in Open-Source projects. *International Journal of Electronic Commerce* 6(3), 25-39.

Heckman, J. J. 1979. Sample Selection Bias as a Specification Error. *Econometrica* 47(1), 153-161.

Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35 (7), 953–969.

Henkel, J. 2009. Champions of revealing - The role of Open Source developers in commercial firms. *Industrial and Corporate Change* 18(3), 435-471.

Jayarama, J. 2007. Supplier involvement in new product development projects: dimensionality and contingency effects. *International Journal of Production Research*, 46(13), 3717-3735.

Jeppesen, L. B., Frederiksen, L. 2006. Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. *Organization Science* 17(1), 45-63.

Laursen, K., Salter, A. 2004. Searching high and low: what types of firms use universities as a source of innovation? *Research Policy* 33, 1201-1215.

Laursen, K., Salter, A. 2006. Open for innovation: the role of openness in explaining innovation performance among U.K. manufacturing firms. *Strategic Management Journal* 27, 131–150

Madey, G., ed., 2009. The SourceForge Research Data Archive (SRDA). University of Notre Dame (February 2009) http://zerlot.cse.nd.edu/

Mockus, A., Fielding, R.T., Herbsleb, J. 2000. A case study of open source software development: the Apache server. *Proceedings of the 2000 International Conference on Software Engineering.*

O'Mahony, S. 2003. Guarding the commons: How community managed software projects protect their work. R*esearch Policy* 32(7), 1179-1198.

O'Mahony, S., Bechky, B. A. 2008. Boundary organizations: Enabling collaboration among unexpected allies. *Administrative Science Quarterly* 53(3), 422-459.

O'Mahony, S., West, J. 2008. The role of participation architecture in growing sponsored Open Source communities. *Industry and Innovation* 15(2), 145-168.

Rucht, D. 2004. Movement allies, adversaries, and third parties. In H. Kriesi (eds.). The Blackwell Companion to Social Movements: 197–216. Malden, MA: Wiley-Blackwell.

Sproull, L., Dutton, W., Kiesler, S. 2007. Introduction to the Special Issue: Online Communities. *Organization Studies* 28(3), 277-281.

Stam, W. 2009. When does community participation enhance the performance of open source software companies? *Research Policy* 38, 1288-1299.

Timmermans, S., Leiter, V. 2000. The redemption of Thalidomide: Standardizing the risk of birth defects. *Social Studies of Science*, 30, 41-71.

von Hippel, E. 1988. The Sources of Innovation. Oxford University Press, Oxford.

von Hippel, E., von Krogh, G. 2003. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science* 14(2), 209-223.

von Krogh, G., Rossi-Lamastra, C., Haeflinger, S. 2012. Phenomenon-based research in management and organization science**: When is it Rigorous and Does it Matter?**. *Long Range Planning* 45(4), 277-298.

Schulze, A., Hoegl, M. 2006. Knowledge Creation in New Product Development Projects. *Journal of Management* 32(2), 210-236.

Shah, S. K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science* 52(7), 1000-1014.

Spaeth, S. 2005. Coordination in Open Source Projects - A Social Network Analysis using CVS data, doctoral dissertation.

Swink, M., Song, M., 2007. Effects of marketing-manufacturing integration on new product development time and competitive advantage. *Journal of Operations Management*, 25(1), 203-217.

Trott, P. 1998. Innovation Management and New Product Development, Prentice Hall.

Wasserman, S., Faust, K. 1994. Social network analysis: Methods and applications. Cambridge: Cambridge University Press.

West, J. 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32(7), 1259–1285.