12-16-2010

Incentivizing Technological Growth: A Symbiotic Relationship in the Computer Software Industry

Alex Usher
University of Puget Sound

Incentivizing Technological Growth: A Symbiotic Relationship in the Computer Software Industry

Alex Usher

12/16/2010

I. Introduction

It is difficult to argue that any given person goes more than a day without exploiting the advancements of computer software technology. Whether it is from the newest release of the video game Call of Duty or the innovative "Snap" feature in Windows 7, people benefit daily from the progression of software innovation. Software is generally defined as a collection of written programs and procedures, usually referred to as code, that instruct a computer what to do. With household computer and internet access up to 68 percent in 2009 and computer software engineer employment expected to increase by 32 percent in the next 10 years, it is also difficult to ignore the growing importance of the software industry in the nation's economy[1] (Computer Use and Ownership) (Computer Software Engineers). It is for this reason, economists have been investigating the economics of the software industry and the role of intellectual property rights in the high rate of technological growth.

Of particular interest to many researchers is the notion of the open source software communities and their ability, and willingness, to stimulate innovation without taking advantage of traditional copyrights or patents. Open source communities consist of developers, sometimes at several different locations or organizations, sharing program code and making it freely accessible to copy, study, refine and modify (Lerner & Jean, 2002).Conversely, proprietary software is exclusively the property of its developers and

[1] According to the Bureau of Labor Statistics, the expected growth of employment in software engineering is much faster than the average of all other occupations. Already holding approximately 1.3 million jobs in 2008, computer software programmers and engineers are expected to see more than 300,000 new jobs created in the next 10 years.

copyrighted or patented against exploitation and duplication. Open source software development has obvious cost-saving traits; stemming from high knowledge diffusion, code reuse, and lowered transaction costs, that proprietary firms cannot replicate but could potentially benefit from.

While proprietary firms and open source communities exhibit many differences, their existence in the computer software industry causes them to both experience a unique industry trait; a high knowledge-based market. The software industry is characterized by network externalities and first-mover advantages that cause innovation to be the key to a firm's success. As a result of these characteristics, new proprietary firms, motivated by the promise of high profits, can only enter and survive in the market with innovative products.

Logically, two questions arise from the notion of open source: Why would developers chose to participate in programming without the incentives of proprietary code or the ability to gain profit from its sale? Secondly, if open source software programmers have strong enough motivations to innovate without profit compensation and do it at lower costs than proprietary firms, what is the need for patents or copyrights? Economists have found the motivations of open source software, or OSS, developers of particular interest and have conducted several studies on the topic.[2] Generally, findings show that a large motivation of OSS programmers is the ability to clearly signal their skills to potential employers through the visibility of open source code. By and large, these employers are proprietary firms, who can afford to hire these programmers due to the profits gained from securing forms of protection of intellectual property rights and then selling software.

[2] Details and findings of these studies will be discussed later in the paper during the literature review.

Copyrights and patents allow proprietary firms to invest in research and development without having to worry about the public good characteristics of computer software leading to free riders undermining their profits. Public goods are goods that exhibit nondiminishability and nonexcludability allowing free riders take advantage of these properties by reaping the benefits of one firm's innovations without experiencing any costs. (Riddell, Shackelford, Schneider, & Stamos, 2009).[3]

Subsequently, even though OSS communities innovate at lower costs, they rely heavily on potential job opportunity motivations from proprietary firms who, in turn, are dependent on copyrights. While copyrights overcome the public good nature of computer software, it also leads to preventable higher fixed costs. I will illustrate that open source software firms help reduce these costs, but would struggle to exist and succeed without the incentives provided from the proprietary industry. By lowering costs, profits for proprietary firms increase which stimulates entry and innovation in the software market. Therefore, a mixed market is favorable due to the interdependence and symbiotic relationship of the two markets inducing lower costs that allow the computer software industry to maintain a high level of technological growth.

The paper's discussion is structured as follows. In Section II I will review relevant literature and note important observations and/or findings. The literature review will also offer a closer look into open source communities by assessing information previous

[3] Nondiminishability refers to the property that one person's consumption does not restrict the amount of any other persons. Nonexcludability refers to the property where "nonpayers" cannot be excluded from using the good.

researchers have acquired on OSS motivations, operations, and cost-saving behaviors. In Section III I will investigate the market failure issues behind the unique characteristics of the software industry and conceptualize the problem diagrammatically. Next, in Section IV I will introduce and examine the symbiotic relationship between open source communities and the resulting profits that lead to a higher level of innovation. Lastly, in Section V I will address limitations of the paper and, finally, I will discuss policy implications in Section VI and Section VII I will conclude.

II. Literature Review

It is important to acknowledge the relationship between the direction of this paper and studies in earlier literature to understand how this research is relevant to current research in the economics of computer science and to demonstrate how this particular argument offers a new analysis. The relevant literature is separated into two general topics; (i) Motivations behind Open Source Communities and (ii) Benefits derived from OSS. Some literature falls into more than once category.

(i) Motivations behind Open Source Communities

The development of open source communities astonished and intrigued economists from the very beginning. It is imperative to understand the underlining incentives for open source developers' since their motivations are arguably the one of the causes for open source communities' dependence on proprietary firms. Frestchman and Gandal (2007) investigate this "economic paradox" with an empirical study and survey of open source developers. They find that output per contributor in OSS projects is higher with less restrictive licenses, implying that the programmers have signaling and status incentives for participating in OSS projects.

Similarly, Lerner and Jean (2002) suggest that even though the "media like(s) to portray the open source community as wanting to help mankind, as it makes a good story" , an explanation based on altruism can only go so far (Lerner & Jean, 2002, p. 2). As a result, they explore the incentives behind OSS and report that there are two distinct motivations; career concerns and ego gratification. "Career concerns" reflects the notion of potential job opportunities resulting from the developer being able to signal his/her skills by being named directly as a contributor to a highly visible piece of programming. Hiring companies are able to evaluate a programmer's skills, creativity and initiative because the source code is free to view. Lerner and Jean (2002) go on to explain that peer recognition is what spurs an "ego gratification" incentive to participate in OSS projects; contributors are able to signal to their ability to fellow peers.

Bizter and Schroder (2007) take this one step further, suggesting that the combination of ego gratification and career concern signaling results in a motivation to produce higher quality programming work than that of a proprietary firm. They also discuss the motivation of the "nature of voluntarism"; explaining that people usually contribute to projects they enjoy doing. This also relates to the argument of altruism as a potential incentive to contributing to OSS development. Collectively, all of these findings offer insight into the OSS communities and their motivation from potential career opportunities that proprietary firms offer.

(ii) Benefits derived from OSS

Generally speaking, most of the literature regarding OSS benefits advocated that the largest gain to be had from implementing open source software was associated with cost-saving characteristics. By demonstrating that open source communities innovate at lower costs, I can point that their existence reduces costs and subsequently increases profits in the software industry.

Returning to Bitzer and Schroder's research, they suggest that signaling motivation behind open source contributors help lower development costs. This is accomplished by generating higher quality work, due to career/ego concerns, which results in lower future debugging costs and lessens the need for technical support.[4] In their research Bitzer and Schroder(2007) also coin the term "*boundless cooperation*", referring to the idea that " because commercial exploitation of newly developed software is not intended, there is no need to keep new ideas secret and therefore barriers against cooperation do not arise". Boundless cooperation leads to lower costs by inducing high knowledge diffusion and prompts combinations of complementary programming skills. The concept of boundless cooperation in open source software also allows for basic code reuse. This idea supports the argument of lowering innovation costs by lowering the time needed to code certain

[4] Debugging refers to fixing "bugs", or defects, in computer program code.

programs. This lessens a developer's opportunity cost of writing the program, allowing them to quickly move onto novel coding aspects.

Similar to Bitzer and Schroder's analysis of OSS benefits, Lerner and Jean (2002) discuss the advantages of lowering costs through cooperation. They discuss the idea of forking, the "splitting of projects into competing developmental streams" and how while forking can be beneficial, it is a delicate balance that sometimes turns into a destructive process. This could happen if competing streams get into disputes about the project, leading to the development of two entirely different programs created for the same purpose, which is clearly a waste of resources. Lastly, Mendez-Duron and Garcia (2009) use social capital, or knowledge flows, to show that investment in OSS communities and the knowledge flows they cultivate results in higher returns.[5] They also suggest that some of these knowledge flows can support healthy forking within communities.

While most of the economic literature investigates OSS and the benefits that are derived from its development, none demonstrate the goal of this paper to show how these benefits can lead to maintaining a higher level of technological growth in a symbiotic mixed market.

III. Knowledge Spillovers and Market Failure in Proprietary Software

With a knowledge intensive industry such as software engineering, the exchange of ideas, or knowledge spillovers, are almost impossible to avoid and can cause serious problems within a market. It is important to understand the difference between *internal*

---

[5] Medez-Duron and Garcia define social capital as " the actual potential resources embedded in relationships amoung actors" or "knowledge flows". They also define "returns" as the projects diffusion over the network.

knowledge spillovers and *external* knowledge spillovers, for they have very different economic implications. Internal knowledge spillovers are interparty exchanges of ideas that can facilitate "creation of new related goods and new ways of producing existing goods" (Carlino, 2001, p. 17). Not surprisingly, internal knowledge spillovers, or what Mendez-Duron and Garcia(2009) called knowledge flows, are exactly what helps open source software practice healthy forking strategies and experience high levels of innovation. Additionally, the significance of these knowledge spillovers is what causes the software industry to rely so heavily on innovation.

However, in opposition, external knowledge spillovers are exchanges of ideas between two different, competing parties. External knowledge spillovers could allow outside companies or firms to benefit from one firm's innovations without incurring any of the costs associated with research and development. This concept is referred to as a positive externality and in section one of this paper this idea was discussed in relation to public goods and free riders.[6] Positive externalities tend to lead to an underallocation of resources and result in less than the socially desirable amount of a good or, in the worst cases, even no production (Riddell, Shackelford, Schneider, & Stamos, 2009). Copyrights and patents allow innovators to internalize these positive externalities by restricting outside parties use, visibility, or replication of a good. More specifically, firms can charge for use of copyrighted or patented information and payments to the firm help them internalize the external benefits.

---

[6] A positive externality is formally defined as when benefits are reaped by a third party; one that is not directly involved in the transaction.

The concept of positive externalities in intellectual property rights can be applied directly to the software industry. Despite the obvious social benefit, the computer software industry experiences little or no demand from the consumers due to free riders taking advantage of positive externalities. Without demand there is no profit, and without profit there is no incentive for firms to produce, even though there is social benefit to be gained. This can result in a social welfare loss; a market failure where it is possible to reallocate resources to better the society. This market failure can be illustrated diagrammatically for the proprietary software industry; however an understanding of software firm's unique marginal costs is needed first.

Marginal cost is defined as the change in total cost that results from a one-unit change in output (Frank, 2008). In the case of software production, the duplication (or creation of another unit of software) of a program is as simple as just writing another disk.[7] The cost of this process is so small that it has virtually no effect on the total cost of operation for a proprietary software firm. As a result of this, the marginal cost can be thought to be essentially zero in the computer software industry. Given this information, the proprietary software market – with no intellectual property rights – can be shown in the following diagram:

[7] This process is also referred to as burning to a CD.

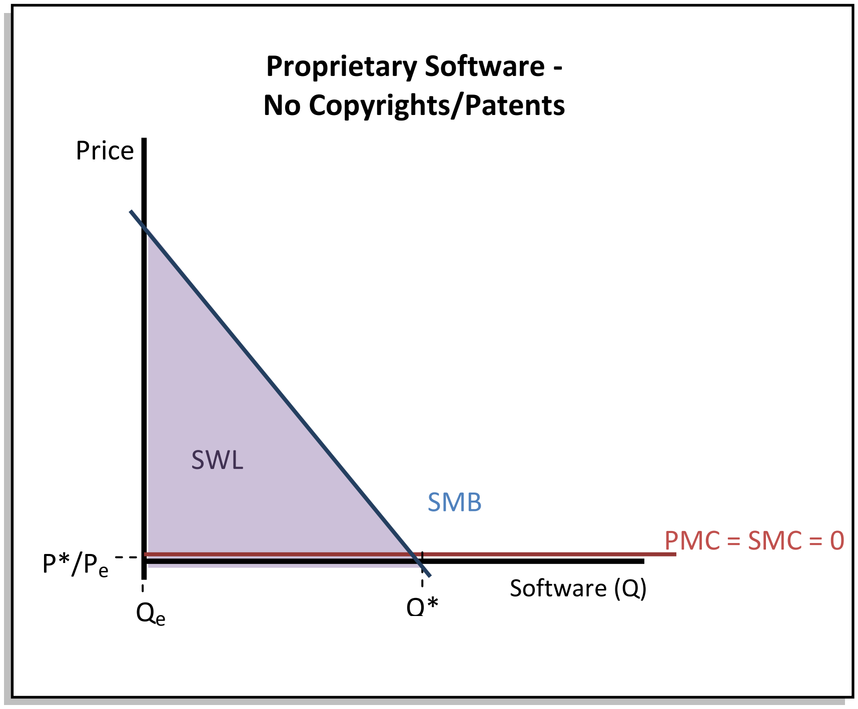

Figure 1

It can be seen in the diagram that market's optimal equilibrium, where social marginal benefit (SMB) intersects the social marginal cost curve (SMC), is at (Q*, P*). However, as mentioned earlier, with no copyrights or patents to protect against free-riding there will be no demand, resulting in no production. This causes the actual equilibrium (Q_e,P_e) to be at (0,0) where there is no production despite the products ability to provide social benefit. Consequently, the market failure can be thought as the forgone benefit of not producing at the optimal level of Q* and is depicted in the diagram as a social welfare loss (shaded triangle).

In an attempt to internalize these benefits and eliminate a free-rider problem, copyrights and patents were introduced to the computer software industry. By abolishing free-riders, copyrights and patents take away consumers ability to enjoy software products without incurring costs. However, since there is still a benefit from the programs to be had, consumers are forced to pay an amount, based on their willingness and ability, for the software and therefore creating a demand in the market that is equal to the marginal social benefit of the programs. Again, this can be depicted graphically:

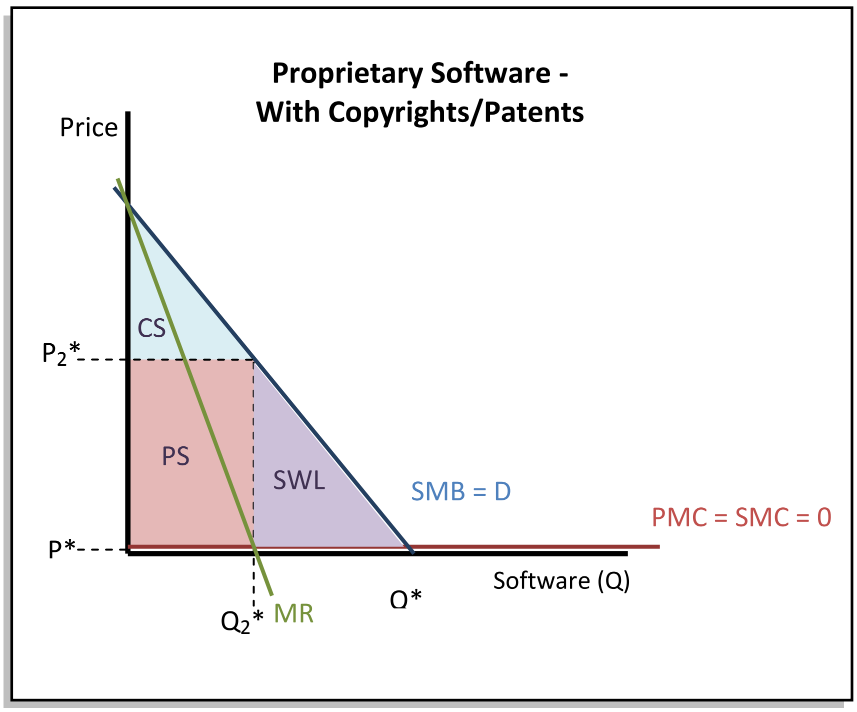

Figure 2

In figure two, the traditional economic theory of monopoly power derived from copyrights is assumed, resulting in the marginal revenue curve having twice the slope of the demand curve.[8] Under monopoly assumptions, the market determined equilibrium is

---

[8] Marginal revenue is the amount of total revenue gained after a one unit increase in output (Frank, 2008).

found at quantity Q_2* and price P_2*. It can be seen that the protection provided by copyrights and patents allow for the market to experience a positive equilibrium and capture both a consumer and producer surplus (CS and PS, respectively) as well as ultimately allowing a firm to experience profits. Additionally, the diagram shows that although there is still a social welfare loss associated with the monopoly power derived from patent protection, the loss is significantly less than the industry without copyrights or patents depicted in Figure 1.

While copyrights and patents help establish a better market for the computer software industry, they also help stimulate innovation. Firms are unable to invest in research and development if they do not have the funding from their profits and they do not experience profit without copyright/patent protection. More specifically, Figure 2 illustrated the creation of a producer surplus, also known to be equal to total fixed costs plus profit, after the copyrights are introduced. With the profits being made, firms are enticed to enter the market and due to the nature of the software industry, a firm can only enter this type of a market with innovative products, thus the promise of high profits provides a large incentive to innovate. Conclusively, it is shown that copyrights and patents induce profits that enable firms to sustain the high level of innovation the industry demands.

IV. Proprietary and Open Source : A Symbiotic Relationship

In order to examine the effect of OSS communities in the software market, the relationship of open source and proprietary firms needs to be investigated. After reviewing previous literature and research in the computer software industry, it became apparent that there existed an interdependent, complementary connection between the two types of parties. The most applicable term for the relationship seemed to be *symbiotic*; defined as a mutually beneficial relationship between different people or groups (Oxford Dictionaries). In this specific case, it is evident that proprietary firms derive benefits from open source communities while OSS developers simultaneously obtain benefits from those proprietary firms. In this section of the paper both sides of this relationship will be investigated and clarified to support the theory of a symbiotic relationship.

Initially, let us examine the prospect of open source developers, and their communities, benefiting from proprietary firms. As introduced during the review of literature, it has been reasoned that OSS contributors have two main motivations that overshadow altruistic incentives; career concerns and ego gratification (Fershtman & Gandal, 2007). Both of these incentives refer to the developer's ability to signal his/her skills (to either a potential employer or fellow peer) due to the extreme visibility of the software and the convention which credits the authors by name. While this illustrates a definite reliance on proprietary firms, it does not *explicitly* illustrate the benefits OSS communities are deriving from the companies. In order to understand these benefits, the implications of potential job opportunities motivating developers needs to be explored. Recall that earlier during the discussion of OSS benefits, it was noted that career incentives motivate open source developers to produce higher quality software. Furthermore, this superior quality of code induces lower costs by lessening the need for technical support and debugging (Bitzer & Schroder, 2007). It should also be noted that these lower costs may be experienced in the development of new and innovative software because higher quality code allows for a more stable foundation for advancement. Lastly, it can now be recognized that this beneficial cost reduction in innovation ultimately stems from the proprietary firms interaction with open source communities because the firms are the subjects inducing cost-saving behavior.

Given that open source communities benefit from proprietary firms has been established, the reverse can be considered. The most obvious potential gain proprietary firms could obtain from open source developers is through their unpatented innovations. Without any property rights preventing proprietary firms from exploiting the visibility and availability of OSS code, they could utilize the information in beneficial way or even employ it to spur their own innovation.

However, the specific licenses used by the OSS communities may dictate the extent to which others, particularly proprietary firms, can benefit from the code by restricting certain aspects or uses. There are two general types of open source licenses: permissive and non-permissive. Non-permissive licenses, also referred to as copyleft or GNU licenses, require that all derived works can only be distributed under the same license terms[9]. This indicates that an open source software program must remain open source after modifications or supplementations if it is being shared or sold. Conversely, permissive licenses allow derived works to be redistributed on more restrictive license terms. For example, an OSS program protected under a permissive license does not require derived works to be open source code. (Lerner & Jean, 2002).

---

[9] GNU is a General Public License used widely with the software industry.

Nevertheless, proprietary firms can still benefit from OSS communities despite which open source licenses they encounter. By either direct exploitation of the open source code or by indirectly employing its concepts, proprietary developers can benefit from OSS communities the same way open source contributors benefit from their own work. They can experience lower costs by code reuse, favorable knowledge diffusion, and combinations of complementary programming skills; all of which lower costs of innovation and code development. Consequently, these observations support the idea that open source communities and proprietary firms experience a symbiotic relationship.

After investigation of both sides of the relationship, it can be noted that almost all the benefits resulting from the symbiotic bond between OSS and proprietary firms are actually reductions in costs, particularly in the creation and innovation of new software. If we now take the previous diagram for the software market into back into consideration, we can capture the innovation implications of open source communities' existence in the software industry. Again, it was concluded that the symbiotic nature of proprietary and OSS caused benefits in the form of lower costs.

Recall that with no marginal costs present for software firms, these lowered costs must be experience in firms' total fixed costs (TFC). Additionally, if these costs are only affecting TFC, a curve that was not incorporated in either diagram, than the producer surplus formed in Figure 2 is not changing with the introduction of OSS communities to the software market. This indicates that the amount of producer surplus is not changing even though total fixed costs are decreasing. For this to be possible, given that producer surplus is equal to TFC plus profit as established earlier, profits must be increasing. With profits increasing, further entry into the market is experienced and with this entry comes even more innovative products.

Ultimately, open source communities' entrance into the market allows for proprietary firms to enhance their profitability, thus increasing incentives to innovate within the computer software industry. This analysis suggests that with a mixed market of both proprietary firms and open source software communities, the computer software industry can reach a higher level of technological growth by exploiting the lower costs derived from a symbiotic relationship.

V. Limitations

As with all research, there are some small limitations that accompany the theories and that sometimes condition relevant findings. In this paper's case, the foremost limitation is the narrowed focus on a very simplistic, pure mixed market of only two types of firms within the computer software industry; proprietary and OSS communities. Other research has been conducted on the distinction between open source *communities* and open source *firms*; the key difference stemming from the pursuit of profit by OSS firms (Linus Dahlander, 2005). However, this paper only analyzes the relationship of OSS communities and proprietary firms in the research and development market.

Similar to the narrowed focus of just one branch of open source population, this paper also only investigated one type of proprietary firm; the firms that do not exercise any of its own open source initiatives. Some companies have recognized the symbiotic relationship between OSS and proprietary and have established their own affiliated open source communities or sectors within their firm[10] (Campbell-Kelly & Swartz, 2010). Dahlander and Magnussion (2005) go into investigate the managerial requirements that make it possible to cultivate a symbiotic relationship, rather than commenalistic or parasitic, in this type of situation.

While both of these factors are limitations, they still provide insight into the validity of argument presented in the paper. It is argued that a mixed market provides a higher, socially desirable level of technological growth in the form of more effective software. In an industry characterized by network externalities and first-mover advantages, innovation is key to a firm's success. It is apparent that companies and communities have realized the advantages of a mixed market and have begun trying to cultivate these benefits through mixed firms to achieve higher levels of innovation.

VI. Policy Implications

While this paper does not lead up to explicit policy suggestions or critiques, its results have a relatively general implication. The conclusions demonstrate that the computer software industry thrives as a mixed market. The symbiotic relationship is a crucial component in technological growth and its contribution to the success of the software industry needs to be taken into consideration before any policies are enacted. Whether it

[10] Interestingly, Cambell-Kelly and Garcia-Swartz (2010) argue that there will soon be a "convergence to the middle" of open source and proprietary industries; implying that there will no longer be two separate industries. They believe all firms will exploit the benefits of both open source and proprietary programming.

entails changes in copyrights and patents or alterations to open source licenses, policies that aim to only influence one particular group of developers will ultimately affect the entire software industry due to the connection and beneficial relationship OSS and proprietary firms enjoy. It is for this reason that policy implementations or modifications need to be taken into serious consideration beforehand.

VII. Conclusion

The atypical economic characteristics in the software industry, and open source attributes in particular, have raised several intriguing questions over the past few years. Open source communities' success, operations, and even existent have been the focus of many research papers in the economic discipline. As a result of this, a review of previous literature was used to establish the motivations behind contribution to open source projects and the benefits that can result from the operations of their communities. It was found that a major motivating factor for OSS developers was the opportunity to better signal their programming abilities to potential employers, i.e. proprietary companies. Subsequently, these incentives allowed open source communities to experience lower technical support and debugging costs as a result of producing higher quality code created to demonstrate their level of coding capability. OSS groups acquired even more cost reductions in innovation from constructive knowledge flows, code reuse and the implications of "*boundless cooperation*".

After analyzing the success of open source communities in technological growth, proprietary firms' reliance on copyrights and patents was questioned. However, it was then shown diagrammatically that proprietary firms use these rights to internalize the positive externalities, create a consumer demand, and thus gain profit from a monopolistic power over their products. By employing copyrights or patents, proprietary companies are incentivized to maintain a higher level of technological growth.

Lastly, it was argued that even though open source communities can achieve a high level of innovation at lower costs than proprietary firms, they would not be able to succeed or maintain a high level of innovation without them. This is because OSS would not be has beneficial if it were not for the career incentives provided by proprietary firms that motivate programmers to contribute high quality work to open source projects in the first place. The apparent interdependence of the two firms was confirmed after exploring the possibility that proprietary firms could be benefiting from interactions with OSS communities and developers. It was established that proprietary companies in fact acquire lower innovation costs derived from the exploitation of open source projects; the extent of such depending on varying OSS licenses. Collectively, the lower costs of innovation being experienced in both open source communities and proprietary firms prompted analysis of its effect in the previously discussed diagrams. Using the analytics from before, it was shown that with a symbiotic-natured mixed market of OSS communities and proprietary companies, the total fixed cost of firms decreases; thus allowing for a higher profits and greater incentive for technological growth.

While the implication of further stimulating technological growth as result of a mixed market has its merits, further research could be done. As mentioned earlier in regards to limitations of the paper, some companies have acknowledge the symbiotic nature of the computer software industry and have began incorporating open source communities and initiatives in their own firms. Some researchers suggest that there will ultimately be a convergence of industries and that all firms will incorporate both open source and proprietary coding into their companies. It would be valuable to investigate whether this convergence is plausible and if it would lessen, maintain, or increase the current level of technological growth. Nevertheless, the computer software industry is currently operating at a high level of technological growth and will continue to do so by utilizing the symbiotic nature of the relationship between proprietary firms and open source software communities.

Works Cited

Bitzer, J., & Schroder, P. J. (2007). Open Source Software, Competition and Innovation. *Industry and Innovation , 14* (5), 461-476.

Campbell-Kelly, M., & Swartz, D. G. (2010). The Move to the Middle: Convergence of the Open Source and Proprietary Software Industries. *International Journal of the Economics of Business* , 223-252.

Carlino, G. A. (2001). Knowledge Spillovers: Cities' Role in the New Economy. *Business Review* , 17-27.

Computer Software Engineers . (n.d.). Retrieved November 7, 2010, from Bureau of Labor Statistics: http://www.bls.gov/oco/ocos303.htm

Computer Use and Ownership. (n.d.). Retrieved November 7, 2010, from US Census Bureau: http://www.census.gov/population/www/socdemo/computer.html

Fershtman, C., & Gandal, N. (2007). Open Source Software: Motivation and Restrictive Licensing. *International Economics and Economic Policy , 4* (2), 209-225.

Frank, R. (2008). *Microeconomics and Behavior .* New York: McGraw-Hill.

Garzarelli. (2008). Open source software and economic growth: A classical division of labor perspective. *Information Technology for Development , 14* (2), 116-135.

Henkel, J. (2009). Champions of Revealing--The Role of Open Source Developers in Commercial Firms. *Industrial and Corporate Change , 18* (3), 435-471.

Lerner, J., & Jean, T. (2002). The Simple Economics of Open Source. *Journal of industrial Economics , 50* (2), 197-234.

Linus Dahlander, M. G. (2005). Relationships between open source software companies and communties: Observations from Nordic firms. *Elsevier* , 482-493.

Mendez-Duron, R., & Garcia, C. E. (2009). Returns from Social Capital in Open Source Software Networks. *Journal of Evolutionary Economics , 19* (2), 277-295.

Open Source Licenses. (n.d.). Retrieved November 10, 2010, from Open Source Initiative : http://www.opensource.org/docs/osd

Orman, W. H. (2008). Giving it Away for Free? The Nature of Job-Market Signaling by Open-Source Software Developers. *B.E. Journal of Economic Analyaia and Policy , 8* (1).

Oxford Dictionaries. (n.d.). Retrieved November 10, 2010, from http://oxforddictionaries.com/view/entry/m_en_us1296460#m_en_us1296460.00

Riddell, T., Shackelford, J., Schneider, G., & Stamos, S. (2009). *Economincs: A Tool for Critically Understanding Society.* Boston: Pearson.