

Association for Information Systems AIS Electronic Library (AISeL)

ICIS 2008 Proceedings

International Conference on Information Systems
(ICIS)

2008

Software Design Strategies in Markets with Open Source Competitors

Zhe Qu

University of Texas - Pan American, quz@utpa.edu

Follow this and additional works at: <http://aisel.aisnet.org/icis2008>

Recommended Citation

Qu, Zhe, "Software Design Strategies in Markets with Open Source Competitors" (2008). *ICIS 2008 Proceedings*. 42.
<http://aisel.aisnet.org/icis2008/42>

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

SOFTWARE DESIGN STRATEGIES IN MARKETS WITH OPEN SOURCE COMPETITORS

*Stratégies de conception de logiciels pour les marchés confrontés à la concurrence
des logiciels libres*

Completed Research Paper

Zhe Qu

Department of Computer Information Systems & Quantitative Methods
College of Business Administration, University of Texas-Pan American
1201 W University Drive, MAGC 3.302
Edinburg TX 78539
Quz@utpa.edu

Abstract

There is a growing body of literature investigating the strategic interaction between proprietary software vendors (PRVs) and their open source (OS) counterparts. Most prior studies focus on software market where PRVs' revenue comes mostly from software license. Nevertheless, open source software (OSS) is prominent in software markets where both software license and software service contract (for the purpose of maintenance, technical support, upgrade, etc) constitute some considerable portion of PRVs' revenue. This paper aims to underline the difference between software characterized by high demand for service and by low demand for service in studying how OSS shapes PRV's software design strategy. The analysis of an economic model implies a magnetic phenomenon in the design of proprietary software with high demand for service, which is not observed in the design of software with low demand for service. The findings imply that in software markets characterized by high demand for service, when OSS is not very sophisticated PRV could introduce more basic version of their software in response to the growth of the OS counterparts. This implication is found to be very much inline with industry practice.

Keywords: Open source software, proprietary software, maintenance, support, competition, software designs, pricing

Résumé

Ce papier examine comment les logiciels libres contraignent la stratégie de conception chez les vendeurs de logiciels propriétaires. Une analyse par modélisation économique indique un phénomène magnétique dans la conception de logiciels, lorsque ceux-ci nécessitent une forte demande de services associés (maintenance...), mais pas pour les logiciels à faible demande de services. Ce résultat est en accord avec les pratiques constatées dans le secteur.

Introduction

Despite all the critiques about open source software (OSS), OSS has made significant inroads in many areas of software development, from operating systems (Linux), programming languages (Perl) and Web browsers (Mozilla Firefox) to Web servers (Apache), database management systems (MySQL), mail management systems (Sendmail) and typesetting engines (Tex) (Bitzer and Schroder 2006; Economides and Katsamakos 2006; Gaudeul 2007; Lerner and Tirole 2002; Lerner and Tirole 2005; Schmidt and Schnitzer 2003; Sen 2007). The growing market share and volume of installed base of some OSS force major proprietary software vendors (PRVs) to seriously consider the competitive position of OSS in the industry and employ a variety of strategies to counter the challenges of successful open source (OS) movement. For instance, Oracle stated in its analysis of key competition in 2007 that “The enterprise software industry is intensely competitive...open source alternatives such as MySQL AB in database, Red Hat, Inc. in middle ware, and SugarCRM Inc. in applications, are also impacting the competitive environment...in the sale of database software...our competitors include ... the open source databases, MySQL and PostgreSQL...In the sale of middleware products..., our competitors include... open source vendors such as Red Hat, Apache and ObjectWeb. ...Business intelligence competitors include...open source vendors Netezza Corporation...” (Oracle annual report 2007) Sun Microsystems similarly stated in its 2007 annual report that “we are seeing increased competition and pricing pressures from competitors offering systems running Linux software and other open source software.” BEA Systems acknowledged that “competitive pressures and open source availability of functionally competitive software could require us to reduce the price of our products and related services, which could harm our business.”

Facing the challenges from OSS, commercial firms employ a number of strategies and build innovative business models to counter the challenges and to capitalize on OS movement (e.g., Bank 2003; Lawton 2007; Tam 2004). For example, Red Hat uses a support business model where it commercially provides complementary services that are not supplied efficiently by the Linux community. Major PRVs such as Oracle, HP, Novell, IBM, SAP and Sun Microsystems are well-known advocates of OS movement (Kerstetter 2004; Stone 2004; Tam 2004; Vara 2006). For instance, Oracle started selling its own technical support for Red Hat's version of Linux in 2007; Sun Microsystems is also determined to offer free downloads of its Solaris software; In 2004, IBM submitted a lightweight database called Cloudscape to the OS community - Apache Software Foundation, an organization that manages the widely used Apache Web server.

Academic researchers have also made significant contribution examining the issues for competitive strategy in the landscape of OS movement (e. g., Bessen 2006; Bitzer 2004; Dahlander and Magnusson 2006; Economides and Katsamakos 2006a; Haruvy et al. 2005; Lerner and Tirole 2002; Lerner and Tirole 2005; Rossi and Bonaccorsi 2006; Schmidt and Schnitzer 2003). Despite the growing body of literature investigating the strategic issues between PRVs and their OS counterparts, most of the prior pieces focus on software market where vendors' revenue mainly comes from software license. Nevertheless, OSS is also prominent in software markets where there is significant demand for maintenance and support service after the sale of software license. That is, both software license and post-sale service contract represent considerable portion of some PRVs' revenue. This is shown in Table 1.

Table 1 is a snapshot of influential OSS. It shows that OSS is not only prominent and known in software markets where users do not require much support and service (e.g., Web browser and typesetting), but also in software markets where a significant portion of PRVs' revenue comes from maintenance and support of software (e.g., infrastructure software and database management systems). For instance, as one of the world's largest database software providers, 33% of Oracle's total revenue in 2007 came from software license, and 46% from software maintenance and product support; maintenance and support also represented the highest margin business unit for Oracle (Oracle annual report 2007). Sun Microsystems, a major provider of infrastructure product, earned

Table 1: Major Open Source Projects

Program	Nature of program	Year of introduction	Competitors	Market penetration	For profit?	Business model
MySQL	DBMS	1995	Oracle, IBM, Microsoft	>1 million installations (2005)	Yes MySQL	Warranty, customer support, optionally additional functionality
Linux	Server operating system	1991	Microsoft Windows	25.7% of total shipments in 2008 (IDC prediction)	Yes Red Hat	Support and other complementary services of Linux
Sendmail	Internet mail transfer agent	1979	Microsoft Exchange	Handle ~80% of Internet e-mail traffic (2002)	Yes Sendmail Inc.	Service
Apache	Web server	1994	Microsoft IIS	~55% (February 2008)	No	N/A
Perl	System administration and programming language	1987	Java (Sun) VB, ActiveX (Microsoft)	Estimated to have 1 million users (2002)	No	N/A
Tex	Typesetting	1982	Microsoft Word	N/A	No	N/A
Mozilla Firefox	Web browser	2004	Microsoft IE	17.83% of the recorded usage share of Web browsers (March 2008)	No	N/A

Note: information presented in the table is from Dahlander and Magnusson (2006), Economides and Katsamakas (2006b), Gaudeul (2007), Lerner and Tirole (2002), Netcraft (2008), Net Applications (2008), Wheeler (2005), Wittig and Inkinen (2004).

28.56% of its total revenue in 2007 from support services consisting primarily of maintenance contract (Sun Microsystems annual report 2007). BEA Systems, Inc., a world leader in enterprise application and service infrastructure software¹, earned 40.9% of its revenue in 2007 from license fees and 59.1% from maintenance and support of its infrastructure software. Generally, customers who purchase enterprise software licenses may also enter into maintenance contracts covering technical support services, regular software maintenance, and software updates and enhancements. These maintenance contracts generate persistent revenue for enterprise PRVs after the transaction of software license (e.g., Banker and Slaughter 1997; Tan and Mookerjee 2005).

This paper is motivated by the significant difference between the two families of software, both of which witness notably dynamic and influential OS movement in recent years. Software like database management systems (DBMS) in general requires much more maintenance service and support than software for typesetting, Web browsing or gaming purpose. The amount of support and service needed is determined in general by the complexity and scope of the software and whether the software is mission-critical². For example, database types of software normally have higher complexity, are of larger scope and are more business-critical than software such as Web

¹ BEA Systems was acquired by Oracle in 2008.

² Poor design in software may also lead to high demand for support. In this paper, I focus on service demand driven by software characteristics and assume good quality in software design.

browser and computer games. As a result the former naturally require much more maintenance and support than the latter do. Furthermore, as support for software is complementary to software characterized by large demand for support service, the software itself and the support associated with it intervene with each other. Hence, it is important to examine them in an integrated framework; while for software with low demand for support service, examining the software alone would be sufficient.

The fundamental difference between the two families of software demands separate treatments in investigating commercial software firms' OS strategy. For example, do PRVs employ different OS strategies for software with low demand for service versus for software with high demand for service? How does the impact of OSS differ on the market structure for the two types of software market? The purpose of this Chapter is threefold. First, I investigate how activities in OS community enhance or replace the economic activities of PRVs. In particular, how does the rise of OSS have an effect on PRV's software design strategy in terms of number of features and functions bundled in the software? Second, I investigate the impact of OSS on software market structure and on pricing strategy of PRV. Third, I examine how the impacts of OSS differ between software market characterized by high demand for service and by low demand for service.

The remainder of the paper is organized as follows. The next Section provides a review of the literature. Then I present a case study of MySQL followed by an economic model. Analysis results of the model are presented in the Section of Equilibrium followed by managerial implications. The last Section concludes.

Literature Review

This paper is related with two streams of research: commercial software firms' OS strategies, and the impact of OS movement on PRS design. There is much discussion among industry experts about commercial software firms' OS strategies (e.g., Bank 2003; Lawton 2007; Tam 2004). The key question among these is: "How do firms seeking to sell products compete with *free*?" (von Krogh and von Hippel 2006) Academic researchers summarize these strategies into four categories: a) market segmentation strategy (i.e., PRVs cover a market segment different from the one covered by OSS) (Bessen 2006; Bitzer 2004); b) complementary product/service strategy (Dahlander and Magnusson 2006; Economides and Katsamakos 2006b; Haruvy et al. 2005; Lerner and Tirole 2002; Lerner and Tirole 2005; Schmidt and Schnitzer 2003); c) subsidy/support strategy (i.e., PRVs participate in or subsidize OS projects) (Bessen 2006; Dahlander and Magnusson 2006; Haruvy et al. 2005; Lerner and Tirole 2002; Lerner and Tirole 2005; Rossi and Bonaccorsi 2006); d) releasing proprietary codes strategy (Lerner and Tirole 2005).

Gaudeul (2005) investigates the direct competition between one OS and one proprietary project in a duopoly model where both the cost and benefit of OSS are compared with PRS. The cost of adopting OSS includes lack of coordination in development and lack of interface. The cost of adopting PRS is that developers may choose to develop a limited number of features. The author found that OSS and PRS can coexist in equilibrium, but OSS is used by low-income customers or by developers. This proves the validity and stability of the market segmentation strategy. Bitzer (2004) similarly showed that decreasing heterogeneity between OSS and PRS can drive the price of PRS below its average cost, leading to PRV's loss of incentive to develop PRS and forcing PRV to exit the market.

The complementary product/service strategy has been brought up in a number of studies. Economides and Katsamakos (2006a) developed a framework to investigate a platform firm's pricing strategy for the platform and for complementary applications to the platform (two-sided pricing strategy of a platform firm) and how the pricing strategy influence the industry structure of the proprietary system as well as the OS system. They found that the overall profit of a proprietary system is larger than an OS system when users have a strong preference for application variety; however the variety of applications in the OS system is larger than in the proprietary system. The authors also examined the competition between OS platform and proprietary platform. They found that a vertically integrated proprietary system (proprietary platform and applications to the platform are provided by one firm) has a larger market share than an OS system if there is a relatively large demand for the proprietary application, a large transition cost for the OS platform and a relatively small demand for application of the OS platform.

Although the payback of market segmentation strategy and complementary product/service strategy to commercial software firms is straightforward, researchers found that the gain of releasing codes to OS community is not guaranteed. Haruvy et al. (2005) examined conditions when a software firm should open its source codes versus close its source codes. They compared the revenue of a software provider under two scenarios: the software is

developed in an OS way, where the provider's revenue comes from complementary product of the OSS; the software is developed in a closed source way, where revenue comes from both software license and service. The authors demonstrated that when there are more productive in-house programmers and when the initial quality of the software is high, opening source codes may not be beneficial to a firm. On the other hand, opening source codes becomes a viable alternative to closed source only when the involvement of OS community in the project reaches a critical level. Moreover, OS is preferred when programmer wage exceeds some threshold, even with highly productive in-house programmers. Lerner and Tirole (2002) argued that PRVs release their proprietary codes to OS community in the hope that they can gain more by providing related services. However, releasing codes could be risky for PRVs if the payoff in the complementary service segment after the release does not exceed the payoff if the codes had not been converted to OS, which explains why the strategy of releasing codes to OS community is often observed in firms that are not major players in the proprietary segment.

Researchers also found that firms may have different incentives subsidizing or supporting OS projects. Some firms invest in OS development if the OSS is complement to the commercial software or hardware they produce (Schmidt and Schnitzer 2003). Bessen (2006) investigated why firms contribute to OSS development when there is PRS from non-rival firms. The author argued that complexity of software limits the number of features included in PRS. Under-provision of features created a market for OS community, which tends to develop more complex and geekier applications. Firms participate in OSS development for their own needs of software features. The author concludes that OSS extends software market rather than replace PRS. That is why OSS can grow in the presence of a competing PRV, and PRS can coexist with OSS because they cover different market segments. Mustonen (2005) investigated when a PRV may support the development of substitute OSS. The author showed that this could happen when the support creates compatibility between the programs and that the programs exhibit network effects. So supporting a rival OSS may allow the PRS and the OSS to share the same network. The resulting strong network effect of the PRS, however, is earned at the price of the PRV losing its network of old PRS, and at the cost of providing support to the OSS. Lerner and Tirole (2002) argued that PRV may want to subsidize the OS movement for strategic reasons such as weaken a rival or lessen the dependence on other vendors.

Another line of research on OS strategy is related to OS license. Kim et al. (2006) examined and compared three software pricing mechanisms: pricing of commercial software, pricing of OS product/service in OS dual license model, and pricing of open source product/service in OS support model. They investigated whether OS models are viable in monopoly and duopoly setting. They found that OS support model is viable in the presence of quality asymmetry among competitors no matter whether the quality of OSS is higher or lower than commercial software.

The second stream of research related to this paper concerns the impact of OS movement on PRS design, in particular, on the design of software features. Bitzer and Schroder (2007) stated that "the question of what impact this unusual development method (OS) has on innovation activity in the software sector has received surprisingly little attention thus far." They built a simple economic model and examined the impact of increased competition on innovation activity of both incumbents and entrants. They found that although incumbent commercial firms claim that technological progress will slow or even stop as a consequence of OSS entry, the model shows that it promotes innovation, that is, enterprises choose to increase the technological levels after the emergence of OSS. Bitzer and Schroder (2006) argued that the emergence of OSS has raised fears of a potential anti-innovation effect on two main issues. First, "the emergence of a no-cost competitor on the software market raised the question of whether commercial enterprise will be able to compete successfully...decreasing profits of commercial software producers will lower their ability to invest in R&D activity, thus resulting in slower technological progress in the software industry"; Second, "an anti-innovative effect of OSS may result if its development process is less efficient than that of commercially organized software." To examine the legitimacy of OSS's anti-innovation effect, the authors used the release history of Windows and Linux, Internet Explorer, Netscape's Web browser and Mozilla Firefox, as a potential indicator of technological progress in software industry. They found that "the data and cases presented do not offer evidence of an anti-innovative impact; on the contrary, if anything, the entry of OSS into commercial segments of the market appears to be associated with increased innovation activity." Economides and Katsamakos (2006b) extended the model developed by Economides and Katsamakos (2006a) and analyzed the innovation incentive in systems of two platforms: the OS and the proprietary platform. Comparing vis-a-vis the innovation investment of OS system and PR system, they found that the investment in application is stronger when the platform is OS. This is because the OS platform is available for free, enabling the application provider to set a larger price and capture a larger profit than the application provider for the proprietary platform. The authors also found that the level of investment in the application affects the level of investment in platform due to the complementarities

between the application and the platform. In particular, the marginal benefit of investing in the platform decreases with the level of investment in the application.

Overall, most of the prior research investigate one type of software in the landscape of OS movement and do not systematically compare software that is different in nature. One exception is the research by Sen (2007). Sen (2007) compared the impact of OSS on PRV's software usability and pricing decisions for two types of software: software with strong network effect versus software with weak network effect. The author found that the two types of software markets reveal very different behavior given the entry of OSS. In software markets with low network effect, PRV is better off in the presence of competition from OSS while PRV in markets with high network effect is threatened by the presence of OSS. Furthermore, PRV has little incentive to improve the usability of their software in markets with low network effect, while the same strategy may drive PRV to exit the market in software market characterized by strong network effect.

A Model

A Case of MySQL

I draw on the case of MySQL to promote the model. MySQL is a popular OS DBMS. It is chosen in the case analysis because DBMS is in general considered to have high demand for service, and there is serious competition between OSS and PRS in this software market. These characteristics of DBMS are essentially the focus of this study, hence the key elements in the model. Therefore MySQL represents a good case in point in this study.

The company MySQL was acquired by Sun Microsystems in February 2008. It was founded in 1995 and is headquartered in Sweden today. It has several offices throughout the world with thousands of community members worldwide. MySQL is in the list of OSS building blocks called LAMP, representing Linux, Apache, MySQL and Perl (Wittig and Inkinen 2004). It is used by Google, Yahoo!, Nokia, YouTube, Amazon and Travelocity among others. MySQL has received much attention of both media (e.g., Bank 2003; Lacy 2006; Lawton 2007; Tam 2004) and academic researchers (e.g., Dahlander and Magnusson 2006; Kim et al. 2006; Lerner and Tirole 2005; Mustonen 2005). Mustonen (2005), Wittig and Inkinen (2004), Wittig (2006), and Dahlander and Magnusson (2006) prepared MySQL cases and documented the history, current market position and business model of MySQL.

MySQL is known not only because of its high growth rate and large volume of installed base, but also because it challenges a high-margin database software category and could reshape the database market as Linux has reshaped the server operating systems market. In their case study of MySQL, Dahlander and Magnusson (2006) mentioned that the most prominent PRSs that are competitors of MySQL are the databases of Oracle and IBM. Industry analysts reported that "facing competition from open-source alternatives", not only Oracle, but also "IBM and Microsoft have each lowered their database prices and created low-end bundles aimed at smaller organizations and partners." (LaMonica 2005)

MySQL used to use a so-called dual license model: the software is available under both OS license GPL and proprietary license. In 2005 MySQL introduced the MySQL Network described as "a subscription service that provides updates, alerts, notifications, knowledge base and production level support, that make it possible for companies to easily manage hundreds or thousands of MySQL servers. The core MySQL database remains open source, but these services are only available to paying customers." MySQL's EVP of sales commented "we are following the Red Hat pattern. The subscription business is growth business for us and that will overtake our OEM business" (Wittig 2006).

MySQL has been improving its product. In October 2005, MySQL released version 5.0 which incorporated comparable features, such as stored procedures and triggers, to those of Oracle, IBM and Microsoft. These features were explicitly outlined by Oracle in 2004 to show the inferiority of MySQL (Wittig 2006). With more advanced features released, MySQL is put in more direct competition with Oracle, IBM and Microsoft.

Model Setup

I model the competition between a PRV and an OSS in a software market characterized by high demand for support and service. The PRV sells software license and support service for its proprietary software; the OSS is available for users free of charge; an OS service provider adopts a support model and sells support service for the

OSS. In the database market, Oracle is the PRV, MySQL database is the OSS, and MySQL is the service provider for the OS database software. I build the model on the literature of vertical differentiation³. Following industry practice, I model the decision making process of the players as a three-stage noncooperative game (with no collusion).

Stage 1: Software Design.

In software market characterized by high demand for service, an OSS with Q features is provided free of charge. A PRV chooses number of features/functions q for its software after observing Q . Let $q, Q \in [0, \bar{q}]$ where \bar{q} represents the maximum number of features/functions that can be developed for a PRV to earn a nonnegative profit. The development cost is $C(q)^4$ where $C'(q) > 0$, and $C''(q) > 0$.

Number of features/functions bundled in the software is treated as the PRV's decision variable in the first stage because of the following reasons. First, the model is intended to build an integrated framework including both software and subsequent support service for the software. The software engineering literature reports that the amount of maintenance service required for one software is closely related with the complexity in software design; more complex software requires more maintenance and support effort⁵ (e.g., Abreu and Melo 1996; Banker et al. 1998; Briand et al. 1999; Li et al. 1995). It has been shown that function point (a unit of measurement to express the amount of business functionality an information system provides to a user) in software is related with complexity and size of software, hence is related with maintenance effort for the software (Banker et al. 1998; Swanson and Beath 1989). Second, functionality is one of the main evaluation criteria for DBMS (Wittig and Inkinen 2004), which further gives good reason for using the number of functions/features as PRV's decision variable in the software design stage.

Stage 2: Software Pricing.

PRV chooses price p for the PRS developed in stage 1. Each consumer acquires one and only one copy of the software from either the OS community or the PRV (software market is fully covered). D denotes the demand for PRS and is determined by Q, q, p and the expected value of PRS service and OSS service (denoted by A and B respectively). The rest of the market adopts OSS. Demand for OSS is denoted as D^o .

Stage 3: Service Pricing⁶.

³ The literature on vertical differentiation is closely related to the literature on how to describe the differentiation between goods. This question has been addressed by Hotelling (1929), Chamberlin (1951, 1962), and Lancaster (1966). According to Tirole (1988) vertical differentiation is applied to a situation where "all consumers agree over the most preferred mix of characteristics" of goods. A widely-adopted approach to model vertical differentiation can be found in Tirole (1988).

⁴ $C(q)$ includes the cost to design and develop q features and to ensure the quality of the software if PRV is an entrant. If PRV is an incumbent, q represents the overall number of features bundled in the software; $C(q)$ includes the cost to design and develop the incremental features, the cost to ensure the quality of the software, including compatibility of new features to the system, maintainability, extensibility, robustness, reliability of the software, after including the incremental features.

⁵ Poor software design may also incur high demand for support service. However, this may not be of major concern in the case of MySQL database and Oracle database. MySQL uses a systematic approach to manage the process of software design and development and its database has an easy-to-use interface (Wittig and Inkinen 2004). In general, OSS has to reach mature stage to be able to compete with an existing PRS. Literature has shown that OSS that reached mature stage is quite comparable with corresponding PRS in terms of quality in software design (Wheeler 2005).

⁶ Some PRVs offer a bundle of software license and support contract. Oracle offers separate prices for license and support. I follow the practice of Oracle and separate the pricing decision for license and support into two stages. The result remains the same if I combine stage 2 with stage 3. The result will change if the price for a bundle with both license and support is different from the sum of the prices if the two are sold separately. Since bundling is not the focus of this paper, I separate the pricing for license and support for simplicity.

As discussed in previous Section, software post-sale service provides additional value to software users. In this stage, PRV chooses service price p_s for the PRS and a service provider specializing in OSS determines the service price p_s^o for OSS. I normalize service cost per customer to zero, so that service price is the profit margin. PRS users purchase service from PRV and OSS users purchase service from OSS service provider if their net surplus of purchasing the service is positive. For the same type of buyers (defined in the next paragraph), I assume service is of higher value to buyers if the software purchased in stage 2 is more complex, that is, the software is bundled with more features/functions. This is reasonable because if software is simple and easy to use, users do not have much need for technical support; while if software is complicated or users lack experience in using it, technical support becomes much more important to keep the software running and to update the software to topnotch condition. Following this logic, I denote the gross value of service for PRS and OSS by $V(q)$ and $V(Q)$ correspondingly where $V'(\cdot) > 0$.

Consumer Preference.

Consumers are uniformly distributed in a unit square where the two axes represent their taste for software features and service respectively⁷. For example, consumers located at (1, 1) have the highest valuation for software features and service; consumers located at (1, 0) have the highest valuation for software features but lowest valuation for service. Consumers know their type about software design at the beginning of stage 2, but get to know their type about service only after they consume the software, i.e., at the end of stage 2. This is because consumers' type for service is closely related with consumers' experience with the software. If the software runs smoothly, they may find less value of the service. If they have difficulty with the software, they may have greater need for the service. As a result, when consumers select software in stage 2, they use the expected surplus they anticipate to derive from the service as part of the input.

The Equilibrium

I start with the last stage to solve the model.

Stage 3. Consumers' utility is $\theta'V(q) - p_s$ if they purchase PRS service and $\theta'V(Q) - p_s^o$ if they purchase OSS service. θ' represents consumers' type about service and is uniformly distributed over the interval [0,1]. Demand for service

is $D_s = (1 - \frac{p_s}{V(q)})D$, $D_s^o = (1 - \frac{p_s^o}{V(Q)})D^o$ for PRS service and OSS service correspondingly. PRV's problem at this stage is to maximize his service profit determined by $p_s D_s$. OSS service provider maximizes his profit determined by $p_s^o D_s^o$.

Proposition 1. The equilibrium service price and the consequential service market coverage satisfy

$$p_s^* = \frac{V(q)}{2}, D_s^* = \frac{D}{2}, p_s^{o*} = \frac{V(Q)}{2}, D_s^{o*} = \frac{D^o}{2}$$

for PRS and OSS correspondingly.

Clearly, service providers for PRS and OSS are monopoly in their individual service market; hence each of them covers half of the service market and prices service at half of the service value to obtain monopolistic service profit.

Stage 2. Incorporating the above results into stage 2, I derive consumers' expected net gain of using PRS service and OSS service respectively.

⁷ This approach is commonly used in modeling consumer preferences in two dimensions. Consumers' tastes for software features and service are assumed to be independent from each other because their relationship could be dependent on consumers' specific requirements for the software, their technical capabilities and prior experience with similar products. Therefore, the relationship between them could be ambiguous. Customers with high valuation for software features may have low or high valuation for service; the same applies to customers with low valuation for software features.

$$A = \int_{\frac{p_s^*}{V(q)}}^1 (\theta'V(q) - p_s^*)d\theta' = \frac{V(q)}{8}, B = \int_{\frac{p_s^{o*}}{V(Q)}}^1 (\theta'V(Q) - p_s^{o*})d\theta' = \frac{V(Q)}{8}$$

Consumers' expected utility of choosing PRS is $\theta q - p + A$. θ represents consumers' type about software features and is uniformly distributed over the interval $[0, 1]$. $\theta q - p$ represents utility from consuming the software and A represents expected utility of adopting PRS service. Consumers' expected utility of adopting OSS is $\theta Q + B$. Let $q > Q$ ⁸, demand is

$$D = 1 - \frac{p - \frac{1}{8}(V(q) - V(Q))}{q - Q}, D^o = \frac{p - \frac{1}{8}(V(q) - V(Q))}{q - Q}$$

for PRS and OSS correspondingly. PRV's payoff at stage 2 is $\pi = \pi(\text{software}) + \pi(\text{service})$ where

$$\pi(\text{software}) = pD, \pi(\text{service}) = p_s^* D_s^* = \frac{V(q) D}{2} \frac{D}{2}.$$

PRV's problem at stage 2 is to

$$\max_p \pi = pD + \frac{V(q) D}{2} \frac{D}{2}$$

Proposition 2. The equilibrium PRS price and the corresponding software market share satisfy

$$p^* = \frac{1}{2}(q - Q - \frac{1}{8}V(q) - \frac{1}{8}V(Q)), D^* = \frac{1}{2} + \frac{\frac{3}{16}V(q) - \frac{1}{16}V(Q)}{q - Q}, D^{o*} = \frac{1}{2} - \frac{\frac{3}{16}V(q) - \frac{1}{16}V(Q)}{q - Q}$$

Proposition 2 shows that given the setup of this model, PRV gains larger market share than OSS in software market characterized by high demand for service as long as both the PRS and the OSS stay in the competition.

Stage 1. PRV's objective at stage 1 is to

$$\max_q \pi = p^* D^* + \frac{V(q) D^*}{2} \frac{D^*}{2} - C(q)$$

where p^* and D^* are determined as in proposition 2. To make the problem tractable, following Thatcher and Pingry (2004), I assume $C(q) = aq^2$ and let $V(q) = bq$ and $V(Q) = bQ$, where $a > 0$, and $b \geq 0$.

Proposition 3. If $b \in (b_1, b_2)$ where b_1 and b_2 are the roots of a quadratic function $(8+3b)^2 - 64abQ - 512aQ = 0$, PRV's payoff function is quasi-concave. The equilibrium number of features bundled in PRS satisfies

$$\frac{1}{2} \left(\frac{3b}{16(q^* - Q)} - \frac{b(3q^* - Q)}{16(q^* - Q)^2} \right) (q^* - Q - \frac{bQ}{8} + \frac{3bq^*}{8}) + \frac{1}{2} \left(\frac{1}{2} + \frac{b(3q^* - Q)}{16(q^* - Q)} \right) \left(1 + \frac{3b}{8} \right) - 2aq^* = 0$$

The sufficient condition for $q^* > Q$ is $Q < 1/(8a)$. Substituting q^* into the expressions of p^* and D^* specified in proposition 2 and into the expressions of p_s^* and p_s^{o*} specified in proposition 1, I derive the equilibrium PRS price and market share, and service prices for PRS and OSS.

Findings and Managerial Implications

⁸ PRVs have initial market power in majority of software markets (e.g., Windows and Linux; Oracle DBMS and MySQL DBMS), hence their PRS has higher market share and more software features than OSS initially. Following this market situation, I assume $Q < q$. Besides, this assumption is consistent with the situation in the DBMS market. MySQL's important target market is unsophisticated users who do not need all the features of Oracle database and MySQL provides a basic, but good-enough product (Wittig and Inkinen 2004). If $q \leq Q$, PRV will have zero demand. The condition will be checked later.

Comparative analysis of the equilibrium is presented in a two-by-two framework in Table 2. Table 2 reports how the impact of OSS differs on software market characterized by low demand for service versus on software market characterized by high demand for service. The findings are summarized in the following propositions.

Table 2: Comparative Analysis

	$Q = 0$ (No OSS)	$Q > 0$ (With OSS)
$b = 0$, software market with low demand for service	$q^* = \frac{1}{8a}, p^* = \frac{q^*}{2} = \frac{1}{16a}, D^* = \frac{1}{2}$	$q^* = \frac{1}{8a}, p^* = \frac{q^* - Q}{2} = \frac{1}{16a} - \frac{Q}{2}, D^* = \frac{1}{2}$
$b > 0$, software market with high demand for service	$q^* = \frac{1}{2a} \left(\frac{1}{2} + \frac{3b}{16} \right)^2$ $p^* = \frac{q^*}{2} - \frac{bq^*}{16}, D^* = \frac{1}{2} + \frac{3b}{16}$	q^* as in proposition 3 $p^* = \frac{q^*}{2} - \frac{bq^*}{16} - \frac{Q}{2} - \frac{bQ}{16}, D^* = \frac{1}{2} + \frac{(3q^* - Q)b}{16(q^* - Q)}$

Proposition 4. In software market characterized by low demand for service, when $Q < 1/8a$,

- OSS does not have an effect on the optimal number of features developed in competing PRS. The optimal level of features in PRS remains at the threshold value $1/8a$;
- PRS is priced lower after the entry of competing OSS;
- PRS remains the same market share after the entry of competing OSS.

The results reported in Proposition 4 are consistent with findings in prior literature that examines software market with low demand for service. For example, Bitzer and Schroder (2007) built a simple economic model to examine the impact of increased competition due to OSS entry on innovation activity of both incumbents and entrants. They found that entry of OSS promotes innovation, that is, enterprises choose to increase the technological levels after the emergence of OSS. Bitzer and Schroder (2006) empirically found support for the proposition in Bitzer and Schroder (2007). Note that although part (a) in Proposition 4 does not show an increase in PRV's investment in software feature after the entry of OSS, it does not conflict with the findings in prior research. This is because the investment decision in my model (Stage 1) refers to service-demand-improving investment in software feature. PRV may invest in other innovation activities in response to the entry of OSS, which may lead to an increase in its overall technological level.

Proposition 5⁹. In software market characterized by high demand for service, when Q is less than a threshold $\underline{q} \in (0, \frac{1}{8a})$,

$$Q < q^* < \frac{1}{2a} \left(\frac{1}{2} + \frac{3b}{16} \right)^2$$

- The equilibrium number of features in PRS satisfies where the upper bound is the equilibrium number of features in PRS when there is no OSS; PRV lowers the equilibrium number of features as OSS includes more features. When Q exceeds the threshold \underline{q} but is less than $1/8a$, the equilibrium number of features in PRS is at the highest level \bar{q} . When Q exceeds $1/8a$, PRV is driven out of the market.
- PRS is priced lower, but with larger market share when there is OSS than when there is no OSS.

Part (a) of proposition 5 implies a *magnetic phenomenon* in the design of PRS with high demand for service that is not observed in the design of software with low demand for service. Figure 1 illustrates the magnetic phenomenon. When the number of features in OSS is below a threshold value, the optimal number of features in PRS with high demand for service is higher than the number of features in the competing OSS; the higher the number of features in the OSS, the lower the number of features in the PRS. This situation resembles one in which

⁹ Proof of proposition 5 is based on the right continuity property of PRV's payoff function at $Q=0$ and by taking derivative of first and second order condition of the payoff function at optimal q with respect to Q . Complete proofs of all propositions are available upon request.

the position of RV is *attracted* to that of OSS. If the position of OSS exceeds the threshold, the design of PRS is placed at the maximal distance away from the position of OSS as if the two *repelled* one another, that is, $q = q^*$.

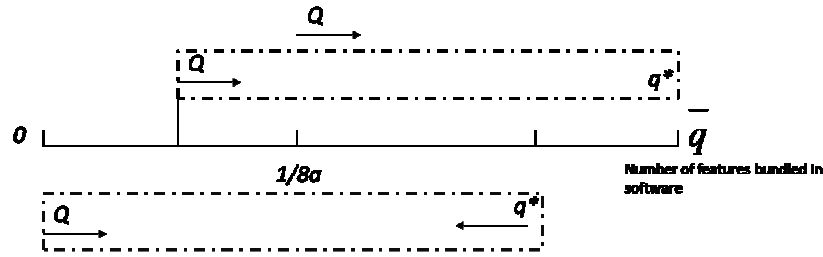


Figure 1: A Magnetic Phenomenon in the Design of PRS with High Demand for Service

This phenomenon can be explained by the leverage of market share effect and strategic effect discussed in the literature of product differentiation (Tirole 1988). According to Tirole (1988), strategic effect and market share effect represent competing firms’ product choice. The strategic effect dominates the market share effect when firms target their products at a maximal difference to avert intense competition; the market share effect dominates the strategic effect when firms target their products more closely to obtain a larger market share. It has been shown that in general, the strategic effect dominates the market share effect; hence firms tend to adopt the strategy of maximal difference. My findings suggest that in software market characterized by high demand for service, PRV’s software design strategy depends on the features in competing OSS: there is a threshold in the number of features in OSS, below which the market share effect dominates, and above which the strategic effect dominates. However, in software markets with low demand for service, the strategic effect always dominates the market share effect.

The intuition of PRV’s choice between market share effect and strategic effect is straightforward. PRVs have three options when determining number of features in response to OS entrant: decrease, increase or remain number of features in the software. For PRS with high demand for service, when the number of features in OSS is below a threshold, decreasing number of features in the PRS will decrease software price and customers’ valuation for service, but the loss of margin is compensated by larger demand for software as well as for service. So the market share effect dominates the strategic effect and PRV’s optimal response is to lower number of features. When OSS is positioned closer to PRS, the loss of margin in software and service exceeds the benefit of market share effect. It is optimal for PRV to play the high-end game, that is, only focus on customers that value large number of features. Hence the strategic effect dominates the market share effect. However, this strategy does not work for PRS with low demand for service. For this software category, the benefit of market share effect is not strong enough to cover the loss of margin in software license because the margin in service is zero. As a result, PRV’s optimal response is to remain the same number of features.

An important implication of Proposition 5 is that in software markets characterized by high demand for service, when OSS is not very sophisticated PRV could introduce more basic version of their software in response to the growth of OS counterparts. This implication is very much inline with industry practice. In the DBMS market, IBM, Oracle and Microsoft “have each lowered their database prices and created low-end bundles aimed at smaller organizations and partners.” (LaMonica 2005a) Oracle released a low-end, free but proprietary edition of its database in October 2005, supposedly as “a reaction to the growing competitive pressure from low-end open source databases.” (LaMonica 2005b) Andrew Mendelsohn, senior vice president of Oracle’s server technologies division, commented on this move: “There is definitely a market there (for low-end databases) and a demand. And we want them to be using Oracle and not MySQL or SQL Server Express.” (LaMonica 2005b)

The strategic handling of market share effect and strategic effect can also be used to explain PRVs OS strategies. Two common OS strategies employed by PRVs are: subsidize OS projects by having their employees working on these projects or making their PRS compatible with OSS (e.g., IBM, HP, SAP, Oracle, Siemens); and submit their proprietary products to the OS community (e.g., IBM submitted its database Cloudscape to Apache; BEA, Computer Associates and Sun also donated their PRS to OS community). Researchers in general believe these strategies provide PRVs with the strategic advantage to hurt rivals or the opportunity to profit through complementary products or services. My finding shows that market share may be additional benefit that drives PRV’s behavior. Both releasing codes and making PRS compatible with OSS help PRVs gain larger market share by

attracting users of OS products. For example, by making its DBMS compatible with Linux, Oracle effectively attracts Linux users to their products run on Linux.

The finding of PRV's pricing strategy (part b in Proposition 5) is also consistent with industry practice. For example, BEA Systems reported in its annual report of 2007 that "competitive pressures and open source availability of functionally competitive software could require us to reduce the price of our products and related services." Some analysts believe that MySQL gave Oracle's corporate customers more price bargaining power than in the past despite the lack of functionality of MySQL compared with Oracle database (Wittig and Inkinen 2004).

The results presented in Propositions 4 and 5 allow me to compare the impact of OSS on market structure of two categories of software markets: software with high service demand and software with low service demand. The finding implies that OSS will be stronger in terms of market share in markets with low demand for service than in market with high demand for service; PRV gains larger market share in software market characterized by high service demand than in market with low service demand. These hold true so long as OSS is not very advanced. The results are consistent with Economides and Katsamakos (2006a). They found that a vertically integrated proprietary system (proprietary platform and applications to the platform are provided by one firm) has a larger market share than OS system if there is a relatively large demand for the proprietary application, a large transition cost for the open source platform and a relatively small demand for application of the open source platform.

The intuition is that in software markets with high demand for service, PRV has the advantage of manipulating software design and optimizing pricing for both software and service. OS service provider however does not have as much influence on the design of OSS as the influence PRV has on its own software. This disadvantage of OS support provider leads to smaller market share of OSS in markets with high demand for service than in markets with low demand for service. To certain extent this resembles the case of an integrated system (proprietary system including software and service) versus a disintegrated system (OS system). Interestingly although MySQL CEO Marten Mickos believed that eventually Oracle, Microsoft, and IBM would realized that MySQL could be a threat to their database business, my model shows that if MySQL does not have full control on the development of MySQL database, and is not able to align its development objective with the objective of MySQL database OS developers, it will not dominate the market of DBMS.

Proposition 6¹⁰. PRS with high demand for service has more features and larger market coverage than PRS with low demand for service regardless of whether there is OSS.

Proposition 6 is intuitive. In software market with high demand for service, PRV can derive higher margin selling service by increasing the number of features. The superiority of this type of PRS in terms of features helps increase demand for corresponding service, which in turn improves the expected value consumers can derive from purchasing the software. Naturally, market coverage for PRS with high demand for service is higher.

Proposition 7¹¹.

- a. OSS hurts PRV's payoff in both types of software markets. When OSS is sufficiently advanced, PRV is driven out of the market.
- b. PRV of software with high demand for service has higher payoff than PRV of software with low demand for service regardless of whether there is OSS.

Proposition 7 is very intuitive. Part (a) implies that competing with OSS in the same market is ultimately a bad strategy for PRV. In the long run, PRV has to find a way to cooperate with OS community.

Conclusions

OSS has become prominent in many categories of software. I investigate how the rise of OSS influence software market structure and PRV's software design strategy. In particular, I examine how the impacts of OSS differ between software market characterized by high demand for service and by low demand for service. I found

¹⁰ Proof of proposition 6 is straightforward, hence is omitted.

¹¹ Proposition 7 can be proved by studying the direct effect of Q and b on PRV's optimal payoff.

that when there is competition between OSS and PRS, PRS is priced lower because of the entry of OSS. OSS have larger market share in markets with low demand for service than in markets with high demand for service. PRV's software design strategy in response to OSS differs between software with high demand for service and software with low demand for service: PRV's software design demonstrates a *magnetic* phenomenon for software with high demand for service, the purpose of which is to increase software market share, which further promotes higher revenue in service; this magnetic phenomenon is not found in software with low demand for service. An important implication of the findings is that in software markets characterized by high demand for service, when OSS is not very sophisticated PRV could introduce more basic version of their software in response to the growth of OS counterparts. This implication is very much inline with the practice in the market of DBMS.

The paper has limitations. Some characteristics of software, such as network effect and switching cost effect, are not considered in the model. Network effect and switching cost may prevent users from adopting OSS or switching to OSS even if the OSS is priced at zero. This explains to some extent why some PRS remains its dominance while competing OSS provides similar features and functionalities free of charge.

There are several directions of future research. The model could be evaluated using primary data source collected through survey or interview whereas in this paper it is evaluated mainly with secondary data. Another direction of future research is to conduct case analysis for other OSS and compare PRVs' strategic response to the competition of OSS in different software markets.

References:

- Abreu, B. F., Melo, W. L. "Evaluating the Impact of Object-Oriented Design on Software Quality", in Proceedings of the Third International *Conference on Software Metrics* Symposium, March 1996.
- Bank, D. "Open Source Database Poses Oracle Threat", in *Wall Street Journal*. (Eastern Edition). New York, July 9, 2003, pp. B1.
- Banker, R., Davis, G., Slaughter, S. "Software development practices, software complexity, and software maintenance performance: a field study," *Management Science* (44:4), 1998, pp. 433-450.
- Banker, R., Slaughter, S. "A field study of scale economies in software maintenance," *Management Science* (43:12), 1997, pp. 1709-1725.
- Bessen, J. "Open source software: free provision of complex public goods", in *The Economics of Open Source Software Development*, Bitzer, J, Schroder, P. (Ed.), 2006, pp. 57-81.
- Bitzer, J. "Commercial versus open source software: the role of product heterogeneity in competition," *Economic Systems* (28), 2004, pp. 369-381.
- Bitzer, J., Schroder, P. "The impact of entry and competition by open source software on innovation activity", in *The economics of open source software development*, Bitzer, J, Schroder, P. (Ed.), 2006, pp. 219-246.
- Briand, L.C., Morasca, S., Basili, V.C. "Defining and Validating Measures for Object-Based High-Level Design," *IEEE Transactions Software Engineering* (25:5), 1999, pp. 722-743.
- Chamberlin, E. 1951. "Monopolistic competition revisited." *Economica* 18:343-363.
- Chamberlin, E. 1962. "The theory of monopolistic competition." 8th edition. Cambridge, Mass.: Harvard University Press.
- Dahlander, L., Magnusson, M. "Business models and community relationships of open source software firms", in *The Economics of Open Source Software Development*, Bitzer, J, Schroder, P. (Ed.), 2006, pp. 111-130.
- Economides, N., Katsamakas, E. "Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry," *Management Science* (52:7), 2006a, pp. 1057-1071.
- Economides, N., Katsamakas, E. "Linux vs. Windows: a comparison of application and platform innovation incentives for open source and proprietary software platforms", in *The Economics of Open Source Software Development*, Bitzer, J, Schroder, P. (Ed.), 2006b, pp. 207-218.
- Gaudeul, A. "Competition between open-source and proprietary software: the LaTeX case study," Working paper, University of East Anglia – Norwich and ESRC Center for Competition Policy, 2005.
- Gaudeul, A. "Do open source developers respond to competition? The LaTeX case study," *Review of Network Economics* (6:2), 2007, pp. 239-263.
- Haruvy, E., Sethi, S., Zhou, J. "Open source development with a commercial complementary product or service," Working paper, University of Texas Dallas, 2005.
- Hotelling, H. 1929. "Stability in competition." *Economic Journal* 39:41-57.
- Kerstetter, J. "Will IBM's Gift Keep on Giving?" *Business Week Online*, 8/4/2004.
- Lacy, S. "Open Source: Now It's an Ecosystem." *Business Week Online*, 10/3/2005.
- LaMonica, M. "Oracle buys open-source database firm." *News.com*, October 10, 2005a.
- LaMonica, M. "Oracle to offer free database." *News.com*, October 28, 2005b.
- Lancaster, K. 1966. "A new approach to consumer theory." *Journal of Political Economy* 74: 132-157.
- Lawton, C. "Business Technology: Linux Shoots for Big League of Servers; Low-Cost Operating System Pushes Into Microsoft, Sun Strongholds," *Wall Street Journal*. (Eastern Edition). New York, N.Y. June 19, 2007, pp. B.5.
- Lerner, J., Tirole, J. "Some simple economics of open source," *Journal of Industrial Economics* (L:2), 2002, pp. 197-234.
- Lerner, J., Tirole, J. "The economics of technology sharing: open source and beyond," *Journal of Economic Perspectives* (19:2), 2005, pp. 99-120.
- Li, W., Henry, S., Kafura, D., Schulman, R. "Measuring Object-Oriented Design," *Journal of Object-Oriented Programming* (8:4), 1995, pp. 48-55.
- Mustonen, M. "When does a firm support substitute open source programming?" *Journal of Economics and Management Strategy*(14:1), 2005, pp. 121-139.
- Net Applications. "Top Browser Market Share for February, 2008," Retrieved on 2008-03-31.
- Oracle annual report 2007: http://www.oracle.com/corporate/investor_relations/earnings/10k-2007.pdf.
- Rossi, C., Bonaccorsi, A. "Intrinsic motivations and profit-oriented firms in open source software: do firms practice what they preach?" *The Economics of Open Source Software Development*, 2006, pp. 83-107.

- Schmidt, K., Schnitzer, M. "Public subsidies for open source? Some economic policy issues of the software market," *Harvard J of Law and Technology* (16:2), 2003, pp. 474-502.
- Sen, R. "A strategic analysis of competition between open source and proprietary software," *Journal of Management Information Systems* (24:1), summer 2007, pp. 233-257.
- Stone, A. "For Novell, Linux Means New Life," *Business Week Online*, 3/31/2004.
- Sun Microsystems annual report 2007: http://www.sun.com/aboutsun/investor/annual_reports/sun_ar07.pdf.
- Swanson, E. B., Beath, C. M. "Maintaining information systems in organizations", John Wiley and Sons, New York, 1989.
- Tam, P. W. "H-P Increases Its Support For 'Open Source' Software," *Wall Street Journal*. (Eastern Edition). New York, N.Y. June 1, 2004, pp. B.4.
- Tan, Y. and Mookerjee, V. "Comparing uniform and flexible policies for software maintenance and replacement," *IEEE Transactions on Software Engineering* (31:3), 2005, pp. 238-255.
- Thatcher, M., Pingry, D. "An Economic Model of Product Quality and IT Value," *Information Systems Research* (15:3), 2004, pp. 268-287.
- Tirole, J. *The theory of industrial organization*, MIT Press, Cambridge, MA, 1988.
- Vara, V. "Oracle Takes Aim At Linux Support Sold by Red Hat." *Wall Street Journal*. (Eastern Edition). New York, N.Y., October 26, 2006. pp. B.4.
- Wheeler, D. "Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!" November 2005 (available online at http://www.dwheeler.com/oss_fs_why.html).
- Wittig, C. "MySQL open source database in 2006 (B)." Stanford case SM-124 (B), 2006, Graduate School of Business.
- Wittig, C., Inkinen, S. "MySQL open source database in 2004." Stanford case SM-124, 2004, Graduate School of Business.
- Netcraft's Survey on web server market share at <http://news.netcraft.com/>

Appendix: Summary of Key Assumptions

1. Fixed cost of service in the model is assumed to be zero. However, the findings still hold for situations when this assumption does not hold. This is because fixed cost of service will be irrelevant in solving for the equilibrium in the third and second stage of the model for the reason that fixed cost of service is dependent on parameters determined or known in the first stage of the game, that is, q for PRV and Q for OS service provider. Therefore including fixed cost of service in the third stage of the model will be equivalent to including it in the first stage payoff function. Furthermore, since both fixed cost of service and fixed cost of software design are determined by software design, they can be combined. This explains why the findings still hold if the assumption of zero fixed cost of service is relaxed.
2. I assume old version of software is out of the market right after the release of new version. That is, multiple versions of the software do not coexist in the market. Although this assumption appears strong, it may not cause serious problem. If old version is present with the new version at the beginning of the second stage in the model, users of the old version basically have three options at the beginning of the second stage: they can stay with the old version, upgrade to the new version or switch to OSS. If majority of existing users of PRS choose to stay with the old version, this game will be the same as the previous one where users select the old version of the software when it is released for the first time. If only a small portion of current PRS users choose to not upgrade, the presence of old version after the release of new version will not have much impact on the findings. If some existing users choose to upgrade while others choose to stay with the old version, the problem becomes more complicated and is out of the scope of this paper.
3. I assume there is no cost of switching from PRS to OSS. Switching cost will deter users from switching software vendors. Hence, if switching cost is positive PRV will have larger market share in equilibrium.