Commercializing Open Source Software

Byung Cho Kim
bck@vt.edu
Pamplin College of Business
Virginia Tech
Blacksburg, VA 24061

Tridas Mukhopadhyay
tridas@cmu.edu
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA 15213

Pei-yu Chen
pychen@temple.edu
Fox School of Business
Temple University
Philadelphia, PA 19122

1 Mailing Address: Byung Cho Kim, 1007 Pamplin Hall (0235), Department of Business Information Technology, Pamplin College of Business, Virginia Tech, Blacksburg, VA 24061; Phone: 540-231-4593.
2 Mailing Address: Tridas Mukhopadhyay, Posner Hall 385B, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213; Phone: 412-268-2307.
3 Mailing Address: Pei-yu Chen, 207E Speakman Hall, 1810 North 13th Street, Philadelphia, PA 19122; Phone: 215-204-8715.
Commercializing Open Source Software

Abstract

In this paper, we examine open source software (OSS) business models with a focus on pricing strategies. We investigate the optimal pricing strategies of software vendors, both proprietary and open source, under competition. Our model considers motivations for, and barriers to OSS adoption, which have been exclusively discussed in practice for the last decade. We characterize the conditions under which the commercial OSS model is viable when potential OSS customers incur cost to switch from an established proprietary software vendor to commercial OSS. Our findings give strategic insights and pricing guidelines to software vendors who consider commercializing their OSS products.

Key words: Open Source Software; Software Pricing; Dual-Licensing Model; Switching Cost
1. Introduction

Open Source Software (OSS) is software for which the source code is available to the public, enabling anyone to copy, modify, and redistribute it (Varian and Shapiro 2003). Proprietary software, by contrast, is software that is distributed under a proprietary license agreement, usually for a fee. Open source was a service mark of the Open Source Initiative (OSI), a non-profit organization that continues to provide an official ‘Open Source Definition’. According to the OSI definition, OSS is software whose source code can be freely modified and redistributed. The redistribution rights do not preclude a company selling such software for profit. Recently, OSS is gaining enormous momentum. According to a recent article in the New York Times, OSS is used heavily by big companies such as IBM, Oracle, Google, Apple, and even Microsoft and thus, it becomes a weapon in corporate warfare (Lohr 2010). A survey by Forrester Research shows that a significant percentage of enterprises have adopted or will adopt the OSS within the next 12 months in various business functions including database (66 percent), web server and networking (63 percent), and security tools (56 percent) (Asay 2009).

An important question is then what motivates enterprises to employ OSS? The Free/Libre and OSS (FLOSS) Survey found that firms are employing OSS on the grounds of cost savings, flexibility and independence from giant software vendors (Wichmann 2002). Firms consider the flexibility of OSS to be important since they believe that vendors of proprietary software routinely downplay the customizability of OSS, arguing customers are not interested in extending software functions themselves. A survey by Actuate confirms that the benefits from OSS adoption include cost savings (55%), vendor independence (49.3%) and flexibility (47.1%) while the main barriers are availability of long-term support (58.2%) and long-term maintenance (44.7%) (McCarthy 2006). Thus, firms that do not have a capable IT management team tend to be reluctant in adopting OSS.

Lately, a new movement in the OSS industry becomes phenomenal. As OSS gains popularity and the market share approaches critical mass on account of its distribution under free license, vendors of such

---

4 The Open Source Definition. http://www.opensource.org/docs/osd
OSS seek to capitalize on the publicity and popularity of their OSS. Two models were proven to be viable for commercializing OSS: dual-licensing and support model. Under the dual-licensing model, the OSS vendor offers the very same software under two different licenses. Open source license allows the licensees to modify, distribute, and use the software for free, but it requires the release of any modifications under the same open source license. The proprietary license permits using the software under standard proprietary terms. Examples include MySQL AB’s database, Oracle Corporation’s Bergeley DB, Qt Software’s Qt development toolkit and Asterisk that is an OSS telecommunications software suite from Digium. Another approach is the support model, under which the vendor sells support service for the OSS which is distributed at no charge. Red Hat and JBoss have been successfully commercializing OSS under the support model. There is a lively debate going on among practitioners about the viability of OSS business models (Moczar 2005, Vaughan-Nichols 2005).

While the number of OSS vendors successfully adopting the aforementioned business models is growing, OSS commercialization is still at an early stage. The viability of OSS business models is still questionable and clear pricing guideline is not yet available to the OSS vendors who consider commercializing their products. For the last decade, a significant number of OSS vendors have successfully built a large installed base through free distribution, leading some of them, if not all, to consider making profits from their OSS. Despite the great potential of OSS business models, little academic research has examined the issue of OSS commercialization. This paper examines the viability of the OSS business model focusing on the dual-licensing regime, under competition with an established proprietary software vendor. Our focus is not the OSS vendors who launch their OSS under the dual-licensing model outright, which is remote from reality, but the OSS vendors who have already reached a critical mass by building a sufficient installed based through distribution of its OSS under a free license and begin offering a commercial license. We identify the key success factors for the OSS business models and characterize the conditions under which OSS commercialization is viable. We only consider firm-level customers since most OSS is distributed freely to individual customers. We model the motivations for and the barriers to OSS adoption and analyze their impact on the viability of the OSS business model.
Our model captures two sources of customer heterogeneity in their valuation of software: taste and technology savviness.

As a benchmark, we first examine a monopoly case in which we compare profits from two different software regimes: proprietary and open source. We investigate how customers’ technology savviness and the OSS-generic benefits and drawbacks affect the monopolistic software vendor’s incentive to choose an OSS regime over a proprietary one. We then extend our model to a realistic two-period setting under which an OSS vendor with a dual-licensing model enters the market which is already dominated by an established proprietary software vendor. Inspired by a real-world software market (e.g., the database management software market where MySQL competes with Microsoft SQL Server), we characterize the conditions under which the OSS dual-licensing model is viable. We examine the impact of switching cost that customers of proprietary software may incur when they switch to OSS on the viability of the commercial OSS licensing model. We also investigate how the optimal pricing strategies of both OSS and proprietary software vendors are affected by the OSS-specific factors, such as quality deficiency of OSS, support costs to high- and low-type customers in different scale, and the scope of switching cost.

Our paper contributes to the literature in the following ways. Firstly, while little academic research examines the issue of OSS commercialization, this study investigate the pricing issues of the commercial OSS licensing model through an economic lens, which has not yet been explored much despite the growing interest in the economics of OSS. We identify the factors that affect the pricing decisions of OSS vendors. Secondly, our model captures the motivations for and the barriers to OSS adoption. We characterize the conditions under which the OSS business models are viable. By doing so, we provide theoretical foundations of commercializing open source software. Finally, our findings can give pricing guidelines for OSS vendors who consider commercializing OSS.

The rest of this article is organized as follows. We discuss the related literature in Section 2. We present our model in Section 3 and examine the benchmark monopoly case in Section 4. In Section 5, we enrich our model and study the impact of switching cost on the viability of the OSS business model under competition. Section 6 concludes the paper.
2. Literature Review

This paper is grounded on two streams of research: (1) information goods pricing, and (2) economics of OSS. Pricing strategies for information goods or IT-enabled pricing strategies have been widely examined by researchers in the domain of Information Systems. Dewan and Mendelson (1990) examine the optimal pricing policy and capacity investment strategy in the context of ASP in the presence of nonlinear delay costs. Other popular topics include versioning (Bhargava and Choudhary 2001; Sundararajan 2004), bundling (Bakos and Brynjolfsson 1999; Hitt and Chen 2005), price discrimination (Choudhary et al. 2005; Dewan et al. 2003), and price dispersion (Clemons et al. 2002). Chen and Png (2003) examine information goods pricing in the context of digital rights management.

Prior literature on OSS focuses heavily on understanding the motivations of individual developers to participate in and contribute to OSS projects (Shah 2006). von Hippel and von Krogh (2003) consider two models of innovation, private and collective, and argue that contributors to OSS projects get intangible private benefits which are not present for free riders, such as personal satisfaction and learning. Franke and von Hippel (2003) examine the motivations of the Apache project participants and find that users’ desire to satisfy their own needs gives incentive for participation. Raymond (1999) suggests that the reputation and the status motivate developers’ participation while Ghosh (1998) argues that enjoyment and creativity matter. Economists argue that existing economic theory can explain OSS project participation (Learner and Tirole 2001, 2002). Hann et al. (2006) argue that participation in OSS projects is driven by career concerns, learning, and reputation. Roberts et al. (2006) develop a theoretical model to examine the system of interrelationship between motivations, participation, and performance. Singh et al. (2007) examines the relationships among OSS developers in their social network.

Modeling competition between proprietary software and OSS is an emerging issue among researchers who study OSS from an economic perspective. Raghunathan et al. (2005) examine the quality debate in OSS by setting up an analytical model and show that OSS quality is not necessarily lower than proprietary software quality. Casadesus-Masanell and Ghemawat (2006) analyze a dynamic mixed
duopoly in which a for-profit proprietary software vendor interacts with an OSS vendor in the presence of demand-side learning effects. Economides and Katsamakas (2006) analyze the optimal two-sided pricing strategy of a platform firm and compare industry structures based on a proprietary platform such as Microsoft Windows with those based on an open source platform such as Linux. Some research discusses the legal issues around commercial OSS (Gomulkiewicz 2004, Välimäki 2003). Our study aims to bridge the gap between information goods pricing and OSS literature by examining the pricing issues around OSS.

3. Model

We analyze the optimal pricing decisions of software vendors and examine viability of the OSS business model in different scenarios. We consider the dual-licensing model for OSS in which the vendor provides the same software under two different licenses: open source license and proprietary license. Users who want to donate their source code to the open source community can license software under an open source license, namely the General Public License (GPL). Under this open source license, the licensees can freely modify, distribute, and use the software at no charge. On the other hand, any users who want to use the OSS for profit-seeking purposes must purchase a proprietary license. MySQL and Mozilla Firefox are the examples of OSS under the successful dual-licensing model. We analyze the optimal pricing of a proprietary license of OSS.

3.1 Customers

We only consider firm-level customers who are obligated to buy the proprietary license of OSS under the dual-licensing regime. We characterize customers by two dimensions constituting heterogeneity: software valuation and technical savviness. Different customers value the same software differently. For example, a firm that heavily uses information technology values the same software more than others since a significant portion of its core business depends on information systems powered by the software. \( v \) is a
taste parameter that captures customer heterogeneity in valuation of software. We assume that \( v \) is uniformly distributed on \([0, 1]\), leading to a linear demand curve. Regarding technical savviness, customers divide into two segments. A proportion \( \mu \) of customers are technologically savvy, and we call them \textit{high-type} customers. These customers have a capable in-house IT management team, which allows them to take advantage of the flexibility of OSS \((f_{OH})\) by customizing the source code. The high-type customers may adopt OSS not only because OSS has cost advantage but also it gives them flexibility. Customers in the remaining fraction \((1 - \mu)\) are \textit{low-type} customers. Cost advantage may be their only incentive for OSS adoption. High-type customers are defined as customers to whom benefit exceeds cost from customization of OSS while low-type customers are ones to whom cost exceeds benefit from customization. Thus, the net flexibility benefit is positive for high-type customers \((f_{OH} > 0)\) while it is zero for low-type customers \((f_{OL} = 0)\) since without technical savviness, no flexibility benefit can be expected. Proprietary software in our paper is closed-source proprietary software, thus, we do not consider the customization of it. Software maintenance and support are costly to all customers. Denote support cost with \( s_j \) where \( j = CH, CL, OH, OL \). We assume that low-type customers suffer more from support and maintenance of any software \((s_{CH} < s_{CL}, s_{OH} < s_{OL})\). To reflect reality, we also assume that support is more costly for OSS than for proprietary software to the same-type customer \((s_{CH} < s_{OH}, s_{CL} < s_{OL})\). We assume risk neutrality of customers.

3.2 Software Pricing Models

We examine the optimal prices for the software under two different pricing schemes: (1) proprietary and (2) OSS dual-licensing. Depending on whether the source code is open or closed, the very same software can lead to different levels of utility. We model such impact of the openness of the source code on the customer utility. In this section, we derive customer utility and vendor profit under each of the three software business models, based on which the equilibrium prices are obtained.
Proprietary Software

We label the high-type customers $H$ and the low-type customers $L$. The proportion of the high-type customers is $\mu$ and the remaining proportion, $1-\mu$ is the low-type segment. Since proprietary software comes as a package with closed source code, the customizability of it is minimal regardless of customer type. Low-type customers suffer from support and maintenance due to limited technical capability. $C$ in the subscript means proprietary software. High-type and low-type customers enjoy utilities, $u_{CH} = v + t - p_c - s_{CH}$ and $u_{CL} = v + t - p_c - s_{CL}$, respectively, where $t$ represents the value difference between proprietary software and OSS and $p_c$ is the price of the proprietary software. Positive (negative) $t$ implies that proprietary software offers higher (lower) value than OSS. Let $q_{CH}$ and $q_{CL}$ be the demand for the software under a proprietary regime from the high-type and the low-type segments, respectively. The profit for the OSS vendor then becomes $\pi_c = (\mu q_{CH} + (1-\mu)q_{CL})p_c$. We assume zero marginal cost, which is reasonable for information goods such as software.

OSS Dual-Licensing Model

$O$ in the subscript represents OSS dual-licensing model. When adopting OSS, only high-type customers enjoy the flexibility. $p_O$ denotes the price of the proprietary license for OSS. The net benefits for high-type and low-type customers become $u_{OH} = v - p_o + f_{OH} - s_{OH}$ and $u_{OL} = v - p_o - s_{OL}$, respectively. We assume that for high-type customers, net flexibility benefit always exceed support cost ($f_{OH} - s_{OH} > 0$). Let $q_{OH}$ and $q_{OL}$ be the demand for the OSS under the dual-licensing model from the high-type and the low-type segments, respectively. The profit for the OSS vendor then becomes $\pi_o = (\mu q_{OH} + (1-\mu)q_{OL})p_O$. Table 1 summarizes the notations used in this paper.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>Customer valuation of software</td>
</tr>
</tbody>
</table>
\[ s_{CH} \quad \text{High-type customer's support cost for proprietary software} \]

\[ s_{CL} \quad \text{Low-type customer's support cost for proprietary software} \]

\[ s_{OH} \quad \text{High-type customer's support cost for OSS} \]

\[ s_{OL} \quad \text{Low-type customer's support cost for OSS} \]

\[ f_{OH} \quad \text{High-type customer's net flexibility benefit from OSS} \]

(i.e., benefit from customization – customization cost)

\[ t \quad \text{Value difference between proprietary software and OSS} \]

\[ t > 0 \text{ means proprietary software gives higher value than OSS} \]

\[ t < 0 \text{ means OSS gives higher value than proprietary software} \]

Table 1. Notations

4. Benchmark: Monopoly Case

In this section, we examine the optimal price and the corresponding maximal profit of a monopolistic software vendor under each of the two pricing regimes: proprietary and OSS. We investigate whether a monopolistic software vendor has an incentive to adopt the OSS business model instead of the proprietary model. Monopoly is worthwhile to examine as a benchmark because a natural monopoly often occurs in software industry.

Proprietary software

Customers who get positive utility would buy the software, resulting in demands for the proprietary software from high-type and low-type customers to be \( q_{CH} = 1 - p_c + t - s_{CH} \) and \( q_{CL} = 1 - p_c + t - s_{CL} \), respectively. The profit for the monopolistic software vendor under the proprietary regime becomes

\[ \pi_c = (\mu q_{CH} + (1-\mu) q_{CL}) p_c = (1 - p_c + t - \mu s_{CH} - (1-\mu) s_{CL}) p_c. \]

Thus, the profit-maximizing price
becomes \( p^*_C = \frac{1}{2}(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}) \), at which, the maximum profit level becomes
\[ \pi^*_C = \frac{1}{4}(1 + t - \mu s_{CH} - (1 - \mu)s_{CL})^2. \]

**OSS Dual-Licensing Model**

Under the dual-licensing regime, demands for the OSS from the high-type and the low-type segments are
\[ q_{OH} = 1 - p_O + f_{OH} - s_{OH} \quad \text{and} \quad q_{OL} = 1 - p_O - s_{OL}, \]
respectively. The monopolistic software vendor gets profit as
\[ \pi_O = (\mu q_{OH} + (1 - \mu)q_{OL})p_O = (1 - p_O + \mu(f_{OH} - s_{OH}) - (1 - \mu)s_{OL})p_O. \]
Thus, the optimal price under the dual-licensing model becomes
\[ p^*_O = \frac{1}{2}(1 + \mu(f_{OH} - s_{OH}) - (1 - \mu)s_{OL}). \]
The maximum profit is then
\[ \pi^*_O = \frac{1}{4}(1 + \mu(f_{OH} - s_{OH}) - (1 - \mu)s_{OL})^2. \]
We compare the OSS pricing model with the proprietary model and identify the conditions under which one regime is more profitable for a monopolistic software vendor than the other. The following proposition summarizes the results from price and profit comparisons between an OSS pricing model and a proprietary model.

**Proposition 1:** A monopolistic software vendor charges higher price and makes higher profit with a proprietary model than with an OSS model \( (p^*_C > p^*_O, \pi^*_C > \pi^*_O) \), when the aggregate value is higher under the proprietary regime than the OSS regime \( (t - \mu s_{CH} - (1 - \mu)s_{CL} > \mu(f_{OH} - s_{OH}) - (1 - \mu)s_{OL}) \).

Otherwise \( (t - \mu s_{CH} - (1 - \mu)s_{CL} < \mu(f_{OH} - s_{OH}) - (1 - \mu)s_{OL}) \), the OSS regime leads to higher price and profit \( (p^*_C < p^*_O, \pi^*_C < \pi^*_O) \).

At the current stage of the software industry, proprietary business model is prevalent. However, given the increasing popularity of commercial OSS, a software vendor may want to consider opening the source code and adopting OSS pricing models before introducing its software to the market. Proposition 1 indicates that the portion of technically savvy customers and their flexibility benefit from the OSS are the key factor for the viability of the OSS dual-licensing model under monopoly. When the aggregate benefit that high-type customers enjoy from OSS outweighs aggregate support cost for low-type customers by
much, even the monopolistic software vendor may be better off with the open source regime than with the proprietary one. In the current state, it is hard to find an example for monopolistic commercial OSS. The findings imply that the current software market may not have sufficient high-type customers and/or not much flexibility benefit from OSS is realized by them given the significant support cost. It is also possible that the support cost from proprietary software is negligible compared to that from OSS, which makes the OSS dual-licensing model less attractive. The monopolistic software vendor may want to consider the OSS dual-licensing model only when it makes sure that many of the potential customers of its software are technologically savvy and that they will appreciate flexibility due to the openness of the source codes. Next, we examine a more realistic scenario under which an OSS vendor enters the market which is dominated by a proprietary software vendor.

5. Sequential Entry and Switching Cost

Commercializing OSS is an emerging but not yet popular concept to both academics and practitioners. Although some OSS vendors have been successfully employing the business models, the general applicability and viability of the OSS business models are still in question. Thus, it is worthwhile to identify the key success factors for the emerging OSS business models and characterize the conditions under which they are viable. In this section, we enrich our model to capture the key drivers for the success of the OSS business models, reflecting what is happening in reality.

We model the competition between an OSS vendor with the dual-licensing model and a proprietary software vendor in the application software market, inspired by the database management software competition between MySQL and Microsoft SQL server. The success of MySQL has been exclusively discussed by software experts. In the database management software market, MySQL has been successfully competing with Microsoft for the segment of small businesses while Oracle and IBM compete for large businesses (Burgelman et al. 2004). Motivated by the competition between MySQL and Microsoft, we extend our model to two-period competition setting in the presence of switching cost. In
reality, OSS vendors with monetary motivation are often late movers, that is, they are entrants to the market which is already dominated by a proprietary software vendor. Thus, investigating the impact of switching cost is important as in other studies of information goods. Researchers have exclusively studied the role of switching cost in the information goods market (e.g., Chen and Hitt 2002). We capture characteristics specific to the OSS such as quality deficiency of OSS, support costs to high- and low-type customers in different scale, and the scope of switching cost as in the benchmark.

5.1. General Description of a Two-Period Model

There are two decision periods. In the first period, there exists only a proprietary software vendor who sets a price $p_{C1}$ for a license which is good for period 1. There is a continuum of customers whose valuations of software, denoted by $v$, lie uniformly on $[0, 1]$ in each period. Among those, the proportion of high-type customers is $\mu$ while the rest $1-\mu$, are low-type. We assume forward-looking behavior of customers. That is, given $p_{C1}$, each customer decides whether to buy proprietary software in period 1 or wait to see the available options in period 2. In the second period, the OSS vendor enters the market and given period-1 price of proprietary software and the market base, both the OSS vendor and the proprietary software vendor set their prices for the period-2 license in a simultaneous manner. In period 2, a new set of customers with the same profile arrive. Thus, there are three different groups of customers: customers who live and buy in period 1 (group 1), customers who live in period 1 and defer their purchase decision to period 2 (group 2), and new customers in period 2 (group 3). In period 2, customers in group 1 can stay with proprietary software by purchasing period-2 license or switch to OSS while incurring switching cost. Customers in groups 2 and 3 choose proprietary software or OSS or neither.

We model the impact of the following parameters on the pricing decisions of software vendors. Customers who switch from proprietary software to OSS in period 2 incur switching cost denoted by $wx_{C1}$ where $x_{C1}$ is the number of customers who buy proprietary software in period 1 and $w$ is a scale parameter which models the scope of switching cost. This way of modeling switching cost is consistent
with the literature in that switching cost increases with the market size due to network externalities. Recall that we assume only high-type customers enjoy flexibility benefit $f_{OH}$ from OSS while all customers suffer support costs from any software in a different scale, denoted by $s_j$ where $j = CH, CL, OH, OL$. Our model also considers the value difference between proprietary software and OSS with a parameter $t$. Positive (negative) $t$ means that proprietary software gives higher (lower) value that OSS. The structure of the game is illustrated in Figure 1.

**Figure 1. Sequential Entry and Market Competition**

### 5.2. Strategic Choice of Forward-Looking Customers

In this section, we analyze the strategic choice of different groups of customers. Group 1 customers who purchase proprietary software in period 1 face two strategic choices in period 2: stay with proprietary software (CC) or switch to OSS (CO). The rest of period-1 customers who decide to wait and see the available options in period 2 (group 2) can buy proprietary software (XC), OSS (XO), or neither (XX). The new customers who arrive in the second period (group 3) have the same strategic choices as group 2 customers, that is, proprietary software (C), OSS (O), or nothing (X). We have three different groups of
customers with two segments in terms of their technological savviness, and they have different sets of strategic choices. Certainly, customer utility from not buying any software (i.e., XX or X) is zero. Table 2 presents customer utility from each of the different strategic choices.

<table>
<thead>
<tr>
<th>Strategic Choices</th>
<th>High-Type ($\mu$)</th>
<th>Low-Type ($1 - \mu$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC (Group 1)</td>
<td>$v - p_{C1} + t - s_{CH} + v - p_{C2} + t - s_{CH}$</td>
<td>$v - p_{C1} + t - s_{CL} + v - p_{C2} + t - s_{CL}$</td>
</tr>
<tr>
<td>CO (Group 1)</td>
<td>$v - p_{C1} + t - s_{CH} + v - p_{O} + f_{OH} - s_{OH} - w_{C1}$</td>
<td>$v - p_{C1} + t - s_{CH} + v - p_{O} - s_{OL} - w_{C1}$</td>
</tr>
<tr>
<td>XC (Group 2)</td>
<td>$v - p_{C2} + t - s_{CH}$</td>
<td>$v - p_{C2} + t - s_{CL}$</td>
</tr>
<tr>
<td>C (Group 3)</td>
<td>$v - p_{O} + f_{OH} - s_{OH}$</td>
<td>$v - p_{O} - s_{OL}$</td>
</tr>
</tbody>
</table>

Table 2. Customer Utility

Now we characterize the conditions under which a customer of each type in each group makes a particular software choice over the other. Consider a high-type customer in group 1. This customer will choose proprietary software in period 2 if $v - p_{C2} + t - s_{CH} > v - p_{O} + f_{OH} - s_{OH} - w_{C1}$. Note that high- and low-type customers with positive utility (i.e., $v - p_{C1} + t - s_{CH} > 0$ and $v - p_{C1} + t - s_{CL} > 0$, respectively) will buy the proprietary software in period 1 which leads to period-1 market base as $\mu = (1 - p_{C1} + t - s_{CH}) + (1 - \mu)(1 - p_{C1} + t - s_{CL}) = 1 - p_{C1} + t - \mu s_{CH} - (1 - \mu) s_{CL}$. Thus, the condition under which a high-type customer in group 1 chooses proprietary software can be rearranged as $wp_{C1} + p_{C2} < p_{O} + t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu) s_{CL})$. OSS will be selected by the high-type customer if $wp_{C1} + p_{C2} > p_{O} + t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu) s_{CL})$. Following the same logic, the condition under which a low-type customer in group 1 plays CC instead of CO is $wp_{C1} + p_{C2} < p_{O} + t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu) s_{CL})$. The low-type customer will choose strategy CO over CC when $wp_{C1} + p_{C2} > p_{O} + t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu) s_{CL})$. 
Since the utility for the same-type customers in group 2 and group 3 is the same as shown in Table 2, the conditions under which proprietary software is preferred to OSS are identical. A high-type customer in groups 2 or 3 will prefer proprietary software if $v - p_{C2} + t - s_{CH} > v - p_o + f_{OH} - s_{OH}$, that is, $p_{C2} < p_o + t - s_{CH} - f_{OH} + s_{OH}$. Otherwise, OSS will be chosen. A low-type customer in groups 2 or 3 will choose proprietary software over OSS when $p_{C2} < p_o + t - s_{CL} + s_{OL}$, while OSS will be selected if $p_{C2} > p_o + t - s_{CL} + s_{OL}$.

Recall that support is assumed to be more costly for OSS than proprietary software regardless of customer type (i.e., $s_{OH} > s_{CH}$ and $s_{OL} > s_{CL}$). Thus, $t - s_{CL} + s_{OL}$ is always positive, meaning that the proprietary software vendor can attract low-type customers in any group by charging period 2 price lower than $p_o + t - s_{CL} + s_{OL}$. Thus, low-type customers’ choice will always be proprietary software no matter to which group they belong. The pricing strategies of the software vendors and the high-type customers’ choices are not trivial. We consider the following three cases under each of which high-type customers behave differently:

**Case I: Better Proprietary software** ($t - s_{CH} - f_{OH} + s_{OH} > 0$)

**Case II: Better OSS, High Switching Cost** ($-w(1 + t - (1 - \mu)s_{CH} - (1 - \mu)s_{CL}) < t - s_{CH} - f_{OH} + s_{OH} < 0$)

**Case III: Better OSS, Low Switching Cost** ($t - s_{CH} - f_{OH} + s_{OH} < -w(1 + t - (1 - \mu)s_{CH} - (1 - \mu)s_{CL})$)

We further analyze each of the aforementioned three cases and present the outcome in the next section.

### 5.3. Software Prices and Customer Choice at Equilibrium

In this section, we analyze pricing strategies of proprietary and OSS vendors. Since boundary solutions are special cases, we are interested in (1) whether there exists a unique interior solution in each of the three cases and if so, (2) what the optimal prices are and (3) what their characteristics are. We then seek managerial implications of these.
Case I: Better Proprietary software \((t - s_{CH} - f_{OH} + s_{OH} > 0)\)

When the high-type customers enjoy high value from OSS, the proprietary software vendor’s optimal strategy is to set prices such that \(wp_{C1} + p_{C2} < p_o + t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL})\) and \(p_{C2} < p_o + t - s_{CH} - f_{OH} + s_{OH}\). By doing so, the proprietary software vendor can serve all high-type customers in all three groups. Since the OSS vendor knows the proprietary software vendor will set the prices which will prevent the OSS vendor from making any positive profit, its best response is not entering the market. Then the customers’ optimal strategies can be summarized as follows:

Group 1: High (CC) Low (CC), Group 2: High (XC) Low (XC), Group 3: High(C) Low (C)

The proprietary software vendor’s profit function becomes

\[
\pi^\text{Case I}_c = (1 - p_{C1} + t - \mu s_{CH} - (1 - \mu)s_{CL})p_{C1} + \mu(1 - p_{C2} + t - s_{CH})p_{C2} \\
+ (1 - \mu)(1 - p_{C2} + t - s_{CL})p_{C2} + (1 - p_{C2} + t - \mu s_{CH} - (1 - \mu)s_{CL})p_{C2} \\
= (1 - p_{C1} + t - \mu s_{CH} - (1 - \mu)s_{CL})p_{C1} + 2(1 - p_{C2} + t - \mu s_{CH} - (1 - \mu)s_{CL})p_{C2}.
\]

In period 2, the proprietary software’s problem is

\[
\text{Max } \pi^\text{Case I}_c \text{ s.t. } p_{C2} < t - s_{CH} - f_{OH} + s_{OH}.
\]

Given the optimal period 2 price \(p^*_c\), the proprietary software vendor solves the following problem in period 1:

\[
\text{Max } \pi^\text{Case I}_c(p^*_c) \text{ s.t. } wp_{C1} + p^*_c < t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}).
\]

A further equilibrium analysis leads to the following Proposition.

Proposition 2: When a high-type customer enjoys higher value from proprietary software than from OSS \((t - s_{CH} > f_{OH} - s_{OH})\), the proprietary software vendor plays a pricing strategy to prevent the OSS vendor from making positive profit.

Proposition 2 indicates that it would not be possible for an OSS vendor to commercialize its product if it does not give any additional benefit to technically savvy customers. It turns out that a market-dominating proprietary software vendor will play a pricing strategy to prevent an OSS vendor from
entering the market, which may explain what is happening in many of the real-world proprietary versus OSS competition cases. This result implies that providing considerable benefits to a significant number of technically savvy customers in the market must precede commercializing OSS. We then investigate whether optimal pricing guidelines can be characterized.

**Proposition 3:** There exists a unique interior solution in which the proprietary software vendor charges equal price for both periods, i.e., \( p_{c1}^* = p_{c2}^* = \frac{1 + t - \mu s_{CH} - (1 - \mu)s_{CL}}{2} \). Other possible sets of optimal prices are characterized as below:

\[
p_{c1}^*(\text{boundary}) = \frac{t - s_{CH} - f_{OH} + s_{OH} + \omega(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}) - p_{c2}^*}{\omega},
\]

where \( p_{c2}^*(\text{interior}) = \frac{1 + t - \mu s_{CH} - (1 - \mu)s_{CL}}{2} \) or \( p_{c2}^*(\text{boundary}) = t - s_{CH} - f_{OH} + s_{OH} \).

The results show that there exists a unique interior solution that is characterized as equal prices for both periods. Under the conditions for the interior solution to exist, charging price at the level of which, single period profit is maximized is optimal.

**Case II: Better OSS, High Switching Cost** \((-w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}) < t - s_{CH} - f_{OH} + s_{OH} < 0)\)

Given the moderate net value difference, the proprietary software vendor wants to offer prices such that \( wp_{c1} + p_{c2} < p_o + t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}) \) to ensure that high-type customers who buy proprietary software will stay with it in period 2. Since \( t - s_{CH} - f_{OH} + s_{OH} < 0 < t - s_{SL} + s_{OL} \), the low-type customers in group 1 will stay as well with such prices. Also, the proprietary software vendor wants to serve low-type customers in groups 2 and 3 by charging period-2 price such that \( p_{c2} < p_o + t - s_{CL} + s_{OL} \). On the other hand, the OSS vendor knows that it can attract high-type customers in groups 2 and 3 by charging the price such that \( p_o < p_{c2} - (t - s_{CH} - f_{OH} + s_{OH}) \). Then the customers’ strategic choices can be summarized as follows:

Group 1: High (CC) Low (CC), Group 2: High (XO) Low (XC), Group 3: High (O) Low (C)
The proprietary software vendor’s profit function can be written as

\[ \pi_{C}^{\text{Case II}} = (1 - p_{C1} + t - \mu s_{CH} - (1 - \mu) s_{CL}) p_{C1} + \mu (1 - p_{C1} + t - s_{CH})(1 - p_{C2} + t - s_{CH}) p_{C2} + (1 - \mu)(1 - p_{C1} + t - s_{CL}) p_{C2} + (1 - \mu)(1 - p_{C2} + t - s_{CL}) p_{C2} \]

In period 2, the proprietary software’s problem is

\[ \max_{p_{C2}} \pi_{C}^{\text{Case II}} \text{ s.t. } p_{C2} < t - s_{CL} + s_{OL}. \]

Given the optimal period 2 price \( p_{C2}^{*} \), the proprietary software vendor solves the following problem:

\[ \max_{p_{C1}} \pi_{C}^{\text{Case II}} \left( p_{C2}^{*} \right) \text{ s.t. } wp_{C1} + p_{C2}^{*} < t - s_{CH} - f_{OH} + s_{OH} + w(1 + t - \mu s_{CH} - (1 - \mu) s_{CL}). \]

Now, the OSS vendor’s profit function can be written as

\[ \pi_{O}^{\text{Case II}} = \mu(p_{C1} - t + s_{CH})(1 - p_{O} + f_{OH} - s_{OH}) p_{O} + \mu(1 - p_{O} + f_{OH} - s_{OH}) p_{O} \]

\[ = \mu(1 + p_{C1} - t + s_{CH})(1 - p_{O} + f_{OH} - s_{OH}) p_{O}. \]

The OSS vendor solves the following problem:

\[ \max_{p_{O}} \pi_{O}^{\text{Case II}} \text{ s.t. } p_{O} < -(t - s_{CH} - f_{OH} + s_{OH}). \]

Further equilibrium analysis leads to the following Proposition.

**Proposition 4:** When a high-type customer enjoys higher value from OSS than from proprietary software but switching cost is high, period 1 customers of proprietary software will not switch to OSS. Among all other customers in period 2, high-type customers will choose OSS while low-type customers will buy proprietary software.

When technically savvy customers derive higher utility from OSS than from proprietary software, the OSS dual-licensing model is viable. With high switching cost, i.e., customers of proprietary software would incur high cost to switch to OSS in the second period, they will stay with the proprietary software. However, the new high-type customers who arrive in the second period will choose OSS over proprietary software.
**Proposition 5:** Proprietary software’s period 1 price always has a boundary solution while period-2 price can be either. The detailed expressions are as follows:

\[
p_{C_1}^* (\text{boundary}) = \frac{t(1 + \omega) + \omega - f_{OH} - (1 + \mu \omega) s_{CH} - (1 - \mu) \omega s_{CL} + s_{OH} - p_{C_2}^*}{\omega},
\]

\[
p_{C_2}^* (\text{boundary}) = t - s_{CL} + s_{OL}.
\]

The optimal price for OSS is independent of proprietary software prices as follows:

\[
p_{O}^* (\text{interior}) = \frac{1}{2}(1 + f_{OH} - s_{OH}),
\]

\[
p_{O}^* (\text{boundary}) = -(t - s_{CH} - f_{OH} + s_{OH}).
\]

It is interesting that the unique solution for period-1 price is boundary. Given the high complexity due to customers’ forward-looking behavior, it turns out that interior solution does not exist. Since no existing customer of proprietary software will not switch to OSS given high level of cost, OSS software vendor’s optimal price level is not affected by the scope of switching cost. We further investigate the impact of switching cost on proprietary software vendor’s prices. Figure 2 illustrates how the scope of switching cost influences proprietary software vendor’s prices. It is reasonable that the proprietary software vendor charges more in period 1 as switching to OSS becomes more costly to the customers.

![Figure 2. Impact of Switching Cost on Proprietary software Prices](image-url)
Case III: Better OSS, Low Switching Cost \((t - s_{CH} - f_{OH} + s_{OH} < -w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}))\)

Group 1: High (CO) Low (CC), Group 2: High (XO) Low (XC), Group 3: High (O) Low (C)

The proprietary software vendor’s profit function can be written as

\[
\pi^\text{CaseIII}_C = (1 - p_{C1} + t - \mu s_{CH} - (1 - \mu)s_{CL})(1 - p_{C1} + t - s_{CL})(1 - p_{C2} + t - s_{CL})p_{C2}
+ (1 - \mu)(p_{C1} - t + s_{CL})(1 - p_{C2} + t - s_{CL})p_{C2} + (1 - \mu)(1 - p_{C2} + t - s_{CL})p_{C2}
= (1 - p_{C1} + t - \mu s_{CH} - (1 - \mu)s_{CL})p_{C1} + 2(1 - \mu)(1 - p_{C2} + t - s_{CL})p_{C2}.
\]

In period 2, the proprietary software’s problem is

\[
\max_{p_{C2}} \pi^\text{CaseIII}_C \quad \text{s.t.} \quad p_{C2} < t - s_{CL} + s_{OL}.
\]

Given the optimal period 2 price \((p_{C2}^*)\), the proprietary software vendor solves the following problem:

\[
\max_{p_{C1}} \pi^\text{CaseIII}_C (p_{C2}^*) \quad \text{s.t.} \quad wp_{C1} + p_{C2}^* < t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}).
\]

Note that \(1 - p_{C1} + t - \mu s_{CH} - (1 - \mu)s_{CL}\) is period 1 demand for the proprietary software which is always positive. Thus, \(p_{C2} < t - s_{CL} + s_{OL}\) is a sufficient condition for \(wp_{C1} + p_{C2}^* < t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL})\).

Now, the OSS vendor’s profit function can be written as

\[
\pi^\text{CaseIII}_O = \mu(1 - p_{C1} + t - s_{CH})(1 - p_{O} + f_{OH} - s_{OH} - \omega(1 - p_{C1} + t - s_{CH}))p_{O}
+ \mu(p_{C1} - t + s_{CH})(1 - p_{O} + f_{OH} - s_{OH})p_{O} + \mu(p_{C1} + t - s_{CH})p_{O}
= \mu(1 - p_{C1} + t - s_{CH})(1 - p_{O} + f_{OH} - s_{OH} - \omega(1 - p_{C1} + t - s_{CH}))p_{O}
+ \mu(1 - p_{C1} - t + s_{CH})(1 - p_{O} + f_{OH} - s_{OH})p_{O}.
\]

Note that \(p_{O} < -(t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}))\) is sufficient to \(p_{O} < -(t - s_{CH} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}))\). The OSS vendor solves the following problem:

\[
\max_{p_{O}} \pi^\text{CaseIII}_O \quad \text{s.t.} \quad p_{O} < -(t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL})).
\]

Further equilibrium analysis leads to the following Proposition.
Proposition 6: When a high-type customer enjoys higher value from OSS than from proprietary software and switching cost is low, all high-type customers will choose OSS while all low-type customers will buy proprietary software in period 2.

Proposition 6 shows that commercial OSS is viable enough to attract existing customers of proprietary software as long as it gives considerable value to the technically savvy customers while switching cost is not significant. This result may encourage OSS vendors who consider commercializing its OSS which might reach a critical mass after years of free distribution. The key success factors include customizability of the OSS, easiness of it, and the number of customers who are technically capable. Next, we examine how optimal prices change with the scope of switching cost.

Proposition 7: There exists a unique solution for period-1 price as
\[ p_{C1}^* = \frac{1}{2}(1 + t - \mu s_{CH} - (1 - \mu)s_{CL}) \] while optimal period-2 price can be
\[ p_{C2}^* = \begin{cases} \frac{1}{2}(1 + t - s_{SL}) & \text{interior} \\ -(t - s_{CL} + s_{OL} + w(1 + t - \mu s_{CH} - (1 - \mu)s_{CL})) & \text{boundary} \end{cases} \]

The OSS vendor’s optimal price is as follows:
\[ p_{O}^* = \begin{cases} \frac{3 + t + (1 + t)^2 w + wp_{C1}^* - f_{OH}(3 - t + p_{C1}^* + s_{CH}) - s_{CH}(1 + 2(1 + t)w - ws_{CH})}{2(3 - t + p_{C1}^* + s_{CH})} & \text{interior} \\ \frac{3s_{OH} - (t - s_{CH})s_{OH} - p_{C1}^*(1 + 2(1 + t)w - 2ws_{CH} - s_{OH})}{2(3 - t + p_{C1}^* + s_{CH})}, & \text{boundary} \end{cases} \]

As summarized in Proposition 6, when OSS allows a high-type customer to enjoy sufficient benefit and switching is not much costly, all high-type customers of proprietary software in period 1 will switch to OSS. Since the condition for period-2 pricing suffices the condition for period-1 pricing, the solution for period-1 price is always interior. Interestingly, optimal OSS price is influenced by switching cost parameter since in this high benefit/low switching cost region, all high-type customers of proprietary software will switch to OSS. Figure 3 illustrates such impact. It shows that OSS price decreases with
switching cost and can be set even higher than proprietary software’s period-2 price. The results imply that the commercial OSS model is convincing in this region.

![Impact of Switching Cost on OSS Price](image)

**Figure 3. Impact of Switching Cost on OSS Price**

6. Conclusion

Commercializing OSS is becoming an important and attractive issue among software vendors as the critical mass has been reached for certain OSS such as Linux. In reality, some OSS vendors (e.g., Red Hat and MySQL) have already employed either of the two available business models for OSS: the dual-licensing model and the support model. The generalizability of such successful commercialization of OSS has been widely discussed among practitioners. Nevertheless, the viability of the OSS business models is still in question and the key success factors have not been clearly identified by academic researchers. In this paper, we examine the OSS business models under various scenarios and characterize the conditions under which the OSS business models are viable.

As a benchmark, we examine the optimal pricing strategy under monopoly. Our result indicates that the total net benefit from OSS is a key driver of a successful dual-licensing model under monopoly.
Inspired by a real-world software markets, we enrich our model to examine the impact of switching cost on the viability of the OSS business models. We set up a two-period game and analyze the customers’ incentive to switch from the market-dominating proprietary software to the newly introduced OSS. We find that the dual-licensing model is profitable in the presence of large switching cost. When the switching cost is low, software vendors split the market in the second period in a way that the proprietary software serves low-type customers and the OSS vendor covers the high-type customer segment. Interestingly, we find that the OSS may dominate the market in the second period when the customers enjoy considerable benefit while not suffering much from switching from proprietary software to OSS.

Our paper contributes to the literature in the following ways. First, this paper examines the issue of pricing OSS through an economic lens. In spite of the growing interest in proprietary OSS among industry experts and jurists, little academic study has studied the pricing strategies under the OSS business models. We identify the factors that affect the viability of the pricing models for OSS and find the conditions under which each model can be successful. Second, our result can give pricing guidelines to OSS vendors, which is not clear in the current state. Finally, we model the motivations for and the barriers to open source adoption, which provides a better picture of the OSS market. Considering such factors based on survey statistics may allow us to better understand the issue of OSS.

Although our findings are significant, this study can be improved in several ways. Firstly, modeling other strategic motivations of OSS vendors than pricing may bring insights to the results. For example, open source has been viewed as a marketing strategy by which the software vendor is able to build a customer base for its new software in a short amount of time. While this paper only considers firm-level customers and focuses on pricing strategies of OSS vendors, it would be an interesting direction to examine the marketing aspects of OSS with individual customers, for example, market expansion due to free distribution of OSS. Secondly, our model captures network externality in the switching cost function which increases with the network size. Isolating network externality and examining its impact may help to understand the interplay between the network size of users of a free version of the OSS and the profitability of a commercial version.
References


