Selling or Subscribing Software under Quality Uncertainty and Network Externality Effect

Jie Zhang
University of Texas at Arlington
jiezh@uta.edu

Abraham Seidmann
University of Rochester
avi.seidmann@simon.rochester.edu

Abstract

We examine the optimal way for a software vendor to license software: perpetual license at a posted price, subscription contract that subscribers receive automatic updates for periodic payment, or a hybrid approach that involves both. By addressing such specific issues in the software market as network effects, quality uncertainty, upgrade compatibility, and the vendor’s ability to commit to future prices in a dynamic environment, we demonstrate how a software vendor can manage the trade-offs of perpetual licensing and subscription to optimize profit, as well as the corresponding welfare effect on consumers. Though the subscription model helps the vendor lock in consumers so as to increase profit when there is a great uncertainty associated with the next version software, it destroys the path dependence in creating network externalities.

Therefore, when the network effect is sufficiently large, it is more profitable for a software vendor to provide both perpetual licensing and subscription.

1. Introduction

Software products have been traditionally sold as a property via a perpetual ownership licensing model. Customers acquire the permanent right to use and own the version of the software by paying upfront. The software publishers have adjusted their licensing models from the traditional perpetual licensing in favor of software subscription models. They deliver application software and services (software maintenance, upgrades, staff training, etc.) over the Web on a lease or subscription basis. For example, Microsoft bundled its Enterprise Subscription Agreement for business users with Software Assurance at an annual rate on a three-year term; vendors like Sun, Oracle, SAS Institute, Computer Associates and BMC also offered their own subscription licenses on their major products. Despite the various forms of licensing models, the vendors all aim at generating perpetual revenue streams by transforming software into a subscription-based service. Industry practitioners claim that the subscription model can lower the cost of ownership and grant users access to up-to-date software at a predictable cost without a large up-front investment. They therefore even claimed, “traditional software is already dead” ([25]).

International Data Corp. (IDC) has also predicted that the software industry will experience “a transition to subscription licensing” ([24]).

Software can be used for a period of time without replacement, though its value may depreciate. In this sense, it is a kind of a durable good. Yet software as a commodity has some special characteristics that differentiate it from other durable goods: (i), In general, software cannot be resold or appropriated due to license restrictions, so a secondhand market like that for used cars does not exist. Therefore we can ignore the resell problem in selling other durable goods; (ii), the production of software which is an information good has cost side economy of scale — it is costly to create the first copy but has negligible marginal production cost for the rest copies; (iii), with the development of information technology, it is relatively easy to improve the value of installed software through upgrades; (iv), the use of software has a strong network externality. That is, the value of using any particular software increases with the number of its adopters. This network externality therefore is like the “chicken-and-egg” problem ([15]). As a result, we cannot apply the established theories about selling or leasing durable goods directly to the licensing models of software.

This paper addresses the above issues by studying the optimal licensing policy of a monopolist software vendor who provides packaged off-the-shelf products through a perpetual licensing or subscription model exclusively, or adopting a mixed strategy of both. In order to provide strategic suggestions to software vendors as well as recommendations to users, we address the following questions by incorporating the special characteristics of software products:

(1) What are the benefits and costs of perpetual and subscription licensing models for a vendor?
(2) Trading off those benefits and costs, what is the optimal way for vendors to license software?
Should they use subscription licensing to completely replace the traditional perpetual licensing?
(3) How are consumers’ purchasing behaviors influenced by different licensing models?

By addressing the specific issues related to licensing models such as compatibility, network externalities, quality uncertainty, and commitment, our paper fills a gap in the literature by examining the impact of quality uncertainty and network externalities on a monopoly seller’s licensing models. With our more generalized findings, our paper extends the previous results of selling and leasing durable goods ([7], [4], [8], [16]) to a broader scope, and it provides key insights to software vendors as well.

Building on prior literature ([1], [5], [6], [10], [11], etc.) that considered either one or two aspects of issues
Consumers are heterogeneous in their quality preference of the product, and support $\theta \in [0,1]$. Consumers are homogeneous in their ability to improve product quality. He offers Version 1 of the software vendor’s optimal licensing policy, taking into account users’ responses. We provide vendors with important insights on licensing policies and improve our understanding of the implications of licensing model selection on consumer surplus and social welfare.

2. Model

We model the intertemporal consumer behavior and the monopolist software vendor’s strategic licensing policy with a two-period model as in [11] and [10]. Following such literature as [19], [21], and [10], we assume that a software user’s net utility of consuming the software product is

$$U(q, p; \theta) = \theta q + e x - p,$$  \hspace{1cm} (1)

where $\theta$ is the intensity of the quality preference, $q$ represents its quality, $e$ measures the intensity of the network externality effect, $x$ is the mass of the adopters of the product, and $p$ is the price paid for the software. Consumers are heterogeneous in their quality preference $\theta$, which is assumed to be uniformly distributed on the support $[0,1]$. Consumers are homogeneous in their sensitivity to network externality $e$. Software “quality” ($q$) includes such dimensions as features, speed, functionalities, user interface, ease of learning, and other characteristics that affect users’ valuation of the product. Potential buyers can assess the quality of existing software, yet the quality of the future (upgrade) version is unknown. Users who subscribe the software with periodic upgrades are exposed to the risk of buying into future upgrades with uncertain quality, while those who purchase the software can choose to upgrade or not in future period when the quality of the new version is realized.

The monopolist vendor continuously invests in R&D to improve product quality. He offers Version 1 of the software in period 1 and an improved version—Version 2—in period 2. When a consumer makes the adoption decision in period 1, the quality of Version 1 $q_1$ is already realized, while the quality of Version 2 $q_2$ is uncertain due to risks in R&D and market condition changes during period 1. Until $q_2$ is realized at the beginning of period 2, its distribution is public knowledge: with probability $\rho$ the quality is high ($q_2 = q^*_2$) and with probability $1-\rho$ the quality is low ($q_2 = q^*_1$). Assume that the quality level cannot go down: $q_2 > q_1$. The quality uncertainty of Version 2 software is measured by the variance of $q_2$, $\text{Var}(q_2) = \rho (1-\rho)(q_2^* - q_1^*)^2$. The vendor can control the quality improvements by controlling R&D costs ([5]). However, in order to simplify the model and focus on our research problem of the optimal licensing strategy, we assume that the R&D investments are sunk and the quality of Version 2 is exogenous.

Software is usually designed to be compatible with previous versions (backward compatibility) to take advantage of the existing network of users, but forward compatibility (compatibility with future versions) is not easy to achieve. For example, the new version of Microsoft Office software will allow users to open files created by previous versions natively, but older software may not open the files created in the new. We study the most common case of software production in terms of compatibility: the software is backward compatible but forward incompatible. That is, the later version of the software can successfully use interfaces and data formats from the earlier, but the original software is not designed in such a way that it can seamlessly accommodate files produced with the planned future version. Hence, consumers who upgrade to or buy later versions can enjoy network externalities from the users of both versions, but those who continue using the older version only have a network of users of the same version.

Consumers discount their future utility gain by a factor of $\beta \in [0,1]$. To reduce the number of cases under consideration, we assume $(1+\beta)q_1 \geq \beta E(q_2)$ so that a consumer with a higher quality preference $\theta$ prefers buying Version 1 in period 1 to waiting to buy version 2 in period 2. This assumption is similar to the one in [11], which only excludes very large improvements between the two versions, but considerably simplifies the analysis.

As suggested by [27], we assume that the monopolist vendor can commit to future prices through means like best-price provisions or limited editions. Relaxing this assumption does not change our major conclusions but significantly complicates the problem. Hence we adopt the commitment assumption in order to separate other effects on the licensing models and focus on the impacts of quality uncertainty and network effect.

The software vendor can offer the software to the market through one-time sales, sales of upgrades, and subscription contracts. Depending on the licensing model adopted, the software vendor sets the selling price $p_1$, upgrade price $p_u$, the second-period selling price $p_2$, and/or the per-period subscription fee $p_r$ at the beginning of period 1. Since the vendor cannot tell whether a buyer in the second period owns Version 1, the upgrade price $p_u$ has to be no higher than the sale price of Version 2 $p_2$; that is, $p_u \leq p_2$.

The penalty for deviating the subscription contract is large enough that the user commits to paying for both periods. Thus it is the total subscription payment over the two periods that matters to the vendor but not how it is allocated over the two periods. Without any loss of generality, we assume the per-period subscription fee $p_r$ stays the same over the two periods, as adopted in the subscription practices of the software industry, for
example, Microsoft’s Enterprise Subscription Model, “Oracle On-demand”, “McAfee Subscription Plus”, etc.

Figure 1 depicts all the available consumer choices. Those who purchase Version 1 in period 1 have the option of upgrading to the new version in period 2 at a cost of $p_u$, but those who enter the subscription contract in period 1 will use the new version software without any additional charges besides the per-period subscription fee. To those consumers who do not adopt in period 1, the software vendor is better off selling only Version 2 in period 2 to achieve a higher profit and avoid market cannibalization. We combine the option of subscribing only in period 2 with that of purchasing in period 2 because: 1) Consumers receive the same level of utility from either of the two options in the two-period model. Therefore they will simply choose the option with a lower price. 2) The subscription fee and sale price in period 2 are not correlated with other decision variables (i.e. other pricing schedules).

![Figure 1: Consumer Strategies](image)

The expected value for a risk-neutral user who buys the software in period 1 is given by

\[ V_p(\theta) = U(q_1, x_0, p_1) + \beta \max \{ E[U(q_2, x_2, p_2), U(q_1, x_1, 0)] \} \tag{2} \]

which is the sum of the net utility of a user with quality preference \( \theta \) in period 1 and the discounted net utility from the second period, when the buyer can decide to upgrade to Version 2 or keep using Version 1. \( x_0 \) denotes the mass of adopters in period 1 and \( x_1 \) denotes the mass of first-period adopters who do not upgrade in period 2. By our backward compatibility assumption, \( x_2 \) is the mass of total adopters of either version in period 2. Under the subscription contract, the user receives continuous supply of software with updates for a per-period payment \( p_r \). The expected value

\[ V_L(\theta) = U(q_1, x_0, p_r) + \beta E[U(q_2, x_2, p_r)] \tag{3} \]

which is the sum of the subscriber’s net utility from using the latest version of the software over the two periods.

Finally, the consumer who is inactive in the first period can either buy the new version of the software in period 2 or remain inactive. Her expected discounted value is

\[ V_I(\theta) = \beta \max \{ E[U(q_2, x_2, p_2)], 0 \} \tag{4} \]

The consumer with quality preference \( \theta \) will choose how to adopt the software by maximizing the expected total discounted value:

\[ V(\theta) = \max \{ V_p(\theta), V_L(\theta), V_I(\theta) \} \tag{5} \]

Consumers’ adopting strategies determine the market segmentations. Taking that into account, the software vendor will decide the pricing schedule for each licensing model and choose the optimal one. Now we solve the market equilibrium under each of the three licensing models.

### 2.1 Perpetual Licensing

We consider the case in which the vendor offers the software through selling: he sells Version 1 to a mass of \( x_0 \) users at price \( p_1 \) in period 1, and in period 2 he sells Version 2 to a mass of \( (x_2 - x_0) \) users at the full price \( p_2 \), and a mass of \( (x_0 - x_1) \) early adopters at the upgrade price \( p_u \).

At the beginning of period 1, a consumer of type \( \theta \) evaluates her expected value of buying in period 1 \( V_p(\theta) \) (2) with the expected value of delaying the decision till period 2 \( V_I(\theta) \) (4). There exists a trade-off between buying and waiting: if buying in period 1, the user can start using the software from that period and has the option of upgrading in period 2; otherwise, she cannot use the software in period 1, but retains the option of buying the new version in period 2 when the quality of Version 2 is realized. Thus, a user has six unique strategies over the two periods: buy and upgrade (BU): buy Version 1 in period 1 and upgrade to Version II in period 2; buy and contingently upgrade (BC): buy version 1 and upgrade only when the quality improvement is large in period 2; buy and hold (BH): buy Version 1 in period 1 and keep using it in period 2; buy Version 2 in period 2 (IB); buy Version 2 in period 2 only when quality of Version is high (IC); and remain inactive in both periods (II). By the assumption we made above about quality improvement, \( (1 + \beta)q_1 \geq \beta E[q_2] \), we have the potential market segmentation as depicted in Lemma 1.

**Lemma 1:** Consumers with quality preference \( \theta \in [0, \theta_0] \) are inactive (II); those with \( \theta \in (\theta_0, \theta_1] \) buy Version 2 in period 2 only when quality of Version 2 is high (IC); those with \( \theta \in (\theta_1, \theta_2] \) wait to buy version 2 in the second period (IB); those with \( \theta \in (\theta_2, \theta_3] \) buy version 1 in period 1 but do not upgrade in period 2 (BH); those with \( \theta \in (\theta_3, \theta_4] \) buy Version 1 in period 1 and upgrade contingent on the realized quality in period 2 (BC); and those with \( \theta \in (\theta_4, 1] \) always use the latest version of the software during the two periods (BU).

The cutoff values \( \theta_0, \theta_1, \theta_2, \theta_3 \) and \( \theta_4 \) make consumers indifferent to two adjacent choices:
\[ \theta_0 = p_1 - q_1, \quad \theta_2 = \frac{p_2 - q_2}{q_2}, \]
\[ \theta_3 = \frac{p_3 - \beta q_3 + e(\beta(x_2 - x_1) - x_0)}{(1 + \beta)q_1 - \beta Eq_2}, \]
\[ \theta_4 = \frac{p_4 - e(x_2 - x_1)}{q_2 - q_1}, \quad \theta_5 = \frac{p_5 - e(x_2 - x_1)}{q_2 - q_1}. \]  
(6)

Because the population of potential user is uniformly distributed on the [0,1] support, we plug
\[ x_0 = 1 - \theta_2, \]
\[ x_1 = (1 - \rho)(\theta_3 - \theta_5) + \theta_3 - \theta_2, \]
\[ x_2 = 1 - \theta_1 + \rho(\theta_1 - \theta_0) \]  
(7)
to (6). Solving them simultaneously, we get the market segmentation of the consumers in response to the seller's price schedule \( p_1, p_2 \) and \( p_u \).

Taking into account the consumers' self-selection behavior, the software vendor decides prices to maximize his discounted total profit over the two periods:
\[ \max_{p_1, p_2, p_u} \Pi(p_1, p_2) = p_1 x_0 + \beta p_2 (x_0 - x_1) + \beta p_3 (x_1 - x_2) \]
\[ \text{s.t. } 0 \leq x_1 \leq x_0 \leq x_2 \]  
(8)

Applying Kuhn-Tucker Theorem, we solve this constrained optimization problem by comparing the profits under different combinations of Kuhn-Tucker conditions. We find that consistent with [11], the optimal solution to Equation (8) involves no users buy only in period 2 (leapfrogging). When the expected quality improvement of the software is reasonably large and the intensity of the network effect on utility is small relative to the quality effect, the resulting market segmentation is (II, BC, BU).

We use the superscript \( P \) to represent the equilibrium under the perpetual licensing strategy. The optimal price schedule for the software vendor is \( (p_1^P, p_2^P, p_u^P) \). The software vendor gets profit \( \Pi^P \). Those equilibrium results are shown graphically in Figure 2 and Figure 7.

\[ \text{(a)} \]

\[ \text{Figure 2. The Market Segmentation in Equilibrium under the Perpetual Licensing Policy (} \beta = 0.8, \ q_1 = 1, \ q_2^1 = 1.4, \ q_2^2 = 1.9, \ (a) \ e = 0.2, \ (b) \ \rho = 0.6) \]

The software vendor will be better off committing to a very high price of Version 2 software to prevent consumers from balking to purchase it in period 2 (leapfrogging). Some users will upgrade contingent on the realized quality of the new version, while some users will upgrade to Version 2 no matter what the realized quality is. The vendor will increase both \( p_1 \) and \( p_u \) with a higher expected quality of Version 2, and thus the market share is reduced due to a higher price margin (Figure 2a). With the increase of the network effect, the vendor will lower the price of Version 1 to attract more users in period 1. With the enlarged installed base, the vendor can charge a higher price for the upgrades and still receive a higher profit (Figure 2b).

\[ \text{2.2 Subscription} \]

Due to a confluence of economic, market and technological factors, the software vendor and the users have gradually adopted subscription as a licensing option. We study the type of subscription models as a contract between the software vendor and the users about delivery of the software product together with the latest upgrades for a fixed periodic payment over a certain number of periods. If a user subscribes, she will pay a rent \( p_r \) at the beginning of each period and enjoy the latest version of the software without any additional charge.

The expected discounted value for a subscriber is given in (3). Consumers with \( \theta \geq \theta_l \) will get positive utility from the subscription contract. The level of \( \theta_l \) can be obtained by solving \( V_1(\theta_l) = 0 \)
\[ \theta_l = \frac{(1 + \beta)(p_r - e)}{q_1 + \beta E q_2 - e(1 + \beta)}. \]  
(9)

Given the consumers' choice, the software vendor sets the optimal subscription fee in the contract by maximizing his expected discounted total profit:
\[ \max_{p_r} \Pi(p_r) = (1 + \beta) p_r (1 - \theta_l) \]
\[ \text{s.t. } \theta_l \leq 1 \]  
(10)

Here we use the superscript \( L \) to represent the equilibrium under the subscription (a leasing type of)
strategy. We have the optimal price schedule for the
software vendor, \( p^*_1 = \frac{q_1 + E(q_2)}{2(1 + \beta)} \). The software vendor
gets profit \( \Pi^*_1 = \frac{(q_1 + \beta E(q_2))^2}{4(q_1 + \beta E(q_2) - E(1 + \beta))} \), which is
affected by the mean of quality of Version 2 software
but not its variance.

2.3 A Hybrid Model of Perpetual Licensing and
Subscription
This section considers the optimal pricing policy for a
hybrid strategy that the monopolist software vendor
offers consumers a choice of either perpetual licensing
or subscribing the software. The consumers are faced
with a decision of whether to pay an upfront fee to buy
the software and have the choice of upgrading in the
second period, or to subscribe and automatically receive
the upgrade with an uncertain quality of the next
version. Consumers with the highest level of quality
preference would like to adopt the latest version of the
software. Hence they receive the same utility through
subscription or buying plus upgrading. When both
policies are available, those consumers will choose the
one with a lower cost. Consequently, these two market
segments will not co-exist unless they cost the same,
under which condition high-end consumers are
indifferent to either buying or subscription. Thus, for the hybrid
case to exist, we need to have the total cost of subscription
lower than that of buying and upgrading
(1 + \beta)p_r < p_1 + \beta p_r. By the same reasoning as in
Lemma 2, the potential market segmentation is as
described below.

The cutoff values \( \theta_0, \theta_1, \theta_2, \theta_3 \) and \( \theta_4 \) make
consumers indifferent to two adjacent choices and we
get the market segmentation of the consumers in
response to the seller’s price schedule \( p_1, p_2 \) and \( p_r \).

Taking into account the consumers’ self-selection
behavior, the software vendor decides prices to
maximize his discounted total profit over the two
periods:

\[
\max_{p_1, p_2, p_r} \Pi(p_1, p_2, p_r) = p_1(\theta_1 - \theta_1) + \beta p_r(\theta_4 - \theta_1) + \beta p_2(\theta_2 - \theta_1) + \beta \rho p_1(\theta_2 - \theta_1) + \beta p_r(\theta_4 - \theta_1) + (1 + \beta)p_r(1 - \theta_1)
\]

s.t.

\[
0 \leq \theta_0 \leq \theta_1 \leq \theta_2 \leq \theta_3 \leq \theta_4 \leq 1
\]

Applying Kuhn-Tucker Theorem, we solve this
constrained optimization problem by comparing the
profits under different combinations of Kuhn-Tucker
conditions. Similar to the equilibrium under the
perpetual licensing model, we find that the optimal
solution to Equation (12) involves no users
leapfrogging. When the expected quality improvement
of the software is reasonably and the intensity of the
network effect on utility is small relative to the quality
effect, the resulting market segmentation is \( (\Pi, \Pi_1, L) \).
We use the superscript \( H \) to represent the equilibrium
under the hybrid model. The optimal price schedule for
the software vendor is \( (p_1^{\Pi}, p_2^{\Pi}, p_r^{\Pi}) \), which is given
by Proposition 1.

**Proposition 1:** When the software vendor adopts
the hybrid licensing model and can commit to future prices,
there exists a feasible set of parameter values such that
the profit-maximizing problem of the software vendor is
joint concave of the prices. Hence, a unique equilibrium
exists under which the software vendor charges Version
1 software \( \frac{1}{2} \left( q_1 + E(q_2) - q_2 \right) \) in
period 1, upgrade price \( \frac{1}{2} (q_2 - q_1) + \frac{1}{2} \left( 40(1 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_2 q_1) + \frac{1}{2} \beta^2(2 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_2 q_1) \right) \) in
period 2, and per period subscription fee

\[
\frac{1}{2} (q_1 + \beta E(q_1)) + \frac{1}{2} \left( p_1 + \beta p_2 \right) - (1 + \beta)p_r = \frac{1}{2} \left( q_2 - q_1 \right) - \frac{1}{2} \left( 4(1 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_2 q_1) \right) + \frac{1}{2} \left( 4(1 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_2 q_1) \right)
\]

The optimal price for Version 1 software is based on its
quality and the vendor will have incentive to lower the
price of the first version to increase the installed base in
period 1 due to the network effect. The vendor charges
the subscription fee based on the quality of Version 1
and the (discounted) expected quality of Version 2. He
also takes into account the network effect, trading off
the conflict of profit margin and market share. The price
discount offered by the subscription contract to those
consumers who always adopt the highest quality product

\[
\frac{1}{2} \left( q_1 + \beta q_2 \right) - (1 + \beta)p_r = \frac{1}{2} \left( q_2 - q_1 \right) - \frac{1}{2} \left( 4(1 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_1 - q_2) \right) + \frac{1}{2} \left( 4(1 + \beta)(q_1 - q_2) - (2 + 2 \beta)(q_1 - q_2) \right)
\]

Figure 3 shows the equilibrium market segmentation
under the hybrid licensing policy with the changes of
probability of a high quality for Version 2 \( \rho \) and of the
intensity of the network effect \( e \).
When the network effect is significant, as those under the pure subscription strategy in §2.2, equilibrium market share and vendor profit are the same if subscribe or stay inactive in both periods. Thus the price of Version 1 to convert those consumers to either contribute to the vendor’s second-period profit. Therefore it is optimal for the vendor to charge a high price of Version 1 but do not upgrade will not contribute to the vendor’s second-period profit. Therefore it is optimal for the vendor to charge a high price of Version 1 to convert those consumers to either subscribe or stay inactive in both periods. Thus the equilibrium market share and vendor profit are the same as those under the pure subscription strategy in §2.2.

When the network effect is significant, \( e > 0 \), a consumer’s utility increases with the installed base, so the vendor has a greater incentive to increase the market share in the early period. Those customers who do not upgrade will increase the installed base of the software, and they thus help increase other users’ utilities to the benefit of the vendor. With an increase in the intensity of the network effect, the software vendor will lower the first-period price to attract more early buyers. As a result, he can charge more users a higher subscription and upgrade price. Summing up the discounted profits over the two periods, the vendor is better off with a greater network effect.

We next compare the three licensing strategies in terms of pricing, profit, consumer surplus and social welfare.

3. Comparison of the Three Licensing Models

Comparing the market structure and the software vendor’s profit among the three license models: perpetual license, subscription, and a hybrid model, we have the following conclusions from our analytical results.

**Proposition 2**: When a software vendor can commit to future prices and there is no network effect,

i) the monopolist software vendor receives less market share but a higher profit by adopting the subscription model than the perpetual licensing model; the hybrid licensing model has the same equilibrium as the subscription model: \( \Pi^L \leq \Pi^H = \Pi^S \);

ii) those consumers who buy and do not upgrade incur a lower total expense under the perpetual licensing model than under the subscription model \( (p_i < (1 + \beta) p_j) \), while those who always use the latest version have to pay more under the subscription model than under the perpetual licensing \( (p_i + \beta p_j < (1 + \beta) p_j) \).

iii) the consumer surplus and social welfare when the vendor chooses the perpetual licensing model are higher than or equal to those of the subscription model: \( CS^L \leq CS^H \leq CS^S \), and \( W^L \leq W^H \leq W^S \).

If there is no network externality effect \( (e = 0) \) and the vendor can commit to future pricing, it is optimal for the vendor to sell only to high-end users who always adopt the latest version software but not to those users who buy-and-hold, those who selectively upgrade, or those who opportunistically leapfrog. The vendor prefers the subscription strategy, under which the vendor targets a smaller market share but extract more surplus from those consumers. The difference of the profits under the subscription and the perpetual licensing models is proportional to \( (q^L_i - q^S_i)^2 \). Thus, the higher the variance of the quality of Version 2 software, the more likely the vendor chooses the subscription licensing model. The vendor’s choice, however, is inconsistent with the social optimum.

Software, however, has a strong network effect because there is value created from file sharing and knowledge exchange among adopters, as well as from compatible products ([2]). Users therefore will value more a software product with a greater population of adopters. When we consider the network externality effect and compare the market equilibria under the three models, we obtain Proposition 3.

**Proposition 3**: When a software vendor can commit to future decisions and there is no uncertainty regarding quality of Version 2,

i) the perpetual licensing or hybrid model weakly dominates the subscription model in terms of both market share and profit: \( \Pi^L \leq \Pi^H \leq \Pi^S \);
(ii) those consumers who buy and do not upgrade incur a lower total expense under the perpetual licensing model than under the subscription model \((p_u < (1 + \beta)p_p)\), while those who always use the latest version have to pay more under the perpetual licensing model than under the subscription model \((p_u + \beta p_p > (1 + \beta)p_p)\); and

(iii) the consumer surplus and social welfare are no higher when the vendor chooses the subscription model:

\[ CS^s \leq CS^p = CS^H, \ W^s \leq W^p = W^H. \]

When the committed vendor sells the software in a market with network effect and there is no uncertainty regarding the quality of Version 2 software, the perpetual licensing model will generate both a greater market share (Figure 4) and a higher profit (Figure 5). The differences of the market share and profit between the perpetual or hybrid licensing model and the subscription model are increasing with the intensity of the network effect \(e\).

![Figure 4. Market Segmentations of Perpetual or the Hybrid Licensing Policy (the dotted curve) vs. Subscription (the solid curve) in Case 1 when \(\beta = 0.7, q_1 = 0.9, q_2 = 2.18\).](image)

![Figure 5. Profit of Perpetual or the Hybrid Licensing Model (the dotted curve) vs. Subscription (the solid curve) when \(\beta = 0.7, q_1 = 0.9, q_2 = 2.18\).](image)

Consumers are also better off under a perpetual or the hybrid licensing model. The perpetual licensing model allows the medium-value consumers to buy Version 1 without the obligation of upgrading, while a subscription model forces them to upgrade by contractual agreement. The purchase option therefore significantly reduces those users’ cost of using the first version of the software. With the consequent increase in the population of users, the network externality effect becomes stronger, enabling the vendor to charge a higher upgrade price to the high value consumers. Figure 8 indicates that consumers who always use the latest version of the software thus incur a higher total cost under the perpetual or hybrid licensing model than under the subscription model. Their loss, however, is outweighed by other consumers’ gains, so the total consumer surplus under the perpetual or hybrid licensing model is greater.

Proposition 3 has important implications for the recent subscription licensing activities in the software industry. Subscriptions with automatic upgrades deprive some medium-value consumers of the flexibility of buying but not upgrading. A subscription model thus forgoes some market share, which has great value for goods with a network effect. We show that when the software vendor can commit to future prices and there is no uncertainty regarding the quality of the future version software, both the vendor and users prefer the perpetual or hybrid licensing model than a subscription model.

The above results suggest that when the quality uncertainty or the network effect is considered alone, the software vendor prefers the subscription model when the uncertainty of the quality of Version 2 is high, and prefers the perpetual licensing model when the network effect is great. When there is both network effect associated with the product and uncertainty in the quality of Version 2 software, it is unclear which licensing model will be a better choice for the vendor. Because the analytical results are not very straightforward to derive comparison results, we use the following numerical example to compare the equilibria of the three licensing models. We pick the values of the parameters to make sure both the quality uncertainty of Version 2 and the network externality effect are not too small and all the assumptions of the model are satisfied: \(\beta = 0.8, q_1 = 1, q_2 = 1.5, q_3 = 2.1, \) and \(\rho = 0.7\).

When the network effect is large enough, the software vendor receives the highest profit (Figure 6) and the greatest market share (Figure 7) through the hybrid licensing model. The vendor has a medium level profit but the smallest market share under the subscription model. The market share and the gain in profit are increasing with the intensity of the network effect. The perpetual licensing model provides users with the option of buying Version 1 without the obligation of upgrading. Those buying-and-holding consumers can increase the software’s user population to the benefit of the vendor. On the other hand, under the perpetual licensing model, the software vendor has to charge the same upgrade price \(p_u\) to all the adopters of Version 1 with different quality preferences. Adding subscription to the perpetual licensing model can better refine those users with the subscription fee \(p_s\) and the upgrade price \(p_u\). By narrowing the segment of conditional upgrading consumers, the vendor can reduce the downside risk of the quality uncertainty on his profit. Therefore when the quality of new version software is uncertain and the network effect is large enough, the vendor should choose the hybrid licensing model.
With the increase of the network effect, the vendor will have a greater incentive to cut the first period prices to enlarge the installed base, so that he can charge a higher upgrade price in the second period to extract part of the consumer surplus increased due to the network effect.

Figure 8 shows that adding perpetual licensing to the subscription model will not only increase the vendor’s profit but also increase consumer surplus. Consumers benefit from the flexibility in updating under the perpetual licensing model, which has the highest consumer surplus. When the network effect is strong enough, the vendor’s licensing model selection (the hybrid model) is consistent with social optimum.

This section provides key theorems with ample analysis and evidence to show how the vendor should trade off the benefit and cost of the perpetual and subscription licensing models to choose the optimal one based on the levels of the network effect and of the uncertainty regarding the quality of future version software: when the network effect is large, he will be more likely to choose the perpetual licensing model to increase the installed base in period 1; when the quality uncertainty is large, he will be more likely to adopt the subscription licensing model to lock in the users from period 1; and when both effects exist and are significant, it may be optimal for the vendor to adopt a mixed strategy of both licensing models.

Even though the above results are based on the assumption that the vendor can commit to future prices, we now discuss that relaxing this assumption does not change our major conclusions presented above. When the vendor cannot commit to the future prices, we will solve the problem as a two-stage game by backward induction. The vendor will announce the second period selling and upgrading prices after the quality of Version 2 is realized.

Therefore given a quality level of Version 2, the vendor will price conditionally and receives a conditional second period profit. Then the vendor will consider the uncertain profit of period 2 when he decides the first period prices by maximizing the discounted total profit of the two periods. Consistent with our results under the committed scenario, a higher uncertainty in quality of Version 2 will cause bigger losses in the vendor’s discounted total profit, and the vendor will have greater incentive to adopt the subscription licensing model to remove the variance in profit. Even when the vendor cannot commit to future prices, the path dependence of the network effect still exists. Hence, those consumers who buy and hold who can only exist in the perpetual licensing model will still add value to the utilities of later adopters and indirectly contribute to the vendor’s profit. When the network effect is large enough, the vendor is better off choosing a hybrid licensing model. We focus on the impacts of network effect and quality uncertainty on the vendor’s licensing model selection assuming the vendor can commit to future prices in this paper because 1) this assumption is supported by literature such as [27]; 2) considering the uncommitted case will lose the tractability of the problem without adding enough new insights; 3) the main insights of the paper will still hold in the uncommitted scenario.

4. Conclusions
Software vendors now can choose to deliver software based on either a perpetual ownership licensing model which offers customers the permanent right to use and own the version of the software they purchased, or a multi-period subscription model which includes a commitment to a periodic payment for a continuous usage of the latest version of the software during the contract periods. Recent speculations of practitioners in the software industry, predict that subscription is likely to become the dominant means of licensing software.
Our paper addresses the above issue by examining the optimal licensing strategy for a monopoly software vendor with a two-period analytical model combining vertical differentiation and inter-temporal price discrimination. We suggest that software vendors should consider the uncertainty of quality improvement and the intensity of the network effect in choosing their optimal licensing models. A high level of quality uncertainty will increase the likelihood of consumers' delaying making the upgrade decisions till the quality is realized. In the presence of quality uncertainty, vendors should consider using subscription licensing to lock consumers in during the first period to prevent their opportunistic behaviors. For example, one of Microsoft’s strategies to fight with the slowing down speed of introducing new version of software products and the competition with itself is providing and actively promoting the subscription licensing policy to its enterprise users.

However, when there is a strong network effect, as is the case with most software, subscription model prevents some earlier adopters from generating positive externalities to the later users, which reduces the value of the software to consumers and hurts the vendor's bottom line. On the other hand, perpetual licensing provides users the right to keep using the same version of the software in later periods without further payment. Due to backward compatibility, those users will contribute to the network size and add value to later adopters. Therefore a strong network effect will increase the vendor’s incentive to lower the price of the first version in order to increase the installed base, thereby increasing the network effect and values of future adopters, which will in turn allow the vendor to extract more surpluses from adopters of later periods. When this effect is pronounced, the vendor is better off choosing a hybrid model rather than the pure subscription model.

Our results are validated by the actual licensing practices in the software market. The majority of the software vendors, such as Microsoft, Sun, Oracle and SAS, are not completely using subscription to replace the perpetual licensing model, but have added subscription as another option in their original licensing models.

Our findings should also help users realize the benefits and costs of each licensing model and help them make the best decision, taking into account their own characteristics, product upgrades, quality uncertainty, and network effect. A subscription model smooths out cash payments and replaces a single lump sum with a per-period payment, which may be attractive when software budgets are limited.

Consumers need to consider the total cost of ownership, however, when evaluating their options. Our model shows that under the subscription model, users can always enjoy the latest version of the software, but they are deprived of the option of not upgrading. The total cost of subscription is higher than buying but not upgrading. Thus, the claim that the subscription model lowers the cost of ownership is misleading and myopic.

Our consideration of consumer surplus and social welfare likewise has implications for policy makers. The subscription model gives the software vendor more market power but deprives consumers of upgrade options. Consumer surplus is higher under the perpetual and hybrid licensing models. Moreover, social welfare is also lowest under subscription model according to our Propositions 2 and 3. Hence policy makers should encourage a mixed of the perpetual and subscription model rather than the pure subscription model in the software market.

Network externalities, quality uncertainty combined with compatibility issues endogenize the externality of a user’s adoption decision into the decision of the other users. Thus, network effects significantly complicate the study of software licensing models. In order to explore the impact of network externalities in particular, we have made several assumptions to simplify the problem. Relaxing those assumptions does not affect our major conclusions but does create interesting opportunities for future research.

References


