

3-2018

A model of competition between perpetual software and software as a service

Zhiling GUO

Singapore Management University, ZHILINGGUO@smu.edu.sg

Dan MA

Singapore Management University, madan@smu.edu.sg

Follow this and additional works at: http://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#), and the [Management Information Systems Commons](#)

Citation

GUO, Zhiling and Dan MA. A model of competition between perpetual software and software as a service. (2018). *MIS Quarterly*. 42, (1), 1-46. Research Collection School Of Information Systems.

Available at: http://ink.library.smu.edu.sg/sis_research/3712

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

A MODEL OF COMPETITION BETWEEN PERPETUAL SOFTWARE AND SOFTWARE AS A SERVICE¹

Zhiling Guo and Dan Ma

School of Information Systems, Singapore Management University, 80 Stamford Road,
Singapore 178902 SINGAPORE {zhilingguo@smu.edu.sg} {madan@smu.edu.sg}

*Software as a service (SaaS) has grown to be a significant segment of many software product markets. SaaS vendors, which charge customers based on use and continuously improve the quality of their products, have put competitive pressure on traditional perpetual software vendors, which charge a licensing fee and periodically upgrade the quality of their software. We develop an analytical model to study the competitive pricing strategies of an incumbent perpetual software vendor in the presence of a SaaS competitor. We find that, depending on both the SaaS quality improvement rate and the network effect, the perpetual software vendor adopts one of three different strategies: (1) an **entry deterrence strategy**, (2) a **market segmentation strategy**, or (3) a **sequential dominance strategy**. Surprisingly, we find that vendor competition does not always result in higher consumer surplus, and it might lead to a socially inefficient outcome under certain conditions. We further show insights into how the incumbent perpetual software vendor can defend its market position by providing incremental quality improvement through patching and/or by releasing consecutive versions with major quality upgrades. Finally, we extend our model to include the vendor's quality improvement cost and users' switching cost. These additional analyses help to identify the effect of different quality and cost factors on the market competitive equilibrium.*

Keywords: Software as a service (SaaS), network effects, pricing and competition, switching cost, game theory, analytical modeling

Introduction

The emergence and growth of software as a service (SaaS) has fundamentally changed how software can be delivered, used, and managed. SaaS represents a new software delivery and pricing model, in which the vendor hosts, maintains, and manages the application from a central location; serves clients through a network; and charges them based on use. In the past decade, SaaS has been one of the fastest growing market segments and is now the third largest component of the total

cloud computing industry. Gartner (2017) forecasts that the SaaS market will increase 20.1% in 2017 to total \$46.3 billion, up from \$38.5 billion in 2016, and reach \$75.7 billion by 2020. Today almost every software segment—office suites, database management systems, supply chain management systems, financial and accounting applications, human resources management software, business intelligence applications, and customer relationship management applications—have been transformed by SaaS (Columbus 2013).

SaaS vendors have exerted strong competitive pressure on traditional perpetual vendors in many software niche markets. A recent example is the competition between Blackboard and Canvas in the learning management system (LMS) marketplace. Blackboard is a perpetual software system that many universities have used since the mid-1990s. Blackboard held 90% market share in the LMS market in 2006, but it more

¹Sulin Ba was the accepting senior editor for this paper. Zhengrui Jiang served as the associate editor.

The appendices for this paper are located in the “Online Supplements” section of the *MIS Quarterly*'s website (<http://www.misq.org>).

recently has dropped to about 34% (Bogage 2015). Blackboard lost many clients to new rivals, among which Canvas is the most remarkable. Canvas is a cloud-based SaaS product that was initially developed in 2008; it quickly became a strong competitor of Blackboard, accounting for 9.3% market share in 2015 (wiki.listedtech.com). Many high-profile clients of Blackboard, such as the University of Texas at Austin and Northwestern, have moved to Canvas. Both researchers and industrial users have been intrigued: How could the new cloud-based entrant successfully compete with the well-established incumbent, which represented the industry standard and held a monopoly in the market for almost 20 years? Our goal in this research is to understand the market competition between these two types of software vendors, which have different pricing schemes, product delivery and operating models, and unequal initial market positions.

Prior research has examined the competition between perpetual software and SaaS vendors in terms of their distinct pricing methods (Fan et al. 2009; Ma and Seidmann 2015), cost structures (Huang and Sundararajan 2011), and internal IT infrastructures (Chen and Wu 2013). However, studies have not yet considered the continuously increasing software quality of SaaS, which is an important factor in the competition. Because of the centralized location and management of SaaS applications, the vendor is able to add new functionalities, offer more features, correct defects, patch the software almost anytime during its life cycle, and therefore continuously increase product quality. More importantly, these quality improvements are free to users. In contrast, the quality improvement of perpetual software takes a step-up form. To satisfy the demand for quality upgrades from existing users and to attract new users, the perpetual software vendor typically adopts a leapfrog strategy: When the current version has been in the market for some period of time, the vendor releases a new, higher quality version as a replacement (Bala and Carr 2009; Ellison and Fudenberg 2000). In many cases, the vendor prices the new version strategically by offering a lower upgrade price to existing customers while charging new clients the full purchase price (Bala and Carr 2009).²

Although (Choudhary 2007) has noted these differences between the two types of vendors' quality improvement strategies, he limits his analysis by assuming that, at the end of the product life cycle, SaaS offers the same quality as perpetual software. In contrast, we allow SaaS quality to be either lower or higher than perpetual software, so that the two vendors might take turns being the higher quality provider

²For example, Microsoft's upgrade price for Office Professional 2007 was 16% less than the full purchase price (\$263.95 vs. \$312.79). In contrast, Adobe charged 69% less for its Photoshop CS3 upgrade (\$195.99 versus \$629.49).

over the planning horizon. As a result, we expect to see strategic users switching from one vendor to the other. This perspective renders our analysis more realistic and interesting, which enables us to generate richer insights than those offered by the extant literature.

We also consider other features of the perpetual software and SaaS competition that have largely been neglected in previous studies. First, perpetual software vendors have a much longer history than their SaaS competitors. In most software niche markets, and for many decades, they have had an established customer base and extensive market power. In contrast, SaaS vendors have only recently begun encroaching on perpetual vendors' territory. This delayed entry by the latter suggests that the two types of vendors are in unequal positions and that the perpetual software vendor has a first-mover advantage. Second, the software market is expanding. Existing perpetual software users who have built in-house systems continue to demand higher quality products; and new users, with new demands, also enter the market. To retain existing users and attract new users, the perpetual software vendor releases a new version of its software during market expansions (Mehra et al. 2014). It also makes the new version backward-compatible with the previous version (Ellison and Fudenberg 2000), so that it can leverage its existing market base to compete with the new SaaS entrant.

In this research, we propose an analytical model to study the competition between an incumbent perpetual software vendor, which has an established market base, and a new SaaS entrant. We aim to answer the following research questions: What are the incumbent perpetual software vendor's competitive pricing strategies in the presence of pressure from a new SaaS entrant? Can it successfully deter the entry of the SaaS vendor by relying on its first-mover advantage? If SaaS quality rapidly increases, how should the perpetual software vendor defend its market share? How do existing and new users make decisions about software adoption? What are the implications for consumer surplus and social welfare?

We find that a SaaS entrant's competitiveness depends on how fast it can improve product quality and how strong the network effect is. In general, the SaaS vendor's competitive power increases when its quality improvement rate is high or when the network effect is weak. Meanwhile, when the SaaS becomes a stronger competitor, the incumbent perpetual software vendor is advised to adjust its pricing strategy accordingly, by moving from an *entry deterrence strategy*, in which it uses low prices to successfully deter the SaaS's entry, to a *market segmentation strategy*, in which the two vendors charge high prices to exploit their respective client segments while avoiding direct competition, and ultimately to a *sequential dominance strategy*, in which the vendors aggres-

sively compete on price and all users first adopt the perpetual software and then switch to the SaaS when the SaaS product quality bypasses the perpetual software. Surprisingly, we find that vendor competition does not always benefit customers. When the SaaS quality improvement rate is low, the new entrant brings a low-quality product to users and thus reduces consumer surplus and social welfare.

Our analysis also suggests that the perpetual software vendor, as the incumbent, can better defend its market position by offering periodic quality improvement using two strategies. One strategy is to provide free incremental quality patching during the software life cycle. In this case, the vendor must consider the tradeoff between the patching time and the magnitude of the quality improvement. We show that the optimal patching time depends on both the SaaS vendor's quality improvement rate and users' perceived total value of the patch. The other strategy is to release consecutive new versions that provide major quality jumps. We find that, compared with the single-period competition, the perpetual software vendor is more likely to adopt entry deterrence and market segmentation strategies over a longer competition horizon.

Furthermore, we extend the baseline model in several ways to gain richer insights into the vendors' and users' decision making. We find that the ongoing quality improvement costs for the SaaS vendor allow the incumbent to more easily block the new rival's entry to the market. We further show that when existing perpetual software users face a switching cost, the perpetual software vendor can more easily preserve its existing installed base and earn a higher profit.

The paper is organized as follows. First, we discuss related past research and highlight our contributions to multiple research fields. We then present our baseline model, including vendors' competition and users' strategies. In the following section, we derive the two vendors' optimal pricing strategies and market equilibrium outcomes. We then provide in-depth analysis on how the perpetual software vendor can defend its market power through appropriate quality improvement strategies. We further extend the baseline model in relation to several aspects of the vendors' and users' decision making. Finally, we discuss the main findings of this research and their implications, conclude the paper, and present limitations and suggestions for future research.

Literature Review

Our work is related to several streams of literature: the SaaS business model, product quality differentiation, and software network externality.

Prior research has examined several interesting features of the SaaS business model, including its on-demand pricing, discontinuous cost structure, contract design, and risk-sharing mechanism. For example, based on transaction-cost economics, Susarla et al. (2009) argue that, because of contractual incompleteness and opportunism by vendors, the SaaS contract design should be multidimensional, should address *ex post* transaction costs, and should offer effective governance structures to protect users. Huang and Sundararajan (2011) identify two types of costs in SaaS provision: infrastructure costs and service costs. They find that, with discontinuous costs and shared infrastructure, the widely adopted full-cost pricing mechanism is typically suboptimal. Huang et al. (2015) propose a hybrid service delivery and pricing mechanism for a monopoly SaaS vendor and demonstrate that their proposed scheme outperforms a pure on-demand subscription pricing approach. Kim et al. (2010) propose a risk-sharing mechanism between SaaS vendors and customers to improve software reliability. August et al. (2014) study the effect of security risks on a software vendor's versioning strategy when it chooses to offer a SaaS variant in addition to an existing on-premise product.

Another line of research compares the SaaS and perpetual software business models and examines their competition. Choudhary (2007) finds that a vendor's incentive to invest in quality is higher under the SaaS model than under the perpetual licensing model. Zhang and Seidmann (2009) show that when the network effect is sufficiently strong, a monopoly software vendor should provide both perpetual licensing and SaaS subscription pricing. Fan et al. (2009) study short-term price competition and long-term quality competition between a SaaS vendor and a traditional shrink-wrap software firm, and they find that the SaaS vendor's high service operation costs affect its competitiveness. In addition, Ma and Seidmann (2015) show that SaaS's lack-of-fit costs play a critical role in the outcome of market competition. In this paper, we study how perpetual software vendors and SaaS vendors compete when they have different pricing schemes, disparate software quality improvement models, and unequal market positions. We also take into consideration the behavior of different user generations with respect to software adoption, upgrades, and switching over a finite time horizon.

Much research has been conducted on product quality differentiation. Firms can adopt different product-differentiation strategies to avoid head-to-head competition (Tirole 1992). They can provide products with heterogeneous features to differentiate horizontally (Hotelling 1929), or they can offer products that have varying quality levels to differentiate vertically (Shaked and Sutton 1983). Product differentiation leads to market segmentation, which enables competing firms to profit by serving their respective customer segments. For example, Salop (1979) shows that a monopolistic competitive

equilibrium exists for two horizontally differentiated firms when each serves only local customers and behaves like a local monopolist in its own market segment. Many other studies (Bhargava and Choudhary 2001; Moorthy 1984; Vandenbosch and Weinberg 1995) suggest that firms can offer products at different quality levels and charge different prices so that they can price-discriminate among users and thus realize higher profits. Interestingly, when competing with vertically differentiated products, the low-quality provider under certain conditions can take a more advantageous position in the market (Lambertini and Tampieri 2012; Ma and Kauffman 2014). A key limitation of these studies is that they assume constant product quality, which neglects and precludes the examination of SaaS's dynamic product quality improvement. In this paper, the SaaS vendor improves its software quality continuously. The quality advantage might switch from one vendor to the other during the product life cycle. This consideration of product quality differentiation in a dynamic environment distinguishes our work from previous studies in this area.

The third stream of literature to which our study relates focuses on product life cycle, upgrade and patch management, and the network effects of software applications. Offering successive upgrades and patching existing software is a common strategy in the software market (Arora et al. 2006; Bala and Carr 2009; Cavusoglu et al. 2008; Kim et al. 2010). Because software has a limited life cycle and becomes obsolete quickly (Mehra et al. 2014), existing users always demand a better product with more features and higher quality to replace the old one. In addition, the market for new products expands over time (Bass 1969; Norton and Bass 1987, 1992). When new users enter the market, they create new business opportunities for software vendors. Prior research has shown that a perpetual software vendor is able to price-discriminate when selling successive software versions by offering a lower upgrade price to its existing users and a full purchase price to new clients (Bala and Carr 2009; Ellison and Fudenberg 2000).

Finally, network effects play an important role in the consumption of software products, which is characterized by network externality (Choi 1994). Ellison and Fudenberg (2000) argue that selling new backward-compatible-only upgrade versions is now a common practice in the software market. Backward compatibility encourages existing users to upgrade, which generates extra network value that, in turn, induces new users to buy the new version. As a result, the vendor with an established customer base has the first-mover advantage in the competition with the new entrant. We propose an integrated framework that takes into account the perpetual software vendor's price discrimination, the SaaS vendor's continuous quality improvement, the network effects

of the software, and user behavior regarding upgrades and switching. Our model offers rich insights into the competitive strategies the two types of vendors can adopt in the software market.

The Baseline Model

In this section, we first introduce our model setup, focusing on the two vendors' distinct pricing methods and quality improvement strategies. We then define users' utility functions and discuss their various software adoption strategies.

The Model Setup

Consider a finite selling horizon $[0, 1]$, which also represents the life cycle of a perpetual software product. At $t = 0$, an incumbent firm—the traditional vendor that offers a perpetual software product—has an installed base of users with mass 1. We call these users *old generation (OG) users*, and the software they have purchased is the *old version perpetual software*. OG users already paid a one-time licensing fee to enjoy a lifetime use of the old version perpetual software. We assume the quality of the old perpetual software is q .

At $t = 0$, market expansion occurs and another group of users with mass 1 enters the market, generating new demand for software use. We call them *new generation (NG) users*.³ In addition, the perpetual software vendor releases a new version of its software with quality ρq , where $\rho > 1$ represents the quality improvement over the old version. The OG users have the option of upgrading to the new version by paying an upgrade price, p_u , while the NG users can purchase the new software by paying a purchase price, $p_n \geq p_u$. A strict inequality means that the vendor offers a price discount to its existing customers. Both the old version and the new version of the perpetual software simultaneously exist in the market, and their qualities remain unchanged throughout the software life cycle.⁴

This market expansion also attracts a potential new competitor, the SaaS vendor. We assume that the SaaS quality is θq at $t = 0$, where $1 < \theta < \rho$. Thus, the initial quality of the SaaS software is higher than the old version perpetual soft-

³Our model considers the market expansion at a fixed time—namely, all NG users enter the market and make the software adoption decision at $t = 0$. In Appendix K, we analyze an alternative setting in which NG users arrive in the market continuously in the time interval $[0, 1]$.

⁴In a later section, we provide an in-depth analysis of the competition when the quality of the perpetual software improves as a step function over time.

ware, recognizing that progress in software technology generally makes a new system better than the old one. However, the SaaS quality is lower than the new version perpetual software, showing the traditional software vendor's technological leapfrog. The SaaS quality continuously increases over time because the vendor is able to release new functionalities, deliver advanced features, and fix identified errors from the centralized server where the SaaS application is installed (Choudhary 2007; Zhang and Seidmann 2009). For simplicity, we assume that SaaS quality is a linear function of time, $q(t) = \theta q + \alpha t$, where α is the rate of quality improvement.

Note that the two vendors have different quality improvement strategies. The perpetual software vendor typically packages major updates into the new version of its software, releases the new version at the end of the life cycle of the existing one, and sells it to the market for new profits. In contrast, the SaaS vendor improves product quality continuously during the software life cycle and offers frequent upgrades free of charge. Consequently, the quality advantage might switch from one vendor to the other during the competition period $[0, 1]$, depending on how fast the SaaS quality increases. In our analysis, we consider the following two scenarios. In the first scenario, α is small (i.e., $\alpha \leq (\rho - \theta)q$), so that $q(1) = \theta q + \alpha \leq \rho q$, implying that the SaaS is always inferior to the new perpetual software during the entire software life cycle. In the second scenario, the SaaS quality improvement rate α is large (i.e., $\alpha > (\rho - \theta)q$), so that $q(1) = \theta q + \alpha > \rho q$, meaning that by the end of the software life cycle, the SaaS quality exceeds the new perpetual software. The SaaS offering becomes the highest quality product on the market. Figure 1 illustrates software quality over the entire time horizon in the case of a significant improvement in SaaS quality.

Furthermore, in terms of pricing, the perpetual software vendor charges a one-time price, while the SaaS vendor charges a price p_s per unit of time, allowing users to pay as they go. Therefore, the SaaS user's total payment to the vendor is calculated as the unit price times the length of the period of use. We assume the SaaS price p_s remains unchanged in the planning horizon.⁵ A complete list of notations is provided in Table A1 in Appendix A.

⁵In reality, many SaaS vendors (e.g., Salesforce.com) have maintained stable prices in the past decade. The strong competition in the software market has led to an industry standard in which most SaaS vendors patch their products and offer upgrades for free. Even with the improved quality over time, peer pressure makes price increases by SaaS vendors unrealistic—especially when the perpetual software vendor maintains a constant price during its product life cycle. In fact, cloud service providers are facing downward pricing pressure, and many of them, including Google, Amazon, and Microsoft, have been involved in a round of price cuts for their cloud services in recent years (Babcock 2013; Sullivan et al. 2012).

User Utility Definition and Strategy Analysis

Both OG and NG users must choose their software adoption strategies over the selling horizon $[0, 1]$. We assume complete information so that users know the two software vendors' prices, (p_u, p_n) and p_s , and they are aware of the quality of the two perpetual software versions, $q, \rho q$, and the SaaS's initial quality θq , as well as its quality improvement rate α . We define a user's *total utility* as the sum of a *base utility* from consuming the software product (without consideration of the network effect) and an additional *network utility* derived from the positive network effect. When network effects exist, users must form expectations about the size of the network. Following the literature of network externality (Katz and Shapiro 1985; Kreps 1977; Stokey 1981), we use the rational (or fulfilled) expectations equilibrium concept, which means that the users' expected software network size is equal to the actual network size in equilibrium.

We first present OG users' feasible strategies and the corresponding base utilities:

Old: The user keeps using the old perpetual software over the entire period $[0, 1]$. The base utility is q .

Old + SaaS: The user keeps using the old perpetual software in the period $[0, t_{s1}]$. It switches to SaaS in the period $[t_{s1}, 1]$.

The base utility is $qt_{s1} + \int_{t_{s1}}^1 (\theta q + \alpha t - p_s) dt$.

SaaS: The user uses the SaaS software over the entire period

$[0, 1]$. The base utility is $\int_0^1 (\theta q + \alpha t - p_s) dt$.

Upgrade: The user upgrades to the new perpetual software at time 0 and uses the new version over the entire period $[0, 1]$. The base utility is $\rho q - p_u$.

Upgrade + SaaS: The user upgrades to the new perpetual software at time 0 and uses it in the period $[0, t_{s2}]$. It switches to SaaS in the period $[t_{s2}, 1]$. The base utility is $\rho qt_{s2} - p_u +$

$\int_{t_{s2}}^1 (\theta q + \alpha t - p_s) dt$.

NG users do not have the option of using the old version perpetual software. Their feasible strategies and corresponding base utilities are:

New: The user purchases the new perpetual software at time 0 and uses it over the entire period $[0, 1]$. The base utility is $\rho q - p_n$.

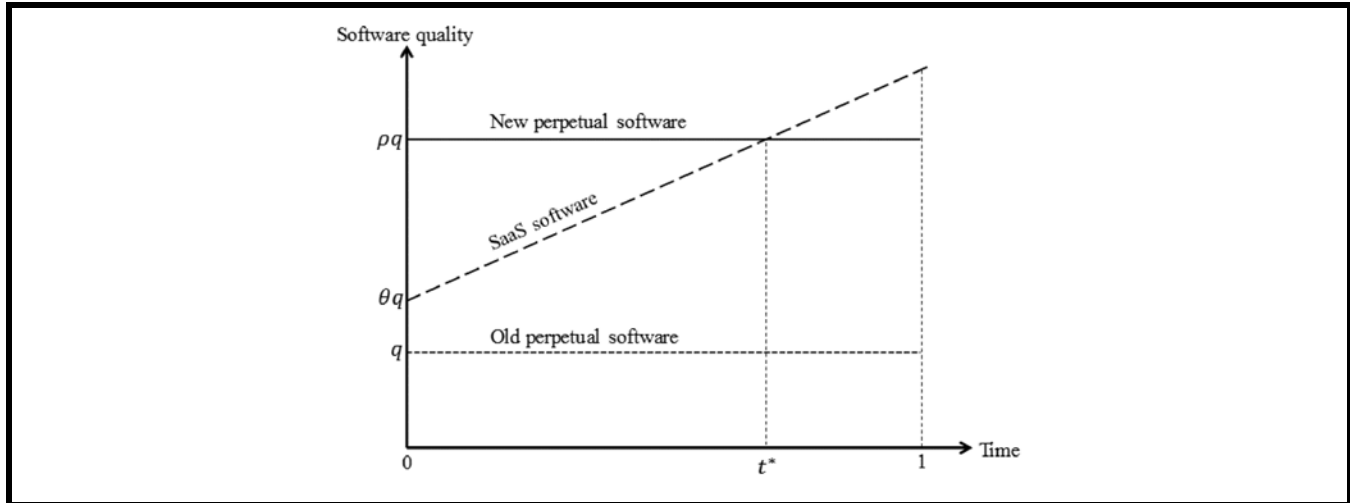


Figure 1. Quality Comparison Between Perpetual and SaaS Software

New + SaaS: The user purchases the new perpetual software at time 0 and uses it in the period $[0, t_{s3}]$. It switches to SaaS in the period $[t_{s3}, 1]$. The base utility is $\rho q t_{s3} - p_n + \int_{t_{s3}}^1 (\theta q + \alpha t - p_s) dt$.

SaaS: The user uses the SaaS software over the entire period $[0, 1]$. The base utility is $\int_0^1 (\theta q + \alpha t - p_s) dt$.

Network effect in the software market is well documented in the literature (Choi 1994; Jing 2007). A user obtains higher utility when more others adopt the same or compatible software. We assume that the SaaS is not compatible with either version of the perpetual software and that the new version perpetual software is backward-compatible with the old version. In our context, the network utility a user derives in the time period $[t_1, t_2]$ is expressed as $\int_{t_1}^{t_2} k n_t dt$, where k is the coefficient to measure marginal network effect, and n_t , where $n_t = 1$ or 2 , is the network size of that software product at time t .

Table 1 presents the strategy matrix. The rows show the OG users' strategies, and the columns show the NG users' strategies. For example, row Upgrade + SaaS, column SaaS, can be understood as the situation in which the OG users upgrade to the new perpetual software for a period of time (until t_{s2}) and switch to the SaaS afterward, while the NG users adopt SaaS over the entire period $[0, 1]$. In this case, OG users' total utility is $(\rho q + k)t_{s2} - p_u +$

$\int_{t_{s2}}^1 (\theta q + \alpha t + 2k - p_s) dt$ and NG users' total utility is $\int_0^1 (\theta q + \alpha t - p_s) dt + [k t_{s2} + 2k(1 - t_{s2})]$. Because the two groups of users adopt incompatible software in $[0, t_{s2}]$, the network size for each group is 1, and each group enjoys a network value k over this time period. After OG users switch to SaaS at time t_{s2} , both groups of users are using the same software, and so both are better off, deriving $2k$ network value over the period $[t_{s2}, 1]$.

In Table 1, we see in total $5 \times 3 = 15$ strategy pairs (SPs). We can safely eliminate some of them from the vendors' profit maximization perspective. For example, vendors would never opt for the strategy pairs in the first row, marked with "Φ." Instead of leaving the OG users completely out of the market and earning zero profit from this group of users, both vendors would compete to serve the OG users by offering them a competitive price. Thus, in equilibrium, OG users are induced to either upgrade to the new perpetual software or to adopt SaaS, so they will not undertake the action "Old."⁶ Similarly, those marked with "Φ" in the last column are overridden because the perpetual software vendor does not make any profit from any group of users in these scenarios; again, they cannot emerge as equilibrium user strategies under the vendor's optimal pricing decision.

⁶Note that we do not directly eliminate "Old+SaaS" here because OG users are served by the SaaS vendor under this strategy, even though we show in a later section that this strategy is not an equilibrium strategy.

Table 1. OG and NG Users' Feasible Strategy Pairs and Corresponding Outcomes

User Strategies	New	New+SaaS	SaaS
Old	Φ	Φ	Φ
Old+SaaS	SP3	SP5	Φ
SaaS	NA	NA	Φ
Upgrade	SP1	NA	SP2
Upgrade+SaaS	NA	SP6	SP4

Also note that four of the strategy pairs are marked with "NA." They are inconsistent among the two groups of users' actions. For example, row SaaS, column New, states that OG users opt for SaaS, while NG users buy the new perpetual software. This option is impossible because the upgrade price (for OG users) should always be lower than or equal to the new purchase price (for NG users). As a result, if the OG users prefer SaaS rather than upgrading to the new perpetual software because the upgrade price is not attractive, then the NG users also prefer SaaS rather than paying a possibly higher price to buy the new perpetual software. Similarly, other NA strategy pairs can be proven infeasible in equilibrium (see Appendix B). Finally, we are left with six SPs, and we number them from SP1 to SP6. Figure C1 and Table C1 in Appendix C provide detailed parameter configurations under which each of the six SPs might appear as equilibrium outcomes.

Especially worthy of mentioning is that both groups of users have the switching option. They can choose to use the perpetual software first and then switch to SaaS at a later time. SP1 and SP2 do not involve user switching; SP3 and SP4 involve switching by only one group of users; and SP5 and SP6 involve switching by both groups of users. Multiple factors play a role in a user's switching decision, including how fast the SaaS quality increases, the SaaS vendor's price, which version of the perpetual software the user currently uses, the expected strategy of other users, and how the network utility before and after switching would change. Overall, users switch if their expected net utility from the SaaS is higher than the utility from the perpetual software that they currently use. Switching time needs to be solved endogenously if switching does happen. For example, the OG users in SP4 switch to the SaaS when the net utility of adopting SaaS equals the utility from continuing to use the new perpetual software, given that the NG users adopt SaaS. The switching time is determined by solving for the equality,

$$\theta q + \alpha_{s2} + 2k - p_s = \rho q + k, \text{ so that } t_{s2}^{SP4} = \frac{p_s + (\rho - \theta)q - k}{\alpha}.$$

Vendor Pricing Strategies and Equilibrium Analysis

We now analyze the vendors' optimal pricing strategies and study the consequent equilibrium outcomes. To help us understand the effect of competition, we take as the benchmark the case in which no threat of SaaS entry arises. We then analyze the competitive outcomes when the SaaS vendor has both a low and a high quality improvement rate α . Finally, we discuss the influence of key factors on the different types of market equilibria.

The Monopoly Benchmark

The monopoly benchmark analysis shows the perpetual software vendor's pricing decisions for its new version software, upgrade price p_u and purchase price p_n , without the competitive pressure of SaaS's entry into the market. We see that the vendor should always set its prices in a way that induces OG users to upgrade and NG users to buy the new version. Because both user groups adopt the perpetual software, the network size is 2. The total utility for OG users is $\rho q + 2k - p_u$ and for NG users is $\rho q + 2k - p_n$. We have the following result:

Proposition 1 (Monopoly Market Equilibrium) *Without the SaaS, the perpetual software vendor serves both user groups: The equilibrium user strategy is SP1 (Upgrade, New), where OG users upgrade and NG users buy the new version of the software; the equilibrium upgrade price is $p_u^M = (\rho - 1)q + k$, and the new purchase price is $p_n^M = \rho q + 2k$.*

We see that $p_u^M < p_n^M$, suggesting that the OG users always get a price discount. The vendor's monopoly profit is $\pi^M = (2\rho - 1)q + 3k$. Consumer surplus for the OG and

NG users is $CS_{OG}^M = q + k$ and $CS_{NG}^M = 0$, respectively. The vendor extracts all surplus from NG users while giving OG users a positive surplus that is equal to their reservation utility from using the old perpetual software, which is just enough to induce them to upgrade. Social welfare is the sum of the vendor's profit and the consumer surplus: $SW^M = 2\rho q + 4k$.

Competition When SaaS's Quality Improvement Rate Is Low

When the SaaS's quality improvement rate is low (i.e., $\alpha \leq (\rho - \theta)q$), the SaaS remains an inferior quality product in the entire software life cycle. In this case, once users have adopted the new perpetual software, they would not switch to SaaS.

We find that the network effect critically affects the equilibrium outcome. We identify a threshold value $K_1 = \frac{\alpha - (\rho - 2\theta + 1)q}{2}$. When k exceeds this threshold, the perpetual software vendor chooses to deter the SaaS from entering the market by reducing prices significantly. When k is below this threshold, the perpetual software vendor shares the market with the SaaS vendor, with each one serving different user groups. These findings are summarized in Propositions 2 and 3.

Proposition 2 (Entry Deterrence Equilibrium) *When the network effect is strong enough (i.e., $k > K_1$), the perpetual software vendor deters the SaaS's entry into the market: The equilibrium user strategy is SP1 (Upgrade, New), where OG users upgrade and NG users buy the new version of the software; the equilibrium prices are $p_u^{SP1} = p_n^{SP1} = (\rho - \theta)q - \frac{\alpha}{2} + k$.*

Compared to the monopoly benchmark, we see that the perpetual software vendor has to reduce both the upgrade and purchase prices significantly ($p_u^{SP1} = p_n^{SP1} < p_u^M < p_n^M$) to deter the SaaS entry. This is because the threat of a SaaS entry creates an outside option for users, which limits the perpetual software vendor's ability to set a high price. In addition, the OG users receive no price discount for upgrading. Intuitively, because both groups of users have the same outside option and derive the same network utility $\rho q + 2k$ under SP1 (Upgrade, New), they pay the same price. Facing the competitive pressure from the SaaS, the perpetual software vendor has to pass some of its profit to users. It hence

earns a lower profit ($\pi_{perp}^{SP1} = 2(\rho - \theta)q - \alpha + 2k < \pi^M$), and both groups of users are better off ($CS_{OG}^{SP1} = CS_{NG}^{SP1} = \theta q + k + \frac{\alpha}{2} > CS_{OG}^M > CS_{NG}^M$). Total social welfare remains the same as in the monopoly benchmark ($SW^{SP1} = 2\rho q + 4k = SW^M$) because the two cases lead to the same user strategy pair, SP1 (Upgrade, New), and thus the total value created in the social system is the same.

Proposition 3 (Market Segmentation Equilibrium— α Low) *When the network effect is relatively weak (i.e., $k < K_1$), the perpetual software and SaaS vendors segment the market: The equilibrium user strategy is SP2 (Upgrade, SaaS), where OG users upgrade to the new perpetual software and NG users adopt the SaaS software. The equilibrium prices are $p_u^{SP2} = (\rho - 1)q$, $p_n^{SP2} = (\rho - 1)q + 2k$, and $p_s^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$.*

When the network effect is weak, the perpetual software vendor gives up on deterring the SaaS vendor's entry. Note that the prices under the entry deterrence equilibrium, as given in Proposition 2, decrease in the network effect k . When k drops below the given threshold value K_1 , the entry deterrence strategy becomes too costly. Instead, the perpetual software vendor benefits from giving up NG users to the SaaS vendor. Declining to compete for NG users enables it to raise the upgrade price for OG users, which results in a higher profit than would be possible if the perpetual software vendor tries to deter the SaaS vendor's entry using very low prices.

In the market segmentation equilibrium, the SaaS vendor serves NG users and earns a positive profit $\pi_{SaaS}^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$, and the perpetual software vendor serves OG users and earns $\pi_{perp}^{SP2} = (\rho - 1)q < \pi_{perp}^M$. The two vendors focus on exploiting their respective user groups. This outcome is similar to the monopolistic competition result established in Salop (1979), in which two competing vendors serve only their own local customer segments and thus mitigate the price competition. However, allowing a low-quality SaaS vendor to enter the market is not socially optimal. Compared to the entry deterrence equilibrium, both generations of users are worse off ($CS_{OG}^{SP2} = q < CS_{OG}^{SP1}$ and $CS_{NG}^{SP2} = q + k < CS_{NG}^{SP1}$). Specifically, the NG users are worse off not only because they pay higher prices, but also because they use a lower quality SaaS product. Social welfare is also reduced as a whole

($SW^{SP2} = (\rho + \theta)q + \frac{\alpha}{2} + 2k < SW^{SP1}$). The stronger the network effect, the greater is the efficiency loss in social welfare. The insight holds even without considering the network effect (i.e., when $k = 0$). Moreover, social welfare in this equilibrium would be lower than the monopoly benchmark ($SW^{SP2} < SW^M$). Figure F2 in Appendix F shows that the consumer surplus registers a significant drop when the market moves from the entry deterrence equilibrium to the market segmentation equilibrium, and that total social welfare is even lower than the monopoly benchmark. These results suggest that allowing the SaaS vendor to enter the market is socially inefficient. Formally, we have the following result:

Corollary 1 (Socially Inefficient Competition) *If the SaaS quality is always lower than the new perpetual software, software vendor competition leads to a loss both in consumer surplus and in social welfare when the network effect is relatively weak.*

Competition When SaaS's Quality Improvement Rate Is High

When SaaS's quality improves at a rapid rate (i.e., $\alpha > (\rho - \theta)q$), the SaaS software eventually outperforms the new perpetual software. In this case, users might switch from the perpetual software to the SaaS at some point to enjoy higher quality, and both vendors engage in a fierce price war.

Because software development costs have been sunk, a vendor can reduce its price to any marginally positive level. Thus, as long as a vendor has the quality advantage over its competitor, it can earn a positive profit by charging a low enough competitive price. This reasoning enables us to eliminate SP1 (Upgrade, New) as an equilibrium, in which the SaaS vendor earns zero profit. Similarly, because the perpetual software vendor has a quality advantage over the SaaS vendor at time 0, it is always able to induce OG users to upgrade and NG users to purchase its new software version by charging low enough prices. This reasoning eliminates SP3, SP4, and SP5 as equilibria. These eliminations allow us to focus on user strategies in SP6 (Upgrade + SaaS, New + SaaS) and SP2 (Upgrade, SaaS). We again find that the equilibrium outcome depends on the network effect, as shown in Propositions 4 and 5.

Proposition 4 (Sequential Dominance Equilibrium) *When the network effect is large enough (i.e., $k > K_2$), the perpetual software and SaaS vendors sequentially dominate the market: The equilibrium user strategy is SP6 (Upgrade + SaaS, NEW + SaaS), where OG users upgrade and NG users purchase the*

new version perpetual software at the beginning, and both switch to the SaaS at $t_{s3}^{SP6} = \frac{\alpha + (\rho - \theta)q}{2\alpha}$. The equilibrium prices are as follows:

(a) *If $(\rho - \theta)q < \alpha \leq (\rho + \theta - 2q)$, then*

$$P_u^{SP6} = P_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha} \text{ and}$$

$$P_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2};$$

(b) *If $\alpha > (\rho + \theta - 2)q$, then*

$$P_u^{SP6} = \frac{k\alpha - (\rho - 1)(\theta - 1)q^2 + [\alpha(\rho - 1) + k(\rho - \theta)]q}{2\alpha},$$

$$P_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}, \text{ and}$$

$$P_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}.$$

The threshold K_2 is given in Appendix E. In this equilibrium, users always choose the vendor that has the quality advantage. Note that once OG users upgrade and NG users buy the perpetual software at time 0, they face the same decision trade-off because payment is sunk. Therefore, they always act in the same way and at the same time. With this knowledge, the SaaS vendor offers a price low enough to induce both groups to switch. As a result, the two vendors take turns serving the whole market—the perpetual software vendor first and the SaaS vendor next; hence, we call it the sequential dominance equilibrium. The SaaS vendor's profit is

$\pi_{SaaS}^{SP6} = \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$. The perpetual software vendor's profit is $\pi_{perp}^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{4\alpha}$ in case (a), which is

higher than $\pi_{perp}^{SP6} = \frac{2[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q] - [\alpha - (\rho + \theta - 2)q]^2}{8\alpha}$

in case (b). Consumer surplus for each user group and total social welfare, under both case (a) and case (b), are reported in Table D1 in Appendix D.

We have several observations to make. First, note that the switching time under SP6 decreases as the SaaS quality improvement rate increases (i.e., $\frac{\partial t_{s3}^{SP6}}{\partial \alpha} < 0$). In addition, the switching time is always greater than $\frac{1}{2}$ (i.e., $t_{s3}^{SP6} > \frac{1}{2}$).

This result implies that, although faster SaaS quality improvement always induces earlier switching, both user groups use the perpetual software through at least half of the software life cycle.

Second, both vendors' equilibrium prices are affected by the SaaS quality improvement rate. A higher quality improvement rate suggests that the SaaS vendor gains the quality advantage more quickly, which thus supports a higher SaaS price p_s^{SP6} . The perpetual software vendor also adjusts its pricing strategy based on the SaaS quality improvement rate. When the SaaS quality improvement rate is relatively small (i.e., case (a)), the SaaS vendor does not create strong competitive pressure. As a result, the perpetual software vendor charges both user generations the same price, $p_u^{SP6} = p_n^{SP6}$, and no price discount is given to OG users. When the SaaS quality improvement rate is relatively large (i.e., case (b)), the SaaS vendor imposes high competitive pressure, and the perpetual software vendor has to offer a price discount to OG users to induce them to upgrade at time 0. Therefore, SaaS quality improvement benefits OG users more than NG users.

Such a sequential dominance equilibrium appears only when the network effect is strong enough. When k is smaller than the threshold K_2 , a market segmentation equilibrium emerges.

Proposition 5 (Market Segmentation Equilibrium— α High) *When the network effect is weak (i.e., $k \leq K_2$), the perpetual software and SaaS vendors segment the market: The equilibrium user strategy is SP2 (Upgrade, SaaS), where OG users upgrade to the new perpetual software while NG users adopt the SaaS software. The equilibrium prices are as follows:*

- (a) If $(\rho - \theta)q < \alpha \leq 2(\rho - 1)q$, then $p_u^{SP2} = (\rho - 1)q$, $p_n^{SP2} = (\rho - 1)q + 2k$, and $p_s^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$;
- (b) If $\alpha > 2(\rho - 1)q$, then $p_u^{SP2} = (\rho - 1)q$, $p_n^{SP2} = \frac{\alpha}{2} + 2k$, and $p_s^{SP2} = k + \alpha - (\rho - \theta)q$.

The two vendors' profits are $\pi_{perp}^{SP2} = (\rho - 1)q$ and $\pi_{SaaS}^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$ in case (a) and $\pi_{SaaS}^{SP2} = k + \alpha - (\rho - \theta)q$ in case (b). The corresponding consumer surplus and social welfare are reported in Table D1 in Appendix D.

In this equilibrium, no users switch during the entire software life cycle. Because of the use of incompatible software products, both OG and NG users can enjoy the network utility only from their own generation. This situation is similar to the equilibrium in Proposition 3 for the low α scenario: Again, it is a monopolistic competition outcome. Two vendors serve the two user groups separately, extracting as much

consumer surplus as possible in their respective market segment. The perpetual software vendor charges OG users a high upgrade price, giving OG users the same level of utility as in the monopoly benchmark. Similarly, the SaaS vendor sets prices aggressively to extract surplus from NG users.⁷ Moreover, as the SaaS quality improvement rate increases, the SaaS vendor charges a higher price to extract more surplus, resulting in a lower consumer surplus for NG users, even though they are using a higher quality software product. The benefits from the SaaS quality improvement therefore are not shared by users.

This insight compares directly with the insight revealed in the sequential dominance equilibrium, in which synergy is created when both groups of users adopt the same software product. The additional utility generated from the network effect increases consumers' willingness to pay, enabling both vendors to charge higher prices and earn higher profits. As a result, all parties are better off.

Summary of Equilibria and Comparative Statics

Figure 2 shows how equilibrium outcomes change under different combinations of the SaaS quality improvement rate and the network effect. Mathematical expressions for $\underline{\alpha}$, K_1 , and K_2 are presented in Appendix E. The solid lines in Figure 2 define three regions for three types of equilibria.

Type I—the entry deterrence equilibrium—appears in Regime I, when the SaaS software is always inferior to the new perpetual software during the entire product life cycle ($\alpha \leq (\rho - \theta)q$) and when the use of compatible software generates high network utility for users ($k > K_1$). This equilibrium is described in Proposition 2. The perpetual software vendor blocks the SaaS vendor's entry via a low-price strategy.

Type II—the market segmentation equilibrium—appears in Regime II, when the SaaS vendor's quality improvement rate is relatively large ($\alpha > \underline{\alpha}$) and when the network effect is small enough ($k < K_1$ and $k < K_2$). The SaaS quality might or might not exceed that of the new perpetual software at the end of the software life cycle. Each vendor serves only one user gener-

⁷The perpetual software vendor deliberately gives up competing for NG users with the SaaS by charging a high new purchase price p_n^{SP2} . The high p_n^{SP2} enables the SaaS vendor to charge a high equilibrium SaaS price p_s^{SP2} , which in turn makes upgrading a more attractive option for OG users than SaaS. Thus the perpetual software vendor can charge a high equilibrium upgrade price p_u^{SP2} .

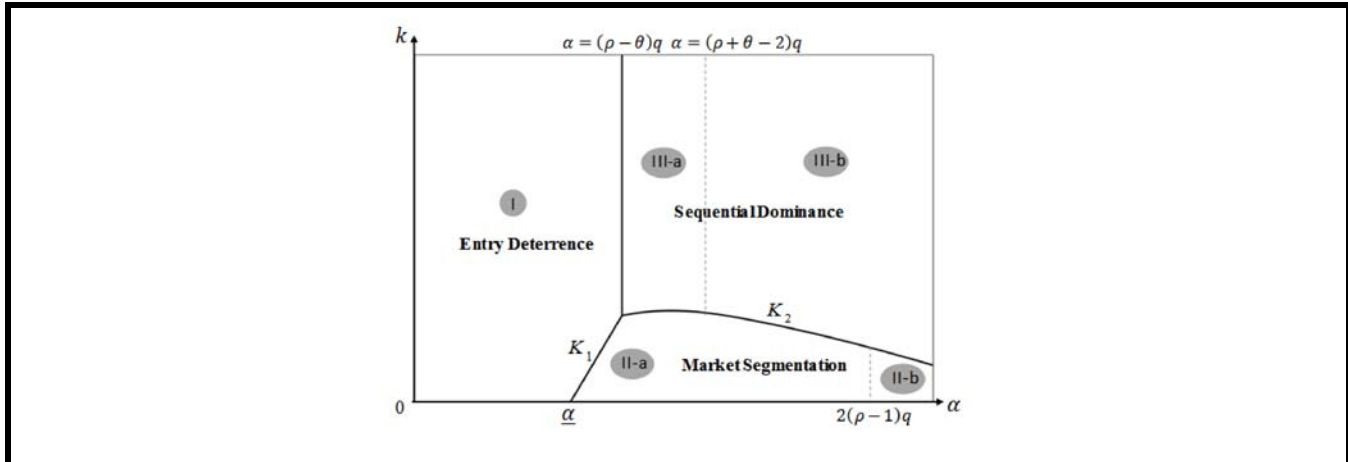


Figure 2. Impact of Network Effect and SaaS Quality Improvement Rate on Equilibria

ation—the perpetual software vendor serves OG users, and the SaaS vendor serves NG users. The dashed line $\alpha = 2(\rho - 1)q$ further divides Regime II into two parts: II-a and II-b, which are described in Propositions 3 and 5, respectively. Note that the area in II-a to the left of the vertical line $\alpha = (\rho - \theta)q$ is the socially inefficient competition regime, as described in Corollary 1. In this regime, the SaaS quality is always lower than that of the incumbent perpetual software vendor; thus, its entry to the market reduces both consumer surplus and social welfare. Also note that this result stems from the unique feature of our model in which the SaaS quality improves continuously over time. As Figure 2 shows, if continuous quality improvement (i.e., $\alpha = 0$) is absent, entry of the lower-quality SaaS will always be deterred. Furthermore, when the network effect is strong enough (i.e., $k \geq K_1$), this socially inefficient outcome is not likely to occur.

Type III—the sequential dominance equilibrium—appears in Regime III, when the SaaS vendor's quality improvement rate is high enough ($\alpha > (\rho - \theta)q$) to exceed the perpetual software quality at some point during the software life cycle, and when the network effect is strong enough ($k \geq K_2$) that the additional utility from using compatible software encourages the two groups of users to act in the same way. The dashed line $\alpha = (\rho + \theta - 2)q$ further divides Regime III into two parts, III-a and III-b, corresponding to the two cases presented in Propositions 4(a) and 4(b), respectively. Under this equilibrium, user switching occurs, and each vendor thus serves both user generations in a sequential manner. The difference between III-a and III-b is the perpetual vendor's pricing strategy: The perpetual software vendor gives OG users a price discount for upgrading in III-b but not in III-a.

Herd behavior and bandwagon pressure have been well recognized in the literature in the study of IT adoption and the

diffusion of innovations across organizations. In reality, innovative user organizations might choose to adopt the SaaS and become early adopters in the market. These early adopters might influence peer organizations, and organizations might imitate each other to adopt SaaS. A “tipping point” often emerges—the point at which a trend catches fire in the population. To some extent this phenomenon can be supported by the sequential dominance equilibrium in our extended model with continuous user arrival (please see Appendix K), where we find that OG users and NG users who arrive before a certain time in the market become early adopters of SaaS, and NG users who arrive thereafter follow the trend to adopt SaaS.

The equilibrium outcome critically depends on both the SaaS vendor's quality improvement rate α and the marginal network effect k . Tables F1 and F2 in Appendix F show the effect of these two key parameters on the vendors' equilibrium prices, profits, consumer surplus, and social welfare. We summarize the main insights in the following paragraphs.

Corollary 2 (*Non-monotonic Impact of α*) *As α increases, the benefits from quality improvement are dispersed as follows: (a) to both generations of users only under the entry deterrence equilibrium; (b) to the SaaS vendor only under the market segmentation equilibrium; and (c) to both vendors and the new generation users when α is low, and to the SaaS vendor and both generations of users when α is high under the sequential dominance equilibrium.*

This finding is illustrated in Figures F1 and F2 in Appendix F. The SaaS vendor's quality improvement rate can be viewed as the indicator of its competitive power. The perpetual software vendor adopts different pricing strategies to cope with a new SaaS rival based on the rival's competi-

tiveness. When α is small, the incumbent perpetual software vendor reduces both upgrade and purchase prices to block the SaaS vendor's entry, resulting in both a lower profit for itself and higher consumer surplus. This outcome suggests that the threat of entrance by a potential competitor benefits customers. As α increases, the SaaS vendor becomes more competitive, and deterring its entry is too costly. (This outcome occurs in the range $\alpha > \underline{\alpha}$.) The perpetual software vendor prefers to increase its prices so that it gives up NG users to the SaaS vendor and focuses on serving the OG users only. Conventional wisdom suggests that users are better off when competition exists. However, in this scenario, users are worse off when the two vendors segment the market. When α increases further, the SaaS gets highly competitive, and the perpetual software vendor changes its pricing strategy again. It reduces prices to compete with the SaaS vendor effectively. The head-to-head price war and the relatively high SaaS quality benefit users. Consumer surplus for both user groups increases, as does the social welfare. This finding demonstrates the positive effect of competition.

Different market participants prefer a distinct SaaS quality improvement rate. Not surprisingly, the perpetual software vendor prefers slow SaaS quality improvement, so that the new rival is weak and its entry can be blocked at a low cost. In contrast, the SaaS vendor prefers to have its quality improvement rate in the middle range—not so low that its entry will be deterred, and also not so high that it will have to compete directly with the perpetual software. The preferred outcome for the SaaS vendor is to be in a market segmentation equilibrium so that it can exploit NG users to the maximum degree and can fully capitalize on this market expansion opportunity. We find that under certain circumstances, a SaaS vendor that has a medium quality improvement rate earns a higher profit than a SaaS vendor that has a high quality improvement capability (see the graphical illustration in Figure F1 in Appendix F). This outcome arises because the first SaaS vendor manages to reap all the benefits from continuous quality improvement under the market segmentation equilibrium, while the latter has to pass some benefits of high quality improvement to users to compete effectively with the incumbent under the sequential dominance equilibrium. This interesting observation suggests that sometimes the SaaS vendor might lack the incentive to improve its quality up to the socially optimal level. As a result, a SaaS vendor might not aggressively patch its software product, thus calling into question its promise that it will always deliver the newest software features and advanced technologies to its clients.

The network effect in the underlying software market is the other important factor in determining the two vendors' relative competitive power. As seen in Table F2 in Appendix F and the following corollary, we find that the value from the

positive network externality is shared by different market participants under different equilibria.

Corollary 3 (*Non-monotonic Impact of k*) *As k increases, the benefits from positive network externality are dispersed as follows: (a) to the perpetual software vendor and both generations of users under the entry deterrence equilibrium and the sequential dominance equilibrium; and (b) to the SaaS vendor and OG users only under the market segmentation equilibrium.*

First, under the entry deterrence equilibrium, both the perpetual software vendor and the users benefit from the positive network effect. All else being equal, a stronger network effect (i.e., $k > K_1$) strengthens the perpetual software vendor's entry deterrence capability because users are more likely to opt for compatible software when the network effect gets stronger. As a result, the perpetual software vendor finds that attracting both generations of users is relatively easy. Second, under the market segmentation equilibrium, the perpetual software vendor passes the full network benefit to the OG users to prevent them from continuing to use the old version perpetual software. In contrast, the SaaS vendor fully exploits the NG users and reserves all network benefit for itself. Third, under the sequential dominance equilibrium, a stronger network effect increases users' willingness to act uniformly. Because the perpetual software vendor has an existing customer base and possesses the initial quality advantage, it always is able to attract both generations of users at the beginning and enjoys the first-mover advantage. In this case, the network value is shared by the perpetual software vendor and both user generations, but not by the SaaS vendor.

Perpetual Software Vendor's Discrete Quality Improvement

Our baseline model assumes the constant quality of perpetual software. In practice, the perpetual software vendor commonly improves its quality in two ways: It offers incremental quality improvement through free patching within the product life cycle, and it provides periodic major quality leapfrogs and sells subsequent updated versions to the market. In the following, we study how these incremental and major quality improvement strategies could affect competition dynamics.

Incremental Quality Improvement Through Free Patching

Consider that the perpetual software vendor offers free quality improvement through patching within the software life cycle

$[0, 1]$. The patching occurs at time $t_\delta \in (0, 1)$ and leads to a small-scale quality jump δq , which is significantly smaller than the quality improvement between two major new releases (i.e., $\delta q \ll (\rho - 1)q$). Such patching is especially meaningful when SaaS's quality improves quickly—namely, $\alpha > (\rho - \theta)q$, when the SaaS is expected to exceed the perpetual software at $t^* = \frac{(\rho - \theta)q}{\alpha} \in (0, 1)$ if the perpetual software vendor does not provide the patching.

We consider two different patching strategies based on the time of patching—patching before the SaaS exceeds the perpetual software quality (S1) and patching after the SaaS exceeds the perpetual software quality (S2). Under S1, denoted as $(\delta_1, t_{\delta 1})$ the perpetual software vendor patches its product and offers a free quality jump $\delta_1 q$ at time $t_{\delta 1} < t^*$. Figures 3(a) and 3(b) illustrate two possible scenarios. In the first scenario, the quality jump $\delta_1 q$ enables the perpetual software vendor to retain the quality advantage until the end of the planning horizon. In the second scenario, the SaaS quality improvement rate is high enough that the perpetual software vendor eventually loses its leading position before the end of the planning horizon. Under S2, denoted as $(\delta_2, t_{\delta 2})$, the perpetual software vendor chooses to patch its product and offers a quality jump $\delta_2 q$ at time $t_{\delta 2} > t^*$. It is illustrated in Figure 3(c).

Figure 4 shows the graphical comparison of equilibrium outcomes between the baseline model and the incremental quality improvement model. The solid lines define regions for each market equilibrium in the baseline model while dashed lines define the regions in this extended model. All mathematical derivations and proofs of the equilibria are detailed in Appendix G.

Consistent with the baseline model without patching, we see that the same three types of equilibria appear, except that the equilibrium regions are shifted toward the right. So no qualitative changes occur in the competition outcomes, and all insights from the baseline model still hold. Intuitively, vendors make a trade-off between the patching time t_δ and its magnitude δq . If a vendor takes more time to fix bugs and patches the software at a later time (i.e., a smaller value of $1 - t_\delta$), it can offer a larger quality improvement (i.e., a larger value of δq). We define $V = \delta q(1 - t_\delta)$ to measure the perceived patching value and denote V_{S1} and V_{S2} as the patching value under S1 and S2, respectively. Note that a later patching always delivers higher quality improvement but not necessarily higher perceived patching value.

We are interested in comparing S1 $(\delta_1, t_{\delta 1})$ —the strategy of an earlier and smaller patch—and S2 $(\delta_2, t_{\delta 2})$ —the strategy of a later and bigger patch. Proposition 6 summarizes the pre-

ferred patching strategies for the perpetual software vendor and the corresponding optimal patching time. Overall, the choice of patching strategy (S1 or S2) and optimal patching time depend on both the SaaS quality improvement rate and the perceived patching value.

Proposition 6 (*Patching Strategy and Optimal Patching Time*)

- (a) *If $\alpha < \alpha_1$, the perpetual software vendor is indifferent between S1 and S2, and the optimal patching time might occur either before or after t^* ; the corresponding market equilibrium is either entry deterrence or market segmentation.*
- (b) *If (i) $\alpha_1 < \alpha < \alpha_2$ and $V_{S2} < v_1$, or (ii) $\alpha > \alpha_2$ and $V_{S2} < v_2$, the perpetual software vendor prefers S1, and the optimal patching time occurs before t^* ; the corresponding market equilibrium is sequential dominance.*
- (c) *If (i) $\alpha_1 < \alpha < \alpha_2$ and $V_{S2} > v_1$, or (ii) $\alpha > \alpha_2$ and $V_{S2} > v_2$, the perpetual software vendor prefers S2, and the optimal patching time occurs after t^* ; the corresponding market equilibrium is market segmentation.*

The threshold values α_1 , α_2 , v_1 , and v_2 are in Appendix G. The optimal patching time under each situation is also derived and presented in Appendix G. To conclude, we find that when the SaaS competitiveness is low (i.e., $\alpha < \alpha_1$), the two strategies have the same effect on the vendor's profit, which linearly increases in the perceived patching value. So the choice of optimal patching time becomes straightforward: If late patching can bring a higher patching value, namely, $V_{S2} > V_{S1}$, then the vendor should opt for S2; otherwise, the vendor is better off with S1.

However, when the SaaS competitiveness is high (i.e., $\alpha > \alpha_1$), delaying patching is not always worthwhile. Specifically, when V_{S2} does not exceed the given threshold value, even if the delay does bring users with a higher perceived patching value (i.e., $V_{S2} > V_{S1}$), the vendor's profit is always lower in S2 than in S1. So the vendor always prefers to patch early. The reason is that if the vendor delays patching after t^* , users are not patient enough to wait for it. Instead, they turn to SaaS, which becomes the temporary quality leader in $[t^*, t_{\delta 2}]$. Knowing that users would switch to SaaS before the perpetual software vendor's patching, the SaaS vendor therefore does not respond to the incumbent's patching action, and it keeps its price the same as in the baseline model. As a result, both the vendor's price and profit in S2 are the same as in the baseline model, and they are lower than the price and profit in S1. In such a situation, the perpetual software vendor is always better off adopting S1. As a result, the sequential dominance

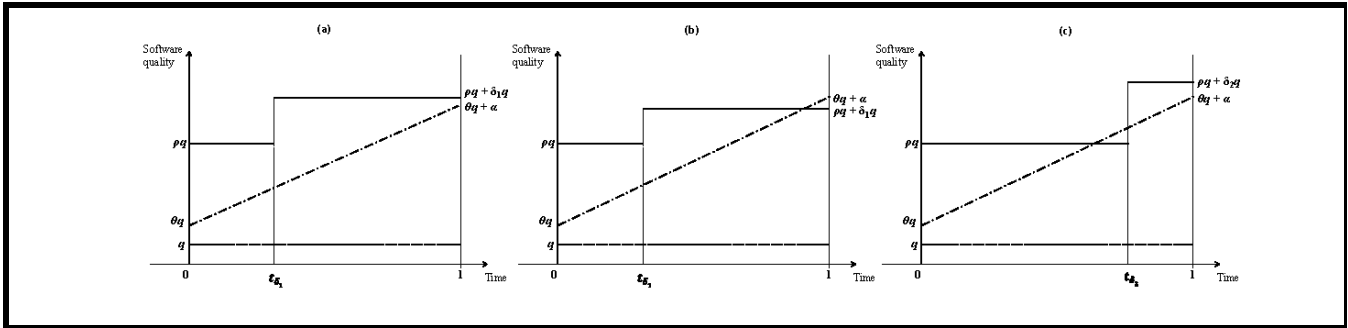


Figure 3. The Perpetual Software Vendor’s Incremental Quality Improvement

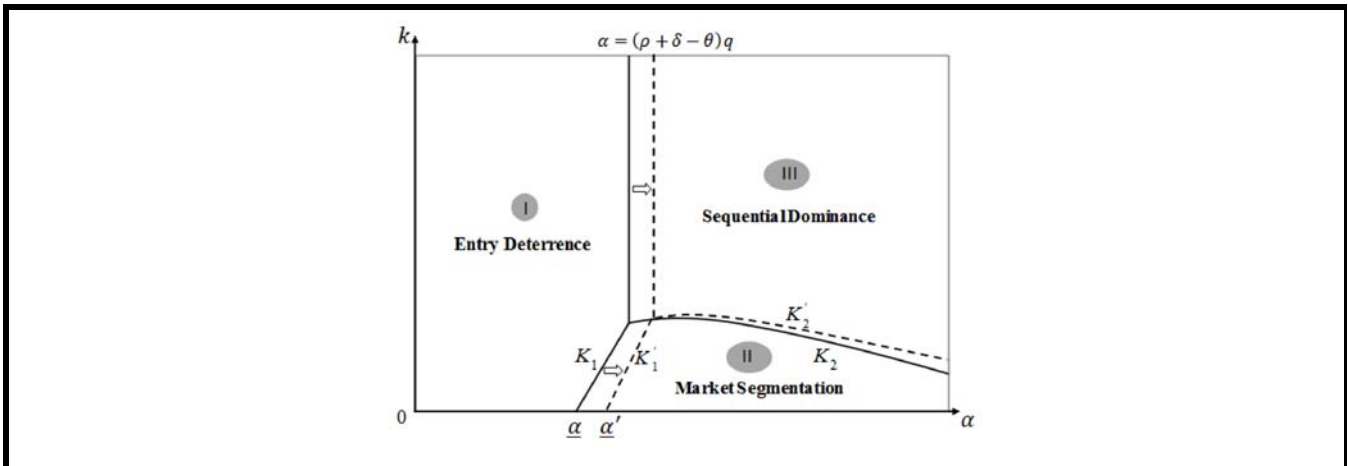


Figure 4. Equilibrium Outcome with Incremental Quality Improvement

equilibrium emerges, in which users’ switching is postponed because of their anticipation of the impending patching. From the users’ perspective, the total value of the perpetual software is enhanced by patching. So the SaaS vendor has to reduce its price in response to the perpetual software vendor’s patching action, which gives the perpetual software vendor more room to increase its profit margin. The perpetual software vendor thus is better off patching before the SaaS exceeds its quality.

In contrast, late patching, in the presence of a strong SaaS entrant, is preferred only if the perceived patching value is sufficiently high, so that users deem it worthwhile to wait for patching despite the fact that SaaS will become the temporary quality leader in $[t^*, t_{\delta 2}]$. The SaaS vendor responds to the perpetual software vendor’s patching action by lowering its price significantly. Consequently, the equilibrium outcome is market segmentation. The SaaS vendor is worse off, and the perpetual software vendor strengthens its competitive position and earns a higher profit than in the baseline model.

Major Quality Improvement Through Consecutive Release

Consider that the perpetual software vendor provides major quality improvement of its software over two periods: the first major release occurs at time zero and spans the first period over $[0,1]$; the second major release occurs at time 1 and spans the second period over $[1,2]$. Consistent with our one-period model, we assume the major quality improvement between two consecutive versions is $(\rho - 1)q$. We further assume that the SaaS does not have absolute quality advantage over the perpetual software in both periods. That is, at the second major release, the perpetual software quality is higher than the SaaS, $(2\rho - 1)q > \theta q + \alpha$; hence, $\alpha \leq (2\rho - \theta - 1)q$.

We extend our baseline model definition of the user strategy pair into a vector that involves a two-period expression. Thus, the one pair becomes double-faceted: [(OG user’s period 1 strategy, OG user’s period 2 strategy), (NG user’s

period 1 strategy, NG user's period 2 strategy)]. We further distinguish the upgrade and new perpetual software purchase strategies in the two periods as follows: Upgrade1 and Upgrade2 denote existing perpetual software users' upgrade strategy in period 1 and period 2, respectively, and New1 and New2 denote new users' strategy of purchasing the perpetual software in period 1 and period 2, respectively. For example, [(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)] indicates that the OG users upgrade to the new version perpetual software at the beginning of period 1, switch to SaaS at some point later, and then switch to the second upgrade of the perpetual software in period 2. The NG users buy the new perpetual software at the beginning of period 1, switch to SaaS at some later time, and then switch to the second upgrade version of perpetual software in period 2.

The number of users' strategy pairs increases significantly compared to the one-period baseline model. We first apply the similar reasoning as in the baseline model analysis to eliminate some of them.⁸ We finally obtain three equilibrium outcomes. The details of derivations and proofs are in Appendix H. Figure 5 compares the equilibrium regions between the two-period model and the baseline model. The solid lines depict the boundaries in the two-period model, while the dashed lines represent the baseline model.

We find that the qualitative insights in the two-period model remain the same as in the baseline one-period model. When the SaaS quality improvement rate is low and the network effect is strong, the perpetual software vendor can deter the SaaS entry. The entry deterrence equilibrium emerges as [(Upgrade1, Upgrade2), (New1, Upgrade2)], where OG users upgrade in both periods and NG users adopt the new perpetual software in the first period and upgrade in the second period. Compared with the one-period model, the region of entry deterrence is pushed down. It suggests that when vendors compete in a longer horizon, it becomes easier to deter the SaaS entry if the SaaS quality improvement is low (i.e., $\alpha \leq (\rho - \theta)q$).

When the SaaS quality improvement rate is high enough and the network effect is not too strong, the market segmentation equilibrium emerges as [(Upgrade1, Upgrade2), (SaaS, SaaS)], where OG users upgrade their perpetual software in both periods and NG users adopt SaaS in both periods. Compared with the one-period model, the region of market

segmentation is pushed up and toward the right. More importantly, when the SaaS possesses the single-period quality improvement advantage over the perpetual software vendor (i.e., $\alpha > (\rho - 1)q$), segmenting the market is more beneficial for the perpetual software vendor than competing directly with the new rival.

Finally, when the SaaS quality improvement rate is high and the network effect is strong enough, the sequential dominance equilibrium can occur. In the two-period setting, we identify two sequential dominance equilibria: [(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)] when $(\rho - \theta)q < \alpha \leq \frac{(3\rho - \theta - 2)q}{3}$ and [(Upgrade1+SaaS, Upgrade2+SaaS), (New1+SaaS, Upgrade2+SaaS)] when $\alpha > \frac{(3\rho - \theta - 2)q}{3}$. In

the first equilibrium, the OG (NG) users upgrade (buy new) perpetual software at the beginning of the first period, switch to SaaS together at a later time in the first period, and upgrade to the perpetual software in the second period. This equilibrium occurs when SaaS quality improvement rate is moderate. Because the quality gap between the two vendors widens over time, the SaaS vendor has a relatively greater quality advantage in the first period than it does in the second period. Thus, it is able to induce users to switch and serves the market in the first period, but it forgoes the opportunity to do so in the second period.

In the second equilibrium, the OG (NG) users upgrade (buy new) perpetual software at the beginning of the first period, switch to SaaS together at a later time in the first period, upgrade to the perpetual software at the time of the second major release, and switch together to SaaS again before the end of the second period. This equilibrium occurs when the SaaS is so competitive that after one period of continuous improvement, its quality is relatively comparable with the perpetual software. The two vendors now engage in fierce price wars, and in each period the vendor that brings higher net utility to users takes over the market.

Other Model Extensions

In this section, we extend the baseline model to consider two types of costs: the SaaS vendor's quality improvement cost and users' switching cost. We examine how they might affect the market outcome. We show that all qualitative insights in our baseline model remain valid, and the results demonstrate the robustness of our model.

⁸For example, [(Upgrade1, Upgrade2), (New1, SaaS)] cannot be a consistent strategy pair because after NG users purchase the perpetual software in the first period, their decision to trade off in the second period becomes the same as that of OG users. We use a similar method to eliminate several other strategy pairs.

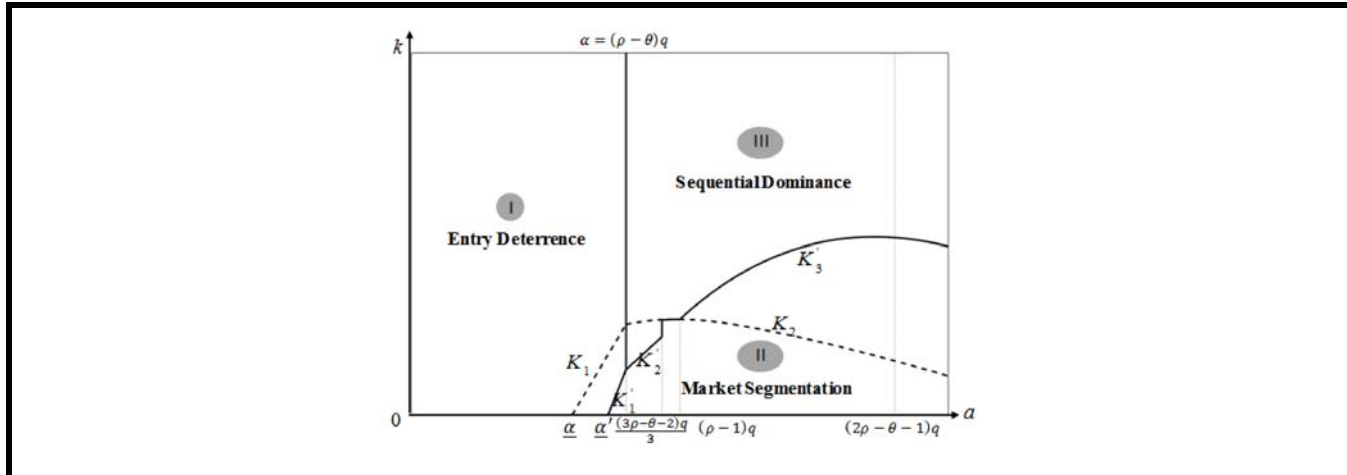


Figure 5. Equilibrium Outcome in a Two-Period Model

SaaS Vendor’s Quality Improvement Cost

We have omitted the SaaS vendor’s cost of improving software quality in our baseline model. In this extension, we include this cost to examine its effect on the results. The continuous quality improvement of SaaS is mainly driven by two facts: The vendor fixes errors and bugs based on clients’ feedback from the software use, and it also applies newly available technologies to offer more features and functionalities as time goes by. This process is different from the R&D process for initial software development, which often requires huge monetary and human capital investments.

Because the SaaS software is centrally maintained and managed by the vendor, the quality improvement cost is a function of the quality improvement rate, rather than the number of adopted users. Therefore, we assume that the SaaS vendor incurs an ongoing quality improvement cost c_α per unit of time, where c_α is an increasing function of α . The total quality improvement cost over the time interval $[0, 1]$ is $\int_0^1 c_\alpha dt = c_\alpha$. We find that including c_α changes the condition under which entry deterrence equilibrium appears, but doing so has no effect on all other equilibrium results. All details related to the new entry deterrence equilibrium are presented in Appendix I. Figure 6 demonstrates how the region of the entry deterrence equilibrium shifts when considering the SaaS quality improvement cost. We use a linear cost function, $c_\alpha = b\alpha$, in this illustration.⁹

⁹Because α is the total value created in the time interval $[0, 1]$ and c_α is the total cost to produce such value, $c_\alpha < \alpha$ is the condition for an efficient production. As a result, $b < 1$. Also note that our result is not limited to the

In Figure 6, the solid lines, $(\rho - \theta)q$ and K_1 , define the border of Region I (entry deterrence equilibrium), as in our baseline model, and the dashed lines, $\frac{(\rho - \theta)q}{1 - b}$ and K'_1 define the border when the SaaS quality improvement cost is considered. Compared with the baseline model, the perpetual software vendor is more likely to block the SaaS vendor’s entry, instead of sharing or segmenting the market with it. Other than this effect, we find no qualitative changes of the insights, compared with the baseline model results. In both regions of the market segmentation and the sequential dominance equilibria, none of the equilibrium prices and vendor profits are affected. Although the cost c_α is in the SaaS vendor’s profit function, it does not affect the vendor’s pricing decision after it enters the market because c_α assembles a fixed cost, instead of a variable cost. The magnitude of the cost is unrelated to the number of SaaS customers.

OG User’s Switching Cost

For users who have already adopted the perpetual software, switching to the SaaS system is similar to IT outsourcing. The user organization must change in many ways to adapt to this new on-demand business model—technologically, organizationally, and economically. This adjustment process might take a long time and involve significant costs, both tangible and intangible—especially for firms that have a long

linear cost function. In the case of nonlinear cost functional forms, the two dashed lines that define the borders of Region I might not be linear, but the same insight still holds.

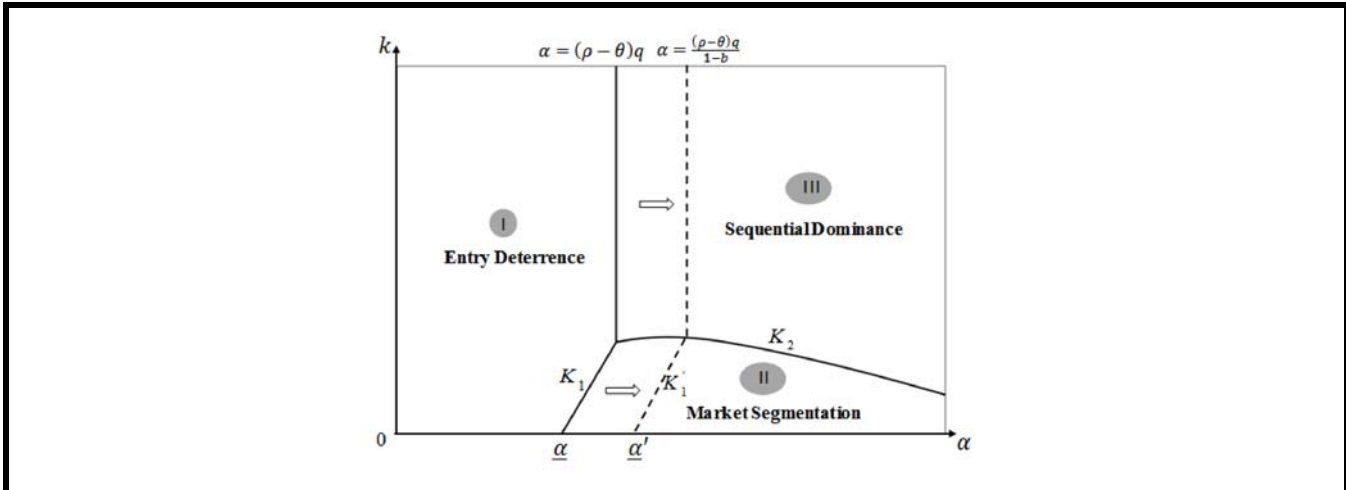


Figure 6. Market Equilibrium with SaaS Quality-Improvement Cost

history of in-house IT experience. To incorporate this reality, we extend our baseline model to include a switching cost for users when they switch from perpetual software to the SaaS. Presumably, firms that have used the traditional in-house software for a longer time face greater challenges because of the technological lock-in by their existing legacy system, which reduces their flexibility and makes moving to the on-demand SaaS model more costly. Without loss of generality, we normalize the switching cost for NG users to be zero. The switching cost for OG users is denoted as c . We assume that no cost is incurred if OG users upgrade from the old to the new perpetual software because of system compatibility. To keep our model tractable and to focus on how the switching cost affects the market competition, we do not consider the network effect in this extension (i.e., $k = 0$). All equilibria and the mathematical proofs are summarized in Appendix J. The four types of equilibria are shown graphically in Figure 7. All the parameter values for A_1 , A_2 , A_3 , C_1 , and C_2 are given in Appendix J.

Compared to the baseline model results, a new type of equilibrium emerges: the *competitive lock-in equilibrium*. Both NG and OG users opt for the new perpetual software at the beginning, but the NG users switch to the SaaS after it takes the quality leader role in the market. The switching time is $t_{SC}^* = \frac{\alpha + (\rho - \theta)q}{2\alpha}$. In contrast, the OG users stay with the perpetual software because they are locked in by switching costs. This equilibrium thus appears only when the SaaS vendor's quality improvement is high enough, $\alpha > A_2$, and when the switching cost for OG users is high enough, $c > C_1$.

We note several interesting observations. First, including the OG users' switching cost does not enhance the perpetual soft-

ware vendor's ability to deter the SaaS rival's entry. We find that the condition for the entry deterrence equilibrium to appear ($\alpha < A_1$) and the optimal prices under this equilibrium are exactly the same as those in the baseline model. Second, considering switching cost creates more room for market segmentation equilibrium to occur. In the presence of OG users' switching cost, the two vendors have the possibility of segmenting the market, even at a very high SaaS quality improvement rate $\alpha \geq A_2$, as long as OG users' switching cost is in a certain range: It is not high enough to support the perpetual software vendor's lock-in capability (i.e., $c < C_1$) and is not low enough to enable both user groups to switch (i.e., $c > C_2$). Finally, we still observe the sequential dominance equilibrium when α is large ($\alpha > A_3$), but it is limited to the low switching cost scenario ($c \leq C_2$).

Conclusion

In this paper we examine the competitive dynamics between an incumbent perpetual software vendor and an SaaS entrant. We find that the incumbent should adopt different strategies to cope with the new rival, based on the SaaS's quality improvement rate and the strength of the network effect. It can adopt an *entry deterrence strategy* to block a weak SaaS vendor with a very low quality improvement rate in the presence of a weak network effect. When the SaaS quality improvement rate is relatively high but the network effect is not so strong, the perpetual software vendor might need to adopt a *market segmentation strategy*, so that it serves existing users and leaves the new users to the SaaS vendor. In the presence of a strong network effect, especially when the SaaS vendor is a strong competitor and its quality improve-

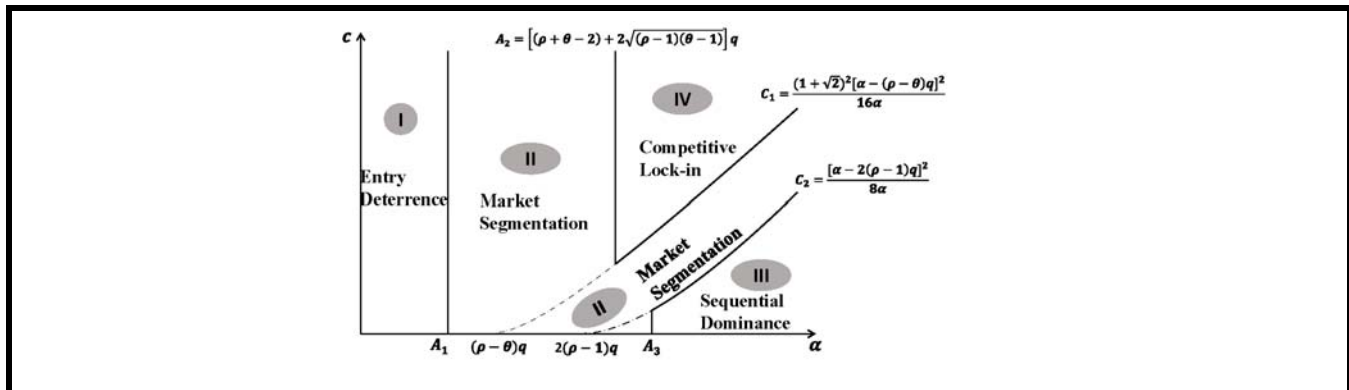


Figure 7. Market Equilibrium with Switching Cost

ment rate is high enough, the perpetual software vendor is better off adopting a *sequential dominance strategy*, by which it aggressively competes with the SaaS vendor. Both the OG and NG users therefore adopt the perpetual software first and then switch to SaaS at a later time.

Our model captures the important features of the SaaS vendor’s pay-as-you-go pricing model and continuous quality improvement pattern, as well as the perpetual software vendor’s one-time pricing method with discrete quality improvement over time. This unique model setup brings several novel insights. First, we find that the SaaS is able to enter the market and make a positive profit even if its quality is inferior to the incumbent perpetual software. This, however, may result in social welfare loss. Second, the possibility that the late entrant becomes the quality leader because its continuous quality improvement leads to the emergence of sequential dominance equilibrium, in which users switch from the incumbent perpetual software vendor to the SaaS vendor. In addition, we suggest that the perpetual software vendor might pursue two quality improvement strategies: It might patch its software with an incremental quality jump, and it might release new versions with major quality leapfrogging. Both serve to defend the incumbent’s market power and strengthen its competitiveness when facing a new SaaS rival. We further make recommendations regarding the vendor’s optimal patching time and identify conditions under which the vendor should patch early in the product life cycle.

Our model and results are supported by empirical observations in the software market. The competition between Blackboard and Canvas is a good business case to demonstrate how a new SaaS entrant can successfully compete with an incumbent. Most users have viewed Blackboard as setting the highest industry standard and as having a near-monopoly in the LSM market for almost 20 years, but after several years

of continuous product improvement, Canvas is considered to be of comparable quality. We have seen Blackboard’s users switch to Canvas, attracted by its in-the-moment cloud computing features, such as mobile access, more up-to-date functionalities, and periodic payment. Although Blackboard remains a major industry force and retains many of its biggest clients, it feels the pressure to defend its position by employing an appropriate quality improvement strategy—for example, offering appealing cutting-edge technologies and new features, such as mobile access capability.

The diversity of the software market structure observed in the real world is also consistent with our analytical findings. For example, Salesforce—the pure SaaS-based customer relationship management (CRM) software company that entered the market in 1999—has been very successful in its competition with the traditional on-premise incumbents, such as SAP, Microsoft, and IBM (Columbus 2014). Many large enterprise users, including Amazon, Morgan Stanley, Nokia, Staples, Target, and Merrill Lynch, have switched from the on-premise CRM system to Salesforce, consistent with our model’s prediction of user switching behavior under strong SaaS competition. However, in some niche markets, new SaaS rivals have experienced failure. For example, new SaaS vendors have rarely succeeded in the enterprise resource planning (ERP) marketplace. Only in the past five years have we started to see existing perpetual software vendors, such as SAP and Oracle, adjusting their business model to offer cloud-based ERP—mainly to complement rather than to substitute for the on-premise ERP (Williams 2011). In this niche, we have seen something more like an entry deterrence relationship between perpetual software vendors and new SaaS-based companies. In fact, we find that when the SaaS quality improvement rate is not high and the network effect is relatively weak, deterring the new SaaS vendor’s entry into the market is socially optimal.

We have generalized our model to consider users' switching cost and the SaaS vendor's ongoing cost of quality improvement, and our main insights from the baseline model are robust. In addition, we suggest that although the SaaS has been a serious threat, the perpetual software vendor, as the incumbent, still has competitive advantages in various ways. For example, many enterprise users have recognized that moving to SaaS requires a great deal of additional effort, including IT governance policies and operating model changes, new types of vendor relationship management (Hsu et al. 2014), and hidden interoperability issues associated with legacy system integration (Ma and Seidmann 2015). These challenges constitute switching costs for users, and we show that, in a competitive situation, they help the perpetual software vendor to preserve its existing market base.

We see several directions for future research. First, we have assumed constant pricing in this paper. Future research might study the vendor's dynamic pricing strategy in a multi-period setting. For instance, software vendors might consider offering a low price to attract new users when the installed user base is small and then raising the price when the user base becomes large enough. Second, in recent years, we have observed that many existing perpetual software vendors are offering SaaS-based versions of their products, too. The SaaS-based and on-premise versions are used together as complements for the vendor to reach broader market coverage. This hybrid strategy has not been explored in this study. It definitely offers a promising direction for future research. Finally, our model does not endogenize vendors' quality decisions. In reality, quality decisions can be made either simultaneously with price decisions or before price decisions are made. In the latter case, a sequential-move game should be solved, which is beyond the scope of our current analysis. We also do not consider software research and development (R&D) costs. Following the literature, the costs might be modeled as a convex function of quality. Taking into consideration both software quality decision and initial R&D cost offers an intriguing avenue for future work.

References

- Arora, A., Caulkins, P. J., and Telang, R. 2006. "Research Note—Sell First, Fix Later: Impact of Patching on Software Quality," *Management Science* (52:3), pp. 465-471.
- August, T., Bucykescym M. F., and Shin, H. 2014. "Cloud Implication on Software Network Structure and Security Risks," *Information Systems Research* (25:3), pp. 489-510.
- Babcok, C. 2013. "Amazon Cuts Cloud Prices Again," *Information Week*, July 10.
- Bala, R., and Carr, S. 2009. "Pricing Software Upgrades: The Role of Product Improvement and User Costs," *Production and Operations Management* (18:5), pp. 560-580.
- Bass, F. M. 1969. "A New Product Growth Model for Consumer Durables," *Management Science* (15:1), pp. 215-227.
- Bhargava, H., and Choudhary, V. 2001. "Information Goods and Vertical Differentiation," *Journal of Management Information Systems* (18:2), pp. 85-102.
- Bogage, J. 2015. "Blackboard Loses High-Profile Clients as its Rivals School it in Innovation," *The Washington Post*, August 22.
- Cavusoglu H., Cavusoglu, H., and Zhang, J. 2008. "Security Patch Management: Share the Burden or Share the Damage?," *Management Science* (54:4), pp. 657-670.
- Chen, P., and Wu, S. 2013. "The Impact and Implications of On-Demand Services on Market Structure," *Information Systems Research* (24:3), pp. 750-767.
- Choi, J. P. 1994. "Network Externality, Compatibility Choice, and Planned Obsolescence," *The Journal of Industrial Economics* (42:2), pp. 167-182.
- Choudhary, V. 2007. "Comparison of Software Quality under Perpetual Licensing and Software as a Service," *Journal of Management Information Systems* (24:2), pp. 141-165.
- Columbus, L. 2013. "Gartner Predicts Infrastructure Services Will Accelerate Cloud Computing growth," *Forbes*, February 8.
- Columbus, L. 2014. "Gartner CRM Market Share Update: 41% of CRM Systems are SaaS-Based, Salesforce Dominating Market Growth," *Forbes*, May 6.
- Ellison, G., and Fudenberg, D. 2000. "The Neo-Luddite's Lament: Excessive Upgrades in the Software Industry," *Rand Journal of Economics* (31:2), pp. 253-272.
- Fan, M., Kumar, S., and Whinston, A. B. 2009. "Short-Term and Long-Term Competition Between Providers of Shrink-Wrap Software and Software as a Service," *European Journal of Operational Research* (196:2), pp. 661-671.
- Gartner. 2017. "Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017," Press Release, February 22, Gartner, Stamford, CT (<http://www.gartner.com/newsroom/id/3616417>).
- Hotelling, H. 1929. "Stability in Competition," *Economic Journal* (39:153), pp. 41-57.
- Hsu, P.F., Ray, S., and Li-Hsieh, Y. Y. 2014. "Examining Cloud Computing Adoption Intention, Pricing Mechanism, and Deployment Models," *International Journal of Information Management* (34:4), pp. 474-488.
- Huang, K. W., and Sundararajan, A. 2011. "Pricing Digital Goods: discontinuous Costs and Shared Infrastructure," *Information Systems Research* (22:4), pp. 721-738.
- Huang, J., Kauffman, R. J., and Ma, D. 2015. "Pricing Strategy for Cloud Computing: A Damaged Services Perspective," *Decision Support Systems* (78), pp. 80-92.
- Jing, B. 2007. "Network Externalities and Market Segmentation in a Monopoly," *Economics Letters* (95:1), pp. 7-13.
- Katz, M. L., and Shapiro, C. 1985. "Network Externalities, Competition, and Compatibility," *The American Economic Review* (75:3), pp. 424-440.
- Kim, B. C., Chen, P., and Mukhopadhyay, T. 2010. "An Economic Analysis of the Software Market with a Risk-Sharing Contract," *International Journal of Electronic Commerce* (14:2), pp. 7-39.

- Kreps, D. M. 1977. "A Note on Fulfilled Expectations Equilibria," *Journal of Economic Theory* (14:1), pp. 32-43.
- Lambertini, L., and Tampieri, A. 2012. "Low-Quality Leadership in a Vertically Differentiated Duopoly with Cournot Competition," *Economics Letters* (115:3), pp. 396-398.
- Ma, D., and Kauffman, J. R. 2014. "Competition Between Software-as-a-Service Vendors," *IEEE Transaction on Engineering Management* (61:4), pp. 717-729.
- Ma, D., and Seidmann, A. 2015. "Analyzing Software as a Service with Per-Transaction Charges," *Information Systems Research* (26:2), pp. 360-378.
- Mehra, A., Seidmann, A., and Mojumder, P. 2014. "Product Life-Cycle Management of Packaged Software," *Production and Operations Management* (23:3), pp. 366-378.
- Moorthy, K. S. 1984. "Market Segmentation, Self-Selection, and Product Line Design," *Marketing Science* (3:4), pp. 288-307.
- Norton, J., and Bass, F. M. 1987. "A Diffusion Theory of Adoption and Substitution for Successive Generations of High Technology Products," *Management Science* (33:9), pp. 1069-1086.
- Norton, J., and Bass, F. M. 1992. "Evaluation of Technological Generations: The Law of Capture," *Sloan Management Review* (33:2), pp. 66-77.
- Salop, S. C. 1979. "Monopolistic Competition with Outside Goods," *The Bell Journal of Economics* (10:1), pp. 141-156.
- Shaked, A., and Sutton, J. 1983. "Natural Oligopolies," *Econometrica* (51:5), pp. 1469-1484.
- Stokey, N. 1981. "Rational Expectations and Durable Goods Pricing," *The Bell Journal of Economics* (12:1), pp. 112-128.
- Sullivan, C., Konary, A., Webster, M., and Mahowald, R. 2012. "The Price Is Right? Microsoft Reduced Pricing on Office 365 and Azure Cloud Services," *IDC Link*, March 30.
- Susarla, A., Barua, A., and Whinston, A. B. 2009. "A Transaction Cost Perspective of Software as a Service Business Model," *Journal of Management Information Systems* (26:2), pp. 205-240.
- Tirole, Y. 1992. *The Theory of Industrial Organization*, Cambridge, MA: The MIT Press.
- Vandenbosch, M. B., and Weinberg, C. B. 1995. "Product and Price Competition in a Two-Dimensional Vertical Differentiation Model," *Marketing Science* (14:2), pp. 224-249.
- Williams, J. 2011. "Will Businesses Opt for SaaS in Next ERP Lifecycle?," *Computer Weekly*, June.
- Zhang, J., and Seidmann, A. 2009. "Perpetual Licensing vs. Subscription of Software: A Theoretical Evaluation," in *Proceedings of the 42nd Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE Computer Society Press.

About the Authors

Zhiling Guo is an associate professor of Information Systems at Singapore Management University. She received her Ph.D. from The University of Texas at Austin. Her recent research focuses on innovative business models for digital payments, resource allocation and management in cloud computing, and data analytics to understand consumer decision making. She has published papers in leading business and management journals, including *Management Science*, *Information Systems Research*, *Journal of Management Information Systems*, *Production and Operations Management*, among others.

Dan Ma is an associate professor of Information Systems at Singapore Management University. She obtained her Ph.D. degree in Computers and Information Systems from Simon School of Business at University of Rochester. She studies the economics of Cloud computing, Software-as-a-Service business model analysis, pricing information goods, competitive strategy, and the decision-making of new technology adoption. Her works have been appeared in research journals including *Information Systems Research*, *Journal of Management Information Systems*, and *IEEE Transactions on Engineering Management*.

A MODEL OF COMPETITION BETWEEN PERPETUAL SOFTWARE AND SOFTWARE AS A SERVICE

Zhiling Guo and Dan Ma

School of Information Systems, Singapore Management University, 80 Stamford Road,
Singapore 178902 SINGAPORE {zhilingguo@smu.edu.sg} {madan@smu.edu.sg}

Appendix A

Modeling Notations

Notation	Definition
$t \in [0, 1]$	Time within the software life cycle [0,1]
q	Quality of the old perpetual software product
ρ	New perpetual software quality improvement ratio over the old version
θ	The SaaS initial quality improvement ratio over the old perpetual software, $1 < \theta < \rho$
α	Rate of software quality improvement for the SaaS product
p_u	One-time upgrade price for existing users to upgrade to the new perpetual software
p_n	One-time purchase price for new users to buy the new perpetual software
p_s	The SaaS price for per unit time use of the software
n_t	The network size at time t , where $n_t = \{1, 2\}$
k	Marginal network effect
δ	Perpetual software incremental quality improvement ratio over the old version
c_a	The SaaS vendor's quality improvement cost per unit time
c	OG users' cost of switching to SaaS

Appendix B

Elimination of Strategy Pairs in Table 1

Given the software quality improvement $\rho q > q$, the OG consumers are willing to pay a positive price to upgrade to the new perpetual software. Because all software development costs have been sunk, the perpetual software vendor can always sell to the OG users at a positive price to earn non-zero profit. So in equilibrium, any strategy pair that involves the OG users that continue to use the old version of perpetual software is dominated by other induced user strategies. We therefore eliminate the first row of strategy pairs in Table 1.

Similarly, (Old + SaaS, SaaS) and (SaaS, SaaS) can be eliminated because the perpetual software vendor earns zero profit. Because the perpetual software has the quality advantage over the SaaS at time 0, the perpetual software vendor, by charging a very small positive upgrade price ε , is able to induce the OG consumers to upgrade and earn a non-zero profit.

Also note that if the OG users choose SaaS, the NG users prefer SaaS as well. The reason is that the OG users are more “sticky” to the perpetual software than the NG users because of their reserve utility from the old perpetual software. Therefore, neither (SaaS, New) nor (SaaS, New + SaaS) can achieve and sustain equilibrium.

Finally, once both OG and NG users adopt the new version perpetual software, they become identical. They should take the same action afterward—either they both continue to use the new version or they switch to SaaS at some time point simultaneously. This rules out (Upgrade, New + SaaS) and (Upgrade + SaaS, New). As a result, only six strategy pairs, SP1 ~ SP6, are possible in equilibrium.

Appendix C

Parameter Configuration for Strategy Pairs SP1 ~ SP6

Figure C1 graphically shows how the six possible strategy pairs can be supported by different combinations of the SaaS quality improvement rate and the SaaS price. The parameter configurations for each strategy pair are presented in Table C1. We observe that the network effect will affect the appearance of SP2, SP4, and SP5. When the network effect is stronger, users tend to choose the same type of software; that is, when the dashed line in Figure C1 shifts up to the left, the appearance of SP2 becomes less likely, while that of E4 and E5 becomes more likely.

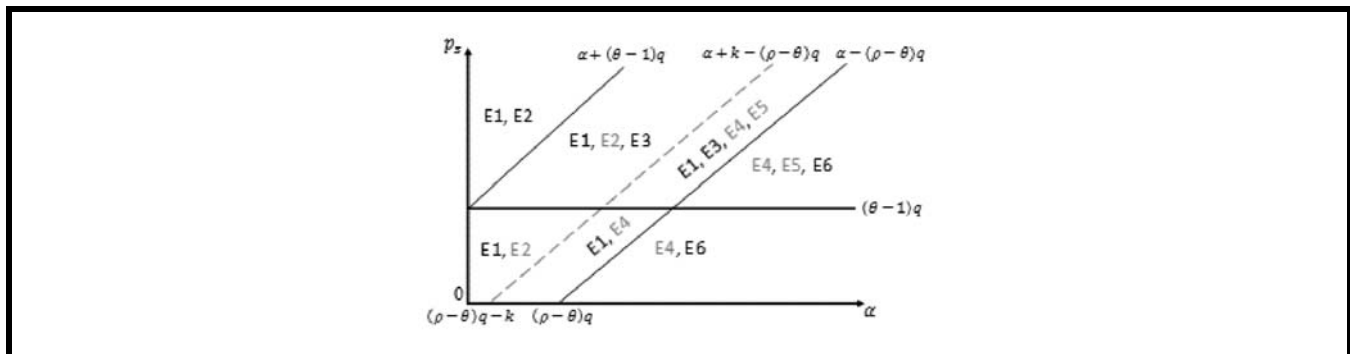


Figure C1. Possible Outcomes and Feasible Regions

Table C1. Parameter Configuration for Each Strategy Pair

Strategy Pair	Feasible Conditions
SP1 (Upgrade, New)	$p_s \geq \alpha - (\rho - \theta)q$
SP2 (Upgrade, SaaS)	$p_s \geq \alpha + k - (\rho - \theta)q$
SP3 (Old+SaaS, New)	$\max[(\theta - 1)q, \alpha + k - (\rho - \theta)q] \leq p_s \leq \alpha + (\theta - 1)q$
SP4 (Upgrade+SaaS, SaaS)	$p_s \leq \alpha + k - (\rho - \theta)q$
SP5 (Old+SaaS, New+SaaS)	$(\theta - 1)q \leq p_s \leq \alpha + k - (\rho - \theta)q$
SP6 (Upgrade+SaaS, New+SaaS)	$p_s \leq \alpha - (\rho - \theta)q$

SP1: Because both groups adopt the new perpetual software, they are identical after adoption. In SP1, no groups switch to SaaS over the entire software life cycle, implying that the SaaS payoff at the end of the software life cycle is no higher than the new perpetual software. Hence, $\theta q + \alpha + 2k - p_s \leq \rho q + 2k$, which leads to $p_s \geq \alpha - (\rho - \theta)q$.

SP2: To prevent the OG users from switching to SaaS, the SaaS payoff at the end of the software life cycle should not be higher than payoff from the new perpetual software for OG users. Note that, without switching, the OG users derive the network utility k ; if switching, they can enjoy the network utility $2k$ because the NG users have adopted SaaS. Hence, $\theta q + \alpha + 2k - p_s \leq \rho q + k$, which leads to $p_s \geq \alpha + k - (\rho - \theta)q$.

SP3: For the OG users to switch but for NG users not to switch during the software life cycle, we have three conditions: (1) the OG users prefer the old perpetual software rather than SaaS at time 0 (i.e., $\theta q + k - p_s \leq q + k$); (2) the OG users prefer SaaS rather than the old perpetual software at the end of the software life cycle (i.e., $\theta q + \alpha + k - p_s \geq q + k$); and (3) the NG users prefer the new perpetual software rather than SaaS at the end of the software life cycle (i.e., $\theta q + \alpha + 2k - p_s \leq \rho q + k$). All together, we have $\max[(\theta - 1)q, \alpha + k - (\rho - \theta)q] \leq p_s \leq \alpha + (\theta - 1)q$.

SP4: For switching to occur, OG users derive higher payoff from SaaS than from the new perpetual software at the end of the software life cycle. Hence, $\theta q + \alpha + 2k - p_s \geq \rho q + k$, which leads to $p_s \leq \alpha + k - (\rho - \theta)q$.

SP5: We have two conditions: (1) the OG users prefer the old perpetual software rather than SaaS at time 0 (i.e., $\theta q + k - p_s \leq q + k$); and (2) the NG users derive higher payoff from SaaS than from the new perpetual software at the end of the software life cycle (i.e., $\theta q + \alpha + 2k - p_s \geq \rho q + k$). Therefore, $(\theta - 1)q \leq p_s \leq \alpha + k - (\rho - \theta)q$.

SP6: Note that both OG and NG users must switch at the same time. They derive higher payoff from SaaS than from the new perpetual software at the end of the software life cycle. Hence, $\theta q + \alpha + 2k - p_s \geq \rho q + 2k$, which leads to $p_s \leq \alpha - (\rho - \theta)q$.

Appendix D

Baseline Model Equilibrium Outcomes

Table D1 presents vendors' optimal prices, profit, consumer surplus, and social welfare under each equilibrium in the baseline model.

Table D1. Equilibrium Prices, Profits, Consumer Surplus, and Social Welfare: Baseline Model

(a) Equilibrium Prices: Baseline Model			
Equilibrium	P_u	P_n	P_s
Monopoly (M)	$(\rho - 1)q + k$	$\rho q + 2k$	NA
Entry Deterrence (I)	$(\rho - \theta)q - \frac{\alpha}{2} + k$	$(\rho - \theta)q - \frac{\alpha}{2} + k$	0
Market Segmentation (IIa)	$(\rho - 1)q$	$(\rho - 1)q + 2k$	$(\theta - 1)q + k + \frac{\alpha}{2}$
Market Segmentation (IIb)	$(\rho - 1)q$	$\frac{\alpha}{2} + 2k$	$\alpha + k - (\rho - \theta)q$
Sequential Dominance (IIIa)	$\frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$	$\frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$	$\frac{\alpha - (\rho - \theta)q}{2}$
Sequential Dominance (IIIb)	$\frac{\alpha k + [\alpha(\rho - 1) + k(\rho - \theta)]q - (\rho - 1)(\theta - 1)q^2}{6\alpha}$	$\frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$	$\frac{\alpha - (\rho - \theta)q}{2}$
(b) Equilibrium Profits: Baseline Model			
Equilibrium	π_{perp}	π_{SaaS}	
Monopoly (M)	$2(\rho - 1)q + 3k$	NA	
Entry Deterrence (I)	$2(\rho - \theta)q - \alpha + 2k$	0	
Market Segmentation (IIa)	$(\rho - 1)q$	$(\theta - 1)q + k + \frac{\alpha}{2}$	
Market Segmentation (IIb)	$(\rho - 1)q$	$\alpha + k - (\rho - \theta)q$	
Sequential Dominance (IIIa)	$\frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{4\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$	
Sequential Dominance (IIIb)	$\frac{2[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q] - [\alpha - (\rho + \theta - 2)q]^2}{8\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$	
(c) Equilibrium Consumer Surplus and Social Welfare: Baseline Model			
Equilibrium	CS_{OG}	CS_{NG}	SW
Monopoly (M)	$q + k$	0	$2\rho q + 4k$
Entry Deterrence (I)	$\theta q + k + \frac{\alpha}{2}$	$\theta q + k + \frac{\alpha}{2}$	$2\rho q + 4k$
Market Segmentation (IIa)	$q + k$	q	$(\rho + \theta)q + 2k + \frac{\alpha}{2}$
Market Segmentation (IIb)	$q + k$	$\rho q - \frac{\alpha}{2}$	$(\rho + \theta)q + 2k + \frac{\alpha}{2}$
Sequential Dominance (IIIa)	$\frac{3\alpha k + [\alpha(\rho + \theta) - k(\rho - \theta)]q}{2\alpha}$	$\frac{3\alpha k + [\alpha(\rho + \theta) - k(\rho - \theta)]q}{2\alpha}$	$\frac{3\alpha^2 + 16\alpha k + 2\alpha(\rho + 3\theta)q + 3(\rho - \theta)^2 q^2}{4\alpha}$
Sequential Dominance (IIIb)	$\frac{\alpha^2 + 12\alpha k + [2\alpha(\rho + \theta + 2) - 4k(\rho - \theta)]q + (\rho + \theta - 2)^2 q^2}{8\alpha}$	$\frac{3\alpha k + [\alpha(\rho + \theta) - k(\rho - \theta)]q}{2\alpha}$	$\frac{3\alpha^2 + 16\alpha k + 2\alpha(\rho + 3\theta)q + 3(\rho - \theta)^2 q^2}{4\alpha}$

Appendix E

Proofs for Baseline Model

Proof of Proposition 1 (Monopoly Market Equilibrium)

Proof. When no entry threat arises from the SaaS vendor, the perpetual software vendor is the monopolist. When the vendor releases the new version software at time 0, it charges a purchase price to the NG users so that it extracts all surpluses from them, and so $p_n^M = \rho q + 2k$. Meanwhile, it charges an upgrade price p_u as high as possible to induce the OG users to upgrade to the new version (i.e., $\rho q + 2k - p_u \geq q + k$). Therefore, $p_n^M = (\rho - 1)q + k$. The vendor's profit is $\pi^M = p_u^M + p_n^M = (2\rho - 1)q + 3k$.

Proof of Proposition 2 (Entry Deterrence Equilibrium)

Proof. This is the case in which $\alpha \leq (\rho - \theta)q$. Because the SaaS quality is always lower than the new perpetual software, users do not switch. The perpetual software vendor can choose either the entry deterrence strategy to serve both user groups and drive the SaaS vendor out of the market or it can choose the market segmentation strategy and serve OG users only. The equilibrium strategy pair corresponding to the former case is SP1 (Upgrade, New), while in the latter case it is SP2 (Upgrade, SaaS).

Consider SP1 (Upgrade, New). Given that NG users adopt the new version perpetual software, the OG users have three strategies to consider.

If they keep using the old version, their total utility is $q + k$; if OG users choose the SaaS at time 0, their total utility is $\int_0^1 (\theta q + \alpha t + k) dt$;

and if OG users choose to upgrade and then keep using the new perpetual software, their total utility is $\rho q + 2k - p_u$.

To ensure that the OG users prefer upgrading to the new version rather than continuing to use the old version, their total utility must be $\rho q + 2k - p_u \geq q + k$, which is $p_u \leq (\rho - 1)q + k$ (IC1). Meanwhile, the perpetual software vendor needs to make sure that OG users prefer upgrading rather than adopting SaaS, even if the SaaS price is reduced to zero. That is, the entry deterrence condition is $\rho q + 2k - p_u \geq$

$\int_0^1 (\theta q + \alpha t + k) dt$, and it gives $p_u \leq (\rho - \theta)q + k - \frac{\alpha}{2}$ (IC2). We can show that (IC1) is not binding.

Similarly, given that OG users choose to upgrade, the NG users' total utility is $\rho q + 2k - p_n$ if they choose the new perpetual software and

$\int_0^1 (\theta q + \alpha t + k) dt$ if they opt for SaaS at time 0 at zero price. To ensure that the NG users prefer the new perpetual software to the SaaS,

even if the SaaS price is zero, their total utility must be $\rho q + 2k - p_n \geq \int_0^1 (\theta q + \alpha t + k) dt$; that is, $p_n \leq (\rho - \theta)q + k - \frac{\alpha}{2}$ (IC3).

Because $p_u \leq p_n$, by (IC2) and (IC3) the perpetual software vendor sets the prices at respective upper bounds:

$p_n^{SP1} = p_u^{SP1} = (\rho - \theta)q + k - \frac{\alpha}{2}$. Consequently, we obtain the perpetual software vendor's profit at $\pi_{perp}^{SP1} = 2(\rho - \theta)q + 2k - \alpha$,

and the SaaS vendor is out of the market.

Finally, we need to prove that the perpetual software vendor earns a higher profit under SP1 than SP2, which is true when

$k \geq K_1 = \frac{\alpha - (\rho - 2\theta + 1)q}{2}$, as shown in the proof of Proposition 3. Hence, the perpetual software vendor deters the SaaS vendor's entry when

$k \geq K_1$.

Proof of Proposition 3 (Market Segmentation Equilibrium— α Low)

Proof. Consider SP2 (Upgrade, SaaS). Given that the NG users adopt SaaS, if the OG users continue to use the old version perpetual software, their total utility is $q + k$; if the OG users choose SaaS, the total utility is $\int_0^1 (\theta q + \alpha t + 2k - p_s) dt$; and if they choose to upgrade and then continue to use the new perpetual software over the entire software life cycle, the total utility is $\rho q + k - p_u$.

To ensure the OG users prefer to upgrade rather than to continue to use the old version, their total utility must be $\rho q + k - p_u \geq q + k$ and thus $p_u \leq (\rho - 1)q$ (IC4). Also, to ensure that the OG users prefer to upgrade rather than opt for SaaS, their total utility must be

$$\rho q + k - p_u \geq \int_0^1 (\theta q + \alpha t + 2k - p_s) dt \text{ and thus } p_s \geq p_u - (\rho - \theta)q - k + \frac{\alpha}{2} \text{ (IC5).}$$

Similarly, given that OG users upgrade, the NG users' total utility is $\rho q + 2k - p_n$ if they choose the new perpetual software and

$\int_0^1 (\theta q + \alpha t + k - p_s) dt$ if they opt for SaaS at time 0. To ensure that the NG users prefer SaaS, their total utility must be

$$\rho q + 2k - p_n \leq \int_0^1 (\theta q + \alpha t + k - p_s) dt; \text{ that is, } p_s \leq p_n - (\rho - \theta)q - k + \frac{\alpha}{2} \text{ (IC6).}$$

To maximize its profit, the perpetual software vendor sets p_n as high as possible so that the SaaS vendor can also charge a high enough price p_s , which in turn allows the perpetual software vendor to charge a high upgrade price p_u . As a result, the perpetual software vendor charges $p_u^{SP2} = (\rho - 1)q$ to make the OG users' IC constraint (IC4) binding. It sets $p_n^{SP2} = (\rho - 1)q + 2k$ so that the SaaS vendor charges the highest possible $p_s^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$ by (IC6) that does not violate (IC5). Finally, under the condition $\alpha < (\rho - \theta)q$, we can verify that the condition for SP2, $p_s > \alpha + k - (\rho - \theta)q$ as specified in Table C1, holds.

Finally, we need to show that the perpetual software vendor's profit under SP2, $\pi_{perp}^{SP2} = (\rho - 1)q$, is higher than its profit under SP1.

Solving $\pi_{perp}^{SP2} > \pi_{perp}^{SP1}$, we have $k < K_1$, where K_1 is defined in Proposition 2. Hence, SP2 (Upgrade, SaaS) sustains as an equilibrium user strategy pair when $k < K_1$. Also note that $K_1 = 0$ when $\alpha = (\rho - 2\theta + 1)q = \underline{\alpha}$.

Proof of Proposition 4 (Sequential Dominance Equilibrium)

Proof. Consider SP6 (Upgrade+SaaS, New+SaaS). The switching time t_{s3} is determined by $\theta q + \alpha t_{s3} + 2k - p_s = \rho q + 2k$, so that

$$t_{s3} = \frac{p_s + (\rho - \theta)q}{\alpha}. \text{ The SaaS vendor's profit is expressed as } 2p_s \left(1 - \frac{p_s + (\rho - \theta)q}{\alpha} \right). \text{ Solving this optimization problem yields the optimal}$$

SaaS price $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$. We can verify that p_s^* satisfies the SP6 condition in Table 3. Consequently, $t_{s3}^* = \frac{\alpha + (\rho - \theta)q}{2\alpha}$. Several incentive compatibility conditions must be satisfied, as follows.

Given that the OG users choose Upgrade+SaaS, the NG users prefer New+SaaS rather than SaaS if $(\rho q + 2k)t_{s3}^* - p_n +$

$$\int_{t_{s3}^*}^1 (\theta q + \alpha t + 2k - p_s^*) dt \geq \int_0^{t_{s3}^*} (\theta q + \alpha t + k - p_s^*) dt + \int_{t_{s3}^*}^1 (\theta q + \alpha t + 2k - p_s^*) dt. \text{ So } p_n \leq \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha} \text{ (IC7).}$$

Given that the NG users choose New+SaaS, the OG users prefer Upgrade+SaaS rather than Old+SaaS if $(\rho q + 2k)t_{s2}^* - p_u + \int_{t_{s2}^*}^1 (\theta q + \alpha t + 2k - p_s^*)dt \geq (q + k)t_{s1}^* + \int_{t_{s1}^*}^{t_{s2}^*} (\theta q + \alpha t + k - p_s^*)dt + \int_{t_{s2}^*}^1 (\theta q + \alpha t + 2k - p_s^*)dt$. The condition gives $p_u \leq \frac{k\alpha - (\rho - 1)(\theta - 1)q^2 + [\alpha(\rho - 1) + k(\rho - \theta)]q}{2\alpha}$ (IC8). Note that the switching time $t_{s2}^* = t_{s3}^*$. The switching time t_{s1} , for Old+SaaS, is determined by $\theta q + \alpha t_{s1} + k - p_s = q + k$, so that $t_{s1} = \frac{p_s - (\theta - 1)q}{\alpha}$. Substituting p_s^* into the expression of t_{s1} , we have $t_{s1}^* = \frac{\alpha - (\rho + \theta - 2)q}{2\alpha}$.

If $\alpha \leq (\rho + \theta - 2)q$, $t_{s1}^* < 0$, so that OG users prefer SaaS. To ensure the OG users prefer Upgrade+SaaS rather than SaaS, we need $(\rho q + 2k)t_{s3}^* - p_u + \int_{t_{s3}^*}^1 (\theta q + \alpha t + 2k - p_s^*)dt \geq \int_0^{t_{s3}^*} (\theta q + \alpha t + k - p_s^*)dt + \int_{t_{s3}^*}^1 (\theta q + \alpha t + 2k - p_s^*)dt$; that is, $p_u \leq \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$ (IC9). So by (IC7) and (IC9) we have $p_u^{SP6} = p_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$, and the perpetual software vendor's profit is $\pi_{perp}^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{4\alpha}$.

If $\alpha > (\rho + \theta - 2)q$, $t_{s1}^* > 0$, by (IC7) and (IC8) we have $p_u^{SP6} = \frac{k\alpha - (\rho - 1)(\theta - 1)q^2 + [\alpha(\rho - 1) + k(\rho - \theta)]q}{2\alpha} < p_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$, and $\pi_{perp}^{SP6} = \frac{2[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q] - [\alpha - (\rho + \theta - 2)]^2}{8\alpha}$.

Under both cases, the SaaS price is $p_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}$, and the SaaS vendor's profit is $\pi_{SaaS}^{SP6} = \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$.

Another outcome under the strategy pair SP2 (Upgrade, SaaS) is solved in Proposition 5. Comparing the two vendors' respective profits under SP2 and SP6, we show that when the network effect k is stronger than a threshold value K_2 (details in the proof of Proposition 5), SP6 (Upgrade+SaaS, New+SaaS) emerges as the final equilibrium user strategy.

Proof of Proposition 5 (Market Segmentation Equilibrium— α High)

Proof. Consider SP2 (Upgrade, SaaS). The analysis is similar to the proof for Proposition 3. The only difference is that when $\alpha > 2(\rho - 1)q$, the constraint $p_s \geq \alpha + k - (\rho - \theta)q$ (refer to Table 3) is binding. Therefore, $p_s^{SP2} = \alpha + k - (\rho - \theta)q$ if $\alpha > 2(\rho - 1)q$. Also, we need to reexamine the IC conditions. (IC5) becomes $p_u \leq \frac{\alpha}{2}$. Because $(\rho - 1)q \leq \frac{\alpha}{2}$, the perpetual software vendor charges $p_u^{SP2} = (\rho - 1)q$ so that (IC4) is binding. By (IC6), we have $p_n^{SP2} \geq \frac{\alpha}{2} + 2k$. As a result, when $\alpha > 2(\rho - 1)q$, the perpetual software vendor's profit is $\pi_{perp}^{SP2} = (\rho - 1)q$, and the SaaS vendor's profit is $\pi_{SaaS}^{SP2} = \alpha + k - (\rho - \theta)q$.

The optimal prices and profits for $\alpha \leq 2(\rho - 1)q$ are the same as in Proposition 3.

Finally, we compare profits of the two vendors under both SP2 (Upgrade, SaaS) and SP6 (Upgrade+SaaS, New+SaaS). The latter is given in Proposition 4. There are three cases:

Case (1) $(\rho - \theta)q \leq \alpha \leq (\rho + \theta - 2)q$. For the perpetual software vendor, $\pi_{perp}^{SP6} < \pi_{perp}^{SP2}$ if $k < \frac{\alpha(\rho - 1)q}{\alpha + (\rho - \theta)q} - \frac{\alpha + (\rho - \theta)q}{4} \doteq k_2$. At both boundary values, $\alpha = (\rho - \theta)q$ and $\alpha = (\rho + \theta - 2)q$, $k_2 = \frac{(\theta - 1)q}{2}$. In addition, we can show that there exists $\hat{\alpha} = [2\sqrt{(\rho - 1)(\rho - \theta)} - (\rho - \theta)]q \in [(\rho - \theta)q, (\rho + \theta - 2)q]$ such that $\frac{\partial k_2}{\partial \alpha} > 0$ for $\alpha \in [(\rho - \theta)q, \hat{\alpha}]$ and $\frac{\partial k_2}{\partial \alpha} < 0$ for $\alpha \in [\hat{\alpha}, (\rho + \theta - 2)q]$. Hence, the perpetual software vendor prefers SP2 if $k < k_2$. For the SaaS vendor, $\pi_{SaaS}^{SP6} < \pi_{SaaS}^{SP2}$ if $k > \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha} - (\theta - 1)q - \frac{\alpha}{2} \doteq k_1$. At $\alpha = (\rho - \theta)q$, $k_1 = -(\theta - 1)q - \frac{(\rho - \theta)q}{2} < 0$, and $\frac{\partial k_1}{\partial \alpha} < 0$. Therefore, the inequality always holds. The SaaS vendor always prefers SP2.

Case (2) $(\rho + \theta - 2)q \leq \alpha \leq 2(\rho - 1)q$. For the perpetual software vendor, $\pi_{perp}^{SP6} < \pi_{perp}^{SP2}$ if $k < \frac{8\alpha(\rho - 1)q + [\alpha - (\rho + \theta - 2)q]^2}{8[\alpha + (\rho - \theta)q]} - \frac{\alpha + (\rho - \theta)q}{4} \doteq k_3$. At $\alpha = (\rho + \theta - 2)q$, $k_3 = \frac{(\theta - 1)q}{2}$. Solving $k_3 = 0$, we get two roots. One is smaller than the lower bound $(\rho + \theta - 2)q$, and the other, $\bar{\alpha} = [(\rho + \theta - 2) + 2\sqrt{(\rho - 1)(\rho - \theta)}]q$, is greater than the upper bound $2(\rho - 1)q$. So $k_3 > 0$ in this range and the perpetual software vendor prefers SP2 if $k < k_3$. For SaaS, the condition is the same as in Case (1). The SaaS vendor always prefers SP2.

Case (3) $\alpha > 2(\rho - 1)q$. For the perpetual software vendor, $\pi_{perp}^{SP6} < \pi_{perp}^{SP2}$ if $k < k_3$. The analysis is the same as in Case (2). For the SaaS vendor, $\pi_{SaaS}^{SP6} < \pi_{SaaS}^{SP2}$ if $k > \frac{-[\alpha - (\rho - \theta)q][\alpha + (\rho - \theta)q]}{2\alpha} \doteq k_4$ and $k_4 < 0$. So the SaaS vendor always prefers SP2.

Overall, define $K_2 = \begin{cases} k_2 & \text{if } \alpha \leq (\rho + \theta - 2)q \\ k_3 & \text{if } \alpha > (\rho + \theta - 2)q \end{cases}$ and we get the results in Proposition 5.

Appendix F

Effect of α and k —Comparative Statics and Graphical Illustration

In this Appendix, we show how the two key parameters, α and k , affect equilibrium prices, profits, consumer surplus, and social welfare using comparative statics, and we also provide a graphical illustration.

Table F1. Comparative Statistics w.r.t. α

Equilibrium	p_u	p_n	p_s	π_{perp}	π_{SaaS}	CS_{OG}	CS_{NG}	SW
Monopoly (M)	—	—	NA	—	NA	—	—	—
Entry Deterrence (I)	↓	↓	—	↓	—	↑	↑	—
Market Segmentation (IIa)	—	—	↑	—	↑	—	—	↑
Market Segmentation (IIb)	—	—	↑	—	↑	—	↓	↑
Sequential Dominance (IIIa)	↑↓	↑↓	↑	↑↓	↑	↓↑	↑	↑
Sequential Dominance (IIIb)	↓	↑↓	↑	↑↓	↑	↓↑	↑	↑

Table F2. Comparative Statistics w.r.t. k

Equilibrium	p_u	p_n	p_s	π_{perp}	π_{SaaS}	CS_{OG}	CS_{NG}	SW
Monopoly (M)	↑	↑	NA	↑	NA	↑	—	↑
Entry Deterrence (I)	↑	↑	0	↑	—	↑	↑	↑
Market Segmentation (IIa)	—	↑	↑	—	↑	↑	—	↑
Market Segmentation (IIb)	—	↑	↑	—	↑	↑	—	↑
Sequential Dominance (IIIa)	↑	↑	—	↑	—	↑	↑	↑
Sequential Dominance (IIIb)	↑	↑	—	↑	—	↑	↑	↑

The graphic demonstrations in Figures F1 and F2 take the following parameter values: $q = 1$, $\rho = 2$, $\theta = 1.2$, and $k = 0.02$. In addition, $\alpha = 0.64$ indicates the equilibrium transition from entry deterrence to market segmentation; $\alpha = 2$ indicates the equilibrium transition from market segmentation II-a to II-b; and $\alpha = 2.25$ indicates the equilibrium transition from market segmentation to sequential dominance.

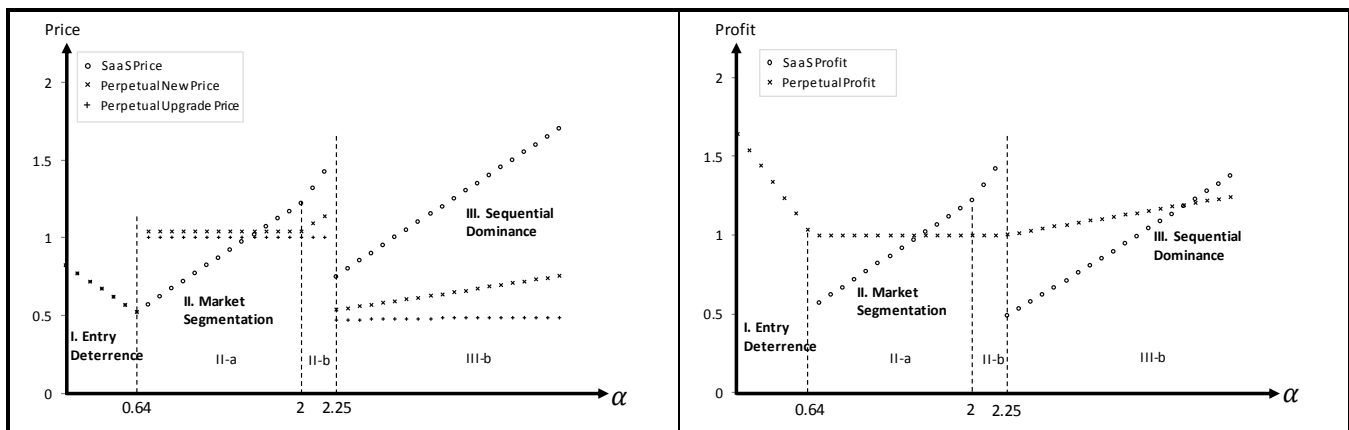
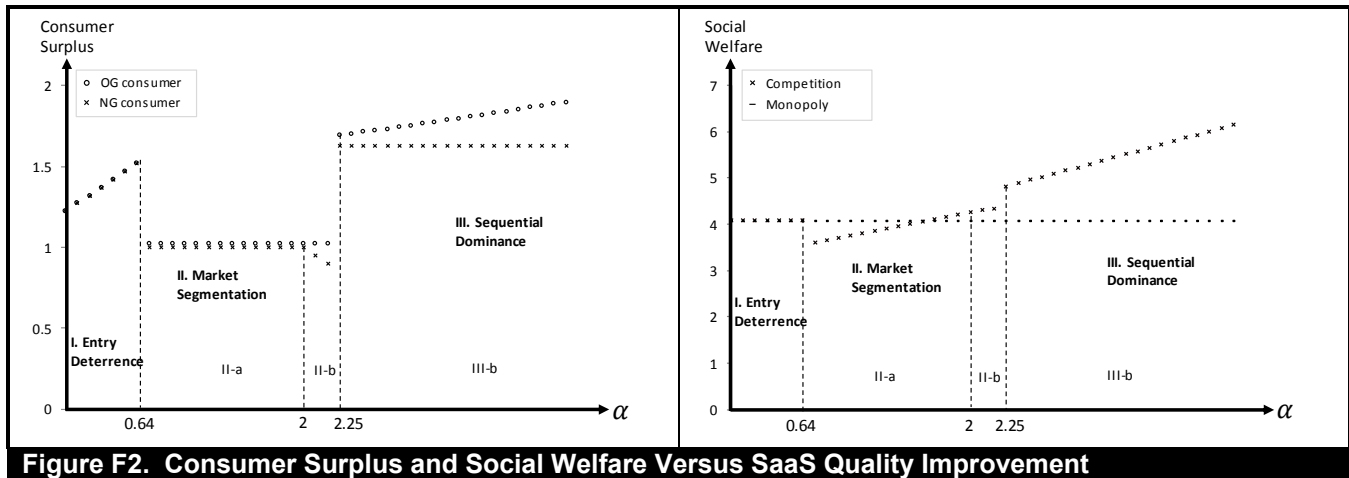


Figure F1. Vendors' Equilibrium Price and Profit Versus SaaS Quality Improvement



As seen in these figures, when the SaaS's quality improves at a low rate ($\alpha \leq 0.64$), the incumbent perpetual software vendor reduces both upgrade and purchase prices to deter the SaaS vendor's entry, reducing its own profit and resulting in higher consumer surplus. This suggests that the threat of entry by a potential competitor benefits customers.

As α further increases, deterring the SaaS vendor's entry becomes too costly. There is a threshold value ($\alpha = 0.64$) beyond which the perpetual software vendor no longer blocks the SaaS vendor's entry into the market. In the intermediate range of the SaaS quality improvement rate ($0.64 < \alpha \leq 2.25$), the perpetual software vendor pursues the market segmentation strategy by giving up NG users to the SaaS vendor and focusing on serving only OG users with a high price. As a result, its price and profit are independent of the SaaS quality. On the other hand, the SaaS vendor is only interested in exploiting NG users. As the SaaS quality increases at a higher rate, we see that the SaaS's price and profit monotonically increase.

Meanwhile, we observe that consumer surplus for both user groups drops significantly when the perpetual software vendor moves from the entry deterrence to the market segmentation equilibrium after $\alpha = 0.64$. As α increases from 2 to 2.25, the OG users' surplus is unaffected, but surprisingly, the NG users' surplus decreases. The intuition is that, when the SaaS has a large quality advantage over the perpetual software in the range, adopting the perpetual software becomes less attractive to NG users. Therefore, the SaaS vendor is able to price aggressively to extract more consumer surplus from NG users without transferring any benefit to them.

Finally, when the SaaS quality improvement rate is high enough ($\alpha > 2.25$), the SaaS becomes very attractive and the perpetual software vendor finds it difficult to prevent OG users from switching to SaaS. Instead, it should reduce both upgrade and purchase prices significantly to compete with the SaaS vendor for both user groups, moving to the sequential dominance strategy. The significant price-reduction pressure from the perpetual software vendor pushes the SaaS vendor to reduce its price as well, which results in a large drop in the SaaS vendor's profit at the transition point ($\alpha = 2.25$). On the other hand, the competition makes users better off, and the consumer surplus for both user groups jumps significantly upward.

As for social welfare, we also observe discrete upward and downward jumps at $\alpha = 0.64$ and 2.25, respectively, when the perpetual software vendor switches its competitive strategy. It is socially inefficient to allow the SaaS vendor to enter the market in the range $0.64 < \alpha < 2$; and after the SaaS vendor enters the market, the resulting social welfare is even lower than the monopoly benchmark. There are two reasons. First, the SaaS software has a low quality in this range. The NG users who adopt the SaaS therefore derive a lower average utility than in the monopoly benchmark, leading to a decrease in social welfare. Second, the SaaS vendor's entry results in a segmented market. Users are not able to enjoy the highest possible network value ($2k$) as they do in the benchmark case. Again, this reduces social welfare.

Appendix G

Perpetual Software Vendor's Incremental Quality Improvement

S1 (δ_1, t_{δ_1}): Patching before the SaaS Exceeds the Perpetual Software Quality

First, consider SP1 (Upgrade, New). Under SP1, the SaaS vendor is out of the market, even if it prices at 0. To ensure that the OG users prefer Upgrade rather than Old, we need $\rho q + \delta_1 q(1 - t_{\delta_1}) + 2k - p_u \geq q + k$; that is, $p_u \leq (\rho - 1)q + \delta_1 q(1 - t_{\delta_1}) + k$ (G1). To ensure that the OG users prefer Upgrade rather than SaaS, even if SaaS is priced at 0, we need $\rho q + \delta_1 q(1 - t_{\delta_1}) + 2k - p_u \geq \int_0^1 (\theta q + \alpha t + k)dt$; that is, $p_u \leq (\rho - \theta)q + \delta_1 q(1 - t_{\delta_1}) + k - \frac{\alpha}{2}$ (G2). To ensure that the NG users prefer New rather than SaaS, even if SaaS is priced at 0, we must have $\rho q + \delta_1 q(1 - t_{\delta_1}) + 2k - p_n \geq \int_0^1 (\theta q + \alpha t + k)dt$; that is, $p_n \leq (\rho - \theta)q + \delta_1 q(1 - t_{\delta_1}) + k - \frac{\alpha}{2}$ (G3). Therefore, the optimal price is $p_u^{SP1} = p_n^{SP1} = (\rho - \theta)q + \delta_1 q(1 - t_{\delta_1}) + k - \frac{\alpha}{2}$. The optimal profit is $\pi_{perp}^{SP1} = 2(\rho - \theta)q + 2\delta_1 q(1 - t_{\delta_1}) + 2k - \alpha$.

Next, consider SP2 (Upgrade, SaaS). To ensure that the OG users prefer Upgrade rather than Old, we need $\rho q + \delta_1 q(1 - t_{\delta_1}) + k - p_u \geq q + k$; that is, $p_u \leq (\rho - 1)q + \delta_1 q(1 - t_{\delta_1})$ (G4). To ensure that the OG users prefer Upgrade rather than SaaS, we need $\rho q + \delta_1 q(1 - t_{\delta_1}) + k - p_u \geq \int_0^1 (\theta q + \alpha t + 2k - p_s)dt$; that is, $p_u \leq p_s + (\rho - \theta)q + \delta_1 q(1 - t_{\delta_1}) - k - \frac{\alpha}{2}$ (G5). To ensure that the NG users prefer SaaS rather than New, we must have $\int_0^1 (\theta q + \alpha t + k - p_s)dt \geq \rho q + \delta_1 q(1 - t_{\delta_1}) + 2k - p_n$; that is, $p_n \geq p_s + (\rho - \theta)q + \delta_1 q(1 - t_{\delta_1}) + k - \frac{\alpha}{2}$ (G6). To ensure that OG users prefers Upgrade rather than SaaS, we need to make sure that at $t = 1$ the net benefit of switching to SaaS cannot exceed that of Upgrade: $\theta q + \alpha + 2k - p_s \leq (\rho + \delta_1)q + k$; that is, $p_s \geq \alpha + k - (\rho + \delta_1 - \theta)q$ (G7). Therefore, the optimal price is $p_u^{SP2} = (\rho - 1)q + \delta_1 q(1 - t_{\delta_1})$, and the optimal profit is $\pi_{perp}^{SP2} = (\rho - 1)q + \delta_1 q(1 - t_{\delta_1})$. The SaaS price is $p_s^{SP2} = (\rho - 1)q - \delta_1 q(1 - t_{\delta_1}) + k + \frac{\alpha}{2}$ if $\alpha \leq 2(\rho - 1)q - \delta_1 q t_{\delta_1}$; otherwise, $p_s^{SP2} = \alpha + k - (\rho + \delta_1 - \theta)q$.

Comparing the perpetual software vendor's profits under SP1 and SP2, we see that $\pi_{perp}^{SP1} > \pi_{perp}^{SP2}$ if $k > K'_1$, where $K'_1 = \frac{\alpha - (\rho - 2\theta + 1)q}{2} - \frac{\delta_1 q(1 - t_{\delta_1})}{2} < K_1$. Consequently, the lower bound value $\underline{\alpha}' = (\rho - 2\theta + 1)q + \delta_1 q(1 - t_{\delta_1}) > \underline{\alpha}$. Both K_1 and $\underline{\alpha}$ are critical values in the baseline model when the perpetual software vendor does not provide a quality jump. Hence, the K'_1 line shifts downward and the lower bound $\underline{\alpha}'$ shifts towards right.

Finally, consider SP6 (Upgrade+SaaS, New+SaaS). The switching time is determined by $\theta q + \alpha t_{IQ} + 2k - p_s = (\rho + \delta_1)q + 2k$; that is, $t_{IQ} = \frac{p_s + (\rho + \delta_1 - \theta)q}{\alpha} > t_{s3}$. The SaaS vendor's profit is expressed as $2p_s \left(1 - \frac{p_s + (\rho + \delta_1 - \theta)q}{\alpha}\right)$. Under the condition $\alpha \geq (\rho + \delta_1 - \theta)q$, solving this optimization problem yields the optimal SaaS price $p_s^{SP6} = \frac{\alpha - (\rho + \delta_1 - \theta)q}{2}$, which is lower than the optimal SaaS price under the baseline case.

To ensure that NG users prefer New+SaaS rather than SaaS, we need $(\rho q + 2k)t_{s4} + \delta_1 q(t_{s4} - t_{\delta_1}) - p_n + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt \geq \int_0^{t_{s4}} (\theta q + \alpha t_s + k - p_s)dt + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt$. Simplifying this inequality we have $p_n \leq \frac{[\alpha + (\rho + \delta_1 - \theta)q][4k + \alpha + (\rho + \delta_1 - \theta)q]}{8\alpha} - \delta_1 q t_{\delta_1}$ (G8). Furthermore, we need to ensure that OG users prefer Upgrade+SaaS rather than Old+SaaS. The switching time for Old+SaaS is $t_{s1} = \frac{p_s^{SP6} - (\theta - 1)q}{\alpha} = \frac{\alpha - (\rho + \delta_1 + \theta - 2)q}{2\alpha}$. If $\alpha > (\rho + \delta_1 + \theta - 2)q$, then the incentive compatibility condition is $(\rho q + 2k)t_{s4} + \delta_1 q(t_{s4} - t_{\delta_1}) - p_u + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt \geq (q + k)t_{s1} + \int_{t_{s1}}^{t_{s4}} (\theta q + \alpha t + k - p_s)dt + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt$. Simplifying this inequality, we have: $p_u \leq \frac{k\alpha - (\rho + \delta_1 - 1)(\theta - 1)q^2 + [\alpha(\rho + \delta_1 - 1) + k(\rho + \delta_1 - \theta)]q}{2\alpha} - \delta_1 q t_{\delta_1}$ (G9). If $\alpha \leq (\rho + \delta_1 + \theta - 2)q$, we need to ensure that OG users prefer Upgrade+SaaS rather than SaaS. Hence, $(\rho q + 2k)t_{s4} + \delta_1 q(t_{s4} - t_{\delta_1}) - p_u + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt \geq \int_0^{t_{s4}} (\theta q + \alpha t + k - p_s)dt + \int_{t_{s4}}^1 (\theta q + \alpha t + 2k - p_s)dt$, which leads to $p_u \leq \frac{[\alpha + (\rho + \delta_1 - \theta)q][4k + \alpha + (\rho + \delta_1 - \theta)q]}{8\alpha} - \delta_1 q t_{\delta_1}$ (G10). Therefore, $p_u^{SP6} = \frac{k\alpha - (\rho + \delta_1 - 1)(\theta - 1)q^2 + [\alpha(\rho + \delta_1 - 1) + k(\rho + \delta_1 - \theta)]q}{2\alpha} - \delta_1 q t_{\delta_1}$ and $p_n^{SP6} = \frac{[\alpha + (\rho + \delta_1 - \theta)q][4k + \alpha + (\rho + \delta_1 - \theta)q]}{8\alpha} - \delta_1 q t_{\delta_1}$ if $\alpha > (\rho + \delta_1 + \theta - 2)q$; and $p_u^{SP6} = p_n^{SP6} = \frac{[\alpha + (\rho + \delta_1 - \theta)q][4k + \alpha + (\rho + \delta_1 - \theta)q]}{8\alpha} - \delta_1 q t_{\delta_1}$ if $\alpha \leq (\rho + \delta_1 + \theta - 2)q$.

Next, we compare the perpetual software vendor's profits under SP2 and SP6. We find that, compared to the K_2 curve in the baseline model, the new K'_2 curve shifts downward. Specifically, if we redefine $\rho' = \rho + \delta_1$, we can write $K'_2 = \frac{\alpha(\rho' - 1)q}{\alpha + (\rho' - \theta)q} - \frac{\alpha + (\rho' - \theta)q}{4} + \delta_1 q t_{\delta_1}$ if $\alpha \leq (\rho' +$

$\theta - 2)q$ and $K'_2 = \frac{8\alpha(\rho'-1)q + [\alpha - (\rho' + \theta - 2)q]^2}{8[\alpha + (\rho' - \theta)q]} - \frac{\alpha + (\rho' - \theta)q}{4} + \delta_1 q t_{\delta 1}$ if $\alpha > (\rho' + \theta - 2)q$. Compared with K_2 , the K'_2 curve shifts towards the right. The upper bound $\bar{\alpha}'_{S1}$ is given by $K'_2 = 0$.

S2 ($\delta_2, t_{\delta 2}$): Patching After the SaaS Exceeds the Perpetual Software Quality

First, consider SP1 (Upgrade, New). The analysis is the same as above. We obtain the same three conditions (G1), (G2), and (G3). So, the solution is also the same: the optimal price is $p_u^{SP1} = p_n^{SP1} = (\rho - \theta)q + \delta_2 q(1 - t_{\delta 2}) + k - \frac{\alpha}{2}$, and the optimal profit is $\pi_{perp}^{SP1} = 2(\rho - \theta)q + 2\delta_2 q(1 - t_{\delta 2}) + 2k - \alpha$.

Next, consider SP2 (Upgrade, SaaS). Following the same analysis, we get the same conditions (G4), (G5), and (G6). In addition, we need to ensure that OG users prefer Upgrade rather than Upgrade+SaaS. If OG users chooses to switch from the upgraded perpetual software to SaaS, it must be at $t^* = \frac{(\rho - \theta)q + p_s - k}{\alpha}$. Note that at t^* , the perpetual vendor has not patched its product yet. To ensure that OG users stay with the perpetual software, their expected value from not switching, after considering the future quality improvement $\delta_2 q$ at $t_{\delta 2}$ should be higher than the expected value from switching to SaaS: $\int_{t^*}^{t_{\delta 2}} (\theta q + \alpha t + 2k - p_s) dt - (\rho q + k)(t_{\delta 2} - t^*) \leq (\rho q + \delta_2 q + k)(1 - t_{\delta 2}) - \int_{t_{\delta 2}}^1 (\theta q + \alpha t + 2k - p_s) dt$. Simplifying and solving this inequality yields $p_s \geq \alpha + k - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}$ (G11). Using (G4), we get the optimal upgrade price $p_u^{SP2} = (\rho - 1)q + \delta_2 q(1 - t_{\delta 2})$. Substituting p_u^{SP2} into (G5), we get $p_s \geq (\theta - 1)q + k + \frac{\alpha}{2}$. Now we compare this lower bound of p_s with the condition (G11): Define $\Delta \equiv (\theta - 1)q + k + \frac{\alpha}{2} - \{\alpha + k - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}\}$. When $\alpha < 2\delta_2 q(1 - t_{\delta 2})$, $\Delta > 0$. When $\alpha \geq 2\delta_2 q(1 - t_{\delta 2})$, $\Delta_{\alpha=2\delta_2 q(1-t_{\delta 2})} > 0$ and $\frac{\partial \Delta}{\partial \alpha} < 0$. So if α exceeds a certain threshold value, $\Delta < 0$. At the largest possible value of $\alpha_{max} = (\rho + \delta_2 - \theta)q$, we find that $\Delta_{\alpha=(\rho+\delta_2-\theta)q} > 0$. Therefore, we always have $\Delta > 0$. Consequently, the optimal SaaS price is $p_s^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$, at which the non-switching condition (G11) is always satisfied. The perpetual software prices are $p_u^{SP2} = p_n^{SP2} = (\rho - 1)q + \delta_2 q(1 - t_{\delta 2})$, and the profit is $\pi_{perp}^{SP2} = (\rho - 1)q + \delta_2 q(1 - t_{\delta 2})$.

Next, we compare the perpetual software vendor's profits under SP1 and SP2: $\pi_{perp}^{SP1} > \pi_{perp}^{SP2}$ if $k > K'_1$, where $K'_1 = \frac{\alpha - (\rho - 2\theta + 1)q}{2} - \frac{\delta_2 q(1 - t_{\delta 2})}{2}$. Note that both the K'_1 line and lower bound value $\underline{\alpha}'$ are as same as in the above Patching Strategy S1.

Finally, consider SP6 (Upgrade+SaaS, New+SaaS). The switching time is determined by $\theta q + \alpha t^* + 2k - p_s = \rho q + 2k$; that is, $t^* = \frac{p_s + (\rho - \theta)q}{\alpha}$. The SaaS vendor's profit is expressed as $2p_s \left(1 - \frac{p_s + (\rho - \theta)q}{\alpha}\right)$. It yields the optimal SaaS price $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$, which is the same as the optimal SaaS price in the baseline model. For SP6 to be an equilibrium, we need to ensure switching does happen. That is, at t^* , it must be $\int_{t^*}^{t_{\delta 2}} (\theta q + \alpha t - p_s) dt - \rho q(t_{\delta 2} - t^*) \geq (\rho q + \delta_2 q)(1 - t_{\delta 2}) - \int_{t_{\delta 2}}^1 (\theta q + \alpha t - p_s) dt$. Simplifying and solving this inequality yields $p_s \leq \alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}$ (G12). Now we check whether the SaaS price $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$ from the above optimization problem satisfies (G12). We can show that if $\delta_2 q(1 - t_{\delta 2}) \leq \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, p_s^* satisfies (G12) and so $p_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}$, $t^* = \frac{\alpha + (\rho - \theta)q}{2\alpha}$, otherwise, p_s^* does not satisfy (G12), and so $p_s^{SP6} = \alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}$, $t^* = \frac{\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}}{\alpha}$.

We need to ensure that NG users prefer New+SaaS rather than SaaS. That is, $(\rho q + 2k)t^* - p_n + \int_{t^*}^1 (\theta q + \alpha t + 2k - p_s) dt \geq \int_0^{t^*} (\theta q + \alpha t + k - p_s) dt + \int_{t^*}^1 (\theta q + \alpha t_s + 2k - p_s) dt$. When $\delta_2 q(1 - t_{\delta 2}) \leq \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, the condition leads to $p_n \leq \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$ (G13); otherwise, $p_n \leq \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}][2k + \alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta 2})}]}{2\alpha}$ (G14).

We also need to ensure that OG users prefer Upgrade+SaaS rather than Old+SaaS. The switching time in Old+SaaS is $t_{s1} = \frac{p_s^{SP6} - (\theta - 1)q}{\alpha}$. According to different values of $\delta_2 q(1 - t_{\delta 2})$, we analyze the following two cases.

Case (a) When $\delta_2 q(1 - t_{\delta 2}) \leq \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, $t_{s1} = \frac{\alpha - (\rho + \theta - 2)q}{2\alpha}$. If $\alpha > (\rho + \theta - 2)q$, $t_{s1} > 0$, and the incentive compatibility condition is $(\rho q + 2k)t^* - p_u + \int_{t^*}^1 (\theta q + \alpha t + 2k - p_s) dt \geq (q + k)t_{s1} + \int_{t_{s1}}^{t^*} (\theta q + \alpha t + k - p_s) dt + \int_{t^*}^1 (\theta q + \alpha t + 2k - p_s) dt$. Simplifying it we have $p_u \leq \frac{k\alpha - (\rho - 1)(\theta - 1)q^2 + [\alpha(\rho - 1) + k(\rho - \theta)]q}{2\alpha}$ (G15). Hence, the optimal perpetual software prices are given by (G13) and (G15). If $\alpha < (\rho + \theta - 2)q$, $t_{s1} < 0$, so the incentive compatibility condition is to ensure that OG users prefer Upgrade+SaaS rather than SaaS: $(\rho q +$

$2k)t^* - p_u + \int_{t^*}^1 (\theta q + \alpha t + 2k - p_s)dt \geq \int_0^{t^*} (\theta q + \alpha t + k - p_s)dt + \int_{t^*}^1 (\theta q + \alpha t + 2k - p_s)dt$, which leads to $p_u \leq \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$ (G16). Hence, the optimal perpetual software prices are given by (G13) and (G16).

Case (b) When $\delta_2 q(1 - t_{\delta_2}) > \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, $t_{s1} = \frac{\alpha - (\rho - 1)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}}{\alpha}$. If $\delta_2 q(1 - t_{\delta_2}) < \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$, $t_{s1} > 0$, and the incentive compatibility condition is to ensure that OG users prefer Upgrade+SaaS other than Old+SaaS. Then we have $p_u \leq [(\rho - 1)q + k] \frac{\alpha - \sqrt{2\delta_2 q(1 - t_{\delta_2})}}{\alpha} - \frac{(\rho - 1)^2 q^2}{2\alpha}$ (G17). Hence, the optimal perpetual software prices are given by (G14) and (G17). If $\delta_2 q(1 - t_{\delta_2}) > \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$, $t_{s1} < 0$, so the incentive compatibility condition is to ensure that OG users prefer Upgrade+SaaS rather than SaaS. Similarly, we get $p_u \leq \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}][2k + \alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]}{2\alpha}$ (G18). Hence, the optimal perpetual software prices are given by (G14) and (G18).

Note that $\frac{[\alpha - (\rho - \theta)q]^2}{8\alpha} > \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$ when $\alpha < (\rho + \theta - 2)q$, and $\frac{[\alpha - (\rho - \theta)q]^2}{8\alpha} < \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$ when $\alpha > (\rho + \theta - 2)q$. As a result, the optimal prices and vendor profits in SP6 can be summarized in the following, depending on both $\delta_2 q(1 - t_{\delta_2})$ and α . Define $\underline{v} = \min\left\{\frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}, \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}\right\}$ and $\bar{v} = \max\left\{\frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}, \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}\right\}$. We have three cases:

(i) $\delta_2 q(1 - t_{\delta_2}) < \underline{v}$: if $\alpha < (\rho + \theta - 2)q$, $p_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}$, $p_u^{SP6} = p_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$, $\pi_{SaaS}^{SP6} = \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$, and $\pi_{perp}^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{4\alpha}$; if $\alpha > (\rho + \theta - 2)q$, $p_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}$, $p_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$, $p_u^{SP6} = \frac{k\alpha - (\rho - 1)(\theta - 1)q^2 + [\alpha(\rho - 1) + k(\rho - \theta)]q}{2\alpha}$, $\pi_{SaaS}^{SP6} = \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$, and $\pi_{perp}^{SP6} = \frac{2[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q] - [\alpha - (\rho + \theta - 2)q]^2}{8\alpha}$.

(ii) $\underline{v} < \delta_2 q(1 - t_{\delta_2}) < \bar{v}$: if $\alpha < (\rho + \theta - 2)q$, $p_s^{SP6} = \frac{\alpha - (\rho - \theta)q}{2}$, $p_u^{SP6} = p_n^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{8\alpha}$, $\pi_{SaaS}^{SP6} = \frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$, $\pi_{perp}^{SP6} = \frac{[\alpha + (\rho - \theta)q][4k + \alpha + (\rho - \theta)q]}{4\alpha}$; if $\alpha > (\rho + \theta - 2)q$, $p_s^{SP6} = \alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}$, $p_n^{SP6} = \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}][2k + \alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]}{2\alpha}$, $p_u^{SP6} = [(\rho - 1)q + k] \frac{\alpha - \sqrt{2\delta_2 q(1 - t_{\delta_2})}}{\alpha} - \frac{(\rho - 1)^2 q^2}{2\alpha}$, $\pi_{SaaS}^{SP6} = \frac{2\sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}[\alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]^2}{\alpha}$, $\pi_{perp}^{SP6} = \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]^2 + 2[(\rho - 1)q + 2k][\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}] - [(\rho - 1)q]^2}{2\alpha}$.

(iii) $\delta_2 q(1 - t_{\delta_2}) > \bar{v}$: $p_n^{SP6} = p_u^{SP6} = \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}][2k + \alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]}{2\alpha}$, $p_s^{SP6} = \alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}$, $\pi_{SaaS}^{SP6} = \frac{2\sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}[\alpha - (\rho - \theta)q - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]^2}{\alpha}$, and $\pi_{perp}^{SP6} = \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}][2k + \alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]}{\alpha}$.

Finally, we compare the perpetual software vendor's profits under SP2 and SP6. The comparison should be done in each region of $\delta_2 q(1 - t_{\delta_2})$. In (i), when $\delta_2 q(1 - t_{\delta_2})$ is small, the perpetual vendor's profit in SP6, π_{perp}^{SP6} , is the same as in the baseline model. Hence, the $K'_2 = K_2 + \frac{\alpha}{\alpha + (\rho - \theta)q} \delta_2 q(1 - t_{\delta_2})$ curve that divides the market segmentation equilibrium (SP2) and the sequential dominance equilibrium (SP6) shifts upward and toward the right, compared to the K_2 curve in the baseline model. Similarly, in (ii), we have $K'_2 = \frac{\alpha(\rho - 1)q}{\alpha + (\rho - \theta)q} - \frac{\alpha + (\rho - \theta)q}{4} + \frac{\alpha}{\alpha + (\rho - \theta)q} \delta_2 q(1 - t_{\delta_2})$ if $\alpha < (\rho + \theta - 2)q$ and $K'_2 = \frac{\alpha[(\rho - 1)q + \delta_2 q(1 - t_{\delta_2})]}{2[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]} + \frac{[(\rho - 1)q]^2 - [\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]^2}{4[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]} - \frac{(\rho - 1)q}{2}$ if $\alpha \geq (\rho + \theta - 2)q$. In (iii), we have $K'_2 = \frac{\alpha[(\rho - 1)q + \delta_2 q(1 - t_{\delta_2})]}{2[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]} - \frac{[\alpha - \sqrt{2\alpha\delta_2 q(1 - t_{\delta_2})}]}{2}$. Under the three cases, the upper bound $\bar{\alpha}'_{S2(i)}$, $\bar{\alpha}'_{S2(ii)}$ and $\bar{\alpha}'_{S2(iii)}$ are given by solving $K'_2 = 0$. Furthermore, $\bar{\alpha}'_{S2(iii)} > \bar{\alpha}'_{S2(ii)} > \bar{\alpha}'_{S2(i)}$, and $\bar{\alpha}'_{S2(i)} > \bar{\alpha}$.

To conclude, in each case, there are no qualitative changes in the competition outcomes, except that the equilibrium regions are shifted.

Proof of Proposition 6 (Optimal Patching Strategy and Time)

We show the proof based on a special case $k = 0$. The reasoning for the general case is similar. We omit the proof because the mathematical expressions are quite lengthy.

Define $\alpha_1 = \bar{\alpha}'_{S1}$ and $\alpha_2 = \bar{\alpha}'_{S2(i)}$ where $\bar{\alpha}'_{S1}$ and $\bar{\alpha}'_{S2(i)}$ are the upper bound in S1 and S2, respectively. When $\alpha < \alpha_1$, the equilibrium under S1 and S2 is the same (either entry deterrence or market segmentation). The perpetual software vendor's profit functions are also the same. Since its profit is linearly increasing in the patching value, the optimal patching time is determined by solving the largest patching value: $t_\delta^* = \underset{t_\delta \in (0,1)}{\text{Argmax}}\{\delta q(1 - t_\delta)\}$. It can be either before or after t^* .

When $\alpha_1 < \alpha < \alpha_2$, for any patching value, the equilibrium under S1 is sequential dominance and under S2 is market segmentation. Next we compare the two equilibrium profits for the perpetual software vendor. Define $v_1 \equiv \frac{[\alpha - (\rho + \delta_1 - \theta)q]^2}{8\alpha} + \frac{(\rho + \delta_1 - 1)[\alpha - (\theta - 1)q]^2}{2\alpha} - (\rho - 1)q - 2\delta_1 q t_{\delta_1}$. If $V_{S2} > v_1$, S2 offers a higher profit than S1. The vendor's profit π_{perp}^{SP2} under S2 is linearly increasing in its patching value. The optimal patching time is given by $t_{\delta}^* = \underset{t_{\delta} \in (t^*, 1)}{\text{Argmax}}\{\delta q(1 - t_{\delta})\}$. So the optimal patching time should be later than t^* . If $V_{S2} < v_1$, S1 offers a higher profit than S2, and the optimal patching time should be earlier than t^* . The optimal patching time is determined by solving the profit maximization problem under π_{perp}^{SP6} : $\underset{t_{\delta} \in (0, t^*)}{\text{Max}} \left\{ \frac{[\alpha + (\rho + \delta t_{\delta} - \theta)q]^2}{8\alpha} + \frac{(\rho + \delta t_{\delta} - 1)q[\alpha - (\theta - 1)q]}{2\alpha} - 2\delta q t_{\delta} \right\}$.

When $\alpha > \alpha_2$, the equilibrium under S1 is sequential dominance. Consider two possibilities. (1) If $V_{S2} < \underline{v}$, the equilibrium under S2 is sequential dominance as in the aforementioned case (i). The perpetual software vendor's profit π_{perp}^{SP6} under S2 is the same as in the baseline model. It does not depend on the patching value V_{S2} at all. So it is always smaller than the profit π_{perp}^{SP6} under S1. The vendor therefore should prefer S1, and its optimal patching time should be earlier than t^* and it maximizes π_{perp}^{SP6} under S1: $\underset{t_{\delta} \in (0, t^*)}{\text{Max}} \left\{ \frac{[\alpha + (\rho + \delta t_{\delta} - \theta)q]^2}{8\alpha} + \frac{(\rho + \delta t_{\delta} - 1)q[\alpha - (\theta - 1)q]}{2\alpha} - 2\delta q t_{\delta} \right\}$. (2) If $V_{S2} > \underline{v}$, under S2, we are in cases (ii) and (iii). However, $\bar{\alpha}'_{S2(ii)} > \bar{\alpha}'_{S2(iii)} > (\rho + \delta_2 - \theta)q$. The resulting equilibrium is market segmentation. Hence, we compare π_{perp}^{SP6} under S1 and π_{perp}^{SP2} under S2. The analysis and results are the same as those in $\alpha_1 < \alpha < \alpha_2$: If $V_{S2} < v_1$, the optimal patching time should be before t^* ; otherwise, the optimal patching time should be after t^* .

Define $v_2 \equiv \max(v_1, \underline{v})$. By combining the above analyses in all regions of α and V_{S2} , we complete the proof of Proposition 6.

Appendix H

Perpetual Software Vendor's Major Quality Improvement (Two-Period Model)

When $\alpha \leq (\rho - \theta)q$, the SaaS quality improvement rate is small such that the perpetual software always has the quality advantage in both periods. In this case, the perpetual software vendor can deter SaaS entry. The corresponding equilibrium strategy pair is SP1'[(Upgrade1, Upgrade2), (New1, Upgrade2)].

When $\alpha > (\rho - \theta)q$, the SaaS entry cannot be deterred. There are two cases. If $(\rho - \theta)q < \alpha \leq (\rho - 1)q$, the single-period quality improvement of SaaS is smaller than that of the perpetual software. Because the SaaS has relative quality advantage in the first period but not in the second period, the possible equilibrium strategies are either SP3'[(Upgrade1+SaaS, Upgrade2+SaaS), (New1+SaaS, Upgrade2+SaaS)] or SP3''[(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)].

If $(\rho - 1)q < \alpha \leq (2\rho - \theta - 1)q$, the single-period quality improvement of SaaS is larger than that of the perpetual software. Because the SaaS has relative quality advantage in the second period but not in the first period, the possible strategies are either SP3'[(Upgrade1+SaaS, Upgrade2+SaaS), (New1+SaaS, Upgrade2+SaaS)] or SP3''[(Upgrade1, Upgrade2+SaaS), (New1, Upgrade2+SaaS)].

Furthermore, because the perpetual software has quality advantage at the beginning of each period, and it has OG users as the established customer base, the perpetual software vendor might consider the market segmentation strategy to give up the NG users in both periods or only in one period. The possible equilibrium strategies are SP2'[(Upgrade1, Upgrade2), (SaaS, SaaS)] for all α , SP2''[(Upgrade1, Upgrade2), (SaaS, New2)] if $(\rho - \theta)q < \alpha \leq (\rho - 1)q$. Note that if $(\rho - 1)q < \alpha \leq (2\rho - \theta - 1)q$, SP2'''[(Upgrade1, Upgrade2), (New1, SaaS)] cannot emerge as equilibrium because after OG users upgrade and NG users adopt the new perpetual software, their actions should be the same.

Entry Deterrence Strategy

Consider SP1'[(Upgrade1, Upgrade2), (New1, Upgrade2)]. Because the SaaS vendor can reduce price to zero, to prevent users from switching to SaaS at anytime between $[0, 2]$, we need $\theta q + \alpha \leq \rho q$; that is, $\alpha \leq (\rho - \theta)q$.

Given that the NG users adopt the perpetual software in both periods, to ensure that the OG users prefer upgrading in both periods rather than just in the first period, we have $\rho q + 2k + (2\rho - 1)q + 2k - 2p_u \geq \rho q + 2k + \rho q + k - p_u$; that is, $p_u \leq (\rho - 1)q + k$ (H1). Similarly, given that the OG users choose to upgrade in both periods, to ensure that the NG users prefer to buy new perpetual software and upgrade in period 2 rather than not upgrading, their total utility must be $\rho q + 2k + (2\rho - 1)q + 2k - p_n - p_u \geq \rho q + 2k + \rho q + k - p_n$, which is the same as (H1).

To ensure that OG users prefer upgrading in both periods rather than adopting SaaS in any period, even if the SaaS price is reduced to zero, the entry deterrence condition is $(\rho q + 2k) + (2\rho - 1)q + 2k - 2p_u \geq \max[\int_0^2 (\theta q + \alpha t + k)dt, \int_0^1 (\theta q + \alpha t + k)dt + (2\rho - 1)q + 2k - p_u, \int_1^2 (\theta q + \alpha t + k)dt + \rho q + 2k - p_u]$. In addition, to ensure that the NG users prefer (New1, Upgrade2) to the SaaS in any period, even if the SaaS price is zero, their total utility must be $\rho q + 2k + (2\rho - 1)q + 2k - p_n - p_u \geq \max[\int_0^2 (\theta q + \alpha t + k)dt, \int_0^1 (\theta q + \alpha t + k)dt + (2\rho - 1)q + 2k - p_n, \int_1^2 (\theta q + \alpha t + k)dt + \rho q + 2k - p_n]$. Solving these inequalities, we have $p_u \leq (\rho - \theta)q + k - \frac{\alpha}{2}$ (H2) and $p_n + p_u \leq (3\rho - 2\theta - 1)q + 2k - 2\alpha$ (H3).

Comparing (H1) and (H2) we see (H1) is not binding. So by (H2) the perpetual software vendor sets the upgrade price at the upper bound $p_u = (\rho - \theta)q + k - \frac{\alpha}{2}$, and by (H3) $p_n = (2\rho - \theta - 1)q + k - \frac{3\alpha}{2}$. We can verify that $p_u < p_n$. Consequently, the perpetual software vendor's profit is $\pi_{perp}^{SP1'} = 3p_u + p_n = (5\rho - 4\theta - 1)q + 4k - 3\alpha$, and the SaaS vendor is out of the market.

Market Segmentation Strategy

Case (1) Consider $SP2'[(\text{Upgrade1}, \text{Upgrade2}), (\text{SaaS}, \text{SaaS})]$. To prevent the OG users from switching to SaaS, the SaaS payoff at the end of each period should not be higher than payoff from the new perpetual software for OG users. Thus, we have $\theta q + \alpha + 2k - p_s \leq \rho q + k$, and $\theta q + 2\alpha + 2k - p_s \leq (2\rho - 1)q + k$. Hence, if $\alpha \leq (\rho - 1)q$, $p_s \geq \alpha + k - (\rho - \theta)q$ (H4); and if $\alpha > (\rho - 1)q$, $p_s \geq 2\alpha + k - (2\rho - \theta - 1)q$ (H5).

Given that the NG users adopt SaaS in both periods, to ensure that the OG users prefer to upgrade in both periods rather than opt for SaaS, their total utility must be $\rho q + k + (2\rho - 1)q + k - 2p_u \geq \int_0^2 (\theta q + \alpha t + 2k - p_s)dt$ and thus $p_u \leq p_s + \frac{(3\rho - 2\theta - 1)q}{2} - k - \alpha$ (H6). To ensure the OG users to upgrade in both periods rather than just in one period, we must have $\rho q + k + (2\rho - 1)q + k - 2p_u \geq \max[2(\rho q + k) - p_u, \rho q + k + (2\rho - 1)q + k - p_u]$; that is, $p_u \leq (\rho - 1)q$ (H7).

Similarly, given that the OG users upgrade in both periods, to ensure that the NG users prefer (SaaS, SaaS) rather than (SaaS, New2), we must have $\int_0^2 (\theta q + \alpha t + k - p_s)dt \geq \int_0^1 (\theta q + \alpha t + k - p_s)dt + (2\rho - 1)q + 2k - p_n$; which is $p_n \geq p_s + (2\rho - \theta - 1)q + k - \frac{3\alpha}{2}$ (H8). To ensure that the NG users prefer (SaaS, SaaS) rather than (New1, Upgrade2), we must have $\int_0^2 (\theta q + \alpha t + k - p_s)dt \geq \rho q + 2k + (2\rho - 1)q + 2k - p_n - p_u$; that is, $p_n + p_u \geq 2p_s + (3\rho - 2\theta - 1)q + 2k - 2\alpha$ (H9).

If $\alpha \leq (\rho - 1)q$, to maximize its profit, the perpetual software vendor charges $p_u = (\rho - 1)q$ and sets p_n high enough such that the SaaS vendor can charge a high enough price p_s , so that the OG users would not opt for SaaS. By binding constraint (H6), we have $p_s = \frac{(2\theta - \rho - 1)q}{2} + k + \alpha$. We can verify that (H4) is satisfied. By (H8) and (H9), $p_n = \max[\frac{3(\rho - 1)q}{2} + 2k - \frac{3\alpha}{2}, 2(\rho - 1)q + 4k]$. The perpetual software vendor's profit is $\pi_{perp}^{SP2'} = 2(\rho - 1)q$, and the SaaS vendor's profit is $\pi_{saas}^{SP2'} = (2\theta - \rho - 1)q + 2k + 2\alpha$.

If $(\rho - 1)q < \alpha \leq \frac{3(\rho - 1)q}{2}$, (H5) can be satisfied and the same solution as above holds.

If $\alpha > \frac{3(\rho - 1)q}{2}$, then we obtain the boundary solution $p_s = 2\alpha + k - (2\rho - \theta - 1)q$. Now, (H8) becomes $p_n \geq 2k + \frac{\alpha}{2}$, and (H9) becomes $p_n + p_u \geq 4k + 2\alpha - (\rho - 1)q$. So $p_u = (\rho - 1)q$ and $p_n = 4k + 2\alpha - 2(\rho - 1)q$. The perpetual software vendor's profit is $\pi_{perp}^{SP2'} = 2(\rho - 1)q$, and the SaaS vendor's profit is $\pi_{saas}^{SP2'} = 4\alpha + 2k - 2(2\rho - \theta - 1)q$.

Comparing $\pi_{perp}^{SP2'}$ with $\pi_{perp}^{SP1'}$ we see that if $k > \frac{3\alpha - (3\rho - 4\theta + 1)q}{4} = K_1'$, then $\pi_{perp}^{SP1'} > \pi_{perp}^{SP2'}$, the entry deterrence strategy dominates the market segmentation strategy. Solving $K_1' = 0$ we get $\underline{\alpha}'$.

Case (2) If $(\rho - \theta)q < \alpha \leq (\rho - 1)q$, consider $SP2''[(\text{Upgrade1}, \text{Upgrade2}), (\text{SaaS}, \text{New2})]$. Given that the NG users adopt (SaaS, New2), OG users prefer (Upgrade1, Upgrade2) rather than (SaaS, Upgrade2) if $\rho q + k - p_u \geq \int_0^1 (\theta q + \alpha t + 2k - p_s)dt$; that is $p_u \leq p_s + (\rho - \theta)q - k - \frac{\alpha}{2}$ (H10). Given that OG users upgrade in both periods, to ensure NG users prefer (SaaS, New2) rather than (New1, Upgrade2), we need $\int_0^1 (\theta q + \alpha t + k - p_s)dt + (2\rho - 1)q + 2k - p_n \geq \rho q + 2k - p_n + (2\rho - 1)q + 2k - p_u$; that is, $p_u \geq p_s + (\rho - \theta)q + k - \frac{\alpha}{2}$ (H11). Because (H10) and (H11) contradict with each other, this user strategy does not support an equilibrium.

Sequential Dominance Strategy

When $\alpha \geq (\rho - \theta)q$, the two competing firms' periodical quality improvement is competitive against each other. There are three possible strategies:

- (1) SP3'[(Upgrade1+SaaS, Upgrade2+SaaS), (New1+SaaS, Upgrade2+SaaS)]. This symmetric strategy can occur in both $\alpha \leq (\rho - 1)q$ and $\alpha > (\rho - 1)q$ ranges.
- (2) SP3''[(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)]. This asymmetric strategy can only occur when $\alpha \leq (\rho - 1)q$; that is, the perpetual software vendor has higher single-period quality improvement than the SaaS vendor.
- (3) SP3'''[(Upgrade1, Upgrade2+SaaS), (New1, Upgrade2+SaaS)]. This asymmetric strategy can only occur when $\alpha > (\rho - 1)q$; that is, the SaaS has higher single-period quality improvement than the perpetual software.

Case (1) Consider SP3'. The sequential dominance strategy involves user switching. If users switch from the new/updated perpetual software to SaaS in the first period, the switching time is determined by $\theta q + \alpha t_{\sigma 1} + 2k - p_s = \rho q + 2k$; that is, $t_{\sigma 1} = \frac{p_s + (\rho - \theta)q}{\alpha}$. If users switch from the updated perpetual software to SaaS in the second period, the switching time is determined by $\theta q + \alpha t_{\sigma 2} + 2k - p_s = (2\rho - 1)q + 2k$; that is, $t_{\sigma 2} = \frac{p_s + (2\rho - \theta - 1)q}{\alpha}$. If users switch from the old version software to SaaS, the switching time is determined by $\theta q + \alpha t_{\sigma 3} + k - p_s = q + k$, so that $t_{\sigma 3} = \frac{p_s - (\theta - 1)q}{\alpha}$.

If the SaaS vendor would like to serve in both periods, we need $0 < t_{\sigma 1} < 1$ and $1 < t_{\sigma 2} < 2$. That is, if $\alpha \leq (\rho - 1)q$, $\alpha - (2\rho - \theta - 1)q < p_s \leq 2\alpha - (2\rho - \theta - 1)q$ (H12); if $\alpha > (\rho - 1)q$, $\alpha - (2\rho - \theta - 1)q < p_s \leq \alpha - (\rho - \theta)q$ (H13). The SaaS vendor's profit is $2p_s(1 - t_{\sigma 1}) + 2p_s(2 - t_{\sigma 2})$. Solving this optimization problem we have interior solution $p_s^* = \frac{3\alpha - (3\rho - 2\theta - 1)q}{4}$. Checking (H12) and (H13) we can verify that this interior solution holds if $\frac{(5\rho - 2\theta - 3)q}{5} < \alpha < (5\rho - 2\theta - 3)q$.

At this interior solution, given that the OG users choose (Upgrade1+SaaS, Upgrade2+SaaS), in order for NG users to prefer (New1+SaaS, Upgrade2+SaaS) rather than (SaaS, New2+SaaS), we need $(\rho q + 2k)t_{\sigma 1} - p_u \geq \int_0^{t_{\sigma 1}} (\theta q + \alpha t + k - p_s)dt$, which is $p_u \leq \frac{[p_s + (\rho - \theta)q + 2k][p_s + (\rho - \theta)q]}{2\alpha}$ (H14). In order for NG users to prefer (New1+SaaS, Upgrade2+SaaS) rather than (SaaS, SaaS), we have $(\rho q + 2k)t_{\sigma 1} - p_n + [(2\rho - 1)q + 2k](t_{\sigma 2} - 1) - p_u \geq \int_0^{t_{\sigma 1}} (\theta q + \alpha t + k - p_s)dt + \int_1^{t_{\sigma 2}} (\theta q + \alpha t + k - p_s)dt$; that is, $p_n + p_u \leq \frac{[p_s + (\rho - \theta)q + 2k][p_s + (\rho - \theta)q] + [p_s + (2\rho - \theta - 1)q - \alpha + 2k][p_s + (2\rho - \theta - 1)q - \alpha]}{2\alpha}$ (H15). Given the NG users choose (New1+SaaS, Upgrade2+SaaS), in order for the OG users to prefer (Upgrade1+SaaS, Upgrade2+SaaS) rather than (Old+SaaS, Upgrade2+SaaS), we need $(\rho q + 2k)t_{\sigma 1} - p_u \geq (q + k)t_{\sigma 3} + \int_{t_{\sigma 3}}^{t_{\sigma 1}} (\theta q + \alpha t + k - p_s)dt$. Solving this inequality we have $p_u \leq \frac{2[(\rho - 1)q + k]p_s + 2k(\rho - \theta)q + (\rho - 1)(\rho - 2\theta + 1)q^2}{2\alpha}$ (H16).

If $\alpha \leq \frac{(3\rho + 2\theta - 5)q}{3}$, $t_{\sigma 3} \leq 0$. In order for the OG users to prefer (Upgrade1+SaaS, Upgrade2+SaaS) rather than (SaaS, Upgrade2+SaaS), we need $(\rho q + 2k)t_{\sigma 1} - p_u \geq \int_0^{t_{\sigma 1}} (\theta q + \alpha t + k - p_s)dt$, which is the same as (H14). If $\alpha > \frac{(3\rho + 2\theta - 5)q}{3}$, $t_{\sigma 3} \geq 0$. Comparing (H14) and (H16) we can verify that (H16) binds. Therefore, for the SP3' interior solution, we have the following:

If $\frac{(5\rho - 2\theta - 3)q}{5} < \alpha \leq (\rho - 1)q$, (H14) binds. So $p_u = \frac{[(\rho - 2\theta + 1)q + 3\alpha + 8k][(\rho - 2\theta + 1)q + 3\alpha]}{32\alpha}$ and $p_n = \frac{[(5\rho - 2\theta - 3)q - \alpha + 8k][(\rho - 2\theta - 3)q - \alpha]}{32\alpha}$. Furthermore, $p_u < p_n$.

If $(\rho - 1)q < \alpha \leq \frac{(3\rho + 2\theta - 5)q}{3}$, (H14) binds. So we have $p_u = p_n = \frac{5\alpha^2 + 8\alpha k + (24k\rho - 16k\theta - 2\alpha\rho - 4\alpha\theta + 6\alpha - 8k)q + (13\rho^2 - 12\rho\theta + 4\theta^2 - 14\rho + 4\theta + 5)q^2}{32\alpha}$.

If $\frac{(3\rho + 2\theta - 5)q}{3} < \alpha < (2\rho - \theta - 1)q$, (H16) imposes an upper bound for p_u . If $k > k_1 = \frac{21\rho^2 q^2 + 4\rho\theta q^2 + 4\theta^2 q^2 - 26\alpha\rho q - 4\alpha\theta q - 46\rho q^2 - 12\theta q^2 + 5\alpha^2 + 30\alpha q + 29q^2}{16[\alpha - (\rho - 1)q]}$, we still have $p_u = p_n = \frac{5\alpha^2 + 8\alpha k + (24k\rho - 16k\theta - 2\alpha\rho - 4\alpha\theta + 6\alpha - 8k)q + (13\rho^2 - 12\rho\theta + 4\theta^2 - 14\rho + 4\theta + 5)q^2}{32\alpha}$. We can verify that the condition $k > k_1$ always holds in this α range.

Now consider the boundary solution. If $(\rho - \theta)q \leq \alpha \leq \frac{(5\rho - 2\theta - 3)q}{5}$, then the SaaS vendor prices at boundary solution $p_s^* = 2\alpha - (2\rho - \theta - 1)q$. Correspondingly, $t_{\sigma 2} = 2$. SP3' degenerates to equilibrium SP3''[(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)]. Substituting p_s^* into (H14) we have $p_u = \frac{[2\alpha - (\rho - 1)q + 2k][2\alpha - (\rho - 1)q]}{2\alpha}$. By (H15) we have $p_u = k + \frac{\alpha}{2}$.

If $\alpha > (5\rho - 2\theta - 3)q$, then the SaaS vendor prices at boundary price $p_s^* = \alpha - (\rho - \theta)q$. Correspondingly, $t_{\sigma_1} = 1$. SP3' degenerates to equilibrium SP3'''[(Upgrade1, Upgrade2+SaaS), (New1, Upgrade2+SaaS)]. However, note that $(5\rho - 2\theta - 3)q > (2\rho - \theta - 1)q$. So the degenerated SP3''' does not occur in the α range we consider.

Case (2) Consider SP3''. Knowing it only serves in one period, the SaaS vendor's optimization problem becomes $2p_s(1 - t_{\sigma_1})$. The optimal interior solution is $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$. The conditions for $0 < t_{\sigma_1} < 1$ and $t_{\sigma_2} \geq 2$ are $2\alpha - (2\rho - \theta - 1)q \leq p_s < \alpha - (\rho - \theta)q$. Checking this condition we see the interior solution holds if $\alpha \leq \frac{(3\rho - \theta - 2)q}{3} < (\rho - 1)q$.

Given that OG users choose (Upgrade1+SaaS, Upgrade2), in order for NG users to prefer (New1+SaaS, Upgrade2) rather than (SaaS, New2), we need $(\rho q + 2k)t_{\sigma_1} - p_u \geq \int_0^{t_{\sigma_1}} (\theta q + \alpha t + k - p_s)dt$, which is the same condition as (H14). In order for NG users to prefer (New1+SaaS, Upgrade2) rather than (SaaS, SaaS), we need $(\rho q + 2k)t_{\sigma_1} - p_n + (2\rho - 1)q + 2k - p_u \geq \int_0^{t_{\sigma_1}} (\theta q + \alpha t + k - p_s)dt + \int_{t_{\sigma_1}}^2 (\theta q + \alpha t + k - p_s)dt$; that is, $p_u + p_n \leq (2\rho - \theta - 1)q + k + p_s - \frac{3\alpha}{2} + \frac{[p_s + (\rho - \theta)q + 2k][p_s + (\rho - \theta)q]}{2\alpha}$ (H17). Given that NG users choose (New1+SaaS, Upgrade2), in order for the OG users to prefer (Upgrade1+SaaS, Upgrade2) rather than (Old+SaaS, Upgrade2), we need $(\rho q + 2k)t_{\sigma_1} - p_u \geq (q + k)t_{\sigma_3} + \int_{t_{\sigma_3}}^{t_{\sigma_1}} (\theta q + \alpha t + k - p_s)dt$, which is the same condition as (H16).

When $\alpha \leq \frac{(3\rho - \theta - 2)q}{3} < (\rho - 1)q$, (H14) binds and we have $p_u = \frac{[(\rho - \theta)q + \alpha + 4k][(\rho - \theta)q + \alpha]}{8\alpha}$ and $p_n = \frac{(3\rho - \theta - 2)q}{2} - \alpha + k$. Furthermore, $p_u < p_n$.

Now consider the boundary solution. If $\frac{(3\rho - \theta - 2)q}{3} < \alpha \leq (\rho - 1)q$, substituting $p_s^* = 2\alpha - (2\rho - \theta - 1)q$ into (H14) we have $p_u = \frac{[2\alpha - (\rho - 1)q + 2k][2\alpha - (\rho - 1)q]}{2\alpha}$, and by (H17), $p_n = k + \frac{\alpha}{2}$.

Case (3) Consider SP3'''. Knowing it only serves in one period, the SaaS vendor's optimization problem becomes $2p_s(2 - t_{\sigma_2})$. The optimal interior solution is $p_s^* = \frac{2\alpha - (2\rho - \theta - 1)q}{2}$. The conditions for $t_{\sigma_1} \geq 1$ and $1 < t_{\sigma_2} < 2$ are $\alpha - (\rho - \theta)q \leq p_s < 2\alpha - (2\rho - \theta - 1)q$ (H18). Checking this condition we can verify that the interior solution does not hold. So the SaaS vendor prices at boundary price $p_s^* = \alpha - (\rho - \theta)q$. Substituting p_s^* into (H14) we have $p_u = \frac{[2\alpha - (\rho - 1)q + 2k][2\alpha - (\rho - 1)q]}{2\alpha}$. By (H15) we have $p_n = k + \frac{\alpha}{2}$.

We see that in the range $(\rho - \theta)q \leq \alpha < (2\rho - \theta - 1)q$, there are two equilibrium strategies: one symmetric (SP3') and one asymmetric (SP3'' or SP3'''). It is worth noting that if an equilibrium pricing strategy consists of boundary price, then the equilibrium is unstable because the vendor can easily deviate from the boundary pricing strategy by lowering its price a little bit, and then end up with entering the feasible pricing region of the other equilibrium. If an equilibrium pricing strategy consists of interior solution, it emerges as the final stable equilibrium at which both vendors have no incentive to deviate given the other vendor's strategy. Comparing the equilibrium profits under the different regions, we can establish the equilibrium outcome in the two-period model. We summarize and present the results in Proposition 7, where K'_2 and K'_3 are determined by solving $\pi_{perp}^{SP3''} = \pi_{perp}^{SP2'}$ and $\pi_{perp}^{SP3'''} = \pi_{perp}^{SP2'}$ in their respective segments. We omit their lengthy mathematical expressions here. In summary, we obtain the following equilibrium outcome.

Proposition 7 (Equilibrium Outcome in the Two-Period Model)

(a) (Entry Deterrence Equilibrium) If $\alpha \leq (\rho - \theta)q$ and $k > K'_1$, the perpetual software vendor deters the SaaS vendor's entry in both periods. The equilibrium user strategy is [(Upgrade1, Upgrade2), (New1, Upgrade2)]. The perpetual software vendor's equilibrium prices are $p_u^* = (\rho - \theta)q + k - \frac{\alpha}{2}$ and $p_n^* = (2\rho - \theta - 1)q + k - \frac{3\alpha}{2}$.

(b) (Market Segmentation Equilibrium) If i) $\alpha \leq (\rho - \theta)q$ and $k \leq K'_1$, or ii) $(\rho - \theta)q < \alpha \leq \frac{(3\rho - \theta - 2)q}{3}$ and $k \leq K'_2$, or iii) $\frac{(3\rho - \theta - 2)q}{3} < \alpha < (2\rho - \theta - 1)q$, and $k \leq K'_3$, the perpetual software vendor and the SaaS vendor segment the market. The equilibrium user strategy is [(Upgrade1, Upgrade2), (SaaS, SaaS)], and the equilibrium prices are as follows:

$$\text{If } \alpha \leq \frac{3(\rho - 1)q}{2}, \text{ then } p_u^* = (\rho - 1)q, p_n^* = \max[\frac{3(\rho - 1)q}{2} + 2k - \frac{3\alpha}{2}, 2(\rho - 1)q + 4k], \text{ and } p_s^* = \frac{(2\rho - \theta - 1)q}{2} + k + \alpha.$$

$$\text{If } \alpha > \frac{3(\rho - 1)q}{2}, \text{ then } p_u^* = (\rho - 1)q, p_n^* = 4k + 2\alpha - 2(\rho - 1)q, \text{ and } p_s^* = 2\alpha + k - (2\rho - \theta - 1)q.$$

(c) (Sequential Dominance Equilibrium) i) If $(\rho - \theta)q < \alpha \leq \frac{(3\rho - \theta - 2)q}{3}$ and $k > K'_2$, the perpetual software vendor and the SaaS vendor sequentially serve the market. The equilibrium user strategy is [(Upgrade1+SaaS, Upgrade2), (New1+SaaS, Upgrade2)]. The equilibrium prices are: $p_u^* = \frac{[(\rho - \theta)q + \alpha + 4k][(\rho - \theta)q + \alpha]}{8\alpha}$, $p_n^* = \frac{(3\rho - \theta - 2)q}{2} - \alpha + k$, and $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$.

ii) If $\frac{(3\rho-\theta-2)q}{3} < \alpha < (2\rho - \theta - 1)q$ and $k > K'_3$, the perpetual software vendor and the SaaS vendor sequentially serve the market. The equilibrium user strategy is [(Upgrade1+SaaS, Upgrade2+SaaS), (New1+SaaS, Upgrade2+SaaS)]. The equilibrium prices are as follows:

$$\text{If } \alpha \leq (\rho - 1)q, \text{ then } p_u^* = \frac{[(\rho-2\theta+1)q+3\alpha+8k][(\rho-2\theta+1)q+3\alpha]}{32\alpha}, p_n^* = \frac{[(5\rho-2\theta-3)q-\alpha+8k][(5\rho-2\theta-3)q-\alpha]}{32\alpha}, \text{ and } p_s^* = \frac{3\alpha-(3\rho-2\theta-1)q}{4}.$$

$$\text{If } \alpha > (\rho - 1)q, \text{ then } p_u^* = p_n^* = \frac{5\alpha^2+8\alpha k+(24k\rho-16k\theta-2\alpha\rho-4\alpha\theta+6\alpha-8k)q+(13\rho^2-12\rho\theta+4\theta^2-14\rho+4\theta+5)q^2}{32\alpha}, \text{ and } p_s^* = \frac{3\alpha-(3\rho-2\theta-1)q}{4}.$$

Appendix I

SaaS Vendor's Quality Improvement Cost

Proposition 8 (Entry Deterrence Equilibrium with c_α) The perpetual software vendor deters the SaaS vendor's entry when the network effect is strong enough or when the SaaS quality improvement cost is high enough. The equilibrium user strategy is SP1 (Upgrade, New), where the OG users upgrade and the NG users adopt the new perpetual software. The equilibrium prices are as follows:

(a) If $c_\alpha \leq \frac{\alpha}{2} + (\theta - 1)q$ and $k \geq K'_1 = \frac{\alpha-(\rho-2\theta+1)q-2c_\alpha}{2}$, then $p_u^{c_\alpha} = p_n^{c_\alpha} = (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$.

(b) If $c_\alpha > \frac{\alpha}{2} + (\theta - 1)q$, then $p_u^{c_\alpha} = (\rho - 1)q + k$ and $p_n^{c_\alpha} = (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$.

Proof. Consider SP1 (Upgrade, New). Similar to the Proof of Proposition 2, we must ensure that the OG users prefer upgrading to the new version rather than continuing to use the old version, which requires $\rho q + 2k - p_u \geq q + k$; that is, $p_u \leq (\rho - 1)q + k$ (I1). Meanwhile, the perpetual software vendor needs to make sure that OG users prefer upgrading rather than adopting SaaS, even if the SaaS price is reduced to the lowest level $p_s = c_\alpha$. That is, the entry deterrence condition is $\rho q + 2k - p_u \geq \int_0^1 (\theta q + \alpha t + k - c_\alpha) dt$, so that $p_u \leq (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$ (I2). Similarly, to ensure that NG users prefer the new perpetual software to the SaaS at $p_s = c_\alpha$, the condition is $\rho q + 2k - p_n \geq \int_0^1 (\theta q + \alpha t + k - c_\alpha) dt$; that is, $p_n \leq (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$ (I3).

If $c_\alpha \leq \frac{\alpha}{2} + (\theta - 1)q$, (I2) is binding. Because $p_u \leq p_n$, by (I2) and (I3) the perpetual software vendor sets the prices at respective upper bounds: $p_n^{SP1} = p_u^{SP1} = (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$. Consequently, we get the perpetual software vendor's profit $\pi_{perp}^{SP1} = 2(\rho - \theta)q + 2k - \alpha + 2c_\alpha$.

If $c_\alpha > \frac{\alpha}{2} + (\theta - 1)q$, (I1) is binding. By (I2) and (I3) we have $p_u^{SP1} = (\rho - 1)q + k$ and $p_n^{SP1} = (\rho - \theta)q + k - \frac{\alpha}{2} + c_\alpha$. Consequently, we get the perpetual software vendor's profit $\pi_{perp}^{SP1} = (2\rho - \theta - 1)q + 2k - \frac{\alpha}{2} + c_\alpha$.

Consider SP2 (Upgrade, SaaS). Similar to the Proof of Proposition 3, we have $p_u \leq (\rho - 1)q$ (I4); $p_u \leq p_s + (\rho - \theta)q - k - \frac{\alpha}{2}$ (I5); and $p_n \geq p_s + (\rho - \theta)q + k - \frac{\alpha}{2}$ (I6).

To maximize its profit, the perpetual software vendor sets p_n as high as possible so that the SaaS vendor can also charge a high enough price p_s , which in turn allows the perpetual software vendor to charge a high upgrade price p_u . As a result, the perpetual software vendor charges $p_u = (\rho - 1)q$ to make the OG users' IC constraint (I4) binding. If $\alpha \leq 2(\rho - 1)q$, the SaaS vendor charges as much as $p_s^{SP2} = (\theta - 1)q + k + \frac{\alpha}{2}$ by (I5), and by (I6) $p_n^{SP2} = (\rho - 1)q + 2k$. If $\alpha > 2(\rho - 1)q$, then the boundary solution $p_s^{SP2} = \alpha + k - (\rho - \theta)q$ as specified in Table C1 holds. By (I4) and (I5) $p_u^{SP2} = (\rho - 1)q$ and by (I6) $p_n^{SP2} = \frac{\alpha}{2} + k$. So $\pi_{perp}^{SP2} = (\rho - 1)q$.

Finally, we compare the perpetual software vendor's profits under SP1 and SP2. We can show that, if $c_\alpha \leq \frac{\alpha}{2} + (\theta - 1)q$, then $\pi_{perp}^{SP1} > \pi_{perp}^{SP2}$ if $k > \frac{\alpha-(\rho-2\theta+1)q-2c_\alpha}{2}$. If $c_\alpha > \frac{\alpha}{2} + (\theta - 1)q$, then $\pi_{perp}^{SP1} > \pi_{perp}^{SP2}$.

Appendix J

OG User's Switching Cost

Proposition 9 (Equilibria with OG User Switching Cost) Both the SaaS quality improvement rate α and users' switching cost c affect the equilibrium outcome as follows:

(a) (Entry Deterrence Equilibrium) If $\alpha \leq A_1$, the perpetual software vendor deters the SaaS vendor's entry. The equilibrium user strategy is SP1 (Upgrade, New). The perpetual software vendor's equilibrium prices are $p_u^* = p_n^* = (\rho - \theta)q - \frac{\alpha}{2}$.

(b) (Market Segmentation Equilibrium) The perpetual software vendor and the SaaS vendor segment the market. The equilibrium user strategy is SP2 (Upgrade, SaaS).

If i) $A_1 < \alpha \leq A_2$, or ii) $\alpha > A_2$ and $C_2 < c \leq C_1$, then equilibrium prices are $p_u^* = p_n^* = (\rho - 1)q$ and $p_s^* = (\theta - 1)q + \frac{\alpha}{2}$.

If $\alpha \leq A_3$ and $c \leq C_2$, then equilibrium prices are $p_u^* = (\rho - 1)q$, $p_n^* = \frac{\alpha}{2} - \sqrt{2\alpha c}$, and $p_s^* = \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$.

(c) (Competitive Lock-in Equilibrium) If $\alpha > A_2$ and $c > C_1$, the perpetual software vendor serves the OG users over the whole time interval $[0, 1]$ and NG users in the time interval $[0, \frac{\alpha + (\rho - \theta)q}{2\alpha}]$. The SaaS vendor serves the NG users in the time interval $[\frac{\alpha + (\rho - \theta)q}{2\alpha}, 1]$. The equilibrium user strategy is SP7 (Upgrade, New+SaaS). The equilibrium prices are $p_u^* = p_n^* = \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha}$ and $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$.

(d) (Sequential Dominance Equilibrium) If $\alpha > A_3$ and $c \leq C_2$, the perpetual software vendor serves both OG and NG users in the time interval $[0, \frac{\alpha + (\rho - \theta)q}{2\alpha}]$, and the SaaS vendor serves both OG and NG users in the time interval $[\frac{\alpha + (\rho - \theta)q}{2\alpha}, 1]$. The equilibrium user strategy is SP6 (Upgrade+SaaS, New+SaaS). The equilibrium prices are $p_u^* = \frac{(\rho - 1)q[\alpha - (\theta - 1)q]}{2\alpha}$, $p_n^* = \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha}$, and $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$.

Our proof involves several steps. First, given user strategies, we analyze four sub-game perfect equilibria and the corresponding vendor prices and profits. Then we derive the final equilibrium outcome under different market conditions.

Entry Deterrence Strategy

Note that SP1 (Upgrade, New) can only occur when $\alpha \leq (\rho - \theta)q$. That is, the quality of SaaS does not exceed the quality of the new perpetual software at the end of the product life cycle.

Given that NG users purchase the new perpetual software, OG users prefer to upgrade rather than continue to use the old version. So we have $p_u \leq (\rho - 1)q$ (J1). Also, OG users prefer to upgrade rather than opt for SaaS. Note that moving to SaaS incurs additional switching costs c . So we get $p_u \leq (\rho - \theta)q - \frac{\alpha}{2} + c$ (J2).

Given that OG users upgrade, NG users prefer to buy the new perpetual software rather than SaaS. This situation gives us $p_n \leq (\rho - \theta)q - \frac{\alpha}{2}$ (J3). In addition, we have the constraint $p_n \geq p_u$.

Putting all these constraints together, we get the perpetual software vendor's prices $p_u^{SP1} = p_n^{SP1} = (\rho - \theta)q - \frac{\alpha}{2}$ and profit $\pi_{perp}^{SP1} = 2(\rho - \theta)q - \alpha$.

Market Segmentation Strategy

Consider SP2 (Upgrade, SaaS), where the perpetual software vendor allows the SaaS vendor to enter the market. It can happen under both $\alpha \leq (\rho - \theta)q$ and $\alpha > (\rho - \theta)q$.

Case (1) $\alpha \leq (\rho - \theta)q$. Given that NG users choose SaaS, we need to ensure that, for OG users, upgrading is better than using the old version and also better than SaaS. Thus, (J1) and $p_s \geq p_u - (\rho - \theta)q + \frac{\alpha}{2} - c$ (J4) must hold. Similarly, NG users prefer SaaS to the new perpetual

software, and so $p_s \leq p_n - (\rho - \theta)q + \frac{\alpha}{2}$ (J5). In addition, $p_n \geq p_u$. So we get $p_u^{SP2} = p_n^{SP2} = (\rho - 1)q$, $p_s^{SP2} = (\theta - 1)q + \frac{\alpha}{2}$. Vendor profits are $\pi_{perp}^{SP2} = (\rho - 1)q$ and $\pi_{SaaS}^{SP2} = (\theta - 1)q + \frac{\alpha}{2}$.

Case (2) $\alpha > (\rho - \theta)q$. When α is large, the SaaS becomes competitive, and switching becomes possible. We first derive the non-switching (NS) condition for OG users. Conditional on the fact that OG users switch, the switching time is when the net payoff from SaaS exceeds the net payoff from the new version of perpetual software. Similar to the baseline case, $t_{s1} = \frac{p_s - (\theta - 1)q}{\alpha}$. Taking into account the switching cost, the condition for switching is $\int_{t_{s1}}^1 (\theta q + \alpha t - p_s)dt - \rho q(1 - t_{s1}) \geq c$. Substituting into t_{s1} and solving this inequality, we get $p_s \geq \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$ (NS).

We can verify that the SaaS price derived in Case (1) satisfies this (NS) condition when $\sqrt{2\alpha c} \geq \frac{\alpha}{2} - (\rho - 1)q$; that is, $c \geq \frac{[\alpha - 2(\rho - 1)q]^2}{8\alpha} \doteq C_2$. Therefore the same optimal solutions apply.

When $c < C_2$, however, the (NS) condition is binding, so $p_s = \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$. Reexamining the incentive compatibility conditions (J1), (J4), and (J5), we get $p_u^{SP2} = (\rho - 1)q$, $p_n^{SP2} = \frac{\alpha}{2} - \sqrt{2\alpha c}$ by (J3), and $p_u^{SP2} < p_n^{SP2}$. The vendor's profits are $\pi_{perp}^{SP2} = (\rho - 1)q$ and $\pi_{SaaS}^{SP2} = \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$.

Competitive Lock-In Strategy

Consider a new strategy pair (Upgrade, New+SaaS). We denote it as SP7. It occurs under the condition $\alpha > (\rho - \theta)q$, where the SaaS quality outperforms the perpetual software quality at some time $t \in [0,1]$. To ensure that OG users do not switch, the (NS) condition must hold. And to ensure that NG users switch, the net payoff from SaaS must be higher than the net payoff from the new perpetual software by time $t = 1$; that is, $\theta q + \alpha - p_s \geq \rho q$. So $p_s \leq \alpha - (\rho - \theta)q$ (J6). In addition, NG users switch at $t_{s3} = \frac{(\rho - \theta)q + p_s}{\alpha}$. The SaaS vendor's profit thus is expressed as $p_s(1 - t_{s3})$, and the optimal SaaS price is $p_s^* = \frac{\alpha - (\rho - \theta)q}{2}$. Accordingly, the optimal switching time is $t_{s3}^* = \frac{\alpha + (\rho - \theta)q}{2\alpha}$. There are two cases:

Case (1) When $c \geq \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, the interior solution $p_s^{SP7} = \frac{\alpha - (\rho - \theta)q}{2}$ satisfies both (NS) and (J6). We now check the incentive compatibility conditions for both groups of users. Given that OG users upgrade, NG users prefer New+SaaS over SaaS if $\rho q t_{s1}^* - p_n + \int_{t_{s1}^*}^1 (\theta q + \alpha t - p_s^*)dt \geq \int_0^1 (\theta q + \alpha t - p_s^*)dt$ (J7). Substituting into p_s^{SP7} and t_{s3}^* and simplifying the condition, we get $p_n \leq \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha} \doteq p_1$. Similarly, given that NG users choose New+SaaS, OG users prefer Upgrade over Old+SaaS if $\rho q - p_u \geq q t_{s1}^* + \int_{t_{s1}^*}^1 (\theta q + \alpha t - p_s^*)dt - c$ (J8), where t_{s1}^* is the switching time if OG users switch from the old version of perpetual software to SaaS, and t_{s1}^* is given by $\theta q + \alpha t_{s1}^* - p_s^* = q$. Using p_s^{SP7} , we have $t_{s1}^* = \frac{\alpha - (\rho + \theta - 2)q}{2\alpha}$. If $\alpha < (\rho + \theta - 2)q$, (J8) is satisfied. So $p_n = p_u = \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha}$. If $\alpha \geq (\rho + \theta - 2)q$, $t_{s1}^* > 0$. Substituting t_{s1}^* into (J8) we get $p_u \leq (\rho - 1) - \frac{[\alpha + (\rho + \theta - 2)q]^2}{8\alpha} + c \doteq p_2$. When $c = \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, $p_2 < p_1$. Because p_2 linearly increases in c , there is a threshold value $c^* = \frac{[\alpha + (\rho - \theta)q]^2 + [\alpha + (\rho + \theta - 2)q]^2}{8\alpha} - (\rho - 1)q$ such that, for $\frac{[\alpha - (\rho - \theta)q]^2}{8\alpha} \leq c < c^*$, $p_1 > p_2$; thus, $p_n^{SP7} = \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha}$ and $p_u^{SP7} = (\rho - 1)q - \frac{[\alpha + (\rho + \theta - 2)q]^2}{8\alpha} + c$; and for $c \geq c^*$, $p_1 < p_2$; thus, $p_n^{SP7} = p_u^{SP7} = \frac{[\alpha + (\rho - \theta)q]^2}{8\alpha}$.

Case (2) When $c < \frac{[\alpha - (\rho - \theta)q]^2}{8\alpha}$, we have a boundary solution $p_s^{SP7} = \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$; accordingly, the switching time becomes $t_{s3}^* = \frac{\alpha - \sqrt{2\alpha c}}{\alpha}$. We next check users' incentive compatibility conditions. Condition (J7) becomes $p_n \leq \frac{\alpha}{2} - \sqrt{2\alpha c} + c \doteq p_1$. For condition (J8), $t_{s1}^* = \frac{p_s^* - (\theta - 1)q}{\alpha} = \frac{\alpha - (\rho - 1)q - \sqrt{2\alpha c}}{\alpha}$. If $\alpha < (\rho - 1)q$, or if $\alpha \geq (\rho - 1)q$ and $c > \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$, (J8) is satisfied. In these cases, $p_n^{SP7} = p_u^{SP7} = \frac{\alpha}{2} - \sqrt{2\alpha c} + c$. If $\alpha \geq (\rho - 1)q$ and $c < \frac{[\alpha - (\rho - 1)q]^2}{2\alpha}$, $t_{s1}^* > 0$, substituting t_{s1}^* into (J8), we get $p_u \leq (\rho - 1) - \frac{[(\rho - 1)q + \sqrt{2\alpha c}]^2}{2\alpha} + c \doteq p_2$. We can verify that $p_1 > p_2$. Hence, $p_n^{SP7} = \frac{\alpha}{2} - \sqrt{2\alpha c} + c$, $p_u^{SP7} = (\rho - 1) - \frac{[(\rho - 1)q + \sqrt{2\alpha c}]^2}{2\alpha} + c$, and $p_n^{SP7} > p_u^{SP7}$.

Sequential Dominance Strategy

This strategy pair is SP6 (Upgrade+SaaS, New+SaaS). It occurs under the condition $\alpha > (\rho - \theta)q$. To ensure that OG users switch to SaaS, the switching condition is $p_s < \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$ (J9), and note that when this condition holds, NG users also switch. Similar to the

baseline model, the switching time is $t_{s3} = \frac{(\rho-\theta)q+p_s}{\alpha}$. The SaaS vendor's profit thus is expressed as $2p_s(1 - t_{s3})$, and the optimal SaaS price is $p_s^* = \frac{\alpha-(\rho-\theta)q}{2}$. Accordingly, the optimal switching time is $t_{s3}^* = \frac{\alpha+(\rho-\theta)q}{2\alpha}$. We get three cases:

Case (1) When $c < \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$, the internal optimal solution $p_s^{SP6} = \frac{\alpha-(\rho-\theta)q}{2}$ satisfies (J9). The solution is the same as the baseline model, as in Proposition 4.

Case (2) When $\frac{[\alpha-(\rho-\theta)q]^2}{8\alpha} \leq c < \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$, we derive the boundary solution $p_s^{SP6} = \alpha - (\rho - \theta)q - \sqrt{2\alpha c}$; accordingly, the switching time becomes $t_{s3}^* = \frac{\alpha-\sqrt{2\alpha c}}{\alpha}$. We reexamine the incentive compatibility conditions. Given that OG users choose Upgrade+SaaS, NG users prefer New+SaaS over SaaS if $\rho q t_{s3}^* - p_n + \int_{t_{s3}^*}^1 (\theta q + \alpha t - p_s^*) dt \geq \int_0^1 (\theta q + \alpha t - p_s^*) dt$. So we get $p_n \leq \frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha} \doteq p_1$. Given that NG users choose New+SaaS, OG users prefer Upgrade+SaaS over Old+SaaS if $\rho q t_{s2}^* - p_u + \int_{t_{s2}^*}^1 (\theta q + \alpha t - p_s^*) dt - c \geq q t_{s1}^* + \int_{t_{s1}^*}^1 (\theta q + \alpha t - p_s^*) dt - c$ (J10), where $t_{s1}^* = \frac{p_s^{SP6}-(\rho-1)q}{\alpha} = \frac{\alpha-(\rho-1)q-\sqrt{2\alpha c}}{\alpha}$. We note that $t_{s1}^* < 0$ when $\alpha < (\rho + \theta - 2)q$, or $\alpha > (\rho + \theta - 2)q$ and $\frac{[\alpha-(\rho-1)q]^2}{2\alpha} \leq c < \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$. So (J10) is satisfied and $p_n^{SP6} = p_u^{SP6} = \frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha}$. When $\alpha \geq (\rho + \theta - 2)q$ and $\frac{[\alpha-(\rho-\theta)q]^2}{8\alpha} \leq c < \frac{[\alpha-(\rho-1)q]^2}{2\alpha}$, $t_{s1}^* > 0$. Substituting t_{s1}^* into (J10), we get $p_u \leq \frac{(\rho-1)q(\alpha-\sqrt{2\alpha c})}{\alpha} - \frac{[(\rho-1)q]^2}{2\alpha} \doteq p_2$. Because $p_1 - p_2 = \frac{[\alpha-(\rho-1)q-\sqrt{2\alpha c}]^2}{2\alpha} > 0$, we have $p_n^{SP6} = \frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha}$, $p_u^{SP6} = \frac{(\rho-1)q(\alpha-\sqrt{2\alpha c})}{\alpha} - \frac{[(\rho-1)q]^2}{2\alpha}$, and $p_n^{SP6} > p_u^{SP6}$.

Case (3) When $c \geq \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$, the condition (J9) does not hold. Thus, SP6 does not appear.

Profit Comparison in All Parameter Regions

To see which strategy pair is the equilibrium, we need to compare the vendor's profits. When $\alpha \leq (\rho - \theta)q$, both SP1 and SP2 are possible; when $\alpha > (\rho - \theta)q$, SP2, SP6, and SP7 are possible. Using Table 3, we have in total 10 parameter regions to study. In the following, we examine one region to show how we obtain the equilibrium; for all the rest of the comparisons, the analysis is similar.

Consider the parameter region $\alpha \geq (\rho - \theta)q$, $\max\left\{\frac{[\alpha-(\rho-1)q]^2}{2\alpha}, \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}\right\} < c < \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$. In this region, SP2, SP6, and SP7 are all feasible strategies. Vendor profits are $\pi_{perp}^{SP2} = (\rho - 1)q$, $\pi_{SaaS}^{SP2} = (\theta - 1)q + \frac{\alpha}{2}$ in SP2, $\pi_{perp}^{SP6} = \frac{[\alpha-\sqrt{2\alpha c}]^2}{\alpha}$, $\pi_{SaaS}^{SP6} = \frac{2[\alpha-(\rho-\theta)q-\sqrt{2\alpha c}]\sqrt{2\alpha c}}{\alpha}$ in SP6, and $\pi_{perp}^{SP7} = \frac{[\alpha+(\rho-\theta)q]^2}{4\alpha}$, $\pi_{SaaS}^{SP7} = \frac{[\alpha-(\rho-\theta)q]^2}{4\alpha}$ in SP7, respectively.

We first compare SP6 and SP7. Because $\Delta\pi_{perp}^{SP7-6} = \frac{[3\alpha+(\rho-\theta)q-2\sqrt{2\alpha c}][-\alpha+(\rho-\theta)q+2\sqrt{2\alpha c}]}{4\alpha} > 0$, the perpetual software vendor prefers SP7 to SP6. For the SaaS vendor, we find that $\partial\Delta\pi_{SaaS}^{SP7-6} / \partial c = \frac{2[(\rho-\theta)q+2\sqrt{2\alpha c}-\alpha]}{\sqrt{2\alpha c}} > 0$. If $(\rho - \theta)q < \alpha < (\rho + \theta - 2)q$, $\underline{c} = \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$ and $\bar{c} = \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$. If $\alpha \geq (\rho + \theta - 2)q$, $\underline{c} = \frac{[\alpha-(\rho-1)q]^2}{2\alpha}$ and $\bar{c} = \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$. We can show that $\Delta\pi_{SaaS}^{SP7-6} < 0$ at \underline{c} and $\Delta\pi_{SaaS}^{SP7-6} > 0$ at \bar{c} . So a value C_1 must exist in this parameter region such that $\Delta\pi_{SaaS}^{SP7-6} = 0$ at C_1 . Solving the equation, we get $C_1 = \frac{(\sqrt{2}+1)^2[\alpha-(\rho-\theta)q]^2}{16\alpha}$. For $c < C_1$, $\Delta\pi_{SaaS}^{SP7-6} < 0$, meaning that the SaaS vendor prefers SP6 to SP7 and so reduces its price to deviate to SP6. Meanwhile, for $c > C_1$, $\Delta\pi_{SaaS}^{SP7-6} > 0$ meaning that the SaaS vendor prefers SP7 to SP6.

We next compare SP2 with SP6 when $c < C_1$, and we compare SP2 with SP7 when $c > C_1$.

Case (1) $c < C_1$. For the SaaS vendor, $\partial\Delta\pi_{SaaS}^{SP6-2} / \partial c = \frac{-2[(\rho-\theta)q+2\sqrt{2\alpha c}-\alpha]}{\sqrt{2\alpha c}} < 0$; and $\Delta\pi_{SaaS}^{SP6-2} < 0$ at $c = \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$. Because in this region all $c \geq \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$, we conclude that $\Delta\pi_{SaaS}^{SP6-2} < 0$ in the whole region. Thus, the SaaS vendor always prefers SP2. For the perpetual software vendor, $\Delta\pi_{perp}^{SP6-2} = \alpha + 2c - 2\sqrt{2\alpha c} - (\rho - 1)q$. We solve $\Delta\pi_{perp}^{SP6-2} = 0$ and get two solutions: $c_1 = \frac{[\sqrt{\alpha}-\sqrt{(\rho-1)q}]^2}{2}$ and $c_2 = \frac{[\sqrt{\alpha}+\sqrt{(\rho-1)q}]^2}{2}$. We can further prove that $c_1 < \frac{[\alpha-(\rho-1)q]^2}{2\alpha}$ and $c_2 > C_1$, and so both roots are outside this region. Hence, $\Delta\pi_{perp}^{SP6-2} < 0$, meaning that the perpetual software vendor prefers SP2. We conclude that when $c < C_1$, the final equilibrium is SP2.

Case (2) $c > C_1$. For the SaaS vendor, $\partial\Delta\pi_{SaaS}^{SP7-2} / \partial \alpha = \frac{-4[\alpha^2+(\rho-\theta)^2q^2]}{16\alpha^2} < 0$; and $\Delta\pi_{SaaS}^{SP7-2} < 0$ at $\alpha = (\rho - \theta)q$. Because in this region all $\alpha \geq (\rho - \theta)q$, we conclude that $\Delta\pi_{SaaS}^{SP7-2} < 0$ in the whole region. Thus, the SaaS vendor always prefers SP2. For the perpetual software

vendor, $\Delta\pi_{perp}^{SP7-2} = \frac{[\alpha+(\rho-\theta)q]^2}{4\alpha} - (\rho-1)q$. We can show that $\Delta\pi_{perp}^{SP7-2} = (1-\theta)q < 0$ at $\alpha = (\rho-\theta)q$, which is the smallest α in this region, and that $\partial\Delta\pi_{perp}^{SP7-2} / \partial\alpha = \frac{4[\alpha+(\rho-\theta)q][\alpha-(\rho-\theta)q]}{16\alpha^2} > 0$. Solving $\Delta\pi_{perp}^{SP7-2} = 0$ for α , we get two solutions: $A_1 = (\rho+\theta-2)q - 2\sqrt{(\rho-1)(\theta-1)q}$ and $A_2 = (\rho+\theta-2)q + 2\sqrt{(\rho-1)(\theta-1)q}$. Note that $A_1 < (\rho-\theta)q$, so it falls outside of the region, and $(\rho+\theta-2)q < A_2 < 2(\rho-1)q$, so it falls within the region. Therefore, when $\alpha < A_2$, $\Delta\pi_{perp}^{SP7-2} < 0$, meaning that the perpetual software vendor prefers SP2 and that, in this sub-region, SP2 is the equilibrium outcome. When $\alpha \geq A_2$, $\Delta\pi_{perp}^{SP7-2} > 0$, meaning that the perpetual software vendor prefers SP7. The perpetual software vendor thus reduces prices to deviate from SP2 to SP7. We conclude that in this sub-region, SP7 is the equilibrium outcome.

Finally, after combining all the conditions and equilibrium results, we obtain the four equilibria shown in Proposition 9 and Table J1.

Table J1. Parameter Conditions, Prices, and Profits Under Switching Cost Model				
(a) Parameter Conditions with Switching Costs				
Strategy Pairs	Regions	Parameter Conditions		
SP1	1	$\alpha < (\rho-\theta)q$		
SP2	2	(1) $\alpha < (\rho-\theta)q$; (2) $\alpha \geq (\rho-\theta)q, c \geq C_2 = \frac{[\alpha-2(\rho-1)q]^2}{8\alpha}$		
	3	$\alpha \geq (\rho-\theta)q, c < C_2 = \frac{[\alpha-2(\rho-1)q]^2}{8\alpha}$		
SP6	4	$(\rho-\theta)q \leq \alpha < (\rho+\theta-2)q, c < \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$		
	5	$\alpha \geq (\rho+\theta-2)q, c < \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$		
	6	(1) $(\rho-\theta)q \leq \alpha < (\rho+\theta-2)q$; (2) $\alpha \geq (\rho+\theta-2)q, \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha} \leq c < \frac{[\alpha-(\rho-\theta)q]^2}{2\alpha}$		
	7	$\alpha > (\rho+\theta-2)q, \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha} < c < \frac{[\alpha-(\rho-1)q]^2}{2\alpha}$		
SP7	8	(1) $(\rho-\theta)q \leq \alpha < (\rho+\theta-2)q, c \geq \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$; (2) $\alpha \geq (\rho+\theta-2)q, c \geq c^*$		
	9	$\alpha > (\rho+\theta-2)q, \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha} \leq c < c^* = \frac{[\alpha+(\rho-\theta)q]^2 + [\alpha+(\rho+\theta-2)q]^2}{8\alpha} - (\rho-1)q$		
	10	(1) $(\rho-\theta)q \leq \alpha < (\rho-1)q$; (2) $(\rho-\theta)q \leq \alpha < (\rho+\theta-2)q, \frac{[\alpha-(\rho-1)q]^2}{2\alpha} \leq c < \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$		
	11	(1) $(\rho-1)q \leq \alpha < (\rho+\theta-2)q, c < \frac{[\alpha-(\rho-1)q]^2}{2\alpha}$; (2) $\alpha \geq (\rho+\theta-2)q, c < \frac{[\alpha-(\rho-\theta)q]^2}{8\alpha}$		
(b) Optimal Prices with Switching Costs				
Strategy Pairs	Regions	p_u	p_n	p_s
SP1	1	$(\rho-\theta)q - \frac{\alpha}{2}$	$(\rho-\theta)q - \frac{\alpha}{2}$	—
SP2	2	$(\rho-1)q$	$(\rho-1)q$	$(\theta-1)q + \frac{\alpha}{2}$
	3	$(\rho-1)q$	$\frac{\alpha}{2} - \sqrt{2\alpha c}$	$\alpha - (\rho-\theta)q - \sqrt{2\alpha c}$
SP6	4	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{\alpha-(\rho-\theta)q}{2}$
	5	$\frac{(\rho-1)q[\alpha-(\theta-1)q]}{2\alpha}$	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{\alpha-(\rho-\theta)q}{2}$
	6	$\frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha}$	$\frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha}$	$\alpha - (\rho-\theta)q - \sqrt{2\alpha c}$
	7	$\frac{(\rho-1)q(\alpha-\sqrt{2\alpha c})}{\alpha} - \frac{[(\rho-1)q]^2}{2\alpha}$	$\frac{[\alpha-\sqrt{2\alpha c}]^2}{2\alpha}$	$\alpha - (\rho-\theta)q - \sqrt{2\alpha c}$
SP7	8	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{\alpha-(\rho-\theta)q}{2}$
	9	$(\rho-1)q - \frac{[\alpha+(\rho+\theta-2)q]^2}{8\alpha} + c$	$\frac{[\alpha+(\rho-\theta)q]^2}{8\alpha}$	$\frac{\alpha-(\rho-\theta)q}{2}$
	10	$\frac{\alpha}{2} - \sqrt{2\alpha c} + c$	$\frac{\alpha}{2} - \sqrt{2\alpha c} + c$	$\alpha - (\rho-\theta)q - \sqrt{2\alpha c}$
	11	$(\rho-1)q - \frac{[(\rho-1)q + \sqrt{2\alpha c}]^2}{2\alpha} + c$	$\frac{\alpha}{2} - \sqrt{2\alpha c} + c$	$\alpha - (\rho-\theta)q - \sqrt{2\alpha c}$

(c) Optimal Profits with Switching Costs			
Strategy Pairs	Regions	π_{perp}	π_{SaaS}
SP1	1	$2(\rho - \theta)q - \alpha$	—
SP2	2	$(\rho - 1)q$	$(\theta - 1)q + \frac{\alpha}{2}$
	3	$(\rho - 1)q$	$\alpha - (\rho - \theta)q - \sqrt{2\alpha c}$
SP6	4	$\frac{[\alpha + (\rho - \theta)q]^2}{4\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$
	5	$\frac{[\alpha + (\rho - \theta)q]^2 + 4(\rho - 1)q[\alpha - (\theta - 1)q]}{8\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{2\alpha}$
	6	$\frac{[\alpha - \sqrt{2\alpha c}]^2}{\alpha}$	$\frac{2[\alpha - (\rho - \theta)q - \sqrt{2\alpha c}]\sqrt{2\alpha c}}{\alpha}$
	7	$\frac{[(\alpha - \sqrt{2\alpha c}) + 2(\rho - 1)q][\alpha - \sqrt{2\alpha c}] - [(\rho - 1)q]^2}{2\alpha}$	$\frac{2[\alpha - (\rho - \theta)q - \sqrt{2\alpha c}]\sqrt{2\alpha c}}{\alpha}$
SP7	8	$\frac{[\alpha + (\rho - \theta)q]^2}{4\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{4\alpha}$
	9	$(\rho - 1)q + c + \frac{\alpha + (\rho + \theta - 2)q}{4\alpha}$	$\frac{[\alpha - (\rho - \theta)q]^2}{4\alpha}$
	10	$\alpha - 2\sqrt{2\alpha c} + 2c$	$\frac{[\alpha - (\rho - \theta)q - \sqrt{2\alpha c}]\sqrt{2\alpha c}}{\alpha}$
	11	$(\theta - 1)q + \alpha - \frac{[(\rho - 1)q + \sqrt{2\alpha c}]^2}{2\alpha} - \sqrt{2\alpha c} + c$	$\frac{[\alpha - (\rho - \theta)q - \sqrt{2\alpha c}]\sqrt{2\alpha c}}{\alpha}$

Appendix K

Continuous NG User Arrival Model

We extend our model to account for NG users’ continuous arrival time. We still focus on the vendors’ price competition on the planning horizon $[0,1]$. The model setup is the same as the baseline model, except that we assume the NG users with mass 1 uniformly and continuously enter the market on the time interval $[0,1]$. Upon arrival, each NG user makes the software adoption decision for a limited use period, which is normalized to 1. Thus, users who arrive at $t < 1$ make a decision based on their expected utility from the software use in the period $[t, 1 + t]$. We use this model setup for several reasons. First, a decision period of the same length provides a fair comparison among all users. Second, the rapid technological obsolescence makes the software value in the far distant future negligible. To cope with the late arrival users’ decision making in the extended time period beyond $t = 1$, we assume that the SaaS software quality continues to increase at rate α after time 1. And at $t = 1$, the perpetual software vendor releases another “newer” software version with a higher quality. We assume the quality improvement between two major software releases remains the same (i.e., $(\rho - 1)q$). Therefore, the “newer” perpetual software’s quality can be calculated as $\rho q + (\rho - 1)q = (2\rho - 1)q$. The continuous user arrival model is depicted in Figure K1.

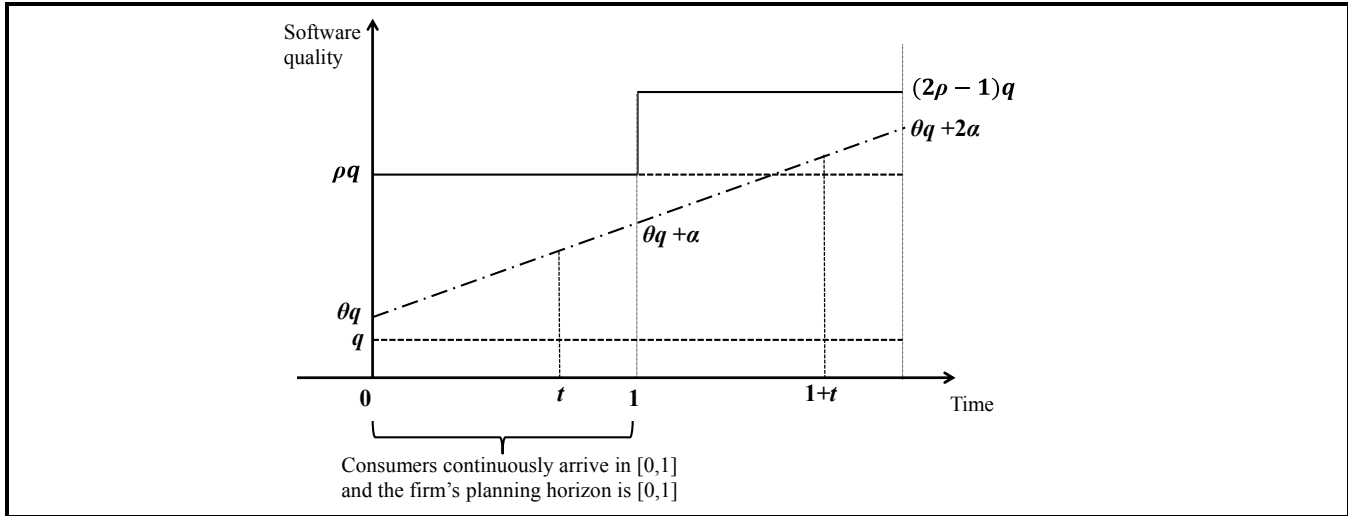


Figure K1. Software Quality Improvement Over Time

In such a dynamic market environment, the installed user base for a software product continues to change. Users who arrive at different times face different expected network values based on both the current number of users and the anticipated future number of users. Even if the current network size is observable, forming the expectation of future network growth is cognitively challenging because it depends on future users’ adoption decisions. We therefore omit the network effect in this continuous arrival model (i.e., $k = 0$).

All OG users’ strategies are the same as in the baseline model in the “User Utility Definition and Strategy Analysis” section of the paper. For each NG user with arrival time $t < 1$, we note five possible strategies.

New: The user purchases the new perpetual software at price p_n at time t and uses it over the entire period $[t, 1 + t]$. The utility is $\rho q - p_n$.

New+Newer: The user purchases the new perpetual software at price p_n at time t , uses it in $[t, 1]$, and then pays an upgrade price p_u to get the newer version at time 1 and uses it for the remaining period $[1, 1 + t]$. The utility is $\rho q(1 - t) - p_n + (2\rho - 1)qt - p_u$.

New + SaaS: The user purchases the new perpetual software at price p_n at time t and uses it in $[t, t_{s3}]$. It switches to SaaS in the period $[t_{s3}, 1 + t]$. The utility is $\rho q(t_{s3} - t) - p_n + \int_{t_{s3}}^{1+t} (\theta q + at - p_s) dt$.

SaaS: The user uses the SaaS software over the entire period $[t, 1 + t]$. The utility is $\int_t^{1+t} (\theta q + at - p_s) dt$.

SaaS+Newer: The user uses the SaaS software in the period $[t, 1]$, buys the newer version perpetual software at price p_n at time 1, and uses this software for the remaining period $[1, 1 + t]$. The utility is $\int_t^1 (\theta q + at - p_s) dt + (2\rho - 1)qt - p_n$.

Following a similar notion as in the baseline model, we solve this continuous user arrival model for equilibrium outcomes. The complete result derivation and proof is attached at the end of this appendix. We summarize our findings as follows.

Proposition 10 (Equilibria with NG User Continuous Arrival) *If NG users continuously arrive in the market, the SaaS quality improvement rate α affects the equilibrium outcome as follows.*

(a) *(Entry Deterrence Equilibrium) If $\alpha \leq (\rho - 2\theta + 1)q$, the perpetual software vendor deters the SaaS vendor’s entry into the market: The equilibrium user strategy is SP1 (Upgrade, New), where the OG users upgrade and all NG users adopt the new perpetual software. The perpetual software vendor’s equilibrium prices are $p_u^* = p_n^* = (\rho - \theta)q - \frac{\alpha}{2}$.*

(b) *(Market Segmentation Equilibrium) If $(\rho - 2\theta + 1)q < \alpha \leq \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$, the perpetual software vendor and the SaaS vendor segment the market: The equilibrium user strategy is SP2 (Upgrade, SaaS), where the OG users upgrade to the new perpetual software and all NG users adopt SaaS. The equilibrium prices are:*

If $(\rho - 2\theta + 1)q < \alpha \leq 2(\rho - 1)q$, then $p_u^* = (\rho - 1)q$, $p_n^* = (\rho - 1)q$ and $p_s^* = (\theta - 1)q + \frac{\alpha}{2}$.

If $2(\rho - 1)q < \alpha \leq \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$, then $p_u^* = (\rho - 1)q$, $p_n^* = \frac{\alpha}{2}$ and $p_s^* = \alpha - (\rho - \theta)q$.

(c) (Sequential Dominance Equilibrium) If $\alpha > \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$, the two vendors serve the market sequentially as follows: During $[0, t_{SD}^*]$, the perpetual software vendor serves all OG users and NG users who arrive during this interval. At t_{SD}^* , these users switch to the SaaS, and in addition, NG users who enter the market in the interval $[t_{SD}^*, 1]$ all choose SaaS during this period. The equilibrium prices are:

$$p_u^* = \frac{(\rho-1)q[-4\alpha+(7\rho-10\theta+3)q+2\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]}{6\alpha}, p_n^* = \frac{[-2\alpha+5(\rho-\theta)q+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]}{18\alpha}, \text{ and } p_s^* = \frac{-2[\alpha-(\rho-\theta)q]+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}}{3}.$$

Overall, we find that all major insights under the discrete model still hold. When SaaS quality improvement is relatively small, the entry deterrence equilibrium emerges; when the SaaS quality improvement is high enough, the sequential dominance equilibrium emerges; and when the SaaS quality improvement is in the intermediate range, the market segmentation equilibrium emerges.

Moreover, we see that both vendors' optimal prices are the same as in the baseline model under the entry deterrence and market segmentation equilibria. The user groups they serve are also the same. However, the sequential dominance equilibrium is different. In the baseline model, the perpetual software vendor might charge an upgrade price that is the same as the new price, while in the continuous arrival setting, it always gives a price discount to OG users to induce them to upgrade. In addition, we also find that the perpetual vendor's new price is higher, the SaaS vendor's price is lower, and the switching time is later than the prices and switching time in the baseline model. As a result, the SaaS vendor earns a lower profit.

In summary, when the SaaS quality improvement rate is relatively high, so that sequential dominance equilibrium emerges, the perpetual software vendor is better off under the continuous arrival model. This outcome occurs mainly because NG users arrive to the market sequentially. The late arrivals are aware of the perpetual software vendor's ability to release a newer version software in the future, so they tend to choose the perpetual software upon arrival to enjoy the lower upgrade price for the future newer version.

Proofs for the Continuous User Arrival Model

Case (1) Entry Deterrence Strategy

Consider the strategy that the perpetual software vendor offers a low enough price to attract all OG users to upgrade to the new software, that NG users who arrive in the market early prefer New, and that NG users who arrive in the market late also prefer New and then upgrade to Newer at $t = 1$. Under this strategy, the SaaS vendor is out of the market, even if it offers $p_s = 0$.

First, to ensure that the OG users prefer Upgrade rather than Old, we need $\rho q - p_u \geq q$; that is, $p_u \leq (\rho - 1)q$ (K1). To ensure that the OG users prefer Upgrade rather than SaaS even if the SaaS price is 0, we need $\rho q - p_u \geq \int_0^1 (\theta q + \alpha t) dt$; that is, $p_u \leq (\rho - \theta)q - \frac{\alpha}{2}$ (K2). To ensure that NG users who arrive at $t = 0$ prefer New rather than SaaS, we need $\rho q - p_n \geq \int_0^1 (\theta q + \alpha t) dt$; that is, $p_n \leq (\rho - \theta)q - \frac{\alpha}{2}$ (K3). In addition, we also need NG users who arrive at $t = 1$ to prefer Newer rather than SaaS, so $(2\rho - 1)q - p_n \geq \theta q + \frac{3}{2}\alpha$; that is, $p_n \leq (2\rho - \theta - 1)q - \frac{3\alpha}{2}$ (K4). We can verify that both (K2) and (K3) are binding.

For NG users who arrive at $t > 0$, they might prefer New+Newer rather than New. The indifference user's entry time is determined by $\rho q - p_n = \rho q(1 - t_c) - p_n + (2\rho - 1)qt_c - p_u$; that is, $t_c = \frac{p_u}{(\rho-1)q}$. The perpetual software vendor's profit over $[0,1]$ is $p_u + p_n$. The first term is the profit from OG users, and the second term is the profit from NG users. Note that the perpetual software vendor generates the Upgrade profit from New+Newer users at $t = 1$. This profit is not counted toward the profit calculation in this software life cycle. Because the profit function increases in p_u , and note that $(\rho - 1)q > (\rho - \theta)q - \frac{\alpha}{2}$, we have $p_u^* = p_n^* = (\rho - \theta)q - \frac{\alpha}{2}$, $t_c^* = \frac{(\rho-\theta)q-\frac{\alpha}{2}}{(\rho-1)q}$, and $\pi_{perp}^{ED} = 2(\rho - \theta)q - \alpha$. Note that the condition for entry deterrence equilibrium is $\alpha \leq 2(\rho - \theta)q$.

Case (2) Market Segmentation Strategy

Consider the strategy in which the perpetual software vendor allows an SaaS vendor to enter into the market. Because OG users are more sticky than NG users, the perpetual software vendor, in giving up the NG users, charges $p_u^* = (\rho - 1)q$ to fully extract the surplus from OG

users. So the perpetual software vendor serves the OG users on the interval [0,1], and the SaaS vendor serves all NG users on the interval [0,1]. Comparing this strategy with the entry deterrence strategy, the SaaS vendor charges a positive p_s .

To ensure that the OG users choose Upgrade rather than SaaS, we need $\rho q - p_u \geq \int_0^1 (\theta q + at - p_s)dt$. So $p_s \geq p_u - (\rho - \theta)q + \frac{\alpha}{2}$ (K5). Substituting p_u^{MS} into (K5), we have $p_s^* \geq (\theta - 1)q + \frac{\alpha}{2}$. To prevent the OG users from switching to SaaS during their lifetime use, we need $\theta q + at - p_s \leq \rho q$; that is, $p_s \geq \alpha - (\rho - \theta)q$ (K6). To ensure that the NG users who arrive at $t = 0$ prefer SaaS rather than New, we need $\int_0^1 (\theta q + at - p_s)dt \geq \rho q - p_n$; that is, $p_n \geq (\rho - \theta)q + p_s - \frac{\alpha}{2}$ (K7). The perpetual software vendor can price the new software at a relatively high price, such that the SaaS vendor attracts the NG users starting from time 0. Because the SaaS vendor's profit is $p_s \int_0^1 t dt$, which linearly increases in p_s , we know that (K7) is binding.

To determine p_n , we need SaaS+Newer to be preferred to SaaS; that is, $\int_0^1 (\theta q + at - p_s)dt + (2\rho - 1)qt - p_n \int_t^{1+t} (\theta q + at - p_s)dt$. So $p_n \leq [(2\rho - 1 - \theta)q - \alpha + p_s]t - \frac{\alpha}{2}t^2$. Since (K7) is binding, substituting into p_s and solving for t we have $t_c = \frac{[(\rho-1)q-\frac{\alpha}{2}+p_n]-\sqrt{[(\rho-1)q-\frac{\alpha}{2}+p_n]^2-2\alpha p_n}}{\alpha}$. The perpetual software vendor earns profit on the interval $[t_c, 1]$. It charges p_n as low as possible. So we have two cases: If $\alpha > 2(\rho - 1)q$, then $p_s^* = \alpha - (\rho - \theta)q$ and $p_n^* = \frac{\alpha}{2}$. The SaaS vendor's profit is $\pi_{SaaS}^{MS} = p_s^* \int_0^1 t dt = \frac{(\theta-1)q}{2} + \frac{\alpha}{4}$. If $\alpha \leq 2(\rho - 1)q$, then $p_s^* = (\theta - 1)q + \frac{\alpha}{2}$ and $p_n^* = (\rho - 1)q$. The SaaS vendor's profit is $\pi_{SaaS}^{MS} = p_s^* \int_0^1 t dt = \frac{\alpha-(\rho-\theta)q}{2}$. Under both cases, $p_u^* = (\rho - 1)q$ and $\pi_{perp}^{MS} = (\rho - 1)q$.

Case (3) Sequential Dominance Strategy

We focus on the two firms' competitive equilibrium. Assume that OG users choose Upgrade+SaaS and NG users choose New+SaaS. Again, the switching time is determined by $\theta q + at_{s3} - p_s = \rho q$; that is, $t_{s3} = \frac{p_s+(\rho-\theta)q}{\alpha}$. At $t = 0$, the OG users prefer Upgrade+SaaS rather than Upgrade if $\rho qt_{s3} - p_u + \int_{t_{s3}}^1 (\theta q + at - p_s)dt \geq \rho q - p_u$, which holds when $p_s \leq \alpha - (\rho - \theta)q$. Similarly, any NG user who arrives at time $t < t_{s3}$ prefers New+SaaS rather than New if $\rho q(t_{s3} - t) - p_n + \int_{t_{s3}}^{1+t} (\theta q + at - p_s)dt \geq \rho q - p_n$; at $t = 0$, this condition gives $p_s \leq \alpha - (\rho - \theta)q$.

The SaaS vendor's profit is expressed as $p_s(1 - t_{s3}) + p_s t_{s3}(1 - t_{s3}) + p_s \int_{t_{s3}}^1 (1 - t)dt = \frac{p_s(1-t_{s3})(3+t_{s3})}{2}$. Note that the computation of profit is different for the two groups of users. The first term is the profit from OG users who switch to SaaS at t_{s3} ; the second term is the profit from the early arrival NG users (i.e., arrivals before t_{s3}) who switch to SaaS at t_{s3} ; the third term is the integral of all NG users who arrive after t_{s3} so they choose SaaS directly. Solving this optimization problem we have $p_s^* = \frac{-2[\alpha-(\rho-\theta)q]+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}}{3}$. We can verify that p_s^* is an interior solution if $\alpha > (2 + \sqrt{2})(\rho - \theta)q$. Substituting p_s^* into the expression of t_{s3} , we get the switching time in the sequential dominance equilibrium $t_{SD}^* = \frac{-2\alpha+5(\rho-\theta)q+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}}{3\alpha} < 1$. We can verify that $t_{SD}^* > 0$ under the condition $\alpha \geq (\rho - \theta)q$. At the boundary solution $p_s = \alpha - (\rho - \theta)q$, $t_{SD}^* = 1$, so (Upgrade+SaaS, New+SaaS) does not sustain as an equilibrium SP.

To ensure that OG users prefer Upgrade+SaaS rather than Old+SaaS, we need $\rho qt_{s2} - p_u + \int_{t_{s2}}^1 (\theta q + at - p_s)dt \geq qt_{s1} + \int_{t_{s1}}^1 (\theta q + at - p_s)dt$, where $t_{s2} = t_{s3}$ and $t_{s1} = \frac{p_s-(\theta-1)q}{\alpha}$ is the switching time for OG users when they choose Old+SaaS; that is, $p_u \leq \frac{(\rho-1)q[2p_s+(\rho-2\theta+1)q]}{2\alpha}$ (K8). Because the OG users are more sticky than the NG users, if the OG users prefer Upgrade+SaaS, then the NG users who arrive at $t = 0$ also prefer New+SaaS. Any NG user arriving before t_{s3} prefers New+SaaS rather than SaaS if $\rho q(t_{s3} - t) - p_n + \int_{t_{s3}}^{1+t} (\theta q + at - p_s)dt \geq \int_t^{1+t} (\theta q + at - p_s)dt$. Simplifying the conditions, we have $p_n \leq \frac{[(\rho-\theta)q+p_s]^2}{2\alpha}$ (K9). When $t > t_{s3}$, NG users' two strategies, SaaS+Newer and SaaS, are equivalent in the analysis because in the current planning period [0,1], the perpetual software vendor's profit for the newer version is not counted and the SaaS vendor's profit is the same.

Substituting p_s^* into (K9) we have $p_n^* = \frac{[-2\alpha+5(\rho-\theta)q+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]}{18\alpha}$. By (K8), $p_u^* = \frac{(\rho-1)q[-4\alpha+(7\rho-10\theta+3)q+2\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]}{6\alpha}$. Note that the perpetual software vendor prices satisfy $p_u^* < p_n^*$. The perpetual software vendor's profit is $\pi_{perp}^{SD} = p_u^* + p_n^* t_{SD}^*$, and the SaaS vendor's profit is $\pi_{SaaS}^{SD} = \frac{p_s^*(1-t_{SD}^*)(3+t_{SD}^*)}{2}$.

In summary, the three equilibria occur in different ranges defined by α . Comparing the vendors' equilibrium profits under different α regions, we can derive the final equilibrium outcome presented in Table K1. For example, in the most complicated case, when $\alpha > (2 + \sqrt{2})(\rho -$

$\theta)q$, both market segmentation and sequential dominance are possible equilibria. Note that π_{perp}^{SD} increases in α , but π_{perp}^{MS} is independent of α . A threshold $\hat{\alpha}$ must exist such that $\pi_{perp}^{SD} > \pi_{perp}^{MS}$. Therefore, if $\alpha \geq \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$, sequential dominance emerges as the final market equilibrium outcome.

Table K1. Equilibrium Prices and Profits Under User Continuous Arrival Model				
(a) Equilibrium Prices with User Continuous Arrival				
Region	p_u^*	p_n^*	p_s^*	
i	$(\rho - \theta)q - \frac{\alpha}{2}$	$(\rho - \theta)q - \frac{\alpha}{2}$	0	
ii	$(\rho - 1)q$	$(\rho - 1)q$	$(\theta - 1)q + \frac{\alpha}{2}$	
iii	$(\rho - 1)q$	$\frac{\alpha}{2}$	$\alpha - (\rho - \theta)q$	
iv	$\frac{(\rho-1)q[-4\alpha+(7\rho-10\theta+3)q+2\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]}{6\alpha}$	$\frac{[-2\alpha+5(\rho-\theta)q+\sqrt{[\alpha+(\rho-\theta)q]^2+12\alpha^2}]^2}{18\alpha}$	$\frac{-2[\alpha - (\rho - \theta)q] + \sqrt{[\alpha + (\rho - \theta)q]^2 + 12\alpha^2}}{3}$	
(b) Equilibrium Prices with User Continuous Arrival Model				
Region	Condition	Equilibrium	π_{perp}^*	π_{saas}^*
i	$\alpha \leq (\rho - 2\theta + 1)q$	Entry Deterrence	$2(\rho - \theta)q - \alpha$	0
ii	$(\rho - 2\theta + 1)q < \alpha \leq 2(\rho - 1)q$	Market Segmentation	$(\rho - 1)q$	$\frac{(\theta-1)q}{2} + \frac{\alpha}{4}$
iii	$2(\rho - 1)q < \alpha \leq \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$	Market Segmentation	$\frac{[\alpha+2(\rho-\theta)q]^2[5\alpha+4(\rho-\theta)q]}{54\alpha^2}$	$\frac{2[\alpha - (\rho - \theta)q]^2[4\alpha - (\rho - \theta)q]}{27\alpha^2}$
iv	$\alpha > \max[(2 + \sqrt{2})(\rho - \theta)q, \hat{\alpha}]$	Sequential Dominance	$p_u^* + p_n^* t_{SD}^*$	$\frac{p_s^*(1-t_{SD}^*)(3-t_{SD}^*)}{2}$