

SOFTWARE UPGRADES UNDER MONOPOLY

Jiří Střelický
Krešimir Žigić

CERGE-EI

Charles University
Center for Economic Research and Graduate Education
Academy of Sciences of the Czech Republic
Economics Institute

Working Paper Series
(ISSN 1211-3298)

478

Software Upgrades under Monopoly

Jiří Střelický
Krešimir Žigić

CERGE-EI
Prague, January 2013

**ISBN 978-80-7343-282-9 (Univerzita Karlova. Centrum pro ekonomický výzkum
a doktorské studium)**

ISBN 978-80-7344-274-3 (Národohospodářský ústav AV ČR, v.v.i.)

Software Upgrades under Monopoly*

Jiří Střelický[†] and Krešimir Žigić[‡]

January 2013

Abstract

We study price discrimination in a monopolistic software market. The monopolist charges different prices for the upgrade version and for the full version. Consumers are heterogeneous in taste for infinitely durable software and there is no resale. We show that price discrimination leads to a higher software quality but raises both absolute price and price per quality. This price discrimination does not increase sales and it decreases the total number of consumers compared to no discrimination. Finally, such discrimination decreases consumers' surplus but increases the developer's profit and social welfare that attains the social optimum in the limit.

Abstrakt

V článku analyzujeme cenovou diskriminaci na monopolním softwarovém trhu. Monopolista nabízí rozdílné ceny za upgrade a za plnou verzi. Uživatelé jsou heterogenní ve vnímání softwaru, který má nekonečnou trvanlivost a který není určený na další prodej. Ukazujeme, že cenová diskriminace vede k vyšší kvalitě softwaru, ale zároveň způsobuje růst ceny, i růst ceny za jednotku kvality. Tato cenová diskriminace nezvyšuje prodej a snižuje celkový počet uživatelů v porovnání s variantou bez diskriminace. V neposlední řadě, tato diskriminace snižuje uživatelský přebytek, zvyšuje profit firmy a celkový společenský blahobyť se limitně blíží optimu.

Keywords: monopoly, durable goods, software, upgrades, price discrimination

JEL Classification: C61, L12; L15;

*Authors would like to thank to Martin Peitz, Paul Belleflamme, Milan Horniaček and Avner Shaked for their valuable comments and Michael Kúnin for strong technical support during shaping the last version of the article.

[†]jiri.strelicky@seznam.cz

[‡]CERGE-EI, a joint workplace of the Center for Economic Research and Graduate Education at Charles University in Prague and the Economics Institute of the Czech Academy of Sciences; kresimir.zigic@cerge-ei.cz

1 Introduction

The software market is one of the largest and fastest growing markets where, for instance, the largest developer (Microsoft) has a revenue of more than \$60 billion per year¹. There are at least several hundred thousand other developers operating on the market, and so it would appear that tough competition is the norm. However, by taking a closer look at every particular sub-market of the software market (not only the operating systems), we could often identify a dominant developer with such an established position (e.g., Adobe, Symantec, Pinnacle, and of course Microsoft) that the whole sub-market could be treated almost as a monopoly. Such a monopolistic market structure may serve as an explanation for some high software prices (Katz and Shapiro, 1998). Moreover, there is a high range of possible price discrimination schemes that such a monopolist may undertake. Since software is “lent” to users, where the identity of both parties is often known, the developer could easily set marginal prices to different groups of users without incentive problems. In such a case, a user prefers to reveal the personal information that qualifies him for a lower price. In the real world, we could observe dozens of prices for identical software: a price for the standard retail user (often called the full version), the OEM version, the upgrade, and student or multi-license versions. On top of that, there are different prices for the university, the army, the public sector, or large corporations, and naturally, the goal is to set prices close to the reservation prices of the respective groups.

Besides the pricing policy, an even more important issue is whether, and how, to improve the *quality* of software that such a monopolist firm generates. By the vague term “quality,” we understand not only software functionality, but foremost software stability, speed, compatibility with hardware, and, as is becoming increasingly important nowadays, security. From this point of view, software evolution may be viewed as socially sub-optimal if a developer rashly introduces poorly tuned software² that leads to welfare losses caused by additional consumer costs³.

Thus, one of the aims of our paper is to analyze the pricing policy of a software monopolist and its impact on software quality evolution. More specifically, a monopolist can either set a single price for its products, or it can stick to price

¹The Microsoft report for the fiscal year 2008.

²For example, more than 20 000 mistakes were known for Microsoft Windows 2000 at the time of the release, and a Service Pack for Microsoft Windows XP was in preparation even before the official XP release.

³Though most users may have in mind frozen programs, software errors can, however, cause more severe losses. United States’ losses due to software problems are estimated at USD 59.5 billion, which is 0.6% of GDP. According to Stanford Research Institute, most of these problems are due to lack of testing.

discrimination based on an upgrade scheme. As we shall see, a different approach to a pricing strategy leads, in turn, to a different evolution of software quality. Software developers often motivate users to switch more frequently to a new version of the product by lowering the price in the case where users own the previous version. Thus, a user faces two options: to switch to every new version to get the upgrade price, or to switch only from time to time for the full price⁴. In the real world, upgrade prices are predominantly used for business software. The ratio between upgrade and full price varies according to the developer's position on the market, the necessity to have updated software, and innovation frequency⁵.

Because the average time between releases of new versions is short⁶, users have to familiarize themselves with new software on a very regular basis. When an old version becomes sufficiently outdated, the users' willingness to switch to new software grows, and software obsolescence becomes crucial to both users' and developers' decision processes. This willingness, however, varies among the users so different users tend to switch to new software with different frequencies and the distribution of software among users is in each period different.

In order to model the above phenomena, we put forward a dynamic model environment in which there is, on the one side, a software monopolist who introduces a new version (quality) of software all the time (or every period), and on the other side, there are consumers (or users) who are heterogeneous in their sensitivity to software quality. That is, they buy software with a different frequency depending on their currently possessed software version. We assume that there is no second-hand market for software and that the developer sets its prices at the outset for the whole horizon. That is, he sets either a single price (in the absence of price discrimination) or a pair of prices, whereby a lower price is charged to the consumers who buy new software every period. Since software is perfectly durable and developers keep introducing new versions all the time, the model has to deal with a growing number of software versions. In other words, at the given point in time, consumers as a whole possess both the newest and the older versions of software while the total number of versions increases over time. Since each new product carries a quality improvement over the preceding one, it is natural to assume vertical product differentiation in such a market. In such a setup we also study the "consumer side" of the market, explaining consumers' dynamics in switching or staying at a given software

⁴Nevertheless, sometimes users need not be motivated to switch, e.g., game players are often willing to pay more for a new version when they own the previous version than when they have no experience with that particular game.

⁵E.g., the full price for MS Windows 7 Home Premium is \$199 and the upgrade price is only \$119. Very often the upgrade price is approximately 50%-70% of the full price.

⁶The major releases are usually between 12 and 24 months.

and explicitly addressing the time distribution of consumers (or demand) over each generation of product (software) in an infinite time horizon.

We first assume that users have imperfect foresight in the sense that they are unable to predict how often they would switch to a new software version in the future. Later on, we extend our model to users with perfect foresight without changing the developer's problem. We shall show that the perfect foresight of users, if known to developers, increases the developers' opportunity to force users to switch more often. The purpose of starting first with the imperfect foresight is that it shows more intuitively the working of the model and makes the more general model of perfect foresight easier to follow and understand.

The key model simplification is the absence of consumers' outside option. In reality, every user has outside options such as using similar software from non-dominant developers (including open-source software) or even using an illegal copy of the software. However, even though software prices have a significant impact on the users' decision whether to undertake piracy or not, what is more important for the extent of software piracy is the role of government⁷ and the expected piracy punishment. Thus, our model well suits markets such as business software in developed countries, where software upgrading is standard, and a high expected punishment limits piracy. Banerjee (2003) analyzed the problem of switching from legal software to piracy, and its social welfare impact with respect to the government's incentives to tolerate piracy (see Belleflamme and Peitz, 2012 for a comprehensive review of software piracy literature). We also assume away the network effects since their significance under monopoly is often suppressed to the role of an entry barrier. The concept of network effects under a monopoly was studied by Ellison and Fudenberg (2000) and Fudenberg and Tirole (2000).

As for the related literature, our approach shares some of the features of behavior-based pricing with multiple products (see, for instance, Fudenberg and Tirole, 1998, and Ellison and Fudenberg, 2000; see also Villas-Boas and Fudenberg, 2007 and Belleflamme and Peitz, 2010 for a survey of this literature). The key feature of this approach is that the monopolist may be able to use his information about the consumers' purchasing history to offer different prices and/or products to consumers with such different histories. The two most common information structures consistent with behavior-based pricing are the "identified consumers" and the "semi-anonymous consumers." While the first category is self-explanatory, semi-anonymous consumers are consumers who can prove that they purchased software in the last

⁷Usually governments only provide a legal environment, whereas actual anti-piracy force is exerted by independent or public organizations.

period if they wish to do so, but they can also pretend not to have bought it if this is in their interest. So the underlying assumption in our model in which consumers qualify for an upgrade if they buy the last version of the software can be thought to fit the “semi-anonymous consumers” assumption. Moreover, much like Fudenberg and Tirole (1998), and Ellison and Fudenberg (2000), we also study the provision of “upgrades” by a monopolist in a setting of vertical differentiation, where there is a consensus among customers that newer versions of the software are better than older ones. Our analysis, however, differs from the above literature in some important aspects. First, we focus on the software market, rather than on a broader class of durable goods. This, among others, implies that the existence of a second-hand market is not appropriate in our setting. We, unlike Fudenberg and Tirole (1998), and Ellison and Fudenberg (2000), use an infinite horizon model to study successive product generations in the software market, and in this sense, we focus on the incentives to innovate and on the social optimal level of innovation rather than on the very pricing decisions. As for monopoly pricing, we do not deal explicitly with the commitment issue, and we are mostly interested in how the introduction of price discrimination based on upgrades and the lock-in of consumers affect the software quality evolution. Moreover, we analyze the two set-ups of imperfect and perfect foresight consumers. Finally, our approach differs also from related infinite horizon models (like Fishman and Rob, 2000 or Anton and Biglaiser, 2010, see below) since we do not use the concept of steady state equilibrium. In our model the numbers of product versions tends to infinity as time goes to infinity and, respectively, the distribution of software versions differs from period to period and never repeats in equilibrium.

The analysis of Fishman and Rob (2000), however, does share some common aspects with our analysis. They consider a durable-goods monopolist that periodically introduces new product models with each new model being an improvement over the preceding one. In this light, our focus on software evolution can be viewed as a particular case of their analysis of R&D into durable-good industries under monopoly. They, however, assume that consumers are homogeneous, and, with their focus on social optimum, their primary finding is that if the monopoly developer can neither shorten the lifetime of its products nor discriminate in prices, then the monopoly developer innovates less frequently and invests less than at the socially optimal level. They also show that if either planned obsolescence or price discrimination based on the age of the product held by the consumer is available, then the developer can both increase his or her profit and implement the social optimum.⁸ Much like

⁸There are many papers that build upon or deal with other aspects of Fishman and Rob (2000).

Fishman and Rob (2000), Anton and Biglaiser (2010) also study an infinite horizon model with steady quality improvement in a durable goods monopoly setup keeping the assumption of the identical (homogenous) consumers. They, however, focus on pricing and adoption, taking innovation as exogenous and, among other things, study the developer's optimal decision on which bundles and prices to offer. Coming back to Fishman and Rob's (2000) paper, a peculiar feature of their analysis is that the homogeneity of consumers leads to every new product being purchased by all consumers. However, real durable-goods markets, and the software market in particular, are characterized by the simultaneous presence of both the newest and the previous products in the consumers' ownership. That is to say, while there are consumers who use the newest product, there also are consumers who keep using previous ones. As each new product represents a quality improvement over the preceding one, vertical differentiation is inherent in such markets; so, as already stated above, we allow for consumer heterogeneity with respect to taste for quality that captures this property. In addition, while Fishman and Rob (2000) show that a price discriminating monopolist is able to fully implement the social optimum, they assume that the price discrimination is of the first-degree type. This is in the sense that the developer is able to freely change the price according to the age of the product currently owned by the consumer. In our approach, the price discount is only offered to those who own the immediately preceding version, which is a more realistic setting of behavior based on third-degree price discrimination. We show that if the developer can set different prices for the upgrade and the full version, then the social welfare resulting from the monopolist's optimal action in the limit approaches the social optimal level irrespective of whether the users have perfect or imperfect foresight. The limit in question is the fraction of users who are not "locked-in", that is, the users who do not switch every period, and this fraction tends to zero. Alternatively, the switching frequency of these users (whose measure tends to zero) goes to infinity. We also show that the popular notion according to which upgrade prices help to spread software and lead to lower price levels does not hold if different prices for the upgrade and the full version are used.

The paper is organized as follows. In the second chapter, we set up the model that serves as the general framework for our analysis. The first part of the article (the

For instance, Atil et al. (2008) show that a competitive environment or after-market leads to a lower investment in R&D. Mehra and Seidmann (2008) focus on the life-cycle management of software products and show how the optimal upgrade changes throughout the life cycle based on whether there are heterogeneous customers, externalities, or product incompatibilities. Nahm (2004) focuses on the interaction between inter-temporal pricing and R&D decisions based on the pricing regimes (net sales vs. buyback). Other papers investigating obsolescence include Echevarría (2005), Wang and Hui (2005), and Inderst (2008).

third, fourth, and fifth sections) deals with the choice of a software upgrade versus the full-price version in an environment where users have imperfect foresight. More specifically, in the third section, we analyze the developer's behavior without price discrimination, and in the fourth section, we focus on price discrimination based on a lower price of upgrade. Finally, in the fifth section, we analyze the impact of price discrimination on prices, software quality evolution, and social welfare. The next three sections constitute the second part of the article, where users are now assumed to have perfect foresight. In sections six to eight, we replicate our analysis from sections three to five, but now with the assumption of perfect foresight. In addition, in section seven, we study the monopolist's tendency to initiate lock-in behavior. In the ninth section, we discuss the results obtained across all sections and point out the differences between an imperfect versus perfect foresight set-up. Section ten concludes.

2 The model

2.1 Basics

There is a monopoly software developer who releases a new version every period. The timeline starts in period 1, when the first version of the software is released, and continues indefinitely.

When a user buys software, he can use the version purchased forever without any additional fee. All versions of software are infinitely durable without depreciation. However, the developer sells only the latest version of the software every period, whereas older versions are not sold anymore. Besides that, the developer outlaws any resale by a license agreement with the user. We assume that users have no outside option like piracy or open-source software, so all users are fully dependent on the developer's offer.

Price discrimination

While only the current software version can be sold in every period, the developer is allowed to price discriminate by setting different prices for the same version along with the eligibility rules for a consumer to qualify for a given price. We assume that in every period it is impossible that there is a user who is completely ineligible to any price offered in the period. In general, the developer can discriminate based on the history of the user's previous actions, and the rules may change between periods. In this paper, we make the following two assumptions.

First, the only thing the developer can observe about a specific user in any period is the software version (if any) possessed by the user. Thus, the only way the developer can price discriminate is based on the age (in periods) of the software version in the user’s hands.

Second, we assume that the developer’s pricing policy is stable over time. Namely, the prices offered in period t and the corresponding eligibility rules do not depend on t . We concentrate on the following two cases:

- The single price, when every version is offered at the same price p to all users in every period.
- The “new user” versus an “upgrade”, when in every period t anyone can buy at the “new user” price p_2 , but the users who purchased the software version immediately preceding the current one, i.e., the version of period $t - 1$, and only those users, are entitled⁹ to buy at the “upgrade” price p_1 , with p_1 and p_2 constant across periods. Here those who own older versions, but not the immediately preceding one, are treated as “new users” just as those who own no version at all.

Remark 1. *One of the key assumptions we make is that the monopolist is able to precommit to its price(s). That is, the monopolist sets price(s) in the first period and keeps them unchanged thereafter. In other words, the famous Coase conjecture that a price setting monopolist in an inter-temporal set-up may not commit to its future prices is not an issue in our set-up. The reason for this may, for instance, be that, besides permanent upgrades, our monopolist has a reputation for sticking to his or her pre-announced price or has fixed production capacity that serves as a commitment device against future price decreases, and so forth.¹⁰ (See more on the durable-good monopoly with commitment in Belleflamme and Peitz, 2010, and also see Fudenberg and Villas-Boa on behavioral based price discrimination and Stole’s survey on price discrimination, 2007.)*

⁹Generally, it is possible to introduce more differentiated upgrade prices, e.g., for users who possess the version from the second latest period, the third latest period, and so on. Doing that would not substantially change our results in any way, but the structure of the model, as well as overall results, would become less transparent.

¹⁰There is, however, an alternative micro-foundation approach which explains the price rigidity by means of rational inattention (see, for instance, Sims, 2005). In the realistic case when this inattention appears on the consumers’ side, a frequent price changes may require consumers to pay a lot of attention to the price. That, in turn, may be irritating for the consumers so in the end, they could decide to consume less. As a result, it would be optimal for the seller to choose his pricing strategy in advance and commits to it (see Matějka, 2010). Moreover, this may also explain the empirical observations that real-life developers are observed to keep roughly constant prices over longer period of time.

Quality and R&D cost

We denote software quality in period t as Q_t . For simplicity, we treat quality as a one-dimensional variable that can be viewed as a weighted linear combination of all characteristics (performance, design, stability, and security, amid others) In fact, exact quality cannot be measured, so we measure the quality indirectly by the willingness to pay for a product.

As resale is outlawed, there is no outside option for the users, and any version is infinitely durable with no depreciation, and the only trigger for new demand is an improvement in software quality that persuades users to replace their older version of software¹¹. In our model, we introduce the developer's cost of software quality improvement ΔQ , which is in fact the cost of R&D. All other developer costs are normalized to zero. We assume that the cost of software quality improvement is increasing in quality $\frac{\partial C(\Delta Q)}{\partial \Delta Q} > 0$, and we assume that costs are convex $\frac{\partial^2 C(\Delta Q)}{\partial \Delta Q^2} > 0$. This condition means that gradual quality improvement (e.g., to improve quality by 1 every period for the next three periods) is cheaper than a significant quality jump (e.g., to improve quality by 3 within one period and then no improvement in next 2 periods). Both conditions are satisfied for quadratic functions; so, we use

$$C(\Delta Q) = \bar{B} \cdot (\Delta Q)^2, \quad (1)$$

where parameter $\bar{B} > 0$ reflects R&D effectiveness.

Improving software is a longtime process based on cumulative activities like learning by doing. Thus, the developer cannot increase software quality simply by hiring a large number of new programmers in one period. To achieve significant progress in quality improvement, the developer must decide about the targeted future quality several periods in advance. The developer does so by deciding on the maximum evolution quality improvement in period t , K_t , at least n periods before period t . Then in period t , software quality improvement is limited to $\Delta Q_t = Q_t - Q_{t-1} \leq K_t$ and cannot be exceeded¹². In reality, n may be several years. In our model, we assume $n \rightarrow \infty$; hence, the developer decides about the evolution capacity once and for all.

In line with the assumption of stable prices and eligibility rules, we assume that the evolution capacity is stable over time, $K_t = K$ for all t . The capacity

¹¹The model can be generalized by introducing the probability of failure. Nevertheless, we then get total demand as a linear combination of demand stemming from technological improvement and demand coming from physical depreciation. The total result will be influenced by the weights put on each source without changing the core of the result.

¹²This is a simplification helping us to focus the paper on price discrimination.

is assumed to be fully utilized in every period. Thus, a monopolist releases new software versions with constant quality improvement at the maximum capacity level, that is, $Q = \Delta Q_t = Q_t - Q_{t-1} = K$ each period t . If the initial quality in period t is Q_t , then the quality in period $t + l$ is $Q_{t+l} = Q_t + l \cdot Q$.

As all developer choice variables are stable (namely the prices and the quality jump), we assume that they are chosen in the very beginning. Note that in our framework, quality adjustment choice and capacity choice are basically equivalent since in selecting capacity K in the beginning, the developer intends to fully use this capacity to achieve quality improvement $Q = K$ in every period.

2.2 Users

We assume an infinite number of heterogenous users on the market. The users differ in their sensitivity θ to product quality, where θ is uniformly distributed and is normalized¹³ to interval $(0, 1)$. Users with θ close to 1 primarily prefer quality, while users with θ close to 0 are very price sensitive. Users' discount factor between periods is $\beta \in [0, 1)$, which is assumed the same¹⁴ for all users.

At the beginning of every period, a user may either own no software version at all or own a software version from some of the previous periods. If no version of software is owned, then a consumer faces a question: "Buy the software now or wait for another period when the consumer value of the new software, because of a higher product quality, will increase." If a previous software version is owned, then the question is: "Buy the new version now or wait until the consumer value is increased and meanwhile continue using the old version." We can consider owning no software as owning the "zero version," whose quality is $Q_0 = 0$.

Consider a user of type θ who owns a version of quality Q_t in period t . If this user decides not to buy the newest version, then this user's utility flow in period t is θQ_t . This implies that users have zero utility flow before they buy software for the first time. If this user buys the newest version, whose quality is Q_t , at price p_t , then this user's utility flow in period t is $\theta Q_t - p_t$. Note that p_t can differ among users due to the developer's eligibility rules. However, if a user happens to be eligible for more than one price in a given period, then the lowest of these prices will be used as p_t .

In the paper, we separately analyze two cases of user behavior:

¹³We could also explicitly introduce a "market size" parameter by assuming that the density of θ is a positive constant rather than exactly 1, but this is effectively captured by parameter \bar{B} in the developer's cost function as demonstrated below.

¹⁴In other words, we assume that all heterogeneity among users is captured by θ .

- Users with *imperfect foresight* or myopic users, who make a decision whether or not to buy software based only on the comparison of the utility of having (or not having) the current product in the current period, while ignoring their eventual switching to new software. In calculating their utility, these users assume that they would keep the current version forever.
- Users with *perfect foresight*, who know exactly in which period they would switch to a new version; these users calculate their utility flow precisely in advance.

The decision of users with imperfect foresight

An imperfect foresight user of type θ who owns no software until period t , switches to software of quality Q_t at price p_t in period t , and who never switches again, has the following infinite utility flow from period t onwards:

$$U_t = \theta(Q_t + \beta Q_t + \beta^2 Q_t + \dots) - p_t = \theta \frac{1}{1 - \beta} Q_t - p_t. \quad (2)$$

For simplicity, denote $q_t = \frac{1}{1 - \beta} Q_t$ and the utility flow of a user θ from buying software of quality q_t for price p_t in period t as

$$U^\theta(q_t, p_t) = \theta q_t - p_t. \quad (3)$$

It is obvious that a new user θ strictly buys software at period t if and only if $U^\theta(q_t, p_t) > 0$. The marginal user who is indifferent¹⁵ between buying and waiting for some next period has sensitivity parameter $\theta = \frac{p_t}{q_t}$.

The previous decision process, however, does not cover a user who already possesses a version of software. Consider now an imperfect foresight user of type θ who possesses the software of quality Q_l from period l . Such a user has already ensured utility flow $U^\theta = \theta Q_l$ at every period $t \geq l$. He decides to switch to new software that would bring him $U^\theta = \theta Q_t$ if and only if the difference in quality offsets the disutility from price p_t . It implies that he buys in period t if and only if

$$U^\theta(q_t, p_t) = \theta q_t - p_t \geq \theta q_l = U^\theta(q_l). \quad (4)$$

If the quality change is not sufficient to compensate for the disutility from the price (if $\theta(q_t - q_l) < p_t$), then the user does not buy now, uses his older version from period

¹⁵Indifferent consumers can either buy or not, and it is useless to restrict them, for example, to buy because their measure is 0 even if their number goes to infinity as t tends to infinity.

l , and waits until the next period $t+1$ when he enters into the decision process again and when he compares the utility flow θq_l with $\theta q_{t+1} - p_{t+1}$.

As can be seen from the user decision process described above, the users do not foresee future software quality levels Q_t and prices p_t so that each period they look at the actual quality offered by the developer and decide based on the current price. In this case, the users cannot foresee the exact time they would switch to a new product. These users simplify their decision process by simply comparing the indefinite utility flows from the old and current software versions ignoring their own future decision on software choice.

The decision of users with perfect foresight

Perfect foresight means that every user can foresee future quality levels Q_t and prices p_t in every period t . Then a perfect foresight user faces the following problem in every period. Let U_t^l be the user's discounted utility flow in period t given that the version from period $l \leq t$ with quality Q_l is used during this period, and let U_t be a shortcut for U_t^t . Let p_t^l be the minimal price the user is eligible for in period t given that the version from period l is owned. If the currently owned version $l < t$ is used for n periods on and then a new version is bought in period $t+n$, then

$$U_t^l = \theta Q_l (1 + \beta + \dots + \beta^{n-1}) + \beta^n U_{t+n}, \quad (5)$$

whereas if the new version t is purchased at price p_t^l and then used for n periods, when another new version is bought, then

$$U_t = -p_t^l + \theta Q_t (1 + \beta + \dots + \beta^{n-1}) + \beta^n U_{t+n}. \quad (6)$$

In every period, the user chooses (i) between buying and not buying and (ii) for how long to keep the version, given the anticipated quality and price development, which can be generally written as a dynamic programming problem. Given our assumptions about the stability of prices, pricing rules, and quality improvement, U_t^l solely depends on the difference $t-l$ in the sense $U_{t+m}^{l+m} = U_t^l$ for any integer $m \geq 0$. An interpretation of this is that the consumer has a guaranteed utility level of θQ_l per period, which amounts to the discounted flow of $\theta \frac{Q_l}{1-\beta} = \theta q_l$, and decides on the basis of added utility so that equations (5) and (6) can be re-written as

$$U_t^l - \theta q_l = \beta^n (U_{t+n} - \theta q_l) \quad (7)$$

if the consumer does not switch in period t , and

$$U_t - \theta q_t = -p_t^l + \theta (Q_t - Q_l) (1 + \beta + \dots + \beta^{n-1}) + \beta^n (U_{t+n} - \theta q_t) \quad (8)$$

$$= -p_t^l + \theta (q_t - q_l) + \beta^n (U_{t+n} - \theta q_t) \quad (9)$$

if the consumer switches in period t .

As a particularly important example, consider a user with a high sensitivity θ , who buys every period and he is aware of this. Given the stability of prices ($p_t = p$ or $p_t = p_1$) and quality improvements ($Q_t = Q_{t-1} + Q$, so that $q_t = q_{t-1} + q$), the last equation takes the form

$$U_t - \theta q_{t-1} = -p + \theta q + \beta (U_{t+1} - \theta q_t) \quad (10)$$

for every t . Infinite iteration of this equation yields (for $\beta < 1$)

$$U_t - \theta q_{t-1} = (-p + \theta q) (1 + \beta + \beta^2 + \dots) = \frac{1}{1 - \beta} (\theta q - p), \quad (11)$$

so that a necessary condition for a user with perfect foresight to switch every period is $\theta q - p \geq 0$.

Notation 1. *We denote the user who buys every period as a high-end user and a user who buys less frequently than every period as a low-end user.*

Regularity of upgrades

In general, if prices, quality improvement, and eligibility rules vary over time, so may vary the users' decision whether to buy the new version or keep the currently held one for another period. However, the ensuing stability assumptions lead to the following result, which substantially simplifies further analysis.

Proposition 1. *Let prices, quality improvement levels, and price eligibility rules be constant over time, and let the developer's pricing policy be either "upgrade" price p_1 versus "new user" price p_2 or single-price $p_1 = p_2 = p$. Then it is optimal for a user with either imperfect or perfect foresight to switch regularly, i.e., there exists a period $T \geq 0$ and a natural number $n = n(\beta, \theta, p_1, p_2, Q)$ such that the user switches in periods $T, T + n, T + 2n, \dots$, and in no other period. In addition, n is non-increasing in θ .*

Proof. The existence follows from the facts that the initial utility is zero, the per-period quality improvement is positive, and the prices are stable. Then $\exists T$ such that $Q_0 + T \cdot Q > \max\{p_1, p_2\}$.

Regularity under the single price stems from the fact that the user faces exactly

the same problem in every period. Regularity under “upgrade” versus “new user” prices follow from the analysis in section 4 (imperfect foresight) and 7 (perfect foresight). ■

In addition, the fact that n is non-increasing in θ follows from the “ $\theta Q - p$ ” utility structure: as θ increases, the user will not decide to purchase new versions less frequently, which means that n will not decrease.

While we show in this proposition that it is an optimal solution to upgrade regularly, it is actually the optimal behavior for all users except for those indifferent between two switching frequencies. However, such users are of measure zero and can be thus neglected.

Detailed descriptions of user decisions will follow in the dedicated sections. For now, only note that the distribution of products across perfect foresight users differs from the distribution of products across imperfect foresight users.

The participation constraint and foresight

From our analysis of the decision process of a user with imperfect foresight, it follows that a necessary condition for such a user to switch every n periods at price p is $\theta(q_{T+n} - q_T) = \theta nq \geq p$. Using the same approach as in the derivation of (11), we can show that the same necessary condition (which is thus the participation constraint in our model) applies for users with perfect foresight. In addition, a user with imperfect foresight will switch to a new product at the earliest possible moment, so that the condition $\theta nq - p \geq 0$ is also sufficient for the minimal n at which it is satisfied. However, a user with perfect foresight may decide to wait till the next period or even longer instead of buying the new version at the first opportunity when the participation constraint is met.

Consider again a high-end user with a high sensitivity θ , who buys every period and he knows that (perfect foresight), and let the developer use the single-price policy. While we have shown that the participation constraint for this user is $\theta q - p \geq 0$, the user has other possibilities, one of which is buying every two periods. Logically, the user will prefer buying every period to buying every two periods if the discounted utility flow over a span of two periods is higher in the former case. Assume the user decides between switching in periods T and $T + 1$ and in period T alone. The discounted (at the beginning of period T) utility flow is then $\theta Q_T - p + \beta(\theta Q_{T+1} - p)$ in the former case and $\theta Q_T - p + \beta(\theta Q_T)$ in the latter case. Therefore, switching every period is more profitable than switching every two periods if

$$\theta(Q_{T+1} - Q_T) = \theta Q \geq p, \tag{12}$$

which is equivalent to $\theta \geq p/Q$. Since $q > Q$ for $\beta > 0$, condition (12) is stronger than the participation constraint. In the part of this paper dealing with perfect foresight, we show that this condition is sufficient for $n = 1$ and derive the corresponding condition for $n > 1$.

Remark 2. *If we assumed that the user initially owns the version from period $T-1$ and chooses between upgrading in periods T and $T+1$, and upgrading in period $T+1$ alone, the result would be the same.*

Note that this decision rule has now capital Q rather than q and recall the difference between the two: Q is the actual quality of software while q is the user-discounted flow of quality, $q = \frac{1}{1-\beta}Q$.

Notation 2. *In the rest of the paper, we will refer to both variables q , Q as “quality” keeping in mind the difference between the two.*

Demand

As every user switches to new software versions regularly, let $d_n = d_n(\beta, p_1, p_2, Q)$ be the measure, in terms of θ , of users who switch exactly every n periods. In general, $d_n \geq 0$ (some switching frequencies can be absent) and $\sum_{n=1}^{\infty} d_n = 1$.

We assume that the developer cannot observe the exact distribution of software versions across users, but the developer knows the function $d_n(\cdot)$. This can be interpreted in the sense that the developer cannot observe in which period t he actually finds himself, so he cannot observe software distribution across users and assumes that all possible distributions are equiprobable (each distribution appears in the whole infinite model just once).

2.3 The developer’s problem

The model deals with heterogenous users over infinite number of periods, and as $t \rightarrow \infty$, the number of software versions the users can own goes to infinity. The distribution of software versions differs from period to period and never repeats. Thus, the developer faces a different demand function each period, which leads to different profit Π_i , so the total profit would be $\Pi_{\infty} = \sum_{i=1}^{\infty} \delta^i \Pi_i$, where δ is the developer’s discount factor.

To simplify our analysis, we assume that the developer values profit from every period equally¹⁶, so that the discounting factor is $\delta = 1$. Thus, the developer’s infinite-time profit maximization is equivalent to maximizing the average profit per

¹⁶A possible interpretation is that an increase in the user base due to network effects exactly offsets the discounting of future earnings.

period. In the rest of the paper, we will assume that the developer maximizes the following profit function:

$$\Pi = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \Pi_i. \quad (13)$$

Remark 3. *Fixing quality jump forever and maximizing average profit per period are key simplifications in the model that would allow us to reach an analytical solution while keeping the rest of the assumptions very flexible (infinite number of different products, infinite heterogenous customer, infinite number of users groups with different decisions). These simplifications are not so far from empirical observations given the fact that real-life developers are observed to keep roughly constant prices over several periods and to improve their products gradually.*

Remark 4. *As we already argued, the simplification with maximizing average profit corresponds to the situation where the developer cannot observe in which period he actually finds himself, thus he assumes that all possible version distributions across consumers are equiprobable. In other words, the developer views profits in different periods as independent and identically distributed random variables (due to price and quality adjustment stability), so that the limit in (13) equals the average per-period profit by the law of large numbers, and maximizes the average profit across all distributions. In fact, this is the same problem as stated in equation (13) except the constant.*

More specifically, the developer knows the demand function d_n , so that in every period the share d_1 of users will buy the new version (at price p_1), whereas for each $n > 1$, only $\frac{1}{n}$ of those users who switch only in every n periods will buy on the average (at price p_2 , note that $p_1 = p_2 = p$ for the single price policy). Then the average per-period revenue equals $p_1 d_1 + p_2 \sum_{n=2}^{\infty} \frac{1}{n} d_n$, whereas the developer's cost equals $\bar{B}Q^2$ per period. Thus, the developer's profit per period can be rewritten as

$$\Pi = p_1 d_1 + p_2 \sum_{n=2}^{\infty} \frac{1}{n} d_n - \bar{B}Q^2. \quad (14)$$

Also denote $B = \bar{B}(1 - \beta)^2$, and recall the notation $q = \frac{Q}{1 - \beta}$, which allows the expression of the developer's cost as Bq^2 , a form we will prefer with imperfect foresight¹⁷.

Notation 3. *The number of high-end users, d_1 , will be also denoted N_H . The average number of low-end users who switch in a given period, $\sum_{n=2}^{\infty} \frac{1}{n} d_n$, will be*

¹⁷Parameter \bar{B} also effectively captures market size, i.e., the density of θ being a constant other than exactly 1. In such a case, the demand d_n will be multiplied by that constant, which is equivalent (as far as profit maximization is concerned) to dividing \bar{B} by the constant in question.

denoted N_L . The average number of users who switch in a given period will be denoted $N = N_H + N_L$.

Remark 5. *Introducing discounting $0 < \delta < 1$ would not lead to analytical solution and would require additional restriction on users without any impact on key results.*

2.4 Welfare

To measure the efficiency of the monopolist developer, we traditionally use social welfare, which consists of consumer surplus and the developer's profit. We use the average per-period welfare. Recall that consumer surplus is calculated based on $q_t = \frac{1}{1-\beta} Q_t$, so it is already discounted, and profits are not discounted by assumption. According to Proposition 2.2, every user starts buying at some finite period and then buys regularly. Then consider the user with quality sensitivity θ , and let $n(\theta)$ and $p(\theta)$ be the frequency and the price at which this user switches to new versions of the product. Then the additional utility accruing to this user at every purchase is $\theta n(\theta)q - p(\theta)$, which corresponds to the participation constraint, and the per-period additional utility equals $\theta q - \frac{p(\theta)}{n(\theta)}$. Therefore, the average per-period consumer surplus equals

$$CS = \int_0^1 \left(\theta q - \frac{p(\theta)}{n(\theta)} \right) d\theta = \frac{q}{2} - \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta.$$

The developer's profit equals revenue minus cost, and the per-period revenue generated by a user with quality sensitivity θ , who switches to a new version every $n(\theta)$ periods at price $p(\theta)$, equals $\frac{p(\theta)}{n(\theta)}$, whence the per-period revenue equals

$$\int_0^1 \frac{p(\theta)}{n(\theta)} d\theta,$$

and the per-period cost equals $\bar{B}Q^2 = Bq^2$.

Thus, the per-period social welfare equals

$$W = CS + \Pi = \frac{q}{2} - \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta + \int_0^1 \frac{p(\theta)}{n(\theta)} d\theta - Bq^2 = \frac{q}{2} - Bq^2. \quad (15)$$

Proposition 2. *The socially optimal quality adjustment is given by*

$$q_0 = \frac{1}{4B} \iff Q_0 = \frac{1}{4B(1-\beta)}.$$

Proof. The claim directly follows from (15). ■

In the third period, the developer releases a software version of quality $3q$ for the same price p . A user who does not own any version of software yet, buys the new version if his sensitivity is $\theta \geq \frac{p}{3q}$ according to (3). A user who is already using software decides according to equation (4). After the second period, users with $\theta \geq \frac{p}{2q}$ are using software of quality $2q$, so the new version of software $3q$ is bought only by users satisfying $\theta \cdot 3q - p \geq \theta \cdot 2q$ (by those with $\theta \geq \frac{p}{q}$). The distribution of software is depicted in Figure 3.

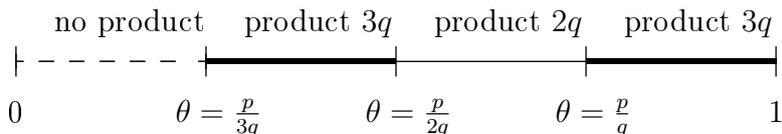


Figure 3: The distribution of products after the third period

When we look at the distribution of software version across users, we see that after the third period software of quality $3q$ is used by users with $\theta \in \langle \frac{p}{3q}, \frac{p}{2q} \rangle \cup \langle \frac{p}{q}, 1 \rangle$, software of quality $2q$ is used by users with $\theta \in \langle \frac{p}{2q}, \frac{p}{q} \rangle$ and no one is using version of quality q from the first period. Users with sensitivity to quality $\theta \in \langle 0, \frac{p}{3q} \rangle$ are not using software at all and wait until the next periods.

For further analysis of software distribution across users, we have to look separately at each group of users $\theta \in \langle \frac{p}{(n+1)q}, \frac{p}{nq} \rangle$ where $n \in \langle 1, \infty \rangle$. For each group, we look which version they are using and when they actually upgrade (replace) the product. Users with $\theta = \frac{p}{nq}$ are marginal users who are, at selected periods, indifferent between switching to the newest version or staying with an older version (or not using the software at all). Continuing with the same approach, we obtain the following distribution of versions across users (see Figure 4).

The analysis presented can be summarized by the following proposition.

Proposition 3. *Users of the highest sensitivity to quality, that is, $\theta \in [\frac{p}{q}, 1]$, buy a new version of the software every period and users with $\theta \in \langle \frac{p}{nq}, \frac{p}{(n-1)q} \rangle$ buy a new version every n th period. As $n \rightarrow \infty$, the measure of those who do not use any version of software, $\theta \in \langle 0, \frac{p}{nq} \rangle$, goes to zero.*

It is interesting that the set of consumers who own a particular software quality is generally non-convex, and it is possible that in a given period a consumer with a higher quality sensitivity owns a lower quality version, as can be seen in Figure 4. However, as we are interested in average per-period values, what really matters is how often consumers upgrade. According to the proposition above, the set of consumers with the same updating frequency is convex, and (by the general result in

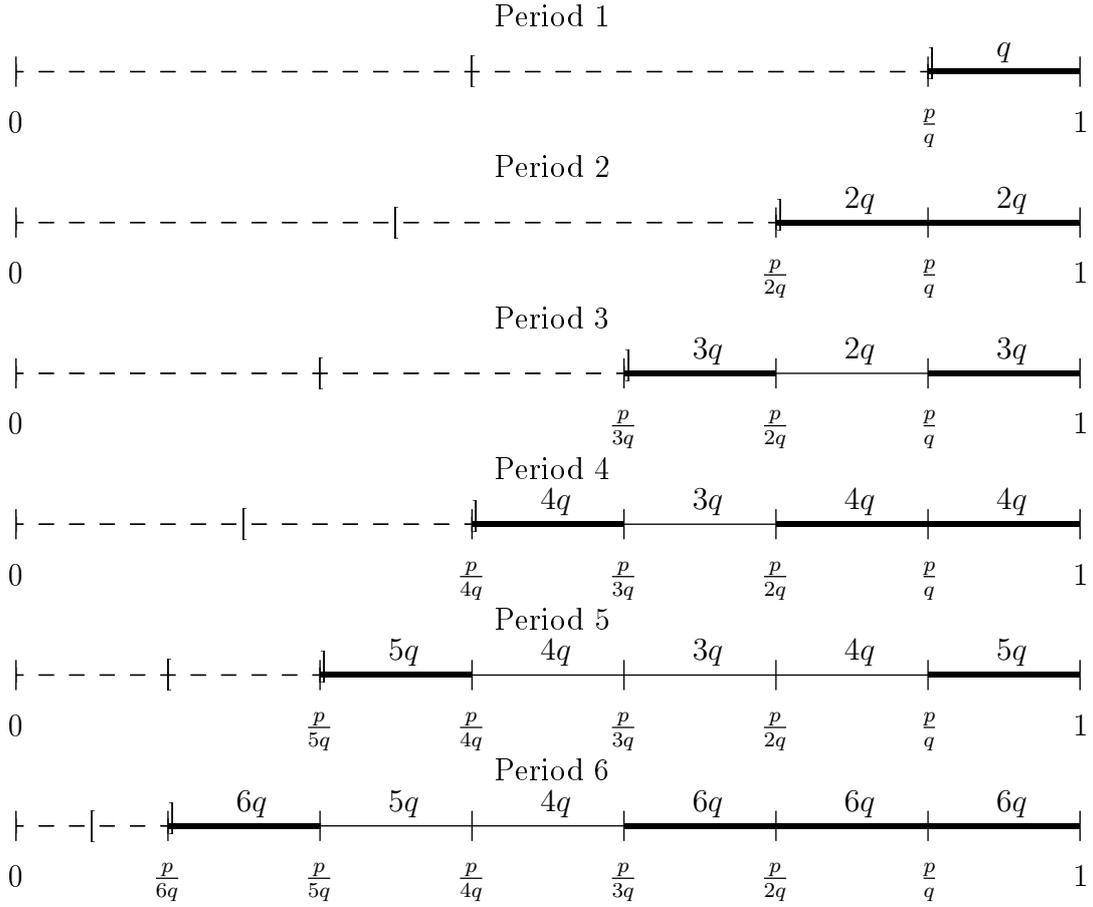


Figure 4: The distribution of products across users

Section 2) this frequency is non-increasing in quality sensitivity. Average frequency of switching is displayed in figure 5.

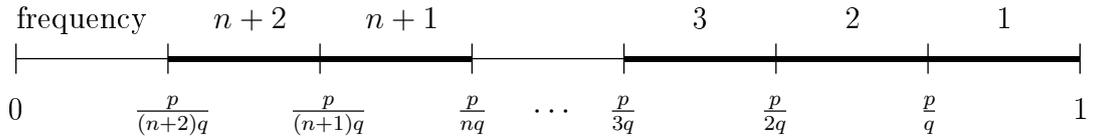


Figure 5: The distribution of switching frequencies across users

This is equivalent to the following demand structure.

$$d_1 = 1 - \frac{p}{q}, d_n = \frac{p}{(n-1)q} - \frac{p}{nq} = \frac{p}{q} \left(\frac{1}{n(n-1)} \right), n \geq 2. \quad (16)$$

3.2 The developer's problem

The developer's profit per period is generally given by (14), which, after taking into account the single price policy and substituting (16), takes the form

$$\Pi = p \left(1 - \frac{p}{q}\right) + p \sum_{n=2}^{\infty} \frac{p}{q} \frac{1}{n^2(n-1)} - Bq^2.$$

Denote

$$D = \sum_{n=2}^{\infty} \frac{1}{n^2(n-1)} = 2 - \frac{\pi^2}{6} \approx 0.355,$$

so that the profit can be written as

$$\Pi = \left(1 - \frac{p}{q}\right)p + \frac{p}{q}Dp - Bq^2. \quad (17)$$

The developer maximizes the profit by setting optimal p and q .

Proposition 3. *If the users have imperfect foresight and the developer uses the single price policy, then the developer's choice of price and quality, and the implied price-quality ratio, are the following.*

$$p^* = \frac{1}{16(1-D)^2 B}, \quad q^* = \frac{1}{8B(1-D)}, \quad \frac{p^*}{q^*} = \frac{1}{2(1-D)}. \quad (18)$$

In addition, equilibrium numbers of users are

$$N_H^* = 1 - \frac{p^*}{q^*} = \frac{1-2D}{2-2D}, \quad N_L^* = \frac{p^*}{q^*}D = \frac{D}{2(1-D)}, \quad N^* = N_H^* + N_L^* = \frac{1}{2}, \quad (19)$$

so that and exactly half of the users (on average) are switching every period.

Proof. The optimal price and quality are obtained from F.O.C.; S.O.C. are checked in Appendix 11.3. Equilibrium numbers of users are obtained directly from (18). ■

Substituting D into the equilibrium, we obtain:

$$p^* = \frac{9}{4B(\pi^2 - 6)^2} \simeq \frac{0.150262}{B}, \quad (20)$$

$$q^* = \frac{3}{4B(\pi^2 - 6)} \simeq \frac{0.193818}{B}, \quad (21)$$

$$\frac{p^*}{q^*} = \frac{3}{\pi^2 - 6} \simeq 0.775273. \quad (22)$$

(Note that the last equality also proves that the assumption $\frac{p}{q} < 1$ was correct.)

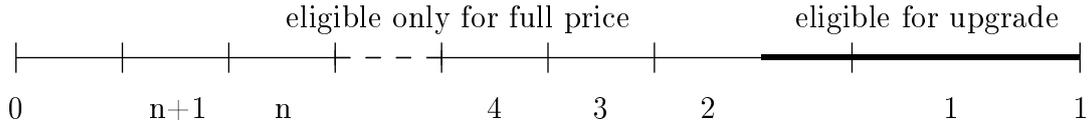


Figure 6: Market distribution in non-lock-in

Substituting back into (17), we obtain the developer’s profit:

$$\Pi^* = \frac{1}{64B(1-D)^2} = \frac{9}{16B(\pi^2 - 6)^2} \simeq \frac{0.037565}{B}. \quad (23)$$

Summary 1. *In this section, we derived monopoly equilibrium and distribution of user if only one price is allowed. As for the equilibrium price, this is in line with Stokey (1979, 1981) who shows that in a dynamic durable-good context when the developer can commit to the time path of prices, the monopolist precommits to the same price in all periods, which coincides with the static monopoly price. Like in many static models, the developer sells on average to exactly half of all users; so equilibrium price and quality are set to reach those half of the users every period. The quality upgrading that the monopolist chooses is below the social optimum due to the standard argument that the single - price monopolist cannot appropriate the whole surplus it generates. Finally, note that the proportion of the users who buy software is independent on factor B .*

4 Imperfect foresight users and price discrimination

In this section, we analyze the situation when the developer offers a lower “upgrade” price p_1 to users who own the version from the previous period. The rest of the users, who are using older products, are not eligible for the lower price and could buy the new version only for a standard “full” price p_2 . Denote the quality offered by the developer in this section Q_e , and $q_e = \frac{Q_e}{1-\beta}$.

If the upgrade price p_1 is only slightly lower than the full price p_2 , some users who would buy every second period for price p_2 now prefer buying every period for price p_1 ; however, all users who buy every three periods do not change their decision and still buy every three periods. This leads to market coverage as in Figure 6.

In the case of a higher discount for upgrading users, not only those who would buy for price p_2 every two periods, but even users who would buy less frequently than every $n \geq 3$ periods would now switch every period. This market situation leads to market coverage as in Figure 7.

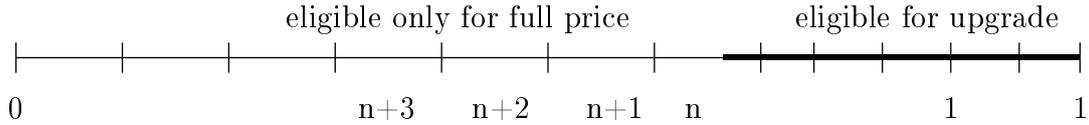


Figure 7: Market distribution in lock-in

We shall refer to the former case, where a user with all switching frequencies are present on the market, as to the “non-lock-in” case, while the latter case, where some frequencies of switching are out of the market, will be referred to as the “lock-in” case. We shall analyze both cases, and we start with the non-lock-in market.

4.1 The “non-lock-in” set-up

Assume in this part that the upgrade price is $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$. This condition will guarantee that all frequencies of switching are present in the market. (We show later that this condition is satisfied in equilibrium.) Let us separate the revenue generated by upgrade versions and by the full versions. We see that only users with sensitivity parameter θ greater than $\frac{p_1}{q_e}$ buy every period. These users generate revenue denoted as R_1 . The rest of the users, who are not eligible for the upgrade price, generate revenue R_2 . The full-price demand is similar to the single price case as in equation (16) with the exception that some users who would switch every two periods at p_2 now switch every period at p_1 . The measure of those users is $\left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)$ and since their frequency of switching was every second period in the case of a single price, we have to adjust the average demand per period for the term $\frac{1}{2} \left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)$. Thus, the revenue functions are:

$$R_1 = \left(1 - \frac{p_1}{q_e}\right) \cdot p_1, \quad (24)$$

$$R_2 = \left(\frac{p_2}{q_e} D - \frac{1}{2} \left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)\right) \cdot p_2,$$

and the developer’s profit is:

$$\Pi = \left(\frac{p_2}{q_e} D - \frac{1}{2} \left(\frac{p_2}{q_e} - \frac{p_1}{q_e}\right)\right) \cdot p_2 + \left(1 - \frac{p_1}{q_e}\right) \cdot p_1 - Bq_e^2. \quad (25)$$

Remark 6. *Imperfect foresight means that the users with θ only slightly higher than $\frac{p_1}{q_e}$, one period after buying, will find it profitable to buy the newest version at p_1 , so they will buy. These users do not consider the possibility that they can be better off buying every two periods at p_2 rather than every period at p_1 whether it*

can actually happen or not.

Proposition 5. *If the users have imperfect foresight, and the developer price discriminates as described, then the developer's choice of prices and quality is the following.*

$$p_1^* = 4 \frac{(1-2D)^2}{(16D-7)^2 B}, \quad p_2^* = 2 \frac{1-2D}{(16D-7)^2 B}, \quad q_e^* = \frac{2D-1}{B(16D-7)}. \quad (26)$$

The equilibrium numbers of users are

$$\begin{aligned} N_H^* &= \frac{3-8D}{7-16D} \simeq 0.120909, \\ N_L^* &= \frac{2D-1}{16D-7} \simeq 0.219772, \\ N^* &= 2 \frac{2-5D}{7-16D} = \frac{5\pi^2-48}{8\pi^2-75} \simeq 0.340681. \end{aligned} \quad (27)$$

Proof. The optimal prices and quality are obtained from F.O.C.; S.O.C. as well as the condition $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$ are checked in Appendix 11.4. The numbers of users are obtained directly from these values. ■

Substituting D into the equilibrium, we obtain:

$$p_1^* = \frac{4}{B} \frac{(\pi^2-9)^2}{(8\pi^2-75)^2} \simeq \frac{0.193200}{B}, \quad (28)$$

$$p_2^* = \frac{6}{B} \frac{(\pi^2-9)}{(8\pi^2-75)^2} \simeq \frac{0.333255}{B}, \quad (29)$$

$$q_e^* = \frac{(\pi^2-9)}{B(8\pi^2-75)} \simeq \frac{0.219772}{B}, \quad (30)$$

$$\frac{p_1^*}{q_e^*} = \frac{4\pi^2-36}{8\pi^2-75} \simeq 0.879090, \quad (31)$$

$$\frac{p_2^*}{q_e^*} = \frac{6}{8\pi^2-75} \simeq 1.516363. \quad (32)$$

substituting (26) into (25), we obtain the developer's profit:

$$\Pi^{D*} = \frac{1}{B} \frac{(2D-1)^2}{(16D-7)^2} = \frac{1}{B} \frac{(\pi^2-9)^2}{(8\pi^2-75)^2} \simeq \frac{0.048300}{B}. \quad (33)$$

Lemma 1. *If the users have imperfect foresight, and the developer uses either the single-price policy or "non-lock-in" price discrimination, then all equilibrium prices, qualities, and profits are dependent on the discount factor β , and all of them are increasing in β . On the other hand, the numbers of switching users N_H^* , N_L^* , N^* are*

independent on β .

Proof. Can be seen immediately from the results above by recalling that $B = \bar{B}(1 - \beta)^2$ and $q = \frac{1}{(1-\beta)}Q$. ■

Remark 7. *If we compare profit made by a single price developer and a developer who undertakes price discrimination based on upgrades, we notice that the profit for price discrimination is higher in the latter case. Clearly, the discriminating developer always has the option to set prices equally $p_1 = p_2$. The exact relationship between the profits is given by:*

$$\Pi^{D^*} - \Pi^* = \frac{1}{B} \frac{(2D - 1)^2}{(16D - 7)^2} - \frac{1}{64B(1 - D)^2} \simeq \frac{0.010734}{B} > 0. \quad (34)$$

Remark 8. *In our analysis, we neglect the fact that in the first period all users should pay full price, including the users who will buy regularly every period afterwards. This simplification is in line with the basic form (13) for profit, which implies that any finite number of periods in the beginning can be neglected. Note again, that imperfect foresight users do not predict future switching to a new version so, for instance, they do not consider the situation that under certain prices p_1, p_2 they can buy a product for the full price p_2 with negative immediate utility and offset it later on with the future positive utility flow from upgrades. Such considerations will be analyzed in the case of perfect foresight.*

4.2 The “lock-in” set-up

In the case of lock-in, we deal with different optimization problems. There are only users who switch every period, and then there are users who switch less frequently than every two periods. Users switching exactly every two periods are not in the market. Consider now the general case, where only users switching every period and then every n or more periods are in the market (further referred to as n -lock-in). Assume $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$. (Again, we will show later that this condition is satisfied in equilibrium.)

Remark 9. *Note that from a mathematical point of view, non-lock-in is a particular case of lock-in at $n = 2$ (“2-lock-in”).*

Denote $\theta_{1,n}$ a user who is indifferent between switching every period and every n periods, and $\theta_{n,n+1} = \frac{p_2}{nq_e}$ is the user who is indifferent between switching every n and every $n + 1$ periods. Denote again revenue from users switching every period as $R_1 = (1 - \theta_{1,n}) \cdot p_1$. From users utility function (3) in view of Remark ??, we see that a user indifferent between switching every period and every n periods is a user with sensitivity to quality $\theta_{1,n} = \frac{p_1}{q_e}$, so the revenue from users switching every

period is $R_1 = \left(1 - \frac{p_1}{q_e}\right) \cdot p_1$.

Revenue from users switching less frequently consists of revenue from users who switch every n periods: $\frac{1}{n}(\theta_{1,n} - \theta_{n,n+1})p_2$ and from users switching even less frequently $\frac{p_2}{q_e}D_{n+1}p_2$:

$$R_2 = \frac{1}{n}(\theta_{1,n} - \theta_{n,n+1})p_2 + \frac{p_2}{q_e}D_{n+1}p_2,$$

where

$$D_{n+1} = \sum_{m=n+1}^{\infty} \frac{1}{m^2(m-1)} = \frac{1}{n} - \psi_1(n+1),$$

where $\psi_1(\cdot)$ is the polygamma function of order 1.

Remark 10. *As could be seen directly from the definition, D_n is decreasing and goes to zero as n goes to infinity.*

Summing up the revenue for both groups R_1 , R_2 , taking into account the developer's costs Bq_e^2 , and re-arranging, we obtain the developer's profit function:

$$\Pi = \frac{p_2}{nq_e}(p_1 + p_2 - np_2\psi_1(n)) + \left(1 - \frac{p_1}{q_e}\right)p_1 - Bq_e^2. \quad (35)$$

Proposition 6. *If the users have imperfect foresight and the developer price discriminates so that no user switches every $2, \dots, n-1$ periods, then the developer's choice of prices and quality is the following.*

$$p_1^* = \frac{n^2}{B} \frac{(n\psi_1(n) - 1)^2}{(4n^2\psi_1(n) - 4n - 1)^2}, \quad p_2^* = \frac{n^2}{2B} \frac{n\psi_1(n) - 1}{(4n^2\psi_1(n) - 4n - 1)^2},$$

$$q_e^* = \frac{n}{2B} \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1},$$

and the average number of switching users per period is:

$$N^{D*} = \frac{n(1 + 2n)\psi_1(n) - 2(1 + n)}{4n^2\psi_1(n) - 4n - 1}. \quad (36)$$

Proof. The optimal prices and quality are obtained from F.O.C.; S.O.C as well as the condition $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$ are checked in Appendix 11.4. The value N^{D*} is then derived by substitution. ■

Substituting p_1^*, p_2^*, q_e^* into the profit function (35), we obtain the equilibrium profit:

$$\Pi^* = \frac{n^2}{4B} \frac{(n\psi_1(n) - 1)^2}{(4n^2\psi_1(n) - 4n - 1)^2}.$$

Remark 11. *By substituting p_1^*, p_2^*, q_e^* into the definition of indifferent users, we see that in equilibrium the indifferent users are:*

$$\theta_{1,n}^* = \frac{p_1^*}{q_e^*} = \frac{1}{2} \left(1 + \frac{1}{4n^2\psi_1(n) - 4n - 1} \right), \quad \theta_{n,n+1}^* = \frac{p_2^*}{nq_e^*} = \frac{1}{4n^2\psi_1(n) - 4n - 1}.$$

Lemma 2. *If the developer uses n -lock-in under imperfect foresight, then as n increases, the average number of switching users decreases, and equilibrium quality change, both prices, both prices per quality change, and the developer's profit increase. In addition, the equilibrium quality converges from below to the socially optimal value $q_0 = \frac{1}{4B}$ as n increases.*

Proof. Directly follows from the results above. ■

The last result implies that the best strategy for the developer in our model is to set n close to infinity. This result is caused by the set-up of our model, where the developer is maximizing average profit. Thus, the developer does not differentiate between profit realized in the first period and profit from a period close to infinity. Maximizing the average profit instead of the discounted flow of profit is a simplification to obtain an analytical solution¹⁸. However, remark that for those who switch every n periods, the developer starts receiving revenue from this group for the first time at period n ; thus, it would be natural to assume that the profit from this group is discounted by β^n , which would immediately imply that for $\beta < 1$, the optimal n is finite.

5 Imperfect foresight users: comparisons and welfare analysis

5.1 A single price versus price discrimination

In this section, we contrast price discrimination with the single price equilibrium, so we compare changes in quality, prices, prices per unit of quality change $\frac{p^*}{q^*}$, the total number of switching users, and welfare for both cases. We do the comparison for the “non-lock-in” case and then extend the results to the general “lock-in” case via Lemma 4.2. The proofs of the results stated here are made by direct arithmetical comparison of the relevant results above and are thus omitted.

Lemma 3. *The product quality in the case of a single price is lower than the quality set by the developer discriminating the users based on upgrades.*

This result is in line with the general view of the impact of price discrimination.

¹⁸Introducing the discounted developer profit would require a numerical solution even for the simplest possible case of the single price and imperfect foresight.

From a purely quality point of view, the developer using price discrimination based on a lower price of the upgrade accelerates quality evolution more than a single price monopoly.

Lemma 4. *If price discrimination based on a lower upgrade price is not possible, the single price p^* is lower than the full price p_2^* , and even lower than the price of an upgrade version p_1^* .*

This result may be counter-intuitive at first sight as we should be aware that the quality $q_e^* > q^*$, so the higher price even for upgrades is justified by a quality change.

Another insightful comparison we obtain if we compare the price per quality¹⁹, $\frac{p^*}{q^*}$.

Lemma 5. *The equilibrium price per quality $\frac{p^*}{q^*}$ is lower if possibility for discrimination does not exist, that is $\frac{p^*}{q^*} < \frac{p_1^*}{q_e^*} < \frac{p_2^*}{q_e^*}$. Consequently, the number of buyers is higher for a developer that charges the single price.*

Lemma 6. *The proportion of users buying in every period in the case of a single price is higher than in the case of price discrimination.*

This is a very interesting result. Generally, the motivation for price discrimination is to increase revenue by increasing the number of users. In standard price discrimination, the price for a more sensitive group is lower than for the other groups, and it is profitable to sell till the price reaches marginal costs.

On the other hand, a lower price for upgrades persuades some users to buy more frequently which, in turn, raises the number of users. The key factor of this price discrimination is that the discount is not offered to the most price sensitive users but to the most quality sensitive users. A possible interpretation of this is that price discrimination by using upgrades helps the developer to better separate the quality sensitive users buying in every period and to accelerate software evolution (recall that the quality change is higher under price discrimination) to fit their demand better. As for the low-end users, the full price for the software is set very high because the developer knows that even though their sensitivity for quality is lower, the fact that they possess an older version means that their valuation of the new software eventually becomes relatively high; thus, these users will buy sometimes as well. Accelerating software evolution enables the developer to charge a higher price to all of them. In a single-price model, the developer values low-end users relatively more, so he must attract a higher number of users by keeping both price and quality lower.

¹⁹Though in the literature the inverse, i.e., quality per price unit, $\frac{q}{p}$ is usually used for comparison, we use $\frac{p}{q}$ as it plays a key role in the distribution of switching customers in our model.

Remark 12. *Note, that in our comparison, when we compare single price and price discrimination equilibria, we have different equilibrium qualities $q^* < q_e^*$. Thus, the higher prices (and the lower number of users) in the price discrimination case is partially caused by a higher equilibrium quality. For a better understanding of this effect, we can decompose “moving” from the single price equilibrium to the price discrimination equilibrium into two steps. In the first step, we fix quality at the single price equilibrium level q^* while allowing price discrimination, and in the second step, we adjust quality (and prices) to their profit-maximizing levels. This analysis is performed in Appendix 11.5.*

The results above apply to the “non-lock-in” case. It follows from Lemma 4.2 that if the developer decides to use n -lock-in, the results above are reinforced as n increases.

5.2 Welfare analysis

In this section, we analyze welfare changes by comparing price discrimination and single-price equilibria. We analyze consumer surplus (CS) for all, low-end and high-end users, as well as social welfare W , which is consumer surplus plus the developer’s profit. For all these variables, we compare their average per-period values (in fact, all these differ from period to period). The values for a single-price developer can be found in Appendix 11.5, and the values for a price discriminating developer can be found in Appendix 11.5. In particular, the average CS per period under a single price equals

$$CS = \frac{1}{32} \frac{1 - 2D}{(1 - D)^2 B} \approx \frac{0.021778}{B},$$

and under price discrimination with n -lock-in, the average CS per period equals

$$CS = \frac{1 - (4n^2\psi_1(n) - 4n - 1)^{-2}}{32B}.$$

The following results are proved by direct comparison of the values derived in Appendices 11.5 and 11.5.

Proposition 7. *The consumer surplus is higher in the case of a single-price monopoly than under non-lock-in price discrimination and decreases in n as n -lock-in is used.*

The consumer surplus for both high-end and low-end users is higher in the case of a single-price developer than under non-lock-in price discrimination, and decreases in n , and the same applies to CS per buyer for high-end users. As for consumer surplus per buyer for all users or for low-end users, they are higher under n -lock-in

than under the single price except for $n = 2$ (non-lock-in) and $n = 3$, and increase in n .

Social welfare is lower in the case of a single-price developer than under non-lock-in price discrimination, increases in n as n -lock-in is used, and approaches the socially optimal level from below as n goes to infinity.

The different relation between CS per buyer for all or low-end users under a single price and under n -lock-in at $n = 2$ and $n = 3$ shows that low end users are less important in lock in and are forced to switch less often (have higher price and quality) and those users are less as n increases, at the same moment developer does not increase the price per quality correspondingly which results to higher CS per buyer —recall that the developer prefers “infinite” lock-in so that in the end those CS are higher under price discrimination. See the discussion in section 8 for better exposition.

The intuition is that as the length of the lock-in period increases, the developer becomes “more precise” in the sense of extracting more and more consumer surplus by setting the appropriate price-to-quality ratio since the targeting group (upgraders) becomes more narrow. Thus, per-period CS in the lock-in case tends to zero as n increases to infinity. In the limit (which is mathematically unreachable), the monopolist makes all consumers pay a price of q^0 per period (i.e., p_2 has an asymptotic behavior of nq^0), where q^0 is also the socially optimal value of quality. Thus, much like in the case of the first degree price discrimination, the developer can in the limit extract the entire consumer surplus. Unlike in the case of the first degree price discrimination, however, this happens due to the dynamic nature of the model: as n increases, the size of the largest consumer group (in terms of switching frequency) in the market decreases, and it is easier to extract surplus from separate smaller groups. This can be interpreted as an example of first-degree price discrimination as the limiting case of third-degree one (see also the Remark before Proposition 11).

6 Perfect foresight users: single price developer

6.1 The user decision

In this section, we analyze the developer’s behavior if all users have perfect foresight. The rest of the set-up is the same as in the previous part, which means that in every period t , the developer introduces a new version of software and sells only this new version. Every new version has a quality improvement Q over the previous version, so that at period t the software quality is $Q_t = tQ$. The decision process for perfect

foresight users is the same as for users with imperfect foresight: users compare the utility flow from keeping the currently possessed version with the utility flow from switching to a new version. In the case of perfect foresight, however, every user can anticipate the optimal frequency of switching to a new version. A user of quality sensitivity θ with perfect foresight calculates the utility flow $U_n(\theta)$ from switching every n periods (for all $n \in \langle 1, \infty \rangle$), and then he decides for such switching frequency n that brings him the maximal utility flow. Naturally, the optimal frequency of switching is fully dependent on the sensitivity to quality θ .

By perfect foresight user, we mean a user who calculates the utility flow and the optimal frequency of switching at the moment of buying a new version and later follows this decision. Alternatively, a user may calculate the utility flow every period and see whether switching would bring him higher utility or whether he should wait till the next period. When such a user calculates the utility flow every period, he takes into account that he will follow the same decision in the future so he incorporates into the calculation his future decisions that are aligned with his current decision. In the case of a single price, these two approaches are equivalent (see Remark 6.1).

According to the analysis presented in section 2, the necessary condition for a user with perfect foresight to switch every n periods at price p is $\theta nq - p \geq 0$, and while this necessary condition is the same as under imperfect foresight, it is no longer sufficient. As mentioned above, when a user decides for an optimal frequency of switching, he must compare the utility flow for all possible frequencies of switching n , where for each frequency he calculates the infinite-time utility flow. When doing so, the user must calculate the utility flow with respect to all possible future switching. To decide between two possible frequencies of switching, the user can just compare the utility flow between two periods when he would always switch to a new version, no matter which of the two frequencies of switching he would select. The simplest example is the decision between switching every period and every two periods, which yields the threshold of $\theta_{12} = \frac{p}{Q}$ in (12) as analyzed in Chapter 2.

For another example, consider a user with quality sensitivity θ who is comparing the utility flow from switching every two periods and every three periods. Assuming that the user decides to buy in period t , the periods when this user always switches to a new version, no matter whether he decides for frequency 2 or 3, are periods $t + 6, t + 12, t + 18, \dots$. If the utility flow from one of those frequencies is higher for the next 6 periods, then this user naturally prefers this frequency in all of his future decision periods $t + 6, t + 12, t + 18, \dots$. Thus, to compare which frequency of switching is better, it is enough in the example above to compare the utility flow

from 6 periods since the decision will be regularly repeated every 6 periods.

Assume now that version available in the market in period t has quality tQ . Buying every three periods implies switching two times to a new version within six periods. For the first time, the user switches in period t to the version of quality tQ and pays the price p , and for the second time, he switches at period $t + 3$ to the new version of quality $(t + 3)Q$ and pays the price p again. Thus, at time t , the discounted utility flow from the next six periods is:

$$U_{n=3}(\theta) = \theta (tQ + tQ\beta + tQ\beta^2 + (t + 3)Q\beta^3 + (t + 3)Q\beta^4 + (t + 3)Q\beta^5) - p - p\beta^3. \quad (37)$$

Similarly, switching every two periods requires switching at periods, $t, t + 2, t + 4$, so that the user obtains the following utility flow from the next 6 periods:

$$U_{n=2}(\theta) = \theta (tQ + tQ\beta + (t + 2)Q\beta^2 + (t + 2)Q\beta^3 + (t + 4)Q\beta^4 + (t + 4)Q\beta^5) - p - p\beta^2 - p\beta^4. \quad (38)$$

Comparing the utility flows $U_{n=3}(\theta)$ and $U_{n=2}(\theta)$, the user sees which frequency of switching is better for him. From the above utility flows $U_{n=3}(\theta)$ and $U_{n=2}(\theta)$, we can immediately derive that the user indifferent between switching every two and every three periods has quality sensitivity

$$\theta_{23} = \frac{p}{Q} \frac{1}{(\beta + 2)}, \quad (39)$$

where the user with quality sensitivity θ prefers switching every two periods to switching every three periods if $\theta > \theta_{23}$ and switching every three periods to switching every two periods if $\theta < \theta_{23}$.

Remark 13. *In this example, the decision process is equivalent to an alternative decision process when a user with quality sensitivity θ at period t possesses the product from period $t - 2$ and decides whether to switch or to wait till the next period. The utility flows are then:*

$$U_{n=3} = \theta((t - 2)Q + (t + 1)Q\beta + (t + 1)Q\beta^2 + (t + 1)Q\beta^3 + (t + 4)Q\beta^4 + (t + 4)Q\beta^5) - \beta p - p\beta^4,$$

and

$$U_{n=2} = \theta (tQ + tQ\beta + (t+2)Q\beta^2 + (t+2)Q\beta^3 + (t+4)Q\beta^4 + (t+4)Q\beta^5) - p - p\beta^2 - p\beta^4.$$

Equating the two expressions above and solving for θ , we see again that the indifferent user is $\theta_{23} = \frac{p}{Q(\beta+2)}$. The same result can be derived for the case when the user owns a version from period $t-1$ and chooses between waiting one or two periods. Thus, the approaches to the decision process are equivalent, and it is not important in which period the user makes a decision about the optimal frequency of switching.

Remark 14. From the derivation of the indifferent user θ_{23} and from the previous remark, we see that the user's decision about optimal switching to a new version is dependent only on β , Q , and p , but independent on the decision period t .

Lemma 7. Given the discount factor β , the software price p , and the per-period quality improvement Q , the user indifferent between switching every n and every $n+1$ periods has quality sensitivity

$$\theta_{n,n+1} = \frac{p}{Q} \frac{(1-\beta)^2}{n(1-\beta) - \beta(1-\beta^n)}. \quad (40)$$

The users with $\theta > \theta_{n,n+1}$ strictly prefer switching every n periods to switching every $n+1$ periods, and the users with $\theta < \theta_{n,n+1}$ strictly prefer switching every $n+1$ periods to switching every n periods. This value does not depend on which version, if at all, is possessed by the user at any given time. Moreover, this threshold decreases in β , is not lower than the corresponding imperfect foresight value (with p and Q fixed), and the utility flow to the indifferent user is non-negative so that the participation constraint holds.

Proof. The derivation of the threshold (40) and the rest of the proof can be found in Appendix 11.6. ■

Remark 15. Note that at $\beta = 0$, (40) yields $\theta_{n,n+1} = \frac{p}{nQ}$, which is the imperfect foresight value (at $\beta = 0$, $q = Q$), and the limiting value at $\beta \rightarrow 1$ is $\theta_{n,n+1} = \frac{p}{Q} \frac{2}{n+n^2}$.

6.2 Product distribution across users

Users who switch every n periods are those whose quality sensitivities satisfy $\theta \in (\theta_{n,n+1}, \theta_{n-1,n})$; thus, substituting (40), we see that the users who switch every n

periods are those with θ from the interval:

$$\theta \in \left(\frac{p}{Q} \frac{(1-\beta)^2}{n(1-\beta) - \beta(1-\beta^n)}, \frac{p}{Q} \frac{(1-\beta)^2}{(n-1)(1-\beta) - \beta(1-\beta^{n-1})} \right), \quad (41)$$

and the number of users switching with every n periods is

$$N_n = \theta_{n-1,n} - \theta_{n,n+1} = \frac{p}{Q} \frac{(1-\beta)^3(1-\beta^n)}{(1-\beta^n - n(1-\beta))(\beta(1-\beta^n) - n(1-\beta))}. \quad (42)$$

The previous equations (40), (41), (42) are valid for all users who switch with frequency $n \in \langle 2, \infty \rangle$. Users who switch every period are just from interval $\theta \in \langle \frac{p}{Q}, 1 \rangle$ (note that $\theta_{12} = \frac{p}{Q}$) and their number is $N_1 = 1 - \frac{p}{Q}$. Summing all switching users over all $n \in \langle 1, \infty \rangle$ together, we obtain that, on the average, the number of switching users per period is:

$$\begin{aligned} N &= \left(1 - \frac{p}{Q}\right) + \sum_{n=2}^{\infty} \frac{1}{n} N_n \\ &= \left(1 - \frac{p}{Q}\right) + \frac{p}{Q} \sum_{i=2}^{\infty} \frac{1}{i} \frac{(1-\beta)^3(1-\beta^i)}{(1-\beta^i - i(1-\beta))(\beta(1-\beta^i) - i(1-\beta))}. \end{aligned}$$

For simplicity, denote

$$L(\beta) = \sum_{i=2}^{\infty} \frac{1}{i} \frac{(1-\beta)^3(1-\beta^i)}{(1-\beta^i - i(1-\beta))(\beta(1-\beta^i) - i(1-\beta))}. \quad (43)$$

Then, given the price p , the quality improvement Q , and the discount factor β , the number of switching users per period on the average, is

$$N = \left(1 - \frac{p}{Q}\right) + \frac{p}{Q} L(\beta). \quad (44)$$

Lemma 8. $L(\beta)$ is increasing in β , with $L(0) = D = 2 - \frac{1}{6}\pi^2 \approx 0.355066$ and, in the limit, $L(1) = 7 - \frac{2}{3}\pi^2 \approx 0.420264$.

Proof. Equation (43) cannot be generally expressed analytically for β other than

0 and 1. To verify the statement, we used a numerical simulation where we showed that $L(\beta)$ increases in β . Selected simulation results can be found in Appendix 11.8. The limiting values can be derived using (42) and Remark 6.1. ■

Remark 16. *It is clear that the introduction of perfect foresight has an impact on the distribution of indifferent users. If the prices and quality adjustment are fixed, then the threshold $\theta_{n,n+1}$ is higher under perfect foresight. This means that the proportion of users switching less frequently is higher than in the case of imperfect foresight.*

6.3 Equilibrium

From a mathematical point of view, the problem faced by the single-price developer when the users have perfect foresight is the same as in the imperfect foresight case. The only difference, in the single price framework, between perfect and imperfect foresight is in the numerical values of the equilibrium solution. The analytic form remains the same with $L(\beta)$ in the place of D , and the distribution of indifferent users is different. We can use all results from the single price case under imperfect foresight. Thus, using the profit function (17) derived for imperfect foresight users, we obtain:

$$\Pi(p, Q, \beta) = \left(1 - \frac{p}{Q}\right)p + p\frac{p}{Q}L(\beta) - \bar{B}Q^2. \quad (45)$$

Using the results from Chapter 4, we obtain:

$$p^*(\beta, B) = \frac{1}{16(1 - L(\beta))^2 \bar{B}}, \quad Q^*(\beta, B) = \frac{1}{8\bar{B}(1 - L(\beta))}, \quad \frac{p^*}{Q^*} = \frac{1}{2(1 - L(\beta))}, \quad (46)$$

and the equilibrium profit:

$$\Pi^*(\beta, B) = \frac{1}{64\bar{B}(1 - L(\beta))^2}.$$

As β increases, future present value of utility flow increases and for the same quality jump Q^* every user is willing to pay more; thus, the developer can increase price p^* . However from the developer's point of view, it is the same as raising the sensitivity to quality; thus, he can also raise produced quality, which in turn increases his profit. We can summarize this in the following lemma:

Proposition 8. $p^*(\beta)$, $Q^*(\beta)$, $\frac{p^*}{Q^*}(\beta)$, and $\Pi^*(\beta)$ are increasing in discount factor β and decreasing in parameter \bar{B} , and the number of switching users in the case of the single-price is independent on β and is equal to $N^* = \frac{1}{2}$, which is the same number as in the case of an imperfect foresight set-up.

Proof. $L(\beta)$ is increasing in β , and as can be immediately seen from the equilibrium, p^* , Q^* , $\frac{p^*}{Q^*}$, Π^* are decreasing in \bar{B} , and they are increasing in L , so that they are all increasing in β . The value N^* is then obtained directly. ■

7 Perfect foresight users and price discrimination by an upgrade version

7.1 General set-up

In the perfect foresight set-up with discrimination by upgrades, every user sees which frequency of switching is the best for him, and a user who switches to a new version every period for a lower upgrade price p_1 knows that not switching in one period means that in the next period he is not eligible for the upgrade price anymore and should pay the higher (full) price p_2 . If the price p_2 is relatively high with respect to the upgrade price p_1 , then there would be users who would rather switch every period for a lower price p_1 than switch for the full price more frequently than every n periods. In this case, the high difference between the prices p_1 and p_2 crowds out users with lower frequencies of switching, e.g. 2, 3, up to $n - 1$.

Assume that the developer sets prices p_1 and p_2 in a way to crowd out users who would switch every 2, ..., $n - 1$ periods, and so, only users switching every period and every n (and more) periods are on the market. Then denote the user indifferent between switching every period at the upgrade price p_1 and every n periods at the full price p_2 as $\theta_{1,n}$, and the developer's profit function combines the features of the profit function (45) from the perfect foresight set-up with single price as well as the profit function from imperfect foresight with price discrimination:

$$\Pi = (1 - \theta_{1,n}) p_1 + \frac{1}{n} (\theta_{1,n} - \theta_{n,n+1}) p_2 + \frac{p_2}{Q} L_{n+1}(\beta) p_2 - \bar{B} Q^2, \quad (47)$$

where the profit function consist of four parts:

1. $(1 - \theta_{1,n}) p_1$ Revenue generated by users switching every period;
2. $\frac{1}{n} (\theta_{1,n} - \theta_{n,n+1}) p_2$ Revenue generated by users switching exactly every n periods;
3. $\frac{p_2}{Q} L_{n+1}(\beta) p_2$ Revenue generated by users switching less than every n periods; and
4. $\bar{B}Q^2$ Cost of product development with quality jump Q .

Here $L_{n+1}(\beta)$ is analogous to $L(\beta)$ in (43) with the initial $n - 1$ terms of the sum not included. The definition of $L_{n+1}(\beta)$ is thus the following:

$$L_{n+1}(\beta) = \sum_{i=n+1}^{\infty} \frac{1}{i} \frac{(1 - \beta)^3 (1 - \beta^i)}{(1 - \beta^i - i(1 - \beta)) (\beta(1 - \beta^i) - i(1 - \beta))}$$

(note that $L_2(\beta)$ is the same as $L(\beta)$).

Necessary conditions for equilibrium existence

- **Condition 1:** There are users who would switch every period:

$$0 \leq \theta_{1,n} \leq 1. \tag{48}$$

- **Condition 2:** There are users who would switch every n periods, but no user would switch every $n - 1$ periods:

$$\theta_{n,n+1} \leq \theta_{1,n} \leq \theta_{n-1,n}. \tag{49}$$

Remark 17. *The whole problem could be generally solved using Lagrangian multipliers and Kunh–Tucker conditions; however, we approach it by using an unconstrained maximization problem with the further identification of binding constraints that are later incorporated into the decision process. This approach gives more insights on model behavior.*

Remark 18. *Other conditions that must hold are participation constraints $\theta_{m,m+1}mq - p_2 \geq 0$ for $m \geq n$ and $\theta_{1,n}q - p_1 \geq 0$. However, these conditions hold by the construction of the respective thresholds as is shown in the corresponding propositions.*

A user indifferent between switching every period and every n periods

As compared to perfect foresight with a single price, the problem of the indifferent user is now more complicated. If a user owns a version which is more than one period older, then this user is only eligible for the full price p_2 so that the decision on the frequency of switching follows the rule (40) with p_2 instead of p , so that the optimal switching frequency *at the full price* $n = n(\theta, p_2, Q, \beta)$ is derived. However, if the user owns the version from the previous period, then the user is also eligible for the upgrade price p_1 , so that two options are available. First, the user can exercise his upgrade price claim, and if this is optimal, then this will be done in every subsequent period as the same choice will be faced. Second, the user may wait $n - 1$ periods so that the version at hand becomes n periods old and then switch every $n = n(\theta, p_2, Q, \beta)$ periods²⁰.

Lemma 9. *Given the discount factor β , the software prices p_1 and p_2 , and the per-period quality improvement Q , the user indifferent between switching every period at the upgrade price p_1 and in every n periods at the full price p_2 has quality sensitivity*

$$\theta_{1,n} = (1 - \beta) \frac{p_1(1 - \beta^n) - p_2(1 - \beta)\beta^{n-1}}{(1 - \beta^n - n\beta^{n-1}(1 - \beta))Q}. \quad (50)$$

The users with $\theta > \theta_{1,n}$ strictly prefer switching every period at p_1 to switching every n periods at p_2 , and the users with $\theta < \theta_{1,n}$ strictly prefer switching every n periods at p_2 to switching every period at p_1 . Moreover, if n is optimal at price p_2 , then the utility flow to the indifferent consumer is non-negative so that the participation constraint is satisfied.

Proof. The derivation of the threshold (50) and the proof that the participation constraint is satisfied can be found in Appendix 11.7. ■

For the sake of convenience, denote

$$X = \frac{(1 - \beta)^2}{n(1 - \beta) - \beta(1 - \beta^n)}, \quad Y = \frac{(1 - \beta)(1 - \beta^n)}{1 - \beta^n - n\beta^{n-1}(1 - \beta)},$$

$$Z = \frac{(1 - \beta)^2 \beta^{n-1}}{1 - \beta^n - n\beta^{n-1}(1 - \beta)},$$

so that the thresholds can be written as $\theta_{n,n+1} = \frac{p_2}{Q}X$ and $\theta_{1,n} = \frac{p_1}{Q}Y - \frac{p_2}{Q}Z$.

²⁰From the optimality of $n(\theta, p_2, Q, \beta)$, it follows that the user will not consider the options involving either switching (at p_2) at other frequencies or waiting for any other number of periods than $n - 1$ and then switching every n periods.

7.2 Equilibria

The approach for calculating equilibria is the following. Given β , we fix n and look for an equilibrium assuming that conditions (48) and (49) are satisfied. If they indeed hold, we have an interior equilibrium. In case some condition is not satisfied, we incorporate this condition into the profit function and calculate the equilibrium again. Then we look for $n \in (2, \infty)$ such that the developer maximizes his profit. Unfortunately, we cannot internalize n into a general solution and the only possibility is to verify the solution for the problem for all n . However, from the solution, it will be clear how the pattern of equilibria changes based on n and β .

As in the case of imperfect foresight, we will distinguish “lock-in” and “non-lock-in” equilibria. By a “lock-in” equilibrium we understand the developer’s strategy when he sets the price difference between the upgrade price p_1 and the full price p_2 so high that there is no user switching every two periods (or even more). Consequently, a “non-lock-in” equilibrium is an equilibrium when prices p_1 and p_2 are in such a relation that there are users switching every two periods. As is shown later, the only condition that may be violated after an unconstrained optimization is $\theta_{1,n} \geq \theta_{n,n+1}$, which effectively means that it is optimal for the developer to increase n . The distribution of switching frequencies across users is qualitatively the same as in the imperfect foresight case, see Figures 6 and 7, though the threshold values are different.

“Non-lock-in” equilibria

Necessary conditions for the existence of an interior non-lock-in equilibrium²¹ are $\theta_{2,3}(p_2^*) \leq \theta_{1,2}(p_1^*, p_2^*) \leq \min\{\theta_{1,2}(p_2^*), 1\}$, and since equilibrium prices and the quality jump are dependent on β , the resulting necessary condition will be fully dependent on β too. As we have shown already in (50), the indifferent user between switching every period at p_1 and every two periods at p_2 has quality sensitivity

$$\theta_{1,2}(p_1, p_2) = \frac{p_1}{Q}(1 + \beta) - \frac{p_2}{Q}\beta. \quad (51)$$

²¹Here $\theta_{1,2}(p_1^*, p_2^*)$ is $\theta_{1,n}$ calculated at $n = 2$, i.e., the user who is indifferent between switching every period at p_1 and switching every two periods at p_2 ; whereas, $\theta_{1,2}(p_2^*) = p_2^*/Q^*$ is $\theta_{n-1,n}$ calculated at $n = 2$, i.e., the user who is indifferent between switching every period at the “full” price p_2 and switching every two periods at p_2 .

From equation (39), we know that a user who is indifferent between switching every second and every third period (at p_2) satisfies

$$\theta_{2,3}(p_2) = \frac{p_2}{Q(\beta + 2)}. \quad (52)$$

Then after substituting $\theta_{1,2}$, $\theta_{2,3}$ from (51), (52) into the general profit function (47), simplifying for $n = 2$, and using F.O.C. (see 11.7 for S.O.C. and condition tests), we obtain a profit function similar to the profit function of the non-lock-in case in the imperfect foresight set-up, see (25):

$$\begin{aligned} \Pi = & \left(1 - \frac{p_1}{Q}(1 + \beta) + \frac{p_2}{Q}\beta\right) p_1 + \frac{1}{2} \left(\frac{p_1}{Q}(1 + \beta) - \frac{p_2}{Q}\beta - \frac{p_2}{Q(\beta + 2)}\right) p_2 + \\ & + \frac{p_2}{Q} L_3 p_2 - \bar{B} Q^2. \end{aligned} \quad (53)$$

Using F.O.C. (see Appendix 11.7 for S.O.C. and the necessary conditions), we obtain equilibrium prices p_1^* , p_2^* and quality improvement Q^* :

$$p_1^* = \frac{4\lambda_2^2}{\Lambda_2^2 \bar{B}}, \quad p_2^* = \frac{2(2 + \beta)(1 + 3\beta)\lambda_2}{\Lambda_2^2 \bar{B}}, \quad Q^* = \frac{\lambda_2}{\Lambda_2 \bar{B}}, \quad (54)$$

where

$$\lambda_2 = (1 + \beta)^2 - 2(2 + \beta)L_3, \quad \Lambda_2 = 6 + 11\beta - \beta^3 - 16(1 + \beta)(2 + \beta)L_3,$$

and these values are positive at every $\beta \in (0, 1)$. The only necessary condition that is not guaranteed to hold is $\theta_{2,3}(p_2^*) \leq \theta_{1,2}(p_1^*, p_2^*)$, which is violated at $\beta \geq B_2 \approx 0.325448$. The violation of this condition means that the developer's profit decreases in p_1 in the entire non-lock-in region, whence it is optimal for the developer to switch to 3-lock-in (and often further).

Substituting the equilibrium values back into the profit function as well as (51), (52), we obtain

$$\Pi^* = \frac{\lambda_2^2}{\Lambda_2^2 \bar{B}}, \quad \theta_{1,2}^* = \frac{2(2 + 4\beta - \beta^2 - \beta^3 - 4(1 + \beta)(2 + \beta)L_3)}{\Lambda_2}, \quad \theta_{2,3}^* = \frac{2(1 + 3\beta)}{\Lambda_2},$$

so that the number of high-end users $N_1 = 1 - \theta_{1,2}$ and the per-period average

number of low-end users $N_2 = \frac{1}{2}(\theta_{1,2} - \theta_{2,3}) + \frac{p_2}{Q}L_3$ in equilibrium are the following.

$$N_1^* = \frac{(1 + \beta)(2 + \beta + \beta^2 - 8L_3(2 + \beta))}{\Lambda_2},$$

$$N_2^* = (1 - \beta)\frac{\lambda_2}{\Lambda_2}.$$

Corollary 1. *Equilibrium prices per quality ratios, obtained directly from (54), equal*

$$\frac{p_1^*}{Q^*} = \frac{4\lambda_2}{\Lambda_2}, \quad \frac{p_2^*}{Q^*} = \frac{2(2 + \beta)(1 + 3\beta)}{\Lambda_2}.$$

As β is increasing, the equilibrium number of high-end users is also increasing. Intuitively, the present value of the future utility flow is growing in β , so it becomes more profitable for the consumers to use the upgrade option. This also leads the developer to decrease p_1 with respect to p_2 , thus attracting more consumers to the upgrade version, until β reaches B_2 , when all consumers buying every two periods are crowded out.

“Lock-in” equilibria

In the previous part, we examined non-lock-in equilibria where users switching every two periods were always present on the market. Now, we will generalize the case in a way that users switching every two periods (or even less frequently) are not present on the market. Taking into account the previous calculation, we derive the following general profit function:

$$\Pi = \left(1 - \frac{p_1}{Q}Y + \frac{p_2}{Q}Z\right)p_1 + \frac{1}{n}\left(\frac{p_1}{Q}Y - \frac{p_2}{Q}Z - \frac{p_2}{Q}X\right)p_2 + \frac{p_2}{Q}L_{n+1}p_2 - \bar{B}Q^2, \quad (55)$$

where n is the lowest frequency of users switching at the regular price present on the market. Then the “lock-in” equilibrium is the following (see Appendix 11.7 for S.O.C. and the necessary conditions).

$$p_1^* = \frac{n^2\lambda_n^2}{\Lambda_n^2\bar{B}}, \quad p_2^* = \frac{n^2(Y + nZ)\lambda_n}{2\Lambda_n^2\bar{B}}, \quad Q^* = \frac{n\lambda_n}{2\Lambda_n\bar{B}},$$

where

$$\lambda_n = X + Z - nL_{n+1}, \quad \Lambda_n = 2nY(2X + Z) - Y^2 - n^2Z^2 - 4n^2YL_{n+1},$$

and these values are positive at every n and $\beta \in (0, 1)$. The only necessary condition that is not guaranteed to hold is $\theta_{n,n+1} \leq \theta_{1,n}$, which is violated at $\beta \geq B_n$, where B_n

is increasing in n with $\lim_{n \rightarrow \infty} B_n = \frac{1}{2}$, tabulated in Appendix 11.8. The violation of this condition means that the developer's profit decreases in p_1 in the entire n -lock-in region, so that it is optimal for the developer to switch to $(n + 1)$ -lock-in. This implies that if $\beta \geq \frac{1}{2}$, then it is optimal in our model for the developer to crowd out lower frequencies of switching infinitely, so no price discrimination equilibrium exists. However, if we introduce discounting into the developer's profit and take into account that the consumers start buying only after the quality accumulated reaches their participation constraint (instead of neglecting this due to no discounting, infinite horizon, and hence per-period optimization), then lock-in equilibria would exist for high values of β . The range of β at which price discrimination equilibria exist would also expand if we assume that some consumers have perfect foresight and others have imperfect foresight as described in section 9.2. It should be also noted that high values of β are not very appropriate to software.

Substituting the equilibrium values back into the profit function (55) as well as into the thresholds, we obtain

$$\Pi^* = \frac{n^2 \lambda_n^2}{4\Lambda_n^2 \bar{B}}, \theta_{1,n}^* = \frac{n(Y(2X + Z) - nZ^2 - 2nYL_{n+1})}{\Lambda_n}, \theta_{n,n+1}^* = \frac{nX(Y + nZ)}{\Lambda_n},$$

so that the number of users, who are switching are

$$\begin{aligned} N_1^* &= 1 - \theta_{1,n}^* = \frac{Y(n(2X + Z) - Y - 2n^2L_{n+1})}{\Lambda_n}, \\ N_2^* &= \frac{1}{n} (\theta_{1,n}^* - \theta_{n,n+1}^*) + \frac{p_2^*}{Q^*} L_{n+1} = (Y - nZ) \frac{\lambda_n}{\Lambda_n}. \end{aligned}$$

Comparative statics for price discrimination equilibria

The following results can be derived from the equilibria above. Recall that non-lock-in is mathematically 2-lock-in. We assume that β is valid, i.e., $\beta \leq B_n$ for the given n .

Lemma 10. *If the developer uses n -lock-in price discrimination under perfect foresight, then the average number of switching users is decreasing in n and increasing in β , whereas the equilibrium quality change, both prices, both prices per quality change, and the developer's profit increase in both n and β . In addition, the equilibrium quality change converges for $\beta < \frac{1}{2}$ from below to the socially optimal level $Q_0 = \frac{1}{4B(1-\beta)}$ as n increases.*

7.3 Equilibria comparison under perfect foresight

Here we compare perfect foresight equilibria under a single price and under price discrimination. As under imperfect foresight, the developer's profit is always higher for price discrimination since the developer always has a possibility to set prices $p_1 = p_2$. The following result, which exactly parallels the imperfect foresight outcome, can be shown to hold.

Lemma 11. *In the case of price discrimination, a developer's quality Q and both prices are higher than in the case of a single price, that is $p^* \leq p_1^* \leq p_2^*$, and the price per unit of the quality jump is higher in the case of price discrimination, that is $\frac{p^*}{Q} \leq \frac{p_1^*}{Q} \leq \frac{p_2^*}{Q}$, no matter the β and B . This result is independent from the perfect or imperfect foresight set-up.*

From this result, it follows that the developer's cost of a higher quality improvement in the case of price discrimination is more than fully compensated for by the increased prices.

In comparing the developer profit for a particular β and n , we see that the best strategy for the developer in our model is to set n close to infinity, which is the same result as in the case of imperfect foresight. This result again stems from our model set-up, where the developer is maximizing the average profit per period. Thus, the developer does not differentiate between the profit realized in the first period and the profit from a very distant period "close to infinity." Note that maximizing the average profit instead of the discounted profit across each period is a simplification to obtain an analytical solution, and the introduction of discounting would lead to a finite optimal lock-in depth.

8 Welfare analysis: perfect foresight set-up

A welfare analysis for the perfect foresight set-up is analogous to the imperfect foresight set-up from section 5. All related results from the imperfect foresight remain valid for the perfect foresight at $\beta = 0$. The following result can be shown to hold²².

Proposition 9. *(i) Consumer surplus per period is higher for the single price developer than for the price discriminating developer for all relevant β , increases in β and decreases in n as n -lock-in is used. This is valid for the total consumer surplus as well as for the consumer surplus for high-end and low-end users separately.*

²²These results were derived similar to those presented in Appendices 11.5 and 11.5, but with perfect foresight equilibria and thresholds. As these results are often mathematically cumbersome, we opted not to include them explicitly but rather leave the *Mathematica* file available upon request.

(ii) Under perfect foresight, the equilibrium per-period CS per buyer (total, high-end, and low-end alike) increases in β , which applies to both a single-price monopoly and price discrimination for every n . Per-period CS per buyer for high-end users is higher for the single price developer than for the price discriminating developer for all relevant β and decreases in n . Per-period CS per buyer for low-end users is lower for the single price developer than for the price discriminating developer for all relevant β except for $\beta < \approx 0.097773$ at $n = 2$ and $\beta < \approx 0.004777$ at $n = 3$, and increases in n .

(iii) Under perfect foresight, total per-period CS per buyer is lower for the single price developer than for the price discriminating developer whenever $n \geq 8$ or $\beta > \approx 0.289509$. It increases in n whenever $n \geq 8$ or $\beta < \approx 0.099358$.

Part (i) exactly corresponds to the one from the imperfect foresight case, and part (ii) is similar to what happens under imperfect foresight (recall that under imperfect foresight, CS per buyer for high-end users follows the same pattern as CS itself; whereas, the total CS per buyer and CS per buyer for low-end users increase in n and are higher under price discrimination except for $n = 2$ and $n = 3$). Thus, part (ii) confirms that the different relation between a single-price and price discrimination CS per buyer for low-end users for low n is just a mathematical property.

Part (iii) shows that as for the total CS per buyer, the effect for low-end users eventually prevails (mainly because the CS for high-end users decreases more quickly than the CS for low-end users). (See Appendix 11.8 for the behavior of the CS per buyer when n is small.) As it is optimal for the developer to “squeeze” non-updating consumers who switch relatively frequently, the CS per buyer eventually increases in n .

As for social welfare, according to Lemmas 7.2 and 7.3, the equilibrium quality is higher under price discrimination and converges from below to the socially optimal level as n increases. Therefore, the same properties apply to the equilibrium social welfare (for $\beta < \frac{1}{2}$, when price discrimination equilibria exist).

9 The comparison between imperfect and perfect foresight

9.1 The comparison of equilibria

As we have already noted, the outcome of the model at $\beta = 0$ is identical under both perfect and imperfect foresight. While the results at $\beta > 0$ are different as

described below, there is one notable exception. Namely, the average number of users who switch to a new version every period is $\frac{1}{2}$ no matter whether they have perfect or imperfect foresight. Therefore, the number of switching users per period under a single price is independent from the discount factor β , and costs \bar{B} .

Before comparing other results, recall that while perfect foresight results are expressed in terms of the developer's quality cost efficiency \bar{B} and the quality choice Q , imperfect foresight results were, for convenience, expressed in terms of $B = \bar{B}(1 - \beta)^2$ and $q = \frac{Q}{1 - \beta}$, so the corresponding change has been made. The following result can be proved.

Proposition 10. *Assume that $\beta > 0$, the price setting is fixed at either a single price or price discrimination by upgrades, and that n is fixed with $\beta \leq B_n$ in the discrimination case. Then the following results hold.*

(i) *The equilibrium prices, qualities, price-to-quality ratios, and per period profits are lower under perfect foresight than under imperfect foresight, and the average per-period number of switching users is higher under perfect foresight than under imperfect foresight.*

(ii) *Under price discrimination, the equilibrium per-period consumer surplus, whether total or per buyer, is higher under perfect foresight than under imperfect foresight, counted for all users or for high-end and low-end consumers separately. However under the single price, the CS is higher under perfect foresight if β is not too high ($\beta < \frac{1}{2}$ is sufficient in all cases) but becomes higher under imperfect foresight as $\beta \rightarrow 1$.*

(iii) *The social welfare is lower under perfect foresight than under imperfect foresight.*

A useful insight into part (i) is provided by Lemmas 6.1 and 7.1, where we show that the indifferent users have a positive utility flow under perfect foresight, and that flow is zero under imperfect foresight. In other words, the developer cannot exert as much monopoly power against users with perfect foresight as against users with imperfect foresight: hence, the result above. This is especially evident as $\beta \rightarrow 1$ when the imperfect foresight values (other than number of users) tend to $+\infty$; whereas, the perfect foresight values, while also increasing in β , have finite limits.

For part (ii)—See Appendix 11.8 for the thresholds under a single-price monopoly—the lower degree of monopoly power in equilibrium also results in a higher consumer surplus under perfect foresight when β is relatively low, including all values of β such that a price discrimination equilibrium under perfect foresight exists. However, at high values of β (which are improbable as noted in the discussion of price discrimi-

nation equilibria under perfect foresight), when the imperfect foresight equilibrium values tend to infinity, CS becomes higher under imperfect foresight.

Part (iii), while following from the fact that equilibrium qualities are lower under perfect foresight, might be surprising as a weaker monopoly is usually associated with higher welfare. However, in our model, while the developer cannot extract as much CS from the users as under imperfect foresight, neither are the users able to capture the entire difference, which results in a loss in welfare. Note that as it is optimal for the price discriminating developer to “squeeze” infinitely, the welfare will tend to its optimal value in both cases.

9.2 The generalization of results

We have shown that the comparative static results are qualitatively independent from using a perfect foresight or an imperfect foresight set-up. Thus, we can generalize the comparative static results on the market set-up where we have mixed users with imperfect and perfect foresight. If the type of foresight of the user is statistically independent from the user’s quality sensitivity, with the latter uniformly distributed on $\langle 0, 1 \rangle$, then all qualitative results of this paper remain valid.

10 Conclusion

In this paper, we have analyzed monopoly price discrimination based on upgrades designed for users who switch every period. It was done in quite a general set-up (infinitely durable products, an infinite number of users with different sensitivity to quality), and our restrictions on the model set-up were only based on the empirically observed patterns of the developers’ behavior. We assumed the prices to be the same in every period since in real markets, we often observe prices for new versions of a product set at almost the same level as for previous versions. Our second restriction was that the quality improvement from one period to another was fixed.

As for our main result, we showed that price discrimination does not lead to lower prices for any user group in the market. This is caused by the character of price discrimination in our model, when the target group of price discounts is not the most price sensitive group of the users, but the discount is provided to the users with the highest frequency of switching, i.e., to the users with the highest sensitivity to quality.

The second key result is that a price discriminating developer accelerates software development more than a single-price developer, which could be perceived as beneficial for users, but on the other hand, the number of switching users in the case

of a price discriminating developer is always lower. Even if we split users into two groups, those who switch every period and those who switch less frequently, we see that in the case of a price discriminating developer, the number of switching users is lower in both groups.

We showed that all those comparative static results are valid no matter whether users have perfect or imperfect foresight.

We can also look at the results from a different perspective: a price discriminating developer can more effectively “separate” high-end users from low-end users and charge them a higher price by offering a higher software quality while at the same time being aware that low-end users become less important from a revenue point of view. Since the developer knows that low-end users would switch to a new version from time to time anyway, the developer sets the price for them relatively higher than in the case of a single price, which ensures for the developer that those users who may consider switching to a new version every second period at the full price would rather switch every period for the lower upgrade price. This effect is more visible for perfect foresight users.

As for social welfare, a price discriminating developer generates (under plausible assumptions) higher per-period consumer surplus per user, but at the same time the number of switching users is lower, and the overall consumer surplus is lower as well. While this also indicates that the developer’s monopoly power strengthens, the social welfare increases as well and approaches the socially optimal level in the limiting case.

The model used in the article was designed for the software market, but it can be easily applied more widely whenever users are heterogeneous and a resale market is outlawed. Infinite durability of the product is not necessary, but the absence of product depreciation eliminates demand generated by product replacement due to physical obsolescence, which simplifies the analytical solution.

11 Appendix

Most of the calculations in this paper were performed using *Mathematica* and other similar software. The *Mathematica* file is available upon request.

11.1 The approach used for S.O.C. verification

In all cases, the objective function is the developer’s profit, which can be either a function of a single price and a quality, $\Pi(p, Q)$, or a function of two prices and a quality, $\Pi(p_1, p_2, Q)$, where Q is replaced with q under imperfect foresight. In the

single-price case, the form of the Hessian used in this paper is

$$H = \begin{pmatrix} \Pi_{pp} & \Pi_{pQ} \\ \Pi_{pQ} & \Pi_{QQ} \end{pmatrix}, \quad (56)$$

and in the two-price case we use

$$H = \begin{pmatrix} \Pi_{11} & \Pi_{12} & \Pi_{1Q} \\ \Pi_{12} & \Pi_{22} & \Pi_{2Q} \\ \Pi_{1Q} & \Pi_{2Q} & \Pi_{QQ} \end{pmatrix}, \quad (57)$$

where subscripts ‘1’ and ‘2’ stand for the derivatives with respect to p_1 and p_2 . In the proofs below, we immediately proceed to listing the principal minors of the Hessians.

11.2 The approach used for consumer surplus calculation

A recurrent task in this paper is to calculate the average per period consumer surplus (CS) as an infinite sum of CS of the consumers who switch every n or more periods, $n \geq 2$. Recall that the structure of the consumers’ utility is $\theta Q - p$, so that the structure of added consumer utility is $\theta q - p$, where $q = \frac{Q}{1-\beta}$, and the range of the consumers who switch every n periods is given by $\theta_{n,n+1} < \theta < \theta_{n-1,n}$, where $\theta_{n,n+1}$ strictly decreases in n and $\lim_{n \rightarrow \infty} \theta_{n,n+1} = 0$. In addition, the usual form of $\theta_{n,n+1}$ is

$$\theta_{n,n+1} = \frac{p}{Q} X_n, \quad \lim_{n \rightarrow \infty} X_n = 0$$

(here p can be p_2 , and Q can be q for imperfect foresight, but see below). Then the average demand per period from the group in question equals

$$\sum_{m=n}^{\infty} \frac{1}{m} (\theta_{m-1,m} - \theta_{m,m+1}) = \frac{p}{Q} \sum_{m=n}^{\infty} \frac{1}{m} (X_{m-1} - X_m) = \frac{p}{Q} L_n.$$

The value L_n depends on the consumers’ discount factor β , and at $\beta = 0$, it turns into the imperfect foresight D_n . The subscript n is usually omitted when $n = 2$.

CS for the group in question is given by $CS_{n+} = \sum_{m=n}^{\infty} CS_m$, where

$$CS_m = \int_{\theta_{m,m+1}}^{\theta_{m-1,m}} \left(\theta q - \frac{p}{m} \right) d\theta = \int_{(p/Q)X_m}^{(p/Q)X_{m-1}} \left(\theta q - \frac{p}{m} \right) d\theta = \frac{p^2}{Q} \left(\frac{X_{m-1}^2 - X_m^2}{2(1-\beta)} - \frac{X_{m-1} - X_m}{m} \right).$$

Note that the infinite sum of the “subtracted” term inside the parentheses is L_n , and as $\lim_{m \rightarrow \infty} X_m = 0$,

$$\sum_{m=n}^{\infty} (X_{m-1}^2 - X_m^2) = \lim_{m \rightarrow \infty} (X_{n-1}^2 - X_m^2) = X_{n-1}^2,$$

so that the final expression for CS is

$$CS_{n+} = \sum_{m=n}^{\infty} CS_m = \frac{p^2}{Q} \left(\frac{X_{n-1}^2}{2(1-\beta)} - L_n \right).$$

Two particularly important cases are $n = 2$ (recall that $X_1 = 1$ for all β), and $n = n + 1$ (when the summation starts at $n + 1$). Then

$$CS_{2+} = \frac{p^2}{Q} \left(\frac{1}{2(1-\beta)} - L \right), \quad CS_{(n+1)+} = \frac{p^2}{Q} \left(\frac{X_n^2}{2(1-\beta)} - L_{n+1} \right).$$

For imperfect foresight the outcome is (note that here $X_n = 1/n$)

$$CS_{2+} = \frac{p^2}{q} \left(\frac{1}{2} - D \right), \quad CS_{(n+1)+} = \frac{p^2}{q} \left(\frac{1}{2n^2} - D_{n+1} \right).$$

11.3 The single price model for imperfect foresight users

S.O.C. verification

The profit function is

$$\Pi = \left(1 - \frac{p}{q} \right) p + \frac{p}{q} D p - B q^2,$$

and the principal minors of the Hessian are $\frac{2}{q}(D-1)$ and $\frac{4B}{q}(1-D)$, so that H is negative definite as $q > 0$, $B > 0$, and $D \approx 0.355$. Therefore, the solution to F.O.C. is a maximum.

11.4 Imperfect foresight users and price discrimination

S.O.C. and validity in non-lock-in

We have to maximize the profit

$$\Pi = \left(1 - \frac{p_1}{q} \right) \cdot p_1 + \frac{p_2}{q} D \cdot p_2 - B \cdot q^2 - \left(\frac{p_2}{q} - \frac{p_1}{q} \right) \cdot \frac{1}{2} \cdot p_2$$

with respect to the conditions $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$ and $\frac{p_1}{q} \leq 1$. Our approach is to start with unconstrained optimization and check the conditions afterwards. (It happens

that the conditions are satisfied, so there is no need to re-calculate.)

F.O.C. result in

$$p_1^* = 4 \frac{(1-2D)^2}{(16D-7)^2 B}, p_2^* = 2 \frac{1-2D}{(16D-7)^2 B}, q_e^* = \frac{2D-1}{B(16D-7)},$$

so that $\frac{p_1^*}{p_2^*} = 2 - 4D$, and $1 - \frac{p_1}{q} = \frac{3-8D}{7-16D}$, which satisfies our assumptions that $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$ and $\frac{p_1}{q} \leq 1$. The principal minors of the Hessian equal $-\frac{2}{q}$, $\frac{7-16D}{4q^2}$, and $-\frac{B(7-16D)}{2q^2}$, so that H is negative definite as $q > 0$, $B > 0$, and $D \approx 0.355$. Therefore, the solution to F.O.C. is a maximum.

S.O.C. and validity in lock-in

The profit function in n -lock-in is

$$\Pi = \frac{p_2}{nq_e} (p_1 + p_2 - np_2\psi_1(n)) + \left(1 - \frac{p_1}{q_e}\right) p_1 - Bq_e^2.$$

As in the non-lock-in case, there are the validity conditions $p_1 \in \langle \frac{p_2}{n}, \frac{p_2}{n-1} \rangle$ and $\frac{p_1}{q} \leq 1$, which are assumed to hold before being checked.

F.O.C. result in

$$p_1^* = \frac{n^2}{B} \frac{(n\psi_1(n) - 1)^2}{(4n^2\psi_1(n) - 4n - 1)^2}, p_2^* = \frac{n^2}{2B} \frac{n\psi_1(n) - 1}{(4n^2\psi_1(n) - 4n - 1)^2},$$

$$q_e^* = \frac{n}{2B} \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1},$$

and the validity conditions can be shown to hold using the properties of the polygamma function.

The Hessian is calculated as in (57), and the principal minors equal

$$-\frac{2}{q}, \frac{4n^2\psi_1(n) - 4n - 1}{n^2q^2}, -\frac{2B(4n^2\psi_1(n) - 4n - 1)}{n^2q^2},$$

so that H is negative definite as $q > 0$, $B > 0$, and it can be shown that $4n^2\psi_1(n) - 4n - 1 > 0, \forall n \geq 2$. Therefore, the solution to F.O.C. is a maximum.

The number of users under imperfect foresight and price discrimination

The number of upgrading users in equilibrium:

$$N_H^* = 1 - \frac{p_1^*}{q_e^*} = \frac{1}{2} \left(1 - \frac{1}{4n^2\psi_1(n) - 4n - 1}\right). \quad (58)$$

The number of users who buy less frequently than every period is:

$$N_L^* = \frac{p_2^*}{q_e^*} D_{n+1} - \frac{1}{n} \left(\frac{p_2^*}{nq_e^*} - \frac{p_1^*}{q_e^*} \right) = \frac{n\psi_1(n) - 1}{4n^2\psi_1(n) - 4n - 1}. \quad (59)$$

Summing N_H^* and N_L^* , we obtain the average demand per period N^* .

11.5 Imperfect foresight users: comparison and welfare analysis

Single price versus price discrimination

A comparison using the single-price equilibrium quality q^* by the price discriminating developer In the equilibrium comparison in the main part, we see that prices and quality are higher in the case of a price discriminating developer. Assume now that the developer has already set the quality at a level of the single price equilibrium $q^* = \frac{1}{8(1-D)B}$, derived in (18), and now is allowed to price discriminate. We now compare how these prices, denoted as p_1^{cs} and p_2^{cs} , differ from the single-price equilibrium value p^* . This comparison will allow us to separate the pure effect of enabling price discrimination and the effect of higher quality in the case of price discrimination.

Proposition 11. *Optimal prices for discrimination for a developer who has already set $q = q^*$ are*

$$p_1^{cs} = \frac{1}{2B} \frac{1 - 2D}{(7 - 16D)(1 - D)}, \quad p_2^{cs} = \frac{1}{4B} \frac{1}{(7 - 16D)(1 - D)}.$$

Proof. If the quality change q is fixed in the non-lock-in price discrimination problem, F.O.C. in prices result in

$$p_1 = \frac{4(1 - 2D)}{7 - 16D}q, \quad p_2 = \frac{2}{7 - 16D}q,$$

the ratio of the prices is $\frac{p_1}{p_2} = 2 - 4D$, so that the non-lock-in condition $p_1 \in \langle \frac{p_2}{2}, p_2 \rangle$ is satisfied, and the principal minors of the Hessian equal $-\frac{2}{q}$ and $\frac{7-16D}{4q^2}$ so that the solution is a maximum. Substituting $q = q^*$ yields the result. ■

By comparing with the optimal single price $p^* = \frac{1}{16(1-D)^2B}$, we immediately see that $\frac{p_1^{cs}}{p^*} \simeq 1.133911$, so that p_1^{cs} is higher than p^* , and since p_2^{cs} is higher than p_1^{cs} , both prices are higher in the case of price discrimination.

Corollary 2. *Enabling price discrimination while fixing q^* from the single-price equilibrium decreases the average number of users who switch to the new product,*

raises the price for both the upgrade and full versions, and increases the price per quality value, $\frac{p^*}{q^*} < \frac{p_1^{sc}}{q^*} < \frac{p_2^{sc}}{q^*}$.

Proof. The result is proven by direct comparison. ■

Thus, the key result from the comparison of the single-price and price discrimination equilibria comes from two effects: enabling price discrimination and increasing equilibrium quality. Further, these two effects reinforce each other as both result in higher prices and a lower number of users.

Welfare analysis: single price developer

Recall that the equilibrium price and quality improvement are

$$p^* = \frac{1}{16(1-D)^2 B}, \quad q^* = \frac{1}{8(1-D)B} \implies \frac{p^*}{q^*} = \frac{1}{2(1-D)}.$$

The asterisk superscript denoting the equilibrium values is implied where needed.

Consumer surplus for high-end users equals

$$CS_H = \int_{\frac{p}{q}}^1 (\theta q - p) d\theta = \frac{1}{2q} (p - q)^2 = \frac{1}{64} \frac{(1-2D)^2}{(1-D)^3 B} \approx \frac{0.0048941}{B}. \quad (60)$$

Consumer surplus per buyer for high-end users is then

$$\frac{CS_H}{N_H} = \frac{CS_H}{1 - \frac{p}{q}} = \frac{1}{2} (q - p) = \frac{1}{32} \frac{1-2D}{(1-D)^2 B} \approx \frac{0.021778}{B}. \quad (61)$$

Consumer surplus for low-end users equals

$$CS_L = \frac{p^2}{q} \left(\frac{1}{2} - D \right) = \frac{1}{64} \frac{1-2D}{(1-D)^3 B} \approx \frac{0.016884}{B}. \quad (62)$$

Consumer surplus per buyer for low-end users is then

$$\frac{CS_L}{N_L} = \frac{CS_L}{\frac{p}{q} D} = p \frac{(1-2D)}{2D} = \frac{1}{32} \frac{(1-2D)}{(1-D)^2 DB} \approx \frac{0.061335}{B}. \quad (63)$$

Total consumer surplus equals

$$CS = CS_H + CS_L = \frac{1}{32} \frac{1-2D}{(1-D)^2 B} \approx \frac{0.021778}{B}. \quad (64)$$

Consumer surplus per buyer is then

$$\frac{CS}{N} = \frac{1}{16} \frac{1 - 2D}{(1 - D)^2 B} \approx \frac{0.043556}{B}. \quad (65)$$

Finally, social welfare under imperfect foresight with a single-price developer equals

$$W = \frac{1}{2}q - Bq^2 = \frac{3 - 4D}{64B(1 - D)^2} = \frac{3}{16} \frac{2\pi^2 - 15}{B(\pi^2 - 6)^2} \approx \frac{0.059344}{B}. \quad (66)$$

A welfare analysis: price discrimination

Here the general n -lock-in case is analyzed. Recall that non-lock-in is mathematically 2-lock-in. Recall that the equilibrium prices and quality change are

$$p_1^* = \frac{n^2 (n\psi_1(n) - 1)^2}{B \gamma_4^2}, \quad p_2^* = \frac{n^2 n\psi_1(n) - 1}{2B \gamma_4^2}, \quad q_e^* = \frac{n n\psi_1(n) - 1}{2B \gamma_4},$$

where $\gamma_m = mn^2\psi_1(n) - mn - 1$, note that $\gamma_m > 0$ for $n \geq 2$, and $m = 2, 3, 4$. The asterisk subscript for the equilibrium values is implied where needed.

Consumer surplus for high-end users equals

$$CS_H = \int_{\frac{p_1}{q_e}}^1 (\theta q_e - p_1) d\theta = \frac{1}{2q_e} (p_1 - q_e)^2 = \frac{n (n\psi_1(n) - 1) \gamma_2^2}{4B \gamma_4^3}. \quad (67)$$

Consumer surplus per buyer for high-end users is then

$$\frac{CS_H}{N_H} = \frac{1 - \gamma_4^{-2}}{32B}. \quad (68)$$

Consumer surplus for low-end users consists of the surplus for those who buy with the frequency of exactly n periods ($\frac{p_2}{nq_e} = \theta_{n,n+1} < \theta < \theta_{1,n} = \frac{p_1}{q_e}$) and for those who buy even less frequently. The former term equals

$$\int_{\theta_{n,n+1}}^{\theta_{1,n}} \theta q_e - \frac{p_2}{n} d\theta = \frac{1}{2} \frac{(np_1 - p_2)^2}{n^2 q_e},$$

and the latter equals

$$\frac{p_2^2}{q_e} \left(\frac{1}{2n^2} - D_{n+1} \right).$$

Substituting the equilibrium values yields

$$CS_L = \frac{n^2 (n\psi_1(n) - 1)^2 \gamma_2}{2B \gamma_4^3}. \quad (69)$$

Consumer surplus per buyer for low-end users is then

$$\frac{CS_L}{N_L} = \frac{n (1 - \gamma_4^{-2})}{16B}. \quad (70)$$

Total consumer surplus equals

$$CS = CS_H + CS_L = \frac{1 - \gamma_4^{-2}}{32B},$$

and total consumer surplus per buyer is CS/N^{D*} , where

$$N^{D*} = \frac{n(1 + 2n)\psi_1(n) - 2(1 + n)}{4n^2\psi_1(n) - 4n - 1}.$$

Finally, social welfare under imperfect foresight when the developer uses price discrimination with n -lock-in equals

$$W = \frac{1}{2}q_e - Bq_e^2 = \frac{n (n\psi_1(n) - 1) \gamma_3}{4B \gamma_4^2}.$$

11.6 Perfect foresight: single price developer

The user indifferent between switching every n and every $n + 1$ periods

Assume that there is a single price p , the user purchased the product in the current period and now decides on whether to switch every n or every $n + 1$ periods. As the user will buy the product $n(n + 1)$ periods from the current one in both cases, the decision is based on (the NPV of) the utility added between the current period and the period $n(n + 1) - 1$ from now. Note that if a user buys a version T periods from now and keeps it for S periods, then the utility added over the periods from T to $T + S - 1$ equals

$$U = \beta^T \left(\theta T Q \frac{1 - \beta^S}{1 - \beta} - p \right).$$

The user switching every $n + 1$ periods buys in periods $n + 1, 2(n + 1), \dots, (n - 1)(n + 1)$, and keeps every version purchased for $n + 1$ periods, so that the

utility added equals

$$U_{n+1}(\theta) = \theta \sum_{i=1}^{n-1} \beta^{i(n+1)} (i(n+1)) Q \frac{1 - \beta^{n+1}}{1 - \beta} - \sum_{i=1}^{n-1} \beta^{i(n+1)} p.$$

The user switching every n periods buys in periods $n, 2n, \dots, n \cdot n$, and keeps every version purchased for n periods, so that the utility added equals

$$U_n(\theta) = \theta \sum_{i=1}^n \beta^{i \cdot n} (i \cdot n) Q \frac{1 - \beta^n}{1 - \beta} - \sum_{i=1}^n \beta^{i \cdot n} p.$$

From these two, we can derive the indifferent user $\theta_{n,n+1}$ after algebraic transformations:

$$\theta_{n,n+1} = \frac{p}{Q} \frac{(1 - \beta)^2}{n(1 - \beta) - \beta(1 - \beta^n)}.$$

As for an alternative decision process, when the user has the version which is n periods old and chooses between buying now and then every n periods and buying in the next period and then every $n + 1$ periods, the outcome is the same as the utilities to compare, $\beta^{-n}U_n(\theta)$ and $\beta^{-n}U_{n+1}(\theta)$ respectively.

Note that at $\beta = 1$, the term $\frac{1 - \beta^S}{1 - \beta}$ is to be replaced with S , and then $\theta_{n,n+1} = \frac{p}{Q} \frac{2}{n+n^2}$. Also note that the threshold can be expressed as

$$\theta_{n,n+1} = \frac{p}{Q} \frac{1}{\sum_{m=1}^n m \beta^{n-m}},$$

so that $\theta_{n,n+1}$ decreases in β .

Recall that the corresponding threshold for imperfect foresight is $\frac{p}{nq} = \frac{p}{Q} \frac{1 - \beta}{n}$, and the ratio of the two equals

$$\theta_{n,n+1} \left(\frac{p}{Q} \frac{1 - \beta}{n} \right)^{-1} = \frac{n(1 - \beta)}{n(1 - \beta) - \beta(1 - \beta^n)} \geq 1$$

(equality holds only at $\beta = 0$), so that the perfect foresight threshold is not lower than the imperfect foresight one. Therefore, the utility received by a user indifferent between switching every n and every $n + 1$ periods, which is zero in the imperfect foresight case, is non-negative, and positive if $\beta > 0$, under perfect foresight.

11.7 Perfect foresight: discrimination by upgrades

The user indifferent between switching every period at the upgrade price and every n periods at the full price

Assume that the user purchased the product in the previous period, so two options are considered. First, the user can exercise the right to buy at the upgrade price p_1 , and if this is optimal, then the same decision will be made in all subsequent periods. Second, the user may decide to wait thus losing his eligibility for the upgrade price, so that the user would wait $n - 1$ periods and then switch every n periods at the full price p_2 , where n is determined as in Appendix 11.6. The easiest way to derive the threshold is to compare infinite discounted added utility flows from these two options. Note that if a user buys at price p a version T periods older than the previously possessed, then the infinite added utility flow at the moment of purchase equals

$$U = \theta Tq - p = \frac{\theta TQ}{1 - \beta} - p.$$

The user switching every period from the current one inclusive at price p_1 has an added utility flow of

$$U_1(\theta, p_1) = \left(\frac{\theta Q}{1 - \beta} - p_1 \right) (1 + \beta + \beta^2 + \dots) = \frac{\theta Q}{(1 - \beta)^2} - \frac{p_1}{1 - \beta}.$$

The user switching $n - 1$ periods from the current one and then every n periods, always at p_2 , has an added utility flow of

$$\begin{aligned} U_n(\theta, p_2) &= \left(\frac{\theta n Q}{1 - \beta} - p_2 \right) (\beta^{n-1} + \beta^{2n-1} + \beta^{3n-1} + \dots) = \\ &= \beta^{n-1} \left(\frac{\theta n Q}{(1 - \beta)(1 - \beta^n)} - \frac{p_2}{1 - \beta^n} \right). \end{aligned}$$

From these two, we can derive the indifferent user $\theta_{1,n}$ after algebraic transformations:

$$\theta_{1,n} = (1 - \beta) \frac{p_1(1 - \beta^n) - p_2(1 - \beta)\beta^{n-1}}{(1 - \beta^n - n\beta^{n-1}(1 - \beta))Q}.$$

Note that as $\beta \rightarrow 0$, $\theta_{1,n} \rightarrow \frac{p_1}{Q}$, which is the imperfect foresight value of this threshold, and as $\beta \rightarrow 1$, $\theta_{1,n} \rightarrow \frac{2(np_1 - p_2)}{n(n-1)Q}$.

If n is optimal at price p_2 , then $\theta_{1,n}nq - p_2 \geq 0$ as is shown in Appendix 11.6, so that $\theta_{1,n}q - p_1 \geq 0$.

S.O.C. and validity in non-lock-in

The profit function is

$$\begin{aligned} \Pi = & \left(1 - \frac{p_1}{Q}(1 + \beta) + \frac{p_2}{Q}\beta\right) p_1 + \frac{1}{2} \left(\frac{p_1}{Q}(1 + \beta) - \frac{p_2}{Q}\beta - \frac{p_2}{Q(\beta + 2)}\right) p_2 + \\ & + \frac{p_2}{Q} L_3 p_2 - \bar{B} Q^2, \end{aligned}$$

and the validity conditions are $\theta_{1,2}(p_2) \geq \theta_{1,2}(p_1, p_2) \geq \theta_{2,3}(p_2)$ and $1 \geq \theta_{1,2}(p_1, p_2)$, which are assumed to hold before being checked. F.O.C. result in

$$p_1^* = \frac{4\lambda_2^2}{\Lambda_2^2 \bar{B}}, p_2^* = \frac{2(2 + \beta)(1 + 3\beta)\lambda_2}{\Lambda_2^2 \bar{B}}, Q^* = \frac{\lambda_2}{\Lambda_2 \bar{B}},$$

where

$$\lambda_2 = (1 + \beta)^2 - 2(2 + \beta)L_3, \Lambda_2 = 6 + 11\beta - \beta^3 - 16(1 + \beta)(2 + \beta)L_3,$$

so that the conditions $\theta_{1,2}(p_2) \geq \theta_{1,2}(p_1, p_2)$ and $1 \geq \theta_{1,2}(p_1, p_2)$ hold for every β , and the remaining condition $\theta_{1,2}(p_1, p_2) \geq \theta_{2,3}(p_2)$ holds for $\beta \leq B_2 \approx 0.325448$.

The minors of the Hessian equal

$$-\frac{2(1 + \beta)}{Q}, \frac{\Lambda_2}{4Q^2(2 + \beta)}, -\frac{\bar{B}\Lambda_2}{2Q^2(2 + \beta)},$$

so that the solution is a maximum.

S.O.C. and validity in lock-in

The profit function is

$$\Pi = \left(1 - \frac{p_1}{Q}Y + \frac{p_2}{Q}Z\right) p_1 + \frac{1}{n} \left(\frac{p_1}{Q}Y - \frac{p_2}{Q}Z - \frac{p_2}{Q}X\right) p_2 + \frac{p_2}{Q} L_{n+1} p_2 - \bar{B} Q^2,$$

and the validity conditions are $\theta_{n-1,n} \geq \theta_{1,n} \geq \theta_{n,n+1}$ and $1 \geq \theta_{1,n}$, which are assumed to hold before being checked. F.O.C. result in

$$p_1^* = \frac{n^2 \lambda_n^2}{\Lambda_n^2 \bar{B}}, p_2^* = \frac{n^2(Y + nZ)\lambda_n}{2\Lambda_n^2 \bar{B}}, Q^* = \frac{n\lambda_n}{2\Lambda_n \bar{B}},$$

where

$$\lambda_n = X + Z - nL_{n+1}, \Lambda_n = 2nY(2X + Z) - Y^2 - n^2Z^2 - 4n^2YL_{n+1},$$

so that the conditions $\theta_{n-1,n} \geq \theta_{1,n}$ and $1 \geq \theta_{1,n}$ hold for every β , and the remaining condition $\theta_{1,n} \geq \theta_{n,n+1}$ holds for $\beta \leq B_n$, where B_n is tabulated in Appendix 11.8.

The minors of the Hessian equal

$$-\frac{2Y}{Q}, \frac{\Lambda_n}{n^2Q^2}, -\frac{2\bar{B}\Lambda_n}{n^2Q^2},$$

so that the solution is a maximum.

11.8 Numeric simulations

Functions $L(\beta)$ and B_n

The calculations were performed in *Mathematica*, where $L(\beta)$ was interpolated with step 10^{-3} .

β	$L(\beta)$	n	B_n
0	0.355066	2	0.325448
0.01	0.355942	3	0.377935
0.05	0.359394	4	0.415119
0.1	0.363599	5	0.441850
0.2	0.371653	6	0.461048
0.3	0.379247	7	0.474634
B_2	0.381107	8	0.483997
0.4	0.386393	9	0.490227
0.5	0.393105	10	0.494211
0.6	0.399388	11	0.496661
0.7	0.405247	12	0.498115
0.8	0.410682	13	0.498954
0.9	0.415691	14	0.499427
0.95	0.418033	16	0.499833
0.99	0.419827	18	0.499953
1	0.420264	20	0.499989

As for B_n , it increases in n , and its limit can be shown to be 0.5.

CS per buyer under perfect foresight for small n

In the following table, we report the ranges of β in which the behavior of the total price discrimination equilibrium CS per buyer is different from the general case,

i.e., it is either lower than under single-price monopoly or decreases in n . No such behavior occurs for $n \geq 8$.

n	$CS_{PD} < CS_{SP}$	$CS_n > CS_{n+1}$
2	[0, 0.220619)	(0.099358, B_2]
3	[0, 0.270516)	(0.211142, B_3]
4	(0.010498, 0.289509)	(0.303669, B_4]
5	(0.045734, 0.285588)	(0.379895, B_5]
6	(0.078875, 0.262546)	(0.442822, B_6]
7	(0.117158, 0.222414)	–

CS comparison under a single-price monopoly

In the following table, we report the maximal values of β until which CS is higher under perfect foresight than under imperfect foresight when the developer charges the same price to all consumers. Note that CS is higher under perfect foresight in all cases when $0 < \beta < \frac{1}{2}$. The threshold for all users is the same because the number of switching users is the same ($N^* = \frac{1}{2}$) in both cases.

	All users	High-end users	Low-end users
CS	0.733526	0.797919	0.699392
CS per buyer	0.733526	0.875735	0.569339

References

- Anton, J., and Biglaiser, G. (2010): “Quality, Upgrades and Equilibrium in a Dynamic Monopoly Market ” Economic Research Initiatives at Duke (ERID) Working Paper No. 37
- Banerjee, D.S. (2003): “Software Piracy: A Strategic Analysis and Policy Instruments,” *International Journal of Industrial Organization*, Vol. 21, pp. 97–127.
- Belleflamme, P., and Peitz, M. (2010): *Industrial Organization: Market Strategies*, Cambridge University Press.
- Belleflamme, P., and Peitz, M. (2012). Digital Piracy: Theory, in: M. Peitz and J. Waldfogel (eds.), *The Oxford Handbook of the Digital Economy*, Oxford University Press, pp. 489-530.
- Echevarría, R.C. (2005): “Technological and Physical Obsolescence and the Timing of Adoption,” Fundación BVVA, Working Paper.
- Ellison, G., and Fudenberg, D. (2000): “The Neo-Luddite’s Lament: Excessive Upgrades in the Software Industry,” *RAND Journal of Economics*, Vol. 31, pp. 253–272.
- Fishman, A., and Rob, R. (2000): “Product Innovation by a Durable-good Monopoly,” *RAND Journal of Economics*, Vol. 31, pp. 237–252.
- Fudenberg, D., and Tirole, J. (2000): “Pricing a Network Good to Deter Entry,” *Journal of Industrial Economics*, Vol. 48, pp.373–390.
- Fudenberg, D., and Tirole, J. (1998): “Upgrades, Trade-ins, and Buybacks,” *RAND Journal of Economics*, Vol. 29, pp. 235–258.
- Fudenberg, D., and Villas-Boas, J. M. (2006): “Behavior-based Price Discrimination and Customer Recognition,” in Hendershott, T.J., ed. *Handbook on Economics and Information Systems*, Elsevier.
- Inderst, R. (2008): “Durable Goods with Quality Differentiation,” *Economic Letters*, Vol. 100, pp. 173–177.
- Katz, L., and Shapiro, C. (1998): “Antitrust in Software Markets,” Freedom Foundation Conference.
- Matějka, F. (2010): “Rigid Pricing and Rationally Inattentive Consumer,” CERGE-EI Working Paper Series, No. 409.
- Mehra, A., and Seidmann, A. (2008): “Optimal Timing of Upgrades over a Software Product’s Life Cycle,” The Bradley Policy Research Center Working Paper, No. FR 08-22.
- Nahm, J. (2004): “Durable-goods Monopoly with Endogenous Innovation,” *Journal of Economics & Management Strategy*, Vol. 13, pp. 303–319.

- Sims, C.A. (2003): “Implications of Rational Inattention,” *Journal of Monetary Economics*, Vol. 50(3), pp. 665–690.
- Shapiro, C. (2000): “Competition Policy in the Information Economy,” in Hope, E., ed. *Competition Policy Analysis*, Routledge.
- Stokey, N., (1979): “Intertemporal Price Discrimination,” *Quarterly Journal of Economics*, Vol. 93, pp. 355–371.
- Stokey, N., (1981): “Rational Expectations and Durable Goods Pricing,” *Bell Journal of Economics*, Vol. 12, pp. 112–128.
- Stole, L.A. (2007): “Price Discrimination and Competition,” in Armstrong, M., and Porter, R., eds. *Handbook of Industrial Organization, Volume 3*, Elsevier.
- Wang, Q.H., and Hui, K.L. (2005): “Technology Timing and Pricing In the Presence of an Installed Base,” EconWPA Working Paper, No. 0512013.
- Werden, G.J. (2001): “Network Effects and Conditions of Entry: Lessons from the Microsoft Case,” *Antitrust Law*, Vol. 69, pp. 87–111.
- Internet Sources: www.tomshardware.com, www.idg.com, www.zive.cz.

Working Paper Series
ISSN 1211-3298
Registration No. (Ministry of Culture): E 19443

Individual researchers, as well as the on-line and printed versions of the CERGE-EI Working Papers (including their dissemination) were supported from institutional support RVO 67985998 from Economics Institute of the ASCR, v. v. i.

Specific research support and/or other grants the researchers/publications benefited from are acknowledged at the beginning of the Paper.

(c) Jiří Střelický and Krešimir Žigić, 2013

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical or photocopying, recording, or otherwise without the prior permission of the publisher.

Published by
Charles University in Prague, Center for Economic Research and Graduate Education (CERGE)
and
Economics Institute ASCR, v. v. i. (EI)
CERGE-EI, Politických vězňů 7, 111 21 Prague 1, tel.: +420 224 005 153, Czech Republic.
Printed by CERGE-EI, Prague
Subscription: CERGE-EI homepage: <http://www.cerge-ei.cz>

Phone: + 420 224 005 153
Email: office@cerge-ei.cz
Web: <http://www.cerge-ei.cz>

Editor: Michal Kejak

The paper is available online at http://www.cerge-ei.cz/publications/working_papers/.

ISBN 978-80-7343-282-9 (Univerzita Karlova. Centrum pro ekonomický výzkum a doktorské studium)
ISBN 978-80-7344-274-3 (Národohospodářský ústav AV ČR, v. v. i.)



CERGE-EI
P.O.BOX 882
Politických vězňů 7
111 21 Praha 1
Czech Republic
<http://www.cerge-ei.cz>