

The Economic Value of Rating in App Market

Abstract

This paper investigates the influence of ratings on an emerging software application market, app market. Given the nature of software application as experience good and the general scarcity of other quality signals such as advertising or branding for majority of app developers, consumers' ex ante belief on app utility essentially relies on the app rating which itself is derived from the ex post utility received by peer consumers. We develop an analytical framework which explicitly characterizes this bidirectional rating-utility conversion process, also named as consumer rating behavior, with a new concept "reservation rating." After integrating consumer rating behavior into consumer's utility formation, we derive the market equilibrium and reveal how the changes in consumer rating behavior affect developers and platform owner's profits and how both developers and the platform owner should optimally react to various consumer rating behaviors in terms of the choices of app price, quality level and revenue sharing percentage. We also reveal how changes in consumer rating behavior would impact the social welfare, suggesting consumers' responsibility, in terms of proper rating behavior, to the overall goodness of app community. The rating-dependent utility function further enables us to derive a self-selection mechanism to achieve the separating equilibrium in which high and low cost rate developers choose differentiated revenue sharing percentages.

Keywords: consumer ratings, mobile app, revenue sharing, self-selection

1. Introduction

The traditional third-party application market is now heading towards its next phase. It has been witnessed that a novel third-party application market, mobile app market, is currently experiencing its explosive growth despite the recent economic downturn. Apple App Store, the world's leading mobile app market launched in July 2008, reached total number of 10 billion downloads in January 2011 while it was just 3 billion in January 2010 (*Wall Street Journal* January 2011). Kaufman Brothers LP estimated that over \$1 billion revenue was generated from more than 350,000 applications inside the Apple App Store in 2010 and Citibank expected this revenue figure to be \$2 billion in 2011. Following Apple's move, Google, Microsoft, Research in Motion (RIM) and Amazon opened their own mobile app stores. Gartner Inc. forecasted that the total revenue of entire mobile app market would hit \$15.1 billion in 2011, almost tripled its \$5.2 billion in 2010. It will further increase to \$35 billion in 2014 according to International Data Corp. (IDC)'s projection. Meanwhile, the tremendous success in this novel market is initiating a new trend which may change the whole climate of future third-party application market. Enlightened by the success of Apple App Store, Apple extended the same business model to desktop and laptop applications. On January 6, 2011, Mac App Store was opened with more than 1,000 computer apps out of the gate. In just 20 days, Pixelmator, a small software company, achieved 1 million dollar sale on January 25 from its one single image processing application sold at unit price \$29.99. Traditional giant software companies such as Autodesk also joined the Mac App Store. The market is so enticing that even Microsoft, Apple's major competitor, is considering bringing its Microsoft Office's Mac version onto Mac App Store. "It's something we are looking at," said Amanda Lefebvre, Microsoft's senior marketing manager (*PCWorld* January 2011).

Concerning this emerging economy, the problem of adverse selection caused by information asymmetry is a serious issue which could potentially dismantle the market (Akerlof 1970) due to the following reasons. First, software products belong to experience goods of which consumers can hardly observe the true quality *ex ante* (Shapiro 1985). Second, even if the "objective" true quality is observed, it could imply different utilities for consumers depending on their subjective valuation of quality. The discounted utility is not necessarily known to consumers *ex ante* (Chen and Xie 2008). Third, a characteristic of app market, that is, the mixture of individual and organizational app developers, generates greater belief dispersion on quality than in the situation where vendors are relatively more homogenous. Thus, without the signals to distinguish the quality of apps, consumers are almost clueless about their own *ex post* utility and can hardly figure out their corresponding willingness to pay.

To counter adverse selection, rating, as derived from consumers' *ex post* experience, is a good source for consumers to formulate the correct *ex ante* beliefs on *ex post* net utility and therefore determine their willingness to pay (Li and Hitt 2010, Sun 2010). Though it is not the only source, it is much more

influential than other factors in app market. A large amount of ratings are aggregated and available at one single site, for instance, iTunes for iPhone apps, and hence easy to be found. Consumers are allowed to rate at that site only after they purchase an app. This helps, to a large extent, to avoid shilling behavior and hence maintain the credibility of ratings. Most importantly, other signaling devices such as advertising and branding are usually weak due to individual developers' budget constraint or lack of marketing capabilities. Therefore, rating is the main criterion that consumers rely on in app market. Besides consumers, developers and the platform owner also pay significant attentions on ratings. The evidence from real business practices shows that rating is one of the hottest topics and the biggest concerns among developers. Tens of thousands of posts on iPhoneDevSDK.com, one of the most popular online iPhone developer communities, are related to how to improve ratings and how ratings affect app sales revenue. Furthermore, developers agree upon revenue sharing contracts with the platform owner whose revenue is also connected with ratings. All these facts indicate that in app market ratings play a very significant role in determining both the success of an app and the prosperity of the market.

Given its great importance, this study focuses on the impact of rating on app market in terms of consumers' choice of purchase, developers' choice of app price and quality level, the platform owner's choice of revenue sharing policy and the social welfare. We first characterize consumer rating behavior and incorporate it into consumers' utility function. We then derive developers' optimal choice of app price and quality level. The platform owner's decision on revenue sharing policy will be associated with the parameters describing consumer rating behavior. We assume that consumer rating behavior consists of two processes. One is to give ratings, also referred to as *ex post* ratings, based on *ex post* user experience which can be quantified by consumers' received net utility after purchase (Kuksov and Xie 2010). The other is to accept *ex post* ratings and translate them into *ex ante* perceived net utility. These two processes, when combined, allow us to construct a bidirectional rating-utility framework in which we call the former utility-to-rating process and the latter rating-to-utility process.

We account for an essential aspect of rating behavior, subjectivity, which is also called "systematic rater error" as one type of rating errors discussed in the vast management literature regarding performance rating (Kane 1994, Kane et al. 1995, Yun et al. 2005, Borman 1977, Saal et al. 1980). In the utility-to-rating process, it is a consumer's subjective opinion to rate, for example 4 or 6 in a typical 10-point rating system, when her received net utility is zero. Similarly, in the rating-to-utility process, exactly which rating that a consumer believes to correspond to her perceived *ex ante* zero net utility is also subjective. Having random rating errors removed, this subjectivity in both processes has been demonstrated to be systematic (Kane 1994) if no effort is made to control its systematic sources. Obviously, this is the case for app market since consumers are under no control for their characteristics such as maturity, conscientiousness and degree of sophistication in using apps, which could serve as such systematic

resources. To model the subjective rating behavior in the utility-to-rating process, we assume a positive linear relationship between *ex post* rating and received net utility, which we define as the rating function. For the rating-to-utility process, we introduce a new concept: reservation rating. Reservation rating is defined as the *ex post* rating which signifies zero received net utility in consumers' *ex ante* perception. Based on reservation rating, a linear function between *ex post* rating and *ex ante* perceived net utility is established with potentially different slope and intercept than the rating function.

This work contributes to the existing literature in the following two aspects. The first is the bidirectional rating-utility framework we construct. While we take app market as our research context, this framework can be applied in any market where consumer rating plays an important role in purchase decisions. The concept of reservation rating introduced for rating-to-utility process is an attempt to systematically model how consumers interpret ratings into net utility. Second, our findings provide the guidance for developers and the platform owner on how to react to the changes in consumer rating behavior by adjusting their choices of quality, price, and revenue sharing percentage. We identified different aspects of rating behavior elicit different choices. We discover that rating leniency is detrimental to app quality. The analysis on social welfare demonstrates the potential misalignment between the interest of platform owner and the social welfare when consumers' rating behavior changes. Consumers' responsibility, in terms of their proper rating behavior, for the welfare of app society is also revealed. We also demonstrate that minimum app price, a common practice in real business, works as an instrument to drive out developers with poor matching performance and high cost rate. It is further shown that the platform owner can discriminate developers through a self-selection mechanism. It achieves a separating equilibrium in which high and low cost rate developers choose different revenue sharing percentages and the platform owner's profit is maximized.

The rest of the paper is organized as follows. In Section 2 we present a review of the related literature. In Section 3 we propose the base model. Section 4 extends the base model by imposing asymmetric rating-utility conversion rates and analyzes the social welfare under such settings. In Section 5 we add minimum app price restriction into the base model and examine how it affects the optimal choices. In Section 6 we study the platform owner's optimal revenue sharing policy when developers' cost rates are unobservable. We derive a self-selection mechanism to achieve separating equilibrium between developers with high and low cost rates. In Section 7, a "Win-Win" effect is identified when consumer reservation ratings are heterogeneous. We conclude and offer the directions for future research in Section 8.

2. Literature Review

The literature on the effects of online word-of-mouth (WOM) has been proliferating rapidly during the last ten years. Most of them are empirical work in the context of movie and book industry, focusing on

the effects of online WOM on predicting or influencing sales revenue (Dellarocas 2004, Chevalier and Mayzlin 2006, Liu 2006, Duan et al. 2008). Dellarocas (2004) demonstrates that online rating is a useful proxy for WOM in movie industry and it serves as one of the predictors for a movie's total revenue. Dellarocas and Narayan (2006) identify three metrics of online word-of-mouth: valence, variance and volume, in which valence is usually denoted by the average numeric average rating, variance is usually measured by its statistical variance or entropy (Godes and Mayzlin 2004), and volume is counted as the number of ratings. Liu (2006) shows that online WOM has significant explanatory power for box office revenue while the volume is a stronger predictor than valence. Duan et al. (2008) reveal that movies' online WOM valence significantly affects the volume which in turn determines the box office performance. For book markets, Chen et al. (2004) finds that the consumer rating is only a predictor for book sales. However, Chevalier and Mayzlin (2006) show that the improvement of online WOM valence will increase the book sales based on the data from Amazon.com and BarnesandNoble.com. Forman (2008) demonstrates that raters give more positive rating to the review with identity information and the disclosure of identity information increases the sales. Researchers have also investigated the connection between consumer ratings and sales in the context of other markets such as beer, DVD and video games (Clemson 2006, Hu et al. 2008, Zhu et al. 2010). However, little attention has been paid to software market. One reason might be that online software selling has not been widely established until app market appears. Zhou and Duan (2010) find that, from CNET download.com, the increase in product variety strengthens the impact of positive consumer reviews but weakens that of negative ones. They also show that positive expert reviews lead to more software download.

Given the above association between online WOM and product sales, it is the natural next step for researchers to conceive firms' optimal strategy to leverage such association. A growing body of literature has been devoted to this topic. Chen and Xie (2005) show that a firm should choose advertising instead of price to adapt consumer review when sufficient consumers value the product's horizontal features. Dellarocas (2006) demonstrates how firms' shilling behavior, i.e. post anonymous messages that exalt their products on the purpose of changing the consumers' perception, will influence firms' profits and consumers' surplus. Chen and Xie (2008) reveal when and how sellers should adjust their marketing communication strategy by changing product attribute information they offer to adapt consumer reviews. Kuksov and Xie (2010) study the optimal pricing and whether the firm should give an unexpected frill to early customers to boost their product experience. Li and Hitt (2010) show, analytically and empirically, that unidimensional ratings are more correlated with the net value, rather than quality, of the product. Firms need to account for price effects and can better serve the consumers by setting up review systems which explicitly separates the perceived value and quality. Jiang and Chen (2007) examine both consumer

review and consumer ratings and find that firms may have the incentive to under-charge in the early period. They also investigate how it will affect the market competition.

Most of the firms' adaption strategies mentioned above are implicitly based on the underlying assumption that ratings can signal the underlying true quality (or true value) to novice consumers at least in an expectation sense. However, this conjecture does not seem to be always true. The topic concerning how precise the ratings reflect the underlying true quality, an issue involved with consumer's rating behavior, becomes increasingly popular in the rating-related research area. Hu et al. (2006) empirically show that in the presence of under-reporting, i.e., only extremely satisfied or extremely unsatisfied consumers would rate, consumers may not extract the true quality from the ratings if only valence (mean value) is known. Sun (2010) analytically demonstrates that novice consumers can figure out product's true quality from the distribution of the ratings given by earlier consumers. Li and Hitt (2008) show that later consumers' perception on the quality may be inaccurate because of the biased ratings left by early buyers whose preferences on quality are different from later ones'. Hu (2009) summarizes two self-selection biases: purchasing bias partially mentioned in Li and Hit (2008), and under-reporting bias mentioned in Hu (2006), as two reasons which potentially lead to biased perception on true quality under certain circumstances. Moe and Trusov (2010) and Moe and Schweidel (2011) empirically show that consumers' rating behavior is significantly affected by previously posted ratings. Lee (2009) demonstrates that social imitation and learning affect can influence user rating generation.

Another important aspect of rating behavior which influences appraisal accuracy is the systematic rater error. While having not been noticed in the context of online WOM, it has been studied in the management literature for decades. Kane (1994) summarizes rating errors into multiple categories and points out that leniency, severity and non-differentiation are three important ones which could be potentially systematic. Kane et al. (1995) demonstrate that rating leniency is the most troublesome rating error and find it to be a relatively stable response tendency. Spence and Keeping (2010) suggest that when managers give performance ratings to their employees, more experienced managers are associated with lower ratings which indicate lower degree of leniency. Berger et al. (2010) empirically show that under the situation where employees' bonus payments are associated with ratings, a forced differentiated distribution requirement on ratings actually leads to higher productivity. This is especially intriguing since by analogy consumers in our context are more or less alike managers who rate developers' apps and consequently developers' revenue is associated with these ratings. This work incorporates the effect of this unique aspect of rating behavior – systematic rater error – on consumers' perception on utility.

3. The Model

Suppose that the platform owner, app developers and app consumers are the three players on an app market. A three-stage dynamic game theoretic model is constructed to study the equilibrium of the market.

In the first stage, the platform owner determines and publicizes developers' revenue-sharing percentage. By observing this percentage at the beginning of the second stage, developers choose either not to adopt and exit the market, or to adopt and then determine the optimal quality level and optimal app price. If developers choose not to adopt, the game ends. Otherwise, in the third stage, consumers decide whether to purchase, and, if a purchase occurs, rate the app based on the received net utility. The third stage is repeated such that after a sufficient time period the rating becomes steady. The goals of the platform owner and developers are to maximize their own benefits in this steady-state, respectively. Developer would choose to participate when the profit is greater than or equal to zero. Likewise, consumer would make the purchase when the expected net utility is greater than or equal to zero. In the following, we start with several essential preliminaries which serve as the foundation of our model.

Rating Function. We assume a linear relationship between the received net utility and the *ex post* rating. Since the rating, r , is bounded between 0 and 1, the mathematical formula is given by:

$$r = \min\{1, \max\{0, ku + r_0\}\}.$$

In the above expression, u is the received net utility. r_0 is the *ex post* zero net utility rating which is an important concept in terms of measuring the degree of severity. A low r_0 indicates that consumers are severe with respect to giving ratings. k is the rating-utility conversion rate. It represents consumers' sensitivity of the *ex post* rating on the difference in received net utility.

Received Net Utility. Following Chen and Xie (2008), we partition consumers into two groups. One consists of all high valuation consumers who consider the app match their taste so that they appreciate the quality. The other consists of all low valuation consumers who find the app a mismatch of their taste. The received net utility, which is associated with the app's "objective" true quality level q , its price p and consumers' valuation for quality, is given by:

$$u_{matched} = q - p, u_{unmatched} = -p.$$

We denote b as the fraction of consumers who belong to high valuation group. It is worth noting that b is an indicator of developers' matching performance. A large b represents a high level of matching performance since most of consumers are "matched." We assume that $b \in (0,1)$.

Reservation Rating. Following the definition of "reservation rating" in the introduction section, we further explain how it fits into economic utility theory. Reservation rating is intuitively a bar which an app needs to pass to be considered for purchase, and it also fundamentally affects consumers' willingness to pay. It works as a "ruler origin" to measure the *ex ante* perceived net utility from an app. For instance, with regard to a 5-star rating system, when a consumer with reservation rating 3.5-star observes an app with current rating of 4.5-star, she would expect some positive net utility from the app. In other words,

combining reservation rating and current *ex post* rating, consumers will figure out their *ex ante* perceived net utility. In the base model, reservation rating r_R is assumed to be homogenous among all consumers. Suppose r is the *ex post* rating given by high valuation consumers. The *ex ante* perceived net utility of high valuation consumers is given by:

$$u_e = \frac{r - r_R}{k}.$$

In order to distinguish the degree of criticism between utility-to-rating and rating-to-utility processes, we define that consumers with low r_0 are “severe” and consumers with high r_R are “critical.” The values of r_R and r_0 are between 0 and 1. Since generally consumers are equally critical or more critical in accepting ratings than giving ratings, we assume that $r_R = r_0$ or $r_R > r_0$.

Now, we derive consumers’ expected net utility $E(U)$. We assume that both true quality level and type of valuation are unknown to consumers before they experience the app. However, by examining the distribution of app’s *ex post* rating they would discover that the probability of being in the high valuation group is b and that in the low valuation group is $1 - b$. Hence, the expected net utility is given by:

$$E(U) = (1 - b) \cdot (-p) + b \cdot u_e.$$

If $E(U) \geq 0$, consumers would make the purchase.

Developers’ profit is given by:

$$u_D = ps - hq^2,$$

where h is the cost rate on quality and $h > 0$. The quadratic form of cost represents the diminishing return of investment on quality (Choudhary 2007). The condition for developers to participate is $u_D \geq 0$. Developers seek to maximize their profit by choosing the optimal p and q under constraint $E(U) \geq 0$.

Table 1. Model Parameters and Decision Variables

Parameters	
k	Rating-utility conversion rate
r_0	<i>ex post</i> zero net utility rating
r_R	Reservation rating
r	<i>ex post</i> rating from high valuation group consumers
b	Fraction of consumers in the high valuation group
h	Developers’ cost rate on quality
Decision Variables	
q	Quality level of the app
p	Price of the app
s	Developers’ revenue sharing percentage

The platform owner's revenue is given by:

$$u_p = p(1-s),$$

where s is bounded between 0 and 1. The platform owner's goal is to find the optimal s^* which maximizes u_p . In our model the platform owner's cost is neglected. Regarding the variable costs such as app hosting cost, we assume that they are already covered by a fixed annual fee paid by developers. For example, Apple iOS platform charges \$99 for annual membership to allow a developer upload her apps on the Apple App Store. We speculate that this membership fee may cover the cost, but not be the major revenue source. The evidence is that nowadays roughly 350,000 apps are available on the Apple App Store. Even under the most extreme case that they correspond to 350,000 developers, the annual revenue from membership is 35 million at most, which account for less than 4% of the total platform owner's app sales revenue. Once a developer decides to participate, the membership fee is a sunk cost which does not affect her choice of app quality level and price.

Table 1 summarizes the notations of the model. Based on the above assumptions, we have the following proposition.¹

Proposition 1. *Given the revenue sharing percentage s , developers' optimal choice of price p^* and quality level q^* are:*

i. *Region 1-1 (self-driven): where $b > b_1$ and $0 < h \leq h_1 s$,*

$$p_1^* = \frac{b(1-r_R)}{k(1-b)}, q_1^* = \frac{br_0 - r_R b - r_0 + 1}{k(1-b)};$$

ii. *Region 1-2 (platform owner driven): where $b > b_1$ and $h_1 s < h \leq h_2 s$,*

$$p_2^* = \frac{1}{2} \frac{sb^2}{h} + \frac{b(-r_R + r_0)}{k}, q_2^* = \frac{1}{2} \frac{sb}{h};$$

iii. *Region 1-3 (poor matching): where $b \leq b_1$ and $0 < h \leq h_3 s$,*

$$p_3^* = \frac{b(1-r_R)}{k(1-b)}, q_3^* = \frac{br_0 - r_R b - r_0 + 1}{k(1-b)}.$$

In all other regions of (b, h) , developers cannot make non-negative profit. The above thresholds are:

$$b_1 = 1 - \frac{1-r_R}{r_R-r_0}, h_1 = \frac{1}{2} \frac{bk(1-b)}{br_0 - r_R b - r_0 + 1}, h_2 = \frac{1}{4} \frac{bk}{r_R - r_0}, \text{ and } h_3 = \frac{kb(1-b)(1-r_R)}{(br_0 - r_R b - r_0 + 1)^2}.$$

Proposition 1 shows that developers' optimal choice of price and quality level is divided into three regions, depending upon their matching performance b and cost rate h . Generally speaking, a higher

¹ Proposition 1 can be shown by solving the corresponding Lagrangian in different regions defined by the model parameters. The sketch of proofs of other propositions and corollaries can be found in the appendix.

quality level q will lead to higher *ex post* ratings from high valuation consumers. The overall *ex post* ratings get elevated; this subsequently increases consumers' *ex ante* perceived net utility as well as their willingness to pay. Thus, developers can gain by charging a higher app price p to exploit the additional willingness to pay. For developers in Region 1-1, because of low cost rate ($0 < h \leq h_1s$), the marginal gain is always greater than marginal cost of quality until the rating r reaches its maximum. Therefore, the optimal quality level q_1^* is the one at which high valuation consumers will give the maximum rating (note that $r = k(q_1^* - p_1^*) + r_0 = 1$). In Region 1-2, when matching performance is good but the cost rate is high ($h_2s \geq h > h_1s$), the marginal cost of quality increases faster than that in Region 1-1 and will surpass the marginal gain from additional consumers' willingness to pay before the maximum $r = 1$ is realized. Thus, the r in Region 1-2 will be between r_R and 1. In Region 1-3, the optimal price p_3^* and quality level q_3^* are of the same analytical forms as Region 1-1. This indicates when their matching performance is poor but the cost rate is relatively low ($0 < h \leq h_3s$), developers' optimal choice is to produce at the quality level which yields $r = 1$. The intuition is that since only a small portion of consumers belong to high valuation type who appreciate high quality and rate positively after purchase, developers need to provide sufficient satisfaction to them in order to make them rate as high as possible. Otherwise the subsequent consumers' *ex ante* perceived net utility as well as their willingness to pay would be too low because the overall rating is too low. Figure 1 depicts developers' p^*, q^*, r^*, u_D^* in Regions 1-1 and 1-2. All of them are non-increasing function of developers' cost rate h .

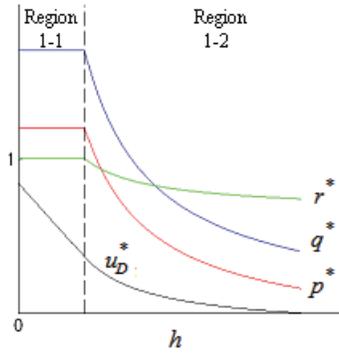


Figure 1 Developers' optimal choices given s

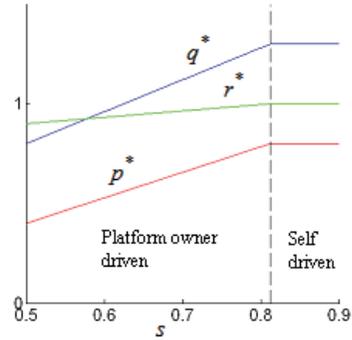


Figure 2 Developers' optimal choice against s

Corollary 1.1. $\partial p_2^* / \partial s > 0$ and $\partial q_2^* / \partial s > 0$.

Corollary 1.1 shows that for developers in Region 1-2, their optimal choice of quality level is driven by the revenue percentage they obtain. By increasing s , the platform owner can provide more incentive for developers to improve the quality of the app. We hence name Region 1-2 as “platform owner-driven,” and refer the developers as “platform owner-driven developers.” In Region 1-1, the optimal quality level

is not affected by s , and therefore the developers are “self-driven.” We further name Region 1-3 as “poor marketing region.” Since the optimality in poor marketing region shares many similarities with self-driven region and it is unlikely to be the main component of the market, we focus our attention on self-driven and platform owner-driven regions.

Notice that the boundaries of the regions (h_1s, h_2s) are proportional to revenue sharing percentage s which is set by the platform owner at the first stage of the game. This indicates that given their cost rate h , whether developers choose to be in self-driven or platform owner region, depends on not only on consumers’ rating behavior (k, r_0, r_R) but also the platform owner’s choice of s . Figure 2 illustrates that as s increases, owner-driven developers are induced to increase their q^* and p^* until the rating r hits the maximum where they become self-driven.

Corollary 1.2. $\partial q_1^* / \partial r_R < 0$; $\partial q_2^* / \partial r_R = 0$. In all regions, $\partial p^* / \partial r_R < 0$.

Corollary 1.2 indicates that when consumers become more critical in accepting ratings (higher r_R), developers in both platform owner-driven region and self-driven region need to charge lower app prices. In platform owner driven-region, charging a lower price according to our rating function will generate a higher r , which is used to satisfy higher r_R . In self-driven region, since r has already reached its maximum, to satisfy higher r_R so that enable $E(U) \geq 0$, the price need to be decreased at an even faster rate (shown in Figure 3). Regarding the quality level, platform owner-driven developers keep the same quality level while self-driven developers choose to lower the quality. This is because in self-driven region, as long as the maximum rating is retained, further quality improvement will not promote consumers’ perception on *ex ante* net utility. Since the price drops at a fast rate, the quality level required to achieve the maximum rating can be set lower.

Corollary 1.3. $\partial q_1^* / \partial r_0 < 0$; $\partial p_1^* / \partial r_0 = 0$. $\partial p_2^* / \partial r_0 > 0$; $\partial q_2^* / \partial r_0 = 0$.

When consumers becomes more lenient in giving ratings (higher r_0), self-driven developers respond by setting a lower quality level but keep the price unchanged. This is because when consumers become more lenient in the self-driven region, even though the quality decreases a little, the rating r will still be at the maximum. Decrease in quality won’t change the consumers’ willingness to pay so that the developer can charge the same price, while decreasing the quality definitely saves the cost. For platform owner-driven developers, the best strategy is to increase the price when consumers become more lenient. In a summary, the developer takes advantage of consumers’ leniency by increasing the price when she is in platform owner-driven region and by decreasing the quality when she is in the self-driven region.

Notice that the boundaries of the regions (h_1s, h_2s) are affected by r_0 and r_R . Figure 3 shows that when r_R increases, platform owner-driven region shrinks but self-driven region expands. However as r_0

increases, both regions expand. Platform owner-driven developers may switch to be self-driven as r_R or r_0 increases. Figure 3 also depicts how developers' choice of optimal quality and price changes across different regions.

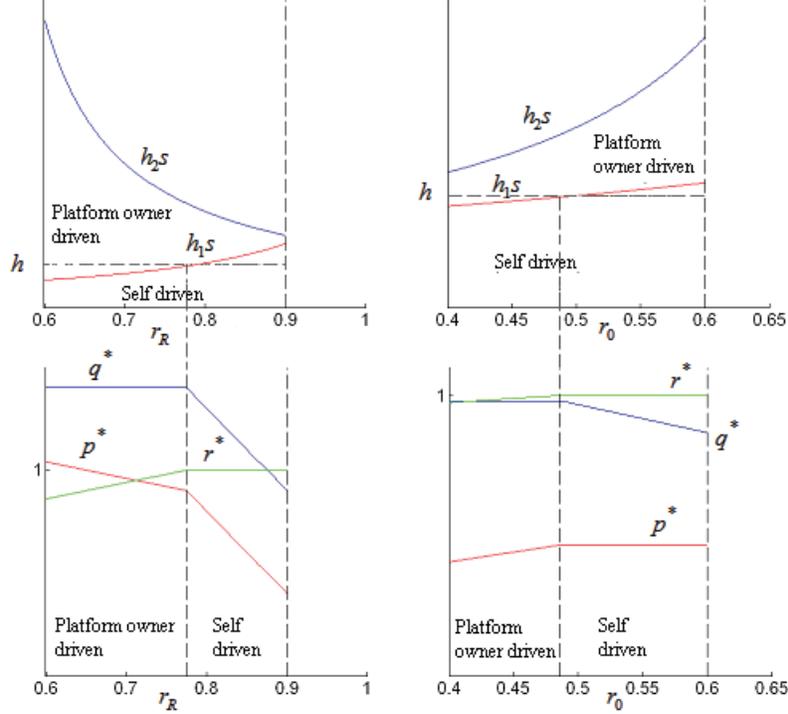


Figure 3. Region distribution and developers' optimal choices across regions

Corollary 1.4. *Change in rating behavior (r_0, r_R) doesn't change the developer's quality incentive in platform owner region.*

Corollary 1.4 results from the combination of Corollary 1.3 and Corollary 1.4. From Figure 3, we can see that the developer in platform owner region doesn't need to adjust optimal quality level when (r_0, r_R) changes. Price is the only factor need to adjust to optimally react to change in (r_0, r_R) in platform owner region.

Figure 3 also reveals an interesting phenomenon. Given developer's cost rate h , either increase in r_0 or r_R could potentially incentivize the developer to switch from the platform owner-driven region to the self-driven region, where the developer makes the $r = 1$. However, the underlying reasons are different. Increase r_R , i.e., the consumers become more critical in accepting ratings, "forces" the developer to render $r = 1$ to guarantee a enough space between r and r_R to sustain the price. Increase in r_0 , i.e. consumers become more lenient in giving ratings, "pleases" the developer to render $r = 1$ to elicit consumers' maximum willingness-to-pay at a lower quality cost.

Proposition 2. *The platform owner's optimal choice of developers' revenue sharing percentage s^* is:*

i. *Region 2-1 (squeezing): when $b > b_1$ and $0 < h \leq h_{1\alpha}$,*

$$s_1^* = \frac{2h(1 + br_0 - br_R - r_0)}{bk(1-b)};$$

ii. *Region 2-2- α (encouragement): when $b > b_1$ and $h_{1\alpha} < h \leq 2h_2/3$,*

$$s_{2\alpha}^* = \frac{1}{2} + \frac{h(r_R - r_0)}{bk};$$

iii. *Region 2-2- β (retention): when $b > b_1$ and $2h_2/3 < h \leq h_2$,*

$$s_{2\beta}^* = \frac{4h(r_R - r_0)}{bk};$$

iv. *Region 2-3: when $b \leq b_1$ and $0 < h \leq h_3$,*

$$s_3^* = \frac{h(1 + br_0 - br_R - r_0)^2}{b(1-r_R)k(1-b)}.$$

The threshold:

$$h_{1\alpha} = \frac{1}{2} \frac{bk(1-b)}{2 + br_0 - br_R - r_0 - r_R}.$$

Proposition 2 characterizes the subgame perfect Nash equilibrium (SPNE). It shows how the platform owner in the first stage should offer different revenue sharing percentage s to different types of developers depending upon their cost rate h and matching performance b . The SPNE endogenizes developers' best response in the second stage. For example, the platform owner knows that the developers in Region 2-1 would maximize their utility u_p by choosing to be self-driven in the second stage, and hence, offers s_1^* to incentivize them to do so.

Corollary 2.1. $\partial s^* / \partial h > 0$ where $s^* = s_1^*, s_{2\alpha}^*, s_{2\beta}^*$, or s_3^* .

Figure 4 depicts the optimal revenue sharing percentage s as a function of cost rate h . Both Regions 2-2- α and 2-2- β correspond to platform owner-driven region in Proposition 1. Figure 1 shows that when platform owner-driven developers' cost rate increases, the optimal quality level q_2^* decreases. Corollary 2.1 suggests that in this situation the platform owner should offer a higher revenue sharing percentage to encourage developers to produce at a higher quality level. The loss due to a lower sharing percentage for the platform owner will be more than compensated by the additional revenue gained from a higher quality level.

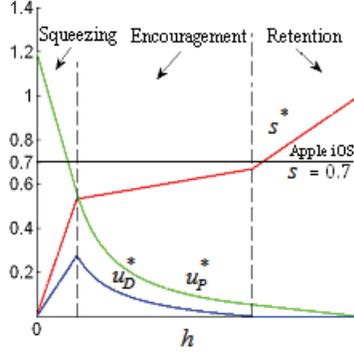


Figure 4. Platform owner's optimal choice of s

Region 2-1 corresponds to self-driven region in Proposition 1 where the optimal quality q_1^* is not a function of revenue sharing percentage. In this region, the platform owner can squeeze developers by giving a lower sharing percentage. Due to their low cost rate, developers can still obtain non-negative utility by choosing the quality level q_1^* at which the rating $r=1$ and charging the price p_1^* . Squeezing developers may be dangerous since developers may leave. One reason why low cost rate developers do not leave this platform even when receiving low revenue sharing percentage lies in our assumption that the platform owner is a monopoly or has the monopoly power on the market such as predominance in consumer base. For example, Apple still grasps roughly 70% of the entire mobile app market and this number is 99.4% two years ago. Developers are allured by the large number of potential customers. Another reason is that consumers using different platforms are usually separated, for example, chances are little that a consumer uses both iPhone and Andriod phone. Since cannibalization is a mild issue, developers can release their app on another platform as long as profitable

Based on the above discussions, we give more intuitive names to the regions in Proposition 2. Region 2-1 is hereafter referred to as “squeezing region.” Developers in squeezing region will be incentivized to choose self-driven region in the second stage. Region 2-2- α is named as “encouragement region.” Region 2-2- β is referred to as “retention region” where developers make zero profit and are hence on the verge of exiting the market. It can be observed in Figure 4 that developers whose cost rate is at the intersection of squeezing and encouragement regions can obtain the highest utility. Lower cost rate developers' utility will be squeezed. The extreme case is when the cost rate is close to zero the platform owner can take almost all the revenue but leave very little share to developers. Developers' participation can still be justified because developing app costs nearly nothing for them.

Corollary 2.2. $s_2^* > 1/2$ where $s_2^* = s_{2\alpha}^*$ or $s_{2\beta}^*$.

This coincides with our observation from real business practice, for instance, Apple iOS platform offers a revenue sharing percentage of 70%. In app market, given the fact that a large portion of developers are individuals or development teams made up of several individuals whose cost rate is

relatively high, the platform owner may consider them to be in encouragement or retention, rather than squeezing region. Therefore, it is the optimal strategy for the platform owner to offer a revenue sharing percentage greater than one half.

Corollary 2.3. $\partial s_1^* / \partial r_R < 0$. $\partial s_2^* / \partial r_R > 0$ where $s_2^* = s_{2\alpha}^*$ or $s_{2\beta}^*$.

According to Proposition 1, when consumers become more critical in accepting ratings, developers in encouragement and retention regions would decrease the price. Such a price drop reduces the total revenue. A raise in sharing percentage would encourage developers to choose a higher quality level and increase the price. The strategy in squeezing region appears to be a little counterintuitive. It states that in squeezing region when consumers' reservation rating increases the platform owner prefers squeezing developers more, rather than incentivizing them by offering higher revenue sharing. The underlying logic is the following. Notice that the threshold $h_{1\alpha}$ increases with r_R . Developers in squeezing region would still choose self-driven region at the second stage when r_R increases. A higher s_1^* would not promote the choice of quality level as in encouragement region but only subsidizing more revenue sharing to developers. Instead, if the platform owner decreases s_1^* and as long as the decreased s_1^* still maintains developers' choice of price and quality level, i.e., staying in self-driven region, the total app sales revenue doesn't change due to the unchanged price. By giving less s to the developer, platform owner would benefit. As illustrated in Figure 2, given developers' cost rate h , when the sharing percentage decreases developers originally in self-driven region may switch to platform owner-driven region (notice that $h_{1\alpha}$ decreases). However, it turns out that moving to platform owner-region will incur greater loss than staying in self-driven region even with less revenue percentage. Choosing between the two "evils," developers will be better off to stay in self-driven region. Hence the platform owner is able to squeeze.

Corollary 2.4. $\partial s_1^* / \partial r_0 < 0$. $\partial s_2^* / \partial r_0 < 0$ where $s_2^* = s_{2\alpha}^*$ or $s_{2\beta}^*$.

When r_0 increases, i.e., consumers generally give higher ratings, developers in encouragement and retention regions would charge a higher price but maintain the same level of quality. Higher price indicates higher revenue in this case and the platform owner can extract part of the additional revenue by decreasing the sharing percentage. For developers in squeezing region, the optimal price does not change but the quality level decreases since consumers are more lenient in giving ratings. The argument why s_1^* decreases with r_0 is similar to the case that r_R decreases s_1^* in squeezing region in Corollary 2.3.

Corollary 2.5. For squeezing, encouragement, and retention regions, in SPNE developers' profit satisfies $\partial u_D^* / \partial r_0 \geq 0$ and $\partial u_D^* / \partial r_R \leq 0$. For the platform owner's revenue, $\partial u_P^* / \partial r_0 > 0$ and $\partial u_P^* / \partial r_R < 0$.

4. Asymmetric Rating-Utility Conversion Rates

In this section, we extend the base model by setting different conversion rates between rating-to-utility and utility-to-rating processes. We denote the conversion rate in the rating-to-utility process as k_R , and that in the utility-to-rating process as k_U . Here we assume that $k_R > k_U$. This assumption suggests that consumers are less sensitive to the difference in received net utility when giving ratings than the change in *ex post* ratings when translating to their perceived net utility as well as corresponding willingness to pay.

4.1 The Platform Owner and Developers' Optimal Choices

The problem can be solved similar to the symmetric (same- k) case in the previous section. The solution is presented in Proposition A1 in the appendix. Similarly, we have identified three regions for developers: self-driven, platform owner-driven, and poor marketing. Their corresponding optimal price and quality are denoted as (p_1^{A*}, q_1^{A*}) , (p_2^{A*}, q_2^{A*}) , and (p_3^{A*}, q_3^{A*}) ,² respectively. Proposition A2 in the appendix describes the optimal revenue sharing percentage. Similar to those in Proposition 2, we have, respectively, s_1^{A*} for squeezing region, $s_{2\alpha}^{A*}$ for encouragement region, $s_{2\beta}^{A*}$ for retention region, and s_3^{A*} . The following corollaries characterize how conversion rates k_U and k_R affect the quality q^{A*} , price p^{A*} , and revenue sharing s^{A*} in these regions.

Corollary 3.1. *In all regions, $\partial q^{A*} / \partial k_R < 0$ and $\partial p^{A*} / \partial k_R < 0$.*

Corollary 3.2. *In self-driven region, $\partial q_1^{A*} / \partial k_U < 0$ and $\partial p_1^{A*} / \partial k_U = 0$. In platform owner-driven region, $\partial q_2^{A*} / \partial k_U > 0$ and $\partial p_2^{A*} / \partial k_U > 0$.*

When k_R decreases, i.e., consumers are willing to pay more for a higher rating, developers in any region should offer a higher quality level and charge a higher price. Corollary 3.2 suggests that when k_U increases, i.e., consumers are more sensitive to the difference in received net utility and willing to give more differentiated ratings, developers in platform owner-driven region should choose a higher quality level as well as a higher price. The reason is that when consumers appreciate high quality apps by giving more differentiated ratings, the *ex post* ratings will be translated into willingness to pay in the rating-to-utility process, hence, the benefit of producing high quality is augmented. On the contrary, if k_U decreases, i.e., consumers don't appreciate high utility apps and give alike ratings for good and bad apps, platform owner-driven developers have less incentive to produce at a higher quality level.

² Here a subscript A is added to denote the asymmetric case.

Developers in self-driven region, however, acts differently when k_U increases. They will keep the price unchanged but choose a lower quality level. This is because whenever the rating r reaches its maximum, any further improvement on quality will no longer be reflected by the rating. Consumers won't recognize such effort in the rating-to-utility process. Therefore self-driven developers cannot take advantage of increased k_U in the same way as platform owner-driven developers. Nonetheless, they can benefit from higher k_U through the reduction of the cost on quality since it would be less costly for them to achieve the quality level which realizes the maximum rating.

It should be noted that when r_0 increases, self-driven developers' quality level q_1^{A*} decreases but for platform owner-driven developers, q_2^{A*} remains unchanged. However, q_2^{A*} increases with k_U . This observation supports the finding in management literature that leniency is a significant problem. Berger et al. (2010) discover empirically in corporate environment a forced distribution on performance ratings will lead to higher productivity. We show that when r_0 decreases (less lenient) and k_U increases (more distributed), the productivity indicator q_2^{A*} increases.

Corollary 3.3. *Change in rating behavior (k_U, k_R) changes the developer's quality incentive in all regions.*

This corollary is a counterpart to Corollary 1.4. Corollary 1.4 states that the change in rating behavior with respect to (r_0, r_R) doesn't change the developer's quality incentive in platform owner-driven region. Here we reveal that change in (k_U, k_R) , on the contrary, changes the developer's quality incentive in all situations. Corollary 3.3 highlights the fact that different aspects of rating behavior are attached to different aspects of developer's incentives. The statement that consumers become more lenient in giving ratings" could be interpreted by either higher r_0 or higher k_U . However, the incentive consequences are quite discrepancy depending on which type of "rating leniency" is.

Corollary 3.4. *In encouragement or retention region, $\partial s_2^{A*} / \partial k_R > 0$ and $\partial s_2^{A*} / \partial k_U < 0$, where $s_2^{A*} = s_{2\alpha}^{A*}$ or $s_{2\beta}^{A*}$.*

When consumers are unwilling to pay differentiated price for high rating app (i.e., decreasing k_R), or consumers are unwilling to give differentiated ratings to high utility app (i.e., increasing k_U), according to Corollaries 3.1 and 3.2, developers in encouragement or retention region choose a lower quality level. Therefore the platform owner should offer higher revenue sharing to developers so as to maintain the quality level and hence the price for the platform owner's revenue.

Corollary 3.5. *In squeezing region, $\partial s_1^{A*} / \partial k_U < 0$. With respect to k_R ,*

i. Case 1: if $r_R \geq \hat{r}_R = (3 + r_0)/4$, $\partial s_1^{A*} / \partial k_R < 0$;

ii. Case 2: if $r_R < \hat{r}_R$ and $k_R > \hat{k}_R = \sqrt{\frac{1-r_R}{1-r_0}} \frac{bk_U}{(1-b)}$, $\partial s_1^{A*} / \partial k_R > 0$;

iii. Case 3: if $r_R < \hat{r}_R$ and $k_U < k_R \leq \hat{k}_R$, $\partial s_1^{A*} / \partial k_R < 0$.

As discussed earlier, when the developer would keep choosing self-driven region at the second stage, i.e., the final choice of price and the total app revenue are not influenced by the first stage s , at first stage the platform owner can decrease the s to make itself better off. Case 1 and Case 3 demonstrate this principle. However, Case 2 seems to violate it. The underlying reason is that when $r_R < \hat{r}_R$, high k_R could lead the developer into the platform owner region. Figures 5 and 6 clearly illustrate this dynamics. Figure 5 presents the situation for Case 1. It shows that when $r_R \geq \hat{r}_R$, i.e., the consumers are sufficiently critical in r_R dimension, developers in self-driven region would always be in that region as k_R increases. However, when the consumers are not sufficiently critical in r_R dimension ($r_R < \hat{r}_R$), which is illustrated by Figure 6 for Cases 2 and 3, developers in self-driven region will stay there when k_R is low, but choose switching to platform owner region when k_R is high. It is not the best interest for the platform owner that developer switches to the platform owner-driven region. Therefore, as illustrated by Figure 2, the platform owner chooses to increase s to make the developer stay in the self-driven region.

The underlying reason for the non-monotonic property shown in Figure 6 is the following. The k_R and r_R represent two types of “criticalness” in accepting ratings. For r_R , it is the bar that r has to pass for making purchase happen. As the “bar” becomes higher (r_R increases), the self-driven developer must adjust the price and quality to pass that “bar” otherwise no purchase will occur. However, for k_R , it is not the “bar” which has to be passed. When k_R is on the low side, meaning that consumers would interpret little extra rating above reservation rating to a great value, the developer doesn’t have to hit the maximum rating. Hence she would choose the platform owner-driven region. When k_R turns to be on the high side, meaning that consumers trust very little value within rating, it becomes not worthy for developer to spend extra quality cost to obtain high rating. As long as r_R is passed, purchase will occur.

Corollary 3.6. For squeezing, encouragement, and retention regions, in SPNE the platform owner’s revenue $\partial u_p^* / \partial k_U > 0$, $\partial u_p^* / \partial k_R < 0$, $\partial u_p^* / \partial r_0 > 0$, and $\partial u_p^* / \partial r_R < 0$.

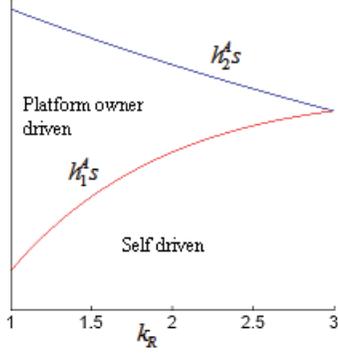


Figure 5. Region distribution as k_R ($r_R \geq \hat{r}_R$)

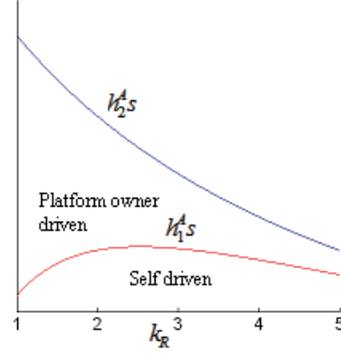


Figure 6. Region distribution as k_R ($r_R < \hat{r}_R$)

4.2 Analysis of Social Welfare

In this subsection, we analyze the impact of consumers' rating behavior on social welfare of app market. The social welfare is defined as:

$$W = bq - hq^2 .$$

Corollary 4.1. *In encouragement or retention region, $\partial W / \partial r_R > 0$ and $\partial W / \partial r_0 < 0$.*

In encouragement or retention region, if r_R increases or r_0 decreases, the social welfare increases. In these regions, there exists unrealized "surplus" because consumers are lenient towards developers. When consumers become more critical in accepting ratings or more severe in giving ratings, such "surplus" will be stimulated out.

Corollary 4.2. *$\partial W / \partial k_R < 0$ in encouragement region and $\partial W / \partial k_R = 0$ in retention region. $\partial W / \partial k_U > 0$ if $h \leq h_{2\alpha}^A$, and $\partial W / \partial k_U < 0$ if $h > h_{2\alpha}^A$ where:*

$$h_{2\alpha}^A = \frac{1}{2} \frac{k_R b k_U^2 (1-b)}{(b k_U - k_R b + k_R)^2 (r_R - r_0)} .$$

In encouragement region, the social welfare increases as k_R becomes small. This indicates that consumers need to appreciate ratings, that is, to accept the signal of utility and be willing to pay higher price for higher rating, for the sake of welfare of the market. If developers' cost rate is relatively lower ($h \leq h_{2\alpha}^A$), the social welfare increases when consumers give more differentiated ratings for good and bad apps (higher k_U). On the other hand, if there are high cost developers ($h > h_{2\alpha}^A$) on the market, more differentiated rating behavior is detrimental to the social welfare.

Corollary 4.3. *In squeezing region, $\partial W / \partial x < 0$ where $x = r_R, r_0, k_R, k_U$.*

In squeezing region, developers' cost rate is very low, being severe in giving ratings is beneficial for the social welfare. The intuition comes from the fact that developers have a great potential to produce high quality app. Consumers' severity in giving ratings helps to realize such potential and improve the

overall social welfare. However, paradoxically, being more critical in accepting ratings will actually lessen the social welfare. This is because it fails to force developers to promote the quality since they are self-driven. Such overly critical behavior would actually reduce developers' incentive since they cannot charge the price as high as before and consequently justify their investment on quality.

Table 2 compares the directions of change for platform owner's u_p^* and the social welfare W in response to the rating parameters r_R , r_0 , k_U and k_R in different regions. Positive sign “+” represents “increase” while negative sign “-” represents “decrease.” “N” means “No effect.” We highlight the situations where the interest of platform owner and the social welfare are always misaligned. We can observe that the adjustment of k_R is always aligned, but that of r_0 is always misaligned. This shows that leniency is always detrimental to the social welfare but always favored by the platform owner.

Table 2 Platform owner's (P.O.'s) interest V.S. Social welfare (S.W.)

		r_R	r_0	k_U	k_R
		+	+	+	+
Squeezing Region	P.O.	-	+	+	-
	S.W.	-	-	-	-
Encouragement Region	P.O.	-	+	+	-
	S.W.	+	-	+	-
Retention Region	P.O.	-	+	+	-
	S.W.	+	-	-	N

5. Minimum App Price

In this section, we impose the minimum price restriction to the base model to study its impact on developers' optimal choice of quality level and price and the platform owner's optimal revenue sharing percentage. We denote the minimum app price as p_m .

Lemma 5.1. *If $p_m \leq \hat{p}_{m1} = (2r_R - r_0 - 1)/k$, developers' optimal choice of price and quality level in self-driven and platform owner-driven regions remains unchanged. p_m affects only poor marketing region.*

Lemma 5.2. *If $p_m = \hat{p}_{m1}$, poor marketing region becomes unavailable.*

It is not very difficult to show $p_1^* > p_2^* > p_3^*$. So when p_m starts to increase from zero to $p_m < \hat{p}_{m1}$, it will affect poor marketing region. When p_m reaches \hat{p}_{m1} , developers in that poor marketing region cannot earn any non-negative net utility and are driven out the market..

Proposition 3. If $p_m > \hat{p}_m$, given the revenue sharing percentage s , developers' optimal choice of price p^* and quality level q^* are:

i. Region 3-1: when $b > b_1^M$ and $h \leq h_1 s$,

$$p_1^{M*} = p_1^*, q_1^{M*} = q_1^*;$$

ii. Region 3-2- α : when $b_1^M < b \leq b_2^M$ and $h_1 s < h \leq h_2^M s$, or, $b > b_2^M$ and $h_1 s < h \leq h_2 s$,

$$p_{2\alpha}^{M*} = p_2^*, q_{2\alpha}^{M*} = q_2^*;$$

iii. Region 3-2- β : when $b_1^M < b \leq b_2^M$ and $h_2^M s < h \leq h_2^M s$,

$$p_{2\beta}^{M*} = p_m, q_{2\beta}^{M*} = \frac{p_m k - r_0 b + r_R b}{b k}.$$

In all other regions of (b, h) , developers cannot make non-negative profit. The above thresholds are:

$$b_1^M = \frac{p_m k}{1 - r_R + p_m k}, b_2^M = \frac{p_m k}{r_R - r_0}, h_2^M = \frac{1}{2} \frac{b^2 k}{b r_R - r_0 b + p_m k} \text{ and } h_2^M = \frac{p_m b^2 k^2}{(b r_R - r_0 b + p_m k)^2}.$$

Figure 7 plots the regions defined in the above proposition. Region 3-1 is derived from self-driven region but cut smaller by p_m . Similarly, Regions 3-2- α and 3-2- β result from platform owner-driven region but are modified by p_m . Proposition 3 reveals that developers with market performance $b < b_1^M$ will be driven out of the market regardless of their cost rate. This is because when $b < b_1^M$, even if developers have the lowest cost rate to receive the maximum rating from high valuation consumers at no cost, there will be sufficient low valuation consumers to give the lowest rating. This would result in the overall *ex ante* perceived net utility not high enough for developers to charge p_m . In this sense, one effect of minimum app price p_m is to drive low matching performance developers out of the market.

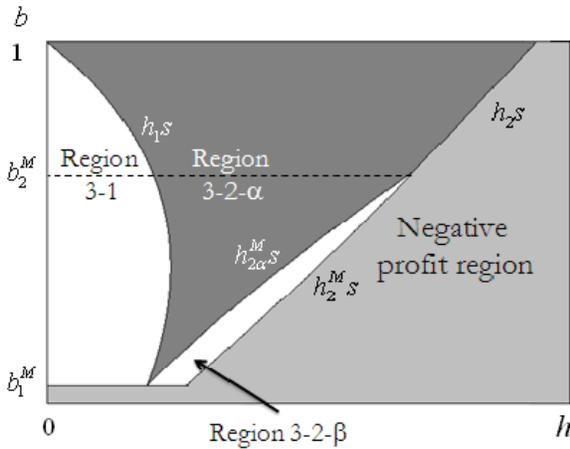


Figure 7. Regions defined in Proposition 3

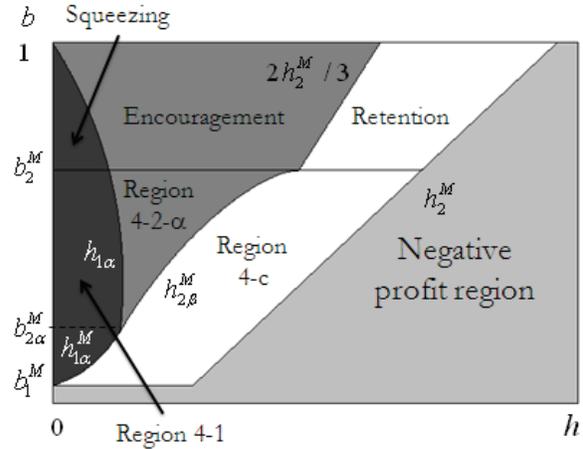


Figure 8. Regions defined in Proposition 4

Corollary 5.1. *When $p_m > \hat{p}_{m2} = (r_R - r_0)/k$, minimum app price shrinks platform owner-driven region to allow developers with lower cost rate.*

Notice that $\hat{p}_{m2} > \hat{p}_{m1}$ and when $p_m > \hat{p}_{m2}$, we have $b_2^M > 1$. Therefore, the sub region ($b > b_2^M$ and $h_1 s < h \leq h_2 s$) in Region 3-2- α becomes unavailable, and platform owner-driven region is defined by ($b > b_1^M$ and $h_1 s < h \leq h_{2\alpha}^M s$). It can be easily shown that $b_1^M > b_1$ and $h_{2\alpha}^M < h_2$. This indicates another effect of minimum app price. A higher p_m can not only control for the entry of poor matching performance developers, but also expel developers with high cost rate from the market.

Lemma 5.3. *When $p_m > \hat{p}_{m1}$ and $b > b_2^M$, the SPNE remains the same as that in Proposition 2 except that Region 2-3 is unavailable and $b > b_2^M$ is required as the lower bound of b .*

Proposition 4. *When $p_m > \hat{p}_{m1}$ and $b_1^M < b \leq b_2^M$ the SPNE is:*

i. *Region 4-1: when $b_1^M < b \leq b_{2\alpha}^M$ and $0 < h \leq h_{1\alpha}^M$, or, when $b_{2\alpha}^M < b \leq b_2^M$ and $0 < h \leq h_{1\alpha}$,*

$$s_1^{M*} = s_1^*; \quad (4.1)$$

ii. *Region 4-c: when $b_1^M < b \leq b_{2\alpha}^M$ and $h_{1\alpha}^M < h \leq h_2^M$, or, when $b_{2\alpha}^M < b \leq b_2^M$ and $h_{2\beta}^M < h \leq h_2^M$,*

$$s_c^{M*} = \frac{h(br_R - r_0 b + p_m k)^2}{p_m b^2 k^2}; \quad (4.2)$$

iii. *Region 4-2- α : when $b_{2\alpha}^M < b \leq b_2^M$ and $h_{1\alpha} < h \leq h_{2\beta}^M$,*

$$s_{2\alpha}^{M*} = s_{2\alpha}^*, \quad (4.3)$$

where $b_{2\alpha}^M \in [b_1^M, b_2^M]$ is a root of $f_s(b) = (b_1^M)^2 b^2 ((1-b)^2 - 2(1-b_1)^2) + (b_2^M)^2 (2b - (1+b)b_1^M)^2$; and

$$h_{1\alpha}^M = \frac{b^3(1-r_R)k(1-b) - p_m b^2 k^2 (1-b)^2}{2(1-r_R)(1+br_0 - br_R - r_0)b^2 - (br_R - br_0 + p_m k)^2 (1-b)^2},$$

$$h_{2\beta}^M = \frac{\left(2p_m k + (r_R - r_0)b - \sqrt{2p_m^2 k^2 - 2(r_R - r_0)^2 b^2}\right)kb^2}{6(r_R - r_0)^2 b^2 + 8p_m k(r_R - r_0)b + 4p_m^2 k^2}.$$

Figure 8 plots the regions defined in Proposition 4. The introduction of minimum app price, as a tool for the platform owner, can lead to considerable changes in the strategy of choosing optimal sharing percentage. One key implication hidden in the Proposition 4 can be separated out in the following corollary.

Corollary 5.2. *For developers in Region 4-c, the optimal sharing percentage s_c^{M*} satisfies: $\partial s_c^{M*} / \partial p_m > 0$, $\partial s_c^{M*} / \partial r_R > 0$, and $\partial s_c^{M*} / \partial r_0 < 0$.*

Corollary 5.2 reveals an important dynamics between minimum app price p_m and the optimal revenue sharing percentage. It especially complies with our observation in real business practice. Compared to Apple iOS platform has minimum app price $p_m = \$0.99$ and a revenue sharing percentage of 70%, RIM sets minimum app price $p_m = \$2.99$ for Blackberry app and offers a revenue sharing percentage 80%, higher than Apple's. Notice that Region 4-c actually could represent a large portion of developers especially individual developers in app market if the gap between r_R and r_0 is not very large.

6. Unobserved Cost Rate

According to Proposition 2 in the base model, the platform owner's optimal choice of revenue sharing percentage is a function of developers' cost rate h . However, h may be unobservable by the platform owner. In this section, we present the optimal choice of sharing percentage when h is unobserved. We also demonstrate that the platform owner can offer developers a menu with combinations of *ex post* ratings and revenue sharing percentage to engage them into a self-selection process.

6.1 Unobservable h

Suppose that the platform owner has no information about h . One approach to model the uncertainty with no prior information is to assume uniform distribution. Assume that h is uniformly distributed in $[0, h_U]$ where $h_U > h_3$. This represents the situation where developers' cost rate space is not fully covered. It is reasonable since there are always developers whose cost rate is so high and who cannot earn non-negative net surplus even when all the revenue goes into them.

Proposition 5. *When h is uniformly distributed, the expected optimal developers' revenue sharing percentage $s^{N*} = 1/2$.*

From the base model we can see the optimal sharing percentage is a linear function of h . Intuitively speaking, when h is on the lower side of the distribution, the optimal sharing percentage is less than one half; whereas when h is on the higher side of the distribution, the optimal sharing percentage is greater than one half. Proposition 5 tells the expectation is exact one half. In real business practice, this percentage is actually adopted by China Mobile, the China's largest wireless carrier, for China's app market. Many commentators question whether this is a right decision to make, especially after comparing to the successful business models such as Apple iOS and Google Android in US app market. Our model provides an explanation that on expectation one half is the optimal under uniform distribution of developers' cost rate. However, if the platform owner believes the distribution is skew toward higher values, the one half will be no longer optimal and in such case the optimal sharing percentage is greater than one half.

6.2 Self-selection

In this subsection, we show that the platform owner can design a menu of (s, r) to engage developers to a self-selection process. Note that developers with different cost rate have different objective for the rating. Figure 1 shows that self-driven developers aim at $r = 1$ while platform owner-driven developers' goal is between r_R and 1. Therefore although cost rate h is not observable, the rating r is an observable factor signaling developers' cost rate h . Hence, if the platform owner can offer different revenue sharing percentages bundled with different realization of r , developers will self-select the combination which is most profitable to them. This turns the information structure of our three-party game into a screening game in conjunction with a signaling game. The screening game is between the platform owner and developers, where the platform owner is the uninformed party. The signaling game is between developers and consumers, where developers are uninformed. In both games, developers are the informed party who would like to signal their app's quality to consumers (signaling game) and also engage into a self-selection generated by the platform owner (screening game). Figure 9 shows the role of rating in a two-sided market.

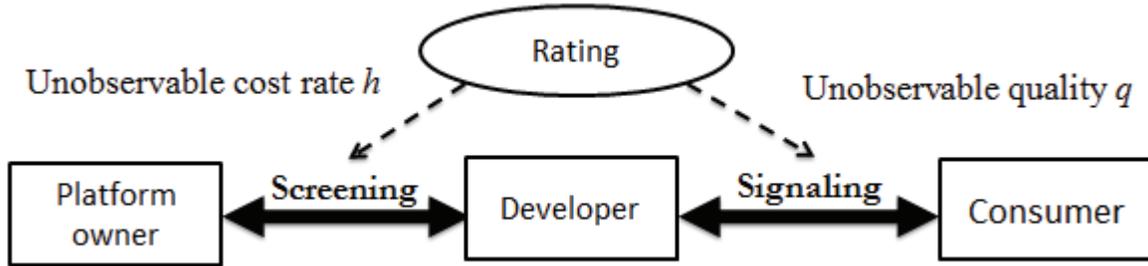


Figure 9. The role of rating in two-sided market

We consider a separating equilibrium between self-driven and platform owner-driven developers. Therefore, suppose that there exists a representative self-driven developer with cost rate h_L in squeezing region and a representative platform owner-driven developer with cost rate h_H in primary encouragement region (Region 2-2). We have the following result.

Proposition 6. *There exists a separating equilibrium in which the representative developers h_L and h_H would self-select to s_L and s_H respectively if the following menu is offered:*

$$s^N = \begin{cases} s_H, & \text{if } r \leq \hat{r}; \\ s_L, & \text{if } r > \hat{r}, \end{cases}$$

where,

$$\hat{r} = r_0 - \frac{(r_R - r_0)(1+b)}{2} - \frac{1}{4} \frac{bk(1-b)}{h_H}, \quad s_L = \frac{2h_L(1+br_0-br_R-r_0)}{bk(1-b)}, \quad \text{and} \quad s_H = \frac{1}{2} + \frac{h_H(r_R-r_0)}{bk}.$$

Proposition 6 characterizes a separating equilibrium for the screening game. By offering this selection to developers, without information on developers' cost rate h , the platform owner can incentivize h_H -type developers to self-select the corresponding sharing percentage s_H and h_L -type developers to s_L , both of which are stated in Proposition 2.

Corollary 6.1. $\partial \hat{r} / \partial r_0 > 0$ and $\partial \hat{r} / \partial r_R < 0$.

Corollary 6.1 indicates that when consumers become more severe in giving ratings, the equilibrium threshold rating \hat{r} should decrease. However, if consumers become more critical in accepting ratings, \hat{r} should also decrease.

7. Heterogeneous Reservation Ratings

In the base model, we assume that consumers have homogenous reservation rating. In this section, this assumption is relaxed to allow a discrete distribution of heterogeneous reservation ratings. Suppose that consumers have either one of the two reservation ratings, r_{RL} and r_{RH} . The corresponding fractions of consumer population are n_{RL} and $n_{RH} = 1 - n_{RL}$. For simplicity, we assume that $r_{RL} = r_0$ and $r_{RH} > r_0$.

This structure of heterogeneous reservation ratings offers developers two choices, targeting on low reservation rating only to attract consumers in such group, or, on high reservation rating to draw consumers from both group. In the based model developers must raise r at least to r_R in order to have the demand. Here, developers only need to reach r_{RL} , which is potentially less than r_R , and enjoy n_{RL} share of the market. Developers would make the decision on selecting between these two alternatives, depending upon their (b, h) .

We make the assumption that $0 < n_{RL} \leq (1 - r_{RH}) / (1 - r_0)$. In app setting, the distribution of reservation ratings is skew towards higher values since most of the people are more severe in giving ratings than accepting ratings. The results of developers' choice of quality and price are presented in Proposition A3 in the appendix. The SPNE resembles the structures stated in Proposition 2. Here we present a special phenomenon when reservation ratings are heterogeneous.

Proposition 7. *If $b > b_{H1}$ and $h \in H^{WW}$, both*

$$s_1^{E*} = \frac{1}{2} \text{ or } s_2^{E*} = \frac{1}{2} + \frac{h(r_{RH} - r_0)}{bk},$$

are Nash equilibriums (NE) while s_2^{E} is the SPNE. s_2^{E*} yields a greater profit than s_1^{E*} for both platform owner and developers. Here,*

$$b_{H1} = 1 - \frac{(1 - n_{RL}^2)(1 - r_{RH})}{(1 + n_{RL}^2)(r_{RH} - r_0)} \text{ and } H^{WW} = \left[\frac{1}{8} \frac{kb(1 - n_{RL}^2)}{r_{RH} - r_0}, \frac{1}{2} \frac{kb(1 - n_{RL}^2)}{(r_{RH} - r_0)(3 + n_{RL}^2)} \right].$$

Proposition 7 suggests that there exists a special situation in which the platform owner can choose between two s^{E*} for developers with their h locating in H_{WW} . Each alternative is a NE. However, by increasing from s_1^{E*} to s_2^{E*} , although the platform owner's share decreases, it incentivizes developers to make a bigger "pie" so both the platform owner and developers' profits are augmented. We call this a "Win-Win" effect and the interval of cost rate H_{WW} is the "Win-Win" region. It is worth noting that when n_{RL} increases, this "Win-Win" region diminishes. This "Win-Win" region confirms that our results under the setting of homogenous reservation rating are considerably robust when the heterogeneity on reservation rating is not severe ($n_{RL} \leq (1-r_{RH})/(1-r_0)$). When n_{RL} is not large, the results are alike between homogenous and heterogeneous cases. Hence, the platform owner can make the decisions by largely ignoring the presence of heterogeneous reservation ratings.

8. Discussion and Future Research

In this paper, we parameterize consumer rating behavior into four parameters (k_U, k_R, r_0, r_R) and construct a bidirectional rating-utility framework which incorporates these parameters into consumers' utility functions. In the equilibrium analysis, we identify three types of developers: self-driven, platform owner-driven and poor marketing, and reveal how the changes in consumer rating behavior (k_U, k_R, r_0, r_R) affect their optimal choices of quality level and app price, as well as the platform owner's optimal choice of revenue sharing policy and the social welfare. We discover the (r_0, r_R) aspect of rating behavior have different quality implication than the (k_U, k_R) aspect. The leniency issue, a well-known problem which has been empirically discovered in many behavioral management studies, is analytically observed in our rating-utility economic model. Our analysis on social welfare reveals that some consumers rating behaviors, though increasing the social welfare, may not be aligned with the platform owner's interest. We find that minimum app price can work as an instrument to rule out developers with low matching performance. The result that the optimal developers' revenue sharing percentage increases with minimum app price appears to explain what we observe in real business practice. When developers' cost rate is unobservable, the platform owner can design a screening game which yields a self-selection process leading to a separating equilibrium for high and low cost rate developers.

It should be noted that the dynamics between platform owner's optimal revenue sharing percentage and (k_U, k_R, r_0, r_R) are different in squeezing and encouragement regions. This discrepancy provides the platform owner the incentive to separate these two regions into multiple storefronts. Organizational developers such as software companies have relatively low cost rate and are more likely to reside in squeezing region. The platform owner may set up an additional storefront, for example "company app

store,” complementary to “personal app store” where individual developers are hosted. The platform owner may design different rating systems and adopt different revenue sharing percentages for these two stores. We believe as the rapid expansion of app market continues, the increased heterogeneity on app supply side would call for this approach.

While this paper takes a first step to incorporate subjective rating behavior into traditional utility-based analysis, one direction of future research may be attributed to studying the impact of rating behavior on other aspects of traditional industrial organization (IO) problems such as competition and product differentiation. Research questions such as how developers would compete on app ratings and how consumer rating behavior affects the platform competition would be interesting. The impact of rating behavior on developers’ product differentiation strategy, such as how developers manipulate ratings to form consumers’ *ex ante* perception on product differentiation, would also be a potential topic to explore. Another direction for future research is from the behavioral angle. Extant behavioral research suggests that the discrepancy of scaling in rating systems, for example 5-star and 10-point scale ratings, would introduce bias in rating behavior. Thus, an interesting question is how the platform owner designs an appropriate rating scale to shape consumer rating behavior to the desirable (k_U, k_R, r_0, r_R) demonstrated in this paper. Finally, our paper speculates the existence of reservation rating r_R . The empirical research might be able to detect it by analyzing online WOM data.

REFERENCE

- Akerlof, G.A. 1970. The Market for “Lemons”: Quality Uncertainty and the Market Mechanism. *The Quarterly Journal of Economics*, Vol. 84, No. 3, pp. 488-500.
- Berger, J., Harbring, C., Sliwka, D. 2010. Performance Appraisals and the Impact of Forced Distribution: An Experimental Investigation. *Working Paper*.
- Borman, W.C. 1977. Consistency of rating accuracy and rating errors in the judgment of human performance. *Organizational Behavior and Human Performance*, Volume 20, Issue 2, Pages 238-252
- Chen, Y., Xie, J. 2005. Third-Party Product Review and Firm Marketing Strategy. *Marketing Sci.* Vol. 24, No. 2, pp. 218–240
- Chen, Y., Xie, J. 2008. Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix. *Management Science*. Vol. 54, No. 3, March 2008, pp. 477–491
- Chevalier, J. A., Mayzlin, D. 2006. The Effect of Word of Mouth on Sales: Online Book Reviews. *Journal of Marketing Research*. Vol. XLIII, 345–354

- Choudhary, V. 2007. Comparison of Software Quality Under Perpetual Licensing and Software as a Service. *Journal of Management Information Systems*. Vol. 24, No. 2, pp. 141–165.
- Forman, C. 2008. Examining the Relationship Between Reviews and Sales: The Role of Reviewer Identity Disclosure in Electronic Markets. *Information Systems Research*. Vol. 19, No. 3, pp. 291–313
- Clemons, E.K., Gao, G., Hitt, L. M. 2006. When Online Reviews Meet Hyperdifferentiation: A Study of the Craft Beer Industry. *Journal of Management Information Systems*. Vol. 23, No. 2, pp. 149–171
- Dellarocas, C., Farag, N.A., Zhang, X. 2004. Using Online Reviews as a Proxy of Word-of-Mouth for Motion Picture Revenue Forecasting. *Working Paper*
- Dellarocas, C. 2006. Strategic Manipulation of Internet Opinion Forums: Implications for Consumers and Firms. *Management Science*. Vol. 52, No. 10, pp. 1577–1593
- Godes, D., Mayzlin, D. 2004. Using Online Conversations to Study Word-of-Mouth Communication. *Marketing Science*, Vol. 23, No. 4, pp. 545-560
- Hu, N., Pavlou, P.A., Zhang, J. 2006. Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of Online word-of-mouth communication. *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, New York, NY, USA, 324-330
- Hu, N., Liu, L., Zhang, J. 2008. Do online reviews affect product sales? The role of reviewer characteristics and temporal effects. *Inf Technol Manage* 9:201–214
- Hu, N., Zhang, J., Pavlou, P.A. 2009. Overcoming the J-shaped Distribution of Product Reviews. *Communications of the ACM*. Vol. 52, No. 10, pp. 144-147
- Jiang, B., Chen, P. 2007. An Economic Analysis of Online Product Reviews and Ratings. *Working Paper*.
- Kane, J.S., 1994. A Model of Volitional Rating Behavior. *Human Resource Management Review*, Vol. 4, No. 3, pp. 283-310
- Kane, J.S., Bernardin, H.J., Villanova, P., Peyrefitte, J. 1995. Stability of Rater Leniency: Three Studies. *The Academy of Management Journal*, Vol. 38, No. 4, pp. 1036-1051
- Kuksov, D., Xie, Y. 2010. Pricing, Frills, and Customer Ratings. *Marketing Sci*. Vol. 29, No. 5, pp. 925–943
- Lee, Y.J., Tan, Y., Hosanager, K. 2009. Do I Follow My Friends or the Crowds? Examining Informational Cascades in Online Movie Reviews. *Proceedings of WITS 2009*.
- Li, X. Hitt, L.M. 2008. Self-Selection and Information Role of Online Product Reviews. *Information Systems Research*. Vol. 19, No. 4, pp. 456–474
- Li, X., Hitt, L.M. 2010. Price Effects in Online Product Reviews: An Analytical Model and Empirical Analysis. *MIS Quarterly*, (34: 4) pp. 809-831.

- Liu, Y. 2006. Word of Mouth for Movies: Its Dynamics and Impact on Box Office Revenue. *Journal of Marketing*. Vol. 70, 74–89
- Moe, W., Schweidel, D.A. 2011. Online Product Opinions: Incidence, Evaluation and Evolution. *Working Paper*
- Moe, W., Trusov, M. 2011. Measuring the Value of Social Dynamics in Online Product Ratings Forums. *Journal of Marketing Research*. forthcoming.
- Saal, F., Downey, R.G., Lahey, M.A. 1980. Rating the ratings: Assessing the psychometric quality of rating data. *Psychological Bulletin*. Vol. 88, Issue 2, pp 413-428
- Shapiro, C. 1985. Optimal pricing of experience goods. *The Bell Journal of Economics*. Vol. 14, No. 2, pp 497-507
- Sun, M. 2010. How Does Variance of Product Ratings Matter? *Working Paper*
- Yun, G.J., Donahue, L.M., Dudley, N.M., McFarland, L.A. 2005. Rater Personality, Rating Format, and Social Context: Implications for Performance Appraisal Ratings. *International Journal of Selection and Assessment*. Vol. 13, No. 2, pp 97-107
- Zhou, W., Duan, W. 2010. Online User Reviews and Professional Reviews: A Bayesian Approach to Model Mediation and Moderation Effects. *ICIS 2010 Proceedings*. Paper 256
- Zhu, F., Zhang, X. 2010. Impact of Online Consumer Reviews on Sales: The Moderating Role of Product and Consumer Characteristics. *Journal of Marketing*. Vol. 74, 133–148

Appendix

A. Additional Propositions

Proposition A1. *Given the revenue sharing percentage s , developers' optimal choice of price p^* and quality level q^* are:*

i. *Region A1-1: (self-driven) when $b > b_1^A$ and $0 < h \leq h_1^A s$,*

$$p_1^{A*} = \frac{b(1-r_R)}{k_R(1-b)}, q_1^{A*} = \frac{(1-r_0)(1-b)k_R + bk_U(1-r_R)}{k_R k_U(1-b)};$$

ii. *Region A1-2: (platform owner driven) when $b > b_1^A$ and $h_1^A s < h \leq h_2^A s$,*

$$p_2^{A*} = \frac{1}{2} \frac{b^2 k_U^2 s}{h(bk_U + k_R - bk_R)^2} - \frac{(r_R - r_0)b}{bk_U + k_R - bk_R}, q_2^{A*} = \frac{1}{2} \frac{bk_U s}{(bk_U + k_R - bk_R)h};$$

iii. *Region A1-3: (poor marketing) when $b \leq b_1^A$ and $0 < h \leq h_3^A s$,*

$$p_3^{A*} = \frac{b(1-r_R)}{k_R(1-b)}, q_3^{A*} = \frac{(1-r_0)(1-b)k_R + bk_U(1-r_R)}{k_R k_U(1-b)}.$$

In all other regions of (b, h) , developers cannot make non-negative profit. The above thresholds are:

$$b_1^A = \frac{(2r_R - 1 - r_0)k_R}{(2r_R - 1 - r_0)k_R + k_U(1 - r_R)}, h_1^A = \frac{1}{2} \frac{bk_U^2 k_R (1 - b)}{(bk_U + k_R - bk_R)((1 - r_0)(1 - b)k_R + bk_U(1 - r_R))},$$

$$h_2^A = \frac{1}{4} \frac{bk_U^2}{(r_R - r_0)(bk_U + k_R - bk_R)}, \text{ and } h_3^A = \frac{k_U^2 bk_R (1 - r_R)(1 - b)}{((1 - b)(1 - r_0)k_R + bk_U(1 - r_R))^2}.$$

Proposition A2. *The platform owner's optimal choice of developers' revenue sharing percentage s^* is:*

i. *Region A2-1: (squeezing) when $b > b_1^A$ and $0 < h \leq h_{1\alpha}^A$,*

$$s_{1\alpha}^{A*} = \frac{2h(bk_U + k_R - bk_R)((1 - r_0)(1 - b)k_R + bk_U(1 - r_R))}{bk_U^2 k_R (1 - b)};$$

ii. *Region A2-2- α : (encouragement) when $b > b_1^A$ and $h_{1\alpha}^A < h \leq 2h_2^A / 3$,*

$$s_{2\alpha}^{A*} = \frac{1}{2} + \frac{h(r_R - r_0)(bk_U + k_R - bk_R)}{bk_U^2};$$

iii. *Region A2-2- β : (retention) when $b > b_1^A$ and $2h_2^A / 3 < h \leq h_2^A$,*

$$s_{2\beta}^{A*} = \frac{4h(r_R - r_0)(bk_U + k_R - bk_R)}{bk_U^2};$$

iv. *Region A2-3: when $b \leq b_1^A$ and $0 < h \leq h_3^A$,*

$$s_3^{A*} = \frac{h((1 - b)(1 - r_0)k_R + bk_U(1 - r_R))^2}{k_U^2 bk_R (1 - r_R)(1 - b)}.$$

The threshold:

$$h_{1\alpha}^A = \frac{1}{2} \frac{k_R bk_U^2 (1 - b)}{(bk_U + k_R - bk_R)((2 - r_R - r_0)(1 - b)k_R + 2bk_U(1 - r_R))}.$$

Proposition A3. *Given the revenue sharing percentage s and consumers are heterogeneous in reservation ratings, the developers' optimal choice of price p^* and quality level q^* are:*

i. *Region A3-1: when $b > b_{H1}$ and $h \leq h_{H1}$,*

$$p_{H1}^* = \frac{b(1 - r_{RH})}{k(1 - b)}, q_{H1}^* = \frac{br_0 - r_{RH}b - r_0 + 1}{k(1 - b)};$$

ii. *Region A3-2: when $b > b_{H1}$ and $h_{H1} < h \leq h_{H2}$,*

$$p_{H2}^* = \frac{1}{2} \frac{sb^2}{h} + \frac{b(r_0 - r_{RH})}{k}, q_{H2}^* = \frac{1}{2} \frac{sb}{h};$$

iii. Region A3-3: when $b > b_{H1}$ and $h_{H2} < h$,

$$p_{L2}^* = \frac{1}{2} \frac{b^2 n_{RL} s}{h}, q_{L2}^* = \frac{1}{2} \frac{n_{RL} s b}{h};$$

iv. Region A3-4: when $b \leq b_{H1}$ and $h \leq h_{H3}$,

$$p_{H1}^* = \frac{b(1-r_{RH})}{k(1-b)}, q_{H1}^* = \frac{br_0 - r_{RH}b - r_0 + 1}{k(1-b)};$$

v. Region A3-5: when $b \leq b_{H1}$ and $h > h_{H3}$,

$$p_{L2}^* = \frac{1}{2} \frac{b^2 n_{RL} s}{h}, q_{L2}^* = \frac{1}{2} \frac{n_{RL} s b}{h}.$$

In all other regions of (b, h) , developers cannot make non-negative profit. The above thresholds are:

$$b_{H1} = 1 - \frac{(1-n_{RL}^2)(1-r_{RH})}{(1+n_{RL}^2)(r_{RH}-r_0)}, h_{H1} = \frac{1}{2} \frac{bks(1-b)}{(br_0 - r_{RH}b - r_0 + 1)}, h_{H2} = \frac{1}{4} \frac{bks(1-n_{RL}^2)}{(r_{RH}-r_0)}, \text{ and}$$

$$h_{H3} = \frac{1}{2} \frac{\left(1 - r_{RH} + \sqrt{(1-r_{RH})^2 - n_{RL}^2(1-r_{RH}b - r_0 + br_0)^2}\right) bks(1-b)}{(1-r_{RH}b - r_0 + br_0)^2}.$$

B. Sketch of Proofs

We provide here the sketch of proofs.

Proof of Proposition 2

The goal is to maximize $u_p = p(1-s)$ on s . We plug each region's optimal price and constraints in Proposition 1 into it and derive each region's optimal u_p^* using Lagrange method. We have, explicitly,

$$\max_s u_p = \frac{b(1-r_R)}{k(1-b)}(1-s) \text{ s.t } h \leq h_1 s \quad , \quad \max_s u_p = \left(\frac{1}{2} \frac{sb^2}{h} + \frac{b(-r_R + r_0)}{k} \right) (1-s) \text{ s.t } h_2 s \geq h > h_1 s \quad \text{and}$$

$$\max_s u_p = \frac{b(1-r_R)}{k(1-b)}(1-s) \text{ s.t } h \leq h_3 s. \text{ Comparing } u_p^* \text{ in different regions, we obtain the platform owner's}$$

optimal choice of sharing percentage as presented in Proposition 2.

Proof of Corollaries 3.1 and 3.2

The results regarding p_1^{A*} and p_3^{A*} are straightforward to prove. We find that $f_1 = h \cdot (\partial p_2^{A*} / \partial k_U)$ is a linear function of h , $\partial f_1 / \partial h > 0$ and $f_1(0) > 0$. So $\partial p_2^{A*} / \partial k_U > 0$. Meanwhile, we consider $\partial p_2^{A*} / \partial k_R$ as

a function of h and then $\partial p_2^{A^*} / \partial k_R < 0$ is guaranteed by $h < \frac{bk_U^2 s}{(r_R - r_0)(k_R - bk_R + bk_U)}$. It is satisfied

when $h < h_2^A s$.

Proof of Corollary 3.5

One can find that $f_2(b) = b \cdot (\partial s_1^{A^*} / \partial k_U)$ is a linear function of b , $f_2(0) < 0$ and $f_2(1) < 0$. So, $\partial s_1^{A^*} / \partial k_U < 0$. Under our assumptions that $r_R > r_0, k_R > k_U$, $f_3(b) = bh^{-1}k_U^2 k_R^2 (-1+b) \cdot (\partial s_1^{A^*} / \partial k_R)$ is a concave quadratic function of b . We find that $f_3(1) > 0$ and $f_3(0) < 0$. It can be shown that when $r_R \geq \hat{r}_R = (r_0 + 3) / 4$, $f_3(b_1^A) \geq 0$. Its root properties in different cases which, when turned into inequalities of k_R , yield this corollary.

Proof of Corollary 3.6

In squeezing region, $f_4(k_R) = k_R \cdot (\partial u_p^* / \partial k_U)$ is a linear function of k_R , $\partial f_4 / \partial k_R > 0$ and $f_4(k_R)|_{k_R=k_U} > 0$. So $\partial u_p^* / \partial k_U > 0$. We find $f_5(h) = \partial u_p^* / \partial k_R$ is a linear function of h , $f_5(0) < 0$ and $f_5(h_{1\alpha}^A) < 0$. So $\partial u_p^* / \partial k_R < 0$. We find $f_6(h) = -k_R^2(1-b)^2 k_U^2 \cdot (\partial u_p^* / \partial r_R)$ is linear function of h and $\partial u_p^* / \partial r_R < 0$ can be shown by $f_6(0) > 0$ and $f_6(h_{1\alpha}^A) > 0$. $\partial u_p^* / \partial r_0 > 0$ is straightforward to prove. The results in other regions can be shown similarly.

Proof of Corollaries 4.1 and 4.2

In encouragement region, $f_7(k_R) = -((1-b)k_R + bk_U) \cdot (\partial W / \partial r_R)$ is a linear function of k_R . $\partial f_7 / \partial k_R < 0$ and $f_7(k_R)|_{k_R=k_U} < 0$ can be guaranteed when $h < bk_U / (r_R - r_0)$, which is satisfied when

$h < 2h_2^A / 3$. So $\partial W / \partial r_R > 0$. We find that $\frac{\partial W}{\partial r_0} = \frac{1}{4} \frac{f_7(k_R)}{(k_R - bk_R + bk_U)k_U^2} < 0$ and

$\frac{\partial W}{\partial k_R} = \frac{1}{8} \frac{b(1-b)f_7(k_R)}{h(bk_U + k_R - bk_R)^3} < 0$. Define that $h \cdot (\partial W / \partial k_U) = -f_7(k_R)f_8(h)$ where $f_8(h)$ is a linear

function of h with a negative slope and root at $h_{2\alpha}^A$. It is not difficult to show that $h_{1\alpha}^A < h_{2\alpha}^A \leq 2h_2^A / 3$.

The results in retention region are straightforward to prove.

Proof of Corollary 4.3

In squeezing region, $\partial W / \partial x$ where $x = r_R, r_0, k_R, k_U$ can be written in the format of $C_x f_9(h)$ where C_x is a positive term without h and $f_9(h)$ is a linear function of h . $f_9(0) < 0$ is not hard to show. $f_9(h_{1\alpha}^A)$ can be written as a function of k_R , and $f_9(h_{1\alpha}^A) < 0$ under our assumption that $k_R > k_U$. So $\partial W / \partial x < 0$.

Proof of Lemma 5.1 and Proposition 3

We have $\sup\{p_2^*\} = p_2^*|_{(h=h_1s)} = \frac{b(1-r_R)}{k(1-b)} = p_1^*$, $\sup\{p_3^*\} = p_3^*|_{(b=b_1)} = \frac{2r_R - r_0 - 1}{k} = \inf\{p_2^*\} = p_2^*|_{(b=b_1, h=h_2s)}$.

Therefore, $p_1^* > p_2^* > p_3^*$. p_m becomes a constraint upon p_1^* , p_2^* and p_3^* . When $p_m \leq \hat{p}_{m1}$, p_m only affects p_3^* , but if $p_m > \hat{p}_{m1}$ it will affect both p_1^* and p_2^* . Adding $p \geq p_m$ as a new constraint to solve the problem in Proposition 1 using Lagrange method, we have Proposition 3.

Proof of Proposition 4

This proposition can be shown by the same approach as in the proof of Proposition 2 but changing the second stage results to those of Proposition 3. The corresponding optimization problems are

$$\max_s u_p = \frac{b(1-r_R)}{k(1-b)}(1-s) \text{ s.t } h \leq h_1s \quad , \quad \max_s u_p = \left(\frac{1}{2} \frac{sb^2}{h} + \frac{b(-r_R + r_0)}{k} \right) (1-s) \text{ s.t } h_{2\alpha}^M s \geq h > h_1s \quad , \quad \text{and}$$

$$\max_s u_p = p_m(1-s) \text{ s.t } h_{2\alpha}^M s < h \leq h_2^M s .$$

Proof of Proposition 5

The platform owner's revenue is $u_p = (1-b_1)(u_{p1} + u_{p2}) + b_1 u_{p3}$ where $u_{p1} = \int_0^{h_1s} p_1^*(1-s) dh$, $u_{p2} = \int_{h_1s}^{h_2s} p_2^*(1-s) dh$ and $u_{p3} = \int_0^{h_3s} p_3^*(1-s) dh$. Solving $\partial u_p^* / \partial s = 0$, we obtain $s^* = 1/2$.

Proof of Proposition 6

s_L, s_H are optimal s^* in its corresponding region so that individual rationality constraints are satisfied. To

check incentive compatibility, we define $f_L(r, s) = \frac{b(r-r_R)s}{k(1-b)} - \frac{h_L(r+br_0-br_R-r_0)^2}{k^2(1-b)^2}$. It can be shown

that $\partial f_L / \partial r > 0$. Then, the incentive compatibility constraint is $f_L(1, s_L) \geq f_L(\hat{r}, s_H)$. To show that it is

also satisfied, we define $F(h_H, h_L) = 16h_H^2 k^2 (1-b)^2 (f_L(1, s_L) - f_L(\hat{r}, s_H))$ in which

$\hat{r} = r_0 - \frac{(r_R - r_0)(1+b)}{2} - \frac{1}{4} \frac{bk(1-b)}{h_H}$. We have $\partial F / \partial h_L > 0$ and $\partial F / \partial h_H > 0$. The second order condition

is not difficult to verify so we have $\inf \{F(h_H, h_L)\} = F(h_{1\alpha}, 0)$. It can be shown that $F(h_{1\alpha}, 0) > 0$.

Proof of Proposition A3

When two groups of reservation ratings r_0 and r_{RH} exist, developers have two scenarios. Scenario 1 is to only satisfy the r_0 consumers and obtain n_{RL} fraction of demand. Scenario 2 is to satisfy the r_{RH} consumers to obtain all the demand. We follow the similar procedure as Proposition 1 using Lagrange method. Comparing and finding the overall optimal solutions for both scenarios yield this proposition.

Proof of Proposition 7

To attain the equilibrium in Region A3-2, which is $s_2^{E*} = \frac{1}{2} + \frac{h(r_{RH} - r_0)}{kb}$, the cost rate h must satisfy

$h \leq ks_{H2}^* (1 - n_{RL}^2) b / (4r_{RH} - 4r_0)$, which requires $h \leq h_2^{WW}$. To attain the equilibrium in Region A3-3,

which is $s_1^{E*} = \frac{1}{2}$, we need to have $h > ks_{L2}^* (1 - n_{RL}^2) b / (4r_{RH} - 4r_0)$, which requires $h > h_1^{WW}$. We can

further show that $h_2^{WW} - h_1^{WW} > 0$ therefore $H^{WW} = [h_1^{WW}, h_2^{WW}]$ exists. In H^{WW} , we find that

$f_{10}(h) = u_{PH2}^* - u_{PL2}^*$ is quadratic function of h . $f_{10}(h) > 0$ when $h < h' = \frac{1}{2} \frac{(1 - \sqrt{n_{RL}}) kb}{r_{RH} - r_0}$. It is not

difficult to show $h_2^{WW} - h' > 0$ when $n_{RL} \in \left(0, \frac{1 - r_{RH}}{1 - r_0}\right]$. So we have $u_{PH2}^* > u_{PL2}^*$. The result that

$u_{DH2}^* > u_{DL2}^*$ can be shown with the same procedure. Thus, the ‘‘Win-Win’’ effect is identified.