Should I Diversify My Mobile App Portfolio? Examining the Effects of App Portfolio Size and Diversity on App Quality and Popularity [1]

(Full Paper, 12,988 Words)

Mei Li
National University of Singapore
limei@comp.nus.edu.sg

Khim-Yong Goh
National University of Singapore
gohky@comp.nus.edu.sg

Huseyin Cavusoglu
University of Texas at Dallas
huseyin@utdallas.edu

ABSTRACT

A critical challenge faced by mobile app developers today is to effectively manage their app portfolio, but this issue has rarely been addressed in the IS academic literature. To address this gap, we focus on the impact of category assortment of developers' app portfolios on their performance. Specifically, we evaluate both the supply and demand of mobile apps with a data set from the Apple App Store. On the supply side, we examine the influence of mobile developers' app portfolio size and diversity on app quality. A panel-level linear model estimation augmented with two identification strategies of Granger causality test and propensity score matching shows a negative impact of portfolio diversity on developers' app quality. However, this negative influence is mitigated by the increasing size of app portfolio. On the demand side, we assess the extent and direction of popularity spillover effects between developers' existing and new apps. Our empirical analysis with a simultaneous equations model shows that popular existing apps of a developer can promote the popularity of new apps in the same category. New apps, in turn, drive demand for a developer's existing apps both within and across categories. Our findings highlight the important role of learning processes in the product portfolio management for human capital intensive industries.

Keywords: product portfolio management, mobile app industry, software developers, portfolio diversity, spillover effect

[1] This study is part of Mei Li's doctoral dissertation and the bulk of the work was done by her.

1. INTRODUCTION

The introduction of mobile smart phones has radically altered the way people use communication and information technology devices. As a complementary product to these devices, mobile apps have gained much attention from both consumers and developers. The US Apple App Store alone has already attracted about 57 million consumers (comScore 2013) and more than 220,000 active developers[2] (148Apps.Biz 2013). Apple (2013) announced its 50 billion accumulated downloads last March, demonstrating the enormous market potential for mobile apps. Moreover, the revenue from mobile apps globally is estimated to soar to US$46 billion by 2016 (CNET 2012). Contrary to the purported lucrative sales in the mobile apps industry, a majority of app developers earn little revenue due to stiff competition in the app markets. A more sobering fact is that most developers lack effective marketing plans and seldom manage their app portfolio strategically (Bergvall-Kåreborn and Howcroft 2011).

A product portfolio refers to an assortment of products or services offered by firms. Product portfolio management plays a critical role in firms' strategic management, including that for mobile app developers. Mobile app stores typically consist of many different app categories such as Books, Games and Navigation. Each category distinguishes itself from others by its functionality and content. App developers have broad flexibilities in choosing the number of apps to develop and the category of apps to specialize in or diversify to. However, most developers in reality rarely plan or manage their app portfolio strategically with market research. Oftentimes, they build apps they like, with the assumption that mobile users will have an affinity for their apps, which often may not be the case (VisionMobile 2013).

In spite of the proliferation of studies on product portfolio management (e.g., Berry 1971; Shankar 2006), the existing ones can hardly provide references for mobile app developers. Unlike the traditional manufacturing industry where physical capital assets are important production factors (Gallivan et al. 2004), human capital plays an influential role in the mobile app industry (Boh et al. 2007). Moreover, mobile platforms have greatly reduced developers' entry barriers and a large proportion of app developers are now not professional software firms but small-scale teams or individuals (Qiu et al. 2011). With less need for a large capital investment and a highly efficient online distribution channel, the mobile app industry is characterized as a market with extremely intensive competition and short innovation cycles (Han and Ghose 2012).

[2] Unless we specifically point out, a developer in this study refers to a publisher of mobile apps, either as an individual person or a software firm.

Despite the increased importance of mobile app industry, there is limited research in the literature. The question of how to manage product portfolio appropriately in such a hypercompetitive environment of the mobile app industry remains unanswered. Our study aims to fill this gap by examining the impact of product portfolio management on supply as well as demand of mobile apps. With respect to product portfolio management, we focus on the category assortment of developers' app portfolio. On the supply side, the scope and concentration of mobile developers' app portfolio across different app categories may influence their app production. A more focused app portfolio helps developers accumulate experience in a certain area and fosters core competency, whereas a more diversified app portfolio allows developers to cater to users with different preferences or needs, although this may require more attention switching between various apps. On the demand side, consumers may infer the quality of a developer's new app from the market response to the developer's existing apps. Simultaneously, the response for this new app may also affect the popularity of the developer's existing apps. In addition, this bidirectional influence may be moderated by the category relatedness[3] between the existing apps and the new app. Therefore, we propose the following research questions in this paper:

(1) How do the size and diversity of mobile app developers' app portfolio influence their app quality?
(2) How, and to what extent, do a developer's new app and existing apps, in either the same or different category, influence each other's popularity?

To answer these questions, we collected a comprehensive data set from the Apple App Store. The data contains information about all mobile apps that were displayed in the store in 2011. For the supply side analysis, we observed monthly app portfolio size and portfolio diversity changes for each developer, in order to examine the effects of category assortment on the quality of apps produced. We used the monthly average user rating valence of apps published by each developer to measure app quality. Our analysis, based on the estimation of a panel-level linear model, shows that app portfolio diversity negatively influences developers' app quality. This effect however decreases with developers' app portfolio size. To support a causal interpretation of our results, we further used two identification strategies of Granger causality test and propensity score matching to reaffirm our findings. Our results imply that while an app portfolio without appropriate specialization is detrimental to developers' performance, an increasing app development experience could mitigate this impact.

[3] Category relatedness refers to whether the existing apps are in the same category as the new app.

For the demand side analysis, we used a simultaneous equations model to evaluate how mobile app category assortment influences the popularity of a developer's existing apps and new apps. Our results suggest that there is a positive popularity spillover from existing apps to new apps within the same category. This spillover effect is not observed when apps are in different categories. Interestingly, the popularity of a developer's new app greatly boosts the popularity of its existing apps regardless of the category relatedness between apps. This spillover effect is also larger in magnitude compared to the effect from existing apps to a new app. Thus, our results imply that, apart from creating fresh revenue streams for developers, new apps serve as a catalyst to induce consumers to explore and purchase developers' existing apps.

Overall, this study makes several significant contributions. First, it expands the study of product portfolio management from traditional industries, where physical capital assets are important production factors, to a human capital intensive industry, namely the emerging mobile app industry. Second, we provide evidence that achieving an alignment between category diversity and development experience through learning is highly crucial for an effective portfolio management strategy in the app market. In particular, we show that developing mobile apps in limited app categories initially facilitates developers' early exploitational learning. Based on substantial knowledge gained in the early exploitation phase in an app category, it is easier for developers to diversify into new categories later. Thus, the negative impact of a diversified portfolio is mitigated through experience over time. Third, our work investigates how app portfolio management influences the branding effect in terms of a developer's' name or identity on user response. Although the popularity of a particular developer's existing and new apps promotes each other simultaneously, such positive spillovers are more prominent within the same category than across different categories. Fourth, we also provide practical suggestions for mobile app developers, who may benefit from a well-planned app portfolio.

2. THEORY AND HYPOTHESES DEVELOPMENT

The focus of this study is to investigate the impact of category assortment on mobile app supply and demand. On the supply side, app quality is a critical metric to predict the success of an app, since product quality directly influences consumers' willingness to pay (Bhargava and Choudhary 2001; Dellarocas 2003). Hence, we discuss the influence of category assortment on the quality of apps produced by the developers in the first part of our theory development.

A major research context of the product portfolio management literature is the traditional manufacturing industry that is mainly characterized by physical capital assets. The IT industry, however, has long been recognized as human capital intensive (Ang et al. 2002; Levina and Xin 2007). Labor forces are soft assets of IT firms and the capabilities of the human assets are influential in shaping firms' performance (Banker et al. 2008). As the mobile app industry is relatively new, most developers are still in the process of learning the intricate dynamics of this industry. A majority of experience gained by developers, whether on technical or managerial aspects, is accumulated in the process of learning by doing (Argote and Miron-Spektor 2011; Boh et al. 2007; Mukhopadhyay et al. 2011; Singh et al. 2011). As such, the accumulation of development experience creates a valuable knowledge repository that can provide reference and guidance for future work.

Every mobile platform has its own software development kit (SDK). Developers need to familiarize themselves with the unique syntax, commands and libraries of the SDK to succeed in this new environment. To many developers, the main vehicle through which learning on development tasks, including framework design, code reuse and bug detection, takes place is the process of publishing apps. This type of learning occurs regardless of the relatedness between the new and existing apps since the underlying architecture and process of app development are similar. An increasing number of apps provide developers with more opportunities to interact with customers and better understand demand dynamics in terms of consumer preferences and needs. Such accruing experiences increase developers' human capital, which in turn improves the quality of the apps they produce. A developer's app portfolio size, defined as the number of apps the developer currently has, reflects its extent of experience in app development. Consistent with the arguments above, we expect a positive impact of mobile app portfolio size on app quality. Hence, we posit that:

Hypothesis 1 (H1): The size of a developer's app portfolio positively influences the developer's app quality.

The diversity of a developer's app portfolio characterizes the extent of the developer's production concentration over app categories. A higher diversity indicates a less concentrated app portfolio. The Apple App Store has 23 major categories, ranging from spare time killers to productivity improvement tools. Each category distinguishes itself from others by its functionality and content. From the developers' perspective, programing skills and other resources required for development of apps in each category may vary. For example, apps in the category of Navigation need a strong support from the server end since frequent changes in maps should be updated frequently to the client's end via the server. In contrast, apps in the category of Games may not need much server-end programming, but probably require sophisticated designs in user interface and game play. In spite of this, developers may reuse code components for some common functions (Haefliger et al. 2008), such as camera access and data storage. However, products in the same category have more features in common than those from different categories (Rosch and Mervis 1975). Compared with apps from different categories, apps in the same category may be able to share more common components due to their greater similarity. In contrast, developing apps for multiple categories can bring about diversification to the app portfolio. Users install apps from different categories according to their own needs and app categories act as natural dividers of consumer segments. Releasing apps in multiple categories helps developers to serve more user segments. Thus, concentration of apps across different app categories can influence the success of mobile app developers.

Previous studies have reached dichotomous conclusions about the impact of a diversified product portfolio on firms' performance. The advocates of diversification contend that a shared sunk cost can reduce the total cost for firms (Bailey and Friedlaender 1982) and diversification helps reduce business risks (Cardozo and Smith Jr 1983). On the contrary, the opponents of diversification argue that firms with a diversified product portfolio disperse their limited resources to multiple areas. As a result, they can hardly foster firm-specific core competency, consequently resulting in a poorer performance (Montgomery and Wernerfelt 1988).

In the mobile app market, most developers are typically small entrepreneurs or startups, who may lack financial assets or other valuable resources (Aggarwal et al. 2012; Qiu et al. 2011). Making full and efficient use of their production capacity is their primary goal because dealing with apps from many different categories necessitates the development of various functional components (Haefliger et al. 2008). Compared with developers who are more focused, developers with a diverse portfolio may encounter more problems. To resolve these problems, they may not have time to gain proficiency with the old functional components. The frequent attention switching renders it difficult for them to obtain an in-depth understanding about a specific domain (Schilling et al. 2003). Hence, developers with a diverse portfolio have to distribute their efforts to a wide range of category areas. This effort distribution, in turn, drains their limited resources, leaving little space for them to build core competency and improve their app quality. As a result, a diverse app portfolio may adversely impact developers' app quality. Therefore, we hypothesize that:

Hypothesis 2a (H2a): The diversity of a developer's app portfolio negatively influences the developer's app quality.

Developers continually learn from previous development experience through *exploitation* and *exploration* (Durcikova et al. 2011). Exploitation refers to "learning gained via local search, experiential refinement, and selection and reuse of existing routines" whereas exploration refers to “learning gained through processes of concerted variation, planned experimentation, and play” (Baum et al. 2000). Exploitation emphasizes the improvement based on existing components and skills, which frequently accumulate in a routinized work. Exploration focuses more on the innovativeness of the task, involving a move to different technological activities (Benner and Tushman 2002). Exploitation and exploration are not independent learning processes. While exploitation involves reuse and refinement of past knowledge, oftentimes, firms not only transfer knowledge to new problems but also learn the methods to learn (Schilling et al. 2003). The learning ability accumulated in exploitational learning sometimes even entail new understanding and solutions to new areas not encountered before, and thus subsequently facilitating explorational learning (Garud and Kumaraswamy 2005).

Developers may face both exploitation and exploration in app development. In early stages, app developers are likely to be unfamiliar with app development processes and requirements due to a lack of experience. Developing similar apps initially helps them resolve similar problems encountered frequently and be acquainted with the general development process. Developers can tap into exploitational learning by reusing and making refinement of existing apps. After sufficient exploitation, app developers can digest wide-ranging development procedures and therefore are better positioned to apply their extensive knowledge to specialized development of apps in different categories. Hence, the experience accumulated through previous exploitational learning may ease the process of exploration later on. Although apps from different categories may share fewer common components, developers are still able to transfer accrued knowledge to new app categories. Therefore, a diversified portfolio may bring positive returns only after sufficient exploitational efforts have been exerted. That is, development experience, reflected by the size of the app portfolio, may mitigate the negative effect of portfolio diversity on developers’ app quality due to developers’ transfer learning. Hence, we posit that:

Hypothesis 2b (H2b): The negative effect of the diversity of a developer’s app portfolio on the developer’s app quality decreases with the size of the app portfolio.

On the demand side, consumers are often imperfectly informed about the quality of products, especially for experience goods (Nelson 1970). As mobile apps are experience goods, individuals’ valuations of an app may vary substantially due to quality uncertainty and heterogeneity in preferences. In assessing experience goods, prospective consumers are likely to be influenced by a product’s rating volume, in addition to rating scores given by individual consumers (Liu 2006). Thus, we are interested in evaluating how the app popularity, as measured by rating volume, of existing apps impacts the popularity of a new app, and whether this popularity externality is mutual (i.e., from a new app to existing apps as well).

The brand management literature (e.g., Montgomery and Wernerfelt 1992; Wernerfelt 1988) contends that consumers' prior use experience or valuation of existing products influences their quality perception of a new product under the same brand. In mobile app stores, developers can release apps in various categories. While apps from different categories are designed to satisfy consumers' different needs, apps in the same category provide common functionalities and therefore fulfill similar needs of users, such as entertainment in the Games category. Having a well-received app signals a developer's ability to develop high-quality apps. Since the apps in the same category share more common features, this quality inference is stronger for a new app within the same category, compared to one belonging to a different category. Specifically, high standings of a developer's existing apps that are in the same category as the new app conveys a message to consumers that the developer is capable of producing superior apps in that category. For apps across different categories, the transfer of quality perception may not be that salient due to the differences in app functionality and design. Thus, we hypothesize that:

Hypothesis 3a (H3a): The popularity of a developer's existing apps positively influences the popularity of its new app releases within the same category.

Hypothesis 3b (H3b): The popularity of a developer's existing apps does not influence the popularity of its new app releases across different categories.

Besides the quality perception transfer from existing products to new products (Wernerfelt 1988), usage experience and reputation of new products, in turn, influence consumers' valuation of existing products (Erdem 1998). This interplay also holds for mobile apps. Thus, the popularity of new apps and existing apps may influence each other simultaneously. We have discussed the popularity spillover from existing apps to a new app in H3a and H3b. At the same time, the popularity of the new app may influence that of existing apps. This effect can materialize via two mechanisms. The first one is through the quality signaling effect of the new app. Existing studies that assess the impact of piracy on the sales of albums find that music piracy, in certain degree, boosts the music sales in legal distribution channels, since pirated music provides consumers a convenient way to sample the artists' music before purchase (Gopal et al. 2006; Hui and Png 2003; Sundararajan 2004). Applying this rationale to the mobile app market, we argue that once consumers have a positive experience with a new app, they may transfer this favorable evaluation to the existing apps of identical or different categories by the same developer. In this case, the usage of the new app acts as a mode of sampling the developer's existing apps. Moreover, consumers' positive evaluation towards the new app may also influence other consumers' quality perception about the developer due to the word-of-mouth effect (Chevalier and Mayzlin 2006; Godes and Mayzlin 2009).

The second mechanism is through the new app's role as a discovery facilitator. A prominent feature of the major mobile app markets, including Apple App Store, is the provision of various ranking charts to help consumers explore the app market and find relevant apps to purchase. Among these, charts specific to newly released apps showcase recently developed apps in each category. The incidence of being displayed in these charts greatly increases the visibility of not only the new app but also the developer, thereby making it easier for consumers to notice other apps by the same developer. Moreover, with the prevalent cross-advertising design[4] of mobile apps, developers also have opportunities to market their existing apps along with new app releases. The prominence of a new app increases the exposure of the existing apps, which are cross-advertised in the focal new app. In addition, developers are able to choose any app from existing apps, either apps within the same category or apps across different categories, to cross-advertise. Thus, through both quality signaling and discovery facilitator effects, a popular new app is able to redirect consumers' attention to other apps by the same developer regardless of their category relatedness. Hence, we hypothesize that:

Hypothesis 4a (H4a): The popularity of a developer's new app releases positively influences the popularity of its existing apps within the same category.

Hypothesis 4b (H4b): The popularity of a developer's new app releases positively influences the popularity of its existing apps across different categories.

3. METHODOLOGY AND RESULTS

3.1. Data Description

To empirically test our hypotheses, we obtained data on mobile apps in the Apple App Store from Mobilewalla (Datta et al. 2012). This data set contained both time-invariant and time-varying information on all mobile apps that were released in the Apple App Store in 2011. Time-invariant information of each app includes the name of the app, developer identity, release date and category classification. We also have user review ratings of each app on a daily basis, which are time-varying.

[4] As the major app stores do not allow in-store advertisements, most developers embed links in their apps to advertise other apps, if any, within their portfolio.

3.2. Supply Side Analysis

Variable Definition and Model Specification

Since our supply-side research hypotheses evaluate the impact of app portfolio size and diversity on developers' app quality, the corresponding unit of analysis is at the developer level. Prior research points out that many amateur developers publish apps on mobile platforms only for fun (Qiu et al. 2011). This observation may confound the objective of our study, which is to provide insights for developers who view creating mobile apps as a business. Therefore, we excluded developers who had less than four[5] apps at the end of our observation period. The reason is that developers with an app portfolio of a moderate to large size are likely to be professional developers. Another reason is that it is pointless to analyze the issue of portfolio management if a developer has only a few apps. Finally, we excluded developers who entered the Apple App Store before 2011 since we did not have complete information about them[6].

Since our analysis aims to uncover the impact of app portfolio size and diversity on developers' app quality, we believe the time granularity of one month is appropriate to capture the change in developers' app portfolio. Hence, we clustered apps by developers and aggregated each developer's app release history to a monthly level. From the data, we can observe developers' monthly app portfolio composition as well as the user rating information of each app. In total, our panel-level (unbalanced) data set consists of 92,018 observations of 11,579 developers, each with about 8 observations on average.

We used the valence of user ratings (i.e., rating score) to measure app quality (Liu 2006). In the Apple App Store, users are asked to evaluate an app by assigning a score from 0 to 5, with an increment of 1. As there are multiple apps in a developer's app portfolio, we averaged the values of user rating valence across apps and used it as our dependent variable (DV). Our DV, hence, is a real number between 0 and 5.

The key independent variables (IVs) are the size and diversity of an app portfolio. The size of an app portfolio is measured as the cumulative number of apps released by a developer in a given month. To measure the diversity of an app portfolio, we used entropy (as shown in Equation (1) and Equation (2)), which is more sensitive to small changes than other alternative measurements and is recommended in prior studies (Jacquemin and Berry 1979). The Apple App Store allows an app to be classified under multiple categories. If an app belongs to more

[5] We also conducted the empirical analyses with alternative cutoffs of 3 and 5 apps. Our results did not change qualitatively.

[6] As a result, our sample contains information about developers who entered the Apple App Store in 2011 and had no less than 4 apps at the end of December 2011. Our observation period covers a whole year and we are able to capture different segments of developers because the launch of the much anticipated iPhone 4 in late 2010 greatly invigorated the supply of mobile developers' output of apps in the market. Many small- to mid-size developers joined the Apple App Store in 2011.

than one category, its production may need development skills for different functions. We use app-category combinations to measure a developer's effort distribution. We define each distinct pair of category and app in a developer's app portfolio as a combination. P_s captures the ratio of the number of combinations that belongs to category s to all the combinations in the developer's app portfolio. The entropy is the sum of ($-P_s * lnP_s$) over the non-empty categories.

$$CategoryEntropy = -\sum_{s \in Z} \left(P_s * lnP_s \right) \tag{1}$$

$$P_s = \sum_{j \in AP} 1(s \in C_j) / \sum_{j \in AP} count(C_j) \tag{2}$$

where Z is the predefined app category set in the Apple App Store, AP is the app portfolio the developer owns. C_j is the set of categories app j belongs to, $count(C_j)$ denotes the number of categories app j belongs to. $\mathbf{1}(s \in C_j)$ equals 1 if app j belongs to category s, and 0 otherwise.

To illustrate the calculation of diversity, consider a developer that has two apps, namely App A and App B. Suppose that App A belongs to Category 1 and Category 2, while App B only belongs to Category 2. We can show that 1/3 of this developer's work is related to Category 1, i.e., $P_{Category1}$ = 1/3, and the other 2/3 is related to Category 2, i.e., $P_{Category2}$ = 2/3. The entropy of this developer's app portfolio is thus – ($P_{Category1}*ln(P_{Category1})$ + $P_{Category2}*ln(P_{Category2})$) = 0.64.

Our econometric model is shown in Equation (3), where subscript i denotes developer and subscript t denotes month. Variable definitions and operationalizations are given in Table 1. β's are the model coefficients, α_i's capture developers' unobserved heterogeneity and $\varepsilon_{i,t}$'s are residual errors with standard assumptions. The changes in app portfolio may take time to affect developers' app quality. Hence, all IVs are lagged by one month[7].

$$\begin{aligned} AvgRatingValence_{it} = {} & \alpha_i + \beta_1 APSize_{i,t\text{-}1} + \beta_2 APDiversity_{i,t\text{-}1} + \beta_3 APDiversity_{i,t\text{-}1} * APSize_{i,t\text{-}1} \\ & + \beta_4 APDiversity_{i,t\text{-}1} * Company_i + \beta_5 FreeRatio_{i,t\text{-}1} + \beta_6 AvgPrice_{i,t\text{-}1} \\ & + \beta_7 StdPrice_{i,t\text{-}1} + \beta_8 PromotionRatio_{i,t\text{-}1} + \beta_9 AvgVersionNum_{i,t\text{-}1} + \beta_{10} Tenure_{i,t\text{-}1} \\ & + \beta_{11} iOSNewApps_{t\text{-}1} + \beta_{12} iOSPromotionApps_{t\text{-}1} + \beta_{13} iOSAppNum_{t\text{-}1} \\ & + \gamma CategoryIndicators + \varepsilon_{i,t} \end{aligned} \tag{3}$$

[Insert Table 1 Here]

To measure the moderating effect of portfolio size on diversity, we use an interaction term between app portfolio diversity and portfolio size. In addition, we account for other factors that may influence developers' app quality. These control factors are categorized into two groups: *developer-level heterogeneities* (e.g., developer's tenure, average price of developer's apps)

[7] We estimated the model with contemporaneous values of the IVs, which generated similar results except for a significantly negative coefficient for *APSize*. We also tried models with other lags (e.g., 2 and 3 months), but most coefficients were insignificant, indicating that developers' app quality in our sample likely reflected the market changes in the previous month.

and *platform-level competition factors* (e.g., numbers of existing and new apps on the platform). Since not all factors that influence developers' performance can be observed and therefore controlled by us, we capture the impact of unobservable factors using developer fixed effects in our econometric model. We include a company indicator to control for the difference between apps by professional companies and individual developers. Since this indicator is time invariant, its main effect will be absorbed by the developer fixed effects. Thus, we add an interaction term for company indicator and portfolio diversity, to capture potential differences between company developers' portfolio diversity and individual developers' portfolio diversity. In addition, we also include category indicators to capture any effects specific to categories in which developers have released apps.

Results and Discussion

The descriptive statistics of all variables are shown in Table 2. The correlation matrix in Table 3 shows three highly correlated variable pairs, namely, *AvgPrice* and *StdPrice*, *Tenure* and *iOSAppNum*, *iOSNewApps* and *iOSAppNum*. In the next paragraph, we explain how we dealt with potential multicollinearity issues in our model estimation.

[Insert Table 2 and Table 3 Here]

Since some of the explanatory variables are cumulative measures (e.g., *APSize* and *AvgVersionNum*), the error term in our model may be serially correlated. In order to test for serial correlation, we followed the approach proposed by Wooldridge (2002) and Drukker (2003). This test rejected the null hypothesis of no serial correlation at the significance level of 0.05. A panel-level ordinary least square estimator with a first-order autoregressive disturbance structure was thus utilized to estimate our econometric model. We first estimated the model without the interaction term between portfolio size and diversity. The results[8] of fixed-effects (FE) model and random-effects (RE) model are reported in columns (1) and (2) of Table 4. The Hausman specification test indicated that the FE model is appropriate (χ^2 = 1356.51, p = 0.000), and thus we focused on the results of FE model to interpret the findings and to test our hypotheses. As both *APSize* and *APDiversity* were significant, we further investigated the moderating effect by incorporating the interaction term. Columns (3) and (4) show the results of the FE model and RE model with the interaction term included. The Hausman specification test again revealed that the FE model is appropriate (χ^2 = 5682.91, p = 0.000). Hence, we interpreted the findings based on the results in column (3).

[8] We initially included all variables in our model estimation. However, estimated coefficients for *AvgPrice* and *StdPrice* were not significant when both were included concurrently. Similarly, coefficients for both *iOSNewApps* and *iOSAppNum* were not significant. To avoid multicollinearity, we dropped *StdPrice* and *iOSNewApps*.

[Insert Table 4 Here]

The coefficient of *APSize* is statistically insignificant in column (3), indicating that app portfolio size does not influence developers' app quality. Thus, H1 is not supported. The coefficient of *APDiversity* is negative and significantly different from zero, which suggests that app portfolio diversity negatively affects app quality, supporting H2a. The interaction term between *APDiversity* and *APSize* is significantly positive. It implies that the negative impact of app portfolio diversity on developers' app quality decreases with app portfolio size. Hence, H2b is also supported. Combining the coefficients of *APDiversity* and *APDiversity**APSize*, we can calculate the net marginal effect of app diversity on developers' app quality as (-0.0258+0.0016**APSize*). Hence, there is a critical inflection point around 16 (i.e., 0.0258/0.0016) in *APSize*. This implies that, on average, when a developer's total app portfolio has less than 16 apps, increasing app diversity in the portfolio brings about poorer quality performance for developers. However, when a developer's total app portfolio is 16 or larger, developers could benefit from a more diversified app portfolio.

There is no significant difference of the portfolio diversity impact between company-based developers and individual developers (since the coefficient of *APDiversity*Company* is insignificant). *FreeRatio* has a significantly positive coefficient, suggesting that developers who have a portfolio with a larger proportion of free apps have higher quality in terms of rating scores. The positive coefficient of *Tenure* indicates that developers with longer tenures on the mobile platform produce better apps compared to the newcomers. Moreover, *iOSAppNum*, which measures the competition intensity in the Apple App Store, is negatively associated with developers' app quality. A possible reason is that as the number of mobile apps grows on the platform, consumers are endowed with more choices, and have more opportunities to compare and contrast apps with other available alternatives. Therefore, consumers are likely to become increasingly discriminating and this is being reflected in their weaker evaluations of the apps. That is, keeping other factors constant, they tend to give lower ratings to apps when more apps are available in the market.

As a robustness check, we used alternative measurement methods, i.e., the *adapted Herfindal-index* (Jacquemin and Berry 1979) and *category count*, to measure the diversity of an app portfolio. Then, we re-estimated our model in Equation (3) using these alternative measures of diversity. The results are presented in Table 5. We calculated the *adapted Herfindal-index* (i.e., *CatHHI*) using the formula in Equation (4). The estimation results with *CatHHI* are given in column (a) of Table 5. We can see that H2a and H2b are both supported. *Category count* (i.e., *CatCount*) refers to the total number of categories to which a developer's apps belong. The estimation results with *CatCount* as a measure of diversity is given in column (b) of Table 5. The results support not only H2a and H2b, but also H1.

$$CatHHI = 1 - \sum_{s \in Z} P_s * P_s \tag{4}$$

[Insert Table 5 Here]

Identification Strategies

Our results in the previous section have provided some evidence of the effect of portfolio diversity on developers' app quality. Although lagged measures of portfolio diversity can help to mitigate potential endogeneity, it may be difficult to establish causality. The observational data we use makes it challenging to establish causal effects in a strict sense. Nevertheless, econometric tests and methods provide us a way to evaluate the existence of causal effects.

The first test we employed is the Granger causality test (Dutta 2001; Granger 1969), based on an autoregressive regression of time series data. The time variable was defined as a developer's tenure (in months) in the Apple App Store. We computed the mean values of all developers' app quality, portfolio size and diversity for each time point to form the time series data. An autoregressive model for app quality was estimated with portfolio size and diversity as predictors. The Granger Causality Wald test showed that portfolio diversity explained a significant incremental variation of developers' app quality (χ^2 = 11.004, p = 0.004). This finding demonstrates that the time series of portfolio diversity Granger-causes the change in app quality and provides statistically significant information of future values of app quality.

Since Granger causality is merely a necessary condition for true causality, our second identification strategy is based on the propensity score matching (PSM) method (Heckman et al. 1998; Rosenbaum and Rubin 1983). Developers' diversification decisions are exhibited when they release new apps. We treat each new app release as an event, during which developers choose their own diversification strategies. Diversification here is considered as a treatment since some developers may release new apps across a wide range of categories while some developers choose to focus on a few categories they previously worked in. Subsequent to the diversification decisions, developers may have different performance outcomes in terms of app quality. If two developers, who shared similar characteristics before but adopted different diversification strategies, and had different app quality subsequently, this difference could be attributed to their diversification decision.

With such logic, we define diversification (coded as a binary variable) as releasing apps in more than two different categories[9]. The aim of PSM is to match the developer who diversified his or her portfolio with another counterpart developer who could have diversified but did not diversify the portfolio. The treatment effect could be calculated by taking the difference between these two developers' app quality. To conduct PSM, we need to identify a set of observable covariates that influence both treatment, i.e., diversification decision, and outcome, i.e., developers' app quality, simultaneously. The goal is to balance the distribution of the covariates to ensure the difference in outcomes is attributable to the treatment effect (Caliendo and Kopeinig 2008).

Developers' previous app publication experience may impact developers' diversification decision and app quality concurrently. The number of existing apps reflects developers' experience with app production and distribution, and also affects the developers' capability in app development and marketing. Experienced developers probably are more willing to diversify their app portfolio since transfer learning helps them to better handle possible different problems in new categories. Current portfolio composition conceivably influences their portfolio diversification decision as well. For example, developers who work in Games may like to release apps to other entertainment related categories, while developers who work in Navigation perhaps prefer to focus on this unique and special category. In addition, previous app performance may affect developers' diversification decision. If previous apps have not met adequate demand targets, developers may wish to release apps in new categories, aiming to expand into or serve different customer segments.

We estimated the propensity score with a Probit model, which is shown in Equations (5) through (7). *APSize* is the number of existing apps in developer *i*'s portfolio, reflecting the developer's experience. P_s has been defined in Equation (2) and captures developer *i*'s app distribution across all categories in the Apple App Store. *PrevRatingVal* and *PrevRatingVol* control for the developer *i*'s app performance one month prior to the new app release, in terms of average user rating valence and volume the developer received. Additionally, developer *i*'s tenure and type (i.e., company or individual) have been included.

$$\begin{aligned} Diversification_i^* = {} & \beta_1 APSize_i + \sum_{s \in Z} \alpha_s P_{si} + \beta_2 PrevRatingVal_i + \beta_3 PrevRatingVol_i \\ & + \beta_4 Tenure_i + \beta_5 Company_i + \varepsilon_i \end{aligned} \tag{5}$$

$Diversification_i = 1$ if $Diversification_i^* > 0$, and $Diversification = 0$ otherwise (6)

[9] The Apple App Store allows an app to be categorized in up to two categories. If developers concentrate on the categories where they released their first app, the number of categories they are working in would not exceed two. In such a case, this can be considered as non-diversification, since they focus on their initial category scope.

$$Prob(Diversification_i = 1 \mid Z_i) = \Phi(Z_i\beta),$$
$$Prob(Diversification_i = 0 \mid Z_i) = 1 - \Phi(Z_i\beta) \quad (7)$$

where Z_i is a vector of covariates that determine diversification propensity as discussed in the prior paragraph.

Table 6 shows the estimation results of propensity score analysis. Using the propensity scores, we matched the treated group with the control group using three matching estimators (i.e., 1 nearest neighbor, 3 nearest neighbors, Caliper with tolerance level 0.01). Then, we compute the average treatment effect on the treated (ATT) by comparing the app quality difference one month after[10] the new app release for each matched developer pair. Table 7 shows the results. All estimators report a significant negative treatment effect, suggesting that app portfolio diversification lowers developers' app quality. This finding is consistent with the main effect of *APDiversity* in our previous analysis (shown in Table 4), thus supporting H2a.

[Insert Table 6 and Table 7 Here]

To evaluate the moderating effect of portfolio size on diversity, we attempt to uncover a threshold value in portfolio size, above which a significant positive impact of portfolio diversity on app quality can be shown. We therefore repeated the PSM with different cutoffs of portfolio size in order to find the threshold value that applies to our sample data set. Consequently, analysis results using this approach suggest a value of 10[11] as the cut-off point. Therefore, the sample is divided into two groups by this threshold. The matching results[12] of Group 1 and Group 2 are shown in Table 7. Clearly, the treatment effect for Group 1 is negative, while for Group 2 is positive. This validates the existence of a moderating effect of portfolio size on portfolio diversity, reaffirming H2b.

3.3. Demand Side Analysis

Variable Definition and Model Specification

On the demand side, we seek to evaluate how the popularity of existing mobile apps impacts the popularity of a new app, and vice versa. Specifically, to examine the popularity spillover effects between existing apps and a new app, we classified each developer's existing apps into two groups. One group contains apps which are in the same category as the focal new

[10] We also examined app quality difference after three and six months. Similar results were obtained.
[11] This cut-off value is different from the one we obtained based on results shown in Table 4. Nevertheless, the qualitative trend of the moderating impact of portfolio size on diversity across the two empirical analysis methods is consistent.
[12] PSM defines the treatment and control groups based on observable covariates, but does not control for unobserved heterogeneities. Therefore, we resorted to the bounding approach proposed by Rosenbaum (2002) to determine how strong the potential effect of the unobservables should be in order to affect the matching inference. The results reveal that the Γ cutoff value is 1.2 (1.25)/ 1.15 (1.2) before a lower/ an upper bound of significance value reaches above 0.05 (0.10) for Group 1/ Group 2. This suggests that unobserved factors could entail a 20%-25%/ 15%-20% odds increase in diversifying a portfolio, in order to undermine the relationship that a higher level of app quality deterioration/ improvement is caused by portfolio diversification. This quantifies the degree to which our results would be biased due to unobserved factors.

app, and the other group contains apps which are in different categories. Since the popularities of the new and existing apps may simultaneously influence each other, a simultaneous equation system, consisting of equations (8a), (8b) and (8c), is proposed to evaluate the effects.

$$\begin{aligned}\ln(RatingVol_{it}) = \alpha_i + \beta_1 \ln(SameCatRatingVol_{it}) + \beta_2 \ln(DifCatRatingVol_{it}) \\ + ControlVars + \varepsilon_{it}\end{aligned} \tag{8a}$$

$$\ln(SameCatRatingVol_{it}) = \kappa_i + \theta \ln(RatingVol_{it}) + ControlVars + \nu_{it} \tag{8b}$$

$$\ln(DifCatRatingVol_{it}) = \eta_i + \varphi \ln(RatingVol_{it}) + ControlVars + \mu_{it} \tag{8c}$$

where *i* denotes the new app and *t* denotes the number of weeks after the release of app ***i***. α_i, κ_i and η_i capture the time-invariant unobserved heterogeneities of the new app, the existing apps in the same category as the new app, and the existing apps in different categories, respectively.

The popularity of a new app is measured by *RatingVol* which records the volume of user ratings for a developer's new app. *SameCatRatingVol* measures the average rating volume of the developer's existing apps in the same category as the new app. Similarly, *DifCatRatingVol* captures the average rating volume of the developer's existing apps that are in different categories. Equation (8a) models the popularity spillover from the existing apps, both within the same and across different categories, to the new app, whereas equations (8b) and (8c) specify the impact of the new app's popularity on the popularity of existing apps, again within the same and across different categories. Apart from these popularity measures, we control for a number of other relevant factors that may play a role in these relationships. We provide definitions and descriptive statistics of our variables in Table 8 and Table 9, respectively.

[Insert Table 8 and Table 9 Here]

Results and Discussion

We focused on new mobile apps released in 2011 to examine the popularity spillover effects between developers' existing and new apps. More than 200,000 new apps were released by 82,010 unique developers in 2011. From these large numbers of new apps and developers, we drew a random sample of 3,396 unique developers, which is about 4.1% of the entire developer population[13]. Based on these developers, we analyzed the popularity of their new apps in the first four weeks after release. Performance in the early period after launch has been recognized as a critical period for the success of a new app (Sangaralingam et al. 2012). As the popularity of existing and new apps influences each other simultaneously, the three-stage least

[13] We also used another random sample of 5,000 developers for the empirical analysis. The model estimation results obtained using this alternative sample were consistent with those reported below and in Table 10.

squares estimator was used to estimate the system of equations in (8a), (8b) and (8c). The model estimation results are shown in Table 10.

[Insert Table 10 Here]

The coefficient of *SameCatRatingVol* is significantly positive, suggesting that existing apps in the same category have a positive spillover effect on the focal new app in terms of popularity. A one percent increase in the popularity of these existing apps translates into 0.168 percent increase in the new app's popularity. H3a is thus supported. The coefficient of *DifCatRatingVol* is insignificant and its magnitude is quite small. This indicates that the popularity of existing apps in different categories does not exert influence on the popularity of new apps across categories; H3b is also supported. Hence, developers can only harness the popularity spillover effect of existing apps in the same category as the focal new app. Another way to think of this is that users do not care about the popularity of a developer' overall portfolio, but care about the popularity of apps in the same category in evaluating the focal new app.

The coefficients of *RatingVol* in both model (8b) and model (8c) are positive. This implies that new apps do not cannibalize existing apps' popularity. Instead, positive feedback effects in app popularity are revealed both within the same and across different categories. These findings are consistent with our theoretical arguments and support both H4a and H4b. These positive spillover effects are even larger in magnitude than that of the spillover effect from existing apps to the new app. The coefficient of *RatingVol* in model (8b) suggests that a one percent increase in the new app's popularity is associated with 0.383 percent of popularity growth in the developer's existing apps in the same category, which is much larger than the impact in the reverse direction (i.e., 0.168). Hence, we can conclude that there is a positive reinforcement loop between existing apps and new apps within the same category. Finally, we observe that the popularity spillover effect of new apps on existing apps is stronger within the same category than across different categories (0.383 vs. 0.180), and therefore indirectly lending support for a concentrated app portfolio.

4. DISCUSSION AND IMPLICATIONS

Our study investigates the impact of developers' portfolio choices and market characteristics on the supply and demand of mobile apps in the Apple App Store. On the supply side, since app quality, as perceived by consumers, is a crucial element of success for developers in the mobile app market, we examine the influence of portfolio size and diversity on app quality. Our empirical results do not support H1, which contends that the more apps a developer has amassed in its portfolio, the higher the average quality of its apps would be. This indicates that learning by doing does not necessarily translate into excellence in performance. A possible reason is that, unlike more structured tasks, such as manufacturing and service work, software development is knowledge intensive and requires a higher cognitive capacity (Boh et al. 2007). Simple repetition without planned summarization, abstraction and contemplation may not be useful for quality improvement of mobile apps. For example, releasing many "copycat" apps quickly into the market with a hope that at least one of them would be a success cannot be the right strategy for developers. This approach leaves little room for developers to carefully think about the flaws and shortcomings of their apps and to take a corrective action to improve quality. Hence, ample experience in publishing apps may not necessarily lead to quality advances.

Our results show that the diversity of a developer's app portfolio negatively influences the developer's app quality (H2a), but the strength of this effect decreases with the developer's app portfolio size (H2b). Hence, our findings suggest that when the size of an app portfolio is small, increasing app portfolio diversity is detrimental to developers' app quality. After the app portfolio size crosses a critical point, diversifying the app portfolio benefits developers in terms of improved app quality because a larger portfolio mitigates the negative effect of diversity. This result echoes the viewpoint of punctuated equilibrium in organizational learning (Burgelman 2002; Levinthal and March 1993; Siggelkow and Levinthal 2003). Punctuated equilibrium suggests that exploitation and exploration are two ends of a continuum (Burgelman 2002). In contrast, the view of ambidexterity maintains that exploitation and exploration are orthogonal and organizations therefore can develop both capabilities simultaneously (Benner and Tushman 2003). Since exploitation and exploration learning compete for scarce resources, organizations need to strike a balance between them. Gupta et al. (2006) contend that when an organization context confines to a single domain, punctuated equilibrium is more appropriate to achieve the balance. Most app developers in our study are small-scale entrepreneurs, who focus on app development with limited resources. It is hard for them to engage in exploitation and exploration learning at the same time. Hence, a reasonable way for them to seek balance between exploitation and exploration is to follow the path of punctuated equilibrium. Specifically, exploitation at the early stages allows developers to deepen their understanding of a particular area by focusing on a small scope of development work, learning concrete knowledge as well as abstract learning methods. The abstract learning methods can establish the foundation for subsequent exploration, which involves expanding app development scope to other different categories.

On the demand side, we investigate the extent and direction of popularity spillover effects between developers' existing and new apps. Our results reveal that a developer's existing and new apps influence each other's popularity simultaneously. As a means to resolve initial uncertainty about app quality, consumers' usage experiences with existing apps provide a good reference for them to infer the quality of new apps released by the same developer. Users with positive (negative) experiences with previous apps are likely to extend their positive (negative) evaluations to recently released apps by the same developer. As proposed in H3a and H3b, our results confirm that there is a spillover effect of popularity from a developer's existing apps to its new apps within the same category, but no such impact is uncovered in apps across different categories. Correspondingly, we show that the popularity of a developer's new app increases the popularity of its existing apps within the same category, thus supporting H4a. Hence, there is a positive reinforcement loop between existing apps and new apps in the same category, creating a virtuous cycle. Existing apps can thus be a vehicle for developers to promote their new apps. New apps, in turn, enable users to discover existing apps of a developer. Moreover, our results reveal that there is a positive popularity spillover effect from new apps to existing apps across different categories, thus supporting H4b. However, this effect is smaller in magnitude than the effect for apps within the same category. Overall, our demand side findings on the popularity spillover effects within and across app categories indirectly support our supply side claim that focusing on fewer categories is conducive for developers to cultivate their core competency in order to improve product quality.

This research contributes to the IS research literature in the following ways. First, our work extends the study of portfolio management from a traditional industry such as manufacturing to an emerging mobile app industry, helping us to understand how portfolio management influences firms' performance in such new markets of the digital economy. Different from the manufacturing sector where production can be automated with machinery, the mobile app industry requires a high level of human capital. Our results highlight the importance of learning as an effective way to increase human capital.

Second, this research provides insights into the portfolio strategies of specialization and diversification through the theoretical lenses of exploitation and exploration learning. In order to achieve superior performance, app portfolio management strategies should be aligned with developers' learning curves. A focused app portfolio involves more local search and reuse of existing knowledge, which is consistent with the practice of exploitation. However, a diverse app portfolio encompasses tasks in different contexts with a higher level of uncertainty and creativity, which is consistent with the practice of exploration. Punctuated equilibrium is thus the appropriate strategy to achieve a right balance between these two types of learning, together with the consideration of scarce resources possessed by mobile app developers. Mapping punctuated equilibrium to app portfolio management requires specialization in earlier phases, followed by diversification later on.

Third, our study also sheds light into how app portfolio management influences the branding effect in terms of a developer's name or identity on users' demand for mobile apps. Users oftentimes draw quality inference from existing apps, but this inference is moderated by category relatedness between new and existing apps. Such a spillover effect in terms of a developer's brand also acts in the opposite direction, possibly due to cross-advertising mechanisms in mobile app markets.

This research also delivers several important insights to practitioners. While the proliferation of electronic exchanges such as eBay, Taobao for physical goods and the Apple App Store and Google Play for information goods provides many small and medium enterprises ample business opportunity, these online exchanges have been criticized as overcrowded, hypercompetitive and lacking viability for new sellers. Our study suggests that fostering core competency is crucial for mobile app developers to cope with the stiff competition. Compared with developing apps for many different categories, focusing app development on one or two categories at the early stage enables developers to utilize their limited resources more effectively. Developers later can consider diversifying their portfolio to more categories when they have accumulated enough experience.

In addition, app developers may want to take advantage of the positive popularity spillovers between existing and new apps to harvest larger market responses. If mobile app developers have a well-received app, releasing a new app in the same category increases users' receptivity and response to this new app. Releasing a new app is also a good way to attract user attention, which may be directed to developers' existing apps subsequently. This attention-direction effects is more salient within the same category than across categories. Therefore, developers can adjust their production plan according to their own marketing needs.

Furthermore, the popularity spillovers between existing and new apps have implications for developers' app release calendar as well. While mobile app developers aim to maximize market shares for their apps with timely app releases, they also need to minimize releases of premature, buggy apps. Releasing an app early may be able to preempt potential competition, but the ensuing app quality may not be guaranteed due to shortened development cycles. Fortunately, mobile app stores allow developers to update their existing apps by releasing new versions. Under such circumstances, releasing an app early and marketing it along with the release of other new apps later offers developers extra time to refine and improve the app. Early release of apps may decrease the extent of potential competition and yet help to shape users' preferences in favor of the early entrant. Once an existing app is capable of providing superior user experiences, the app developer can cross-advertise it with its new app release. This asynchronous product release and marketing plan can be a feasible path for mobile app developers' new product strategy.

5. CONCLUSIONS

In this study, we examine the influence of category assortment on developers' app quality and popularity using a large data set from the Apple App Store. Our analyses reveal a number of results that are novel and interesting. First, we show that, unlike our expectations, portfolio size does not influence the quality of apps released by developers. Second, concentrating on fewer categories through exploitation, or increasing specialization in app portfolios, helps developers to focus their limited resources on a narrow production scope and produce quality improvements. Third, the negative impact of portfolio diversification on app quality decreases with the size of developers' portfolio. With growing exploitation, developers are able to apply the learning methods gained from previous experience to new domains. As a result, portfolio diversification pays off only for experienced developers. Fourth, the popularity of existing apps of a developer positively influences the popularity of its new apps within the same category. Hence, consumers carry over their prior experiences into the apps released by the same developer. Fifth, the popularity of a developer's new app also promotes the popularity of its existing apps, irrespective of category relatedness among apps. Hence, a new app release not only expands a developers' app portfolio but also is an effective way to facilitate the discovery of the developer's existing apps. Overall, we conclude that there is a positive reinforcing loop between existing apps and new apps, creating a virtuous cycle.

However, our study does not come without limitations. First, we used the category classification scheme defined by the Apple App Store to quantify the diversity of developers' app portfolio. The category classification of each app is set by developers. It may reflect developers' perception of category affiliation of their apps or it may be a result of developers' intentional marketing strategy. Some developers may categorize their apps into popular categories (e.g., Games, Entertainment) rather than the most relevant categories so as to reach a larger user base. Hence, future research may use an alternative classification scheme.

Second, although we controlled for portfolio-level and market-level characteristics, developers' portfolio management strategy may also be influenced by factors that were not accounted for, such as the size of the development team, career experience of team members and funding conditions. While we do not have information on these characteristics, such attributes are partially controlled by developer fixed effects in our study.

Last, we investigated the impacts of factors such as portfolio size and diversity on app quality and app popularity. Although these two quantities of interest are important for developers, app downloads and total revenue from sales are probably the primary concerns of developers. Since the Apple App Store keeps app downloads and revenue data confidential, the general public is unable to access this crucial information, which is acknowledged by other related studies (e.g., Liu et al. 2012; Zhong and Michahelles 2013). Though a method (Garg and Telang 2013) to infer app downloads from daily rankings is available, this method is not useful for unranked apps, which is the case for most apps in our data set. Instead, we rely on user ratings to measure developers' performance. Nevertheless, future studies would immensely benefit from proprietary app downloads or sales data, if they become available. Despite these limitations, our study paves the way for further research to study the success factors in mobile app market

References

148Apps.Biz. 2013. "App Store Metrics." Retrieved November 26, 2013, from http://148apps.biz/app-store-metrics/?mpage

Aggarwal, R., Gopal, R., Gupta, A., and Singh, H. 2012. "Putting Money Where the Mouths Are: The Relation between Venture Financing and Electronic Word-of-Mouth," *Information Systems Research* (23:3-Part-2), pp. 976-992.

Ang, S., Slaughter, S., and Ng, K.Y. 2002. "Human Capital and Institutional Determinants of Information Technology Compensation: Modeling Multilevel and Cross-Level Interactions," *Management Science* (48:11), pp. 1427-1445.

Apple. 2013. "Apple's App Store Marks Historic 50 Billionth Download." Retrieved November 30, 2013, from http://www.apple.com/pr/library/2013/05/16Apples-App-Store-Marks-Historic-50-Billionth-Download.html

Argote, L., and Miron-Spektor, E. 2011. "Organizational Learning: From Experience to Knowledge," *Organization Science* (22:5), pp. 1123-1137.

Bailey, E.E., and Friedlaender, A.F. 1982. "Market Structure and Multiproduct Industries," *Journal of Economic Literature* (20:3), pp. 1024-1048.

Banker, R., Wattal, S., Liu, F.-C., and Ou, C.-S. 2008. "Value Contributed by Education in IT Firms," *Proceedings of the 29th International Conference on Information Systems*, Paris, France.

Baum, J.A., Li, S.X., and Usher, J.M. 2000. "Making the Next Move: How Experiential and Vicarious Learning Shape the Locations of Chains' Acquisitions," *Administrative Science Quarterly* (45:4), pp. 766-801.

Benner, M.J., and Tushman, M. 2002. "Process Management and Technological Innovation: A Longitudinal Study of the Photography and Paint Industries," *Administrative Science Quarterly* (47:4), pp. 676-707.

Benner, M.J., and Tushman, M.L. 2003. "Exploitation, Exploration, and Process Management: The Productivity Dilemma Revisited," *Academy of Management Review* (28:2), pp. 238-256.
Bergvall-Kåreborn, B., and Howcroft, D. 2011. "Mobile Applications Development on Apple and Google Platforms," *Communications of the Association for Information Systems* (29:1), p. 30.
Berry, C.H. 1971. "Corporate Growth and Diversification," *Journal of Law and Economics* (14:2), pp. 371-383.
Bhargava, H.K., and Choudhary, V. 2001. "Information Goods and Vertical Differentiation," *Journal of Management Information Systems* (18:2), pp. 89-106.
Boh, W.F., Slaughter, S.A., and Espinosa, J.A. 2007. "Learning from Experience in Software Development: A Multilevel Analysis," *Management Science* (53:8), pp. 1315-1331.
Burgelman, R.A. 2002. "Strategy as Vector and the Inertia of Coevolutionary Lock-In," *Administrative Science Quarterly* (47:2), pp. 325-357.
Caliendo, M., and Kopeinig, S. 2008. "Some Practical Guidance for the Implementation of Propensity Score Matching," *Journal of Economic Surveys* (22:1), pp. 31-72.
Cardozo, R.N., and Smith Jr, D.K. 1983. "Applying Financial Portfolio Theory to Product Portfolio Decisions: An Empirical Study," *Journal of Marketing* (47:2), pp. 110-119.
Chevalier, J.A., and Mayzlin, D. 2006. "The Effect of Word of Mouth on Sales: Online Book Reviews," *Journal of Marketing Research* (43:3), pp. 345-354.
CNET. 2012. "Mobile App Revenue Set to Soar to $46 Billion in 2016." Retrieved November 26, 2013, from http://news.cnet.com/8301-13506_3-57379364-17/mobile-app-revenue-set-to-soar-to-$46-billion-in-2016/
comScore. 2013. "Comscore Reports August 2013 U.S. Smartphone Subscriber Market Share." Retrieved January 16, 2014, from http://www.comscore.com/Insights/Press_Releases/2013/10/comScore_Reports_August_2013_US_Smartphone_Subscriber_Market_Share
Datta, A., Dutta, K., Kajanan, S., and Pervin, N. 2012. "Mobilewalla: A Mobile Application Search Engine," in *Mobile Computing, Applications, and Services*. Springer, pp. 172-187.
Dellarocas, C. 2003. "The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms," *Management Science* (49:10), pp. 1407-1424.
Drukker, D.M. 2003. "Testing for Serial Correlation in Linear Panel-Data Models," *Stata Journal* (3:2), pp. 168-177.
Durcikova, A., Fadel, K.J., Butler, B.S., and Galletta, D.F. 2011. "Research Note—Knowledge Exploration and Exploitation: The Impacts of Psychological Climate and Knowledge Management System Access," *Information Systems Research* (22:4), pp. 855-866.
Dutta, A. 2001. "Telecommunications and Economic Activity: An Analysis of Granger Causality," *Journal of Management Information Systems* (17:4), pp. 71-96.
Erdem, T. 1998. "An Empirical Analysis of Umbrella Branding," *Journal of Marketing Research* (35:3), pp. 339-351.
Gallivan, M.J., Truex III, D.P., and Kvasny, L. 2004. "Changing Patterns in IT Skill Sets 1988-2003: A Content Analysis of Classified Advertising," *Data Base for Advances in Information Systems* (35:3), pp. 64-87.
Garg, R., and Telang, R. 2013. "Inferring App Demand from Publicly Available Data," *MIS Quarterly* (37:4), pp. 1253-1264.
Garud, R., and Kumaraswamy, A. 2005. "Vicious and Virtuous Circles in the Management of Knowledge: The Case of Infosys Technologies," *MIS Quarterly* (29:1), pp. 9-33.

Godes, D., and Mayzlin, D. 2009. "Firm-Created Word-of-Mouth Communication: Evidence from a Field Test," *Marketing Science* (28:4), pp. 721-739.

Gopal, R.D., Bhattacharjee, S., and Sanders, G.L. 2006. "Do Artists Benefit from Online Music Sharing?," *The Journal of Business* (79:3), pp. 1503-1533.

Granger, C.W. 1969. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods," *Econometrica* (37:3), pp. 424-438.

Gupta, A.K., Smith, K.G., and Shalley, C.E. 2006. "The Interplay between Exploration and Exploitation," *Academy of Management Journal* (49:4), pp. 693-706.

Haefliger, S., Von Krogh, G., and Spaeth, S. 2008. "Code Reuse in Open Source Software," *Management Science* (54:1), pp. 180-193.

Han, S.P., and Ghose, A. 2012. "Estimating Demand for Applications in the New Mobile Economy," *Proceedings of the 33th International Conference on Information Systems*, Orlando, Florida.

Heckman, J.J., Ichimura, H., and Todd, P. 1998. "Matching as an Econometric Evaluation Estimator," *Review of Economic Studies* (65:2), pp. 261-294.

Hui, K.-L., and Png, I. 2003. "Piracy and the Legitimate Demand for Recorded Music," *Contributions in Economic Analysis & Policy.* (2:1).

Jacquemin, A.P., and Berry, C.H. 1979. "Entropy Measure of Diversification and Corporate Growth," *The Journal of Industrial Economics* (27:4), pp. 359-369.

Levina, N., and Xin, M. 2007. "Research Note—Comparing IT Workers' Compensation across Country Contexts: Demographic, Human Capital, and Institutional Factors," *Information Systems Research* (18:2), pp. 193-210.

Levinthal, D.A., and March, J.G. 1993. "The Myopia of Learning," *Strategic Management Journal* (14:S2), pp. 95-112.

Liu, C.Z., Au, Y.A., and Choi, H.S. 2012. "An Empirical Study of the Freemium Strategy for Mobile Apps: Evidence from the Google Play Market," *Proceedings of the 33th International Conference on Information Systems*, Orlando, Florida.

Liu, Y. 2006. "Word of Mouth for Movies: Its Dynamics and Impact on Box Office Revenue," *Journal of Marketing* (70:3), pp. 74-89.

Montgomery, C.A., and Wernerfelt, B. 1988. "Diversification, Ricardian Rents, and Tobin's Q," *RAND Journal of Economics* (19:4), pp. 623-632.

Montgomery, C.A., and Wernerfelt, B. 1992. "Risk Reduction and Umbrella Branding," *Journal of Business* (65:1), pp. 31-50.

Mukhopadhyay, T., Singh, P., and Kim, S.H. 2011. "Learning Curves of Agents with Diverse Skills in Information Technology-Enabled Physician Referral Systems," *Information Systems Research* (22:3), pp. 586-605.

Nelson, P. 1970. "Information and Consumer Behavior," *Journal of Political Economy* (78:2), pp. 311-329.

Qiu, Y., Gopal, A., and Hann, I.-H. 2011. "Synthesizing Professional and Market Logics: A Study of Independent iOS App Entrepreneurs," *Proceedings of the 32th International Conference on Information Systems*, Shanghai, China.

Rosch, E., and Mervis, C.B. 1975. "Family Resemblances: Studies in the Internal Structure of Categories," *Cognitive Psychology* (7:4), pp. 573-605.

Rosenbaum, P.R., and Rubin, D.B. 1983. "The Central Role of the Propensity Score in Observational Studies for Causal Effects," *Biometrika* (70:1), pp. 41-55.

Sangaralingam, K., Pervin, N., Ramasubbu, N., Datta, A., and Dutta, K. 2012. "Takeoff and Sustained Success of Apps in Hypercompetitive Mobile Platform Ecosystems: An Empirical Analysis," *Proceedings of the 33th International Conference on Information Systems*, Orlando, Florida.

Schilling, M.A., Vidal, P., Ployhart, R.E., and Marangoni, A. 2003. "Learning by Doing Something Else: Variation, Relatedness, and the Learning Curve," *Management Science* (49:1), pp. 39-56.

Shankar, V. 2006. "Proactive and Reactive Product Line Strategies: Asymmetries between Market Leaders and Followers," *Management Science* (52:2), pp. 276-292.

Siggelkow, N., and Levinthal, D.A. 2003. "Temporarily Divide to Conquer: Centralized, Decentralized, and Reintegrated Organizational Approaches to Exploration and Adaptation," *Organization Science* (14:6), pp. 650-669.

Singh, P.V., Tan, Y., and Youn, N. 2011. "A Hidden Markov Model of Developer Learning Dynamics in Open Source Software Projects," *Information Systems Research* (22:4), pp. 790-807.

Sundararajan, A. 2004. "Managing Digital Piracy: Pricing and Protection," *Information Systems Research* (15:3), pp. 287-308.

VisionMobile. 2013. "Developer Economics 2013: The Tools Report."

Wernerfelt, B. 1988. "Umbrella Branding as a Signal of New Product Quality: An Example of Signalling by Posting a Bond," *The RAND Journal of Economics* (19:3), pp. 458-466.

Wooldridge, J.M. 2002. *Econometric Analysis of Cross Section and Panel Data*. The MIT press.

Zhong, N., and Michahelles, F. 2013. "Google Play Is Not a Long Tail Market: An Empirical Analysis of App Adoption on the Google Play App Market," *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal: ACM, pp. 499-504.

Table 1 Variable Definitions and Operationalizations

Variable	Definition and Operationalization
$AvgRatingValence_{it}$	Average user rating score of the apps in developer *i's* portfolio in month *t*, calculated based on the rating scores on the last day of month *t*.
$APSize_{it}$	Total number of apps in developer *i's* portfolio in month *t*, calculated based on developer *i*'s app portfolio on the last day of month *t*.
$APDiversity_{it}$	Category entropy of developer *i*'s portfolio in month *t*, calculated using Equation (1) and Equation (2).
$Company_{it}$	Indicator for a company-based developer (=1, company developer; =0, individual developer). We classified developers based on the last word of their names since most company-based developers' names end with words such as "Ltd.", "Inc.", etc.
$FreeRatio_{it}$	Number of free apps divided by the total number of apps in developer *i's* portfolio in month *t*. Free apps are apps with zero average price in month *t*.
$AvgPrice_{it}$	Average price of all the apps in developer *i's* portfolio in month *t*. We first calculated the average price of each app in month *t*, and then computed the mean value of the average prices of apps in developer *i's* portfolio.
$StdPrice_{it}$	Standard deviation of prices of all the apps in developer *i's* portfolio in month *t*. We first calculated the average price of each app in month *t*, and then computed the standard deviation of the average prices of apps in developer *i's* portfolio.
$PromotionRatio_{it}$	Number of apps that have been promoted divided by the total number of apps in developer *i*'s portfolio in month *t*. We first obtained the standard deviation of prices of each app in month *t*. Apps with non-zero price standard deviation were considered as having been promoted in month *t*.
$AvgVersionNum_{it}$	Average number of versions an app has in developer *i*'s portfolio in month *t*. We first counted the total number of versions that each app had on the last day of month *t*, then calculated the average version number for developer *i*'s apps.
$Tenure_{it}$	Count of months passed since developer *i* released the first app till month *t*.
$iOSNewApps_t$	Number of new apps released in the Apple App Store in month *t* (in thousands).
$iOSPromotionApps_t$	Number of apps promoted in the Apple App Store in month *t* (in thousands). The operationalization of promoted apps is the same as that of *PromotionRatio*.
$iOSAppNum_t$	Total number of apps available in the Apple App Store in month *t*, excluding the new apps (in hundred thousands).
CategoryIndicators	A set of binary variables that indicate which categories developer *i* have released apps in.

Table 2 Descriptive Statistics (obs.= 92,018)

Variable	Mean	Std. Dev.	Min	Max
AvgRatingValence	0.958	1.288	0.000	5.000
APSize	7.777	9.713	1.000	307.000
APDiversity	0.993	0.485	0.000	2.656
Company	0.370	0.483	0.000	1.000
FreeRatio	0.515	0.383	0.000	1.000
AvgPrice	1.834	8.859	0.000	899.990
StdPrice	1.203	7.784	0.000	494.968
PromotionRatio	0.000	0.003	0.000	0.471
AvgVersionNum	1.304	0.813	0.500	30.000
Tenure	7.201	3.156	3.000	15.000
iOSNewApps (in 1k)	23.189	2.621	18.935	27.654
iOSPromotionApps (in 1k)	4.309	3.582	2.003	36.833
iOSAppNum (in 100k)	5.611	0.767	3.697	6.593

Table 3 Correlation Matrix

Variable	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***
1. AvgRatingValence	1.00												
2. APSize	-0.06	1.00											
3. APDiversity	-0.10	0.18	1.00										
4. Company	0.09	-0.03	-0.06	1.00									
5. FreeRatio	0.04	-0.18	-0.04	0.10	1.00								
6. AvgPrice	-0.04	0.01	-0.05	-0.03	-0.19	1.00							
7. StdPrice	-0.01	0.01	0.00	-0.01	-0.08	0.61	1.00						
8. PromotionRatio	0.02	-0.01	-0.01	0.00	0.00	0.00	0.00	1.00					
9. AvgVersionNum	0.55	-0.05	-0.05	0.05	0.04	-0.02	0.00	0.04	1.00				
10. Tenure	-0.03	0.15	0.15	0.01	0.03	-0.01	0.02	-0.02	0.00	1.00			
11. iOSNewApps	0.07	-0.08	-0.08	0.00	0.01	0.00	-0.01	0.01	0.05	-0.42	1.00		
12.iOSPromotionApps	0.02	-0.02	-0.02	0.00	0.00	0.01	-0.01	0.11	0.03	-0.07	0.04	1.00	
13. iOSAppNum	-0.13	0.13	0.13	-0.01	-0.02	0.01	0.02	-0.03	-0.10	0.60	-0.67	-0.13	1.00

Table 4 Model Estimation Results of Supply Side Analysis

Variable		(1) FE	(2) RE	(3) FE with Interaction	(4) RE with Interaction
APSize		0.0019*** (0.0004)	-0.0043*** (0.0003)	0.0000 (0.0008)	-0.0085*** (0.0007)
APDiversity		-0.0213*** (0.0063)	-0.0807*** (0.0059)	-0.0258*** (0.0065)	-0.0938*** (0.0061)
*APDiversity*APSize*				0.0016*** (0.0006)	0.0035*** (0.0005)
*APDiversity*Company*		-0.0049 (0.0098)	0.0271*** (0.0082)	-0.0049 (0.0098)	0.0286*** (0.0082)
FreeRatio		0.0190** (0.0081)	0.0303*** (0.0075)	0.0191** (0.0081)	0.0306*** (0.0075)
AvgPrice		-0.0000 (0.0002)	-0.0004* (0.0002)	-0.0000 (0.0002)	-0.0004* (0.0002)
PromotionRatio		-0.0372 (0.0281)	0.0560** (0.0251)	-0.0375 (0.0281)	0.0556** (0.0251)
AvgVersionNum		-0.0007 (0.0031)	0.1205*** (0.0028)	-0.0009 (0.0031)	0.1200*** (0.0028)
Tenure		0.0208*** (0.0079)	0.0237*** (0.0025)	0.0206*** (0.0079)	0.0239*** (0.0025)
iOSPromotionApps		-0.0000 (0.0001)	0.0001 (0.0001)	-0.0000 (0.0001)	0.0001 (0.0001)
iOSAppNum		-0.1696*** (0.0325)	-0.2507*** (0.0102)	-0.1703*** (0.0326)	-0.2501*** (0.0102)
Constant		1.6600*** (0.0331)	2.0889*** (0.0451)	1.6720*** (0.0332)	2.1006*** (0.0451)
Category Indicators		YES	YES	YES	YES
R^2	Within	0.0084	0.2272	0.0086	0.2293
	Between	0.0120	0.1661	0.0118	0.1672
	Overall	0.0183	0.1601	0.0181	0.1611

Notes: 1. Standard errors in parentheses, *** p<0.01, ** p<0.05, * p<0.1.
2. Number of observations = 92,018, number of developers = 11,579.

Table 5 Alternative Measurements of Portfolio Diversity

Variable		(a) FE - CatHHI	(b) FE - CatCount
APSize		-0.0010 (0.0012)	0.0012* (0.0006)
APDiversity		-0.0565*** (0.0142)	-0.0047** (0.0019)
*APDiversity*APSize*		0.0047*** (0.0018)	0.0001* (0.0001)
*APDiversity*Company*		-0.0084 (0.0214)	-0.0014 (0.0027)
FreeRatio		0.0190** (0.0081)	0.0199** (0.0081)
AvgPrice		-0.0000 (0.0002)	-0.0000 (0.0002)
PromotionRatio		-0.0372 (0.0281)	-0.0371 (0.0281)
AvgVersionNum		-0.0008 (0.0031)	-0.0007 (0.0031)
Tenure		0.0206*** (0.0079)	0.0209*** (0.0079)
iOSPromotionApps		-0.0000 (0.0001)	-0.0000 (0.0001)
iOSAppNum		-0.1699*** (0.0325)	-0.1706*** (0.0326)
Constant		1.6763*** (0.0332)	1.6613*** (0.0333)
Category Indicators		YES	YES
R^2	Within	0.0085	0.0084
	Between	0.0114	0.0118
	Overall	0.0177	0.0181

Notes: 1. Standard errors in parentheses, *** p<0.01, ** p<0.05, * p<0.1.
2. Number of observations = 92,018, number of developers = 11,579.

Table 6 Propensity Score Estimation

Variable	Coefficient	Variable	Coefficient
APSize	0.027*** (0.001)	*Tenure*	0.064*** (0.002)
PrevRatingVal	-0.038*** (0.004)	*Company*	-0.054*** (0.011)
PrevRatingVol	-0.000*** (0.000)	Category Distribution	YES
		Pseudo-R^2	0.1158

Notes: 1. Standard errors in parentheses, *** p<0.01, ** p<0.05, * p<0.1.
2. A set of P_s, as shown in Equation (2) has been incorporated to control for category distribution. Their coefficients have been omitted due to space limitations, but are available upon request.

Table 7 Matching Results

Developer Group	Estimator	Treated	Controls	ATT	T-value	R^2 After
All developers	1NN	0.883	1.020	-0.137	-7.00	0.012
	3NN		0.967	-0.084	-5.11	0.011
	Caliper		1.020	-0.137	-7.01	0.012
Group1 (*APSize* < 10)	1NN	0.872	1.051	-0.178	-8.12	0.013
	3NN		0.973	-0.101	-5.66	0.010
	Caliper		1.051	-0.178	-8.12	0.013
Group 2 (*APSize* >= 10)	1NN	0.897	0.812	0.085	2.35	0.035
	3NN		0.775	0.122	3.85	0.035
	Caliper		0.811	0.086	2.38	0.035

Table 8 Variable Definitions

Variable	Definition	Control Variable
$RatingVol_{it}$*	Total number of user ratings for new app *i* by week *t*.	
$SameCatRatingVol_{it}$*	Average number of user ratings by week *t* for existing apps by the developer who developed the new app *i*, and that are in the same category as the new app *i*.	
$DifCatRatingVol_{it}$*	Average number of user ratings by week *t* for existing apps by the developer who developed the new app *i*, and that are in different categories from that of the new app *i*.	
$SameCatNewRatingVol_{it}$*	Average number of user ratings by week *t* for other new apps by the developer who developed the new app *i*, and that are in the same category as the new app *i*.	8a
$DifCatNewRatingVol_{it}$*	Average number of user ratings by week *t* for other new apps by the developer who developed the new app *i*, and that are in different categories from that of the new app *i*.	8a
$SameCatApps_{it}$*	Cumulative number of existing apps by week *t*, that are released by the developer who developed the new app *i*, and which are in the same category as the new app *i*.	8a
$DifCatApps_{it}$*	Cumulative number of existing apps by week *t*, that are released by the developer who developed the new app *i*, and which are in different categories from that of the new app *i*.	8a
$SameCatNewApps_{it}$*	Total number of other new apps in week *t* released by the developer who developed the new app *i*, and which are in the same category as the new app *i*.	8a
$DifCatNewApps_{it}$*	Total number of other new apps in week *t* released by the developer who developed the new app *i*, and which are in different categories from that of the new app *i*.	8a
$Free_{it}$	= 1 if new app *i* is free; = 0 otherwise.	8a
$Price_{it}$	Price of new app *i* in week *t*.	8a
$RatingVal_{it}$	Average user rating score of the new app *i* at the end of week *t*.	8a
$Promotion_{it}$	= 1 if new app *i* is on promotion, = 0 otherwise. New	8a

	app i is considered as being promoted if its price standard deviation in week t is non-zero.	
$NewAppsInCat_{it}$*	Total number of new apps that were released by all developers in week t that are in the same category as new app i.	8a
$SameCatFreePerc_{it}$	Percentage of free existing apps in week t in the portfolio of the developer who released the new app i, and that are in the same category as the new app i.	8b
$DifCatFreePerc_{it}$	Percentage of free existing apps in week t in the portfolio of the developer who released the new app i, and that are in different categories from that of the new app i.	8c
$SameCatPrice_{it}$	Average price of existing apps in week t in the portfolio of the developer who released the new app i, and that are in the same category as the new app i.	8b
$DifCatPrice_{it}$	Average price of existing apps in week t in the portfolio of the developer who released the new app i, and that are in different categories from that of the new app i.	8c
$SameCatRatingVal_{it}$	Average user rating of existing apps in week t in the portfolio of the developer who released the new app i, and that are in the same category as the new app i.	8b
$DifCatRatingVal_{it}$	Average user rating of existing apps in week t in the portfolio of the developer who released the new app i, and that are in different categories from that of the new app i.	8c
WeekDummies	A set of week dummies to capture the time trend.	8a, 8b, 8c

* Variable enters the equations with log-transformations.

Table 9 Descriptive Statistics (obs.= 48,084)

Variable	Mean	Std. Dev.	Min	Max
RatingVol	69.65	1754.13	0.00	111085.00
SameCatRatingVol	190.25	1459.14	0.00	69609.00
DifCatRatingVol	30.30	581.63	0.00	32612.00
SameCatNewRatingVol	7.51	273.29	0.00	20212.70
DifCatNewRatingVol	0.12	4.99	0.00	777.50
SameCatApps	16.79	49.19	0.00	653.00
DifCatApps	11.74	51.05	0.00	706.00
SameCatNewApps	0.83	3.65	0.00	67.00
DifCatNewApps	0.62	4.33	0.00	80.00
Free	0.52	0.50	0.00	1.00
Price	1.83	10.75	0.00	499.99
Promotion	0.01	0.08	0.00	1.00
RatingVal	0.98	1.72	0.00	5.00
NewAppsInCat	858.74	461.92	12.00	2850.00
SameCatFreePerc	0.33	0.40	0.00	1.00
SameCatPrice	1.99	12.54	0.00	466.66
SameCatRatingVal	1.02	1.38	0.00	5.00
DifCatFreePerc	0.19	0.34	0.00	1.00
DifCatPrice	0.58	1.67	0.00	44.24
DifCatRatingVal	0.56	1.09	0.00	5.00

Table 10 Model Estimation Results of Demand Side Analysis

Variable		Model (8a) Focal New App	Model (8b) Apps in the Same Category	Model (8c) Apps in Diff. Categories
Portfolio-level Factors	*SameCatRatingVol*	0.168*** (0.007)		
	DifCatRatingVol	0.001 (0.012)		
	SameCatNewRatingVol	-0.051*** (0.003)		
	DifCatNewRatingVol	-0.021*** (0.006)		
	RatingVol		0.383*** (0.037)	0.180*** (0.014)
	SameCatApps	-0.022*** (0.005)		
	DifCatApps	0.088*** (0.008)		
	SameCatNewApps	0.023*** (0.003)		
	DifCatNewApps	0.032*** (0.004)		
Own Characteristics	*Free*	0.030 (0.027)		
	SameDifCatFreePerc		0.071*** (0.008)	
	DifCatFreePerc			0.055*** (0.008)
	Price	-0.002 (0.008)		
	SameCatPrice		0.001*** (0.000)	
	DifCatPrice			0.006*** (0.001)
	RatingVal	0.196*** (0.024)		
	SameCatRatingVal		0.438*** (0.004)	
	DifCatRatingVal			0.501*** (0.003)
	Promotion	0.067*** (0.008)		
Competition Intensity	*NewAppsInCat*	-0.071*** (0.017)		
Constant		0.000 (0.001)	-0.000 (0.001)	0.000 (0.000)
WeekDummies		Yes	Yes	Yes
Observations		48,084	48,084	48,084
R^2		0.220	0.371	0.410

Notes: 1. Standard errors in parentheses, *** p<0.01, ** p<0.05, * p<0.1.
2. To avoid ln(0), we use ln(x+1) in log-transformations.
3. A fixed-effects estimator was utilized to account for app-specific heterogeneity.