The Value of Technology Releases in the Apple iOS App Ecosystem

Jin-Hyuk Kim
Associate Professor, University of Colorado Boulder

Yidan Sun
Illinois Institute of Technology

Liad Wagman
Professor of Economics, Illinois Institute of Technology

EODS Economics of Digital Services @Penn

CTIC Center for Technology, Innovation and Competition

The Warren Center for Network & Data Sciences

Penn University of Pennsylvania

Economics of Digital Services (EODS) is an initiative of the University of Pennsylvania's Center for Technology, Innovation and Competition (CTIC) and The Warren Center for Network & Data Sciences. Its aim is to generate independent research on the economics of digital services and the role of data and algorithms in the business strategies of digital platforms. The initiative was funded by a major grant from the John S. and James L. Knight Foundation to support scholarly inquiry and novel approaches in the evolving digital age.

To learn more about the initiative, visit www.law.upenn.edu/digitaleconomics/.

Center for Technology, Innovation and Competition (CTIC)

CTIC is an interdisciplinary academic center at the University of Pennsylvania Carey Law School that bridges law and technology for academia and students. Focusing on intellectual property, antitrust, internet law, and privacy law and policy. CTIC delivers foundational research that shapes the way legislators, regulatory authorities, and scholars develop policy and regulatory frameworks. CTIC also produces programming that explores the full range of scholarly perspectives, engages with industry experts, and prepares the next generation of technology scholars, lawyers, and policymakers.

www.pennCTIC.org

The Warren Center for Network & Data Sciences

The University of Pennsylvania's Warren Center fosters research and innovation in interconnected social, economic, and technological systems. Collaborating with Penn affiliates, it focuses on the role of data and algorithms to understand networked systems and how they can improve lives. The center also produces events that connect researchers, students, and entrepreneurs across the spectrum of network science.

www.warrencneter.upenn.edu

The Value of Technology Releases in the Apple iOS App Ecosystem*

Jin-Hyuk Kim[†] Yidan Sun[‡] Liad Wagman[§]

August 29, 2021

Abstract

Companies operating in the digital space regularly release new devices, platforms, and functionalities, often accompanied by Software Developer Kits (SDKs), which enable app developers to take advantage of technological innovations. Using a comprehensive dataset on Apple's iOS mobile app market, we describe the evolution of key time-series variables (iPhone sales, app releases, and SDK releases) and examine the drivers of mobile app releases. We find that SDK releases have a much larger impact on the introduction of new apps than rising smartphone sales, highlighting the role of this technology layer in the app ecosystem. We then estimate the consumer surplus generated from iOS apps, to estimate the value attributable to technological developments associated with SDKs.

Keywords: Mobile app; consumer surplus; software development

JEL Codes: L23, L96, M21

---

*This research was funded by the Data Catalyst Institute and by the University of Pennsylvania's Center for Technology, Innovation and Competition and Warren Center for Network and Data Sciences. The views and analyses in this paper are entirely the authors' own and are independent of any institution or agency with whom any of the authors may be affiliated. We are thankful to the participants at the University of Pennsylvania's Knight Foundation Program on the Economics of Digital Services 2020 launch event, the George Mason University 2021 PEP Scholars Research Roundtable, the 2021 International Industrial Organization Conference, and the 2021 Munich Summer Institute for helpful comments. Any errors are our own.

[†]University of Colorado Boulder. Email: jinhyuk.kim@colorado.edu.

[‡]Illinois Institute of Technology. Email: ysun100@hawk.iit.edu.

[§]Illinois Institute of Technology. Email: lwagman@stuart.iit.edu.

1 Introduction

Since Microsoft's release of the Windows operating system in 1985, advancements in hardware and operating systems (software that acts as an intermediary between applications and computing hardware) of various types have helped bring to life new kinds of electronic devices over the last few decades that are connected to communication networks. While Windows is probably the most well-known legacy OS and maintains over 80% market share to date in the desktop and laptop OS markets, it has a share of less than 1% in the mobile OS market. Instead, the mobile OS market has been largely divided between Android (developed by Google/Alphabet) and iOS (developed by Apple), with respective market shares of roughly 70% and 30% (NetMarketShare, 2020).

A mobile OS facilitates the offering of software applications (henceforth, apps) on mobile devices. This aspect of app markets has been extensively studied in the academic literature under the rubric of two-sided (or multi-sided) markets, where a platform brings together the supply side (app developers) and the demand side (phone users) to a joint marketplace. However, the existing literature has paid relatively little attention to the underlying technological drivers of the app market, or the developments of third-party technology platforms that have facilitated the rapid growth of mobile apps. That is, besides the OS, various tech platforms provide building blocks and/or services through their release of Software Developer Kits (SDKs), which facilitate the development of apps and enable numerous app features.[1]

In our data, there are more than a thousand SDKs that have been utilized over the history of the mobile app market. The vast majority of them (over 90%) have been released by third parties — commonly by other tech platforms — rather than by Apple, with millions of apps having collectively utilized those SDKs. Tech platforms are able to release SDKs in large part due to the functionalities offered by the hardware and operating systems of mobile devices. In turn, by adopting SDKs, app developers can leverage services and data provided by various platforms — including third-party platforms — helping facilitate the development of innovative apps that generate substantial consumer surplus. This technological layer of the app ecosystem

[1] The functions of SDKs are numerous. The most commonly offered SDKs in our dataset are associated with functionalities concerning "DEV_TOOL", "ANALYTICS", "GEO_LOCATION", "DEV_PLATFORM", and "AD_NETWORK." By releasing SDKs, technology platforms benefit from apps connecting to and utilizing their platforms, which could in turn result in a broadening of their user bases and/or direct monetary rewards (e.g., from licensing fees; although such fees are often privately negotiated, some SDKs are open source).

is essential because typical app developers, such as an individual or a small startup, often do not have the capacity or ability to develop all of the requisite infrastructure themselves for their apps — infrastructure that SDKs can provide.

This paper investigates the effect and the value of SDK releases in the mobile app ecosystem, leveraging comprehensive data on the iOS app market.[2] The analysis is conducted in two steps. The first step examines macro-level time-series data on smartphone sales, SDK releases and app releases, using vector autoregression (VAR) models. Since the 1980s, VARs have been a time-tested tool for capturing rich dynamics in multiple time series, providing a credible approach to data description, forecasting, and inference (Sims (1980); Stock and Watson (2001)). Such VAR models, when applied to industry-level time-series, can yield results that complement the analyses that are more commonly conducted by industrial organization researchers in economics.

As we elaborate below, a frequently used VAR model today is the so-called structural VAR, where, loosely speaking, a structure is often motivated by some underlying theory and/or institutional knowledge and aims to solve the identification problem, differentiating between correlation and causation. For instance, to determine the effect of an unexpected increase in smartphone sales on app releases, or the effect of an increase in SDKs on apps, or the proportion of the long-run variation in app releases that is due to device sales as opposed to SDK releases, etc. In Section 4, we attempt to answer these questions using three-variable VARs under a couple of identifying assumptions.

We find that if there is a 15% (one standard deviation) positive shock to the number of iPhones sold in a quarter, then there is a 2% to 5% increase in the number of new iOS apps released in the second and the third years after this shock. If there is a 40% (one standard deviation) positive shock to the number of SDKs released in a quarter, then there is an immediate 3% to 8% rise in the number of iOS apps, with this effect lasting over three years. In fact, when we calculate how important each shock is in explaining the variation in long-run app releases, we find that after two years, a shock to SDKs explains 42% of the variation while iPhone sales explain 17%. Hence, our analysis suggests that SDKs have a larger impact on app

[2] We do not examine the Android market because, unlike Apple, Android vendors sold mobile devices with multiple OS versions, and they do not report unit sales quantities, so accurate time-series data on smartphone sales could not be obtained. Similarly, unlike the Apple iOS App Store, Android has multiple app marketplaces, so comprehensive data (from, e.g., China) were not available. While we acknowledge that there are some differences between the market strategies of Android and iOS, we think that our results can still shed some light on the Android market as well.

releases than smartphone sales.

Our second step is to assess the consumer surplus of the iOS app market to end users, and attribute the surplus created to the three drivers of the mobile app ecosystem. To do so, we take a simple approach of directly estimating the market demand function for apps and using it to calculate the consumer surplus. Generally known under the rubric of the Hotelling (1938)-Clawson and Knetsch (1966) model, this approach has been widely used in the applied evaluation literature for its computational and analytical simplicity, and can yield a sensible result if appropriate care is taken to address endogeneity issues.[3] For instance, we use an instrumental variable for app's number of downloads, and a proxy variable to control for users' level of engagement with an app.

Across all game and non-game iOS app categories, we find that the iOS app market generates over \$60 million in consumer surplus each day.[4] The figures are relatively stable across consecutive days. At the app level, a game app generates, on average, \$130 to \$140 in consumer surplus on a given day, while a non-game app generates, on average, \$110 to \$120. However, for 'active' apps with a positive number of downloads per day, there are more non-game apps (about 420,000) than game apps (about 100,000), so the total consumer surplus is higher for non-game apps than for game apps. In sum, our results suggest that the roughly \$60 million in daily consumer surplus from the iOS app market is attributable to innovations in SDKs (42%), apps (41%), and iPhones (17%).

There are some limitations to our study. For instance, we are unable to further break down SDKs by function or type for the analysis. This is because adding more variables to our VAR model creates a dimensionality problem, where the number of parameters increases as the square of the number of variables. Given the relatively short quarterly time-series (2009:Q1 to 2020:Q1), using more than three variables is beyond the scope of our model. In addition, our demand function and accompanying consumer surplus analysis are based on download revenues and in-app purchases as a measure of consumers' willingness to pay for apps, but we are unable

[3]We note that estimating demand functions is not a micro-founded approach — discrete-choice models have been widely used in the literature to estimate consumers' preference parameters. However, with roughly half a million active apps each day in our dataset, it is a challenge to define a plausible consideration set for discrete choice. Even a single app category can include tens of thousands of apps, so a much narrower choice set needs to be devised. We thus think that there are pros and cons of these empirical approaches.

[4]The consumer surplus we calculate pertains directly to consumer demand for apps, and not to the broader economic impact of apps (e.g., in facilitating e-commerce transactions).

to take into account, for instance, a nuisance cost from mobile advertising, a potential erosion of privacy, and any other potential social and societal harms.

The remainder of this paper is organized as follows. Section 2 discusses the relevant literature, and Section 3 describes the dataset. Section 4 contains the VAR model, and Section 5 presents the consumer surplus analysis. Section 6 concludes.

2 Related Literature

There has been a growing interest in mobile platform markets in the literature. Bresnahan et al. (2014) assert that while mobile platforms lowered the costs of developing and distributing apps, the search and matching processes between consumers and the wide variety of apps may result in some inefficiencies. In particular, they argue that app rankings may fail to fully reflect the value created by an app. In the same spirit, we focus on examining the supply side of the app market, while broadening the scope of relevant technology to the various utility components made available via SDKs. We also take advantage of a longer time horizon and comprehensive data that capture the history of the iOS app market.

To be more specific, our analysis of app demand and consumer surplus builds on the examination in Bresnahan et al. of the joint distribution of app prices and monthly active users. While Bresnahan et al. are concerned with app download statistics being potentially bought or manipulated, the number of downloads in a given period is a more conservative measure than the number of active users, which reflects cumulative downloads. Further, the so-called Average Revenue Per User (ARPU), commonly used by app analytics services, is measured per download rather than per active user. Hence, we estimate app demand as well as consumer surplus based on app downloads, while using a measure of user engagement as a control.

More recently, Liu (2017) examines app developers' entry decision, selecting a mobile platform for which to develop apps. Using the average utility that users experience as a proxy for app quality, Liu finds that in the Google Play store, the presence of lower-quality apps induces more lower-quality apps to enter, while Apple's App Store exhibits stronger entries of higher-quality apps. Similarly, Ershov (2018) examines the effect of superstar apps on app entry, from the standpoint of product market competition. Using the emergence of superstar apps as unexpected events, Ershov finds that the app categories where superstar apps enter subsequently tend to experience more entry by other apps, but the new apps tend to be of lower quality. Liu's and Ershov's findings can thus provide micro-founded interpretations of the time-series shocks to app entries that occur at the platform level.

Perhaps closest to our study in this strand of the literature is Leyden (2019), who examines an app developer's product updating behavior. Leyden finds that updates increase app demand, and monetization opportunities increase the likelihood that a given update is a feature-adding update. While we only examine app and SDK entries based on the first release date and the first installed date, respectively, feature updating is likely to be related to the app developer's later adoption of SDKs. Hence, Leyden's study focuses on the surplus created at the intensive margin (i.e., post entry), while our VAR analysis focuses on the surplus created at the extensive margin (at the initial entry). On the other hand, both our analysis and Leyden's findings suggest that a share of consumer surplus created in the app market is due to the availability of new features and technologies that support them.

On the other hand, our approach and research questions are somewhat different from those in the aforementioned works, which focus on product market competition. For instance, apps need to be actively competing with each other in order to apply a discrete choice model variant for analysis. As an example, Leyden utilizes natural language processing to classify a subset of 356 ranked apps in a single (Productivity) category, to define 29 competitive submarkets. In contrast, our focus is on platform-level effects and aggregate consumer surplus, for which it may not be desirable to drop the long tail of app sales; instead, it would be important to have all possible active (e.g., downloaded) apps and SDKs. Given the different levels of granularity and identifying assumptions, these aforementioned studies are thus complementary to our focus.

The value of technology platforms has also been explored in the literature. For instance, Brynjolfsson et al. (2019) take an experimental approach. Using consumer choice surveys, they elicit the survey respondent's willingness to accept to stop using platforms such as Facebook and YouTube. Thus, they focus on access to individual platforms rather than to an overall app marketplace. Further, Facebook and YouTube (Alphabet/Google) also release SDKs that make available some of their services and features to third-party app developers, in addition to providing their own native apps. Thus, the value of tech platforms or SDK releases in spurring innovations in other apps would need to be counted, directly or indirectly, in order to obtain a comprehensive picture of the mobile app value chain.[5]

Finally, our study is generally related to the broader literature on app markets. For instance, Savage and Waldman (2015) and Kummer and Schulte (2019) show that less expensive apps tend to ask for more privacy-sensitive permissions, and that consumers are only willing to pay a few dollars at most to protect the privacy of their data. These findings suggest that free apps may, in effect, not be free, due to the potential costs of sharing some user information with the apps. This is different from in-app purchases and/or advertising which directly accrue revenues to the app developers. Thus, one might think that such a concern would matter for determining a consumer's willingness to pay. While plausible, we do not have any ready means to quantify a privacy cost for each app in our sample; thus, we prefer to use objective data for estimating a benchmark consumer surplus, which can then be pitted against potential harms from data privacy if and when such estimates become available from researchers.

There are also industry reports that relate to the economic impact of app markets. For instance, Borck et al. (2020) examine total billings and sales generated via the iOS platform in 2019, where the former refers to payments that occur on the App Store, and the latter refers to funds spent by users "outside the App Store" (e.g., e-commerce transactions). Borck et al. find that out of the total $519 billion in annual billings and sales, 80% was from e-commerce revenues and only 12% was from downloads and in-app purchases (with the remaining 8% attributed to in-app advertising revenues, which our data largely lack). We assert that the e-commerce volumes that flow through mobile apps are not the same as consumers' actual willingness to pay for the apps.[6] On the other hand, in-app advertising revenues may reflect consumers' willingness to pay, but at the same time, it may also lead to nuisance costs from the ads. Thus, under these caveats, our focus here is on the objective metrics of download prices and in-app purchases for estimating consumer surplus.

[5]Note that one could potentially survey app developers to inquire how much they value certain SDKs; however, it would be costly for researchers to survey a large number of app developers, and such a developer survey may not reveal the full surplus created by apps given that the surplus ultimately comes from the apps' users.

[6]If some of the volume of e-commerce would not have taken place through other means (e.g., a desktop or laptop Internet browser) outside the App Store, then some of the volume of sales may be attributed to the app usage. Whether that is the case and, if so, how much the extra surplus would be is beyond the scope of this paper.

3 Data

We use two datasets. The first is a time-series at the platform level, and the second is a cross-section of iOS apps. The quarterly time series comprises the number of iPhone sales, SDK releases, and iOS app releases in each quarter from 2009:Q1 to 2020:Q1. As previously mentioned, Apple smartphone devices utilize only the iOS operating system, and Apple published quarterly iPhone unit sales until it stopped doing so in its fiscal year (FY) 2019. While some third parties have published estimates of iPhone unit sales in industry reports for FY 2019 onward, for a more accurate analysis we prefer not to mix the two (estimates and official) sources of data.[7] However, we have used the industry estimates of iPhone sales post FY 2019 and found that the qualitative nature of the results does not change.

Given that the App Store opened in July 2008, data on new app and SDK releases are available from 2008:Q3; however, we decided to drop the initial two quarters because many third-party app developers and tech platforms prepared to release apps and SDKs in light of the App Store's opening, making the initial few months a rapidly growing period that may be qualitatively different from the ensuing periods. There are no official statistics for SDK and app releases. It is possible to scrape data on apps from the App Store; however, doing so only provides limited information, such as top ranks and ranges of cumulative downloads (10,000–50,000; 50,000–100,000, etc), and tracking millions of apps throughout a decade would be practically impossible for individual researchers. Therefore, we rely on a proprietary data source that collects large-scale app data from the OS, programming codes, and also from a large number of partner app publishers. Our data source is Apptopia.com. Apptopia is one of several firms providing app analytics and intelligence services for the mobile app community.[8] Apptopia's primitive data source includes real-time data feeds from more than 125,000 apps, and its database covers millions of apps (Kay, 2020). To be precise, there are 4.9 million iOS app IDs in the database with initial release dates. These numbers far exceed the quantity of currently available apps on the App Store (around 1.8 million). We thus believe that the app

[7] Android vendors (e.g., HTC, Samsung and Motorola) sold their devices preloaded with multiple OS (e.g., Windows Mobile, Symbian, Bada) and do not report unit sales. Further, industry estimates for Android phone sales disagree with each other to a substantial degree.

[8] There are other app analytics firms such as App Annie. Apptopia and App Annie were founded around the same time (2011 and 2010, respectively) and are the most well known. Our reading of various online commentaries and reviews suggests that the data quality is comparable between the two.

release data closely resemble the entire history of the app market, and we aggregate app releases by quarter.

Apptopia also provides unique SDK intelligence data based on automated script analyses. To do so, Apptopia pulls the programming codes for all free-to-download apps (which include 'freemium' apps, or apps that are free-to-download but require in-app purchases), whenever an app is newly released or a new version is made available. Apptopia does not pull the codes for all pay-to-download apps, because of the financial implications of doing so. However, over 90% of all iOS apps are free-to-download; hence, the database approximates the universe of all iOS SDKs ever released. Specifically, Apptopia tracks the date on which each SDK was installed in each app, and we identify the release quarter using the date on which an SDK was first ever installed in any app, and aggregate new SDK releases to each quarter.

Second, we use the daily cross-sections of apps in the Apptopia database during the first week of June 2020. This includes the daily number of downloads, active users, download revenues, and in-app purchase revenues for each app at the world-wide iOS platform level. We chose these dates to estimate the consumer surplus because they are two years post the VAR sample period, so the long-run result of the VAR analysis is applicable. We could readily apply our analysis to any other particular date, but we find that the estimates are not too sensitive to other nearby dates or the prior or subsequent year.

The sample universe of apps for which Apptopia tracks performance and shows one or more downloads on any given day averages about half a million apps (as of June 2020). To be more precise, Apptopia tracks all apps that were ever ranked in any app category in any country around the world. These include the aggregate rankings (e.g., top paid, top free) as well as the (sub)category rankings. Importantly, once an app is ever ranked (even on a single day), it remains in the Apptopia sample and its performance is tracked, although it may drop out of the app store chart rankings and/or even have zero downloads and no active users.

A caveat is that, just like any app analytics provider, Apptopia's app performance data is in part based on extrapolations from actual app data feeds from partner apps that share their analytics (more than 125,000 in Apptopia's case). Hence, the quality of the performance data is a function of the scope of the real data feed and the proprietary prediction algorithms that Apptopia uses. Note that inferring sales from sales ranking is often used in the literature (e.g., Brynjolfsson et al. (2003);Chevalier and Goolsbee (2003)), and Apptopia calibrates and fine-tunes their algorithm and has ample data every day to validate their model.

The performance data include an app user engagement metric: Each app has the number of daily active users (DAU) as well as monthly active users (MAU), which are the shares of the total installed user base who logged into the app over the prior 24 hours or the prior 30 days. How long a user sticks around after installing an app is a primary interest to app developers, and Apptopia sees or predicts the retention rate for each app and install cohort, and then aggregates the information. The engagement metric is then calculated as the ratio, DAU/MAU (most of which are below 0.5). We include this variable as a proxy to control for app quality.[9]

Average Revenue Per User (ARPU) refers to the download price of the app, plus the average revenue per download from In-App Purchases (IAP). IAP revenue is often associated with a business model where users pay nothing to download but are offered in-app purchases for additional, premium or sometimes necessary features. IAP revenue is defined as any payments within an app that flow through the App Store. Our understanding from Apptopia is that IAP include purchases such as game items and removing ads from an app, as well as recurring subscription fees and one-time payments processed within apps.

We note that some apps steer subscribers to their own websites,[10] in which case associated revenues would not be captured by Apptopia. Similarly, Apptopia does not have any visibility into e-commerce apps like Amazon and Walmart. However, since we estimate the consumer surplus from apps rather than from firms/websites, e-commerce revenues would not substantially affect the consumer surplus. Apptopia data would still provide valuable insight into the average app's business. Moreover, by utilizing ARPU as the effective price consumers pay for an app, our estimation of app demand is improved relative to those that do not account for revenues from IAP.

Descriptive statistics of the cross section of iOS apps are reported in Table 1. As can be expected, most of the variables have right-skewed distributions, except for the average review ratings whose distribution is left-skewed (the majority of apps have 4 or above ratings out of 5). Review ratings can be potentially manipulated and the effects are also heterogeneous.[11] Hence,

[9]While this proxy is imperfect (e.g., some apps, such as movie editing, may not be used as frequently), within a given app category it can provide a crude proxy for app quality.

[10]To a limited extent up until August 2021, as this has been a contentious issue, see, e.g., Nicas (2021); though some apps, such as e-commerce apps, may conduct their billings entirely outside of the App Store.

[11]See, e.g., Berger et al. (2010) and Mayzlin et al. (2014).

we instead use the Engagement metric as a proxy for objective app quality, which is indeed right-skewed and yields a more plausible result. Since the metric is based on user retention, it pairs well with ARPU.

4 VAR for App Ecosystem

In this section, we take a system-wide perspective to analyze the interrelationship between three time-series variables that help describe the app ecosystem, namely, smartphone sales, SDK releases, and app releases. While there is a burgeoning literature on the app market (or more generally two-side markets), the technological drivers of the app market have been relatively neglected, though there is a growing number of studies relating to it such as those mentioned in Section 2. This may be because addressing the dynamics of multiple layers of the supply side, as well as consumers, can be challenging and is, to an extent, still a developing area in the literature. Hence, this section aims to demonstrate how time-series analysis can help guide further development in the growing literature on digitization and technology platforms.

We start by briefly introducing VARs and explaining how the identifying assumptions of VARs are similar to the exclusion restriction of instrumental variable regressions that are widely used in the microeconomic empirical literature (see, e.g., Stock and Watson, 2001). A VAR is an n-equations, n-variables linear system of equations, in which each endogenous variable is explained by its own lagged values, plus current and lagged values of the remaining $n-1$ variables and an error term. This model is particularly suitable when there are multiple interrelated endogenous variables; for instance, two or three-variable VARs estimated using quarterly data have been staple specifications in macroeconomic research for forecasting and model validation. The so-called structural VAR model can be written as follows:

$$A_0 y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + e_t, \tag{1}$$

where the A's are 3×3 matrices of parameters, y_t is a 3-vector comprising the natural logs of iOS app releases, iPhone unit sales, and SDK releases in quarter t (possibly lagged by p quarters), and e_t is a 3-vector of structural 'shocks' which, by assumption, are serially uncorrelated, orthogonal disturbances (in fact, the variance-covariance matrix of shocks is normalized to an identity matrix). Pre-multiplying equation (1) with A_0^{-1} turns the system into a reduced form VAR, where each equation can be estimated by ordinary least squares (OLS):

$$y_t = A_0^{-1} A_1 y_{t-1} + ... + A_0^{-1} A_p y_{t-p} + \xi_t, \tag{2}$$

where the error terms in $\xi_t \equiv A_0^{-1} e_t$ are in general correlated across equations; however, equation by equation OLS is consistent because the error terms are uncorrelated with the lagged values of y_t. Note that the variance-covariance matrix of the error terms can be written as $\Sigma \equiv A_0^{-1} A_0^{-1\prime}$; thus, to recover the structural parameters in A_0^{-1}, we solve for the elements in A_0^{-1} given an estimate of $\hat{\Sigma}$ from the OLS, which turns out to be an over-identified system. This is where an identification problem arises.

The identification problem in VAR models can be solved by judicious use of theories and institutional knowledge, similarly to the case of instrumental variable regressions. Here, the most well known method to identify the A_0^{-1} matrix is to impose so-called 'recursive' or 'zero' restrictions, which are equivalent to computing a Choleski factorization of $\hat{\Sigma} = A_0^{-1} A_0^{-1\prime}$, where A_0^{-1} is lower triangular. Hence, the endogenous variables in y_t need to be carefully ordered, because the current (time t) value of the variable ordered later in a sequence is excluded as a regressor from the reduced-form equation for the variables ordered earlier in the sequence.[12] These restrictions essentially spell out the causal links among the variables, which allows for correlations to be interpreted causally in a VAR model.

We use two sets of identifying assumptions for robustness. The baseline assumption is to order (1) apps, (2) phones, and (3) SDKs. That is, an unexpected increase in app releases can motivate consumers and tech platforms to buy more iPhones and release more SDKs, respectively, within the same quarter. This is plausible because apps can add to the consumer's utility and app developers are the customers of SDKs. Similarly, an unexpected increase in iPhone unit sales can lead tech platforms to release more SDKs within the same quarter. Note that information about smartphone sales is a vital metric in the app market and is also readily available from market research firms, so it can induce tech platforms, for instance, to deploy

[12]Since the inverse of a lower triangular matrix is also lower triangular, recursive restrictions on A_0 mean that the variable ordered first in y_t only depends on the lagged values of the three variables, while the variable ordered second depends on the current value of the first-ordered variable as well as the lagged values of the three, and the variable ordered third depends on the current values of both first- and second-ordered variables as well as the lagged values of all three. Similarly, A_0^{-1} specifies how the structural shocks e_t affect y_t.

resources to leverage a broadened user base.

On the other hand, app developers may not be able to respond to some shocks to the system within a quarter, because it takes time (often 4–6 months) to build an average app (e.g., Yarmosh, 2019). Thus, the above ordering assumes that shocks to either iPhone unit sales or SDK releases do not affect app releases contemporaneously (within a quarter), but only through their lagged effects. Similarly, we assume that a shock to SDK releases does not affect iPhone unit sales contemporaneously because it is unlikely that consumers will take significant notice of SDKs or the underlying technology releases.

With these identifying assumptions, we estimate the above VAR model in log levels with $p = 4$ lags, as is common in macroeconomic time-series analyses that utilize VAR models (i.e., to include one-year's length of lags, or 4 lags for quarterly data). There are also statistical criteria behind this choice; for instance, Ivanov and Kilian (2005) recommend, based on simulations, that the Schwarz's Bayesian Information Criterion (SBIC) produces the most accurate impulse response estimates for small sample sizes. When calculating the selection-order statistics, where the maximum lag length can be higher than 4, we find that the SBIC further points to lag $p = 4$ as optimal.

Additionally, when we individually tested the variables in y_t for unit roots (i.e., non-stationarity), as one might expect, they exhibit a unit root. However, they also tend to co-move (or be co-integrated, so a linear combination of them is likely to be stationary), in which case first-differencing would lead to misspecifications, whereby estimating VARs in levels is robust and it yields consistent estimates. The numerous parameters in VAR models often go unreported in the literature, since the estimation results can be more informatively conveyed through impulse response functions and forecast error variance decompositions.

Impulse responses show the response of current and future values of the variables in y_t to a one-time increase in each of the structural shocks in e_t, while setting all other shocks to zero. Since our interest is in the app market, the impulse responses of app releases to each of the three shocks to the system are shown in Figure 1, along with bootstrapped one- and two-standard error bounds. (It is not uncommon in the VAR literature to use the 68% instead of 95% confidence bounds.) Hence, the solid line traces the response to one standard deviation Cholesky shocks, and given an estimate of our A_0^{-1}, it has the following interpretation: A shock to iPhone sales has no impact on app releases in the same quarter, but app releases begin to gradually increase, reaching a peak 4 quarters out, plateauing for another 4 quarters and then gradually decreasing to zero for the next 6 quarters. Hence, the effect lasts about 3 years, with the largest effect being about 5.3% after 8 quarters. Similarly, a shock to SDK releases has no impact on apps in the immediate period, but app releases begin to increase gradually, reaching a peak after 7 quarters with an effect size of 7.5%, and then decreasing back to zero by the fourth year. Thus, the response to SDKs is stronger and more persistent than to device sales.

On the other hand, a shock to apps increases app releases by about 9.8% on impact, which can be consistent with the findings in the aforementioned literature. The effect remains for 3 more quarters and from then on, the effect decreases relatively quickly to zero; hence, the effect is shorter in duration but stronger in impact than the effects from the other two shocks. Given the pace of rapidly developing technologies in the mobile ecosystem, the duration of the effects (lasting 2 to 3 years) seems to make some intuitive sense, although the impulse responses hinge on the structural assumptions regarding how the variables react to shocks. However, as we will further elaborate, these results are robust to alternative structural assumptions.

Next, a (forecast error) variance decomposition in VARs is the percentage of the variance of the error made in forecasting a variable in y_t due to shocks to a specific variable in y_t as opposed to shocks to all variables in y_t. Hence, it is a way to quantify or attribute how important each type of shock is in explaining the evolution of the variables in the system over any given time horizon (or forecasting length). Figure 2 demonstrates the results. Given short time horizons (1 year or less), app releases are basically explained by the shocks to apps themselves. Given longer time horizons (2 years or more), in contrast, a significant proportion of the variation in app releases is due to the shocks to SDK releases and, to a lesser extent, the shocks to iPhone sales.

To be more specific, after about 10 quarters into forecasting, 17.5% of the variation in quarterly app releases is explained by the shocks to iPhone sales, while 31% is explained by the shocks to SDKs. Accordingly, the percentage of the variance explained by the shocks to apps falls to a little over 50%. Thus, if we were to forecast quarterly app releases two years from a given time period, then as much as half of the variation is due to shocks that will be occurring to phone sales or SDK releases. In particular, SDK accounts for more variation in app releases than smartphone sales, highlighting the effect of technological drivers (e.g., in addition to network effects from an increasing smartphone user base).

As previously indicated, an important criticism of structural inference utilizing VAR modeling is that the variables are "backward looking" or only reactive to unanticipated shocks (i.e., the Lucas 1976 critique; see also Christiano et al. 1999). Hence, in what follows, we replicate the above VAR model using alternative identification assumptions that app developers may be "forward looking," that is, some app developers may anticipate shocks to iPhone sales or SDK releases, such as new features and functions of hardware and software, which they may use to get a head start on app development.

In terms of the Cholesky decomposition, this amounts to changing the order of the variables as follows: (1) devices, (2) SDKs, and (3) apps. We maintain the ordering between (1) and (2) because consumers are unlikely to be predicting SDK releases by technology providers. On the other hand, that app releases are now ordered last means that app developers can react within a quarter to a sudden increase in iPhone sales and/or SDK releases, because they may have been anticipating such events (e.g., app developers may obtain beta access to a new operating system). In return, consumers and tech platforms may not react within a quarter to a sudden increase in apps.[13]

The new impulse responses of app releases to the three shocks are shown in Figure 3. The general shape and the magnitude of the responses are similar to those of Figure 1. The difference is that apps now respond to contemporaneous shocks in the same quarter. Specifically, app releases decrease when there is a sudden increase in iPhone sales, which may be due to iOS updates, while apps increase when there is a sudden increase in SDK releases. This leads to an overall fall of the response to an iPhone sales shock and an overall rise of the response to a SDK shock.

This translates into the variance decomposition as shown in Figure 4. Given short time horizons (one year or less), shocks to iPhone sales and to SDK releases each account for about 10% of the variation in app releases, leaving less than the full variation to the shocks to apps. More importantly, given longer time horizons (after 10 or so quarters), about 42.5% of the variation in apps is now due to SDK shocks, while 17% is still due to shocks to iPhone sales.

[13] This may be because apps are so numerous that it takes some time for consumers and tech platforms to take notice of the sudden increase in apps. However, it is difficult to know the exact time period after which such awareness increases sufficiently to trigger a response — it is beyond the scope of this paper to test which of the two structural assumptions is more plausible than the other. Our preference is to equally weigh their findings, and for them to serve as robustness checks, given that the results are similar.

Hence, the relative importance of SDKs in explaining the evolution of app releases increased in comparison to that in Figure 2.

The reason why SDKs explain a larger part of the variation under the assumption of forward-looking developers than under backward-looking developers is that the previous VAR model did not consider the possibility that app developers anticipate technology releases and rather viewed it as an unanticipated shock. Thus, the backward-looking VAR may have underestimated the true effect of SDK shocks, which is now corrected. While we cannot empirically discern which of the two identifying assumptions is more plausible than the other, the range of inferences is similar across the two cases.

5 Consumer Surplus from Apps

The preceding analyses showed that, in addition to app developers, Apple as the platform's operator, as well as third-party technology platforms via SDKs, are responsible for building a substantial part of the app ecosystem through the distribution of smartphone devices and SDKs. However, it is unclear how much consumers value the app market, separately from the use-value of smartphones (e.g., Lee, 2018). Consumer surplus is the welfare measure government agencies often use to value goods, services, and the effects of competition, as well as to allocate economic resources. Thus, estimating the consumer surplus from the app market and attributing its sources can help us better understand the growing mobile sector, as well as help guide policymakers on the magnitude and relevant focus areas of policy objectives.

In this section, we aim to provide some empirical estimates of the consumer surplus that users derive from iOS apps. We do so by taking the neoclassical approach of estimating a market demand function, which is a mapping between the price of an app (approximated by its ARPU) and its daily number of downloads, allowing for individual app-specific intercepts.[14] We estimate three functional forms (linear, quadratic, and translog) to fit the demand function, and show that the translog specification provides a considerably better fit than the other two. We then integrate the demand function for each app from zero up to to the number of app

[14]In the app market, the effective price consumers pay is plausibly defined as the sum of the price-to-install and the average in-app purchase (IAP) per download. This is how ARPU is defined by various app analytics platforms and what we refer to as the 'price' in this section. IAP do not include product sales billed outside of the app store.

downloads to calculate the consumer surplus from each app.

This approach requires a number of simplifications. For instance, we assume that there is a ‘representative consumer’ for each app, who generates an aggregate demand function, and shares the same price elasticity within a relatively broad app category. The latter is a strong assumption and ignores the heterogeneity among apps. However, we cannot identify individual app-level elasticities because app prices rarely change. Thus, a certain level of homogeneity needs to be imposed, and we do so by estimating the demand function separately for game and non-game apps. On the other hand, we include the Engagement measure that shifts the individual apps’ demand functions within game and non-game apps; hence, we think that our framework still captures sufficient variation among apps.[15]

An alternative approach would be to consider demand for apps by using a random-utility model, which offers some advantages over the neoclassical approach because the representative consumer considers multiple apps at the same time. However, in our setting, doing so requires narrowing down the choice set of apps considerably, which is challenging given the millions of apps in our dataset. We do not in fact have much information on apps (or users) to define smaller choice sets. Further, albeit anecdotally, when we inquired with university students about their app downloading behavior, some indicated that they do not mind using 2-3 similar apps; that is, even within a small, competing set of apps, it does not necessarily appear that consumer behavior is best described by a discrete choice model at the platform level.

Hence, our goal is to estimate the following simple form of inverse market demand equations, which are linear (and quadratic) in the downloads, for game and the non-game apps, separately:

$$Price_i = \beta_0 + \beta_1 Engagement_i + \alpha_0 Downloads_i + \alpha_1 Downloads_i^2 + \epsilon_i, \tag{3}$$

where ϵ_i is independent across apps i, but need not be identically distributed. We use the ‘sandwich’ estimator of error variance, so the standard errors are robust to some model misspecification. Similarly, a translog specification can be written as follows:

$$ln(Price_i) = \beta_0 + \beta_1 Engagement_i + \alpha ln(Downloads_i) + \epsilon_i, \tag{4}$$

[15]To be precise, we also need to assume that app spending is a relatively small portion of a consumer’s budget, so there are no income effects. Further, we estimate a demand function for game and non-game apps separately, because complementarity (positive or negative) between the two types of apps seems to be unlikely.

where we do not take a log of Engagement, which is DAU/MAU, because it is relatively bell-shaped mostly between 0 and 0.5 (although taking a log of Engagement does not significantly change the results).

We aim to estimate a static market demand equation on each day, during the first week of June 2020. The primary concern in estimating a market demand function is that the price and the quantity (downloads) can be driven by omitted variables. Specifically, a higher-quality app may require a higher price and, at the same time, have more downloads. To tackle this issue, we first show a scatter plot between price and the quantity in Figure 5. The samples include all iOS apps with one or more downloads on June 1, 2020, having non-zero reviews. In order to give a closer look, the top 1% of downloads and prices are not shown in this graph, but we do not exclude any outlier from our estimation.

Figure 5 shows that most of the data points exhibit a clear negative relationship between price and quantity; however, there are indeed samples populating the upper-right region, which indicates a potential positive correlation between the two variables. Our method for addressing this is twofold. First, we include the Engagement measure as a direct control of an app's quality. Second, our identification strategy is to leverage a plausible instrument for the number of downloads, namely, the number of app reviews/ratings. We do so because finding a good instrument for the price of an app is difficult; on the other hand, we can estimate an (equivalent) inverse market demand function if there is a suitable instrument for the quantity.

That is, our identification assumption is that the number of app reviews is uncorrelated with the error term in the above inverse market demand function. This is so because it is unlikely that apps will change their download price or in-app purchase prices based on a factor that also influences the number of app reviews. For instance, users of an app may want to post positive reviews about the app being of high quality, but they may equally want to post negative reviews about the app being of low quality, so overall the number of reviews would not systematically affect the price that app publishers demand in either direction. On the other hand, the number of app downloads and the number of reviews are naturally positively correlated, so we find this identification assumption plausible.

Table 2 reports the estimation results for each functional form on June 1, 2020 (other days are similar). Given the exponential decay observed in Figure 5, it is unsurprising that the translog specification fits the data considerably better than the linear and the quadratic specifications in terms of its R^2. Note that the translog function requires the price (ARPU) to be nonzero, and we hold the nonzero-price sample constant for the other functional forms. Thus, the difference in the model fit is not driven by any difference in the sample. The estimated linear and the quadratic demand functions are too flat to capture the negative relationship between the price and the quantity, failing to explain much of the data variation.[16] Thus, we use the traslog function to estimate the consumer surplus.

On the other hand, the OLS estimate of the curvature parameter (α) is downward-biased, due to the simultaneity bias, relative to the IV estimate for non-game apps, although the OLS estimate is similar to the IV estimate in the case of game apps. Therefore, for both game and non-game apps, we use the IV specification in column (4), to estimate the consumer surplus for each app on a given day as follows: Given the estimates of $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\alpha}$, the demand function can be rewritten as $Price_i = \hat{b}_i / {Downloads_i}^{-\hat{\alpha}}$, where $\hat{b}_i = exp(\hat{\beta}_0 + \hat{\beta}_1 Engagement_i)$, which is drawn as a downward-sloping, exponential-decay curve, converging to zero from above as *Downloads* goes to infinity. The scale parameter $\hat{b}_i$ allows for heterogeneity across apps depending on their Engagement metric, while the elasticity parameter $\hat{\alpha}$ varies by game or non-game category.

The demand function can be interpreted as being ordered from the left to the right with a decreasing order of willingness to pay, so the Marshallian consumer surplus is calculated as the summation of the difference between each consumer's willingness to pay and the price (ARPU) from the first to the last unit. This area is approximated with the formula, $\hat{b}_i + \int_1^{\bar{q}_i} (\frac{\hat{b}_i}{q^{-\hat{\alpha}}}) dq - \bar{q}_i (\frac{\hat{b}_i}{\bar{q}_i^{-\hat{\alpha}}})$, where $\bar{q}_i$ is the actual number of downloads for app i. Note that the first term is the (highest) value of the first download, and the remainder of the sum can be approximated by an integral from one to the last unit $\bar{q}_i$. The last term in the formula is the producer's revenues (from downloads and in-app purchases).[17] We then repeat this procedure for six different days in the same week and aggregate the consumer surplus by game or non-game category, which is shown in Figure 6.

The figure plots the daily aggregate consumer surplus (CS) generated from about half a

[16] If we include zero-price samples in the estimations of the linear and quadratic demand functions then the results, including R^2, do not significantly change. We have also fitted a semilog function, with and without zero-price samples, and its R^2 remains below 0.01 for game apps and just above 0.01 for non-game apps.

[17] To be precise, this is the producer's surplus given a representative consumer. Each app's actual revenue will differ and deviate from this figure; however, it would not change the calculation of consumer surplus if the change in revenues uniformly comes from all downloads, whereby the demand curve shifts vertically.

million iOS apps each day during the first week of June 2020. We observe that the numbers are relatively stable across days, although the fluctuation in the monetary unit (millions of dollars) can be large. Specifically, the daily CS ranges from $45 million to $52 million in that week for the non-game category, whereas it ranges from $14 million to $16 million for the game category. As we will show below, the difference between the two categories is largely due to the difference in the number of apps being downloaded on a given day, where the former has about four times as many apps being actively downloaded as the latter.

Unless (or until) there is a drastic regime change in the mobile app ecosystem, we can reasonably assume that these figures will be more or less stable in the near term. That is, while the identities of the actively-downloaded app samples may change over time, the cross-sectional distribution of downloads and prices may remain relatively stable, as illustrated by Figure 5. Hence, we can extrapolate annual aggregate consumer surplus generated from iOS apps at the platform level by multiplying the daily figures by 365 days, which aggregates to $21 to $24 billion. Note that these are only from third-party apps, not including the apps that come pre-installed with the iPhone device.

Therefore, given the above forecast error variance decompositions and using the $21 billion annual figure for CS, about $6.5 and $8.9 billion of the aggregate annual CS from the iOS app market in and around 2020 are attributable to unexpected SDK releases under the "backward"- and "forward"-looking app developer assumptions, respectively. Similarly, about $3.7 and $3.6 billion are due to unexpected iPhone sales, and about $10.8 and $8.5 are due to unexpected app releases. These may serve as the benefit figures that can be compared to the costs of innovating in each driver of the ecosystem.

Table 3 further breaks down the consumer surplus by subcategories (as recorded in the iOS App Store). Column (1) shows how the average daily consumer surplus varies across subcategories, hence, their varying importance at the platform level (as of June 2020). However, the number of apps in each subcategory varies widely (from thousands to tens of thousand), so the consumer surplus at the subcategory level does not imply how much a typical app in each subcategory is valued. Thus, column (2) lists the average daily CS per app. Although there are notable exceptions, it can be said that the average app generates between $100 and $200 in consumer surplus for most of the subcategories, with game apps generally having a higher mean figure than non-game apps.

Columns (3) and (4) of Table 3 show the top 3 and top 10 consumer surplus concentrations in each subcategory. For non-game apps, with the exception of category (1), which groups several small-sample subcategories, the consumer surplus does not appear to be concentrated in the top apps under either measure, though they are still have disproportionately large shares given the large number of apps in each subcategory. On the other hand, game subcategories have consumer surpluses relatively more concentrated in top game apps than non-game subcategories. This suggests that games may be riskier for new app developments than non-games (i.e., more difficult to succeed, but success entails a larger reward). This also implies that top game apps may have more bargaining power vis-à-vis the platform than non-game apps.

6 Conclusion

We examined the main factors that describe the mobile app ecosystem, using expansive, proprietary data that approximate sector-level historical statistics as well as a cross section of iOS apps. We found that technology (SDK) releases by third-party platforms account for app releases, moreso than hardware (smartphone) sales. In particular, an unexpected shock to SDK releases and to phone sales boost app releases for about three years, while a shock to app releases does so for less than two years, despite having a stronger initial effect.

We found that the iOS app market collectively generates around \$24 billion per year (as of 2020) in aggregate consumer surplus. We do not know how many iPhone users participated in the app market in 2020, but a rough back-of-the-envelop calculation would be to use a range between the 195 million estimated iPhone device sales in 2020 (Graham, 2020) and the estimated 1 billion active iPhone devices as of the end of 2020 (Kastrenakes, 2021). Then the annual consumer surplus from apps would be between \$24 and \$123 on a per-user basis, which is lower than the price of an iPhone, but over its useful lifetime can comprise a significant share of the device price.

References

Borck, J., J. Caminade, and M. von Wartburg (2020): "How Large Is the Apple App Store Ecosystem?" Accessed August 1, 2021. https://www.apple.com/newsroom/pdfs/

app-store-study-2019.pdf.

Berger, J., A. T. Sorensen, and S. J. Rasmussen (2010): "Positive Effects of Negative Publicity: When Negative Reviews Increase Sales," *Marketing Science*, 29, 815–827.

Bresnahan, T. F., J. P. Davis, and P.-L. Yin (2014): "Economic Value Creation in Mobile Applications," *The Changing Frontier: Rethinking Science and Innovation Policy*, 233–286.

Brynjolfsson, E., A. Collis, and F. Eggers (2019): "Using Massive Online Choice Experiments To Measure Changes in Well-being," *Proceedings of the National Academy of Sciences*, 116, 7250–7255.

Brynjolfsson, E., Y. Hu, and M. D. Smith (2003): "Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers," *Management Science*, 49, 1580–1596.

Christiano, L. J., M. Eichenbaum, and C. L. Evans (1999): "Monetary Policy Shocks: What Have We Learned and to What End?" *Handbook of Macroeconomics*, 1, 65–148.

Chevalier, J. and A. Goolsbee (2003): "Measuring Prices and Price Competition Online: Amazon.com and BarnesandNoble.com," *Quantitative Marketing and Economics*, 1, 203–222.

Clawson, M. and J. L. Knetsch (1966): "Economics of Outdoor Recreation," Baltimore: Johns Hopkins Press.

Ershov, D. (2018): "Competing with Superstars in the Mobile App Market," Working Paper.

Graham, J. (2020): "Apple iPhone again Best Tech Seller of the Year, Thanks to the Working-from-home Trend," Accessed August 1, 2021. https://www.usatoday.com/story/tech/2020/12/28/apple-iphone-best-selling-tech-device-year-work-home-helped/4048111001/.

Hotelling, H. (1938): "The General Welfare in Relation to Problems of Taxation and of Railway and Utility Rates," *Econometrica*, 6, 242–269.

Ivanov, V. and L. Kilian (2005): "A Practitioner's Guide to Lag Order Selection for VAR Impulse Response Analysis," *Studies in Nonlinear Dynamics & Econometrics*, 9, Article 2.

Kastrenakes, J. (2021): "Apple says there are now over 1 billion active iPhones," Accessed August 1, 2021. https://www.theverge.com/2021/1/27/22253162/iphone-users-total-number-billion-apple-tim-cook-q1-2021.

Kay, J. (2020): "Apptopia's Data Algorithms Explained," Accessed August 1, 2021. https://medium.com/@Jon_35384/apptopias-data-algorithms-explained-64b65ed11d56.

Kummer, M. and P. Schulte (2019): "When Private Information Settles the Bill: Money and Privacy in Google's Market for Smartphone Applications," *Management Science*, 65, 3470–3469.

Lee, S. (2018): "Quantifying the Consumer Surplus from Smartphones," Working Paper.

Leyden, B. T. (2019): "There's an App (Update) for That: Understanding Product Updating Under Digitization," Working Paper.

Liu, Y. (2017): "Mobile App Platform Choice," Working Paper.

Lucas, R. E. (1976): "Econometric Policy Evaluation: A Critique," *Carnegie-Rochester Conference Series on Public Policy*, 1, 19–46.

Mayzlin, D., Y. Dover, and J. Chevalier (2014): "Promotional Reviews: An Empirical Investigation of Online Review Manipulation," *American Economic Review*, 104, 2421–2455.

NetMarketShare (2020): "Operating System Market Share," Accessed August 1, 2021. https://netmarketshare.com/operating-system-market-share.aspx.

Nicas, J. (2021): "Apple Avoids Another Legal Fight With App Developers," Accessed August 26, 2021. https://www.nytimes.com/2021/08/27/technology/apple-settles-app-store-lawsuit.html.

Savage, S. J. and D. M. Waldman (2015): "Privacy Tradeoffs in Smartphone Applications," *Economics Letters*, 137, 171–175.

Sims, C. A. (1980): "Macroeconomics and Reality," *Econometrica*, 48, 1–48.

Stock, J. H. and M. W. Watson (2001): "Vector Autoregressions," *Journal of Economic Perspectives*, 15, 101–115.

Yarmosh, K. (2019): "How Long Does it Take to Make an App?" Accessed August 1, 2021, https://savvyapps.com/blog/how-long-does-it-take-to-make-an-app.

Figure 1: Impulse Response of App Releases (Log Diff.), Model 1

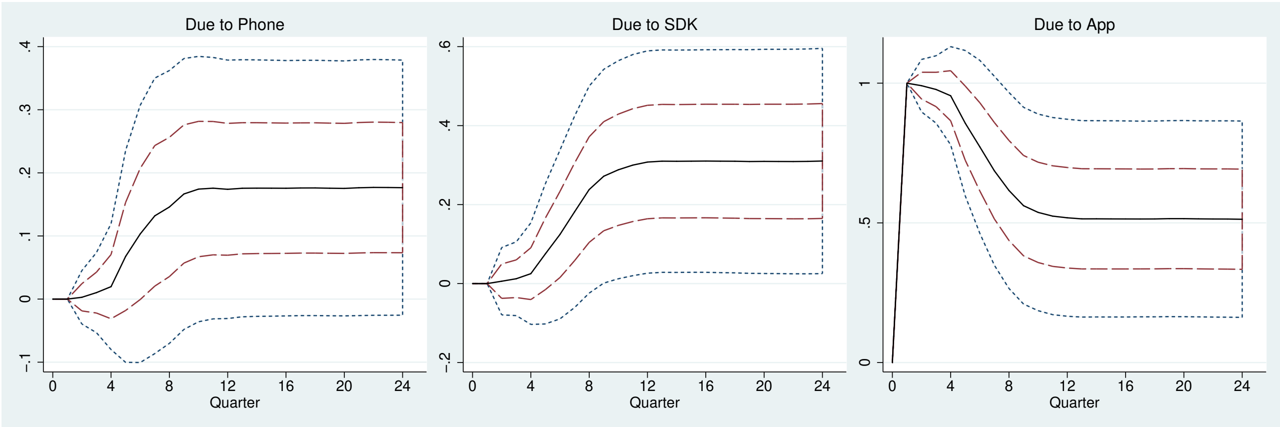

Figure 2: Variance Decomposition of App Releases (% Point), Model 1

Figure 3: Impulse Response of App Releases (Log Diff.), Model 2

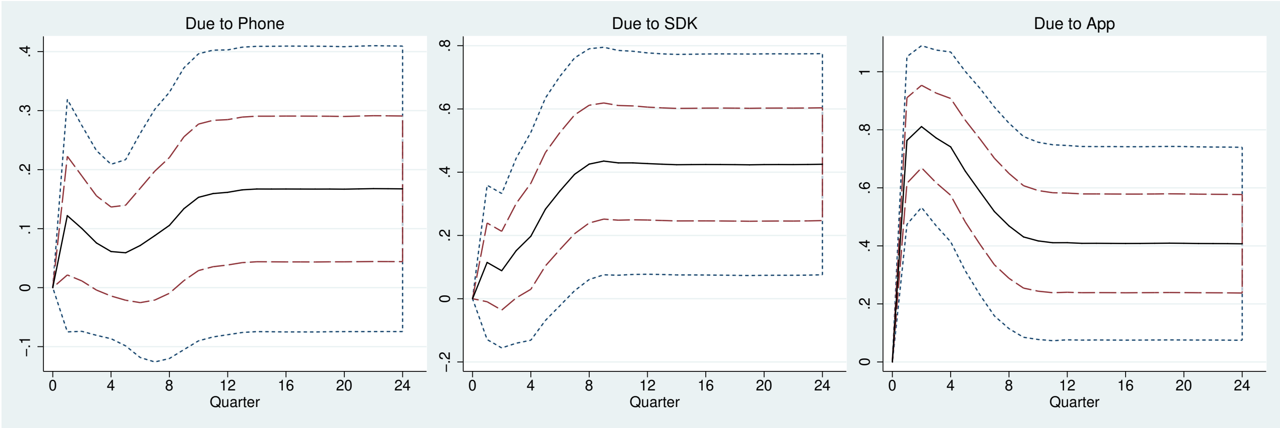

Figure 4: Variance Decomposition of App Releases (% Point), Model 2

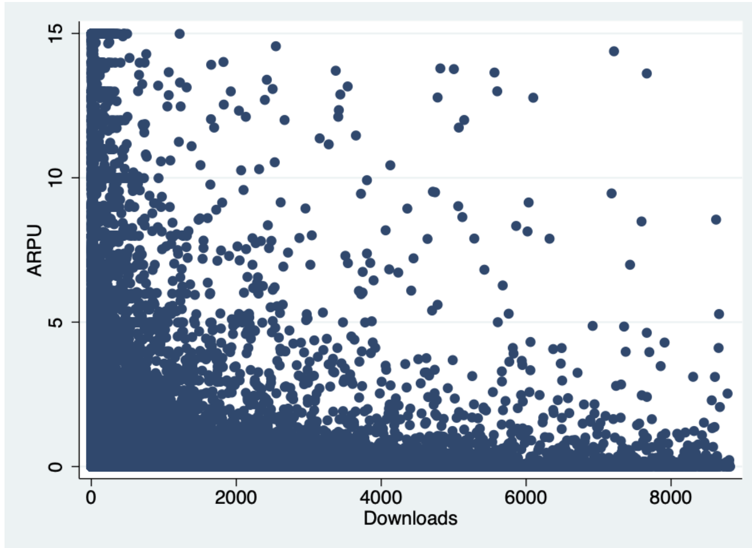

Figure 5: Scatter Plot for iOS Apps (on June 1, 2020)

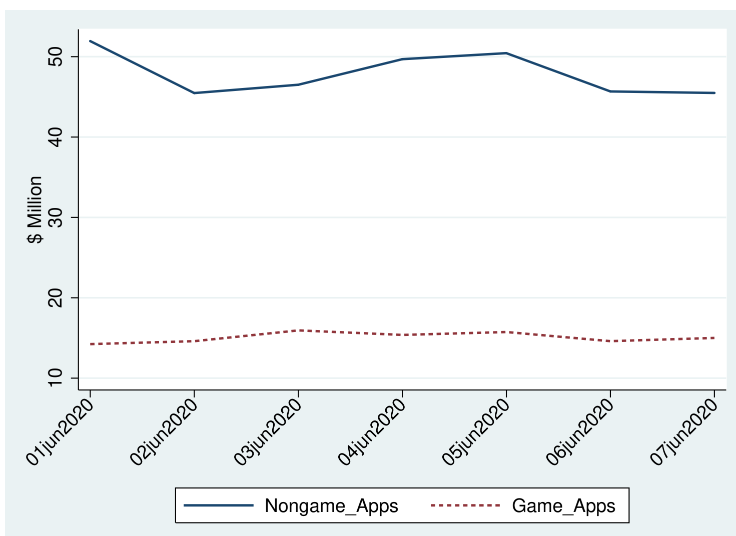

Figure 6: Consumer Surplus from iOS Apps (June 2020)

Table 1: Cross Sectional Descriptive Statistics

	Mean	Std. dev.	Min.	Max.
Downloads	261.121	2668.75	1	483,047
ARPU	1.11190	14.5417	0	2290
Engagement	.131682	.107397	0	1
Avg. rating	3.84646	.976059	1	5
Review count	7072.38	207,258	0	5.92e+07

Note: The samples include all actively downloading iOS apps on a day in June 2020 (some statistics are also redacted to protect the business interests of Apptopia). The summary statistics are similar for other days.

Table 2: Inverse Market Demand Function

	(1) Linear OLS	(2) Quadratic OLS	(3) Translog OLS	(4) Translog IV
Non-game				
Constant	3.490347	3.462388	1.543188	1.984424
	(.2224052)***	(.2227526)***	(.0233846)***	(.0374167)***
Engagement	-1.128799	-.7312506	4.389221	5.496077
	(.9179982)	(.9257075)	(.0957037)***	(.1222755)***
Downloads	-.0000489	-.0001414		
	(.0000105)***	(.0000126)***		
Downloads^2		4.28e-10		
		(6.79e-11)***		
ln(Downloads)			-.5859702	-.7330117
			(.0056905)***	(.0112855)***
R^2	0.0005	0.0010	0.2242	
Game				
Constant	1.08856	1.088331	.5145571	.4496219
	(.1311033)***	(.1310778)***	(.0425835)***	(.0720374)***
Engagement	4.9948	5.236118	7.365729	7.249287
	(.9147323)***	(.9341817)***	(.1740566)***	(.2124915)***
Downloads	-.0000332	-.0000858		
	(8.26e-06)***	(.0000174)***		
Downloads^2		6.49e-10		
		(1.47e-10)***		
ln(Downloads)			-.5541422	-.5378301
			(.0081449)***	(.0169854)***
R^2	0.0025	0.0028	0.1988	

Note: The sample used is the cross section of iOS apps on June 1, 2020. (Other dates are similarly estimated.) Heteroscedasticity-consistent standard errors are shown in parentheses. Statistical significance is denoted by *** for the 1% level, ** for the 5% level, and * for the 10% level.

Table 3: Average Daily Consumer Surplus

	(1) Total CS (in US $)	(2) Per-app CS (in US $)	(3) Top 3 CS (% share)	(4) Top 10 CS (% share)
Non-game				
Business	2,185,516	70	1.43%	4.01%
Weather	901,341	190	5.38%	8.18%
Utilities	3,385,170	124	2.30%	6.12%
Travel	2,795,638	111	2.04%	4.64%
Sports	2,697,196	148	1.61%	4.65%
Social Networking	2,172,389	158	3.35%	8.18%
Reference	1,548,377	108	2.74%	7.21%
Productivity	1,941,602	97	3.40%	7.14%
Photo & Video	1,953,588	138	6.71%	10.85%
News	1,783,924	121	1.91%	4.71%
Navigation	1,435,066	114	8.88%	11.40%
Music	1,843,912	112	6.48%	12.14%
Lifestyle	3,127,041	110	1.97%	4.69%
Health & Fitness	2,122,936	96	2.83%	5.39%
Finance	3,659,411	152	2.74%	5.37%
Entertainment	3,044,374	136	2.66%	6.50%
Education	3,045,836	84	1.13%	3.08%
Book	1,617,822	125	2.33%	6.17%
Medical	1,264,767	82	4.09%	5.79%
Food & Drink	2,200,185	91	1.34%	3.60%
Shopping	3,123,773	139	4.06%	9.21%
(1)	39,885	108	41.42%	48.64%
Game				
Action	1,660,066	190	15.95%	28.51%
Adventure	1,051,731	133	14.96%	32.17%
Arcade	2,085,688	226	11.32%	24.11%
Board	504,771	84	21.99%	31.29%
Card	648,323	143	8.58%	21.42%
Casino	346,525	116	12.85%	26.45%
Family	631,956	66	20.45%	34.77%
Music	213,790	93	19.00%	31.66%
Puzzle	1,163,785	95	12.39%	28.12%
Racing	312,008	71	17.48%	33.55%
Role Playing	1,543,409	232	26.92%	41.78%
Simulation	1,408,147	137	14.12%	31.19%
Sports	437,515	106	22.50%	36.79%
Strategy	893,824	163	13.46%	28.04%
Trivia	854,491	147	14.45%	39.52%
Word	462,860	93	27.18%	43.90%
(2)	854,404	1236	11.13%	31.46%

Note: Figures are averaged across 7 days (June 1, 2020 to June 7, 2020). (1) Four non-app subcategories (Magazines, Catalogs, Dev Tools, Graphics) are combined due to small sample sizes. (2) Game apps that have no subcategory information are collected.