CESifo WORKING PAPERS

9113
2021

Original Version: May 2021
This Version: May 2023

Platform Design and Innovation Incentives: Evidence from the Product Rating System on Apple's App Store

Benjamin T. Leyden

CESifo

Impressum:

CESifo Working Papers
ISSN 2364-1428 (electronic version)
Publisher and distributor: Munich Society for the Promotion of Economic Research - CESifo GmbH
The international platform of Ludwigs-Maximilians University's Center for Economic Studies and the ifo Institute
Poschingerstr. 5, 81679 Munich, Germany
Telephone +49 (0)89 2180-2740, Telefax +49 (0)89 2180-17845, email office@cesifo.de
Editor: Clemens Fuest
https://www.cesifo.org/en/wp
An electronic version of the paper may be downloaded
· from the SSRN website: www.SSRN.com
· from the RePEc website: www.RePEc.org
· from the CESifo website: https://www.cesifo.org/en/wp

CESifo *Working Paper No. 9113*

Platform Design and Innovation Incentives: Evidence from the Product Rating System on Apple's App Store

Abstract

I study how the aggregation of product ratings on digital platforms affects the strategic behavior of third-party firms. I leverage an unexpected and exogenous change in the rating system on Apple's App Store marketplace to show that for nearly a decade, the manner in which the App Store aggregated customer ratings led to less frequent product updating by developers. I show that products that were more reliant on the rating system were more responsive to the incentives created by this policy, and I provide suggestive evidence that this policy led to a decrease in developer effort on the platform.

JEL-Codes: L150, L860, O300.

Keywords: reputation systems, online platforms, product innovation.

Benjamin T. Leyden
Dyson School of Applied Economics and Management
Cornell University / Ithaca / NY / USA
leyden@cornell.edu

May 16, 2023
I am grateful to Gaurab Aryal, Lukas Jürgensmeier, Ambre Nicolle, Michael Ward, Georgios Zervas, and seminar participants at the Centre for Competition Policy at the University of East Anglia, the Toulouse School of Economics Online Seminar on the Economics of Platforms, the CESifo Area Conference on the Economics of Digitization, the Toulouse School of Economics 14th Digital Economics Conference, the 12th Paris Conference on Digital Economics, the 19th Annual International Industrial Organization Conference, the 20th ZEW Conference on the Economics of Information and Communication, and the 2022 NBER IT and Digitization Summer Institute for helpful comments.

1 Introduction

Product rating and review systems are a commonly implemented solution on digital platforms for the information asymmetries that exist between one side of the platform and another. The introduction of such a system, however, trades an information asymmetry problem for an information aggregation problem. Given customers are unlikely to wade through an endless list of ratings and reviews, platforms must choose how to aggregate and present this information. There is reason to believe platforms may not always convey this information in an optimal manner as doing so in the presence of reviewer heterogeneity, reputation inflation, and other concerns can be difficult (Dai et al., 2018; Filippas, Horton, and Golden, 2022). Additionally, Lizzeri (1999) has shown in the context of certifying intermediaries that a monopoly platform might optimally choose to not provide full information, even absent other concerns. Ultimately, the manner in which platforms aggregate and present product ratings has the potential to shape the competitive behavior of platform participants. In this paper, I study how the aggregation of information under these systems can affect firms' strategic behavior.

Using data on product ratings and development from Apple's App Store marketplace, I show that the manner in which ratings information is aggregated can have a real and economically meaningful effect on the strategic behavior of firms. In particular, I show that a longstanding policy governing the aggregation of product ratings on the App Store led developers to engage in less frequent product updating. Additionally, I find that developers who are likely to be more reliant on the platform's reputation system showed signs of a greater discouragement effect. Finally, I provide suggestive evidence that the policy led developers to decrease their effort on the platform and produce fewer product updates, versus exerting the same level of effort and bundling updates into less frequent releases.

My analysis centers on a longstanding policy on Apple's App Store regarding how product ratings are aggregated. For nearly a decade, the App Store would reset an app's salient average rating each time the app was updated, which, in theory, discouraged product updating from higher-quality apps, and potentially encouraged more frequent updating from lower-quality apps. I leverage an exogenous—from the perspective of an app developer—change in this policy to show that the policy indeed led to a suboptimal level of product updating among higher-quality products. While measuring the welfare impact of this policy using the empirical methods employed herein is difficult, I use data on the content of the product updates to provide suggestive evidence that in many, although likely not all cases, this policy resulted in lost, and not just delayed, product updating. These findings provide evidence that the design of a rating system can have a first-order effect on intra-platform competitive behavior, and, ultimately, may lead to socially suboptimal outcomes.

There has been a significant amount of research in recent years about the role and importance of rating systems (see, e.g., Jin and Leslie (2003); Ghose, Ipeirotis, and Li (2014); Mayzlin, Dover, and Chevalier (2014); Luca (2016); Luca and Zervas (2016); Vana and Lambrecht (2021); Acemoglu et al. (2022)), but to date, there is relatively limited research on how the design of rating systems affects firm behavior. Early work in this area focused on the relationship between reputation and

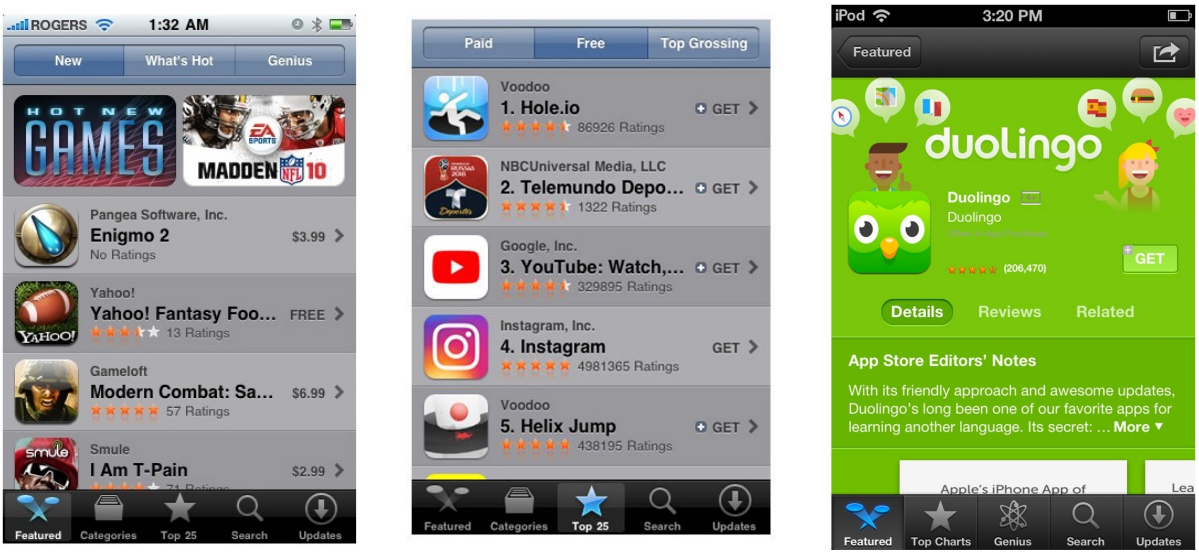
market participation. In the context of eBay, Cabral and Hortaçsu (2004) provide evidence that prospective sellers may first build a reputation as a buyer, and then leverage that reputation as they increasingly engage in sales. In related work, Cabral and Hortaçsu (2010) find that lower-rated eBay sellers are more likely to exit the platform (often amidst a flurry of negative reviews), although Klein, Lambertz, and Stahl (2016) show that when eBay improved the quality of its review system, there was no change in the composition of sellers on the platform, and that seller effort likely increased following the design change.

Other work has focused on how ratings can affect seller behavior, and particularly seller effort on the platform. Davis, Chhabra, and Yin (2016) provide evidence that multi-product developers reallocate effort across products based on the feedback they receive via user ratings. Hui et al. (2018) show that a change in the standard for being labeled a top seller on eBay led to a bifurcated response by incumbent sellers, with some engaging in greater effort to improve quality and stay above the bar, while others reduced effort, having determined the label was not worthwhile. Klein, Lambertz, and Stahl (2016) also find evidence of quality improvements following the policy change they study. Hunter (2020) shows that firms engage in short-run gaming of ratings in a context where average ratings are rounded to the nearest half-star. Ananthakrishnan, Proserpio, and Sharma (2023) present evidence that hotels whose managers show signs of engagement with online reviews are more likely to improve in quality and that subsequent quality improvements appear to address concerns raised in prior reviews. Finally, Huang, Li, and Zuo (2022) study how high-quality sellers can leverage a low introductory price to both signal their quality and quickly build a reputation.

Beyond rating and review systems, there is growing evidence about the role a platform's design can have on competitive outcomes. Claussen, Kretschmer, and Mayrhofer (2013) study a policy change on Facebook that made the number of notifications an app on the platform could send a function of how compelling (proxied by user engagement) the app's past notifications were. Meant to discourage spam notifications, Claussen, Kretschmer, and Mayrhofer provide evidence that this policy change led to an increase in the average quality of an app on Facebook. In the context of Google's mobile app store, the Google Play Store, Ershov (2022) shows that the introduction of new product categories lowered search costs and resulted in increased entry. Ershov finds that the change to the platform's organizational structure led to welfare increases, although those effects are mitigated by both a decline in the quality of the marginal entrant and an increase in congestion on the platform. Finally, Jia, Jin, and Wagman (2021) investigate how changes in platform regulations by Airbnb affect rental listings on the platform and its competitor VRBO, as well as how competition between platforms interacts with these effects.

This paper contributes to these areas of research by showing how a specific platform design decision—how product ratings are calculated and displayed—can distort the innovative behavior of the firms competing on the platform. This builds on two recent papers that suggest platform design can have a meaningful impact on innovation by platform participants. Using data on Apple's App Store and the Google Play Store, Comino, Manenti, and Mariuzzo (2018) argue that the level of

Figure 1: Display of App Ratings



(a) App Store Homepage

(b) "Free" Ranking List

(c) App Storefront

Reproduced with permission from Hackett (2018)

quality control by a platform can affect the returns to product updating, and therefore affect the incentive to engage in product innovation. Leyden (2022) shows that new technologies provided to developers by Apple's mobile platform, namely the ability to push automatic updates to consumers, as well as to monetize product use via ads, add-on purchases, and other means, leads to both increased and more substantial product innovation.

2 Apple's App Store and the Policy Change

Introduced in 2008, Apple's App Store is the exclusive marketplace for buying and selling software for Apple's mobile products.[1] The market is large—with an estimated 700 million iPhones in use worldwide in 2017, the year of the rating policy change, developers earned approximately $26.5 billion dollars worldwide (Staff, 2017; Apple, 2018).

The App Store employs a product review system in order to better inform potential customers about the quality of an app. Customers have the ability to rate and/or review apps they have previously purchased on an integer, 1-5 star scale. All text reviews are also accompanied by a rating on this scale. An app's average rating is displayed to users in search results, app ranking lists, and on the app's individual store page. Figure 1 provides examples of these displays. In cases where an app has not yet received at least five ratings the phrase "No Ratings" is displayed (see, "Enigmo 2" in Figure 1a).

When the App Store debuted in July 2008, the salient user ratings for an app were those for

[1] In addition to the iPhone, the App Store is the exclusive software marketplace for Apple's tablet and smartwatch platforms, and an optional marketplace for the Mac computing platform.

the app's *current* version. As a result, any update to the app by the developer automatically resets the score to "No Ratings." This *ratings reset policy* received significant criticism from developers, who felt it penalized high-quality work by imposing an additional, reputational cost of updating. As one developer put it in 2014,

> "With App Store ratings getting reset on every minor version, we are actively discouraged from updating our apps once they have good ratings. I've just submitted a new version of Delicious Library 3, and I'm scared out of my head that the first two or three people who review it won't like it, which will tank it to the extent where nobody discovers it any more, so there won't be any positive reviews to balance them out."
>
> — Wil Shipley, Developer (Shipley, 2014)

In July 2017, Apple unexpectedly announced plans to change this policy, with the executive in charge of the App Store explaining,

> "... some developers don't like submitting their updates, because it resets the rating. So they would get upset saying, 'Oh, man, I have a choice, fix some bugs and blow away my ratings or keep my high rating. I have a 4.7, I don't want to submit it. And I thought that was kind of stupid."
>
> — Philip Schiller, Apple SVP of Worldwide Marketing (Gruber, 2017)

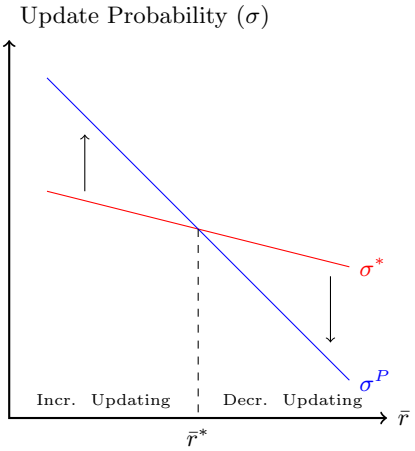
Apple's new policy, enacted with the release of the iOS 11 operating system on September 19, 2017, gave developers the option of resetting their average rating with a product update—consistent with the original policy—or keeping their existing score.

3 The Effect of the Ratings Policy on App Updating

In this section, I outline the expected impact of Apple's rating reset policy (henceforth, reset policy) on app developers' product updating decisions. Consider an app developer, with the developer and their app indexed by j. Developers maximize their discounted profit stream by choosing each period whether to update their app. A developer's per-period profits depend on the revenue the app earns in period t and the fixed cost of producing an app update, should the developer decide to update their product. I assume that updates increase demand, and therefore revenue, which is consistent with prior empirical work (Ghose and Han, 2014; Comino, Manenti, and Mariuzzo, 2018; Leyden, 2022). In each period, developers weigh the benefits associated with an update (increased revenue in the current and, possibly, future periods) against its cost. With this in mind, we can consider a developer's optimal updating policy, which I represent by the probability of updating in a given period, $\sigma^*_{j,t}$.

Next, consider a rating system where users have the option of rating the app. The existence of a rating system slightly modifies the developer's original optimization problem. To understand how, first note that a number of studies have shown that ratings affect demand (Jin and Leslie, 2003;

Figure 2: Effect of Review Reset Policy on Updating Decisions



A policy that resets product ratings with each app update will distort developers' updating behavior from their optimal updating policy, σ^*, to σ^P, which reflects the encouraging effect of this policy to apps with an average rating $\bar{r} < \bar{r}^*$, and the discouraging effect for apps with an average rating $\bar{r} > \bar{r}^*$.

Chevalier and Mayzlin, 2006; Luca, 2016; Luca and Zervas, 2016; Jabr et al., 2022).[2] Furthermore, if ratings reflect (at least on average) the quality of the product, an app update will have both a direct, positive effect on profits, and an indirect, positive effect on profits via its effect on the product's future average rating. Thus, the firm's updating decision may depend on the value of its average rating, $\bar{r}_{j,t}$ at the time of the update decision. Without loss of generality, I assume that this relationship is downward sloping ($\frac{\partial \sigma^*}{\partial \bar{r}} < 0$). That is, I assume that apps with a higher average rating are less likely to update.

To understand how Apple's original policy of resetting an app's rating after an update affects a developer's product updating decision, we can focus our analysis on how the policy might affect the slope of the firm's updating policy function σ with respect to $\bar{r}$ ($\frac{\partial \sigma}{\partial \bar{r}}$). If a policy is enacted that resets ratings with each product update, then when an app updates it will lose whatever amount of demand was due to its prior average rating. For example, if a 5-star average rating causes a 10% increase in demand (relative to a baseline of no rating), then the first-order effect of resetting ratings will be to lose that 10% boost in demand. Of course, the update will still increase the quality of the product (at least on average), therefore increasing demand (and future average ratings), and so many developers will still update. Overall, if ratings have a positive effect on demand then developers of highly rated products will become less likely to update, all else equal, in the presence of the reputational penalty imposed by this policy.

The effect of this policy on developers of lower-rated apps is less clear and ultimately depends on the consumer response to a product with no ratings. One possibility is that consumers will treat a product with no ratings as if it has an average rating of 0. If this is the case, then we can expect the reset policy to result in a reduction in updating across all products. Alternatively, if consumers

[2]Consistent with the mechanism developed in this section, Jabr et al. (2022) provide evidence that low-reputation (high-reputation) apps on the App Store saw their demand increase (decrease) following update-induced rating resets.

view especially low ratings as a stronger signal of low quality than no ratings, then the policy would actually encourage increased updating by low-quality apps, as the rating reset would boost demand for the product (at least until new ratings arrive). If we define $\bar{r}^*$ to be the average rating that consumers view as equivalent to having no ratings, then we can expect that Apple's reset policy will *discourage* updating for apps rated $\bar{r}_{j,t} > \bar{r}^*$, and *encourage* updating for apps rated $\bar{r}_{j,t} < \bar{r}^*$, relative to the optimal updating frequency σ^*.[3] This response is illustrated in Figure 2.

As discussed in Section 2, the policy change that was enacted in 2017 with the release of iOS 11 gave developers the option of keeping or resetting their ratings with each update. This preserves any potential benefits of the old policy, which would accrue to apps rated $\bar{r}_{j,t} < \bar{r}^*$, but eliminates any discouraging effect for apps rated $\bar{r}_{j,t} > \bar{r}^*$ (assuming developers optimally select when to reset their ratings). Therefore, we should expect to see the update response function flatten with respect to $\bar{r}$ after the policy change for highly-rated apps ($\bar{r} > \bar{r}^*$), as higher-rated apps become relatively more likely to update, but remain steeper than σ^* for lower-rated apps ($\bar{r} < \bar{r}^*$).

4 Data

To better understand how Apple's rating reset policy, enacted on September 19, 2017, affected product updating, I collected weekly app characteristic and ratings data for all products on Apple's App Store marketplace beginning on September 19, 2016, and ending on September 19, 2018, from the data provider Airnow Data (Airnow, 2016). A primary challenge in defining a sample for this analysis is that many of the apps on the App Store are not actively engaged in competition, having been previously abandoned by the developer or produced by a hobbyist. To identify the set of apps that are competitively active on the platform, I use Apple's three category-specific, daily product ranking lists. Within each product category, these lists rank 1) free apps based on new downloads, 2) paid (i.e., not free) apps based on new downloads, and 3) all apps based on gross revenue.[4] I construct my sample using all apps that have maintained a ranking of 200^{th} or better on one or more of these lists for at least four weeks, and at least 25% of the time that the app is available on the store during the sample period, and average at least one user rating per month.

In Table 1, I present summary statistics on the likelihood of product updates and on product ratings broken down by product category and by the period before and after the policy change. Overall, developers update their products roughly once every two months, although this average varies from as low as roughly once every four months (Reference apps) to once every five weeks (Shopping apps). The overall likelihood of updating increases following the policy change. As is typical in online markets, average ratings are skewed toward the high-end, with an average overall rating of 4 stars. Finally, apps are labeled as having "No Rating" in 17.9% of observations. This rate fell from 30.3% to 6.7% following the policy change. This sharp decline is consistent with the

[3] Note that the previous case, where the "No Ratings" display is viewed as a rating of zero, is a special case of this approach, with $\bar{r}^* = 0$.

[4] The "top grossing" lists account for all revenue processed directly through the platform and thus exclude most advertising revenue.

Figure 3: Release Notes Update Classification Training Rules

Feature Update	**Bug Fix Update**
• Adds additional functionality.	• Any direct mention of fixing bugs.
• Adds additional content.	• Performance improvements or anything indicating "under-the-hood improvements."
• Adds support for a new language.	• Adjustments to maintain compatibility with the latest version of iOS.
	• Changes to existing functionality that imply minor improvements to existing functionality, not the addition of more functionality.

This figure, reproduced from Leyden (2022), outlines the classification rules under which the SVM training set was hand-coded.

developer sentiments regarding the reset policy discussed in Section 2.

In addition to studying the product updating decisions by developers, I also consider the content of those updates. I classify the content of updates in two ways.

Version Number (VN) Classification First, I follow a traditional method of classifying software updates using each product's "version number." When a developer updates their app, they increment the product's version number, and the manner in which these numbers change can provide a proxy for the content of the update. Traditionally, changes to the number before the first decimal of a version number (e.g., from 12.1 to 13.0) indicates a "Major" revision of the product, while changes of any of the subsequent numbers (e.g., 12.1 to 12.2, or 12.1.1 to 12.1.2) indicate a "Minor" update. However, this measure is a noisy proxy of update content because the App Store does not enforce a formal versioning nomenclature, and while the approach described above is historically common, developers are free to use alternative approaches. Indeed, a new approach that is gaining popularity on Apple's platform is to set version numbers using the year and an incrementing count (e.g., 2020.1, 2020.2, etc.)—an approach that conveys no information about the content of an update.

Release Notes (RN) Classification To address these potential concerns with the VN classification approach, I take advantage of the fact that developers also provide a written set of "release notes" with each update, documenting how the update changes the product. To do so, I employ the approach developed in Leyden (2022) to classify the content of updates using these documents.

Table 1: Summary Statistics

	Update		No Ratings		Average Rating ($\bar{r} > 0$)		
	Mean	SD	Mean	SD	Mean	SD	N
Category							
Books	0.102	0.303	0.247	0.431	4.178	0.820	53,041
Business	0.148	0.355	0.243	0.429	3.834	1.038	67,022
Education	0.105	0.306	0.156	0.363	4.043	0.792	84,373
Entertainment	0.104	0.305	0.121	0.326	3.850	0.914	90,630
Finance	0.152	0.359	0.205	0.404	3.912	1.011	61,192
Food and Drink	0.136	0.343	0.237	0.425	3.808	1.055	44,158
Games	0.175	0.380	0.036	0.186	4.314	0.490	118,726
Health and Fitness	0.137	0.344	0.140	0.347	4.257	0.790	80,159
Lifestyle	0.130	0.336	0.203	0.402	3.905	0.987	81,607
Medical	0.106	0.308	0.272	0.445	4.160	0.860	49,766
Music	0.112	0.315	0.165	0.371	4.102	0.790	70,036
Navigation	0.121	0.326	0.301	0.459	3.857	0.951	45,282
News	0.132	0.338	0.248	0.432	3.789	1.026	62,813
Photo and Video	0.128	0.334	0.109	0.312	4.181	0.767	84,063
Productivity	0.133	0.339	0.142	0.349	4.079	0.791	81,815
Reference	0.073	0.260	0.171	0.376	4.222	0.835	53,829
Shopping	0.212	0.409	0.162	0.369	3.996	0.963	45,512
Social Networking	0.197	0.398	0.190	0.393	3.805	0.948	61,491
Sports	0.131	0.337	0.274	0.446	3.749	1.052	59,307
Travel	0.159	0.366	0.234	0.424	3.953	1.005	59,418
Utilities	0.092	0.289	0.126	0.332	3.882	0.920	82,954
Weather	0.094	0.292	0.288	0.453	4.075	0.835	46,623
Time Period							
Pre-iOS 11	0.127	0.333	0.303	0.459	3.996	0.947	707,583
Post-iOS 11	0.135	0.342	0.067	0.249	4.035	0.853	776,234
Total	0.131	0.338	0.179	0.384	4.020	0.893	1,483,817

Table 2: Release Notes Classification Statistics

(a) Update Classification Precision and Recall

	Precision	Recall	N
Bug Fix	0.96	0.88	207
Feature	0.55	0.82	38
Weighted Average/Total	0.90	0.87	245

Precision indicates the ratio of the number of correctly predicted cases for a given update type to the total number of cases predicted to be of that type. Recall is the ratio of the number of correctly predicted cases for a given update type to the total number of cases of that type.

(b) Percentage of Updates, by Type

	Minor	Major	Total
Bug Fix	56.90	2.80	59.60
Feature	36.70	3.70	40.40
Total	93.60	6.40	100.00

This cross-tabulation shows the percentage of updates under the Version Number (columns) and Release Notes (rows) classification systems.

Specifically, I use natural language processing techniques to process and analyze these documents, and I use a hand-coded set of release notes to train a support vector machine (SVM), a supervised machine learning technique, which can then be used to classify the content of product updates based on the content of their release notes. I classify app updates as either "Feature" updates, which add new features or functionality to the product, or as "Bug Fix" updates, which correct software bugs or make other incremental changes. Figure 3 outlines the definition for each of these update types, and Table 2a reports details on the fit of the trained SVM model. While Major and Feature (or Minor and Bug Fix) updates can be viewed as somewhat analogous, I maintain the distinct vocabularies to help differentiate the two approaches throughout this paper.

Under both approaches, the less substantial type of update (Minor/Bug Fix) are more common than the more substantial updates (Major/Feature). Notably, the overlap is not perfect—roughly two-fifths of the Minor updates, under the VN classification approach, appear to have added additional features or functionality to the product according to the RN classification approach. This cross-tabulation is presented in Table 2b.

5 Empirical Analysis

I now turn to estimating how the relationship between average rating and product updating changed following the 2017 policy change. I first estimate the overall effect, and how that varied across product categories. I then consider how these effects varied with the extent to which developers relied on the rating system. Finally, I look at how the content of product updates changed following the policy change in order to provide some evidence on whether the original rating policy discouraged developer effort on the platform.

Table 3: The Effect of Apple's Review Reset Policy on Updating Behavior

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0165***			-0.0001	-0.0092***
	(0.0007)			(0.0013)	(0.0024)
Avg Review Score (Large Penalty)		-0.0182***	-0.0139***		
		(0.0008)	(0.0008)		
Avg Review Score (Small Penalty)		-0.0113***	-0.0260***		
		(0.0015)	(0.0017)		
Avg Review Score X Post iOS 11	0.0107***			0.0043*	0.0040
	(0.0007)			(0.0017)	(0.0030)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0115***	0.0082***		
		(0.0008)	(0.0007)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0041***	0.0077***		
		(0.0012)	(0.0022)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,483,817	1,483,817	1,483,817	192,763	192,763

Robust standard errors are reported in parentheses. Other control variables included in all regressions, but whose coefficients are not reported, are the age for the current version, log file size, and category-week and app fixed effects. See Appendix A for the full set of estimated coefficients.

5.1 Overall Response

In order to understand how Apple's initial rating resetting policy affected developers' product updating behavior, I estimate the following linear probability model using OLS,

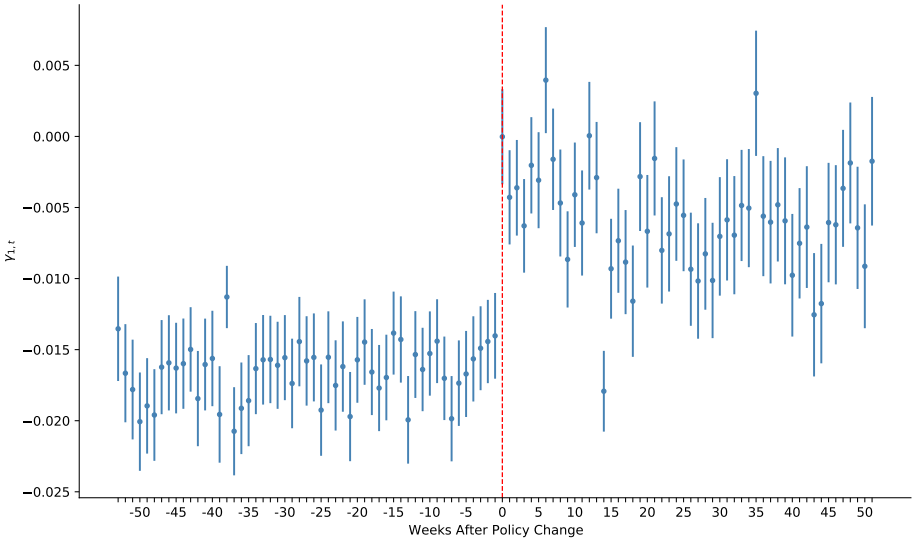
$$\begin{aligned}\mathbb{1}(update)_{j,t} = \beta_0 \bar{r}_{j,t} + \beta_1 \bar{r}_{j,t} \times \mathbb{1}(\text{Post-iOS 11})_{j,t} \\ + \beta_2 \mathbb{1}(\text{No Rating})_{j,t} + \beta_3 \mathbb{1}(\text{No Rating})_{j,t} \times \mathbb{1}(\text{Post-iOS 11})_{j,t} \\ + \beta_4 p_{j,t} + \beta_5 \text{version-age}_{j,t} + \beta_6 \text{size}_{j,t} + \mu_{c,t} + \mu_j + \epsilon_{j,t}\end{aligned} \tag{1}$$

where, $\mathbb{1}(update)_{j,t}$ is an indicator for whether app j was updated in week t, and $\bar{r}_{j,t}$ is the average rating of the app's current version at the start of period t (i.e., prior to an update, if one occurs). $\mathbb{1}(\text{Post-iOS 11})_{j,t}$ indicates whether period t is after the policy change, which occurred with the September 19, 2017 release of Apple's iOS 11 operating system. $\mathbb{1}(\text{No Rating})_{j,t}$ indicates whether an app is currently displaying the "No Ratings" label, which is applied to an app version with fewer than five ratings. In order to control for other relevant factors that might affect a developer's updating decision, how long the current version of the app has been on the App Store ($\text{version-age}_{j,t}$), and the size of the app measured in megabytes ($\text{size}_{j,t}$). Additionally, I include category-week and app fixed effects, $\mu_{c,t}$ and μ_j.

The coefficient of interest is β_1, which represents the extent to which the estimated relationship between an app's current (i.e., prior to the start of week t) rating $\bar{r}_{j,t}$ and the developers' updating decisions changes following the removal of the reset policy. Connecting this to the stylistic model depicted in Figure 2, the slope of σ^P is β_0 and the slope of σ^* is $\beta_0 + \beta_1$.

Column (1) of Table 3 shows the results from estimating Equation (1) on the full sample of

Figure 4: Monthly Estimates of the Relationship Between Average Rating and Product Updating



Results of estimating Equation (2), which allows the relationship between average rating and updating to vary by month. Bars indicate 95% confidence intervals.

products. The first row shows a negative relationship between an app's rating and its likelihood of updating. In the fourth row, I find a positive effect of the removal of Apple's resetting policy on this relationship (β_1). The estimated slope coefficient of 0.0107 implies the removal of the reset policy led to a 0.043 percentage point increase in the frequency of updating for an average rated app, or a 33% increase relative to the pre-iOS 11 period average. This translates to an increase from an average of roughly one update every two months to one update every six weeks.

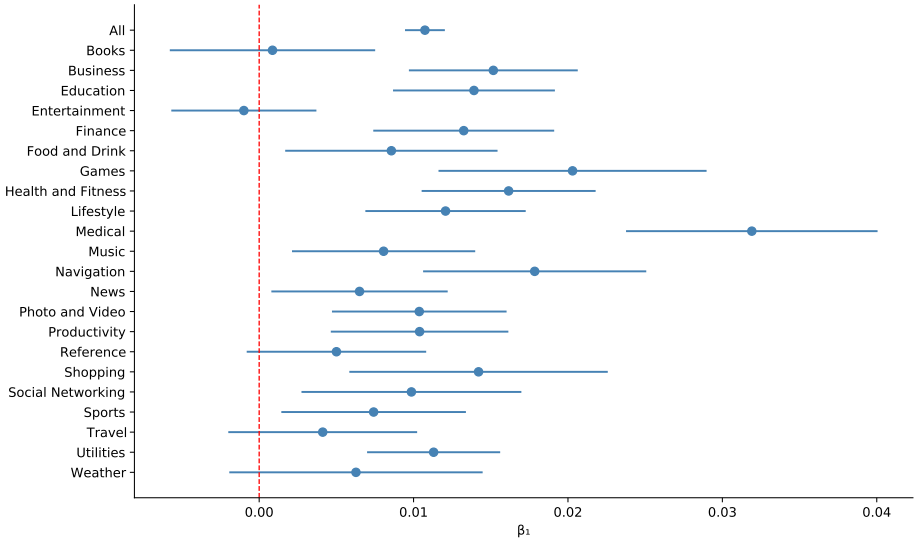
I next estimate the following variation of Equation (1) that allows the relationship between an app's current average rating ($\bar{r}_{j,t}$) and updating to vary by month, rather than just before and after the release of iOS 11.

$$\begin{aligned}\mathbb{1}(update)_{j,t} = &\sum_{t=-52}^{52} \gamma_{1,t}\bar{r}_{j,t} + \gamma_2 \mathbb{1}(\text{No Rating})_{j,t} + \gamma_3 \mathbb{1}(\text{No Rating})_{j,t} \times \mathbb{1}(\text{Post-iOS 11})_{j,t} \\ &+ \gamma_4 p_{j,t} + \gamma_5 \text{version-age}_{j,t} + \gamma_6 \text{size}_{j,t} + \delta_{c,t} + \delta_j + \nu_{j,t}\end{aligned} \tag{2}$$

This specification provides insight as to 1) whether there's a delay before the policy change affects developer behavior, and 2) whether the effect dissipates over time as developers acclimate to the new policy regime. Here the set of coefficients $\{\gamma_{1,t}\}_{t=-52}^{52}$ are the coefficients of interest. I present these estimates in Figure 4, which shows that the change in updating behavior once developers have the opportunity to preserve past customer ratings is immediate and persistent.

Having established that developers responded to the policy change by reducing the frequency of updates, and that this response was both immediate and persistent, I next consider the possibility that this respone varied by product category. In Figure 5, I present the results of estimating

Figure 5: The Effect of Apple's Review Reset Policy on Updating Behavior By Category



Results of estimating Equation (1) separately for each category. Bars indicate 95% confidence intervals.

Equation (1) separately for each product category. I find that while the effect of the policy change on the relationship between an app's average rating and the decision to update varies by category, with the largest effects in the Medical, Games, and Navigation categories, the effect is generally similar across categories. Notably, some categories, specifically Books, Entertainment, News, Reference, and Travel show no statistically significant evidence of an effect at the 5% level. Ultimately, Figure 5 suggests that rating systems play a heterogeneous role within a platform, which could be the case due to varying levels of concern about information asymmetries, the informativeness of reviews, and/or the reliability of reviews.

Across Table 3 and Figures 4 and 5, the estimates of the effect of the policy change on the average ratings-updating relationship are broadly consistent with the idea that Apple's original policy of resetting product ratings when an app was updated decreased the frequency of updating among higher-quality apps. In light of these results, it also seems likely that Apple's reset policy led developers of lower-rated apps to update more frequently than they otherwise would. However, because Apple's policy change in 2017 provided developers with the option of reseting their rating with each update—which means that low-rated apps were not necessarily "treated" by the policy change—I am unable to directly test this.

5.2 Response Heterogeneity

Having shown that, on average, the rating reset policy decreased the frequency of product updates, I now consider whether and how this response varied across developers. The extent to which an app's updating propensity was affected by the policy likely depends on the degree to which the app faced a meaningful reputational penalty from losing its past rating. In particular, the severity of

this penalty could vary by the size and length of the penalty.

The size of the penalty reflects how dependent a given app is on its rating to attract new customers. For example, it is unlikely that sales of Microsoft Word, a dominant word processing application, are reliant on its current App Store rating, while Ulysses, a more niche word processing app, may depend heavily on having a good rating to attract new customers. As a result, the negative demand shock associated with an update-induced rating reset should be much larger for Ulysses than for Microsoft Word.

The length of the penalty reflects the fact that once an app's rating has been reset, the app is dependent on users rating the app in order to rebuild its reputation. Recall from Section 2 that following a reset, Apple displays a "No Ratings" indicator until at least five consumers have rated the product. For some apps, this may take only a few minutes or hours until a new rating is displayed, while for others it could take days or weeks. For apps in the former group, it is unlikely that the reset policy would significantly affect their innovative behavior, because the cost would be short-lived.

In order to investigate whether apps with particularly high demand, or those with the highest arrival rate of reviews are less distorted by the initial rating reset policy, I allow the estimate of the effect of the policy change to differ for the top quartile of the sample according to the app's average download ranking (Penalty Size) and according to the app's average number of new reviews per week (Penalty Length). I present the results of re-estimating Equation (1) in this way in columns (3) and (4) of Table 3.

Column (3) of Table 3 shows a larger response to the policy change among apps facing a larger penalty size, proxied by the apps' average rating. Indeed, the response by large penalty apps is nearly 60% more than those facing a smaller penalty. In column (4) of Table 3, I show the results of allowing the effect of the policy change to differ depending on the length of the reputational penalty, proxied by the arrival rate of new reviews. Developers facing a longer penalty—i.e., those with a lower arrival rate of new reviews—are more responsive to the policy change. Taken together, the results in this stage of the analysis further support the view that Apple's policy created a real reputational penalty that affected developers' innovative behavior on the platform, and that developers' response to the policy varied with both the size and the length of the penalty they faced.

5.3 Content of Product Updates

I have shown that the original reset policy led to a decrease in the frequency of product updates. This result can be explained both by decreased developer effort—more product updates would have been produced but for the policy—or it could be the case that while developers updated their products less frequently when the reset policy was in place, the total amount of effort remained the same, and was just distributed across less frequent, more aggregated releases. While this latter option would still result in a policy-induced welfare loss—if an update is delayed for a week, the value that would have accrued to consumers over the course of that week often cannot be

recovered—it would be a smaller loss than if the product innovation were lost entirely.

If the rating reset policy resulted in sub-optimally bundled updates, then we should expect to see an increase in the relative likelihood of Minor/Bug Fix updates following the policy change in 2017. This is because, in the presence of the reset policy, Minor/Bug Fix updates impose the same reputational penalty on developers as Major/Feature updates, despite typically being of lesser importance. Thus, the removal of the penalty would allow developers to return to their optimal release schedule for small and large updates.

If, instead, the rating reset policy resulted in decreased effort, the effect on the relative frequency of Minor/Bug Fix updates in the post period is theoretically ambiguous and would depend on the nature of the lost updates. If developers responded to the reputational penalty by fixing fewer bugs, then we would expect to see the relative likelihood of Minor/Bug Fix updates increase. Alternatively, if developers responded to the reputational penalty by developing fewer new product features, then we would expect the relative likelihood of Minor/Bug Fix updates to decrease following the policy change.

I investigate whether updates were lost or simply delayed by estimating a variation of Equation (1) on only the set of observations where developers chose to update, using an indicator for a Minor or Bug Fix (as opposed to Major or Feature, respectively) updates as the left-hand side variable. As before, the coefficient of interest is β_1, which indicates how the observed relationship between an app's average rating and the likelihood of creating a Minor/Bug Fix update conditional on updating changes after the rating reset policy is changed. If product updates are simply delayed, then estimates of β_1 should be positive. Otherwise, the data is consistent with the view that innovation was lost as a result of the rating reset policy. Column (5) Table 3 displays the results of estimating this model using the Version Number (VN) classification approach (Major/Minor updates), and column (6) shows the results of estimating the model using the Release Notes (RN) classification approach (Feature/Bug Fix updates).

The overall results using the VN approach, presented in column (5) of Table 3, show that Apple's initial policy resulted in less frequent Bug Fix updates than developers would produce absent the policy. This overall finding of an increase in the relative frequency of Bug Fix updates is consistent with *both* the lost innovation theory and the theory that developers engaged in a practice of delaying updates in order to bundle small and large revisions in less frequent releases. Estimating this model using the more precise Release Notes (RN) classification, which is constructed using developers' own descriptions of how they have updated their products, I fail to find statistically significant evidence of a positive relationship between the removal of the reset policy and the content of updates when looking across all categories (column (6) of Table 3). This provides inconclusive, but suggestive evidence that the initial policy may have resulted in diminished developer effort, and not just delayed updating, as discussed above.

6 Robustness

In this section, I consider the possibility that the findings in Section 5 were driven by developers anticipating the policy change, and I reproduce Table 3 under alternative sample definitions.

6.1 Excluding the Summer of 2017

As noted in Section 2, while Apple announced its plans to change its rating reset policy in June 2017, the actual policy change did not occur until September of that year. It could be the case that developers strategically withheld updates until after the policy change as a result of this early announcement. To address the concern that this potential behavior may have affected the findings of this paper, I reproduce Table 3 using a sample that excludes all observations from the time of the announcement until the end of September 2017. I present summary statistics for this alternative sample in Table 4, and the reproduction of the main results using this sample in Table 5. We can see in column (1) of Table 5 that the primary effect is nearly identical under this alternative sample. The remaining specifications in the table also closely match the primary results in Table 3, with the exception that penalty length no longer appears to be associated with different responses. Overall, this indicates that the findings in this paper were not driven by the early summer announcement of this policy.

6.2 Alternative Sample Definitions

As discussed in Section 4, the primary sample in this paper is constructed using a set of criteria that help identify apps that are actively competing on the platform, as opposed to those that have long since been abandoned by the developer, or that purely exist as a hobby project. In this section, I reproduce the primary results under two alternative sample criteria.

First, in Table 6, I define the sample using a "drop out" variation of the primary sample which allows firms to enter and exit the sample whenever their ranking falls below the specified ranking threshold (200^{th} or better on one or more of the category-level sales and revenue ranking lists). Comparing these results to the primary results presented in Table 6, we can see that removing apps from the sample when their ranking falls below this threshold does not have a meaningful effect on the results.

Second, I consider a more restrictive ranking threshold, limiting the sample to apps that rank 100^{th} or better for 25% of their time in the sample. Considering this sample of higher-performing apps in Table 7, I find again that the reset policy led developers to strategically reduce the frequency of updates. Interestingly, the differential responses to the penalty size and length are more pronounced than in the primary sample. Finally, I fail to find evidence of an increase in the relative frequency of small-to-large updates under both classification systems, which provides supportive evidence for the view that the reset policy led developers to invest less on the platform.

Table 4: Summary Statistics, Excluding Summer 2017

	Update		No Ratings		Average Rating ($\bar{r} > 0$)		
	Mean	SD	Mean	SD	Mean	SD	N
Category							
Books	0.103	0.304	0.218	0.413	4.170	0.815	44,423
Business	0.147	0.355	0.210	0.407	3.827	1.028	56,049
Education	0.106	0.307	0.137	0.344	4.040	0.786	70,624
Entertainment	0.106	0.308	0.106	0.308	3.858	0.904	75,716
Finance	0.154	0.361	0.180	0.384	3.920	0.990	51,247
Food and Drink	0.138	0.345	0.213	0.409	3.819	1.041	37,108
Games	0.177	0.382	0.031	0.173	4.317	0.483	99,641
Health and Fitness	0.141	0.348	0.124	0.330	4.255	0.781	67,122
Lifestyle	0.132	0.338	0.186	0.389	3.906	0.977	68,082
Medical	0.110	0.313	0.241	0.428	4.149	0.858	41,675
Music	0.115	0.318	0.145	0.352	4.103	0.783	58,468
Navigation	0.124	0.329	0.269	0.443	3.859	0.943	37,854
News	0.135	0.342	0.224	0.417	3.793	1.013	52,469
Photo and Video	0.130	0.337	0.096	0.295	4.177	0.763	70,304
Productivity	0.134	0.341	0.124	0.329	4.078	0.781	68,518
Reference	0.074	0.262	0.153	0.360	4.220	0.826	45,126
Shopping	0.215	0.411	0.141	0.348	4.001	0.949	38,164
Social Networking	0.200	0.400	0.168	0.374	3.814	0.929	51,294
Sports	0.132	0.339	0.239	0.427	3.755	1.037	49,648
Travel	0.158	0.365	0.211	0.408	3.953	0.995	49,797
Utilities	0.094	0.291	0.111	0.314	3.883	0.916	69,313
Weather	0.096	0.295	0.260	0.439	4.061	0.832	39,062
Time Period							
Pre-iOS 11	0.131	0.337	0.310	0.462	3.985	0.949	479,976
Post-iOS 11	0.134	0.341	0.063	0.244	4.036	0.851	761,728
Total	0.133	0.340	0.159	0.365	4.019	0.884	1,241,704

Table 5: The Effect of Apple's Review Reset Policy on Updating Behavior, Excluding Summer 2017

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0162***			0.0010	-0.0098***
	(0.0009)			(0.0016)	(0.0029)
Avg Review Score (Large Penalty)		-0.0177***	-0.0136***		
		(0.0010)	(0.0009)		
Avg Review Score (Small Penalty)		-0.0112***	-0.0251***		
		(0.0017)	(0.0020)		
Avg Review Score X Post iOS 11	0.0113***			0.0055**	0.0028
	(0.0008)			(0.0020)	(0.0035)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0120***	0.0085***		
		(0.0009)	(0.0008)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0045**	0.0084***		
		(0.0014)	(0.0025)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,241,704	1,241,704	1,241,704	163,225	163,225

This table presents the results of reproducing Table 3 using a sample that excludes all observations from the time of the policy announcement until the end of September 2017. Robust standard errors are reported in parentheses. Other control variables included in all regressions, but whose coefficients are not reported, are the age for the current version, log file size, and category-week and app fixed effects. See Appendix A for the full set of estimated coefficients.

Table 6: The Effect of Apple's Review Reset Policy on Updating Behavior, "Drop out" Sample

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0221***			-0.0012	-0.0080**
	(0.0010)			(0.0015)	(0.0029)
Avg Review Score (Large Penalty)		-0.0240***	-0.0198***		
		(0.0011)	(0.0011)		
Avg Review Score (Small Penalty)		-0.0158***	-0.0287***		
		(0.0019)	(0.0020)		
Avg Review Score X Post iOS 11	0.0124***			0.0041*	0.0049
	(0.0009)			(0.0020)	(0.0036)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0121***	0.0095***		
		(0.0011)	(0.0009)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0072***	0.0087***		
		(0.0016)	(0.0025)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	956,684	956,684	956,684	138,068	138,068

This table presents the results of reproducing Table 3 with a sample that drops apps when they fall below a ranking 200^{th} on a sales or revenue ranking list. Robust standard errors are reported in parentheses. Other control variables included in all regressions, but whose coefficients are not reported, are the age for the current version, log file size, and category-week and app fixed effects.

Table 7: The Effect of Apple's Review Reset Policy on Updating Behavior, Alternative Ranking Threshold

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0203***			-0.0006	-0.0088**
	(0.0010)			(0.0015)	(0.0029)
Avg Review Score (Large Penalty)		-0.0220***	-0.0188***		
		(0.0011)	(0.0010)		
Avg Review Score (Small Penalty)		-0.0147***	-0.0268***		
		(0.0019)	(0.0022)		
Avg Review Score X Post iOS 11	0.0125***			0.0037	0.0062
	(0.0009)			(0.0021)	(0.0037)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0124***	0.0110***		
		(0.0010)	(0.0009)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0071***	0.0041		
		(0.0016)	(0.0030)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	913,573	913,573	913,573	130,290	130,290

This table presents the results of reproducing Table 3 with a sample that places a more restrictive set of criteria on which products are included. Robust standard errors are reported in parentheses. Other control variables included in all regressions, but whose coefficients are not reported, are the age for the current version, log file size, and category-week and app fixed effects.

7 Conclusion

In this paper, I study how the manner in which information from consumer reviews is aggregated affects firms' strategic behavior. I find that Apple's longstanding ratings reset policy led developers to reduce the frequency of product updates, and that developers who are likely to be more dependent on the product rating system were more responsive to the discouragement effects of the policy. Finally, I provide preliminary evidence that this policy may have led to decreased developer effort and lost product updates, although in some cases developers may have simply delayed updating in response to the policy by choosing to bundle together small and large product updates.

As the prominence of digital platforms continues to increase, it is important that policy-makers and platform owners understand how the design of these centralized markets can affect intra-platform competition and innovation, and, ultimately, welfare (Cremer, de Montjoye, and Schweitzer, 2019; Scott Morton et al., 2019; Furman et al., 2019). In particular, socially inefficient platform policies can persist given the relative dearth of platform-level competition. My finding of (likely) lost product innovation provides empirical evidence of how the design of a platform can have a first-order effect on competitive behavior, and, ultimately, lead to a reduction of consumer welfare. Notably, even delayed investment by developers—à la the bundled updates mechanism—would result in welfare losses, though this reduction would be smaller than if the product innovation had been lost entirely.

That said, there are a number of ways in which these welfare losses could be mitigated. The preceding analysis does not account for endogenous changes in product monetization, or effects on

entry and exit, all of which might offset the welfare effects of lost product updates. Indeed, Vellodi (2021) shows that suppressing the ratings of top-rated firms could lead to increased entry and, consequently, improve consumer welfare. While mechanically distinct, Apple's initial reputation system is in many ways akin to the upper certification policy Vellodi considers. Ultimately, there remains substantial room for further research on how the manner in which platforms aggregate and provide information, and platform design decisions more generally, affect consumer and overall welfare.

References

Acemoglu, Daron, Ali Makhdoumi, Azarakhsh Malekian, and Asu Ozdaglar. 2022. "Learning from Reviews: The Selection Effect and the Speed of Learning." *Econometrica* 90 (6):2857–2899. 2

Airnow. 2016. "AppMonsta App Details." 7

Ananthakrishnan, Uttara M., Davide Proserpio, and Siddhartha Sharma. 2023. "I Hear You: Does Quality Improve with Customer Voice?" Working paper, SSRN. URL https://ssrn.com/abstract=3467236. 3

Apple. 2018. "App Store kicks off 2018 with record-breaking holiday season." *Apple Newsroom* . 4

Cabral, Luís and Ali Hortaçsu. 2004. "The Dynamics of Seller Reputation: Theory and Evidence from eBay." Working Paper 10363, National Bureau of Economic Research. URL http://www.nber.org/papers/w10363. 3

———. 2010. "The Dynamics of Seller Reputation: Evidence from eBay." *The Journal of Industrial Economics* 58 (1):54–78. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6451.2010.00405.x. 3

Chevalier, Judith A and Dina Mayzlin. 2006. "The Effect of Word of Mouth on Sales: Online Book Reviews." *Journal of Marketing Research* 43:345–354. 6

Claussen, Jrg, Tobias Kretschmer, and Philip Mayrhofer. 2013. "The Effects of Rewarding User Engagement: The Case of Facebook Apps." *Information Systems Research* 24 (1):186–200. 3

Comino, Stefano, Fabio M. Manenti, and Franco Mariuzzo. 2018. "Updates Management in Mobile Applications : iTunes vs Google Play." *Journal of Economics and Management Strategy* 28 (3):392–419. 3, 5

Cremer, Jacques, Yves-Alexandre de Montjoye, and Heike Schweitzer. 2019. "Competition Policy for the Digital Era." . 19

Dai, Weijia, Ginger Jin, Jungmin Lee, and Michael Luca. 2018. "Aggregation of consumer ratings: an application to Yelp.com." *Quantitative Marketing and Economics* 16 (3):289–339. 2

Davis, Jason P., Yulia Chhabra, and Pai-ling Yin. 2016. "Entrepreneurial Diversification and the Scope of New Firms: Multiproduct Innovation in iPhone Apps." *Working Paper* . 3

Ershov, Daniel. 2022. "Variety-Based Congestion in Online Markets: Evidence from Mobile Apps." *Working Paper* . 3

Filippas, Apostolos, John Joseph Horton, and Joseph Golden. 2022. "Reputation inflation." *Marketing Science* 41 (4):305–317. 2

Furman, Jason, Diane Coyle, Amelia Fletcher, Derek McAules, and Philip Marsden. 2019. "Unlocking digital competition: Report of the digital competition expert panel." . 19

Ghose, Anindya and Sang Pil Han Han. 2014. "Estimating demand for mobile applications in the new economy." *Management Science* 60 (6):1470–1488. 5

Ghose, Anindya, Panagiotis G. Ipeirotis, and Beibei Li. 2014. "Examining the Impact of Ranking on Consumer Behavior and Search Engine Revenue." *Management Science* 60 (7):1632–1654. 2

Gruber, John. 2017. "Live from WWDC 2017." *The Talk Show* :Podcast audio. 5

Hackett, Stephen. 2018. "10 Years of App Store: A Timeline of Changes." 4

Huang, Yangguang, Chenyang Li, and Si Zuo. 2022. "Price signaling and reputation building: evidence from a service platform." *HKUST Business School Research Paper* 2022-062. URL https://ssrn.com/abstract=4080448. 3

Hui, Xiang, Maryam Saeedi, Giancarlo Spagnolo, and Steven Tadelis. 2018. "Certification, Reputation and Entry: An Empirical Analysis." Working Paper 24916, National Bureau of Economic Research. URL http://www.nber.org/papers/w24916. 3

Hunter, Megan. 2020. "Chasing Stars: Firms' Strategic Responses to Online Consumer Ratings." *Available at SSRN 3554390* . 3

Jabr, Wael, Dominik Gutt, Jurgen Neumann, and Dennis Kundisch. 2022. "My Reviews are taken away, what about my Reputation? The Asymmetric Impact of Resetting the Review History on Mobile App Platforms." *Working Paper* . 6

Jia, Jian, Ginger Zhe Jin, and Liad Wagman. 2021. "Platform as a Rule-Maker: Evidence from Airbnb's Cancellation Policies." Working Paper 28878, National Bureau of Economic Research. URL http://www.nber.org/papers/w28878. 3

Jin, Ginger Zhe and Phillip Leslie. 2003. "The effect of information on product quality: Evidence from restaurant hygiene grade cards." *The Quarterly Journal of Economics* 118 (2):409–451. 2, 5

Klein, Tobias J, Christian Lambertz, and Konrad O Stahl. 2016. "Market transparency, adverse selection, and moral hazard." *Journal of Political Economy* 124 (6):1677–1713. 3

Leyden, Benjamin T. 2022. "There's an App for That: Understanding Product Updating Under Digitization." *Working Paper* . 4, 5, 8

Lizzeri, Alessandro. 1999. "Information Revelation and Certification Intermediaries." *The RAND Journal of Economics* 30 (2):214–231. 2

Luca, Michael. 2016. "Reviews, reputation, and revenue: The case of Yelp. com." *Harvard Business School Working Paper* 12-016. 2, 6

Luca, Michael and G. Zervas. 2016. "Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud." *Management Science* 62 (12):3393–3672. 2, 6

Mayzlin, Dina, Yaniv Dover, and Judith Chevalier. 2014. "Promotional Reviews: An Empirical Investigation of Online Review Manipulation." *American Economic Review* 104 (8):2421–2455. 2

Scott Morton, Fiona, Pascal Bouvier, Ariel Ezrachi, Bruno Jullien, Roberta Katz, Gene Kimmelman, A Douglas Melamed, and Jamie Morgenstern. 2019. "Committee for the Study of Digital Platforms: Market Structure and Antitrust Subcommittee Report." . 19

Shipley, Wil. 2014. "Begging for app ratings." *Loop Insight* . 5

Staff, Fortune. 2017. "Apple iPhone use worldwide." *Fortune* . 4

Vana, Prasad and Anja Lambrecht. 2021. "The effect of individual online reviews on purchase likelihood." *Marketing Science* 40 (4):708–730. 2

Vellodi, Nikhil. 2021. "Ratings Design and Barriers to Entry." *Working Paper* . 20

Appendix A Full Results

In this appendix, I present the full set of coefficients estimates for the specifications reported in the paper, as well as additional specifications not reported in the paper. Specifically,

- In Table A.1, I report the full set of coefficient estimates for the primary table of results in the paper (Table 3).
- In Table A.2, I report the full set of coefficient estimates for Table 5, which uses the sample that excludes the summer of 2017 period between the announcement of the policy change and the end of September, 2017 (see Section 6.1).
- In Tables A.3 and A.4 I consider whether controlling for the number of ratings an app has received affects the estimated effect of the ratings policy change. Specifically, I include log(Ratings Count + 1) as an additional control variable. Under this alternative specification, the magnitude of the estimated effect of the policy change declines, but the overall findings of this paper remain the same.

Table A.1: The Effect of Apple's Review Reset Policy on Updating Behavior

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0165***			-0.0001	-0.0092***
	(0.0007)			(0.0013)	(0.0024)
Avg Review Score (Large Penalty)		-0.0182***	-0.0139***		
		(0.0008)	(0.0008)		
Avg Review Score (Small Penalty)		-0.0113***	-0.0260***		
		(0.0015)	(0.0017)		
Avg Review Score X Post iOS 11	0.0107***			0.0043*	0.0040
	(0.0007)			(0.0017)	(0.0030)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0115***	0.0082***		
		(0.0008)	(0.0007)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0041***	0.0077***		
		(0.0012)	(0.0022)		
No Ratings	-0.0635***			0.0041	-0.0046
	(0.0030)			(0.0055)	(0.0103)
No Ratings X Post iOS 11	0.0403***			0.0074	-0.0103
	(0.0031)			(0.0081)	(0.0142)
Version Age	0.0000***			-0.0008***	-0.0005***
	(0.0000)			(0.0000)	(0.0000)
Log Download Size	0.0907***			-0.0234***	-0.0605***
	(0.0014)			(0.0029)	(0.0050)
No Ratings (Small Penalty)		-0.0420***	-0.1251***		
		(0.0057)	(0.0077)		
No Ratings (Large Penalty)		-0.0698***	-0.0521***		
		(0.0035)	(0.0031)		
No Ratings X Post iOS 11 (Small Penalty)		0.0142*	0.0477***		
		(0.0057)	(0.0143)		
No Ratings X Post iOS 11 (Large Penalty)		0.0431***	0.0325***		
		(0.0037)	(0.0031)		
Version Age (Small Penalty)		0.0000***	0.0000***		
		(0.0000)	(0.0000)		
Version Age (Large Penalty)		0.0000***	0.0000***		
		(0.0000)	(0.0000)		
Log Download Size (Small Penalty)		0.0815***	0.1197***		
		(0.0033)	(0.0032)		
Log Download Size (Large Penalty)		0.0922***	0.0780***		
		(0.0016)	(0.0016)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,483,817	1,483,817	1,483,817	192,763	192,763

Robust standard errors are reported in parentheses.

Table A.2: The Effect of Apple's Review Reset Policy on Updating Behavior, Excluding Summer 2017

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0162***			0.0010	-0.0098***
	(0.0009)			(0.0016)	(0.0029)
Avg Review Score (Large Penalty)		-0.0177***	-0.0136***		
		(0.0010)	(0.0009)		
Avg Review Score (Small Penalty)		-0.0112***	-0.0251***		
		(0.0017)	(0.0020)		
Avg Review Score X Post iOS 11	0.0113***			0.0055**	0.0028
	(0.0008)			(0.0020)	(0.0035)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0120***	0.0085***		
		(0.0009)	(0.0008)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0045**	0.0084***		
		(0.0014)	(0.0025)		
No Ratings	-0.0630***			0.0097	0.0021
	(0.0035)			(0.0066)	(0.0123)
No Ratings X Post iOS 11	0.0464***			0.0115	-0.0227
	(0.0036)			(0.0092)	(0.0162)
Version Age	0.0000***			-0.0008***	-0.0005***
	(0.0000)			(0.0000)	(0.0000)
Log Download Size	0.0885***			-0.0226***	-0.0570***
	(0.0015)			(0.0031)	(0.0054)
No Ratings (Small Penalty)		-0.0442***	-0.1222***		
		(0.0068)	(0.0090)		
No Ratings (Large Penalty)		-0.0680***	-0.0525***		
		(0.0041)	(0.0037)		
No Ratings X Post iOS 11 (Small Penalty)		0.0188**	0.0550***		
		(0.0067)	(0.0159)		
No Ratings X Post iOS 11 (Large Penalty)		0.0490***	0.0377***		
		(0.0043)	(0.0037)		
Version Age (Small Penalty)		0.0000***	0.0000**		
		(0.0000)	(0.0000)		
Version Age (Large Penalty)		0.0000***	0.0000***		
		(0.0000)	(0.0000)		
Log Download Size (Small Penalty)		0.0815***	0.1168***		
		(0.0035)	(0.0034)		
Log Download Size (Large Penalty)		0.0895***	0.0757***		
		(0.0017)	(0.0017)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,241,704	1,241,704	1,241,704	163,225	163,225

Standard errors are in parentheses, and are clustered at the app level.

Table A.3: The Effect of Apple's Review Reset Policy on Updating Behavior (with Rating Count)

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0181***			0.0005	-0.0069**
	(0.0007)			(0.0013)	(0.0025)
Avg Review Score (Large Penalty)		-0.0202***	-0.0137***		
		(0.0009)	(0.0008)		
Avg Review Score (Small Penalty)		-0.0106***	-0.0323***		
		(0.0015)	(0.0018)		
Avg Review Score X Post iOS 11	0.0085***			0.0056**	0.0088**
	(0.0007)			(0.0018)	(0.0031)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0087***	0.0084***		
		(0.0008)	(0.0007)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0049***	0.0007		
		(0.0012)	(0.0022)		
No Ratings	-0.0593***			0.0031	-0.0083
	(0.0030)			(0.0055)	(0.0103)
No Ratings X Post iOS 11	0.0351***			0.0109	0.0030
	(0.0031)			(0.0082)	(0.0144)
Log Rating Count	0.0042***			-0.0016**	-0.0060***
	(0.0003)			(0.0005)	(0.0010)
Version Age	0.0000***			-0.0008***	-0.0004***
	(0.0000)			(0.0000)	(0.0000)
Log Download Size	0.0893***			-0.0231***	-0.0591***
	(0.0014)			(0.0029)	(0.0050)
No Ratings (Small Penalty)		-0.0458***	-0.1225***		
		(0.0058)	(0.0077)		
No Ratings (Large Penalty)		-0.0658***	-0.0534***		
		(0.0035)	(0.0032)		
No Ratings X Post iOS 11 (Small Penalty)		0.0156**	0.0281		
		(0.0057)	(0.0144)		
No Ratings X Post iOS 11 (Large Penalty)		0.0362***	0.0329***		
		(0.0037)	(0.0031)		
Log Rating Count (Small Penalty)		-0.0024**	0.0082***		
		(0.0008)	(0.0006)		
Log Rating Count (Large Penalty)		0.0051***	-0.0007		
		(0.0004)	(0.0004)		
Version Age (Small Penalty)		0.0000***	0.0000		
		(0.0000)	(0.0000)		
Version Age (Large Penalty)		0.0000***	0.0000***		
		(0.0000)	(0.0000)		
Log Download Size (Small Penalty)		0.0822***	0.1186***		
		(0.0033)	(0.0032)		
Log Download Size (Large Penalty)		0.0904***	0.0782***		
		(0.0016)	(0.0016)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,483,817	1,483,817	1,483,817	192,763	192,763

Robust standard errors are reported in parentheses.

Table A.4: The Effect of Apple's Review Reset Policy on Updating Behavior, Excluding Summer 2017 (with Rating Count)

	Overall	Heterogeneity		Update Content	
	(1)	(2)	(3)	(4)	(5)
	—	Penalty Size	Penalty Length	VN	RN
Avg Review Score	-0.0176***			0.0016	-0.0078**
	(0.0009)			(0.0016)	(0.0030)
Avg Review Score (Large Penalty)		-0.0195***	-0.0134***		
		(0.0010)	(0.0009)		
Avg Review Score (Small Penalty)		-0.0106***	-0.0300***		
		(0.0017)	(0.0021)		
Avg Review Score X Post iOS 11	0.0092***			0.0070***	0.0078*
	(0.0008)			(0.0021)	(0.0036)
Avg Review Score X Post iOS 11 (Large Penalty)		0.0093***	0.0088***		
		(0.0009)	(0.0008)		
Avg Review Score X Post iOS 11 (Small Penalty)		0.0052***	0.0017		
		(0.0014)	(0.0025)		
No Ratings	-0.0593***			0.0084	-0.0021
	(0.0035)			(0.0066)	(0.0123)
No Ratings X Post iOS 11	0.0413***			0.0158	-0.0085
	(0.0036)			(0.0094)	(0.0164)
Log Rating Count	0.0039***			-0.0018**	-0.0059***
	(0.0004)			(0.0006)	(0.0012)
Version Age	0.0000***			-0.0008***	-0.0004***
	(0.0000)			(0.0000)	(0.0000)
Log Download Size	0.0871***			-0.0221***	-0.0555***
	(0.0015)			(0.0031)	(0.0054)
No Ratings (Small Penalty)		-0.0477***	-0.1189***		
		(0.0069)	(0.0090)		
No Ratings (Large Penalty)		-0.0644***	-0.0544***		
		(0.0041)	(0.0038)		
No Ratings X Post iOS 11 (Small Penalty)		0.0202**	0.0364*		
		(0.0067)	(0.0160)		
No Ratings X Post iOS 11 (Large Penalty)		0.0424***	0.0383***		
		(0.0043)	(0.0037)		
Log Rating Count (Small Penalty)		-0.0022**	0.0070***		
		(0.0008)	(0.0007)		
Log Rating Count (Large Penalty)		0.0045***	-0.0011*		
		(0.0004)	(0.0005)		
Version Age (Small Penalty)		0.0000***	0.0000		
		(0.0000)	(0.0000)		
Version Age (Large Penalty)		0.0000***	0.0000***		
		(0.0000)	(0.0000)		
Log Download Size (Small Penalty)		0.0822***	0.1158***		
		(0.0035)	(0.0034)		
Log Download Size (Large Penalty)		0.0879***	0.0760***		
		(0.0017)	(0.0017)		
App FEs	✓	✓	✓	✓	✓
Category-Week FEs	✓	✓	✓	✓	✓
N	1,241,704	1,241,704	1,241,704	163,225	163,225

Standard errors are in parentheses, and are clustered at the app level.