International Journal of Research in Marketing 42 (2025) 467–486

Contents lists available at ScienceDirect

International Journal of Research in Marketing

journal homepage: www.elsevier.com/locate/ijresmar

Full length article

If it ain't broke, should you still fix it? Effects of incorporating user feedback in product development on mobile application ratings

Zeynep Aydin Gokgoz [a,*], M. Berk Ataman [b], Gerrit H. van Bruggen [c]

[a] Sabanci Business School, Sabanci University, Turkey
[b] Faculty of Business, Özyeğin University, Turkey
[c] Rotterdam School of Management, Erasmus University, Netherlands

ARTICLE INFO

Article history:
First received on 12 July 2021 and was under review for 4½ months
Available online 17 October 2024

Area Editor: Peter C. Verhoef
Accepting Editor: Alina Sorescu

Keywords:
Mobile Apps
User Reviews
Updates
Product Development
Automated Text Analysis

ABSTRACT

How can firms utilize collective user feedback in reviews to better tailor their products to improve customer satisfaction? Based on an automated text analysis of 1,075,704 reviews and a content analysis of 3255 mobile application updates, observed over 460 apps' first year on the market, this paper investigates the role of collective user feedback on user ratings during the process of developing successive mobile app generations. The results reveal that the rewards associated with responding to user feedback and the penalties due to ignoring this feedback can be substantial. The impact of the match/mismatch between user feedback and product development decisions depends on the topic of the feedback and the timing of the update. The findings provide app developers with guidance on the challenge of prioritizing possible development paths: (1) *improve ratings* by promptly matching user feedback that require new content or smooth functioning of the app; (2) *avoid rating penalties* by continuously improving existing content or ensuring compatibility with most recent operating systems and devices. The implications shed light on the fundamental question of whether and when to pro-act on or re-act to the voice of the customer.
© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid progression of technology, coupled with the proliferation of platforms that publish customer reviews (e.g., manufacturer/retailer websites, customer review platforms, online blogs, and social media), has provided firms with ubiquitous access to valuable insights, such as latent user needs (e.g., Timoshenko & Hauser, 2019) or market structures (e.g., Netzer et al., 2012). Embedded within these reviews, either explicitly stated or implied, is customer feedback that might benefit firms in their ongoing product development endeavors. The effective utilization of this readily accessible and abundant feedback in continuous product development remains a promising avenue for research.

Firms can address the feedback in user-generated content in two different ways: reactive response, which involves merely replying to the comments, or proactive response, which entails taking direct action based on the feedback in the comments. Extant research has investigated the effects of the former (Chevalier et al., 2018; Proserpio & Zervas, 2017; Wang & Chaudhry, 2018), leaving the latter a relatively under-explored area (Dabrowski et al. 2022). Accordingly, we pose the fol-

* Corresponding author.
E-mail addresses: zeynep.aydin@sabanciuniv.edu (Z. Aydin Gokgoz), berk.ataman@ozyegin.edu.tr (M.B. Ataman), gbruggen@rsm.nl (G.H. van Bruggen).

https://doi.org/10.1016/j.ijresmar.2024.10.004
0167-8116/© 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

lowing empirical questions in this study: How can firms utilize the information in user reviews in the process of continuous product development? What are the effects of incorporating the feedback extracted from user reviews in the development of successive generations of a product on user evaluations? To what extent do these effects vary depending on the content of the requests, the nature of the implemented changes, and as a function of time?[1]

The greatest potential for capitalizing on the feedback in customer reviews lies in digital, technology-driven, and dynamic contexts, where firms can rapidly modify the value propositions of their offerings based on such feedback. Accordingly, we study this interlinked process of user reactions and subsequent product releases in the mobile applications (app hereafter) market, focusing on the early stages of apps' lifecycles. The rationale is as follows: The idea of commencing with a "minimum viable product" and pivoting has quickly taken root in the start-up world, replacing the "big design up front" approach (Blank, 2013). This trend has resonated most strongly with none other than app developers; experimentation and continuous innovation have become the norm in this industry (Anadea 2017). Consequently, the app market provides a unique setting to study whether firms can harness the opportunity of continuous interaction with their customers. The constant flow of reviews from users generates large amounts of data, from which one can extract feedback. Moreover, unlike traditional markets where product improvement usually takes a long time and is costly, app developers refine their products at relatively lower costs and in shorter timeframes. Frequent updates by developers provide many instances for analyzing the effects of (not) incorporating customer feedback in the development process.[2]

To address the effect of incorporating user feedback in the development of new product versions, we conduct an in-depth analysis of app reviews and textual descriptions of app updates. For this purpose, we amassed publicly available data on user reviews (1,075,704 reviews), the description of updates (3255 updates listing 12,268 specific changes), and user ratings for 460 apps in the Apple App Store during the first year of their existence. Content analysis of update descriptions led to the identification of four overarching update categories: bug fixes, compatibility with operating systems and devices, addition of new features or content, and improvement of existing features or content.[3] Adopting a lexicon-based approach to automated text analysis, we extracted actionable user feedback from reviews corresponding to these four broad categories. Given that lexica for such specific feedback did not previously exist, we created custom dictionaries. Using these data, we examined how the effect of releasing an update of a particular type on the average rating score of an app's new version varies with the intensity of the corresponding user feedback and the versions it takes the developer to respond to the feedback with an update. The expectations for the effects are guided by the Expectancy Confirmation Theory (Oliver, 1980) and the literature on customer complaints and inquiries (e.g., Davidow, 2003).

Our results reveal a nuanced view of the impact of (not) acting upon user feedback in terms of product evaluations, with implications for successive product generation development strategies. For instance, being available to a wide range of users with different devices and operating systems upfront appears crucial. We find that the negative effect of accumulated compatibility requests cannot be reversed by updates. This finding emphasizes the importance of adopting a *continuous product improvement strategy to maintain* the app's average rating score. A similar strategy is needed for updates pertaining to the improvements on existing app content, but for a different reason: users expect regular improvement of existing content and may penalize the developers for not doing so. In the case of issues concerning the essential functioning of the app (i.e., bugs) or requests for new app content, developers that respond fare even better. However, these rewards have an expiration date, at most as early as the first two versions following the feedback. This suggests prioritizing timely *compliance* on these fronts *to improve* the ratings.

The contribution of our research is primarily to the growing body of mobile apps literature. While the effects of updates and user reviews on app performance have been extensively studied in this domain, they are often studied in isolation. By acknowledging the inherent interdependency between updates and user reviews, we identify the circumstances in which app developers can extract value from user feedback and improve their product offerings to meet the needs of both existing and potential users. Additionally, we contribute to scholarly work at the intersection of voice of the customer and product development, particularly the emerging research on successive product generations (i.e., Bornemann et al., 2020). We demonstrate how and when firms can proactively utilize substantial amounts of organic customer feedback (i.e., mining reviews to identify problems, expectations, and growth opportunities) to steer development decisions for successive product generations (e.g., Palomba et al., 2018) and offer nuanced insights on how these decisions impact app performance. To our knowledge, our paper is among the first to conduct a comprehensive investigation at a large scale on how and when firms can benefit from feedback extracted from unstructured customer review data in continuous product development process.

[1] Time in this context refers to different app versions released sequentially, rather than traditional calendar time.

[2] Investigation of update descriptions confirms that developers actively monitor user reviews and update their apps in line with user requests: "You asked for it, so we built it: a new review mode lets you polish up either individual lessons or else all of the content you've completed so far.", "You asked for it and we're thrilled to deliver. This latest version includes your recent orders in "The Usual" section. Plus, reorder your favorites with one click.", "Soundscapes/white noise, you asked for it and we added it:)", etc.

[3] For brevity, we refer to "features or content" as "content" and "compatibility with operating systems and devices" as "compatibility" hereafter.

2. Research background

2.1. Mobile Apps: The role of user reviews and the impact of updates

Because of the lucrative business opportunities apps present, the market's rapid emergence, and its highly competitive nature, the mobile app market has drawn considerable interest from scholars (for comprehensive reviews, see Aydin Gokgoz et al. (2022) and Stocchi et al. (2022)). Research exploring the roles of user reviews and updates in mobile app success demonstrates that favorable product evaluations, as reflected in average ratings, increase an app's perceived utility (Hao et al., 2011), ultimately driving demand (Ghose & Han, 2014). Furthermore, this body of literature provides empirical evidence that updates are beneficial: demand is higher for regularly updated apps (e.g., Carrare, 2012; Ghose & Han, 2014). However, upon closer inspection, the evidence also suggests that the favorable impact of updating is contingent on the nature of the update. For instance, Aydin Gokgoz et al. (2021) discovered that major updates lead to more downloads than mid-level or minor updates, with the different levels of updates classified based on version number digit changes. Moreover, they show that the effects of various update types change over the life cycle of the apps. Kübler et al. (2018) advocated for localization of updates across countries for better performance. While all this evidence indicates that the nature of the update matters, extant research has not addressed how the specific nature of updates influences app performance.

More importantly, these studies investigate the effects of user reviews and updates on app performance independently, overlooking the inherent interplay between these two processes. Jang and Chung (2015) work is a notable exception. Turning their attention to the relationship between updates and reviews, the authors investigate the impacts of customer-to-customer and customer-to-firm interactions on app performance. They find that communication activities among customers and between customers and firms affect sales directly and product lifetime indirectly through increased R&D activity (i.e., updates). While Jang and Chung (2015) identify the dynamic nature of the app market as a fertile ground for examining the interplay between developers and users, their study overlooks a crucial facet of user reviews and updates: their content.

Moreover, app developers often grapple with resource constraints and aim to selectively incorporate the customer's voice into their development decisions. Yet, they struggle with the absence of a systematic approach.[4] Against this backdrop, we investigate how and when developers should mold their apps, offering updates of varying nature, to better meet user needs by leveraging insights from review content.

2.2. User Reviews: The voice of the customer

Online word-of-mouth has been a widely investigated topic ever since the rapid diffusion of digital platforms led to the proliferation of user reviews. As consumers heavily rely on the opinions of others' in their purchase decisions, early research in this area concentrated on the effect of user reviews, specifically examining the valence, variance, and volume of reviews (Dellarocas & Narayan, 2006), on sales performance (e.g., Anderson & Magruder, 2012; Chevalier & Mayzlin, 2006; Dhar & Chang, 2009; Duan et al., 2011; Forman et al., 2008; Liu, 2006; Luca, 2016; Vermeulen & Seegers, 2009; Ye et al., 2011; Zhu & Zhang, 2010). With advances in text mining and widespread availability of text analysis software allowing for an in-depth analysis of reviews' linguistic features, more recent studies have advanced our understanding of how certain linguistic characteristics impact the persuasiveness of reviews (e.g., Packard & Berger, 2017; Villarroel Ordenes et al., 2019) and, thereby, sales.

In addition to the relationship between (the linguistic characteristics of) user reviews and performance, reviews have been shown to provide a better, or an alternative, understanding of the drivers of choice, latent needs, market structures, and competitive positioning of brands (e.g., Archak et al., 2011; Büschken & Allenby, 2016; Lee & Bradlow, 2011; Netzer et al., 2012; Timoshenko & Hauser, 2019; Tirunillai & Tellis, 2014). Collectively, these studies illustrate the potential to extract useful information from reviews beyond their linguistic features.

One particularly valuable piece of information is the feedback that customers provide regarding their experiences with firms' offerings. While the motivations for writing a review may differ among customers, review content is replete with evaluations, complaints, and specific requests directed at, or at least informative for, the firm (Pagano & Maalej, 2013).[5,6] In the

[4] In addition to the evidence provided earlier (i.e., based on developer communication with users in app update descriptions), we interviewed two app developers to gain further insights. One developer managed a portfolio of 12 games with 50 million monthly active users spanning multiple countries and the other managed a finance app with 1 million active users. The interviews revealed that they take user reviews into account in their app development process, yet this process could benefit from a systematic approach. The need for a systematic approach is prevalent in both categories, though for different reasons. For finance apps, paying extra attention to functionality related feedback often overshadowed user feedback in other categories. For games, the immense volume of user feedback emanating from a huge userbase spread over a diverse set of tastes and preferences overwhelmed the developers. Both believed that they would greatly benefit from an overarching framework in the classification and the prioritization of actionable feedback. Also see Palomba et al. (2018) for a more comprehensive survey.

[5] Customers write reviews with different motives, including egoistic (e.g., receiving an economic benefit or enhancing one's reputation), intrinsic (e.g., benefiting from social interaction and identification), and altruistic motives (e.g., concern for other consumers or helping the company) (Hennig-Thurau et al., 2004; Yoo & Gretzel, 2008).

[6] In the app market, platforms are used extensively to communicate with the developer. Users may be internally motivated to do so, but they are also encouraged by the developers. Developers may dedicate a sentence in the update descriptions inviting users to provide feedback (e.g., "We are already working on version 1.2 iTuner Radio. Please send us suggestions of new features."; "Please review this latest app version! Sharing your feedback helps us improve the app with the new features you want.").

context of digital products, this feedback can span a spectrum of insights including, but not limited to, product functionality and performance (e.g., stability, responsiveness), features and capabilities (e.g., useful features, potential enhancements), user interface and experience (e.g., design assessment, ease of navigation), compatibility and integration with other software or devices, installation and setup complexity, and technical hurdles.

Regarding how managers handle this feedback, one can differentiate between passive and active forms of response. Firms' public responses to reviews can be seen as a reactive, or passive, form of response. Centered around review positivity/negativity, several studies have examined how firms' public responses to reviews influence subsequent product evaluations and provided mixed insights. For instance, Proserpio and Zervas (2017) discover that firms' responses to reviews lead to more positive subsequent reviews by increasing the potential cost of writing a negative review. Conversely, Chevalier et al. (2018) find that firms' responses encourage customers to write about their complaints, resulting in more negative reviews. Wang and Chaudhry (2018) distinguish between positive and negative reviews. They find that a public response to positive (negative) reviews can result in more negative (positive) subsequent reviews because it may raise questions about the sincerity of the response and can be interpreted as promotional (can signal that management is adopting a strategy to address the complaints).

While these studies shed light on how interactions between companies and customers, in the form of firms' public responses to reviews, affect subsequent product evaluations, it remains unclear whether the proposed solutions or promised changes actually lead to improved customer evaluations when implemented. Furthermore, it is uncertain whether the absence of a public response indicates that the firm is not listening to their customers' voices. Actively responding to customer feedback in the reviews, on the other hand, involves releasing an updated version of the current offering that incorporates the received feedback and falls under the purview of the product development literature.

2.3. New product development: Bringing in the voice of the customer

The new product development (NPD) literature strongly emphasizes the need to consider customer needs and integrate the voice of the customer throughout the various stages of the NPD process (Cooper, 1999; Griffin & Hauser, 1993). Research in this field spans studies on establishing mechanisms to incorporate customer requirements into the new product design phase (e.g., Bailetti & Litva, 1995) and the performance implications of customizing products based on customers' expressed preferences (e.g., Franke et al., 2009). Meta-analyses on the determinants of new product success emphasize the significance of meeting customer needs and engaging them both in the early and late stages of NPD (e.g., Chang & Taylor, 2016; Henard & Szymanski, 2001).

Coviello and Joseph (2012) illustrate that successful technology firms employ an unconventional approach to new product development, which involves iterative tasks and the involvement of consumers assuming diverse roles. They identify early engagement and conversation, as well as mindful trial and error, with a sufficiently diverse customer mix—central characteristics of the app market—as building blocks of a successful new product co-development process. Viewing customers as co-innovators who serve as sources of latent needs, and firms as facilitators of information sharing and mutual learning, the authors conceive of NPD for major innovations as a socially constructed and generative process. They advocate an approach akin to "almost ready, fire, do a bit more, refire" for its success, suggesting a prolonged, dynamic, and organic co-creation process. Consistent with this perspective, practices such as crowdsourcing ideas on digital platforms for product development have emerged as popular strategies for many established businesses beyond the technology sector in recent years (e.g., My Starbucks Idea, Lego Online Factory).

Traditionally, firms employed a mix of qualitative (e.g., in-depth interviews, focus groups, and observational studies) and quantitative research methods (e.g., surveys, conjoint analysis, concept tests, and experiments) to listen to the voice of the customer and guide their innovation efforts. This input can be characterized as being company-solicited, collected on an as-needed basis, and (semi-)structured. However, with the advent of digital technologies, big data and advanced data analytics have started to supersede traditional data sources and research methods, resulting in significant improvements in new product development performance (Cheng & Shiu, 2023; Del Vecchio et al., 2023). Chief among these data sources and research methods are online user reviews (i.e., a steady stream of customer-volunteered input) and automated text analysis techniques (i.e., tools for extracting useful information from unstructured data), which have proven useful for sourcing innovation ideas (e.g., Choi et al., 2020; Goldberg & Abrahams, 2022; Park et al., 2023; Qi et al., 2016; Shen et al., 2022).

While the existing body of work in this area adeptly distills valuable insights from user-generated content to guide firms in making informed product development decisions, it exhibits certain limitations. First, it tends to view the product holistically, primarily as a collection of tangible attributes, and perceives the next-generation product as a slightly differently configured set of the same tangible attributes. These, in combination, hinder the effective differentiation of alternative development paths. Additionally, most studies do not rigorously evaluate the effectiveness of implementing the development recommendations. Some assess the sensibility of the overall direction identified or the specific changes proposed by the algorithms by consulting a group of product owners or development engineers (e.g., Qi et al., 2016). Few evaluate, in a somewhat rudimentary manner, whether the set of identified changes holds the potential to enhance customer evaluations (e.g., Chen et al., 2019; Ciurumelea et al., 2017; Palomba et al., 2018; Zhou et al., 2021). In their review of the software engineering literature on advanced technical tools for mining informative content from app reviews, Dąbrowski et al. (2022) observe that only 3 % of the studies conduct a comprehensive impact analysis. Furthermore, most of these studies are confined to a single-period timeline, commencing arguably with the product launch and concluding with the hypothetical

launch of its improved version (i.e., the end of the review collection period). This decision restricts their ability to address time-related considerations in product development opportunities mined from user reviews (i.e., the urgency to promptly address certain issues versus the flexibility to take more time for others), an issue that is particularly relevant when resources are constrained.

This body of work that focuses primarily on the task of extracting diverse insights from user-generated content that can inform product development decisions, reflects a product-centric and data-driven decision-making mindset. Conversely, we adopt a more customer-centric and decision-driven data analytics mindset and place the emphasis on extracting value from the insights rather than extracting the insights themselves. To that end, we study the effect and timing of incorporating the extracted feedback in product development decisions on user satisfaction.

Consequently, our research lies at the intersection of works on voice of the customer and product development. It brings together these two inherently related, yet not always explicitly integrated bodies of work in the mobile apps domain (Stocchi et al., 2022), wherein the development process is characterized by continuous interactions with a sufficiently diverse user mix to understand their needs and an iterative and generative approach to product development.

Table 1 displays where exactly our paper is positioned at the juncture of these two streams of research. The rows in Table 1, reflecting the voice of the customer, differentiate the feedback obtained from unstructured customer-volunteered information from more structured, company-solicited feedback. The columns in Table 1, reflecting how firms handle the feedback, distinguish an active form of response, where the feedback guides the development of a successive product generation, from a passive form of response, where the feedback is merely acknowledged with a public response, yet its implementation is unobservable, rendering assessment of its effectiveness impossible.

3. The research setting: Reviews and updates in a mobile app's timeline

An app's launch marks the beginning of the timeline of our research. Following the initial release, the developer *updates* the app at specific, yet not necessarily regular, points in time. Each update is the result of a development effort, where the developer fixes, modifies, and/or enhances the app. Running parallel to the sequence of updates is the continuous process of evaluation by the app's users. Upon downloading the most up-to-date version of the app and interacting with it for some time, users can express their overall satisfaction with a *star-rating*. These evaluations are often accompanied by a review, wherein the users justify their ratings for the benefit of other users and/or write *feedback* to the developer to request certain changes to the app. Consequently, an app's observed timeline can be expressed as a repeated cycle of two continuous processes: updating (by the developer) and reviewing (by the users). Each cycle begins with an update that outsets a newer version of the app, which is then rated and reviewed by the users, and ends with the release of the next version. Fig. 1 illustrates these cycles using a gaming app's daily star rating distribution and updates. To demonstrate the phenomena under investigation, we subsequently discuss the interplay between user reviews and updates across the cycles, examining their impact on user evaluations. Fig. 2 displays the general research setting. In both figures, the developer's timeline is presented in the upper half of the figure and that of the users in the lower half.

The substantial changes in the star rating distribution in Fig. 1 following the 4th, 7th, and 8th updates are particularly noteworthy. A quick inspection of the review content preceding the 4th update of the game reveals that (1) user feedback upon which the developer can act is virtually non-existent and (2) the little actionable feedback is scattered across different topics, failing to steer the development efforts. Juxtaposed against this rather blank canvas, the developer releases the 4th update that enhances the app's value proposition through new and improved content (see Fig. 1 for update descriptions), which leads to a flurry of favorable ratings (see Fig. 1 for a sample of user reviews).

Different from this example, when the developer releases the 7th update, the game stops functioning properly. Investigation of the subsequent reviews reveals that users predominantly complain about a bug in the new version and the proportion of 1-star ratings increases. Timely realization of this problem and the developer's effort to fix this issue in the 8th update results in praise in the reviews following the new version's release and the recovery of the app's ratings.

Also likely to observe, unfortunately not in this select example, are cases wherein the feedback points to a clear development direction but the developer does not address the feedback or respond to it in a timely manner. Such cases beg the question whether the failure to comply or the delayed action will have undesirable effects on users' evaluation of the apps.

4. Conceptual framework and expectations

Updating an app has the potential to improve its rating, as with each update the developer offers an improved version of its product to its current and potential users. To guide the decision of what to change in the new version, the developer may turn to the reviews of the most up-to-date version of the app, wherein users voice their specific demands and request certain changes to the app. Some of these demands will overlap and be voiced by many (e.g., the reviews following the 7th update in Fig. 1), suggesting a specific route for the next set of development tasks. Others will be scattered and voiced by only a few (e.g., the reviews preceding 4th update in Fig. 1). The developer can choose to update the app (1) in line with user demands or (2) different from users' feedback, either because specific requests have not been voiced by many or they choose to ignore the feedback.

Table 1
Customer Feedback vs. Company Response.

		Response to Feedback	
		Passive *(Implementation not traceable)*	Active *(Implementation traceable)*
Customer Feedback	Company-solicited(semi-) structured feedback	Post-launch satisfaction surveys, focus groups, in-depth interviews, or observational studies (e.g., Kenett & Salini, 2011)	Traditional pre-launch NPD research utilizing a mix of qualitative and quantitative techniques informing development decisions(e.g., Griffin & Hauser, 1993)
	Customer-volunteered unstructured feedback	Monitoring social media conversations, tracking user-generated content (i.e., reviews) and issuing public responses(e.g., Chevalier et al., 2018; Proserpio & Zervas, 2017; Wang & Chaudhry, 2018)	Extracting and integrating a steady stream of user feedback in the development of successive product generations **This paper**

Given that customer satisfaction can be expressed as a function of the difference between the experience with the product and initial expectation (Anderson & Sullivan, 1993), we posit that user feedback will gradually build up product expectations. The developer can intervene this process and impact expectations through updates. Specifically, we propose that the content and timing of the update, as well as the alignment between the user requests and implemented changes, will determine how favorably the new version will be evaluated by users.

To develop our expectations, we turn to Oliver's Expectancy Confirmation Theory (1980) (referred to as ECT hereafter), which consists of four constructs: expectations, perceived performance, disconfirmation of beliefs, and satisfaction. Table 2 depicts the possible scenarios along the user feedback and update axes, positioned in four different regions. While developing and visualizing our expectations, we consider the extreme cases of feedback intensity at an almost dichotomous scale for simplicity. However, in our empirical application, we treat user feedback as a continuous variable ranging between zero (i.e.,

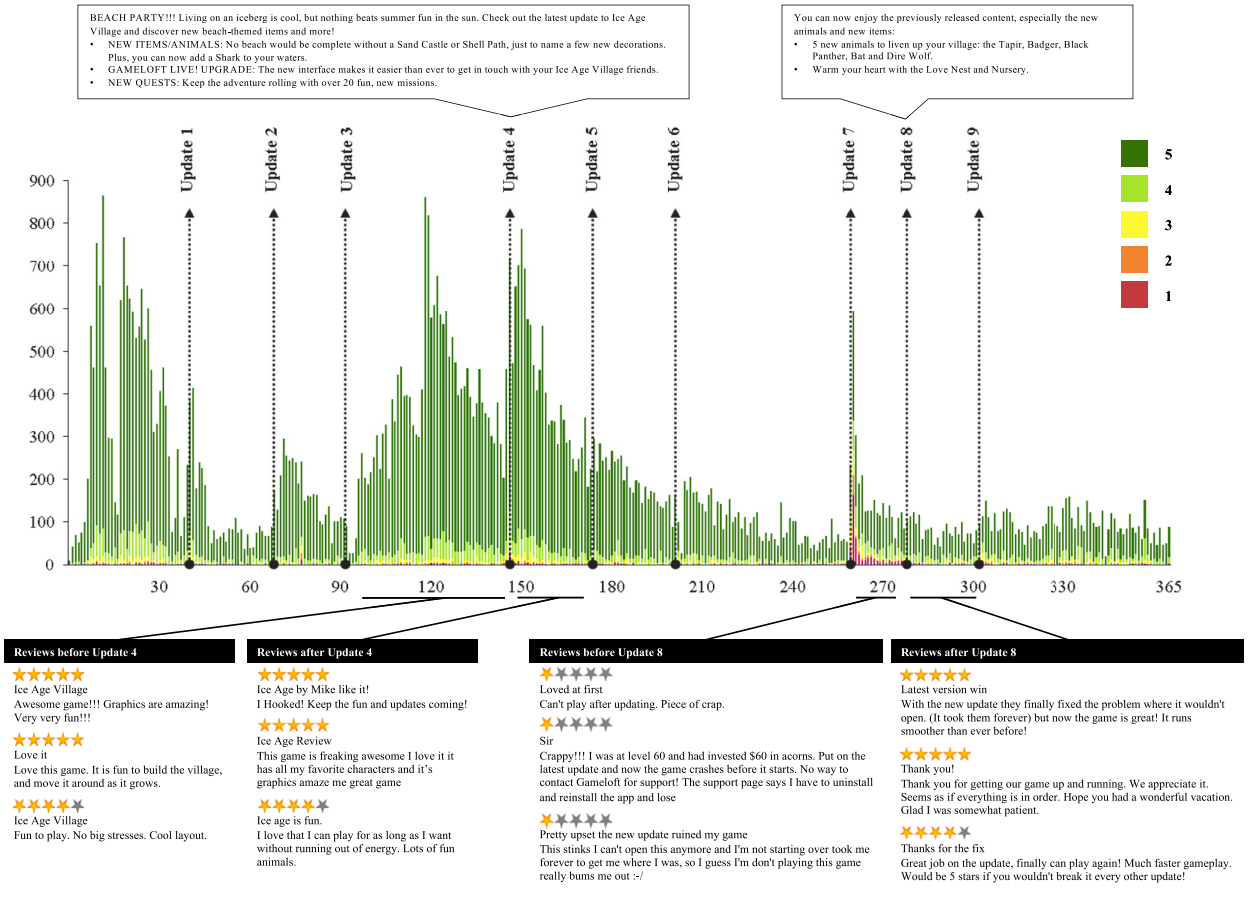


Fig. 1. Daily Review Distribution and Update History of a Gaming App.

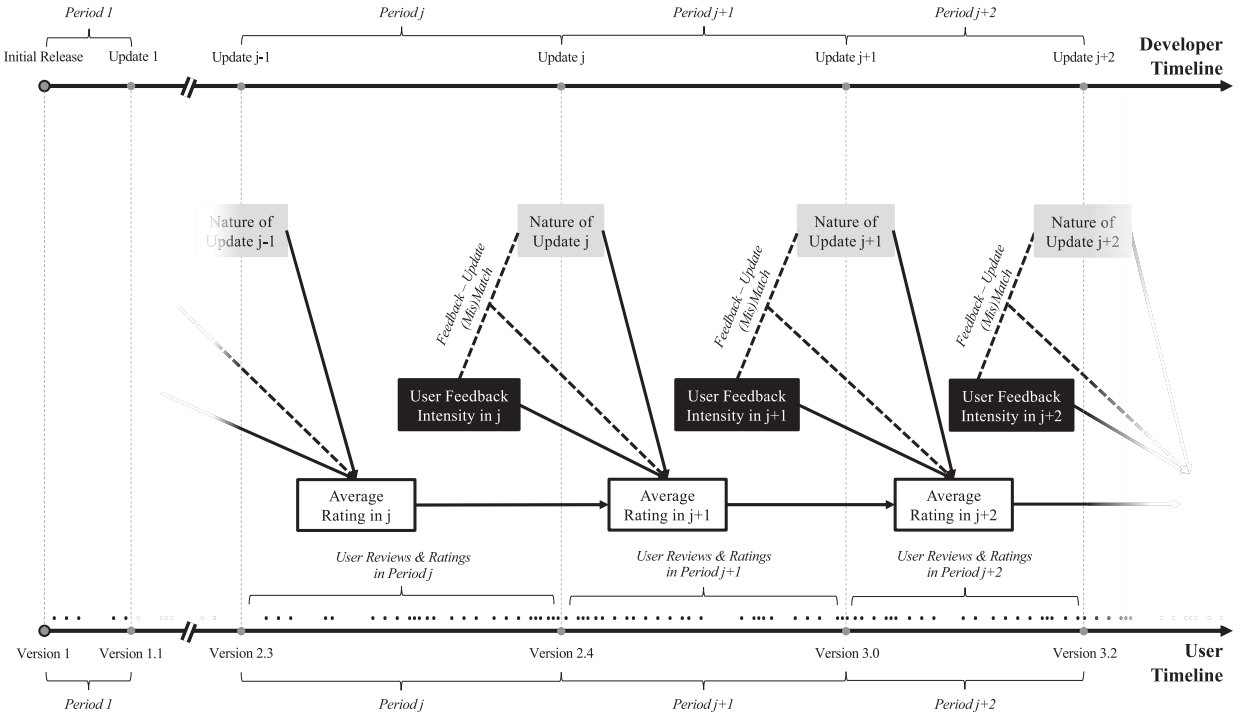


Fig. 2. The Research Setting: Reviews and Updates in a Mobile App's Timeline.

none of the review writers mention a specific issue) and one (i.e., all reviews mention a specific issue), hence the label 'feedback intensity'.

Of the three possible and meaningful scenarios, the first one corresponds to the situations wherein users request specific changes with a unified voice (i.e., feedback intensity is high), and the developer incorporates these requests in the update. We refer to this scenario as *compliance*. Meta-analysis results on the drivers of NPD success (e.g., Chang & Taylor, 2016; Henard & Szymanski, 2001), as well as confirmation cases in ECT (i.e., zero disconfirmation), would suggest that the new version of the app should be evaluated more favorably than the former version. Regarding the timing of compliance, based on insights from the literature on customer complaints and inquiries (Davidow, 2003; Istanbulluoglu, 2017; Moore & Moore, 2004; Wang & Chaudhry, 2018), we expect that user evaluations will be lower as the developer takes longer to address these topical unmet needs in an update.

The second scenario is the reversal of the former situation. It corresponds to cases wherein the users almost unanimously voice specific complaints or explicitly request certain changes (i.e., feedback intensity is high), but the developer either chooses to ignore or fails to respond to these requests in the next version of the app. We refer to this scenario as *non-compliance*. In ECT, non-compliance corresponds to negative disconfirmation, as user expectations are left unmet. We expect a non-compliant developer to be penalized and the app's new version to be rated lower than the previous version.

Regarding how penalties for non-compliance change with delay, we cannot afford a directional prediction. On the one hand, the literature on customer complaints and inquiries cited earlier would suggest that the severity of the penalty will increase as the developer continues to ignore user feedback. On the other hand, one could argue that the feedback will be forgotten, become gradually invisible to other users, or its relevance will diminish over time. Moreover, the users may habituate to the current version of the app (Chen et al., 2022), barring all changes along other aspects, or develop workaround solutions to get what they need from the app. These latter points suggest that the severity of the penalty will decrease over time.

In the final scenario, users do not request specific changes or voice specific complaints (i.e., feedback is virtually non-existent), yet the developer anticipates latent needs and proactively provides them with an update that meets these needs. Given the absence of feedback and the release of an update intended to be the new and improved version of the app, we refer to this scenario as *surprise*. In line with ECT, the developer should enjoy higher ratings for the new version to reflect the awe factor for offering a superior app (i.e., positive disconfirmation).

Inasmuch as user expectations are raised following this experience, it may be harder for the developer to enjoy similar benefits in the near future. Rust and Oliver (2000) argue that firms that delight their customers can benefit from forgetting, enabling the opportunity to recreate the delightful experience. However, letting several updates pass before surprising users can have an undesirable effect as well. Over time, customers develop certain usage patterns. The release of an update featuring a "surprise" may distort these patterns and impose learning costs, ultimately lowering user satisfaction. However, the findings of Chen et al. (2022) reveal that, although these fears are real, they are short lived, and the developer may ulti-

Table 2
Summary of Expectations.

		Developer Update	
		No	Yes
User Feedback	Present	Scenario: *Non-Compliance* Expected Effect: *Negative* ECT Reason: *Negative disconfirmation* Timing: *Intensify or Diminish with Delay*	Scenario: *Compliance* Expected Effect: *Positive* ECT Reason: *Zero disconfirmation* Timing: *Diminish with Delay*
	Absent		Scenario: *Surprise* Expected Effect: *Positive* ECT Reason: *Positive disconfirmation* Timing: *Intensify with Delay*

mately benefit from breaking these usage patterns. Accordingly, we expect the rewards associated with surprising users to increase with delay.

To recap, we posit that the extent to which a developer incorporates the voice of the user into update decisions will play a critical role in users' satisfaction with the app. Specifically, we argue that the change in ratings between two consecutive versions of an app will be influenced by the content of the update that outsets the newer version and its correspondence – or lack thereof – to the feedback from user reviews for the preceding version. We expect an increase in new version rating scores in *compliance* and *surprise* cases, and a decrease in *non-compliance* cases. Furthermore, we expect compliance effects to diminish and surprise effects to intensify as developers take longer to update. Delays may or may not be beneficial for the developer in non-compliance cases. Fig. 3 displays the corresponding conceptual framework.

5. Data

To investigate whether and to what extent the correspondence between user feedback and update content – or lack thereof – influences the evaluation of an app's new version, we have put together a unique data set on user reviews and app updates for a total of 1011 mobile apps – a stratified random sample taken from 40,000 newly-released apps on the Apple App Store in 2012.[7] As experimentation and continuous innovation are more likely to be observed early on in an app's lifecycle, we focus on the first year of each app's existence. For the sampled apps, we crawled the publicly available (1) update descriptions provided by the developers and (2) reviews written by app users from the Apple App Store and data.ai (formerly App Annie), a leading mobile analytics company. The initial data set contained approximately 1.3 million reviews and 6200 update descriptions with a total of 16,000 lines indicating specific changes. We screened out (i) apps that were never updated or updated only once, as we need at least two fully observed periods for the analysis (i.e., two updates after the launch), and (ii) apps with less than 50 reviews in total to ensure we have enough reviews per period. The final data set comprises 460 apps, with a total of 1,075,704 user reviews and 3255 updates, detailing 12,268 specific changes.

5.1. Identifying update categories and classification of update descriptions

An essential component in our study is the developer's actions; specifically, the nature of changes a developer implements in an app's new version. To infer the content of an update, we turn to the update description texts accompanying each new version (i.e., What's new in Version #.#?). These are concise and organized descriptions provided by the developers, for the benefit of their current and potential users, listing the specific changes made in the app's new version (see Appendix B1 for an example).

To map these specific changes onto different update types, we followed a two-pronged procedure: first, identifying different types of updates and, second, coding the specific changes into these update types. For the former, we developed an initial list of update types by content analyzing update descriptions and reviewing popular publications and blogs. Next, we fine-tuned this list through discussions with a 'user-experience research and design' expert. We provided the expert with the initial list of update types, their definitions, and a random sample of update descriptions extracted from another app store. We then asked the expert to (1) merge or split update types, (2) propose new ones, and (3) evaluate the coverage and representativeness of their final proposal by applying it to the provided sample. Table 3 presents the final list of update types (2nd column), their definitions obtained after this iteration (3rd column), and examples from our sample (4th column).

Because the number of update descriptions in our final sample render human coding possible, we took this opportunity to code the specific changes into the identified update types. To that end, we enlisted a research assistant with a background in

[7] The list of new apps released in the first five months of 2012 was acquired from a leading mobile analytics company Distimo, now owned by data.ai (formerly known as App Annie), as part of a larger data set and complemented with publicly available data later. To accurately reflect the app business model (i.e., free or paid) and app category (e.g., entertainment, lifestyle, social networking, games, etc.) distribution in the Apple App Store, we drew a random sample from each business type-category combination proportional to its share in the App Store. Table A1 and Figure A1 in the appendix show, respectively, the definitions and examples of app categories in the App Store and the distribution of apps across categories in the App Store and our sample.

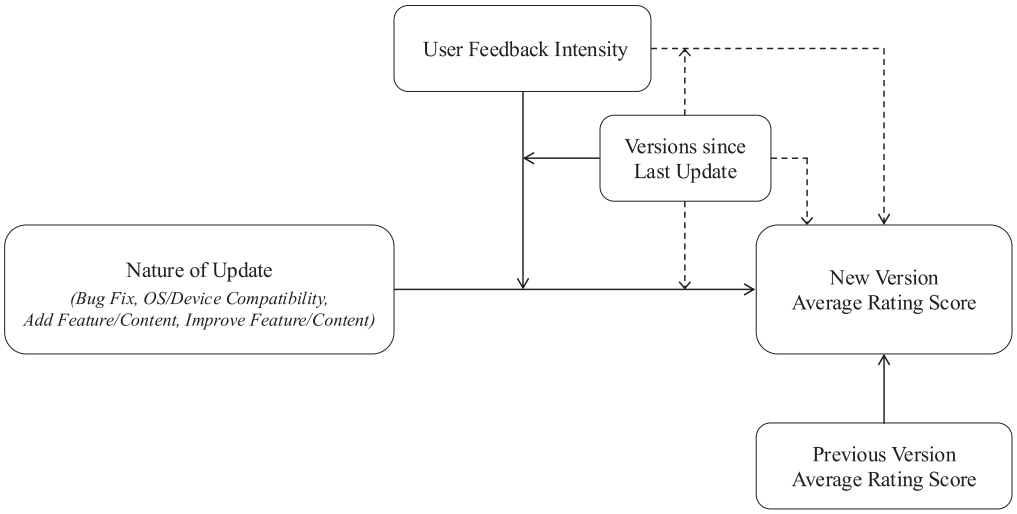


Fig. 3. Conceptual Framework.

computer engineering, who was pursuing a graduate-level degree in marketing at the time of coding. We provided the research assistant with the final list of update types, along with their definitions, some examples, and a set of instructions (see Appendix B2). The research assistant then independently coded the entire update description dataset, consulting the researchers for clarification when needed (see Appendix B1 for an example).

A careful examination of the content and the definitions of different update types reveals seven higher-order update categories (1st column in Table 3): Bug Fix (i.e., indicates minor improvements that ensure the app works as intended), OS / Device Compatibility (i.e., ensures that the app works on different devices or functions properly in most up-to-date operating systems), Add Content / Feature (i.e., adds new content to the app or new features, such as allowing users to personalize, customize, work offline, work in harmony with other apps, etc.), Improve Content / Feature (i.e., improves the existing content or features of the app), Remove Content / Feature (i.e., disables undesirable content or features, such as intrusive ads), Sales / Offers (i.e., enables in-app purchases or encourages cross-selling), and Other (i.e., update description entries, such as feedback requests, that do not indicate an improvement or that are not clear enough to be classified).

In our empirical application, we focus on the four higher-order update categories of "Bug Fix", "OS/Device Compatibility", "Add Content/Feature" and "Improve Content/Feature".[8] We exclude updates that fall under "Remove Content/Feature" due to insufficient number of observations, "Sales/Offers" because price changes would limit the analyses to a subset of the sample (i.e., paid apps or apps offering in-app purchases) and they are not aligned with updates, and "Other" because they are not about improvements to the app. The frequency distribution of these four update categories is presented in Table 4, Panel A.

Of the 3255 updates across all apps, 2128 are classified as "Bug Fix", 1704 as "Add Content/Feature", 1542 as "Improve Content/Feature", and 492 as "OS/Device Compatibility" updates.[9] On average, an app is updated 7.1 times, with the majority being classified as bug fixes (4.6 bug fix updates per app), followed by updates adding new features or content (averaging 3.7 times). Updates improving existing features or content occur at a slightly lower frequency, (averaging 3.4 times). Updates addressing OS or device compatibility issues are the least frequent (averaging just 1.1 updates).

5.2. Extracting feedback from user reviews

The feedback users provide to the developers is the other essential component of our study. To decide how best to approach the task of feedback extraction, we explored the reviews to understand common user-requests and the way these requests were expressed. The exploration revealed that the reviewers use a daily and informal language with plenty of abbreviations and content-specific words, rendering a human-driven investigation more suitable than machine learning techniques. Accordingly, we decided to adopt a lexicon-based approach to feedback extraction. However, as lexica for user feedback corresponding to our specific update categories do not exist, we developed *feedback-category dictionaries* from scratch, following a systematic approach (see Berger et al. (2019) and Humphreys and Wang (2018) for details pertaining to dictionary construction and lexicon-based automated-text analysis).

[8] As indicated by the 'Update Type' column in Table 3 and the coding instructions provided in Appendix B2, we initially employed a more detailed breakdown of the nature of updates. However, due to an insufficient number of observations per update type, we were compelled to conduct the analysis at the higher-order update-category level.

[9] Nearly 34% of all bug fixes (720 update instances), 18% of compatibility updates (89 instances), 20% of updates that add new features or content (348 instances), and 14% of improve existing feature/content updates (214 instances) happen in isolation.

Table 3
Definition and Examples of Update Categories.

Update Category	Update Type	Definition	Example
Bug Fix	Bug Fix	Bug fixes are minor updates that ensure that the application works, as it should	"Minor bug fixes and tweaks", "Fixed a bug with crashes when a device is offline"
Compatibility	OS/Device Compatibility	Developer updates the application to ensure that it works on different devices or functions properly in most up-to-date operating systems	"ios6 compatibility release", "Compatible with more devices"
Add Content/ Feature	Personalization/ Customization	Allows the user to customize the application to her taste and remembers previous choices	"Your language selection is saved but can be changed again any time later", "Remembers login"
	Offline Functionality	Allows users to access the application/use it without connecting to a network	"In the new tab "Other" you can now (even in the free version) download all mp3 audio files for all 100 lessons and in all language combinations"
	Location Sensitivity	The application is location sensitive (GPS enabled)	"GPS Track View, Play, and Record functions has been added. User could see GPS Track on the Map with Photos. GPS Track also can be played and saved (recorded)"
	Sharing/Social Networking	Allows the user to connect with others in her social network	"You can now share with Facebook, Twitter or another social network!!"
	Feedback	Allows the user to connect with the developer directly	"You can now write us directly from the settings screen. We love your feedback!"
	New Content	Offers the user content that was not present in the previous version of the application	"New levels", "Helicopters, Shields and Weapons"
	Integration/ Synchronization	Allows the application to interact with other applications/web-sites	"Heyzap integration (The largest community of mobile gamers)", "Share Link option for better integration w/ other Web Browsers"
	Localization	Offer the users local features such as new languages, cities, states, geographies, etc.	"New localizations: Spanish, French and Italian are now supported", "English and Chinese languages are now supported"
Improve Content/ Feature	Personalization/ Customization	See above.	"Optimize the Theme Customization process", "More customization supports"
	Offline Functionality	See above.	"Save, browse and load song files"
	Location Sensitivity	See above.	"Improved Current Location functionality by implementing the Google Maps My Location marker"
	Sharing/Social Networking	See above.	"Made some Enhancements to sharing a story when posting it", "Changed login process to use new Facebook standard"
	Feedback	See above.	"Fine-grained feedback"
Improve Content/ Feature	Existing Content	See above.	"Improved libraries"
	User Interface	Includes enhancements in the user interface (i.e., design and/or placement of buttons, etc.)	"Images and colors can be used as backgrounds (16 custom wallpapers included)", "Hour and battery left on menu button"
	Gestures	Includes enhancements related to gestures (i.e., gestures needed to work and control the app).	"Paddle at the bottom of the screen to move the ball and fired to modify the area to be distinguished", "In this version you can enter your own currency rate by long press on currency in the list"
	Performance	Smaller file size, improved speed etc.	"Changed database access code to avoid locking on faster phones", "About 20% reduction in memory usage for most of themes!! More fluid now"
Remove Content/ Feature	Content/ Features	See above.	"Building underwater no longer possible"
	Advertisements	Removal of advertisements within the app	"New: No more ads!", "Banner ads removed so there will be more space for windows"
Sales/Offers	In-App Purchase	Enables In-App Purchase option for the application for the first time	"Add in app purchase function", "Dietary Supplement – now you can shop right from our App to purchase D'Adamo Personalized Nutrition® supplements (see More, then Dietary Supplements)"
	Cross-selling	Ads of other applications by the developer	"Have you tried our other game in the series, Grandpa's Workshop?"
	Special Offers	Seasonal Promotions such as "Christmas" or "Valentine's day" or limited time offers	"4th July Sale 50% off Imperial Store items!"

Table 3 (continued)

Update Category	Update Type	Definition	Example
Other	Feedback Requests	Asks the user to connect with the developer to rate the app, like app's Facebook page, etc.	"Feel free to write us about any bugs or improvements to blueplopgames@gmail.com"
	News/ Information	Information about upcoming features of the app or any other news	"Coming soon:- App support for hands-free calling with Bluetooth™ compatible phones."
	Tips	Guides the user through various features of the application	"TIP: You can always double tap any day to mark them as complete or to unmark them."
	"Thank you!" Notes	Includes thank you notes for users from developers	"Thanks for your awesome reviews and feedback!"
	Cryptic	The description does not allow the coder to classify the update	Empty Updates or Updates with version history only.

Table 4
Data Descriptives.

Panel A: Summary Statistics of App Updates and User Reviews

					Percentiles		
	Total	Mean	Std. Dev.	Min	25th	50th	75th	Max
Number of Updates	3255	7.076	4.454	2	4	6	9	35
Bug Fix	2128	4.626	3.654	0	2	4	6	24
OS/Device Compatibility	492	1.070	1.037	0	0	1	2	5
Add Feature/Content	1704	3.704	3.240	0	2	3	5	21
Improve Feature/Content	1542	3.352	2.946	0	1	3	5	21
Period Length		44.024	47.695	1	13	28	56	373
Number of Reviews	1,075,704	330.477	3564.325	1	10	29	111	170,997
Bugs	63,472	19.500	89.174	0	1	3	11	2321
OS/Device Compatibility Issues	32,746	10.060	49.028	0	0	2	6	2040
New Content/Feature Requests	42,806	13.151	54.391	0	0	2	9	1998
Improve Content/Feature Requests	34,064	10.465	50.979	0	0	2	6	1282

Panel B: Summary Statistics of Model Variables

				Percentiles			
	Mean	**Std. Dev.**	**Min**	**25th**	**50th**	**75th**	**Max**
Average Rating	3.739	1.015	1.000	3.080	4.046	4.539	5.000
Update							
Bug Fix	0.654	0.476	0	0	1	1	1
OS/Device Compatibility	0.151	0.358	0	0	0	0	1
Add Feature/Content	0.524	0.500	0	0	1	1	1
Improve Feature/Content	0.474	0.499	0	0	0	1	1
Feedback Intensity							
Bugs	0.147	0.180	0	0.017	0.085	0.209	1
OS/Device Compatibility Issues	0.090	0.137	0	0.000	0.038	0.119	1
New Content/Feature Requests	0.103	0.138	0	0.000	0.059	0.140	1
Improve Content/Feature Requests	0.069	0.102	0	0.000	0.038	0.095	1
Versions Since Last...							
Bug Fix Update	0.581	1.242	0	0	0	1	11
OS/Device Compatibility Update	2.482	2.725	0	0	2	4	18
Add Content/Feature Update	0.777	1.344	0	0	0	1	11
Improve Content/Feature Update	1.041	1.785	0	0	0	1	17

First, we created initial feedback-category dictionaries from the language used in update descriptions, as updates are aligned with specific user-request categories. This was achieved by identifying common words and phrases used in user reviews to complain, comment, or request content/features. Next, we expanded these initial feedback-category dictionaries in two stages. In the first stage, a stratified random sample of user reviews was tagged by judges, identifying words and phrases related to specific feedback categories. The second stage involved extracting unigrams, bigrams, and trigrams from user reviews, and adding any category-specific phrases that were absent based on their frequency. The dictionaries were further augmented by seeking related words online. Finally, the dictionaries were fine-tuned by eliminating positive or neutral phrases and adding negative phrases to focus on actionable developer feedback. This multi-step process resulted in comprehensive feedback-category dictionaries. For a detailed view of the process and a sample of the final dictionaries, we refer the reader to section C in the Appendix.

Using the custom-built dictionaries, we extracted the specific user feedback from all 1,075,704 reviews in the sample following the standard steps of automated text analysis (e.g., Berger et al., 2019). Each user review was normalized and parsed before extraction. Specifically, we tokenized all reviews, removed all non-alphanumeric and non-Ascii characters, and converted all letters to lowercase. We then stemmed all words to their roots using Snowball Stemmer (i.e., Porter2 Stemmer). The same stemming was applied to feedback-category dictionaries. To ease computational burden, we removed stop-words using a context-specific list and uncommon words (i.e., words that occur less than 10 times).[10]

Our feedback-category extraction algorithm simply matches the words/phrases extracted from the reviews to the set of words/phrases in each of the four feedback-category dictionaries. Specifically, a review is assumed to contain user feedback on a specific topic if it contains one or more of the dictionary entries of the corresponding feedback-category. As a review may contain feedback on multiple categories or none, we represent the result of the matching algorithm as a binary array of size four.[11] The summary statistics pertaining to the extracted feedback are presented in Table 4, Panel A.

Out of a total of 1,075,704 reviews, a significant portion focuses on bugs, with 63,472 reviews pointing out functionality issues that need fixing. OS/device compatibility issues also garner considerable attention, with 32,746 reviews mentioning problems in this area. Additionally, users are proactive in requesting new content or features, with 42,806 reviews. Finally, there are 34,064 reviews demanding improvement of existing content or features. On average, an app in our sample receives 330.5 reviews between two consecutive versions (SD = 3564.3).[12] Of these reviews, 19.5 (SD = 89.2) mention bug related issues, 13.2 (SD = 54.4) request new content or features, 10.5 (SD = 51.0) suggest improvements to existing content or features, and 10.1 (SD = 49.0) complain about compatibility issues.

Using the binary arrays, we calculate the intensity of feedback for each of the four categories. To that end, we count number of reviews containing feedback on a specific feedback category posted between two consecutive versions of the app and normalize it by the total number of reviews posted in the same period. The summary statistics of the four feedback intensity variables are presented in Table 4, Panel B.

6. Model specification

To arrive at a specification that examines how the (mis)match between user feedback and developer updates affects users' evaluation of a mobile app, and how these effects change with delay, we begin with a simple model structure that assumes that the rating of the current app version is an update-dependent adjustment of the previous version's rating:

$$ARS_{i,v} = \lambda ARS_{i,v-1} + g(U^{j}_{i,v}) \tag{1}$$

where $ARS_{i,v}$ represents the *average rating score* of version v of app i. It is calculated as the average of all ratings the app receives within a time window that starts with the launch of the new version and ends with the release of the next version. $U^{j}_{i,v}$ is a dummy variable indicating whether the improvements made in version v of app i belong to *update category* j, where j = 1 for bug fixes, j = 2 for add new content, j = 3 for improve existing content, and j = 4 for compatibility and $g\left(U^{j}_{i,v}\right) = \sum_{j=1}^{4}\left(\beta_{0j} + \beta_{1j}U^{j}_{i,v}\right)$.

As we posit that the impact of offering (and not offering) an update of a particular kind on the average rating score depends on corresponding user feedback, we express $\beta_{kj} = \alpha_{k0j} + \alpha_{k1j}FI^{j}_{i,v-1}$, where $k \in \{0,1\}$ and $FI^{j}_{i,v-1}$ is the *intensity of user feedback* in j-th feedback category for the former version of app i. Feedback intensity, $FI^{j}_{i,v-1}$, is calculated as the ratio of the number of reviews referencing a specific feedback category to the total number of reviews for the former version of the app. This yields the model with the following structure:

$$ARS_{i,v} = \lambda ARS_{i,v-1} + \sum_{j=1}^{4}\left(\alpha_{00j} + \alpha_{01j}FI^{j}_{i,v-1} + \alpha_{10j}U^{j}_{i,v} + \alpha_{11j}\left(FI^{j}_{i,v-1} \times U^{j}_{i,v}\right)\right) \tag{2}$$

Dropping the update and feedback category identifier j for simplicity, α_{10} in equation (2) captures the effect of offering an update on the average rating score in the absence of corresponding user feedback (i.e., $FI_{i,v-1} = 0$). We refer to this as the surprise effect. Similarly, α_{01} measures how the average rating score changes with user feedback when there is no update (i.e., $U_{i,v} = 0$). This coefficient captures the potentially deleterious effect of not responding to user requests (i.e., penalties).

[10] First, for details on the creation of this context-specific stop-words list, we refer the reader to section C2 in the Appendix. Second, there are 127,613 unique word stems in user reviews, with a total count of 7,849,239 stems. However, the tail of the word stem frequency distribution is extremely long. 11,138 unique stems (8.7% of all unique stems) appear 10 times or more and make up 97.7% of the entire corpus. The 116,475 unique stems (91.3% of all unique stems) that appear less than 10 times have a total count of only 182,678 stems, making up 2.3% of the entire corpus. Following Fu et al. (2013), we consider words appearing less than 10 times as uncommon and treat them as stop-words. Considering that higher frequency cutoffs have been applied in the construction of feedback-category dictionaries (please see Appendix C), removal of uncommon words has no bearing on the feedback extracted.

[11] Alternatively, one can 'score' user feedback by counting the number of tokens in a review matching the entries in the feedback-category dictionary. The reasons for refraining from scoring are twofold. First, we do not want to place undue weights to verbose review-writers. Second, it is the absence or presence of feedback on a specific issue that matters for the developer, not its length.

[12] These reviews accumulate over a period (i.e., the time window bookended with the release of two consecutive versions of an app) that lasts 44.0 days on average (SD = 47.7).

Therefore, we refer to it as the non-compliance effect. α_{11} measures how the effect of an update changes with varying levels of feedback intensity. As this coefficient captures whether and to what extent an update that addresses user requests offsets the potentially undesirable effects of non-compliance (i.e., countervailing rewards), it is referred to as the compliance effect.

Finally, to explore how the effects of the (mis)match between user feedback and developer updates on users' evaluation of a mobile app change with delay, we express the coefficients in equation (2) as functions of the number of *versions since the last update*. Specifically, $\alpha_{klj} = \phi_{kl0j} + \phi_{kl1j} VSLU^{j}_{i,v}$, where $k, l \in \{0, 1\}$ and $VSLU^{j}_{i,v}$ counts the number of versions released since the developer last offered an update in category j.[13] The structure of the full model is:

$$ARS_{i,v} = \lambda ARS_{i,v-1} + \sum_{j=1}^{4} \left(\phi_{000j} + \phi_{001j} VSLU^{j}_{i,v} + \phi_{010j} FI^{j}_{i,v-1} + \phi_{011j} \left(FI^{j}_{i,v-1} \times VSLU^{j}_{i,v} \right) + \phi_{100j} U^{j}_{i,v} + \phi_{101j} \left(U^{j}_{i,v} \times VSLU^{j}_{i,v} \right) + \phi_{110j} \left(FI^{j}_{i,v-1} \times U^{j}_{i,v} \right) + \phi_{111j} \left(FI^{j}_{i,v-1} \times U^{j}_{i,v} \times VSLU^{j}_{i,v} \right) \right) \quad (3)$$

With the update and feedback category identifier j omitted for simplicity, the coefficients ϕ_{100}, ϕ_{010}, and ϕ_{110} measure the surprise, non-compliance, and compliance effects, respectively, when the previous app version includes the particular update (i.e., $VSLU^{j}_{i,v} = 0$). Meanwhile, the coefficients ϕ_{101}, ϕ_{011}, and ϕ_{111} capture how the rewards from surprise, penalties from developer non-compliance, and countervailing rewards from compliance change with delay, respectively.

Because we have data from multiple apps with distinct characteristics (categories, business models, etc.), capturing systematic differences across ratings and controlling for the effects of the potentially omitted time-invariant unobservables (e.g., app popularity, inherent quality) is critical. Therefore, we estimate the regression model presented in equation (4), which features ϕ_i, an app-specific fixed effect.[14] Moreover, the presence of lagged rating scores also helps us to control for the effects of omitted unobservable time-varying factors.

$$ARS_{i,v} = \phi_i + \lambda ARS_{i,v-1} + \sum_{j=1}^{4} \left(\phi_{001j} VSLU^{j}_{i,v} + \phi_{010j} FI^{j}_{i,v-1} + \phi_{011j} \left(FI^{j}_{i,v-1} \times VSLU^{j}_{i,v} \right) + \phi_{100j} U^{j}_{i,v} + \phi_{101j} \left(U^{j}_{i,v} \times VSLU^{j}_{i,v} \right) + \phi_{110j} \left(FI^{j}_{i,v-1} \times U^{j}_{i,v} \right) + \phi_{111j} \left(FI^{j}_{i,v-1} \times U^{j}_{i,v} \times VSLU^{j}_{i,v} \right) \right) + \varepsilon_{i,v} \quad (4)$$

The summary statistics of the variables in equation (4) are presented in Table 4, Panel B.

7. Results

Table 5 displays the parameter estimates of the three models presented in Section 6. In what follows, we (1) present the findings of the full model, (2) assess the robustness of our findings by exploring whether and how they change depending on app type and app category, and (3) conclude with a discussion of the interplay of the effects of updates, their timing and feedback intensity for each update category.

7.1. Surprise, non-compliance, and compliance effects

We find that offering an update of a certain kind regularly (i.e., in all versions, $VSLU^{j}_{i,v} = 0$) when users do not explicitly request these changes (i.e., $FI^{j}_{i,v-1} = 0$) fails to *surprise* the users, as we do not observe any strong effects for *update dummies* (−0.025, n.s. for bug fix, −0.008, n.s. for compatibility, −0.061, n.s. for add new content, and −0.021, n.s. for improve existing content).

We observe that all *feedback intensity* variables have negative coefficient estimates, two of which are statistically significant (−0.515, $p < 0.01$ for bug fix, −0.196, n.s. for compatibility, −0.379, $p < 0.05$ for add content, −0.259, n.s. for improve content). This result evidences our custom feedback dictionaries' ability to detect requests and complaints expressed in user reviews. It also confirms our prediction that unmet user requests (i.e., $FI^{j}_{i,v-1} > 0$ and $U^{j}_{i,v} = 0$) may drive down the ratings of the next version (i.e., $VSLU^{j}_{i,v} = 0$). For example, holding all else constant, if 25 % of the reviews contain language asking the developer to *add new content* to the app and the developer fails to respond immediately in the next version, the app's average rating score will decline by 0.09 points (the decline is 0.13 points for bug fixes). Considering that the magnitude of the absolute change in ratings between versions is 0.57 on average, developer *non-compliance* effects are rather substantial.

The signs of the parameter estimates for the *update dummy* and *feedback intensity* interaction terms and their significance levels suggest that developer *compliance* pays off, but only occasionally. Responding immediately (i.e., $VSLU^{j}_{i,v} = 0$) to the mentions of bugs and requests to add new content offsets the deleterious impacts of such feedback on ratings partially or completely (0.337, $p < 0.10$ and 0.511, $p < 0.05$, respectively). Though directionally the same, the impact of an immediate response to requests to improve the app's existing content is statistically insignificant (0.270, n.s.). Interestingly, the findings

[13] For instance, if two consecutive versions of the app contain an update of the same nature, say "add content", then $VSLU^{2}_{i,v} = 0$. If the developer adds new content in one version, then updates the app to fix a bug, then updates it again to improve compatibility, and then releases another version with new content, $VSLU^{2}_{i,v} = 2$.

[14] Hausmann test favors fixed app effects over random app effects.

Table 5
The Effects of Updates, Feedback Intensity, and Update Timing on Ratings.

	Updates only	Updates & Feedback	Full Model
Variable	**Coefficient**	**Coefficient**	**Coefficient**
Constant	3.687***	3.871***	3.885***
	(0.081)	(0.095)	(0.098)
Average Rating (Lag)	0.019	0.002	−0.013
	(0.020)	(0.021)	(0.021)
Update Category: Bug Fix			
Update	0.023	−0.023	−0.025
	0.032)	(0.039)	(0.043)
Intensity		−0.502***	−0.515***
		(0.151)	(0.166)
Update × Intensity		0.342**	0.337*
		(0.163)	(0.182)
Versions Since Last Update			−0.016
			(0.022)
Update × Versions Since Last Update			0.013
			(0.031)
Intensity × Versions Since Last Update			−0.062
			(0.096)
Update × Intensity × Versions Since Last Update			−0.023
			(0.130)
Update Category: OS/Device Compatibility			
Update	−0.115***	−0.019	−0.008
	(0.039)	(0.048)	(0.067)
Intensity		−0.307**	−0.196
		(0.131)	(0.168)
Update × Intensity		−0.851***	−0.747**
		(0.252)	(0.317)
Versions Since Last Update			0.012*
			(0.007)
Update × Versions Since Last Update			−0.011
			(0.016)
Intensity × Versions Since Last Update			−0.029
			(0.042)
Update × Intensity × Versions Since Last Update			−0.003
			(0.058)
Update Category: Add Content/Feature			
Update	0.005	−0.022	−0.061
	(0.029)	(0.036)	(0.042)
Intensity		−0.194	−0.379**
		(0.158)	(0.179)
Update × Intensity		0.281	0.511**
		(0.195)	(0.229)
Versions Since Last Update			−0.035*
			(0.018)
Update × Versions Since Last Update			0.048*
			(0.028)
Intensity × Versions Since Last Update			0.261**
			(0.104)
Update × Intensity × Versions Since Last Update			−0.313*
			(0.181)
Update Category: Improve Content/Feature			
Update	−0.064**	−0.080**	−0.021
	(0.039)	(0.035)	(0.040)
Intensity		−0.025	−0.259
		(0.183)	(0.218)
Update × Intensity		0.217	0.270
		(0.267)	(0.315)
Versions Since Last Update			0.046***
			(0.012)
Update × Versions Since Last Update			−0.068***
			(0.020)
Intensity × Versions Since Last Update			0.274**
			(0.119)
Update × Intensity × Versions Since Last Update			−0.122
			(0.206)
N apps	460	460	460
N observations	2795	2795	2795
R-squared overall	0.0754	0.1036	0.0594

Notes. * p < 0.10, ** p < 0.05, *** p < 0.01. Standard errors are shown in parentheses.

indicate that the negative, yet insignificant, effect of user complaints about the app's compatibility with new operating systems and devices may be exacerbated when addressed immediately (-0.747, $p < 0.05$).

As to how surprise, non-compliance, and compliance effects vary as a function of the *timing of the updates* (i.e., versions), some of the noteworthy findings are as follows. First, we focus on the *surprise* effect. Though surprising users with new or improved content in every version does not increase the average rating score of an app (-0.061, n.s. and -0.021, n.s., respectively), our results suggest that failing to add new content lowers the rating in the long run (-0.035, $p < 0.10$), while deferring updates improving the existing content increases it (0.046, $p < 0.01$). The positive and significant interaction between an update that adds new content and the number of versions released since an update of the same nature (0.048, $p < 0.10$) suggests that the developer can potentially make up for the deleterious effect of not having offered new content for some time. However, the same interaction is negative for improve content category (-0.068, $p < 0.01$), indicating that the developer will be penalized for skipping versions between improvement updates.

Second, as to the *non-compliance* effect, the positive and significant coefficient estimates of feedback intensity and versions since last update interaction terms in the "add new content" and "improve existing content" categories (0.261, $p < 0.05$ and 0.274, $p < 0.05$, respectively) suggest that the deleterious effects of non-compliance in these areas diminish with delay (i.e., forgetting takes place). Though statistically insignificant for "bug fix" and "OS/Device compatibility" categories, the signs of the same interaction terms' coefficients are in the opposite direction, suggesting that delay may accentuate the potentially deleterious effects of non-compliance in these areas (i.e., frustration builds up).

Finally, regarding the *compliance* effect, we observe that the favorable effect of complying with users' requests to add new app content and/or features (0.511, $p < 0.05$) diminishes with delay (-0.313, $p < 0.10$), highlighting the importance of being prompt.

7.2. Robustness checks

To assess whether the same pattern of results holds across different app types and categories, we estimate the model separately for (1) free vs. paid apps and (2) games vs. non-games. The results are presented in Appendix D.

To assess whether the results differ across different business models for apps, we compare the coefficient estimates in the 2nd, 3rd, and 4th columns of Table D5. The comparison reveals that the pattern of findings discussed above mainly holds for free apps. For paid apps, we find that fixing bugs when users haven't yet started voicing any concerns; and responding, even promptly, to compatibility requests hurt ratings. The deleterious effect of bug updates is dampened when made in response to used feedback. Moreover, not having to improve any content over versions also impact ratings positively. These findings collectively suggest that developers of paid apps are better off offering a ready-for-the-market, high-quality app that works on all most up-to-date operating systems and a wide range of devices upfront. Launching a (minimally) viable paid app and improving it on the way based on user feedback appears to backfire in terms of ratings, except when fixing bugs.

To assess whether the results differ across different app categories, we compare the coefficient estimates in the 2nd, 3rd, and 4th columns of Table D6 and observe that the pattern of results discussed above resembles those of non-games. We note that bug-related feedback and bug fix updates are particularly important for game evaluations. Though users' complaints about bugs lower the game's average rating score, releasing a bug-free version of the game counteracts the deleterious impact of the feedback. Finally, it is worth noting the rather substantial and deleterious compatibility compliance effects in games. Similar to our observations pertaining to paid apps, if users have to bring up this issue before it is dealt with, the game's average rating score suffers.

7.3. The interplay among updates, feedback intensity, and update timing

The discussion above reveals that user feedback and updates in certain categories as well as their timing influence an app's average rating score; however, their combined effect is hard to judge. To understand the interplay between user feedback and updates, and to provide insights for app developers, we calculate the average marginal effect of providing an update on the average rating score of an app at different values of feedback intensity and versions since the last update, separately for each update category, using the parameter estimates of the full model. In our calculations, we restricted the ranges of feedback intensity (between 0 and 0.5 with increments of 0.1) and versions since last update (between 0 and 5 with increments of 1) wherein more than 90 % of our data lies. Fig. 4 displays the results. When feedback intensity is at 0, the average marginal effects represent the incremental gains due to surprise over the number of app versions released since the last update in that category (i.e., the lowest slice in the plots). For all non-zero values of feedback intensity, the average marginal effects represent the incremental gains due to developer compliance over versions since last update of the same kind (i.e., the remaining regions).

Fixing Bugs. Panel A in Fig. 4 shows the incremental gains from an update containing a bug fix at different levels of feedback intensity and versions since the release of a similar update. Inspection of the figure's lowest slice (i.e., when feedback intensity is zero) reveals that a developer cannot *surprise* the users by fixing bugs regardless of the number of versions since the last bug fix update. However, once a critical mass is achieved, a developer who *complies* promptly with these user complaints about bugs enjoys significantly higher ratings than a developer who ignores them.

Compatibility. Similar to bug fixes, a developer cannot *surprise* users with a compatibility update (see the lowest slice in Panel B, Fig. 4). Moreover, updating the app after users start talking about compatibility issues appears to be futile, and the developer is penalized regardless.

Improving Existing Content. Panel C in Fig. 4 shows the incremental gains from improving existing content as a function of feedback intensity and versions since the release of an improvement update. The figure reveals that developers cannot *surprise* the app's users by improving existing content in every version but will certainly be worse off if they skip versions. Moreover, having to offer such improvements at the behest of users fails to improve and may even hurt ratings if not addressed quickly.

Adding New Content. The incremental gains presented in Panel D, Fig. 4 suggest that, if user feedback contains new content requests, addressing them can be rewarding. Yet, the magnitude of *compliance* rewards change depending on the intensity of the feedback and the timing of the update. Similar to bug fixes, the developers can reap off rewards only if a substantial portion of users requests new content, and that the response to these requests are timely. A delayed response fails to improve version ratings.

8. Conclusion

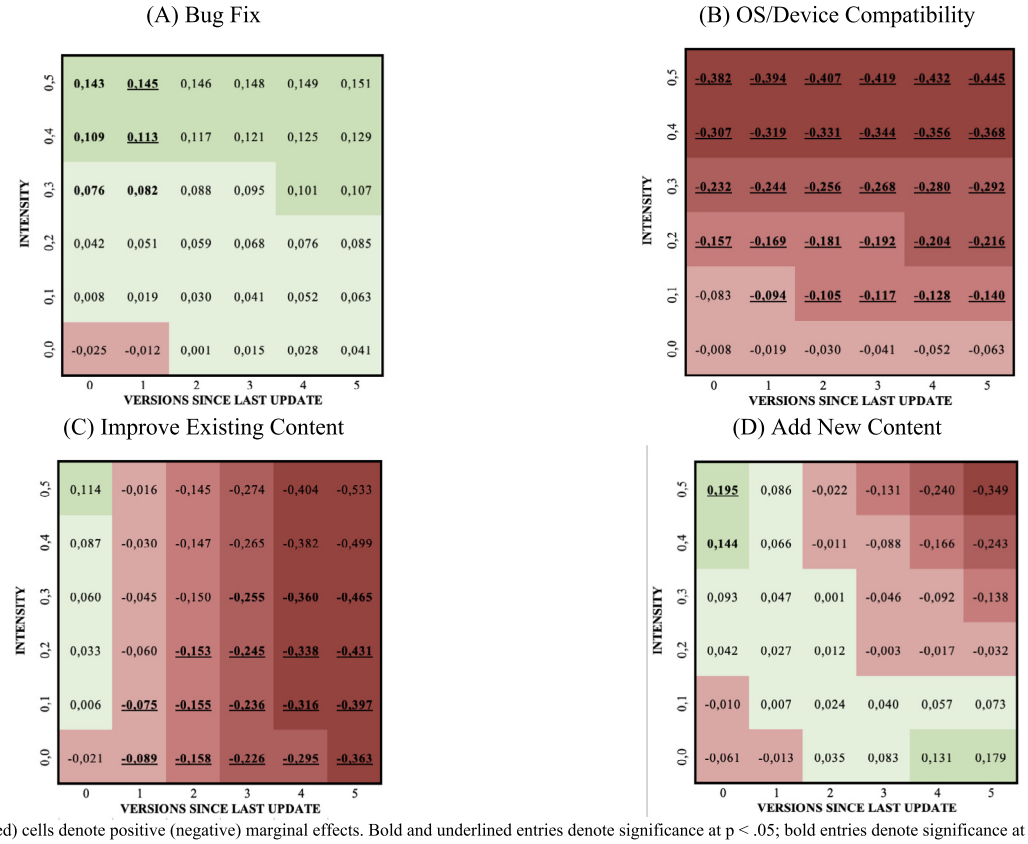
Based on an automated text analysis of 1,075,704 user reviews and content analysis of 3255 updates released by brand new apps from different app categories and business models, observed in their first year on the market, this paper assesses the impact of the alignment between the content of app updates and the feedback in user reviews on users' evaluations of an app's new version. Specifically, we quantify the rewards (or penalties) developers experience, determined by their attentiveness (or disregard) to user feedback in deciding the specifics and timing of the update. To provide fine-grained insights, we distinguish among four broad topics of feedback, corresponding to four areas of improvement identified by the content analysis of update descriptions. Our results indicate that the rewards and penalties associated with actively responding to user feedback or failing to do so are sensitive to the nature of the feedback and the improvements as well as the timing of the update.

8.1. Contributions

This research makes significant headway in advancing our understanding of mobile app success by examining the intricate interplay between user reviews and updates. Our study reveals that the effectiveness of updates varies significantly depending on their nature and timing. Moreover, unlike the majority of prior studies that analyzed user reviews and updates separately, our research is among few that integrates them, underscoring the need for developers to consider both factors when determining update specifics, and researchers when examining their effectiveness. Consequently, it addresses a key gap in the existing literature by delving into the nuanced interactions between users and developers, shedding light on how these exchanges impact app performance. Exploration of the intricacies of these interactions and their performance implications affords actionable insights for app developers striving to optimize user satisfaction.[15]

In addition to these contributions, our research introduces a fresh perspective by advocating for a more customer-centric, decision-driven analytics mindset in the context of mining customer reviews to inform product development decisions. We demonstrate that insights obtained from social listening – a less costly alternative to direct engagement with users – can guide firms in their product re-development decisions. Specifically, we show how firms can extract actionable user feedback from reviews through simple, custom-built dictionaries and proactively utilize it. As such, it emphasizes the importance of extracting value from insights, rather than merely extracting insights themselves. This approach differs from the primarily product-centric and data-driven decision-making mindset common in extant work, offering a potentially more forward-looking way of incorporating customer feedback into the development of successive product generations. By addressing some of the shortcomings in previous studies, particularly in evaluating the effectiveness of implementing different development recommendations mined from customer reviews, this research aims to enhance the practical applicability of its findings for managers. Altogether, this study represents a comprehensive and multifaceted examination of the interplay between customer reviews, product redevelopment, and performance, offering valuable insights for academia and industry practitioners alike.

[15] In one of our interviews with the developers, we had the chance to discuss the findings of our study and its practical implications. The developer who manages the finance app thought that the results were very beneficial in understanding user requests at a holistic level. The results and implications pertaining to bug fixes, compatibility, and improving existing content and/or features resonated with them. The potential to surprise users with new content and/or features somewhat puzzled them. When probed, it became clear that the developer had a narrower perspective on adding new content and/or features: they recalled the resistance they faced to user interface changes that users did not explicitly ask. Reminding the developer that "add content and/or features" category encompassed a diverse range of changes including changes made to the user interface solved the initial puzzlement. Yet, this discussion highlights an unavoidable limitation of our study and the potential for a fruitful avenue for future research.



Notes. Green (red) cells denote positive (negative) marginal effects. Bold and underlined entries denote significance at $p < .05$; bold entries denote significance at $p < .10$ levels.

Fig. 4. Incremental Gains in Ratings due to Updating as a Function of Feedback Intensity and Update Timing.

8.2. Implications for managers

By examining numerous scenarios along different levels of feedback intensity, and the existence and promptness of the response per update category, our study provides app developers guidance on the challenge of prioritizing possible development paths. Specifically, our results facilitate careful allocation of scarce resources between *compliance* and *continuous improvement* efforts and offer insights as to why.

The findings highlight the importance of adopting a *continuous improvement* approach *to maintain* apps' average rating scores on two fronts: (i) ensuring optimal performance on various operating systems and availability for a wide range of devices and (ii) improving the existing content of the app. However, the need for continuous improvement in these two areas is for different reasons. Being optimized for various operating systems and being available for a wide range of devices appears to be of crucial importance, as the deleterious effects of incompatibility are almost irreversible once users start complaining. Accordingly, the continuous improvement efforts are undertaken to prevent users from raising these issues in the first place. Continuously improving the existing content of the app, on the other hand, is critical because our results indicate that users come to expect these developments and penalize the developer for delivering these updates sporadically.

Compliance is necessary on the remaining two fronts, not just to maintain but *to improve* ratings. Our findings demonstrate that attending to issues about the essential functioning of the app (i.e., fixing bugs) and extending its value proposition guided by user feedback (i.e., delivering the new features and content requested) are beneficial for the developers. However, developers have a limited time window to comply, preferably in the update immediately following the chatter or the one after that. Failing to promptly address these concerns yields the same unsatisfactory result as not addressing them at all. Operating in this extremely crowded market with relatively low switching costs, developers may seek to prioritize compliance either to keep up with or to get ahead of their competitors.

The conclusion that developers cannot afford to neglect improving existing content or maintaining the app's compatibility in every single version raises the question: Can users ever be surprised? Unfortunately, our data fails to reveal such an effect. Attempts at fixing bugs that users do not reckon are unavailing, and introducing new app features or content does not produce a positive and statistically significant surprise effect. This finding is consequential.

Prioritizing compliance (i.e., responding to bug fix and new content requests) can deplete the resources for continuous improvement and cause delays in other updates (i.e., improving content and compatibility), resulting in lower ratings. A

favorable surprise effect, through bundling of updates, could have offset the deleterious effects stemming from the perceived inferiority of the app or insufficient coverage of operating systems and devices. However, since our data shows no support for a surprise effect, striking the right balance in allocating resources between *compliance* and *continuous improvement* becomes crucial for the developers.

Finally, although particularly important for the mobile app market, the implications of our findings may extend to domains where experimentation is natural, and firms may adopt a minimum viable product strategy. Examples of domains with similar dynamics include the development of webpages or software, the growing market of smart products that enable continuous development via updates, and (digital) services that allow for quick customization. Even in more traditional service settings, adhering to complying with user requests and maintaining a mindset of continuous improvement remain essential. In such settings, any direct customer interaction (e.g., face-to-face feedback in healthcare, retail, or hospitality or through call centers and company e-mails) can serve as valuable opportunities for gathering organic customer feedback. Depending on the type of the feedback received, firms can guide their responses along paths focused on either compliance or continuous improvement, with frontline employees making real-time decisions about which approach to follow. Furthermore, industry-specific dictionaries can be customized to capture the unique nuances relevant to each sector.

8.3. Limitations and avenues for future research

While our work paves the way for optimizing the implementation of organically generated feedback in the process of product development, it can be improved upon in several directions. First, a prevalent observation that surfaces from the analysis of user reviews is that users often discuss matters related to pricing and in-app purchases. Exploring how developers respond to these requests and the rewards or penalties they incur due to compliance or non-compliance present an interesting avenue for future research. However, response to such requests do not align with updates and, therefore, cannot be tracked through studying update descriptions. Moreover, app performance indicators other than average rating scores (e.g., downloads or revenues) might be more appropriate to assess the effectiveness of responses to price-related requests.

Second, this study treats the actionable feedback in all user reviews, if any, as being equally important. An interesting direction for future research can involve distinguishing between users and testing whether update decisions based on specific users' feedback produce more desirable results in terms of ratings. Related, in our interviews with the app developers, we observed efforts to categorize users based on their customer value and prioritize updates based on the feedback from different user strata. Given the potential for significant benefits, we believe there is merit in further developing this approach. Depending on the app's lifecycle stage, app developers may have different objectives, such as acquiring new users, retaining existing users, or increasing user engagement. Given a goal, exploring the effects of the (mis)match between different updates and goal-relevant feedback could provide useful insights. However, such an undertaking would require access to a much richer data set with app usage behaviors, including those of the review writers, and numerous app performance indicators.

Third, our study utilized a substantial dataset, covering 460 brand new ranked and unranked apps releasing 3255 new versions in their first year of existence, incorporating a total of 12,268 specific changes, which garnered feedback from 1,075,704 users. Despite its considerable size, our analysis could only focus on the broader categories of updates, such as improving existing content or features versus OS/Device compatibility. Therefore, the depth of our insights is constrained by the scope of these categories. A more extensive dataset would afford the opportunity to delve into finer details, examining the effects of specific update types, such as improving location sensitivity versus personalization features. Moreover, such an undertaking holds the potential to yield insights and implications at the app-category level, such as distinguishing between finance, social networking, and health and fitness apps. We defer this detailed exploration to future research endeavors.

Finally, future research can examine how our findings relate to the existing processes in continuous innovation settings in digital products such as smart products, software, and PC or console games. In a broader sense, our feedback and improvement categories are applicable not only to digital products but also to traditional ones. Future research, contingent on data availability, could explore whether our results hold true in diverse product and market contexts.

CRediT authorship contribution statement

Zeynep Aydin Gokgoz: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **M. Berk Ataman:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Formal analysis, Conceptualization. **Gerrit H. van Bruggen:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Conceptualization.

Data availability

The data that has been used is confidential.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ijresmar.2024.10.004.

References

Anadea (2017), "Why You Need an MVP for Your Mobile App," https://medium.com/swlh/why-you-need-an-mvp-for-your-mobile-app-4f8046ac52e0, last accessed on October 28, 2023.

Anderson, E. W., & Sullivan, M. W. (1993). The antecedents and consequences of customer satisfaction for firms. *Marketing Science, 12*(2), 125–143.

Anderson, M., & Magruder, J. (2012). Learning from the crowd: Regression discontinuity estimates of the effects of an online review database. *The Economic Journal, 122*(563), 957–989.

Archak, N., Ghose, A., & Ipeirotis, P. G. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management Science, 57*(8), 1485–1509.

Aydin Gokgoz, Z., Ataman, M. B., & van Bruggen, G. (2021). There's an app for That! understanding the drivers of mobile application downloads. *Journal of Business Research, 123*, 423–437.

Aydin Gokgoz, Z., Ataman, M. B., & Van Bruggen, G. (2022). The rise of mobile marketing: A decade of research in review. *Foundations and Trends® in Marketing, 17*(3), 140–226.

Bailetti, A. J., & Litva, P. F. (1995). Integrating customer requirements into product designs. *Journal of Product Innovation Management, 12*(1), 3–15.

Berger, J., Humphreys, A., Ludwig, S., Moe, W. W., Netzer, O., & Schweidel, D. A. (2019). Uniting the tribes: Using text for marketing insight. *Journal of Marketing, 84*(1), 1–25.

Blank, S. (2013). Why the lean start-up changes everything. *Harvard Business Review, 91*, 63–72.

Bornemann, T., Hattula, C., & Hattula, S. (2020). Successive product generations: Financial implications of industry release rhythm alignment. *Journal of the Academy of Marketing Science, 48*(6), 1174–1191.

Büschken, J., & Allenby, G. (2016). Sentence-based text analysis for customer reviews. *Marketing Science, 35*(6), 831–998.

Carrare, O. (2012). The impact of bestseller rank on demand: Evidence from the app market. *International Economic Review, 53*(3), 717–742.

Chang, W., & Taylor, S. A. (2016). The effectiveness of customer participation in new product development: A meta-analysis. *Journal of Marketing, 80*(1), 47–64.

Chen, L., Zhang, P., Li, S., & Turner, S. F. (2022). Growing pains: The effect of generational product innovation on mobile games performance. *Strategic Management Journal, 43*(4), 792–821.

Chen, R., Wang, Q., & Xu, W. (2019). Mining user requirements to facilitate mobile app quality upgrades with big data. *Electronic Commerce Research and Applications, 38* 100889.

Cheng, C. C., & Shiu, E. C. (2023). The relative values of big data analytics versus traditional marketing analytics to firm innovation: An empirical study. *Information & Management, 60*(7) 103839.

Chevalier, J. A., & Mayzlin, D. (2006). The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research, 43*(3), 345–354.

Chevalier, J. A., Dover, Y., & Mayzlin, D. (2018). Channels of impact: User reviews when quality is dynamic and managers respond. *Marketing Science, 37*(5), 688–709.

Choi, J., Oh, S., Yoon, J., Lee, J. M., & Coh, B. Y. (2020). Identification of time-evolving product opportunities via social media mining. *Technological Forecasting and Social Change, 156* 120045.

Ciurumelea, A., Schaufelbühl, A., Panichella, S., & Gall, H. C. (2017). Analyzing reviews and code of mobile apps for better release planning. In: 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 91-102.

Cooper, R. G. (1999). From experience: The invisible success factors in product innovation. *Journal of Product Innovation Management, 16*(2), 115–133.

Coviello, N. E., & Joseph, R. M. (2012). Creating major innovations with customers: Insights from small and young technology firms. *Journal of Marketing, 76*(6), 87–104.

Dąbrowski, J., Letier, E., Perini, A., & Susi, A. (2022). Analysing app reviews for software engineering: A systematic literature review. *Empirical Software Engineering, 27*(1).

Davidow, M. (2003). Organizational responses to customer complaints: What works and what doesn't. *Journal of Service Research, 5*(3), 225–250.

Del Vecchio, P., Mele, G., Passiante, G., & Serra, D. (2023). Knowledge generation from big data for new product development: A structured literature review. *Knowledge Management Research & Practice, 21*(4), 892–907.

Dellarocas, C., & Narayan, R. (2006). A statistical measure of a population's propensity to engage in post-purchase online word-of-mouth. *Statistical Science, 21*, 277–285.

Duan, W., Gu, B., & Whinston, A. B. (2011). The dynamics of online word-of-mouth and product sales: An empirical investigation of the movie industry. *Journal of Retailing, 84*(2), 233–242.

Dhar, V., & Chang, E. A. (2009). Does chatter matter? the impact of user-generated content on music sales. *Journal of Interactive Marketing, 23*(4), 300–307.

Forman, C., Ghose, A., & Wiesenfeld, B. (2008). Examining the relationship between reviews and sales: The Role of reviewer identity disclosure in electronic markets. *Information Systems Research, 19*(3), 243–396.

Franke, N., Keinz, P., & Steger, C. J. (2009). Testing the value of customization: When do customers really prefer products tailored to their preferences? *Journal of Marketing, 73*(5), 103–121.

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J. I., & Sadeh, N. M. (2013), Why People Hate Your App: Making Sense of User Feedback in a Mobile App Store, In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Ghose, A., & Han, S. P. (2014). Estimating demand for mobile applications in the new economy. *Management Science, 60*(6), 1470–1488.

Goldberg, D. M., & Abrahams, A. S. (2022). Sourcing product innovation intelligence from online reviews. *Decision Support Systems, 157* 113751.

Griffin, A., & Hauser, J. R. (1993). The voice of the customer. *Marketing Science, 12*(1), 1–27.

Hao, L., Li, X., Tan, Y., & Xu, J. (2011). The economic value of ratings in app market. *SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.1892584.

Henard, D. H., & Szymanski, D. M. (2001). Why some new products are more successful than others. *Journal of Marketing Research, 38*(3), 362–375.

Hennig-Thurau, T., Gwinner, K. P., Walsh, G., & Gremler, D. D. (2004). Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the internet? *Journal of Interactive Marketing, 18*(1), 38–52.

Humphreys, A., & Wang, R. J. H. (2018). Automated text analysis for consumer research. *Journal of Consumer Research, 44*(6), 1274–1306.

Istanbulluoglu, D. (2017). Complaint handling on social media: the impact of multiple response times on consumer satisfaction. *Computers in Human Behavior, 74*, 72–82.

Jang, S., & Chung, J. (2015). How do interaction activities among customers and between customers and firms influence market performance and continuous product innovation? An empirical investigation of the mobile application market. *Journal of Product Innovation Management, 32*(2), 183–191.

Kübler, R., Pauwels, K., Yildirim, G., & Fandrich, T. (2018). App popularity: Where in the world are consumers most sensitive to price and user ratings? *Journal of Marketing, 82*(5), 20–44.

Kenett, R. S., & Salini, S. (2011). Modern analysis of customer satisfaction surveys: Comparison of models and integrated analysis. *Applied Stochastic Models in Business and Industry, 27*(5), 465–475.

Lee, T. Y., & Bradlow, E. T. (2011). Automated marketing research using online customer reviews. *Journal of Marketing Research, 48*(5), 881–894.

Liu, Y. (2006). Word of mouth for movies: Its dynamics and impact on box office revenue. *Journal of Marketing, 70*(3), 74–89.

Luca, M. (2016), Reviews, Reputation, and Revenue: The Case of Yelp.com, *Harvard Business School NOM Unit Working Paper*, No. 12–016 doi:10.2139/ssrn.1928601.
Moore, R., & Moore, M. (2004). Customer inquiries and complaints: The impact of firm response time to email communications. *Marketing Management Journal, 14*(2), 1–12.
Netzer, O., Feldman, R., Goldenberg, J., & Fresko, M. (2012). Mine your own business: market-structure surveillance through text mining. *Marketing Science, 31*(3), 521–543.
Oliver, R. L. (1980). A cognitive model of the antecedents and consequences of satisfaction decisions. *Journal of Marketing Research, 17*(4), 460–469.
Packard, G., & Berger, J. (2017). How language shapes word of mouth's impact. *Journal of Marketing Research, 54*(4), 572–588.
Palomba, F., Linares-Vásquez, M., Bavota, G., Oliveto, R., Di Penta, M., Poshyvanyk, D., & De Lucia, A. (2018). Crowdsourcing user reviews to support the evolution of mobile apps. *Journal of Systems and Software, 137*, 143–162.
Park, J., Yang, D., & Kim, H. Y. (2023). Text mining-based four-step framework for smart speaker product improvement and sales planning. *Journal of Retailing and Consumer Services, 71* 103186.
Proserpio, D., & Zervas, G. (2017). Online reputation management: estimating the impact of management responses on consumer reviews. *Marketing Science, 36*(5), 645–665.
Qi, J., Zhang, Z., Jeon, S., & Zhou, Y. (2016). Mining customer requirements from online reviews: A product improvement perspective. *Information & Management, 53*(8), 951–963.
Rust, R. T., & Oliver, R. L. (2000). Should we delight the customer? *Journal of the Academy of Marketing Science, 28*(1), 86–94.
Shen, Y., Zhou, J., Pantelous, A. A., Liu, Y., & Zhang, Z. (2022). A voice of the customer real-time strategy: An integrated quality function deployment approach. *Computers & Industrial Engineering, 169* 108233.
Stocchi, L., Pourazad, N., Michaelidou, N., Tanusondjaja, A., & Harrigan, P. (2022). Marketing research on mobile apps: Past, present and future. *Journal of the Academy of Marketing Science*, 1–31.
Timoshenko, A., & Hauser, J. R. (2019). Identifying customer needs from user-generated content. *Marketing Science, 38*(1), 1–20.
Tirunillai, S., & Tellis, G. J. (2014). Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research, 51*(4), 463–479.
Vermeulen, I. E., & Seegers, D. (2009). Tried and tested: The impact of online hotel reviews on consumer consideration. *Tourism Management, 30*(1), 123–127.
Villarroel Ordenes, F., Grewal, D., Ludwig, S., De Ruyter, K., Mahr, D., & Wetzels, M. (2019). Cutting through content clutter: How speech and image acts drive consumer sharing of social media brand messages. *Journal of Consumer Research, 45*(5), 988–1012.
Wang, Y., & Chaudhry, A. (2018). When and how managers' responses to online reviews affect subsequent reviews. *Journal of Marketing Research, 55*(2), 163–177.
Ye, Q., Law, R., Gu, B., & Chen, W. (2011). The influence of user-generated content on traveler behavior: An empirical investigation on the effects of e-word-of-mouth to hotel online bookings. *Computers in Human Behavior, 27*(2), 634–639.
Yoo, K., & Gretzel, U. (2008). What motivates consumers to write online travel reviews? *Information Technology & Tourism, 10*(4), 283–295.
Zhou, Y., Su, Y., Chen, T., Huang, Z., Gall, H., & Panichella, S. (2021). User review-based change file localization for mobile applications. *IEEE Transactions on Software Engineering, 47*(12), 2755–2770.
Zhu, F., & Zhang, X. M. (2010). Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of Marketing, 74*(2), 133–148.
Pagano, D., & Maalej, W. (2013), User Feedback in the Appstore: An Empirical Study, In: *21st IEEE International Requirements Engineering Conference (RE)*, Rio de Janeiro, 125-134.