Do User Reviews Matter? Empirical Evidence on the Role of User Involvement in App Performance

Miriam Erne[1], Zhiying Jiang[2] & Vanessa Liu[2]

[1] Sole Proprietor, Cyprus, e-mail: miriam.erne@googlemail.com
[2] Singapore University of Social Sciences, School of Business, Singapore, Singapore, e-mail: jiangzhiying@suss.edu.sg, vanessaliusw@suss.edu.sg

Abstract The extant literature often presumed that user involvement was positively associated with software performance. In the context of mobile applications (apps), user reviews were collected to enlighten app developers on improvement of app quality through identifying bugs or suggesting new features. However, the value of user reviews varied a great deal due to their unmanageable volume and content irrelevance. In this study, over 40,000 user reviews with 50 apps were analyzed to empirically examine the association between customer led improvement and the revenues from the apps. Our findings indicated that customer led improvement produced significant increase in quarterly revenues. Greater growth in revenues was also observed if the developers responded to the user reviews faster. These results showed empirical support for the value of co-creation of apps with users, as customers could contribute to continuous improvement of the apps by providing experienced-based solutions.

Keywords: user involvement, customer led improvement, user reviews, mobile apps, empirical evidence.

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-362-3.11
ISBN 978-961-286-362-3

1 Introduction

Nowadays application distribution platforms such as Apple App Store and Google Play provide millions of different mobile applications (apps) to users. As of the fourth quarter of 2019, there were around 2.57 million apps for android users and 1.84 million apps for App Store users available (Statista, 2020). Survival in such a "hyper-competitive" mobile market was challenging to app developers (Comino et al., 2016). It is therefore becoming increasingly important for app developers to optimize app performance based on user needs (e.g., see Maalej et al., 2016; Maalej and Hadeer, 2015; Chen et al., 2014). One way to do so is through user involvement.

User involvement often takes the form of user reviews in mobile app development. Unlike regular reviews for products and services, user reviews for mobile apps have a more direct and influential impact over the life span of mobile apps. Poorly-rated or unpopular apps could be phased out very shortly after launch, resulting in a waste of development cost and effort. Most apps actively elicit customer comments as they are useful to the app developers, who might not always be able to spot a non-working feature. With so many different versions of smart phones and frequent software updates (e.g., iOS 10 and iOS 10.3.2), one app feature may work in one but not in another. Through spotting bugs, user reviews often offer valuable information to enable continuous improvement of the apps. Users could submit their feedback on their needs and experiences with an app like a missing feature or poor functionality (Khalid et al., 2015; Panichella et al., 2015). Complaints from users are actually of great value to further improvement of the app quality as they direct developers to be more customer-focused (Barlow and Moeller, 1996).

However, the number of user reviews received could be immeasurable and unmanageable. For instance, online gurus like Facebook could generate as high as at least 2,000 user reviews per day (Chen et al., 2014). The aspects covered in the reviews could be highly diverse, ranging from the price of the apps to the frequency of advertisements. Manual processing and management of these reviews is simply impossible, costly and overwhelming. More importantly, not all feedback is useful. Almost 65% of app reviews were found to be noisy and irrelevant (Chen et al., 2014). Some suggestions might be solely emotional and commercially infeasible, throwing little light on what concrete corrections could be made.

Most prior researchers focused on the development of analytical tools for categorization of user reviews (e.g., Maalej et al., 2016; Maalej and Hadeer, 2015), seldom questioning the actual benefits of the ideas from the user on app development. It was presumed that user involvement (in form of user reviews) could always lead to better app performance. Our study therefore aims to address this gap. Though the notion of performance is multifaceted and could refer to various aspects such as success, effectiveness, usability, comprehensibility, and satisfaction etc., our study adopted a financial approach and focused on performance in terms of revenues yielded from apps.

Specifically, we categorized and analyzed over 40,000 user reviews associated with about 50 apps. We conceptualized user reviews with bug-fixing suggestions as "customer led improvement" and examined its impact on revenues of apps. We also took into consideration the time taken for app developers to respond to the user reviews and examined the moderating role of developers' responsiveness.

The remainder of this paper is structured as follows: first, we will explain the conceptual framework and the related past studies. The research methodology and the data analysis procedure will then be presented. Finally, the findings will be discussed and the theoretical and managerial implications will be drawn.

2 The Conceptual Model

The development of our research model was grounded on the user involvement literature. It sought to explain the effect of user involvement on app performance. We relabeled user involvement as "customer led improvement" to align with the focus of this study on bug-reporting user reviews. Developer responsiveness was included as a moderating variable. Our research model is presented in Figure 1.

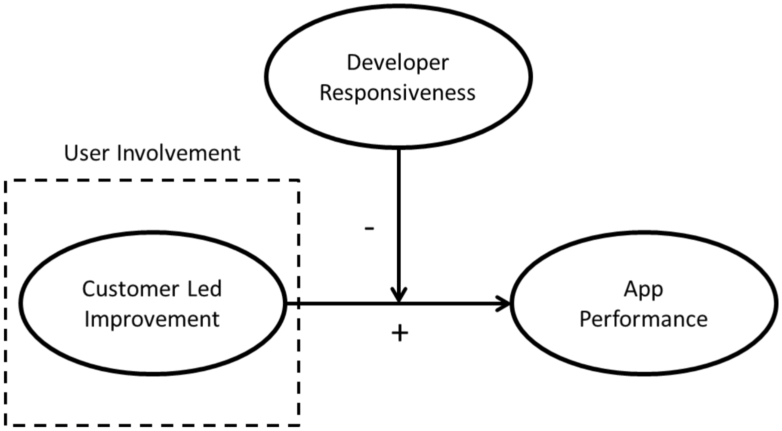

Figure 1: The Research Model

2.1 User Involvement and App Performance

The notion of user involvement was well documented in the literature, referring to the level of personal relevance and importance attached by users to the system (Barki and Hartwick, 1989). In broad terms, it is defined as "direct contact with users" (Kujala, 2003). User involvement may take different forms with varied levels and degrees. It can be informative, consultative or participative in nature (Kristensson et al., 2008). It is only helpful if certain involvement roles and development conditions are fulfilled (Ives and Olsen, 1984). These conditions include, who should be involved, which type of software with which the users should be involved, and in which stage (i.e., when) of the software development the users should be involved.

Recently, it was observed that customers had become more and more involved in the product development (Prahalad and Ramaswamy, 2013). User involvement was essential and indispensable for system/ software developers as it helped to collect more accurate user requirements and enable quality improvement, resulting in better fulfillment of user needs and higher user satisfaction (Kujala, 2008; Kaulio, 1998). User involvement was therefore recognized by previous researchers as beneficial to the improvement of quality and performance (Berger et al., 2005). Terms such as co-creation or co-design had emerged to describe the collaboration between developers and users. Other terms included quality function deployment (QFD), user-oriented product development, concept testing, Beta testing, consumer idealized design, lead user method and participatory ergonomics (Kaulio, 1998). In the collaborative process, users may assume the roles of providers of information, commentators or objects for observations.

User involvement could be totally undesirable when technical expertise is needed. While the potential value of user feedback is not deniable, it may not always be economically justified for developers to translate user feedback into actual software features (Ives and Olsen, 1984).

2.1.1 Users vs. Customers in Mobile App Development

In the context of traditional system design, users may only be engaged in user need elicitation or user acceptance test. Their involvement is minimal in other phases of system implementation. In the context of mobile apps, app users are often customers in nature. They go through similar purchase cycles like a customer. For example, a user may perform app search and app comparison in the initial stage, followed by order placement (for paid apps) or downloading and installation (for both paid and free apps). After-sale service may take the form of making inquiries at the helpdesk of the app developers. As the roles of users and customers have become blurred in the context of mobile apps, the terms "users" and "customers" are used interchangeably in this study.

2.1.2 Customer Led Improvement

User reviews, if carefully and properly screened and processed, could be vital to ongoing improvement of app performance. For example, a user might point out specific problems of how usage of an app led to slowing down of his/her iPhone. With many varieties of smartphones available, it was difficult for app developers to detect bugs specific to a particular phone model. Frequent software updates (e.g., iOS) rendered it even more complicated to test functionalities and compatibility of apps. User reviews could be a good source to identify usability issues. Though some users may be tech-non-savvy, the problems experienced by them might never be foreseen in the development process. Their feedback could still help developers to enhance user-friendliness of the apps. Appropriately addressing user reviews could be of strategic value to developers (Gutt et al., 2019). In this study, we focused on user reviews of the nature of bug-reporting. We conceptualized user reviews with suggestions on improvement as customer led improvement. It denotes reports from users about unwanted errors, bugs, annoying advertisements and other usability problems. Customer led improvement offer insights to developers to improve features and performance of apps, resulting in greater efficiency of development and higher user satisfaction (Kujala, 2008). Accordingly, we hypothesized that:

H1: Customer led improvement has a positive impact on app performance.

2.2 Developer Responsiveness to User Reviews

The time taken by developers to respond to user reviews on app improvement may matter (Vaniea and Rashidi, 2016). After a user submitted his/her feedback, he/she may tend to expect the developer to address the bug quickly. This is particularly important to individuals who are current users of the app. If the developer response is slow, the individual may continue to experience the bugs in the regular app usage and may eventually rescind usage or even uninstall the app. Conversely, users may tend to be more positive about the app if their concerns and problems were addressed promptly. The shorter the time taken to respond to user reviews, the greater the effect is the reviews on improvement of app performance. Accordingly, we hypothesized that:

H2: Developer responsiveness negatively moderates the relationship between customer led improvement and app performance.

3 Research Methodology

3.1 Research Context

The data was collected through a business intelligence company that retrieved panel data on a range of health and fitness apps, including the app user reviews and revenues generated from each app. Health and fitness apps were considered appropriate for our research focus as they tended to be used personally and users were likely to have more feedback on what improvement could be made. Another reason for the choice of these apps was that their target users were ordinary people. This should enable our research to be generalizable to other apps of general interest.

Only apps that had been active for at least one year were included in the sampling. Active apps should provide more valid results as it was common in the mobile apps industry that numerous apps could have been removed before their official launch. A total of 50 apps were selected for our analysis as their revenue constituted almost 75% of the total revenue in the health and fitness apps market. There were 189,527 user reviews available for these selected apps.

In order to measure the effect of user reviews on app performance, a specific research time frame was defined. Only reviews posted after the second last updated version and before the latest version of the apps were included in our samples. This enabled us to examine whether the user reviews led to improvement in the resultant update of the apps. The final sample consisted of a total of 40,619 user reviews, representing 21.4% of the total reviews associated with the selected apps.

3.2 Measurement

3.2.1 Customer Led Improvement

User reviews were used as proxies for improvement suggestions provided by customers. A subtraction and categorization process were conducted to identify the reviews that specifically pertained to improvement suggestions. Many tools were developed to support the search, screening, and extraction of useful information from user reviews. A review of the current literature showed that different tools were built with different mining objectives. Examples included MARK (Mining and Analyzing Reviews by Keywords) (Vu et al., 2015), MARA (Mobile App Review Analyzer) (Iacob and Harrison, 2013), ALERTme (Guzman et al., 2017), and AR-Miner (App Review Miner) (Chen et al., 2014). These tools made use of techniques like natural language processing, topic modeling, clustering and machine learning algorithms to search, classify, extract, group and rank user reviews based on pre-defined keywords or categories. In our study, Python coding was used to perform the screening of user reviews.

The screening took two steps. First, *generic* reviews were subtracted to isolate the *specific* reviews (Chen et al., 2014). *Generic* reviews were noisy and irrelevant reviews that did not provide any information on ways of improvement Examples of such reviews were "*by far the best app on meditation!*" and "*I love this app and have done since the* *moment I started using it. Potentially helped me get through a period of anxiety…*". *Specific* reviews, on the other hand, were those that stated a specific actionable function, that is, a function that the app developer can fix or improve. A total of 7,654 specific reviews were identified.

Next, the specific reviews were categorized to shortlist the improvement-related reviews. Consistent with previous studies, reviews concerning bugs and too many advertisements (embedded in the apps) were considered customer led improvement reviews (Maalej et al., 2016; Maalej and Hadeer, 2015).

A bug review reported on an unwanted error in an app. It could be any kind of problems with the app, a crash, an error or a performance issue arising from programming failure by the developer (Maalej and Hadeer, 2015). Examples of such review were "*it's not letting me sign up and I deleted the app and re-downloaded it but it's not working*" and "*if you open the app in the watch it tries to connect for a minute (literally a minute) then crashes*". Keywords used to screen for bug reviews were "bug", "fix", "problem", "issue", "defect", "crash", and "solve" (Maalej and Hadeer, 2015).

A review complaining about too many advertisements indicated that ads popped up too frequently and caused user annoyance. Reducing the number of ads might be room for improvement for the developer. Examples of these reviews were "*paid for the ap. Still get ads pushed to me. Don't advertise to me if I paid the money for the non-ad version*" and "*The avalanche of ads makes it unusable unless you pay $3 each and every month*".

3.2.2 Developer Responsiveness

Developer responsiveness was measured by the time interval (number of days) from the first posted date of the user review to the update date when the bug was fixed or the advertisements were removed.

3.2.3 App Performance

App performance could be operationalized in a number of ways such as the number of downloads and app ratings etc. In this study, app performance was measured using the revenue generated from the app during the research time frame. This allowed us to examine the financial impact on the app developers more directly.

Revenues could include purchases of apps, micro-transactions within an app or in-app advertisement (IADV) (Ghose and Han, 2014). The revenues for each app was computed by a summation of the daily revenues for the research time frame.

3.3 Data Analysis

Regression Analysis was conducted to analyze the correlations in the measurement model. It enabled us to examine the significance and the magnitude of the impact from the independent variable (customer led improvement) on the dependent variable (performance in terms of revenues) (Mooi and Sarstedt, 2011).

As the data for customer led improvement took the form of written user reviews, steps were taken to convert the text data into numerical data. Each review was enumerated with a Python code respectively according to its match with the categories of "bug" or "too many advertisements".

4 Results and Discussion

The results of the regression analysis were presented in table 1 below.

Table 1: Results

	Unstand. Coefficients	**Stand. Coefficients**			
	B	**Std. Error**	**Beta**	**t**	**Sig**
Model 1 – Customer Led Improvement	7292.185	2456.607	.523	2.968	.005
Model 2 – Customer Led Improvement	29,805.931	12,470.637	2.139	2.390	.021
Moderator of Customer Led Improvement	-612.97	316.019	-1.736	-1.94	.058

The overall model was significant with a p-value of 0.007. H1 was supported (0.005), meaning that customer led improvement has a positive significant impact on app performance in terms of revenues. The overall R-square was 0.190, which was satisfactory and typical for exploratory research (Mooi and Sarstedt, 2011).

The moderating effect of developer responsiveness was only supported with an alpha of .1 (.058). It was negatively associated with the link between customer led improvement and app revenues. With shortened response time, the impact of customer led improvement might increase the revenues from $7,292.19 to $29,805.93, demonstrating the negative moderating effect of developer responsiveness.

5 Implications and Directions for Future Research

Our findings provided empirical evidence on the value of user reviews on bugs and other usability issues. It was challenging for app developers to identify all possible bugs in view of the jungle of different smartphone models and rapid software updates. It was more cost-effective to adopt the approach of management by exception and rely on user reviews to report on problems and errors. As customers acquired hands on experience with usage of the app, they were more able to suggestion solutions. In other words, co-creation of apps with users should be encouraged to enable continuous improvement of the app performance (Gustafsson et al., 2012). It should lead to strategic value to the developers in the long run (Gutt et al., 2019). It would be worthwhile for app developers to invest in management of user reviews.

We also tested the moderating effect of developer responsiveness on the relationship between customer led improvement and app performance. Though significant, the effect was not very strong. One plausible explanation is that developers may have to launch app updates very frequently after addressing each bug or improvement suggested by users. The recurring need to update the app may be annoying to users (Vaniea and Rashidi, 2016) and discourage them from the continued usage of the app. However, existing users might also grow impatient if bug fixing took excessive time and the non-workable features constantly caused disruptions. App developers should therefore ensure user reviews on improvement were properly addressed within a reasonable time frame (Schenchk, 2013; Armerding, 2012).

In future research, this study could be tested using other app performance measures, such as the number of downloads, user ratings, or app rankings. We only examined user reviews with bug-reporting. Other researchers could investigate the impact of user reviews with a different nature, such as those focusing on making innovative suggestions to the developers.

References

Armerding, T. (2012). Why users don't often upgrade software when they should. Available at https://www.csoonline.com/article/2132061/security-awareness/References 48 why-users-don-t-often-upgrade-software-when-they-should.html.

Barki, H., & Hartwick, J. (1989). Rethinking the concept of user involvement. MIS Quarterly, 53–63.

Barlow, J., & Møller, C. (1996). A complaint is a gift: using customer feedback as a strategic tool. Berrett-Koehler Publishers.

Berger, C., Möslein, K., Piller, F., & Reichwald, R. (2005). Co-designing modes of cooperation at the customer interface: learning from exploratory research. European Management Review, 2(1), 70-87.

Chen, N., Lin, J., Hoi, S., Xiao, X., & Zhang, B. (2014). AR-miner: mining informative reviews for developers from mobile app marketplace. Proceedings of the 36th International Conference on Software Engineering, 767–778.

Comino, S., Manenti, F. M., & Mariuzzo, F. (2016). Updates management in mobile applications. iTunes vs google play.

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

Gustafsson, A., Kristensson, P., & Witell, L. (2012). Customer co-creation in service innovation: a matter of communication? Journal of Service Management, 23(3), 311–327.

Gutt, D., Neumann, J., Jabr, W., & Kundisch, D. (2019). The App Updating Conundrum: Implications of Platform's Rating Resetting on Developers' Behavior. In Proceedings of the International Conference on Information Systems.

Guzman, E., Ibrahim, M., & Glinz, M. (2017). A little bird told me: Mining tweets for requirements and software evolution. In Requirements engineering conference (re), 2017 IEEE 25th International (pp. 11–20).

Iacob, C., & Harrison, R. (2013). Retrieving and analyzing mobile apps feature requests from online reviews. In Mining software repositories (MSR), 2013 10th IEEE Working Conference on (pp. 41–44).

Ives, B., & Olson, M. H. (1984). User involvement and mis success: A review of research. Management Science, 30(5), 586–603.

Kaulio, M.A. (1998). Customer, consumer and user involvement in product development: A framework and a review of selected methods. Total Quality Management, 9(1), 141–149.

Khalid, M., Asif, M., & Shehzaib, U. (2015). Towards improving the quality of mobile app reviews. International Journal of Information Technology and Computer Science (IJITCS), 7(10), 35.

Kristensson, P., Matthing, J., & Johanson, N. (2008). Key strategies for the successful involvement of customers in the co-creation of new technology-based services. International Journal of Service Industry Management, 474-491.

Kujala, S. (2003). User involvement: a review of the benefits and challenges. Behavior & Information Technology, 22(1), 1–16.

Kujala, S. (2008). Effective user involvement in product development by improving the analysis of user needs. Behavior & Information Technology, 27(6), 457–473.

Maalej, W., & Hadeer, N. (2015). Bug report, feature request, or simply praise? on automatically classifying app reviews. In Requirements Engineering Conference (RE), 2015 IEEE 23rd International.

Maalej, W., Kurtanović, Z., Nabil, H., & Stanik, C. (2016). On the automatic classification of app reviews. Requirements Engineering, 21(3), 311–331.

Mooi, E., & Sarstedt, M. (2011). A Concise Guide to Market Research. Springer Berlin Heidelberg.

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., & Gall, H. C. (2015). How can I improve my app? Classifying user reviews for software maintenance and evolution. In Software maintenance and evolution (ICSME), 2015 IEEE International Conference on (pp. 281–290).

Prahalad, C. K., & Ramaswamy, V. (2013). The future of competition: Co-creating unique value with customers. Harvard Business Press.

Schenck, Stephen. (2013). Why I don't update apps. Available at: http://pocketnow.com/2013/09/25/dont-update-apps.

Statista (2019). Number of apps available in leading app stores as of 4th quarter 2019. Available at: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

Vaniea, K., & Rashidi, Y. (2016). Tales of software updates: software. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 3215–3226).

Vu, P. M., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015). Mining user opinions in mobile app reviews: A keyword-based approach (t). In Automated software engineering (ASE), 2015 30th IEEE/ACM International Conference on (pp. 749–759).