

The Role of App Update and User Feedback

Pramod K. Sharma*, Charu Shree, Devendra K., Daksh

Abstract:

Developing a mobile app that users will love can be a challenging task even with a great idea and a talented development team. It's impossible to anticipate every user's need or preference. That's where user feedback comes in. Collecting feedback from app users is essential for improving app's functionality, user experience, and overall success. It provides us with valuable insights into what users like, what they don't like, and what they want to see improved. In turn, this information allows us to make data-driven decisions about our app's development. Our work explores the role of user reviews in app updates based on release notes. For this purpose, we collected user reviews and released notes for Spotify, the number one app in the music category in Apple App Store, as the research data. Then, we manually removed non-informative parts of each release note and manually determined the relevance of the app reviews with respect to the release notes. Our empirical results show that more than 60% of the matched reviews are actually irrelevant to the corresponding release notes. When zooming in at these relevant user reviews, we found that around half of them were posted before the new release and referred to requests, suggestions, and complaints. Whereas the other half of the relevant user reviews were posted after updating the apps and concentrated more on bug reports and praise.

Keywords: Anticipate, data-driven decisions, release notes, non-informative, empirical, irrelevant, bug reports.

I. Introduction:

With the rapid progress on mobile techniques and smart phones, the number of mobile applications rises every year in which user feedback plays an important role. As if we are consistently listening to customers and making user-feedback-based decisions at every stage, we can ensure the specifications of our project are on target and we will be able to figure out our top priorities and thus make better business decisions. User-based feedback not only helps to determine our priorities but also helps us to design an improved final product. By collaborating with the user base, a wider range of ideas can be generated by which we can give more time and effort to ideas that get the most positive feedback and change (or scrap) ideas that are generating mostly negative feedback. If a certain tool is not working properly in the prototype, then we can make decisions accordingly on what customers need.

As the most common and widely used app data, user reviews and ratings have been exploited to obtain user requirements or app features for software evolution and maintenance. However, there is no clarity on user reviews that they are actually taken into account by developers and implemented in the new release.

There is found a smaller number of discoveries on the influence of user reviews on app updates based on app release notes from the point of view of developers. This study helps us to explore the role of user reviews in app updates according to app release notes. To our best knowledge, very limited research revealed the characteristics of the user feedback that got attention from developers. By some studies it was found that the app stores include information useful for app developers such as user requirements, ideas for improvement, user sentiments about specific features and descriptions of experiences with these features. We should use natural language processing techniques to identify and find grained app features in the reviews and become able to extract the user sentiments about the identified features and give them a general score across all reviews.

By user's feedback, the technical issues coming in any tool are demolished as the development team could spend considerable time on a single bug fix. Or if we talk about new features, it can be developed on demand of customers which saves time and helps to improve the overall experience. This way also stops the wastage of resources which is through by gathering quality feedback from users like as after spending countless months on developing a new feature if in the end it fails to meet the needs of the users, allocate resources to features will add value to customers which improves the overall experience. In this paper, we propose an automated approach that helps developers filter, aggregate, and analyze user reviews.

Corresponding Author: Pramod K. Sharma

Assistant Professor, Information Technology, Arya Institute of Engineering and Technology, Jaipur, Raj.

Associate Professor, Department of Computer Science, Arya Institute of Engineering and Technology, Jaipur, Raj.

Science Student, Ganesh Public Sr. Sec. Sch., Kotputli, Raj.

Science Student, RPS Public School, Narnaul, Haryana

II. Literature Reviews:

Building on previous research, we conducted two studies on android app updates from a user's perspective. The first study involved a survey on update behaviors and attitudes, while the second study was a field study testing different update interphases. In our online survey, we analyzed users' behavior patterns and evaluated the effectiveness of current notifications.

Based on our findings, we proposed a new design for app update notifications that alerts users when a potential privacy risk is identified or expected. Additionally, we included reviews of the update within the notification, as some users reported checking reviews explicitly before updating.

To validate our new design, we conducted an online between-group experiment using simulated updates of two popular apps. We discussed the design detail and analyzed the results of the two studies in the subsequent sections.

In the past few years, many researches have contributed to the extraction and analysis of app review for software maintenance and evolution. However, app reviews mainly concentrated on the extraction of app features, in order to get the key issues or topics for developers in updating apps. For example, both AR-Miner and Casper were proposed to extract user requirements from app reviews. On the other hand, existing studies, on release notes intend to explore the characteristics and development trends of app updates, by analyzing the posting time and content of release notes. For example, Mellroy et al. explore the update frequency of 10713 mobile apps across 30 mobile app categories which indicate that 14% of the apps are updated frequently, while 45% of these frequently-updated apps do not provide users with any information about the rationale for the new updates. He also observed that frequently-updated apps are highly ranked by users.

Other researchers tried to match user reviews with bug reports written by developers as Villarroel et al. proposed in approach called CLAP (Crowd Listener for Release Planning) to categorize user reviews, cluster together related reviews and prioritize the clusters of reviews to suggest app release planning.

In the study of Haring et al, he proposed in automatic approach deep matcher to match problem reports in app reviews to bug reports in issue trackers by using deep learning algorithms.

Unlike these authors' studies, our work is based on the release notes, trying to identify the most relevant reviews to determine the degree of relevance and response between release notes and user reviews.

III. Conclusion:

User feedback serves as the compass, guiding app updates, which, in turn, enhances the app's quality and user identification. App updates and user feedback are too important aspects of mobile app developments app updates are necessary to keep the up-to-date with the latest technologies and operating systems, fix bugs, and add new features, updating an app regularly is also a great way to keep users interested and engaged with the app User feedback is a valuable source of information for developers to improve their apps.

User feedback can help identify bugs, usability issues, and feature requests. However, it's important to note that not all feedback is useful or actionable. Developers need to be careful when interpreting feedback and prioritize the most critical issues additionally; acting on customer feedback can lead to increased customer satisfaction and loyalty.

User feedback is a valuable tool in mobile app development in which by involving users in the development process we can ensure that our app is meeting our needs and preferences. By user feedback, our goal is to identify and analyze the traits and distinctions of publicly available hybrid mobile apps from and users' perspective. Feedback helps developers and product managers to see what popular features are not in their content. It provides valuable information about their app's use and features needing improvement.

App updates enhance the security of one's device, which helps to protect our personal details. If some apps of our own device are outdated, hackers could make the usage of their outdated style. By updating system apps there are changes and patches that the apps developer had added on the app. Bug fixes and UI change are usually what the updates of the apps contain. We can encourage researchers to conduct further researchers in these areas to better understand the phenomena of social media marketing to benefit both academics practitioners.

IV. Future scope:

Experts predict that various applications will be developed using technologies such as machine learning algorithms and predictive analytics. In this way android app developers will provide a more enhanced personalized experience to users.

Apps will become more personalized based on user feedback. Developers will tailor updates to individual user preferences, making the user experience more relevant and engaging. App updates will continue to play a crucial role in delivering new features, fixing bugs, and enhancing security. As technology evolves, apps need to keep up with the latest trends and user expectations. The user expects exceptional, frictionless experiences and support when there is a need, throughout the user journey. They want just accept being able to self-serve answers, they will expect to be able to contact a business at a time convenient to them through a channel of their choice, accessible from any device, anywhere on the planet.

Fresh works vision of the future of user's feedback embodies what the modern consumer of today and tomorrow expects and serves as a guide to stay relevant and competitive.

Fresh works AI engine, to constantly learn and improve on the routing performance to minimize wait times. Freddy (an Omni channel routing engine, enables businesses with multi skilled support agents to automatically assigned user queries to agents with available bandwidth for faster, better service) knows when to intervene and when to pass on the query to a human agent while making sure that the transition is seamless and also gives full context about the user to the agent.

References:

1. Carreño, L.V.G., Winbladh, K.: Analysis of user comments: an approach for software requirements evolution. In: International Conference on Software Engineering, pp. 582–591. IEEE (2013)
2. Cerf, V.G.: Apps and the web. *Commune. ACM* 59(2), 7–7 (2016)
3. Chen, N., Lin, J., Hoi, S.C., Xiao, X., Zhang, B.: Ar-miner: mining informative reviews for developers from mobile app. marketplace. In: International Conference on Software Engineering, pp. 767–778. ACM (2014)
4. Cugola, G., Ghezzi, C., Pinto, L.S., Tamburrelli, and G.: Adaptive service-oriented mobile applications: A declarative approach. In: International Conference on Service-Oriented Computing, pp. 607–614. Springer (2012)
5. R. K. Kaushik Anjali and D. Sharma, "Analyzing the Effect of Partial Shading on Performance of Grid Connected Solar PV System", 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-4, 2018.
6. R. Kaushik, O. P. Mahela, P. K. Bhatt, B. Khan, S. Padmanaban and F. Blaabjerg, "A Hybrid Algorithm for Recognition of Power Quality Disturbances," in *IEEE Access*, vol. 8, pp. 229184-229200, 2020.
7. Kaushik, R. K. "Pragati. Analysis and Case Study of Power Transmission and Distribution." *J Adv Res Power Electro Power Sys* 7.2 (2020): 1-3.
8. Finkelstein, A., Harman, M., Jia, Y., Martin, W., Sarro, F., Zhang, Y.: App. store analysis: Mining app. stores for relationships between customer, business and technical characteristics. *Research Note of UCL Department of Computer Science* 14, 10 (2014)
9. Gorla, A., Tavecchia, I., Gross, F., Zeller, A.: Checking app. behavior against app. descriptions. In: International Conference on Software Engineering, pp. 1025–1035. ACM (2014)
10. Guzman, E., Maalej, W.: How do users like this feature? A fine-grained sentiment analysis of app. reviews. In: International Conference on Requirements Engineering, pp. 153–162. IEEE (2014)
11. Hao, Y., Wang, Z., Xu, X.: Empirical study on the interface and feature evolutions of mobile apps. In: International Conference on Service-Oriented Computing, pp. 657–665. Springer (2016)
12. Iacob, C., Harrison, and R.: Retrieving and analyzing mobile apps feature requests from online reviews. In: IEEE Working Conference on Mining Software Repositories, pp. 41–44. IEEE (2013)
13. Iacob, C., Harrison, R., Faily, S.: Online reviews as first-class artifacts in mobile app. development. In: International Conference on Mobile Computing, Applications, and Services, pp. 47–53. Springer (2013)
14. Khalid, H., Shihab, E., Nagappan, M., Hassan, A.E.: What do mobile app users complain about? *IEEE Soft.* 32(3), 70–77 (2015)
15. Maalej, W., Nabil, H.: Bug report, feature request, or simply praise? On automatically classifying app. reviews. In: IEEE International Conference on Requirements Engineering, pp. 116–125. IEEE (2015)
16. Martin, W., Sarro, F., Harman, M.: Causal impact analysis applied to app. releases in google play and windows phone store. *Research Note of UCL Department of Computer Science* 15, 07 (2015)
17. Martin, W., Sarro, F., Jia, Y., Zhang, Y., Harman, and M.: A survey of app. store analysis for software engineering. *IEEE Transactions on Software Engineering*, PP(99), 1–32 (2016)
18. McIlroy, S., Ali, N., Hassan, and A.E.: Fresh apps: an empirical study of frequently-updated mobile apps in the google play store. *Emp. Soft. Eng.* 21(3), 1346–1370 (2016)
19. Sharma, Richa and Kumar, Gireesh. "Availability Modeling of Cluster-Based System with Software Aging and Optional Rejuvenation Policy" *Cybernetics and Information Technologies*,
20. G. Kumar and R. Sharma, "Analysis of software reliability growth model under two types of fault and warranty cost," 2017 2nd International Conference on System Reliability and Safety (ICSRS), Milan, Italy, 2017, pp. 465-468, doi: 10.1109/ICSRS.2017.8272866.

21. Kumar, G., Kaushik, M. and Purohit, R. (2018) "Reliability analysis of software with three types of errors and imperfect debugging using Markov model," *International journal of computer applications in technology*, 58(3), p. 241. doi: 10.1504/ijcat.2018.095763.
22. Gireesh, K., Manju, K. and Preeti (2016) "Maintenance policies for improving the availability of a software-hardware system," in 2016 11th International Conference on Reliability, Maintainability and Safety (ICRMS). IEEE.
23. Sharma, R. and Kumar, G. (2017) "Availability improvement for the successive K-out-of-N machining system using standby with multiple working vacations," *International journal of reliability and safety*, 11(3/4), p. 256. doi: 10.1504/ijrs.2017.089710.
24. Jain, M., Kaushik, M. and Kumar, G. (2015) "Reliability analysis for embedded system with two types of faults and common cause failure using Markov process," in *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*. New York, NY, USA: ACM.