Developers' Eyes on the Changes of Apps: An Exploratory Study on App Changelogs

Chong Wang, Ju Li, Peng Liang
School of Computer Science
Wuhan University
Wuhan, China
{cwang, liangp}@whu.edu.cn

Maya Daneva, Marten van Sinderen
Service and Cyber Security Group
University of Twente
Enschede, the Netherlands
{m.daneva, m.j.vansinderen}@utwente.nl

***Abstract*— Release planning for mobile apps has only recently become an area of active research. As a result, little is known about the types of requirements that app developers pay the most attention to when releasing an app. This research uses the changelogs of apps to shed light on this. We report the results of an exploratory study in which we analyzed the requirements that dominate the changes of apps, according to a set of 3000 changelogs collected from 120 apps from three categories in the Apple App Store: Travel, Social networking, and Books. We analyzed the changelogs in terms of functional and non-functional requirements, from a developers' perspective. Our results suggest that developers' releases are by far more concerned with non-functional requirements than with functional requirements. We also found that usability and maintainability are the most frequently mentioned non-functional requirements (NFRs) in the changelogs. Surprisingly, reliability requirements formed only a fraction of the total number of NFRs addressed in all changelogs of apps in the three selected App Store categories.**

***Index Terms*—Requirements Engineering, Non-functional Requirements, Release Planning, Changelogs, App Store, Empirical study**.

I. Introduction

With the progress on mobile techniques and smart phones, the number of mobile applications (apps for short) are growing much faster in recent years. The first quarter of 2019 marked the availability of 2.1 million apps for Android users to choose, whereas in Google Play, 1.8 million apps are there to download [1]. Meanwhile, the number of either new apps or new releases of existing apps are even continuously growing, to satisfy the emerging demands of users and to be the winner in the market competition. Since app repositories offer such a huge number of apps, it is difficult to understand what changes in the apps make them get more downloads in the market of app users. For this purpose, this exploratory study intends to employ app changelogs as the data source to get a comprehensive understanding on the trend of apps development and developers' concerns on release planning of apps.

App changelogs are posted by software vendors or developers regularly in weeks or months. These official texts are written in a standardized way and comprise the primary changes of the releases of each app. Moreover, app changelogs have been employed as one of the data sources for the analysis on app stores. For example, a 2018 ICSE study [2] has successfully employed app changelogs to identify emerging issues in app reviews. Plus, Hassan et al. [3] conducted an empirical study on emergency updates for top Android mobile apps. Specifically, in [3], the content of app changelogs (i.e., release note in [3]) was only analyzed to confirm whether they provided useful information about the rationale for the emerging updates. Unlike the study of Hassan et al. [3], our exploratory research employed the app changelogs as the data source to investigate the changes of apps as related to requirement types, from a developers' perspective.

The rest of this paper is organized as follows. Section II provides background and related work. Section III presents the research questions and describes the process of data collection and labelling. Section IV presents the descriptive results of our exploration. Section V discusses the study results, followed by the limitations in Section VI. Finally, Section VII concludes and gives direction to further research.

II. Background and Related Work

For the purpose of this research, we use the term mobile app (or just an app) to refer to applications designed specifically for the current generation of mobile devices such as smart phones and tablets [4]. Typically, mobile apps are distributed through a platform specific, and centralized app market [13]. Release engineering aspects of these apps have only recently become an area of active research, unlike release planning and engineering for web and desktop applications, which has been an established area for many years [5]. From a Requirements Engineering (RE) perspective, the majority of empirical studies on apps took users' perspective, e.g. many studies focused on app users' reviews and their importance for requirements elicitation and software evolution.

There are only a few empirical studies that focused on the analysis of release notes from developers' perspective. The survey research of Nayebi et al. [5] focused on uncovering the ways in which mobile app developers organize their releases and the release strategies they employ. These authors found that half of the developers participating in the survey had a clear strategy for their app releases. Furthermore, the empirical analysis of McIlroy et al. [6] looked into the update frequency of the top 10,713 mobile apps across 30 mobile app categories. These authors indicated that 14% of the apps are updated frequently, while 45% of these frequently-updated apps do not provide the users with

978-1-7281-5165-6/19/$31.00 ©2019 IEEE
DOI 10.1109/REW.2019.00042

any information about the rationale for the new updates. McIlroy et al. also observed that frequently-updated apps are highly ranked by users. Next, Hassan et al. [3] analyzed 1000 emergency updates of apps in the Google Play Store to identify patterns of updates and their effect on the user experience. The authors conclude that most emergency updates are due to simple mistakes and that developers should avoid these patterns if they want to improve the user experience. Finally, a 2018 empirical study of Nayebi et al. [7] investigated how developers consider deletion and addition of functionality to apps. The authors grounded their research on Lehman's laws of software evolution which states that the functionality of programs has to increase over time to maintain user satisfaction. Their study however found that in the domain of mobile apps developers consider *"deletion of functionality to be equally or more important than the addition of new functionality"* [7].

III. Research Design

A. Research Questions

The main objective of our work is to explore the changes of apps from a developers' perspective, by analyzing app changelogs in terms of requirement types: Functional and Non-Functional Requirements (FRs and NFRs). To this end, we formulate two Research Questions (RQs):

RQ1: *Which type of requirements (FR vs. NFRs) dominate the changes of apps, according to app changelogs?*

RQ2: *Which types of NFRs are the foci of app updates, from a developers' perspective?*

In general, app changelogs describe the main and/or the most important changes of the newly released apps, compared to their previous releases. The answer to RQ1 is needed to understand that which type of software and user requirements (FR vs. NFR) get more attention when developers maintain or update the apps in order to catch the eyes of users. Then, RQ2 zooms in on the types of NFRs that developers prefer to list in the official changelogs of apps.

To answer these RQs, we set up an exploratory study [8]. Our exploration mainly focused on the mobile apps available in the Apple App Store, particularly the apps in three Apple-store-defined categories [13]: Books, Travel and Social Networking. This choice was justified with the authors' collective familiarity with these app types and experience in using them. We note that the belongingness of a specific app to a category is pre-determined by Apple [13], and not by the authors.

Furthermore, to get a more comprehensive understanding on the changes of apps, we included in our data collection and data analysis app markets in six regions, namely: Asia, North America, South America, Europe, Africa, and Oceania. These app markets were reviewed to collect changelogs of the top 10 free-to-download mobile apps in our three selected categories. Our data analysis draws on the content analysis technique of Krippendorff [9]. We chose this analytical approach because of its suitability to our research context and also because of its reliance on coding and categorizing of the data, which makes the qualitative analysis particularly rich [10]. Below, we describe our data collection process and our data analysis in more detail.

B. Data Collection and Preprocessing

The data collection was performed in January 2019, covering each of the six selected regions in the Apple App Store. Our raw dataset was formed as follows. We first excluded both duplicate apps and the apps with a very low number of releases. This resulted in 120 apps: 40 apps in the 'Social Networking' category, 50 apps in the 'Travel' category, and 30 apps in the 'Books' category. Next, for each of these 120 apps, we excluded the app changelogs written in non-English language. In total, we collected 17024 changes in 8647 app changelogs of these 120 apps. The data collection process and the selected changelogs are summarized in Table I (see the 2nd to 4th column).

When zooming in on the app changelogs, we observed that each app changelog consists of multiple app changes, and each app change is listed as one numbered sentence in app changelogs. However, we found that the changes in 'What's New' and 'Recent Updates' overlapped much in most releases of those 120 apps. To filter redundant app changes in the collection of app changelogs, each app changelog was first decomposed into changes. In our work, each app change is denoted by one numbered sentence in one app changelog. Next, the changes in 'Recent Updates' were removed if these changes were the same as the ones in 'What's New'. This resulted in 8325 app changes, by excluding 8639 duplicate changes. Furthermore, we observed that between January 2012 and December 2013 only 161 versions were released, which accounted for 1.86% of our raw dataset. As this percentage is too small, app changes in those 161 releases were also removed, in order to balance the dataset. As a result, we got 8037 app changes posted between January 2014 and December 2018.

Note that in this exploratory study, the decomposition of an app change was implemented manually, and the identification and removal of duplicate app changes was automatically conducted by Microsoft Excel.

TABLE I. Summary of selected app changelogs (2014-2018)

Category of apps	No. of apps	No. of app changelogs	No. of app changes	
			Before exclusion	*After exclusion*
Social Networking	40	3631	6469	2652
Travel	50	3166	6494	3207
Books	30	1850	4061	2178
Total:	120	8647	17024	8037

C. Data Labeling

To understand the updating trend of apps based on app changelogs, this paper employed manual labeling, analysis and synthesis [9] to identify and statistically analyze different types of requirements in the collected app changelogs. In general, manual labeling is a time-consuming task, and it is hard for the authors to manually label all the 8037 app changes in the short term. Therefore, for the purpose of this exploratory study, we took randomly 3000 ($1000\ \text{changes} \times 3\ categories$) out of

the 8037 included app changes. These 3000 changes formed the final set used in this study.

Following Krippendorff [9], we applied the 'a priori coding' process, in which a data classification schema is established prior to the analysis based upon some theory. As Stemler suggests [10], professional colleagues agree on the categories in a classification schema to use, and the coding is applied to the data. Revisions of the classification schema could be made as necessary, and the categories are tightened up to the point that maximizes mutual exclusivity and exhaustiveness [10]. In this research, we use a NFRs classification schema based on the ISO 25010 standard [11] for NFRs. This means the app changes that addressed NFRs were classified according to the NFRs types treated in the standard.

The manual content analysis and labeling for the sampled 3000 changes was performed by two bachelor students majoring in Computer Science, including the second author of this paper. First, we briefed these two coders in a meeting to introduce the task and explain the NFR standard (ISO 25010 [11]) using some examples for labeling NFRs with some examples. Then, these two coders were asked to conduct the first round of pilot labeling on 120 app changes ($40\ \text{changes} \times 3\ categoris$). This pilot labeling task was completed within 35 minutes (two coders worked in parallel and spent 30 and 35 minutes respectively), and resulted in 63% inter-coder agreements [12] on app changes. After discussing and resolving all the disagreements (the remaining 37%), we developed a coding guide to precisely define each type of requirements and increase the quality of manual labeling. Next, additional 500 app changes were selected randomly as the input of the second-round pilot labelling task. The agreement of the second-round labelling was 73.8%, so the two coders updated the coding guide after resolving all the disagreement in the second-round pilot labeling. Finally, these two bachelor students completed labeling and came to an agreement on all the included 3000 app changes.

While labelling, we found that most app changes specified only one type of requirements. However, there were still a few changes describing more than one requirements types. In order to facilitate the analysis and synthesis in app changelogs, those app changes containing more than one types of requirements were decomposed into several app change sentences. This was needed in order to (1) ensure that only one requirement type can be identified in each change, and (2) make the labelled app changes fit for training and testing the classifiers of supervised machine learning algorithms in the near future.

TABLE II. EXEMPLARY APP CHANGES FALLING IN THE TYPES OF REQUIREMENTS.

Type of requirements	Examples of app changes
Usability	*Read magazines in a more friendly interface format.*
Reliability	*Email notification works again.*
Maintainability	*Unlocking picture is fixed.*
Performance	*Establishing video session takes less time.*
Portability	*Support for 3D Touch on iPhone 6 and 6s.*
FR	*We have added a password function; you can set a password to enter the program.*
Others	*Thank you for your use, please continue to pay attention.*

Note that ISO 25010 treats eight types of NFRs: *Functional suitability*, *Performance efficiency*, *Compatibility*, *Usability*, *Reliability*, *Security*, *Maintainability*, and *Portability*. During the pilot labeling, however, we found that functional suitability, compatibility, and security were seldom observed in app changelogs. Therefore, we included only five types of NFRs (*Usability*, *Reliability*, *Maintainability*, *Performance*, and *Portability*) in this research, plus the type of functional requirements (FR) and a requirements type that we labeled 'Others' to refer to those requirements that are neither FR nor fit the five NFRs indicated above. Examples of requirements falling in each category are shown in Table II. We notice that the statements in the "Others" category are usually non-informative for RE as they are either thank-you notes or some very general recommendations, such as "keep the good work" or "please continue to pay attention" as indicated in the last row of Table II.

IV. RESULTS

A. Overview of App Changes

Figure 1 shows the overview of 8037 app changes collected from 2014 to 2018. We observed that the number of both app changelogs (see the blue line) and app changes (see the orange line) are continuously increasing over the last five years. Moreover, Figure 1 indicates that the increase of app changes grows faster than that of app changelogs. One reason could be that the app changelogs contain more and more changes that have been implemented in the new releases.

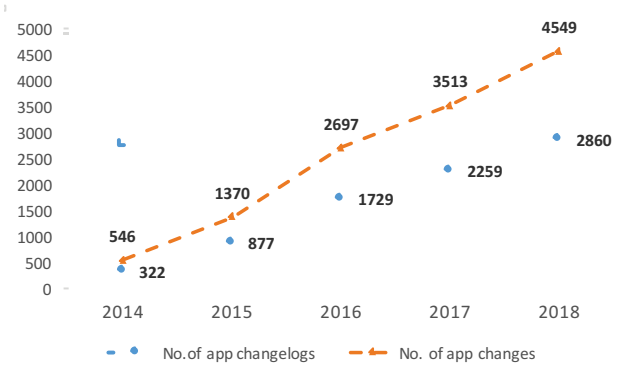
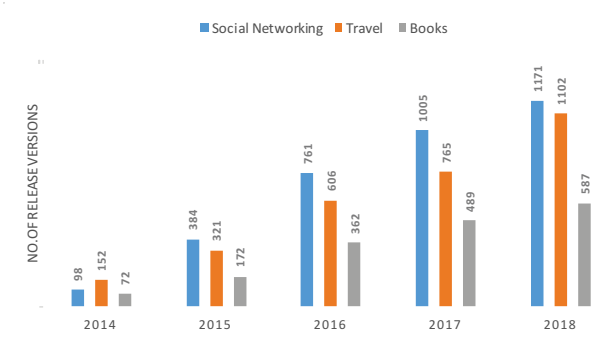


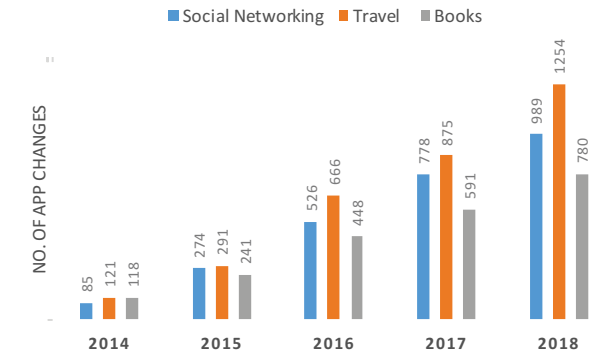
Fig. 1. Growth of app changelogs and app changes over year.

Figure 2 zooms in on the changes of apps from the perspectives of the three categories of these apps, i.e. Social Networking apps, Book apps, and Travel apps. More specifically, Figure 2 (a) shows that compared to apps in the 'Books' category, apps developed for both social networking and travelling were updated much more frequently. Plus, apps in the category of 'So-

cial Networking' usually have the largest number of new releases. Considering the number of app changes shown in Figure 2 (b), we found that apps in the 'Travel' category released much more updating details in less app changelogs, compared with the apps for social networking. Similarly, apps in the category of 'Books' have the least number of app changes.



(a) Numbers of app changelogs over year.



(b) Numbers of app changes over years.

Fig. 2. The changes of apps over different categories of apps.

B. Answer to RQ1

Table III shows the distribution of app changes over the types of requirements, i.e. FR or NFRs. We found that 92.5% of the selected 3000 app changes (2777/3000) are informative for RE (see the total number of FRs and NFRs versus the requirements in the "Others" category which are not uninformative). Specifically, 63.6% of 3000 app changes mentioned NFRs and 28.9% referred to FR.

TABLE III. REQUIREMENTS TYPES IDENTIFIED IN APP CHANGES OVER DIFFERENT CATEGORIES OF APPS

Category of apps	No. of app changes specifying the type of requirements		
	FR	***NFR***	***Others***
Social Networking	283	664	53
Travel	300	643	57
Books	285	602	113
Total:	868	1909	223

Figure 3 takes a closer look at the types of requirements identified in the 3000 app changes over different categories of apps. We observed that for each of these three categories, there is no big difference in the percentages of FR and NFRs that were subjected to changes. Relatively, however, NFRs got the most attention when developers updated the apps in the social networking category. Regarding the apps in the category of 'Books', more app changes are uninformative for RE, as indicated by 22.3% of app changes typed as 'Others'. Whereas, the updates of apps for travelling took a little more attention to FR, compared to the apps in the other two categories.

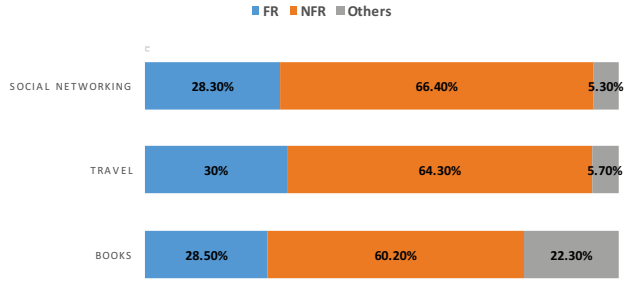


Fig. 3. Percentages of different requirements types identified in app changes over different categories of apps.

C. Answer to RQ2

Based on the answer to RQ1, this sub-section zooms in on the distribution of app changes pertaining to the five NFRs. Figure 4 indicates that usability and maintainability are the NFRs subjected to the most changes in all three categories of apps. Surprisingly, app changes referring to reliability are the least in terms of numbers in all three categories of apps. Furthermore, we gave a closer look at the NFRs to which the highest changes belong, per app category. For example, usability is observed as the NFR with the highest proportion in the 'Travel' category of apps. Whereas, maintainability is the NFR with the highest proportion in the apps of 'Social Networking' category, but the lowest proportion in the apps of 'Travel' category. Regarding *Reliability*, *Performance*, and *Portability*, the proportion of the app changes labelled as these types are not much different per app category.

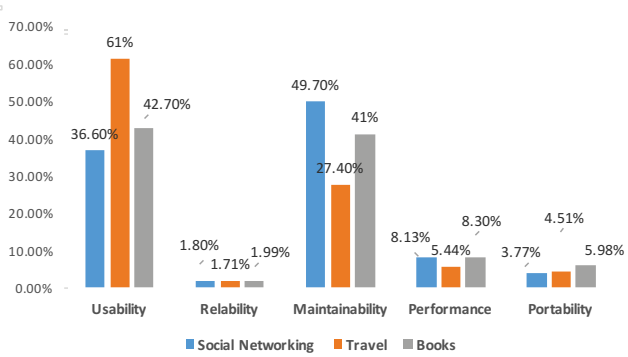


Fig. 4. Percentages of the five NFRs over categories of apps.

Since *Usability* and *Maintainability* are the top two NFRs dominating the app changes, we further investigated the growth over time of these two NFRs in the app changes. Figure 5 shows that over the years, for the apps in each of these three categories, developers' attention to usability is growing. Regarding the categories of 'Social Networking' and 'Books', the attention is increasing more slowly in recent years. Whereas, apps in the 'Travel' category continuously focused on the changes relevant to usability, and the interest is even growing much faster in 2018.

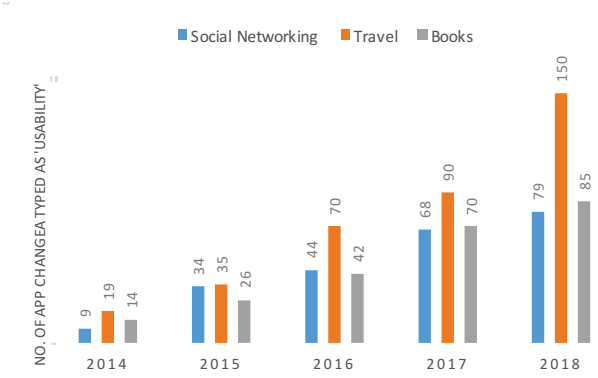


Fig. 5. Growth of app changes typed as Usability.

Similarly, Figure 6 shows the growth of app changes types as *Maintainability* over app categories. We found that although apps in the categories of 'Travel' and 'Books' kept increasing in the past five years, apps in 'Social Networking' category still released the highest number of app changes referring to *Maintainability*. Plus, we observed that the growth of app changes typed as *Maintainability* seems to be stable in the past two years.

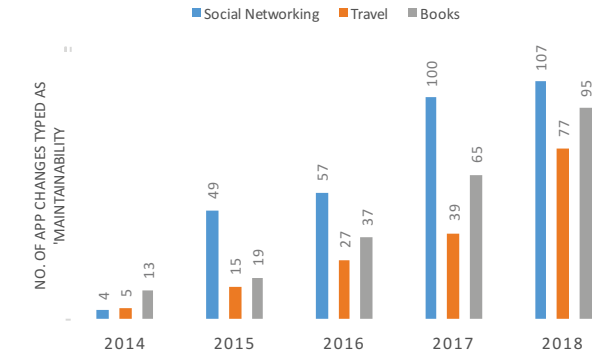


Fig. 6. Growth of app changes typed as Maintainability.

V. Discussion

A. Reflection on the overview of app changes

Regarding the increasing number of app changelogs and app changes, we observed that the updating of apps is more and more frequent with larger number of changes. One reason could be that the number of app users is increasing, so that more changes are needed to quickly respond users for the purpose of maintaining current users and attracting potential users.

When looking into the changelogs of apps from the perspective of app categories, one observation is that apps for social networking are updated the most frequently. The reason could be that social networking apps are becoming much more popular and users are updating their profiles frequently while demanding new functionality and quality levels. The more frequent use of social networking apps leads in turn to more issues that users expect to be resolved, and more FRs and NFRs to be improved or implemented in the new releases. Another observation is that the apps in the category of 'Travel' usually posted more changes, compares to the apps in the other two categories. The reason could be that the apps for travelling usually have closer relations with apps in other categories, and much more changes are needed for coordination with other apps. For example, Booking.com activates Google map to show the map from your current position to the destination labelled in Booking.com.

B. Reflection on the answer to RQ1

For RQ1, we first observed that FRs and NFRs dominate the changes of apps. This agrees with our understanding of official app changelogs: they provide information with less noise compared to reviews and could be treated as a high-quality data source for RE purposes. Second, we found that compared to the number of app changes (868) typed as FRs, the number of NFRs-typed app changes (1909) has more than doubled. The possible reason could be that app developers now focus much more on how to improve user experience on the new releases, rather than providing new functions. Also, we observed that apps in the category of 'Travel' gave more new FRs in the new releases, compared with the apps in the other two categories. The reason could be that when using travelling apps (e.g. Booking.com), more FRs are needed to coordinate and help switch to other apps (e.g. Google Map). Whereas, the apps for social networking might all have the same or very similar functionalities, as these applications are more mature. Therefore, the changes of apps in the category of 'Social Networking' are more referring to NFRs to maintain registered users and absorb new ones.

C. Reflection on the answer to RQ2

Regarding different types of NFRs, it was observed that *Usability* and *Maintainability* are the top two NFRs mentioned in app changelogs. One reason could be that app developers prefer to satisfy the users' needs and resolve their problems when using the apps. Another reason could be that if zooming in on the growth of these two NFRs per app category in recent years, more usability requirements are expected for apps in the category of 'Travel', and more bugs are needed to fix in the apps for social networking. The reason could refer to the nature of these two categories of apps. Travel apps compete on ease of use and user-friendliness of the app interface. Whereas, developers of social networking apps assume that users will keep updating their profiles on regular basis, and therefore take care of the ease and speed with which an app can be restored to operational status after a failure occurs.

VI. Limitations

This exploratory study has some limitations concerning the generalizability [8] of our findings. First, we included applications belonging to three categories only. One might think that we could have obtained different results if we have included other categories among those pre-defined in the Apple App Store [13]. We consider this an important issue that warrants future research. In addition, all our applications are from the Apple Store. It might be possible that results could be different if we analyzed applications of the same three categories but from Google Play. We however believe that the app sector is a very competitive one and if an app is known for certain outstanding features and qualities, then one can safely assume that the competitors will follow suit and would try as soon as possible to implement changes to their apps so that these apps can be as close as possible to the leading apps in each respective category.

Second, in this exploration, all the 3000 app changes were analyzed and labelled by two bachelor students majored in computer science. To get high-quality labels for analysis and synthesis, we took the following measures. Regarding the coders, before labelling, the two coders learned the body of knowledge of requirements engineering and software engineering from specialized courses in their bachelor program. Considering the process of labelling, as mentioned in Section III.C, we conducted a two-round pilot labelling to help the two coders get a consensus on understanding the meanings of different types of requirements, especially for NFR. Furthermore, 500 out of the 3000 labelled app changes were randomly selected and then checked by the first author to ensure the quality of labelling, for the purpose of reducing the effects of requirements-engineering and domain knowledge of these two coders. The agreement of this randomly checking is 98%, which means that the labels of those 3000 app changes could basically guarantee the quality of the analysis process and the conclusions.

VII. Conclusions and Future Work

This exploratory study looked into the changes and the changelogs pertaining to 120 apps in the Apple App Store, from a developers' perspective in regard to functional and non-functional requirements (FR and NFRs). We found that the majority of the changes in fact refer to NFRs. Developers seem to be busy with improving the quality aspects of their apps, and relative fewer changes referred to FR, according to the apps investigated in this study.

Our study included apps from three categories (Travel, Social networking, and Books); however, we found that usability and maintainability seem to be the NFRs that dominate the app changes across all three categories. It came as a surprise that reliability requirements formed only a fraction of the total number of non-functional requirements (NFRs) addressed in all changelogs of apps in the three selected App Store categories.

Our immediate future research is to explore how to use machine learning techniques to facilitate the automatic classification of app changes, in order to reduce the cost of manual labelling and then help developers quickly understand the changes of apps. Next, we plan to extend our exploration by including other categories of apps, e.g., well-being and health care apps in Apple App Store and Google Play. We consider this important in order to improve generalizability of the study results. Meanwhile, we also plan to conduct a survey from the developers of investigated apps, in order to confirm the findings from this study as well as the usefulness of the results for practitioners.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (No. 2018YFB1003800) and the National Natural Science Foundation of China (Nos. 61702378 and 61672387).

References

[1] Statista. Number of apps available in leading app stores as of 1st quarter 2019. 2019. Available at: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[2] C. Gao, J. Zeng, M. R. Lyu and I. King, "Online App Review Analysis for Identifying Emerging Issues," in Proc. of the 40th International Conference on Software Engineering (ICSE), Gothenburg, Sweden, ACM, 2018, pp.48-58.

[3] S. Hassan, W. Shang, A.E. Hassan, "An empirical study of emergency updates for top android mobile apps," Empirical Software Engineering, 2017, vol. 22, no. 1, pp.505-546.

[4] M. Nagappan, E. Shihab, "Future Trends in Software Engineering Research for Mobile Apps," in Proc. of Leaders of Tomorrow Symposium: Future of Software Engineering (FOSE) at the 23rd IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Osaka, Japan, IEEE, 2016, pp.21-32.

[5] M. Nayebi, B. Adams, G. Ruhe, "Release Practices for Mobile Apps - What do Users and Developers Think?", in Proc. of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER), Osaka, Japan, IEEE, 2016, pp.552-562.

[6] S. McIlroy, N. Ali, and A. E. Hassan, "Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store," Empirical Software Engineering, 2016, vol. 21, no. 3, pp.1346-1370.

[7] M. Nayebi, K. Kuznetsov, P. Chen, A. Zeller, G. Ruhe, "Anatomy of functionality deletion: an exploratory study on mobile apps," in Proc. of the 15th International Conference on Mining Software Repositories (MSR), Gothenburg, Sweden, 2018. ACM, pp. 243-253.

[8] R.K. Yin, Case Study Research: Design and Methods, Sage, 2008.

[9] K. Krippendorff, K. Content Analysis: An Introduction to its Methodology. Newbury Park, CA: Sage, 1980.

[10] S. Stemler, "An overview of content analysis". Practical assessment, research & evaluation, 2001, vol. 7, no. 17, pp.137-146.

[11] ISO, ISO/IEC 25010, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. FDIS, 2011.

[12] X. Zhao, J. Liu, K. Deng, "Assumptions behind intercoder reliability indices," Annals of the International Communication Association, 2013, vol. 36, no. 1, pp.419-480.

[13] T. McCain, The Art of the App Store: The Business of Apple Development, Wrox, 2011.