Empir Software Eng (2018) 23:3442–3475
https://doi.org/10.1007/s10664-018-9604-y

Studying the consistency of star ratings and the complaints in 1 & 2-star user reviews for top free cross-platform Android and iOS apps

Hanyang Hu[1] · **Cor-Paul Bezemer**[1] · **Ahmed E. Hassan**[1]

Published online: 22 March 2018

Abstract How users rate a mobile app via star ratings and user reviews is of utmost importance for the success of an app. Recent studies and surveys show that users rely heavily on star ratings and user reviews that are provided by other users, for deciding which app to download. However, understanding star ratings and user reviews is a complicated matter, since they are influenced by many factors such as the actual quality of the app and how the user perceives such quality relative to their expectations, which are in turn influenced by their prior experiences and expectations relative to other apps on the platform (e.g., iOS versus Android). Nevertheless, star ratings and user reviews provide developers with valuable information for improving the overall impression of their app. In an effort to expand their revenue and reach more users, app developers commonly build cross-platform apps, i.e., apps that are available on multiple platforms. As star ratings and user reviews are of such importance in the mobile app industry, it is essential for developers of cross-platform apps to maintain a consistent level of star ratings and user reviews for their apps across the various platforms on which they are available. In this paper, we investigate whether cross-platform apps achieve a consistent level of star ratings and user reviews. We manually identify 19 cross-platform apps and conduct an empirical study on their star ratings and user reviews. By manually tagging 9,902 1 & 2-star reviews of the studied cross-platform apps, we discover that the distribution of the frequency of complaint types varies across platforms. Finally, we study the negative impact ratio of complaint types and find that for some apps, users have higher expectations on one platform. All our proposed techniques and

Communicated by: Andrian Marcus

✉ Cor-Paul Bezemer
bezemer@cs.queensu.ca

Hanyang Hu
hyhu@cs.queensu.ca

Ahmed E. Hassan
ahmed@cs.queensu.ca

[1] Queen's University, Kingston, ON, Canada

our methodologies are generic and can be used for any app. Our findings show that at least 79% of the studied cross-platform apps do not have consistent star ratings, which suggests that different quality assurance efforts need to be considered by developers for the different platforms that they wish to support.

Keywords Mobile apps · Cross-platform apps · Android · iOS · Star rating · User reviews

1 Introduction

More than 1.2 billion smartphones were sold in 2014, which is an increase of 28.4% as compared to 2013 (Laurence and Janessa 2015). Most of these smartphones are running the Google Android or Apple iOS platform. As of the second quarter of 2015, Android and iOS together have a market share of 96.7% worldwide (Ramon et al. 2015b). Both Android (Google Play Store) and iOS (App Store) have an official app store with more than 1.5 million apps available as of July, 2015 (Ramon et al. 2015a).

As shown in earlier work (Mudambi and Schuff 2010), users rely heavily on star ratings and reviews to decide whether an app is worth downloading or not. A 2015 survey (Poschenrieder 2015) shows that 77% of the users will not download an app that has a star rating which is lower than 3 stars. In addition, Nayebi et al. (2016) find that app developers often deviate from their time-based release scheduling in order to address issues that are raised in user reviews. Consequently, it is important for apps to achieve high star ratings and good user reviews. Hence, developers are constantly seeking ways to improve the star ratings and user reviews of their apps.

Developers can make their app available for multiple platforms to reach more users and increase revenue. In order to maximize the revenue, apps must be top apps on all platforms for which they are available. Hence, developers should strive for delivering high-quality apps on all platforms. Indeed, Joorabchi et al. (2013) find that developers aim to provide a consistent user experience for such cross-platform apps. A consistent user experience is defined by various aspects, from having screen-to-screen functional consistency (Joorabchi et al. 2013) to receiving consistent star ratings and sentiments that are reflected in user reviews across platforms. However, the consistency of star ratings and user reviews of cross-platform apps has not been explored in depth.

As an understanding of the star ratings and user reviews is so critical for developers, we look into whether developers succeed in providing consistent star ratings and user reviews for their cross-platform apps. We base our investigation on a comparison of the star ratings and user reviews for apps that are available in both the Google Play Store and the App Store. We manually identify the 19 cross-platform apps that exist in the top 50 free apps chart of both stores. We study apps that exist in both top 50 charts, to ensure that we study apps that receive a similar level of attention of developers for both platforms due to the popularity of these highly-ranked apps. We collect two snapshots of star ratings and user reviews for the 19 studied apps. The snapshots are almost one year apart. In total, we analyze 34,258 star ratings that are collected for 19 cross-platform apps. We then collect and manually analyze 9,902 1 & 2-star reviews from the two snapshots. Throughout our paper, we focus on the star ratings and user reviews of cross-platform apps at both the platform level (i.e., considering the reviews and ratings of all apps on a specific platform together) and the app level (i.e., considering the reviews and ratings per app) to ensure that our comparisons control for the actual apps.

We notice that the results from analyzing two snapshots of star ratings and user reviews are similar yet some differences do exist. The goal of our study is not to report the exact values that are derived from studying one snapshot. Instead, our key goal is to highlight the commonalities and differences among the two snapshots, while providing a window into the life of the app developers of these cross-platform apps across a one-year time period. Therefore, our findings will reflect the overall trend across both the snapshots. We also discuss findings that varied across both snapshots. We address the following three research questions:

RQ1 How consistent are the star ratings of cross-platform apps?
At least 79% of the studied cross-platform apps do not receive a consistent distribution of star ratings on both platforms. 58% of the studied cross-platform apps are rated higher on Android. The difference in the distribution of star ratings of cross-platform apps suggests that users perceive the quality of these apps differently across platforms.

RQ2 How consistent are the 1 & 2-star reviews for cross-platform apps?
For 59% of the studied cross-platform apps, users complain in 1 & 2-star reviews more frequently about the crashing of the iOS versions of apps. On both platforms, users complain the most often about functional errors. For the same app, users complain differently across platforms. Developers can benefit from analyzing reviews for consistency-improving guidances.

RQ3 Are the most negatively impacting complaints consistent across platforms?
A complaint type has a higher negative impact on the star ratings if more 1-star reviews than 2-star reviews are found. We find that the negative impact ratio varies even for apps that have identical distributions of star ratings. The same complaint type for the same app can have a different impact on the star ratings across both platforms.

This paper is further organized as follows. Section 2 discusses related work. Section 3 presents the design of our empirical study. The findings of our study are presented in Section 4. In Section 5, we first discuss the implications of our findings, and we discuss automated approaches in tagging user reviews. We then discuss the generalizability of using the most recent 500 star ratings and reviews. Section 6 addresses the threats to the validity of our study. Finally, Section 7 presents our conclusion.

2 Background and Related Work

In this section we discuss prior work that is related to our study.

2.1 Cross-Platform Apps

Cross-platform apps have recently gained much attention in mobile app research. Ali et al. (2017) studied 80,000 app-pairs across app stores. In contrast to their study, we focus on top apps. In addition, Ali et al. (2017) discuss their results at the platform level, while our study provides a more detailed analysis at both the platform and app level. Joorabchi et al. (2013) conducted a survey on mobile app developers to explore the challenges that they encounter when developing apps for multiple platforms. The results of this survey show that developers treat the apps for each platform as separate projects, while the developers

manually try to preserve behavioral consistency. In more recent work (Joorabchi et al. 2015), Joorabchi et al. present a tool for automated verification of the aforementioned behavioral consistency. Chen et al. (2013) compared maturity ratings of cross-platform apps across Android and iOS. They observed that there is a considerable portion of Android apps that have a lower maturity rating than their iOS counterparts.

Developers use the native platform API or frameworks based on web technologies (e.g., HTML5) to build cross-platform apps. Several studies have examined the preferred strategy for building cross-platform apps (Hartmann et al. 2011; Heitkötter et al. 2013; Palmieri et al. 2012; Dalmasso et al. 2013). These studies conclude that each strategy has its own advantages and challenges. While web technologies provide an easier method for maintaining cross-platform behavioral consistency, native development may result in more responsive apps. For example, in order to provide a "fast, reliable experience", Facebook discarded their iOS app built in HTML5 in 2012 and switched to natively developed apps for each of its supported platforms (Dann 2012). Mercado et al. (2016) studied the differences in complaint types between apps that were built using cross-platform frameworks. These frameworks either use a shared codebase (i.e., using HTML5) to achieve cross-platform compatibility, or they generate native apps for each supported platform from a common language. Mercado et al. found that HTML5 apps tend to be more prone to complaints than generated apps.

While prior work on cross-platform apps focuses on the quality of an app from the developer's perspective, our work focuses on the quality of an app as reflected by the star ratings and user reviews that are provided by app users. Joorabchi et al.'s studies found that behavioral inconsistency does exist, yet these studies never confirmed that this behavioral inconsistency results in an inconsistent perception of app quality (in terms of ratings and reviews) across platforms. A multi-method approach, including surveys (as provided by Joorabchi et al.) and data analysis (as provided by our work), is necessary to fully understand inconsistency across app markets. The survey that was conducted by Joorabchi et al. gives only a partial view, i.e., that of the developers, on inconsistency across app stores. In our work, we complement this partial view by providing the remaining view, i.e., that of the user.

2.2 Analyzing Star Ratings and Reviews of Apps

Several prior studies have investigated the overall impression of users of mobile apps based on user-provided information. In most cases, such investigations were based on analyzing star ratings and user reviews. For a more thorough overview of studies that focus on analyzing star ratings and user reviews, we refer to the survey that was done by Martin et al. (2017).

2.2.1 Studies Using Manual Analysis

Harman et al. mined 32,108 BlackBerry apps and found a strong correlation between the average star rating of an app and the number of downloads (Harman et al. 2012). Khalid et al. (2015) manually categorized the 1 & 2-star reviews for 20 free iOS apps into 12 complaint categories. Khalid et al. find that users complain the most about functional errors and give more 1-star rating than 2-star ratings to issues that are related to privacy and ethics. Pagano and Maalej (2013) studied how and when users provide feedback on iOS apps and discovered that most user feedback is provided by app users shortly after a new app release. Pagano and Maalej applied manual analysis on 1 to 5-star reviews and found 17 topics in user feedback. In another study by Khalid et al. (2014), they mined user reviews of 99 free apps from the games category (and other app categories) in Android to develop strategies

for developers on how to test their apps. Khalid et al. found that while the number of device models that are mentioned in reviews is large, a small number of device models account for most of the reviews. Hassan et al. (2017) studied emergency updates for top Android apps, and found that emergency updates are often associated with simple mistakes, such as including a low quality image in the app.

2.2.2 *Studies Using Automated Analysis*

Tian et al. (2015) studied 28 different factors, such as the app size, for low and high-rated apps. They found that install size, the number of promotional images and the target SDK version are the most related to app rating. Martin et al. (2016) conducted causal impact analysis on the impact of releases on app success, and found that higher priced releases are more likely to be significant and to have a positive impact on the success of an app. Noei et al. (2017) studied the relation of mobile device attributes with the user-perceived quality of Android apps, and found that some device attributes (such as CPU) have a stronger relation with user-perceived quality than some app attributes (such as the number of UI inputs).

Chen et al. implemented "AR-Miner", a tool that automatically collects reviews and applies LDA modeling on the collected reviews (Chen et al. 2014). Their goal is to find and present the most informative reviews, i.e., reviews that provide valuable information. Fu et al. built WisCom, which is capable of detecting inconsistencies and identifying topics in user reviews using LDA (Fu et al. 2013). They conducted their study on Android apps. Guzman and Maalej combined LDA with sentiment analysis (Guzman and Maalej 2014). Their study outlined an automated approach for extracting reviews that contain feature-related information for requirements evolution tasks. McIlroy et al. (2015) studied 20 Android apps and 4 iOS apps and built an automated approach to categorize reviews according to the raised complaints. Yichuan et al. (2016) proposed a framework named "CrossMiner" which identifies seven types of issues that are raised in 1 & 2-star reviews of cross-platform apps. Using "CrossMiner", Yichuan et al. (2016) conducted two case studies on the reviews of two cross-platform apps: eBay and Spotify. They found that for both eBay and Spotify, users are concerned about different issues across the platforms on which these two apps are available.

Palomba et al. (2015) presented an approach to link user reviews to source code changes. In a study of 100 Android apps, Palomba et al. show that in most cases, developers take user reviews into account when updating their apps. Panichella et al. (2015) used a combination of natural language processing, sentiment analysis and text analysis to automatically classify reviews into categories, with the goal of improving the app maintenance and evolution process. Villarroel et al. (2016) proposed an automated approach for release planning based on the information that is available in app reviews. Their approach identifies bug reports and feature requests and clusters reviews that address the same issue together. The clusters are then prioritized, so that developers can address them in future releases. Di Sorbo et al. (2016) automatically generate user summaries of app reviews. Palomba et al. (2017) proposed an approach that recommends source code changes based on user reviews.

Bavota et al. (2015) and Linares-Vásquez et al. (2013) studied the impact of API change and fault-proneness on the user ratings of Android apps. They found that highly-rated apps use APIs that are less change and fault-prone than low-rated apps.

Most existing work on users' overall impression of the app as reflected by the user-provided information (i.e., star ratings and reviews) focuses primarily on a single app store. Our work focuses on the differences of users' overall impression across app stores. Yichuan et al.'s study analyzes cross-platform apps, however they study only two apps.

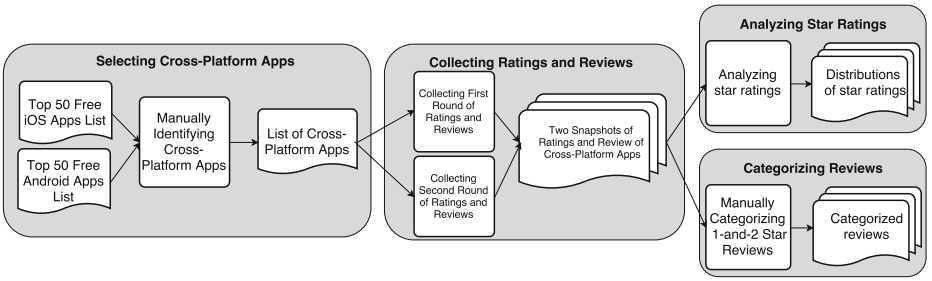


Fig. 1 Overview of our empirical study

3 Empirical Study Design

In this section, we describe the design of our empirical study of star ratings and user reviews of cross-platform apps. Figure 1 gives an overview of the steps of our study. In the remainder of this section, we describe each step in more detail.

3.1 Selecting Cross-Platform Apps

We focus on apps from the Google Play Store and App Store due to their dominating market share (Ramon et al. 2015b). In particular, we focus on popular apps in the U.S. version of the stores as these are the most likely to receive a large number of star ratings and reviews. AppAnnie (2015) tracks the top apps for both app stores. Prior research shows that the quality and maturity of apps vary considerably within the market, which echos with similar findings in the studies of Sourceforge (Howison and Crowston 2004) and GitHub (Kalliamvakou et al. 2014). It is advisable to start with a well-defined selection strategy instead of attempting to cover a large number of apps. Hence, we decided to study the top free 50 apps charts from Oct 30, 2015. Our intuition is that if a cross-platform app can reach the top 50 free apps chart in both app stores, we are more confident that both versions of the app 1) are likely to have reached a mature stage and 2) have plenty of user-feedback to analyze. We are also confident that the companies that produce such apps are committed to delivering their apps on both platforms. We use free apps instead of paid apps due to the fact that free apps account for more than 90% of total app downloads (Christy Pettey and Rob van der Meulen 2012). Also, our study of free apps helps us better control for confounding factors that affect the star ratings and user reviews (e.g., user reviews are likely influenced by how much one paid for an app).

Cross-platform app developers do not always use identical names for the same app across platforms. For example, Skype is a popular app for communication and it exists in both top 50 apps charts. In the App Store, the full name of Skype is *Skype for iPhone* while in the Google Play Store, the full name is *Skype - free IM & video calls*. Due to the naming discrepancies, we did not use an automated approach to match apps across stores. Instead, we manually inspected the top 50 apps charts and identified 19 cross-platform apps in these 50 apps.

3.2 Collecting Star Ratings and Reviews

Neither the App Store nor the Google Play Store provide public APIs to collect the entire set of reviews for apps. However, given an app ID, such as 284882215 for the *Facebook* app,

Apple offers a public RSS (Rich Site Summary) feed that allows us to collect the 500 most recent star ratings and reviews for the app (Apple 2008). We use a crawler for the Google Play Store (Akdeniz 2014) to collect the star ratings and reviews for Android apps. The 500 most recent star ratings and reviews provide only a momentary snapshot of the star ratings and user reviews of the app. Martin et al. warned about using reviews that are collected during a short time frame for review analysis (Martin et al. 2015). Therefore, we collect two snapshots of the 500 most recent star ratings and reviews for the studied cross-platform apps. The first snapshot of star ratings and reviews are collected from Nov 11, 2015 to Nov 20, 2015 across both stores. The second snapshot of star ratings and reviews is collected from September 1, 2016 to September 7, 2016. We removed all reviews that exist in both snapshots, to avoid bias in our study. In total, we removed one iOS review and zero Android reviews from our data set.

3.3 Analyzing Star Ratings

We intend to analyze the star ratings from both the perspective of a platform and from the perspective of an app. In particular, we first analyze the distribution of all collected star ratings at the platform level. We then analyze the distribution of all star ratings at the app level. We are interested in whether the distribution of star ratings that is observed at the app level is reflected as well at the platform level.

3.4 Tagging Reviews

A 2015 survey shows that 77% of the users will not download an app whose star rating is lower than 3 stars (Poschenrieder 2015). Therefore, understanding what users are complaining about in 1 & 2-star reviews is critical for developers who want their apps to receive higher star ratings. In their prior studies of analyzing the user issues that are raised in the reviews of mobile apps, McIlroy et al. (2015) conduct a sentiment analysis and find that 1 & 2-star reviews have the most negative sentiment in them.

We manually tag all 1 & 2-star reviews from both snapshots of the 500 most recent reviews for the studied cross-platform apps. The first author is a first year master's student and the second author is a postdoctoral researcher. Both first and second authors have and frequently use smartphones powered by Android and iOS. We start our tagging process with the 12 existing complaint types as identified by Khalid et al. (2015) in their study of complaints made in the user reviews of 20 free iOS apps. We do not use the tags that are identified in Pagano and Maalej's (2013) study because their tags are not oriented toward user complaints (e.g., they have tags about "praise"). Table 1 shows the identified complaint types and their descriptions. In case we encounter a new complaint type, we add it to the list and restart the tagging process. If a review does not contain a specific complaint or does not contain descriptive content, e.g., *"this app is bad"*, this review is tagged as 'Not Specific'. Every review can be tagged with multiple complaint types.

To validate our manual tagging process, we follow the following procedure. To validate the definition of complaint types, the first and second author of this paper tag the same 50 reviews. Then, any difference in tags were discussed and the definition of the complaint types is firmed up when necessary. Next, the first author tags 1,000 reviews for iOS and 1,000 reviews for Android using those definitions. The second author tags a statistically representative sample (95% confidence level and 10% confidence interval) of 88 reviews for both iOS and Android and the results are compared with those of the first author. We find that after the consolidation of definition, the first and the second author agree on the tags

Table 1 List of 12 complaint types with description and example review

# complaint type	Complaint type	Description	Example review
1	App Crashing	The app is crashing or cannot be opened or installed	*"Use to work. Now it crashes every time I tap the app. Doesn't even open"*
2	Compatibility	The app does not work as expected on a specific device or OS version	*"The app will not open on my iPad since new OS update."*
3	Feature Removal	A feature should be removed	*"There are way to commercials in every TV show. The are at least 20 in 1."*
4	Feature Request	A feature should be added	*"Bring back hashtags !!!!!!!!!!!!!!"*
5	Functional Error	A feature does not work as it is supposed to	*"Since This new update I can't log in...fix this immediately"*
6	Hidden Cost	The app requires unexpected monetary investment for the full experience	*"Charges???? This is NOT OK !!!!"*
7	Interface Design	The design, interface or visuals of the app are below user expectation	*"Get rid of the bottom right follow button"*
8	Network Problem	The app has network-related issues	*"Every time I launch the app, it always says, Re-establishing connection..."*
9	Privacy and Ethical	The app violates the privacy of the user or is considered unethical	*"Some of the picture on here are way to inappropriate but are showed anyway"*
10	Resource Heavy	The app consumes excessive system resources	*"1 hour on screen. 10.3 background, that is absurd."*
11	Uninteresting Content	Content of the app is unappealing	*"you guys haven't added any more movie that are latest."*
12	Unresponsive App	The app is slow to respond to user input or does not respond at all (i.e. freezes)	*"The app is getting slower and more unresponsive with each update."*
13	Not Specific	No specific complaint	*"Gets worse over time"*

of the majority of the user reviews. We find that 87.5% of the reviews have identical tags. 5.7% of the reviews have one or more common complaint tags and 6.8% of the reviews have completely different tags. We investigate reviews that have completely different tags and find that the discrepancies in most cases stem from reasonable but different interpretations. For example, one Netflix user complains that: *"No empire - Won't let me watch empire ... Everything else works :(empire Is the only thing I watch"*. The first author interpreted this review as a complaint about not having the show "Empire" on Netflix and tagged the complaint as 'Uninteresting Content'. The second author interpreted this review as a complaint about a 'Functional Error' that prevents this user from watching the show "Empire" on Netflix. We believe that both tags are justifiable. Therefore, differences in the tags of such reviews are difficult to overcome and cannot be completely resolved. Nevertheless, the first and second author have discussed and reinforced the descriptions for complaint types with the intention of minimizing the ambiguity of the compliant types. Finally, the first author restarts the manual analysis of all reviews with the updated descriptions of complaint types. All our data is made publicly available on our website (Hu et al. 2017).

4 Empirical Study Results

In this section, we present the results of our empirical study. In each section we discuss the motivation, approach and results for a research question. For each result, we discuss how we came to the result, based on the collected data from the two snapshots. The first snapshot is denoted as snapshot S1, whereas the second snapshot is denoted as snapshot S2.

4.1 How Consistent are the Star Ratings of Cross-Platform Apps?

Motivation Star ratings are the simplest quantification of users' overall impression of an app. Mudambi and Schuff (2010) show that Amazon users rely heavily on star ratings and reviews for deciding which product to purchase. Kim et al. (2011) find that word of mouth is the overall top purchase determinant in app stores. The results from these two studies show that app users will more likely download apps with higher star ratings. In order to increase the revenue of cross-platform apps, developers of such apps aim to maximize the star rating of their apps on all platforms. To find whether developers succeed in achieving consistently high star ratings on all their supported platforms, we analyze whether there is a difference in the star ratings that are given by users for the same app across different platforms. In particular, for a cross-platform app, we want to know whether the distribution of star ratings is similar across platforms. Differences in the distribution of star ratings of the same app on two platforms indicate a difference in users' overall impression of the app. Hence, developers of such apps need to look deeper into their user reviews to better understand the differences in users' impression of their app across platforms.

Approach We compare the distribution of the star ratings for both versions of each cross-platform app in three ways, using the average star rating, a Mann-Whitney U test, and the skewness and kurtosis of both distributions. We use all available star ratings, namely 1 to 5-star ratings, so the distribution of the star ratings can be viewed as a list of integers from 1 to 5.

We first calculate the average star rating for each cross-platform app on both studied platforms. In addition, we count the number of 1 & 2-star reviews as these are the most likely to contain user complaints (Khalid et al. 2015). To decide whether the star ratings for the iOS and Android-version of the same app differ significantly, we perform a Mann-Whitney's U

test with the default significance level ($\alpha = 0.05$). Thus, if the p-value computed by the Mann-Whitney's U test is smaller than 0.05, we conclude that the two input distributions are significantly different.

The p-value only indicates whether two distributions have a statistically significant difference. However, the p-value does not represent whether this difference is large enough to be noticeable in practice. Therefore, p-values should always be reported together with an effect size. The effect size quantifies the difference between two distributions, i.e., two distributions that are very different will have a large effect size. Hence, we include

Cliff's delta d (Long et al. 2003) effect size to quantify the difference in the distributions of star ratings for each cross-platform app. Cliff's delta returns a real number between -1 and 1. The absolute value of the returned number is used to assess the magnitude of the effect size. We use the following threshold for interpreting d, as provided by Romano et al. (2006):

$$\text{Effect size} = \begin{cases} negligible(N), & \text{if } |d| \leq 0.147. \\ small(S), & \text{if } 0.147 < |d| \leq 0.33. \\ medium(M), & \text{if } 0.33 < |d| \leq 0.474. \\ large(L), & \text{if } 0.474 < |d| \leq 1. \end{cases}$$

We calculate the skewness and kurtosis for the review ratings of both studied versions of each cross-platform app. The skewness of a distribution captures the level of symmetry in terms of mean and median, i.e., the skewness of the distribution of star ratings represents how positive or negative users feel about that version of the app. A negative skew means that users feel negative (i.e., more lower star ratings) about the app, while a positive skew means that users feel positive (i.e., more higher ratings). A skewness smaller than -1 or larger than 1 means that the skew is substantial (Graphpad Software 2015).

Kurtosis explains the peakedness of a distribution. The Gaussian distribution has a kurtosis of 3. A kurtosis higher than 3 means that the distribution has a higher peak than the Gaussian distribution, while a kurtosis lower than 3 means that the distribution is flatter. A high kurtosis means that users have a relatively strong consensus on the average star rating of the app, while a low kurtosis means that there is no clear consensus (i.e., agreement) between the users.

Results In snapshot S1, we collect 9,500 reviews of Android cross-platform apps and 9,500 reviews of iOS cross-platform apps.[1] In snapshot S2, we collect 8505 reviews of Android cross-platform apps and 6,753 reviews of iOS cross-platform apps. In total, we analyze 34,258 star ratings and reviews across both snapshots of the studied cross-platform apps. Table 2 shows that 17 out of 19 cross-platform apps have different distributions of star ratings in snapshot S1 and 15 out of 19 cross-platform apps have different distributions of star ratings in snapshot S2. These numbers indicate that at least 79% of the studied cross-platform apps do not receive the same distribution of star ratings on Android and iOS. We now present the results of our analysis on the star ratings of cross-platform apps.

The difference between star ratings in Android and iOS is negligible or small when considering all apps together. Table 2 shows that the distribution of star ratings in Android (with an average of 3.7) and iOS (average of 3.5) in snapshot S1 are significantly different according to the Mann-Whitney's U test. However, Cliff's Delta shows that the difference is negligible. The total number of low-star (i.e., 1 or 2-star) reviews in snapshot S1 is 3,059

[1]Note that we removed one review from the iOS snapshots (see Section 3.2).

Table 2 Statistics of the star ratings of cross-platform apps (500 reviews for each cross-platform app on each platform in snapshots S1 and S2)

App name	App type	Average rating			M.W.[b]	p-value	Effect[c]	1&2-star
		Android	iOS	Ratio[a]			Size	Ratio[a]
Snapshot S1:								
Amazon	Shopping	**4.0**	**3.0**	1.4	Y	2.3e-25	M	0.4
Facebook	Social	2.9	2.4	1.2	Y	6.1e-06	S	0.8
Hulu	Entertainment	3.0	2.7	1.1	Y	4.1e-04	N	0.8
Instagram	Photo & Video	4.2	4.1	1.0	N	3.7e-01	–	0.8
Kik	Social	2.9	3.2	0.9	Y	4.3e-03	N	1.2
Madden NFL	Games	4.3	3.8	1.1	Y	4.2e-07	S	0.6
McDonald's	Food & Drink	**3.4**	**2.3**	1.5	Y	2.8e-23	M	0.6
Messenger	Social	**3.5**	**2.5**	1.4	Y	4.5e-22	M	0.5
Netflix	Entertainment	3.4	2.9	1.2	Y	1.0e-06	S	0.8
Pandora	Music	**3.7**	**4.7**	**0.8**	Y	1.7e-32	M	5.0
Pinterest	Social	**4.2**	**3.5**	1.2	Y	4.2e-09	S	0.5
Pop the Lock	Games	4.1	4.4	0.9	Y	1.2e-04	N	1.7
Skype	Social	3.5	3.9	0.9	Y	1.6e-05	N	1.4
Snapchat	Photo & Video	2.4	2.2	1.1	Y	4.5e-02	N	1.0
SoundCloud	Music	**4.3**	**4.5**	**1.0**	Y	1.7e-02	N	1.8
Spotify Music	Music	**3.5**	**4.7**	**0.7**	Y	3.2e-50	M	5.4
Subway Surfers	Games	4.4	3.9	1.1	Y	1.6e-12	S	0.5
Twitter	Social	3.8	3.4	1.1	Y	5.8e-04	N	0.7
WhatsApp	Social	4.2	4.3	1.0	N	6.1e-02	–	1.1
Values across all apps		**3.7**	**3.5**	1.0	Y	7.8e-12	N	**0.93**
Snapshot S2:								
Amazon	Shopping	4.0	2.9	1.4	Y	6.5e-23	M	0.4
Facebook	Social	3.3	2.8	1.2	Y	1.8e-03	S	0.7
Hulu	Entertainment	3.4	2.2	1.6	Y	9.7e-24	M	0.5
Instagram	Photo & Video	4.1	2.7	1.5	Y	2.8e-15	M	0.2
Kik	Social	4.1	2.8	1.5	Y	5.5e-25	M	0.3
Madden NFL	Games	3.9	4.1	1.0	Y	1.8e-02	N	1.3
McDonald's	Food & Drink	3.6	1.9	1.8	Y	1.3e-11	L	0.4
Messenger	Social	2.9	2.3	1.3	Y	3.1e-08	S	0.7
Netflix	Entertainment	4.2	3.0	1.4	Y	4.2e-36	M	0.3
Pandora	Music	**3.9**	**4.2**	0.9	Y	6.3e-04	N	1.4
Pinterest	Social	**4.6**	**2.9**	1.6	Y	1.9e-59	L	0.1
Pop the Lock	Games	4.2	4.0	1.0	N	2.6e-01	–	0.5
Skype	Social	3.7	2.9	1.3	Y	9.4e-14	S	0.6
Snapchat	Photo & Video	2.7	2.8	1.0	N	3.7e-01	–	1.1
SoundCloud	Music	**4.0**	**4.5**	0.9	N	5.4e-02	–	3.5
Spotify Music	Music	**4.4**	**4.5**	1.0	N	1.0e-01	–	0.8
Subway Surfers	Games	4.2	4.7	0.9	Y	1.4e-12	S	3.3
Twitter	Social	4.2	3.3	1.3	Y	1.1e-21	S	0.3

Table 2 (continued)

App name	App type	Average rating			M.W.[b]	p-value	Effect[c]	1&2-star
		Android	iOS	Ratio[a]			Size	Ratio[a]
WhatsApp	Social	4.3	4.1	1.0	Y	2.4e-02	N	0.7
Values across all apps		**3.9**	**3.3**	1.2	Y	7.5e-91	S	**0.60**

[a]Ratios in this and following tables are calculated by Android / iOS

[b]Mann-Whitney's U test: Y: p-value smaller than 0.05. N: otherwise

[c]Effect size: N: negligible. S: small. M: medium

for iOS and 2,605 for Android. The last row of Table 3 shows that the skewness and kurtosis for both platforms are similar, which shows that both distributions are evenly symmetric and peaked. The histogram of the star ratings given for all studied Android and iOS apps in Fig. 2 confirms that similarity.

At a platform level, the difference of star ratings that are collected in snapshot S2 changes from negligible in snapshot S1 to small. As shown in Table 2, Cliff's Delta shows that

Table 3 Skewness and kurtosis ratio for the distribution of star ratings of the studied cross-platform apps in snapshot S1 and S2

App Name	Skewness ratio[a]		Kurtosis ratio[a]	
	Snapshot S1	Snapshot S2	Snapshot S1	Snapshot S2
Amazon	**−30.8**	−11.8	2.1	2.0
Facebook	0.1	−1.4	0.7	1.0
Hulu	−0.1	−0.5	0.9	0.6
Instagram	1.0	−4.0	1.1	2.6
Kik	−0.4	−8.6	1.0	2.6
Madden NFL	1.8	0.7	1.9	0.7
McDonald's	−0.6	−0.5	0.7	0.6
Messenger	−1.0	0.1	1.1	0.6
Netflix	−3.6	34.5	1.0	2.9
Pandora	0.2	0.7	**0.2**	**0.6**
Pinterest	2.6	−24.8	2.3	7.3
Pop the Lock	0.6	1.4	0.5	1.6
Skype	0.5	−10.8	0.7	1.4
Snapchat	0.9	1.6	0.9	1.1
SoundCloud	0.8	0.6	**0.6**	**0.5**
Spotify Music	0.1	1.0	**0.1**	**1.0**
Subway Surfers	2.0	0.5	2.3	0.3
Twitter	1.8	6.0	1.4	3.1
WhatsApp	0.9	1.2	0.8	1.4
Ratings of all apps	1.4	2.9	1.2	1.7

[a]Ratios in this and following tables are calculated by Android / iOS

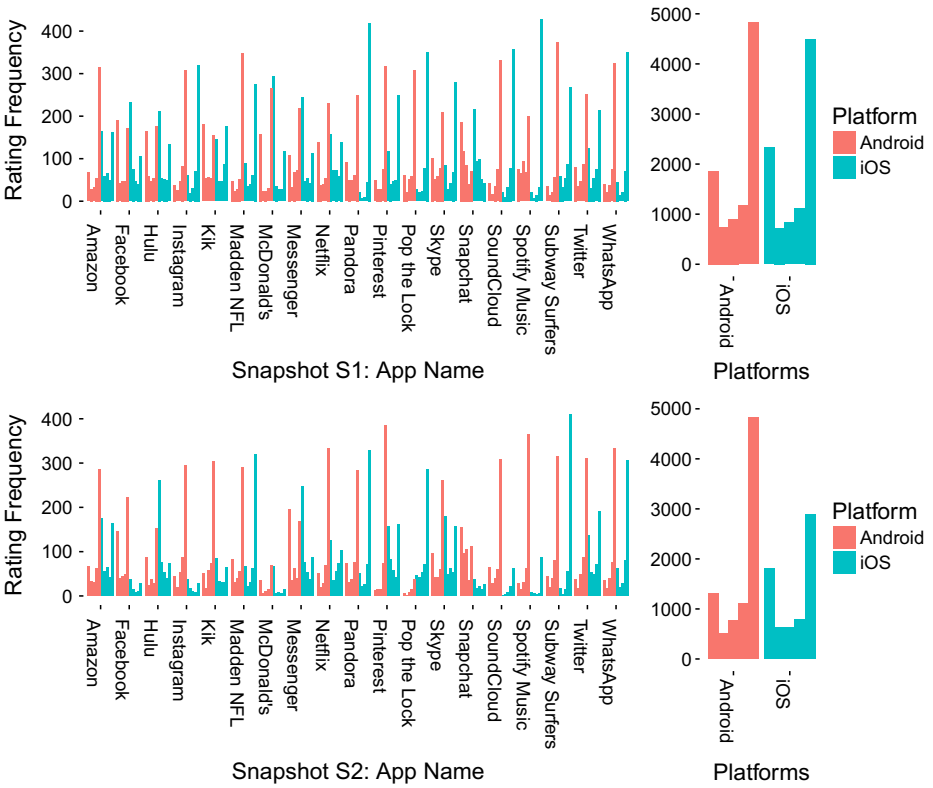


Fig. 2 Distribution of star ratings of cross-platform apps on Android and iOS across snapshots S1 and S2. (The leftmost bar for each app represents the frequency of 1-star ratings, while the rightmost bar represents the frequency of 5-star ratings)

the difference is small even though the p-value from Mann-Whitney's U test indicates the difference is significant. The average star rating at a platform level is 3.9 in Android and 3.3 in iOS, whereas in snapshot S1, the average star ratings are 3.7 and 3.5 respectively. The 1 & 2-star ratio decreases from 0.93 in snapshot S1 to 0.60 in snapshot S2. Even though our crawler collects more star ratings from Android users in snapshot S2, iOS users provide more 1 & 2-star reviews. The graph on the right side of Fig. 2 shows that while the number of 5-star ratings is clearly larger in Android, cross-platform apps in iOS receive more 1-star ratings and the number of 2-star ratings remains similar across platforms. Thus, both of our snapshots of star ratings suggest that at a platform level, the difference between star ratings in Android and iOS is negligible or small.

Almost twice as many cross-platform apps have a higher average star rating in Android than in iOS. This observation highlights the importance of performing our analysis at the app level, since app-level differences are likely not to be visible when the reviews are examined as a whole. In snapshot S1, when we examine closely the star ratings for each app, we find that 11 cross-platform apps have average ratings that are significantly higher on Android than on iOS (note that two have a negligible effect size). Table 2 shows the average star ratings for all apps. Out of these 11 cross-platform apps, McDonald's, Amazon and Messenger show the largest difference in average star rating on Android and iOS. For these

three apps, the differences in the average star ratings are larger than 1. The effect sizes that are calculated for the star rating distributions of these apps indicate that the star ratings differ at a medium level. We notice that the Android versions of these three apps have half as many 1 & 2-star reviews as their iOS counterparts. Table 3 shows that the ratio of skews for these apps have a negative sign, which means that they are skewed in different directions. For Messenger, the skewness ratio shows that users of that app have a polarized impression of the app: for the Android version, users are as positive as they are negative for the iOS version. The skew of -30.80 for Amazon shows that the star ratings for the Android version is much more skewed than the star ratings for the iOS version.

In snapshot S2, we find 13 cross-platform apps whose average star ratings are significantly higher in Android (note that two have a negligible effect size), while the number is 11 in snapshot S1. We find that the differences in average star ratings change over time. For example, the difference in average star ratings for the app Pinterest increases from 0.7 to 1.7 stars, and the effect size for Pinterest is large in our snapshot S2 in contrast to small in snapshot S1. This change highlights the dynamic nature of star ratings over time. Nevertheless, we continue to observe more versions of the cross-platform apps getting higher average star ratings in Android than their iOS counterparts.

Users are more satisfied about music apps in iOS than in Android. In snapshot S1, we find three music apps, Pandora, SoundCloud, and Spotify Music, that have higher average star ratings in iOS. In fact, they have the highest average star ratings of the studied cross-platform apps. The kurtosis ratios of these apps are smaller than 0.7, which indicates that the star rating distribution of these apps have a higher kurtosis for iOS. This suggests that there exists a strong agreement across iOS users regarding the quality of these apps, while Android users have more scattered opinions on the quality of these apps.

Using star ratings from snapshot S2, we find the average star ratings for these three music apps remain higher in iOS than in Android. As shown in Table 2 for snapshot S2, the average star ratings of Pandora, SoundCloud and Spotify are 0.3, 0.5 and 0.1 star higher in iOS respectively. Nevertheless, the differences in the star ratings for this type of cross-platform app across the two studied platforms are becoming much smaller.

4.2 How Consistent are the 1 & 2-Star Reviews for Cross-Platform Apps?

Motivation In order to improve the star ratings and user reviews of a cross-platform app, developers must know what unsatisfied users are complaining about. Knowing which types of complaints are commonly made about a specific version of their cross-platform app, helps developers better understand the challenges that are associated with developing for the platform on which that version is running. In addition, by analyzing the differences in user complaints, developers can get a deeper understanding of the differences in the expectations and concerns of users across all supported platforms.

Approach To find out what users complain about in cross-platform apps, we analyze all 1 & 2-star reviews for the studied apps. We manually tag all 1 & 2-star reviews as described in Section 3, starting from the set of complaint types found by Khalid et al. (2015) (see Table 1). For each cross-platform app, we apply the Mann-Whitney U test on the distribution of complaint types using the same setting as described in Section 4.1.

In addition, we perform frequent itemset mining to find the complaint types that are frequently mentioned together in one review. ECLAT is a frequent itemset mining algorithm introduced by Zaki (2000). ECLAT requires two parameters, minimum support (i.e., how often do the complaint types occur together) and minimum pattern length (i.e., how

many complaint types must at least occur together). We use the default settings for these parameters, i.e., we set the minimum support to 0.1 and the minimum pattern length to 2.

Finally, we run Twitter-LDA (Zhao et al. 2011) to extract the issues (topics) that are discussed in the negative (1 & 2-star) reviews. Compared to the original LDA by Blei et al. (2003), Twitter-LDA is able to identify more meaningful topics in short documents, such as tweets and app reviews. Since it is difficult to compare extracted topics from different LDA runs, we run Twitter-LDA on the combined set of iOS and Android reviews of an app in one snapshot. Then, we compare the top-3 topics (ranked by the number of reviews that discuss each particular topic) of an app across platforms. We conduct the following preprocessing steps on the reviews:

- Change all words to lower case (e.g. *App* to *app*)
- Remove punctuation
- Remove English stop words (e.g. *the*)
- Stem user reviews using the Porter stemmer (Porter 1997) from the Python NLTK library (e.g. *meeting* to *meet*)

We run Twitter-LDA on apps that have more than 100 negative reviews combined in iOS and Android in a snapshot, to increase the chances of extracting meaningful topics. As a result, we run Twitter-LDA on all studied apps except Pop the Lock, SoundCloud, Spotify Music and Subway Surfers. Hence, we run Twitter-LDA twice (for each snapshot) for 15 apps in total. We configure Twitter-LDA to extract 10 topics. The top-3 extracted topics for each app are available on our website (Hu et al. 2017).

Results In snapshot S1, we find 2,605 1 & 2-star reviews in the 9,500 reviews of Android cross-platform apps users. We find 3,059 1 & 2-star reviews in the 9,500 reviews of iOS cross-platform apps users. In snapshot S2, we find 1,809 1 & 2-star reviews in the 8,505 reviews of Android cross-platform apps users. Finally, we find 2,429 1 & 2-star reviews in the 6,753 reviews of iOS cross-platform apps users. In total, we manually tag all 9,902 1 & 2-star reviews from snapshots S1 and S2 for the studied cross-platform apps. We now present the results of our manual analysis of the 1 & 2-star user reviews.

68% of the studied cross-platform apps have a different distribution of complaint types between their Android and iOS counterparts. In snapshot S1, we find 15 out of 19 cross-platform apps whose distribution of complaint types are different across platforms. For these 15 cross-platform apps that receive different types of complaints, we check whether their most frequent complaint type is the same across the Android and iOS versions. 10 out of these 15 apps do not share a common most frequent complaint type. For example, Amazon users complain most about a crashing app in iOS but they complain most about a functional error in Android. A deeper investigation of these 10 apps shows that 5 out of them have 'App Crashing' as the most frequent complaint type on one platform and 'Functional Error' on another. In snapshot S2, we find 13 cross-platform apps that have a different distribution of complaints between their Android and iOS counterparts. 9 of them do not share a common most frequent complaint type. The 4 cross-platform apps that share a common most frequent complaint type all have functional error as the most frequent complaint type.

In addition, we compared the top-3 occurring topics that were extracted by Twitter-LDA for the 15 apps that had more than 100 negative reviews for iOS and Android combined (per snapshot). Table 4 shows that for only two apps in snapshot S1 (Kik and Pinterest) and one app in snapshot S2 (McDonald's), two out of three topics in the top-3 were the same across platforms. For Kik (the *hashtag* feature) and Pinterest (the *boards* feature), the complaints were mostly about wanting a removed feature back. For McDonald's, the complaints were

Table 4 The number of common top-3 topics that were extracted by Twitter-LDA across platforms

App name	Common top-3 topics	
	Snapshot S1	Snapshot S2
Amazon	1	1
Facebook	0	0
Hulu	1	1
Instagram	0	0
Kik	2	0
Madden NFL	0	1
McDonald's	1	2
Messenger	0	0
Netflix	1	1
Pandora	1	0
Pinterest	2	1
Pop the Lock	–[a]	–[a]
Skype	0	1
Snapchat	0	0
SoundCloud	–[a]	–[a]
Spotify Music	–[a]	–[a]
Subway Surfers	–[a]	–[a]
Twitter	0	1
WhatsApp	0	1

[a]The number of negative Android and iOS reviews combined was less than 100, so we did not run Twitter-LDA on the reviews of these apps

about the app crashing. However, no further explanations were given for the crashes. For five apps in snapshot S1 and eight apps in snapshot S2, one out of three topics were the same, and respectively eight and six apps had zero top-3 topics in common. Our Twitter-LDA findings reiterate that not only the complaint types are different across platforms, but also the actual content of the complaints.

Users have different complaints even for apps whose overall star ratings are consistent. In snapshot S1, we find that for 2 apps, Instagram and WhatsApp, their overall star ratings are not significantly different across platforms. However, the 2 apps have large differences in their distributions of complaints. Table 5 shows the ratio of the frequency of complaint types in Android and iOS for each app. We find Instagram has 8.7 times more complaints about a network problem in Android. WhatsApp has 10 (i.e., ratio is 0.1) times more complaints about 'Compatibility' in iOS. We manually studied the reviews of the apps that are discussed in this and the following paragraph, but we did not observe commonalities in the complaints. Hence, we do not discuss the differences in more detail.

In snapshot S2, we find 4 cross-platform apps whose distributions of star ratings are not significantly different across platforms. They are: Pop the Lock, Snapchat, SoundCloud and Spotify Music. For those 4 cross-platform apps, we find that the distributions of complaint types vary. For example, as shown in Table 6, Snapchat has 10 times more complaints about 'Feature Removal' on iOS. This finding emphasizes that star ratings alone are not sufficient for studying the differences in user reviews of cross-platform apps.

For more than 59% of the studied cross-platform apps, users complain more frequently about a crashing app in iOS. At a platform level, 'App Crashing' is one of the most frequent

Table 5 Ratio of the frequency of complaint types in snapshot S1 and S2

Complaint type	Percentage of[a] (%)		Percentage
	Android	iOS	Ratio
Snapshot S1:			
App Crashing	13.55	23.14	0.59
Compatibility	8.71	11.31	0.77
Feature Removal	2.76	6.96	0.40
Feature Request	10.86	15.14	0.72
Functional Error	32.71	30.37	1.08
Hidden Cost	1.31	2.39	0.55
Interface Design	2.07	6.24	0.33
Network Problem	6.79	3.27	2.08
Privacy and Ethical	5.95	7.26	0.82
Resource Heavy	2.00	2.03	0.99
Uninteresting Content	6.68	7.88	0.85
Unresponsive App	8.41	10.85	0.78
Not Specific	19.88	7.49	2.65
Snapshot S2:			
App Crashing	15.04	20.58	0.73
Compatibility	7.41	7.04	1.05
Feature Removal	4.98	12.23	0.41
Feature Request	3.98	13.59	0.29
Functional Error	33.33	30.22	1.10
Hidden Cost	3.15	1.48	2.13
Interface Design	1.71	5.60	0.31
Network Problem	6.08	4.16	1.46
Privacy and Ethical	4.42	5.31	0.83
Resource Heavy	3.10	1.85	1.67
Uninteresting Content	9.51	10.21	0.93
Unresponsive App	7.85	7.78	1.01
Not Specific	22.61	9.14	2.47

[a]Note that these percentages do not add up to 100 as a complaint can be tagged with multiple types

complaint types for both Android and iOS. In snapshot S1, we find that 13.55% of the complaints are about 'App Crashing' in Android and 23.14% in iOS (see Table 5). In particular, 14 of the studied apps have more 'App Crashing' complaints (i.e., ratio < 1), as shown in the first column of Table 6. In snapshot S2, we find that 15.04% of the complaints are about 'App Crashing' in Android and 20.58% in iOS. At a platform level, iOS users complain more about 'App Crashing' than Android users.

Compared to the results in snapshot S1, in which we find that users are complaining about a crashing app for every cross-platform app, we find 17 out of 19 cross-platform apps that receive complaints about a crashing app in snapshot S2. As shown in Table 6, we find 10 cross-platform apps whose users complain more frequently about crashing app in iOS. Although the number of cross-platform apps that receive more complaints about the crashing in iOS is decreasing from 14 to 10 apps, our results still suggest that cross-platform developers need to pay extra attention to the crashing issues in the iOS version of their apps.

Table 6 Ratio of the percentage of frequency of complaint types in snapshot S1 and S2

App name	Ratio of the percentage of frequency of complaint types												
Complaint type #	1	2	3	4	5	6	7	8	9	10	11	12	13
Snapshot S1:													
Amazon	**0.5**	**0.2**	0.9	0.4	1.6	8.0	0.7	–	1.9	13.4	3.3	0.8	5.1
Facebook	1.3	**0.6**	0.6	1.0	0.6	–	0.1	2.1	0.2	0.4	0.6	1.5	3.2
Hulu	**0.6**	1.1	0.2	–	0.7	0.6	0.1	6.8	0.6	–	0.7	0.8	1.7
Instagram	**0.5**	**0.4**	–	0.6	1.1	–	–	**8.7**	0.4	–	–	0.6	5.0
Kik	**0.7**	**0.8**	0.3	0.9	1.7	–	0.4	0.8	0.8	0.4	1.1	0.4	2.3
Madden NFL	**0.3**	**0.6**	–	2.8	0.5	–	–	0.4	0.5	–	2.9	1.4	2.5
McDonald's	1.7	**0.7**	–	0.6	0.9	–	0.2	0.3	1.3	0.5	0.5	0.8	1.6
Messenger	**0.6**	**0.6**	0.3	1.7	0.4	–	0.7	0.4	1.9	3.5	0.9	1.1	3.0
Netflix	**0.5**	**0.7**	–	0.2	1.1	–	0.1	2.8	2.6	–	1.0	0.9	5.5
Pandora	**0.6**	2.3	0.5	0.7	1.0	–	–	0.1	0.2	–	–	–	2.1
Pinterest	**0.9**	**0.4**	0.7	0.5	1.4	–	0.7	4.5	3.1	0.7	0.4	1.3	4.0
Pop the Lock	**0.4**	–	1.9	0.9	0.6	–	–	–	0.6	–	1.7	0.7	1.0
Skype	2.0	**0.7**	1.5	1.7	0.7	–	2.6	0.2	2.2	1.1	0.5	1.0	2.4
Snapchat	**0.2**	1.4	–	2.3	4.6	0.1	1.1	2.5	–	2.3	3.4	0.6	3.4
SoundCloud	**0.4**	**0.8**	–	0.4	1.8	–	0.6	–	–	–	1.5	**3.0**	6.7
Spotify Music	1.4	**0.7**	0.4	0.4	1.8	2.1	–	1.9	0.3	–	**0.2**	0.7	1.3
Subway Surfers	**0.1**	–	1.2	0.6	0.1	–	–	–	2.5	–	0.3	0.6	15.9
Twitter	3.2	6.7	0.1	0.1	4.1	–	0.8	–	4.8	3.8	1.4	0.8	10.9
WhatsApp	**0.3**	**0.1**	–	1.2	0.4	–	–	1.7	1.2	–	2.3	0.2	4.6
Snapshot S2:													
Amazon	4.5	1.5	–	0.2	0.8	1.0	0.1	0.6	1.7	1.7	1.2	1.7	1.2
Facebook	1.0	0.9	0.4	0.1	0.8	–	–	10.7	0.9	0.3	0.3	0.6	2.3
Hulu	1.2	0.5	0.7	0.0	1.4	8.0	–	0.5	2.1	1.0	1.3	3.0	1.7
Instagram	1.1	2.5	0.2	0.1	1.7	–	0.5	–	0.7	–	0.5	3.0	7.4
Kik	**0.6**	–	4.2	1.1	0.4	–	–	–	0.9	–	10.5	0.3	12.6
Madden NFL	2.6	1.4	**3.1**	1.9	0.5	0.7	–	–	–	–	0.8	1.5	1.3
McDonald's	**0.3**	–	–	–	1.3	–	–	–	1.2	–	0.9	0.6	1.8
Messenger	**0.3**	0.9	0.7	0.3	1.4	–	0.6	3.4	1.7	1.2	0.6	0.6	2.7
Netflix	**0.6**	1.9	–	–	1.0	6.5	0.4	0.4	2.2	–	0.9	0.6	1.7
Pandora	**0.2**	0.7	2.0	–	17.6	–	–	5.9	0.7	1.3	1.5	–	3.5
Pinterest	**0.4**	–	4.1	0.5	0.5	–	0.8	–	2.9	–	0.9	0.5	4.3
Pop the Lock	–	–	0.8	–	–	–	–	–	–	–	1.5	–	4.0
Skype	**0.8**	0.2	0.2	0.8	0.9	–	0.3	0.7	0.4	0.3	1.3	0.4	5.6
Snapchat	**0.7**	3.1	**0.1**	0.5	1.6	–	0.1	0.4	0.3	1.1	3.8	1.3	0.9
SoundCloud	–	–	0.3	–	0.6	–	–	–	–	–	0.7	–	1.1
Spotify Music	**0.7**	–	0.5	0.7	0.9	–	–	–	–	–	0.1	–	–
Subway Surfers	**0.7**	0.3	0.2	1.7	0.4	–	–	–	–	–	1.2	0.3	3.3
Twitter	1.4	0.7	0.8	0.4	0.5	–	1.7	4.2	0.3	1.4	–	0.5	4.6
WhatsApp	1.7	0.3	–	–	0.9	–	–	–	0.2	–	–	–	3.9

Example: Amazon has 11.3% of reviews in Android and 24.7% of reviews in iOS tagged as 'App Crashing' Hence, Amazon's ratio of the percentage frequency of 'App Crashing' is 0.5 (11.3% divided by 24.7%)

Note: "–" represents that the number of complaints is 0 in one or both platforms

Entries in bold are discussed in the text

Identifying the root cause of this phenomenon is difficult. Three of the possible scenarios that may help to explain the phenomenon are: 1) iOS users may be more sensitive to crashing issues, 2) apps may crash more often in iOS or 3) backwards compatibility is more difficult to provide in iOS. For developers, these three possible scenarios require them to spend more effort on their iOS app to avoid crashing issues. We also discuss the difficulty in identifying the root cause of platform differences in Section 5.

To identify the complaint types that frequently occur together in a review with the 'App Crashing' complaint type, we apply the ECLAT algorithm. In both snapshots, 'Functional Error' and 'Compatibility' are the complaint types that occur together most frequently with the 'App Crashing' complaint type. Table 7 shows the complaint types that occur together with the 'App Crashing' complaint type with a support of at least 0.1. In both snapshots, the support for the 'Functional Error' and 'Compatibility' complaint types is at least 0.3. In snapshot S1, the support for the 'Compatibility' complain type is twice as high as for the 'Functional Error' complaint type in iOS reviews. In snapshot S2, the support for the 'Compatibility' complaint type is also higher than for the 'Functional Error' complaint type.

As the higher occurrence of the 'Compatibility' complaints together with complaints about a crashing app in iOS is counterintuitive, due to the strict app publishing guidelines (Apple 2017) and the lower number of different iOS devices, we manually studied the iOS reviews. However, reviewers generally do not give concrete reasons as to why an app crashes, e.g.: *"Since the release of iOS nine there have been at least two revisions to the Amazon app but it still crashes on my iPhone every time I get to an individual product page."* One possible explanation is that the app is simply of low quality. Another possible explanation is that the problem is on the reviewer's side, for example, the device needs to be rebooted before the app will work (Apple 2016). We discuss further possible reasons for the higher rate of complaints about a crashing app in iOS in Section 5.1.

In more than 62% of the studied apps, iOS users complain more about compatibility issues. In snapshot S1, we find that 13 out of 19 (68%) studied cross-platform apps receive more complaints about compatibility issues from iOS users. In snapshot S2, we find that 62% of the studied cross-platform app receive more complaints about compatibility issues from iOS users. As shown in Table 6, we find 13 studied cross-platform apps in S2 that

Table 7 Complaint types that occur the most frequently together with the 'App Crashing' complaint type in Android and iOS

Platform	Complaint type[a]	Support
Snapshot S1:		
Android	Functional error	0.50
	Compatibility	0.48
iOS	Compatibility	0.62
	Functional error	0.31
Snapshot S2:		
Android	Functional error	0.46
	Compatibility	0.43
	Network problem	0.11
iOS	Compatibility	0.45
	Functional error	0.36
	Feature removal	0.12
	Interface design	0.12

[a]Compliant types in this table have a support of at least 0.1

Table 8 Complaint types that occur the most frequently together with the 'Compatibility' complaint type in Android and iOS

Platform	Complaint type[a]	Support
Snapshot S1:		
Android	Functional error	0.46
	App crashing	0.31
iOS	App crashing	0.45
	Functional error	0.44
Snapshot S2:		
Android	Functional error	0.59
	App crashing	0.27
iOS	Network problem	0.38
	Functional error	0.25
	App crashing	0.19
	Feature removal	0.19

[a]Compliant types in this table have a support of at least 0.1

receive complaints about compatibility issues, and 8 of these apps have a ratio that is less than 1, which indicates that iOS users complain more often about compatibility issues.

To identify the complaint types that frequently occur together with compatibility complaints, we again apply the ECLAT algorithm. As shown in Table 8, 'Functional Error' and 'App Crashing' are the complaint types that occur most frequently together with the 'Compatibility' complaint type in Android. For iOS, the complaint types that occur most frequently together with 'Compatibility' differ across snapshot S1 and S2. 'App Crashing' and 'Functional Error' have the highest support in snapshot S1 but 'Network Problem' has the highest support in snapshot S2. After manual inspection of the reviews, we found that at the time of recording the S2 snapshot, some apps had issues with connectivity to Apple TV or Chromecast devices on iOS. These devices are used to project the content of an app onto a larger screen (for example, Netflix movies). Hence, most complaints about network problems and compatibility for iOS apps in snapshot S2 are from users who were not able to connect their apps to one of these devices.

Surprisingly, despite the availability of more than 24,000 Android devices (Morani 2015), users complain more often about compatibility issues for the iOS version of a cross-platform app. One possible reason is that iOS users try to run a new app on an older device, which may not be compatible. A Snapchat user complains: *"I like snapchat but... I would really like it if snapchat could make it possible to use the lenses on the IPod 5th gen."* We discuss further possible reasons for the higher rate of complaints about compatibility in iOS in Section 5.1.

4.3 Are the Most Negatively Impacting Complaint Types Consistent Across Platforms?

Motivation In addition to the frequency of complaints, we are also interested in their severity, namely, the negative impact on the star rating of the app. Negative impact of a complaint type refers to how badly app users experience an issue corresponding to its complaint type.

As discussed earlier, app users consider star ratings as the most important determinant for app purchases. Understanding what types of complaints have a high negative impact on the star rating help developers to understand what users dislike the most.

Approach The negative impact ratio of a complaint type is calculated by examining all the 1 & 2-star reviews that have that complaint, then dividing the number of 1-star reviews by the number of 2-star reviews (Khalid et al. 2015). We calculate the negative impact ratio at both the platform level (Table 9) and the app level (Table 10). A complaint type that has a negative impact ratio that is larger than 1 implies a higher negative impact than a complaint type of which the negative impact ratio is lower than 1. For example, if an equal number of users are complaining about 'App Crashing' and 'Network Problem', and 'App Crashing' receives a higher negative impact ratio than 'Network Problem', developers may want to solve crashing issues first since such 1-star complaints will have a stronger impact on the overall star rating of the app.

We also calculate the ratio of negative impact ratio for each complaint type for each app across both platforms. The ratio of negative impact ratios is calculated by dividing the negative impact ratio in Android by that in iOS. A ratio of negative impact ratios larger than 1 for a complaint type indicates that users are more annoyed when they make such complaints in Android relative to the same complaint type in iOS.

Results Negative impact ratios vary even for apps of which the distributions of star ratings across platforms are identical. In snapshot S1, we find two cross-platform apps, Instagram and WhatsApp, that have the same distribution of star ratings across platforms, as shown in Section 4.1. However, the ratios of negative impact ratio for certain complaint types show differences in users' star ratings toward different complaint types, as shown in Table 10. For example, for Instagram, the negative impact ratio for 'Feature Request' is 12 times higher in Android, and for WhatsApp, the negative impact ratio for 'Feature Request' is 5 times higher in iOS. We examine the complaints for Instagram regarding 'Feature Request'. We find that iOS users are requesting support for HD video, support for Arabic hashtags, and a picture adjustment tool. In Android, users are requesting HD video, a feature to copy URLs, and auto-save of pictures. For WhatsApp, we also find that users are requesting different features. Different requested features might help to explain the big difference in negative impact ratio because users may deem some features essential and others optional. For example, this user considers animated GIF support for WhatsApp essential for iOS: *"Add gif support!!!! This would get my 5 stars right away!"*

In snapshot S2, we also find that the negative impact rating varies even for apps of which the distribution of star ratings across platforms is identical. As shown in Table 2, there are 4 apps whose distributions of star ratings across platforms are identical: Pop the Lock, Snapchat, SoundCloud, Spotify Music. We examine the negative impact ratio in Table 10 and find that the negative impact ratio varies in one or multiple complaint types. For example, for the Spotify Music app, the negative impact ratio for the fifth complaint type, 'Functional Error' is 0.5. This indicates that iOS users give more 1-star ratings regarding a functional error of the Spotify Music app. Although the negative impact ratio of a complaint type does not remain the same over time, the differences in the negative impact ratio varies even for apps whose distributions of star ratings across platforms are identical. To improve the overall star ratings, developers can use negative impact ratios to identify the complaints that annoy users the most for each of the platforms on which their app is available.

Table 9 Negative impact ratio of complaint types in snapshots S1 and S2

Complaint type	Negative impact in Android	Rank in Android	Negative impact in iOS	Rank in iOS	Ratio[a] of negative impact ratio
Snapshot S1:					
App Crashing	2.1	8	3.4	6	0.6
Compatibility	2.0	10	2.8	11	0.7
Feature removal	1.7	13	3.2	7	0.5
Feature request	2.4	7	2.8	12	0.9
Functional error	2.0	9	2.9	9	0.7
Hidden cost	4.7	2	3.9	4	1.2
Interface design	1.2	14	2.0	14	0.6
Network problem	2.4	6	2.8	10	0.8
Privacy and ethical	5.5	1	4.9	2	1.1
Resource heavy	1.7	11	4.6	3	0.4
Uninteresting content	2.8	4	2.9	8	1.0
Unresponsive app	1.7	12	2.5	13	0.7
Not specific	3.5	3	6.6	1	0.5
All categories	2.6	5	3.5	5	0.7
Snapshot S2:					
App crashing	2.1	4	3.3	5	0.6
Compatibility	2.0	6	3.2	7	0.6
Feature removal	2.0	8	2.2	13	0.9
Feature request	1.1	14	1.7	14	0.7
Functional error	1.6	11	2.7	10	0.6
Hidden cost	4.2	3	17.0	1	0.2
Interface design	1.2	13	2.2	12	0.5
Network problem	5.2	1	2.9	8	1.8
Privacy and ethical	4.8	2	5.5	2	0.9
Resource heavy	2.0	7	3.5	4	0.6
Uninteresting content	2.1	5	2.6	11	0.8
Unresponsive app	1.6	10	3.2	6	0.5
Not specific	1.3	12	5.0	3	0.3
All categories	1.7	9	2.9	9	0.5

[a]Ratios in this and following tables are calculated by Android / iOS

Users have higher expectations on one platform than on the other for some apps. In snapshot S1, we find that for Netflix, its ratios of negative impact ratios for all complaint types are greater than 1. In other words, Android users tend to give more 1-star ratings than 2-star ratings for any type of issues for Netflix. We also find that for Hulu, Messenger and WhatsApp, the ratio of negative impact ratio is always lower than 1 except for one complaint type. In snapshot S2, we find that Hulu, Messenger and WhatsApp have most ratios of negative impact ratio lower than 1 except for one complaint type. In S2, we find

Table 10 Ratio of negative impact ratios for cross-platform apps in snapshots S1 and S2

App name	Ratio of negative impact ratios												
Compliant type #	1	2	3	4	5	6	7	8	9	10	11	12	13
Snapshot S1:													
Amazon	0.3	–	1.0	2.2	1.0	–	1.7	–	0.3	–	0.5	0.3	1.1
Facebook	1.6	2.5	–	3.3	1.3	–	–	–	–	0.9	1.3	0.8	0.6
Hulu	**0.3**	**0.9**	**0.6**	–	**0.7**	–	–	**0.6**	**0.8**	–	**0.9**	**0.5**	**2.1**
Instagram	–	–	–	**12.0**	0.4	–	–	–	–	–	–	–	–
Kik	–	1.0	–	1.1	1.4	–	–	1.0	–	–	–	0.3	–
Madden NFL	0.6	0.2	–	2.0	0.4	–	–	–	1.1	–	1.2	1.6	0.6
McDonald's	0.3	0.4	–	1.5	1.1	–	–	–	0.6	0.1	0.8	0.9	0.9
Messenger	**0.9**	**0.5**	**0.1**	**0.7**	**0.8**	–	–	–	**1.4**	**0.5**	**0.4**	**0.4**	**0.4**
Netflix	**1.4**	**2.4**	–	**1.3**	**1.6**	–	–	**1.5**	**3.0**	–	**4.1**	**2.9**	**1.2**
Pandora	0.2	–	0.6	–	1.3	–	–	–	–	–	–	–	–
Pinterest	2.0	–	0.4	–	0.7	–	0.5	–	1.5	–	1.3	0.5	–
Pop the Lock	–	–	11.2	0.5	–	–	–	–	–	–	2.3	1.3	0.9
Skype	0.6	0.4	–	0.7	1.0	–	0.4	–	–	0.1	0.5	1.1	–
Snapchat	0.4	0.8	–	0.7	0.7	0.3	–	1.8	–	–	–	0.7	**2.5**
SoundCloud	0.6	1.5	–	0.7	0.6	–	–	–	–	–	–	–	–
Spotify Music	0.1	–	0.3	1.0	0.4	–	–	–	8.0	–	–	–	–
Subway Surfers	–	–	–	–	–	–	–	–	–	–	–	0.6	1.5
Twitter	–	–	1.0	0.2	1.2	–	0.2	–	4.0	–	–	–	2.3
WhatsApp	**0.4**	–	–	**0.2**	**0.3**	–	–	–	**1.0**	–	–	–	**0.6**
Snapshot S2:													
Amazon	0.2	1.6	–	0.7	0.5	–	–	–	2.6	0.5	2.7	–	0.3
Facebook	**4.8**	**9.5**	–	**0.3**	**2.5**	–	–	–	–	–	**1.0**	**24.0**	**1.2**
Hulu	**1.7**	–	**0.8**	–	**1.1**	–	–	**1.8**	–	–	**2.7**	**1.3**	**0.2**
Instagram	3.0	–	–	0.6	0.3	–	3.0	–	–	–	–	–	–
Kik	–	–	–	3.9	0.4	–	–	–	0.3	–	–	–	–
Madden NFL	**1.9**	**2.0**	–	–	**1.1**	**0.9**	–	–	–	–	**0.9**	–	**0.8**
McDonald's	–	–	–	–	0.3	–	–	–	–	–	–	–	–
Messenger	**0.7**	**0.6**	**5.5**	–	**2.5**	–	**0.3**	**4.6**	**0.8**	**1.8**	**0.3**	**0.8**	**0.3**
Netflix	–	1.1	–	–	0.3	–	–	0.1	–	–	0.9	–	0.7
Pandora	0.9	1.0	11.0	–	–	–	–	–	–	–	2.9	–	1.5
Pinterest	1.3	–	0.3	–	0.2	–	–	–	–	–	0.6	–	2.2
Pop the Lock	–	–	**1.9**	–	–	–	–	–	–	–	–	–	–
Skype	**0.1**	–	–	**1.3**	**0.8**	–	–	**2.6**	**0.2**	–	**0.2**	–	**0.2**
Snapchat	**0.4**	**0.4**	**0.3**	**0.4**	**0.2**	–	–	–	**1.0**	**1.6**	–	**0.2**	**0.0**
SoundCloud	–	–	–	–	–	–	–	–	–	–	–	–	–
Spotify Music	–	–	–	–	**0.5**	–	–	–	–	–	–	–	–
Subway Surfers	–	–	–	–	0.1	–	–	–	–	–	–	–	1.9
Twitter	0.9	–	0.5	–	0.2	–	0.7	–	–	–	–	3.2	0.9
WhatsApp	–	–	–	–	**1.9**	–	–	–	–	–	–	–	**0.7**

"–" represents the number of complaints is 0 in one or both platforms

Entries in bold are discussed in the paper

that there are four complaint types for Netflix whose ratios of negative impact ratio are smaller than 1. The differences between snapshots indicate that the ratio of negative impact ratio can change over time.

Having a larger ratio of negative impact ratios for most complaint types on a single platform implies that users believe that every type of issue of the app is more severe and critical, which eventually suggests that users on this platform have a higher expectation on the quality of the app. In order to achieve consistent star ratings and user reviews for apps for which users have a particularly high expectation on one platform, developers need to spend more time and effort in solving user-raised issues on this specific platform or to identify new strategies to properly meet the high expectations of users.

5 Discussion

In this section, we first discuss the implications of our findings. Second, we discuss the influence of the preferred platform of a cross-platform app on the star ratings and user reviews. We then discuss the disadvantages of existing automated approaches in tagging reviews, to motivate our manual tagging process. We also discuss using the most recent 500 reviews as the source of our data. Finally, we discuss the impact of our sampling approach on our results.

5.1 Implications of our Findings

App users across platforms complain about different issues. In Section 4.2 we observed that for the same app, users across platforms complain about different issues. In addition, we observed in Section 4.3 that users across platforms assign different (negative) value to the same complaint types. One possible explanation is that behavioral consistency is not achieved across platforms, despite the efforts of the developers (Joorabchi et al. 2013, 2015). Another possible explanation is that users of different platforms have different expectations or interests. For example, iOS users may be more interested in technology (Benenson et al. 2013). Hence, they may expect that app developers follow up quickly with new functionality of the latest iPhone, as demonstrated by the following review: *"Great cross platform messaging app, but it needs improvements such as: 1) TouchID support, 2) Message recall and 3) Message expiration."* While there exists work on the differences between iOS and Android users (Benenson et al. 2013; Shaw et al. 2016), larger and more thorough studies are necessary to draw definitive conclusions. In particular, studies are needed to identify whether the expectations and interests of iOS and Android users are changing over the years. For example, Benenson et al. (2013) find that Android users are more privacy-aware. Our findings appear to contradict this, as in Section 4.2 we find that iOS users complain more often about privacy and ethical issues.

Cross-platform apps appear to have more compatibility issues on iOS than on Android. In Section 4.2, we observed that users of the iOS version of a cross-platform app complain more about compatibility issues than users of the Android version. As there exist more than 24,000 different Android devices (Morani 2015) and less than a hundred different iOS devices (Patterson 2016), and as Apple has strict quality guidelines (Apple 2017) for publishing an app, it seems counterintuitive that iOS apps have more compatibility

issues than Android apps. In Section 4.2, we explained that reviews generally do not contain many details about the reason for the compatibility complaint. Instead, we give possible explanations for the higher rate of compatibility issues on iOS:

- **Android users may be more technologically literate.** Prior studies on the differences between Android and iOS users reported that Android users are more likely than iOS users to have a technical job (Hixon 2014), or preference for advanced technical features (Benenson et al. 2013). Hence, Android users may be more capable of correctly identifying specific types of issues themselves without complaining to the app developer.
- **The heavily heterogenous landscape of Android devices may promote better testing for compatibility.** Because of the large number of different Android devices, Android developers may test their apps on a larger set of different devices to ensure compatibility across devices. In addition, as Android is open source software, mobile phone vendors and ISPs may provide customized versions of the operating system to their users. As new Android updates need to be customized by these vendors and ISPs, it takes longer for an update to reach the user (Android Authority 2017; Raphael 2016). Therefore, Android developers may be more careful when updating their apps, as their userbase may run a large variety of different Android versions. In addition, as Android users are more likely to run an older version of the operating system on their devices, app developers have to worry less about unexpected situations due to evolving versions of the operating system (Syer et al. 2015).
- **The perceived quality of an app is dependent on the number of users.** Herraiz et al. (2011) showed that the number of defects that are reported for the Debian GNU/Linux distribution are limited by the number of installations. It is possible that the iOS versions of the apps that we studied have more users and therefore, more bugs are discovered. Hence, future studies should investigate how the number of complaints for an app is related to the number of downloads and/or active users.

Both ratings and reviews need to be considered when studying the consistency of users' overall impressions of a cross-platform app across platforms. As demonstrated in Section 4.2, cross-platform apps may receive different complaints across platforms, despite the receipt of consistent star ratings. Hence, developers need to study both the ratings and reviews of both app versions to understand the overall impression of users of their app across platforms.

5.2 Influence of the Preferred Platform of an App on Star Ratings and User Reviews

It is possible that cross-platform app developers have a preference for a particular platform over another. For example, an app might have a much larger user base for one of the platforms. As a result, more time and effort are spent on the preferred platform version of the app, which may result in an inconsistent overall impression of users and inconsistent feature sets across both platforms. The preferred platform may help explain the differences in star ratings and user reviews that we observed in this paper. For example, we are interested in whether apps on the preferred platform receive higher or lower star ratings and user reviews.

Table 11 Comparisons of the number of releases and installation size of cross-platform apps

AppName	Number of releases ratio[a]	Package size ratio[a]
Amazon	0.8	0.3
Facebook	1.4	0.3
Hulu	0.7	0.7
Instagram	0.8	0.4
Kik	1.1	1.5
Madden NFL	0.8	0.7
McDonald's	1.0	0.8
Messenger	1.1	0.2
Netflix	1.0	0.5
Pandora	1.0	0.1
Pinterest	1.0	0.6
Pop the Lock	1.0	0.2
Skype	1.8	0.4
Snapchat	1.1	0.4
SoundCloud	2.0	0.5
Spotify Music	1.2	0.3
Subway Surfers	1.3	0.6
Twitter	1.5	0.3
WhatsApp	0.7	0.4
All apps	1.1	0.4

[a]Ratios in this and following tables are calculated by Android / iOS

To verify the hypothesis that the preference for a platform can explain the inconsistency in star ratings and user reviews, we must identify whether an app has a preferred platform and which platform this is.

Unfortunately, information regarding the preferred platform of an app is hard to obtain. Almost all of the 19 studied cross-platform apps are developed by large companies such as Facebook. It is unlikely that developers from such companies would disclose such competitive information about the preferred platform of an app.

We examine the total number of releases for each platform in an effort to identify whether an app has a preferred platform with a much higher number of releases. Specifically, we collect the number of releases in 2015 and the installation size for the current version as of Jan 29, 2016 for cross-platform apps through APK4Fun (2016) and AppAnnie (2015). We present our findings in Table 11.

We find that using the number of releases or package size (i.e., size of the APK file or IPA file) alone does not help in identifying the preferred platform of an app. For example, Instagram receives identical average star ratings on Android and iOS, but the number of releases differs by 10. On the other hand, the average star rating for SoundCloud is 4.3 (Android) and 4.5 (iOS) in snapshot S1, 4.0 (Android) and 4.5 (iOS) in snapshot S2, while it has twice as many releases for Android. We also observe that the installation size of 18 out of 19 cross-platform apps is at least twice as large in iOS.

5.3 Automated Approaches in Tagging User Reviews

Recently, researchers have began to tag user reviews using automated approaches. Unfortunately, existing approaches for automated tagging are not very successful at tagging issues that are raised in reviews. McIlroy et al. (2016) tag app reviews automatically. Their automated tagging approach achieves 65% precision and 64% recall. The approach that is presented by Panichella et al. (2015) has a higher precision (75%) and recall (74%). However, we consider their review categories too broad. For example, they have four categories for user reviews: 1) feature request, 2) problem discovery 3) information seeking and 4) information giving. The category 'problem discovery' is too general for developers to fetch more detailed issues. The two example reviews that are classified as 'problem discovery' in Table VII of Panichella et al.'s paper provide a specific example. The first example review, *"App crashes when new power up notice pops up"* should be classified as 'app crashing' whereas the second example review, *"Please fix the syncing issues with the iPad app"* should be classified as 'network problems' and 'compatibility issues'. The above example reviews show that automated tagging is still limited. To achieve a better accuracy while providing specific issue information for developers, we decided to tag reviews manually. The manual process is considerably more resource intensive nevertheless we are much more confident in our observations over observations that are derived from automated approaches.

5.4 Using the Most Recent 500 Star Ratings and Reviews

We collect star ratings and user reviews at two snapshots and in each snapshot, we collect the most recent 500 star ratings and reviews. Martin et al. (2015) recommends researchers to avoid collecting star ratings and user reviews in a short time frame. We are aware of Martin et al.'s recommendation and therefore design our study using two snapshots. We also show that the exact values of a phenomenon derived from a type of user-provided information (e.g. star rating) of cross-platform app can change, and we focus on the overall trend in the two snapshots as reported in findings in Section 4. Also, our two snapshots are collected almost one year apart, so it is unlikely that our findings are formed by chance alone. Therefore, our approach for collecting star ratings and reviews is reasonable.

5.5 The Impact of the Sampling Approach on our Results

For RQ2 and RQ3, we analyze only the 1 & 2-star reviews of the studied cross-platform apps. In total, we manually tag 9,902 1 & 2-star reviews. On average, the number of analyzed 1 & 2-star reviews of a studied cross-platform app is 510, which is sufficient for analyzing what users are complaining about in reviews. Lastly, we use a Mann-Whitney U test to analyze the distribution of complaint types, and the Mann-Whitney U test does not have a minimum sample size requirement.

6 Threats to Validity

6.1 External Validity

The findings of our RQ1 highlight the risk of analyzing star ratings at a platform level, since such an analysis reveals limited information for developers to improve the star ratings of

their own cross-platform apps. Instead, developers should focus on comparing their cross-platform apps on an app versus app basis. While our specific findings might not generalize, our findings do highlight the existence of differences across platforms for the same app. Moreover, all our proposed techniques and our methodologies are general and can be used for any app.

6.1.1 Threats Due to our Selection of Cross-Platform Apps

The studied cross-platform apps are apps that exist in the top 50 apps charts in both Google Play Store and App Store. The number of studied apps is small compared to the number of available apps in the entire app stores. In addition, the app categories of the studied apps do not cover all categories that are available in app stores. It is possible that the studied cross-platform apps do not represent all cross-platform apps across all store categories. For example, we did not study apps from the Google Play store category 'Parenting'. However, the goal of our study is not to derive a wide ranging theory about cross-platform apps. We think that such a theory is not achievable and it would also vary between apps. The 19 studied cross-platform apps are selected by carefully going through the top 50 free apps in both the iOS App Store and Google Play Store. Future studies are necessary to find whether our findings apply to non-top apps. In addition, as our findings are based on a comparison between iOS and Android apps, our findings do not necessarily apply to single platform apps. We studied ratings and reviews of free cross-platform apps only. One of the main reasons for not studying paid cross-platform apps is that the pricing of an app is very likely to act as a major confounding factor. Developers and users of paid cross-platform apps may have different expectations and attitudes than developers and users of free apps. For example, Lim et al. (2015) showed that users from Canada are more likely to be influenced by the price of an app. Hence, our findings are valid for free apps only. In addition, we focused on the US versions of the stores. Future studies should address the differences of users' overall impression of cross-platform apps across different countries.

In our study, we did not distinguish between different versions of an app on a platform. Hence, our study may be biased by the versions that the reviews in our snapshot cover. For example, a snapshot may contain only reviews for a buggy version, which will bias our study. However, there are several major challenges that make a study of reviews from a specific version not feasible:

- Reviews for Android apps are not explicitly linked to a version of an app. In addition, because reviews can be posted at any time, we cannot safely assume that a review is for the latest available version of an app. Therefore, we cannot identify which version of an Android app is reviewed. At the time of collecting the S1 snapshot, the Apple store did not always explicitly mention the version of an app for a review. We have the version information only for snapshot S2.
- Versioning on Android and iOS is not necessarily parallel, as shown in Section 5.2. As explained by Joorabchi et al. (2013), different versions of a cross-platform app are usually developed by different teams and treated as different products. Hence, these teams do not necessarily follow the same release schedule, as different bugs and updates may be necessary across platforms. Therefore, we cannot compare apps across platforms on a version-to-version basis (since it is impossible in many cases to create such a version mapping across platforms).

The median number of studied iOS versions of an app in snapshot S2 is two. In addition, we studied two snapshots that were recorded almost a year apart. Hence, our results are at least not strongly biased by studying only one version of an app. However, future studies on larger periods of data (e.g., years) are necessary to further mitigate this threat.

6.1.2 Threats Due to Different Expectations of Users on Different Platforms

Different expectations on the overall impression of an app between Android and iOS users have been mentioned and discussed by various researchers and industry experts. For example, Benenson et al. (2013) finds that if users have a technical background, then they are more likely to have an Android phone. Benenson et al. also find that having an Android phone is positively correlated to being more security-aware. Interestingly, Schick claims that iOS app users are often richer than Android users, based on annual income (Schick 2014). The abovementioned examples show that different expectations on the overall impression by users of different platforms do exist to some extent, but the impact of such differences on our study is not clear. For instance, it is not clear whether the 'fact' that iOS app users are often richer than Android users makes iOS users more tolerant toward issues related to 'Hidden Cost'. Or, whether the fact that Android users being more security aware makes them more critical of security or privacy issues in an app. Moreover, completely understanding the differences in expectations is more complicated than understanding the overall impression of users of cross-platform apps. The target audience of our paper (i.e cross-platform app developers) is not able to change the expectation of users. However, such developers must adapt to such varying expectations in order to achieve better star ratings and more favorable user reviews. Thus, future studies should carefully study the differences in expectations of users across platforms.

6.1.3 Threats Due to Feature Inequalities of Cross-Platform Apps

The existence of feature inequalities of cross-platform apps is no secret among both mobile app users and researchers. Feature inequalities can be divided into two types based on the source of introduction: 1) feature inequalities that are introduced by developers and 2) feature inequalities that are introduced by the platform. We acknowledge the fact that feature inequalities that are introduced by developers may exist but we also recognize the efforts by developers who try to make the user experience as consistent as possible across platforms. Developers' efforts to make the user experience consistent across platforms are noted in Joorabchi et al. (2013) survey. For example, Joorabchi et al. find that developers would like their mobile apps to behave similarly across platforms and they highlight feature equality as a subset of the behavior consistency. This suggests that developers are actively avoiding feature inequalities that are introduced by themselves. On the other hand, feature inequalities that are introduced by platform may be difficult or impossible for cross-platform app developers to resolve. For example, Apple allows developers to use Touch ID, a fingerprint recognition feature, for their mobile apps. This feature is only available for iOS devices such as the iPhone. Another example are widgets in Android, which are part of the home screen customization. Developers use widgets to display a 'quick glance' of the status of their apps. Widgets are not available on iOS devices. Solving the feature inequality in the above example is unrealistic for developers. In short, feature inequalities do exist and they may have an impact on the results of our study. However, developers of the top cross-platform apps are actively minimizing feature inequalities.

6.2 Internal Validity

In this paper, we conducted a manual tagging of user complaints. In S1, we notice that 19.88% of the complaints in Android and 7.49% of the complaints in iOS are tagged as 'Not Specific'. In S2, we find 22.61% of the complaints in Android and 9.14% of the complaints in iOS are tagged as 'Not Specific'.

We find that there are three main reasons a review gets tagged as 'Not Specific': 1) insufficient information about the app itself, 2) use of foreign language and 3) no information given at all. We counted the number of words in 1 & 2-star reviews for the studied cross-platform apps and we find that on average, a 1 & 2-star review for the studied cross-platform app on Android has 22 words, while in iOS, the average number of words per review is 39. The smaller average number of words in reviews in Android may help to explain the rationale for Android having a larger number of reviews that are tagged 'Not Specific'. Nevertheless, as these non-specific complaints also reflect the characteristics of users with respect to platforms, we have not removed them from the results.

7 Conclusions

Analyzing the star ratings and reviews of cross-platform apps, i.e., mobile apps that are available on multiple platforms, provides app developers a unique insight of the overall impression of app users of their apps across different platforms. The majority of prior work on mobile apps is done from a developer's perspective or limits the app selection to one app store.

In this paper, we study the overall impression of users of cross-platform apps using two snapshots of star ratings and user reviews. The snapshots are collected almost one year apart. We analyze 34,258 star ratings collected for 19 cross-platform apps and discover that at least 79% of the cross-platform apps receive different distributions of star ratings across platforms. By manually examining 9,902 1 & 2-star reviews, we tag user reviews in 12 complaint types and analyze the frequency as well as the negative impact of such complaints on star ratings of complaint types. The most important findings of our work are:

1. In order to understand how users rate the quality of a cross-platform app, it does not suffice to only analyze the received star ratings received across all supported platforms.
2. At least 79% of the cross-platform apps in our study do not receive the same distribution of star ratings on Android and iOS.
3. Users have different complaints for the iOS and Android version of the same cross-platform app, even though that app may have received similar star ratings on both platforms.
4. For the same app, users on two platforms judge the severity of issues differently.

Our findings show that cross-platform apps are far from achieving consistent star ratings and user reviews across platforms. Consistency in star ratings and user reviews is far more complex than delivering the same high quality top apps. Developers must be aware that users of different platforms have different priorities and expectations. Hence, the main implication of our results is that while developers are treating cross-platform apps as separate projects already in many cases, as shown in Joorabchi et al.'s study (Joorabchi et al. 2013), they should consider adjusting development priorities and requirements to the differing desires of the users of each platform for which their app is available.

References

Akdeniz (2014) Google play crawler JAVA API. https://github.com/Akdeniz/google-play-crawler, (last visited: May 30, 2017)

Ali M, Joorabchi M, Mesbah A (2017) Same app, different app stores: a comparative study. In: Proceedings of the IEEE/ACM international conference on mobile software engineering and systems (mobilesoft). IEEE Computer Society, p 12

Android Authority (2017) Android 7 nougat update tracker. http://www.androidauthority.com/android-7-0-update-679175/, (last visited: May 30, 2017)

APK4Fun (2016) Version history of the app facebook in Android. http://www.apk4fun.com/history/2430/, (last visited: May 30, 2017)

AppAnnie (2015) U.s version of the top 50 apps chart of the app store. https://www.appannie.com/apps/ios/top/, (last visited: May 30, 2017)

Apple (2008) RSS Feed provided by apple for the app facebook. https://itunes.apple.com/us/rss/customerreviews/id=284882215/page=1/json, (last visited: May 30, 2017)

Apple (2016) If an app you installed unexpectedly quits, stops responding, or won't open. https://support.apple.com/en-nz/HT201398, (last visited: May 30, 2017)

Apple (2017) App store review guidelines. https://developer.apple.com/app-store/review/guidelines/, (last visited: May 30, 2017)

Bavota G, Linares-Vásquez M, Bernal-Cárdenas CE, Penta MD, Oliveto R, Poshyvanyk D (2015) The impact of API change- and fault-proneness on the user ratings of Android apps. IEEE Trans Softw Eng (TSE) 41(4):384–407

Benenson Z, Gassmann F, Reinfelder L (2013) Android and iOS users' differences concerning security and privacy. In: Extended abstracts on human factors in computing systems (CHI), pp 817–822

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res (JMLR) 3:993–1022

Chen Y, Xu H, Zhou Y, Zhu S (2013) Is this app safe for children?: a comparison study of maturity ratings on Android and iOS applications. In: Proceedings of the 22nd international conference on world wide web (WWW). ACM, New York, pp 201–212

Chen N, Lin J, Hoi SCH, Xiao X, Zhang B (2014) AR-Miner: mining informative reviews for developers from mobile app marketplace. In: Proceedings of the 36th international conference on software engineering (ICSE). ACM, New York, pp 767–778

Dalmasso I, Datta SK, Bonnet C, Nikaein N (2013) Survey, comparison and evaluation of cross platform mobile application development tools. In: 9th international wireless communications and mobile computing conference (IWCMC), pp 323–328

Dann J (2012) Under the hood: Rebuilding facebook for iOS. https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920/, (last visited: May 30, 2017)

Di Sorbo A, Panichella S, Alexandru CV, Shimagaki J, Visaggio CA, Canfora G, Gall HC (2016) What would users change in my app? Summarizing app reviews for recommending software changes. In: Proceedings of the 24th ACM SIGSOFT international symposium on foundations of software engineering (FSE), ACM, pp 499–510

Fu B, Lin J, Li L, Faloutsos C, Hong J, Sadeh N (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 (KDD). ACM, New York, pp 1276–1284

Graphpad Software (2015) Interpreting results: skewness and kurtosis. http://www.graphpad.com/guides/prism/6/statistics/index.htm?stat_skewness_and_kurtosis.htm, (last visited: May 30, 2017)

Guzman E, Maalej W (2014) How do users like this feature? a fine grained sentiment analysis of app reviews. In: 22nd international requirements engineering conference (RE). IEEE, pp 153–162

Harman M, Jia Y, Zhang Y (2012) App store mining and analysis: MSR for app stores. In: 9th working conference on mining software repositories (MSR). IEEE, pp 108–111

Hartmann G, Stead G, DeGani A (2011) Cross-platform mobile development. Mobile Learning Environment, Cambridge

Hassan S, Shang W, Hassan AE (2017) An empirical study of emergency updates for top Android mobile apps. Empir Softw Eng (EMSE) 22(1):505–546

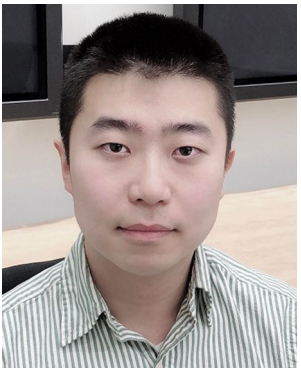
Heitkötter H, Hanschke S, Majchrzak TA (2013) Evaluating cross-platform development approaches for mobile applications. In: Web information systems and technologies, springer, pp 120–138

Herraiz I, Shihab E, Nguyen THD, Hassan AE (2011) Impact of installation counts on perceived quality: a case study on Debian. In: 18th working conference on reverse engineering (WCRE), pp 219–228

Hixon T (2014) What kind of person prefers an iphone? https://www.forbes.com/sites/toddhixon/2014/04/10/what-kind-of-person-prefers-an-iphone/, (last visited: May 30, 2017)

Howison J, Crowston K (2004) The perils and pitfalls of mining SourceForge. In: International workshop on mining software repositories (MSR), pp 7–11
Hu H, Bezemer CP, Hassan AE (2017) Supplementary material: Studying the consistency of star ratings and 1 & 2-star user reviews for top free cross-platform Android and iOS apps. http://sailhome.cs.queensu.ca/replication/cross-platform-mobile-apps/, (last visited: May 30, 2017)
Joorabchi M, Mesbah A, Kruchten P (2013) Real challenges in mobile app development. In: International symposium on empirical software engineering and measurement (ESEM). IEEE/ACM, pp 15–24
Joorabchi ME, Ali M, Mesbah A (2015) Detecting inconsistencies in multi-platform mobile apps. In: 26th international symposium on software reliability engineering (ISSRE). IEEE, pp 450–460
Kalliamvakou E, Gousios G, Blincoe K, Singer L, German DM, Damian D (2014) The promises and perils of mining GitHub. In: Proceedings of the 11th working conference on mining software repositories, MSR 2014, pp 92–101
Khalid H, Nagappan M, Shihab E, Hassan AE (2014) Prioritizing the devices to test your app on: A case study of Android game apps. In: Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering (FSE). ACM, pp 610–620
Khalid H, Shihab E, Nagappan M, Hassan AE (2015) What do mobile app users complain about? IEEE Softw 32(3):70–77
Kim HW, Lee HL, Son JE (2011) An exploratory study on the determinants of smartphone app purchase. In: The 11th international DSI and the 16th APDSI joint meeting, Taipei, Taiwan
Laurence G, Janessa R (2015) Gartner says smartphone sales surpassed one billion units in 2014. http://www.gartner.com/newsroom/id/2996817, (last visited: May 30, 2017)
Lim SL, Bentley PJ, Kanakam N, Ishikawa F, Honiden S (2015) Investigating country differences in mobile app user behavior and challenges for software engineering. IEEE Trans Softw Eng (TSE) 41(1):40–64
Linares-Vásquez M, Bavota G, Bernal-Cárdenas C, Di Penta M, Oliveto R, Poshyvanyk D (2013) API Change and fault proneness: a threat to the success of Android apps. In: Proceedings of the 9th joint meeting on foundations of software engineering (ESEC/FSE). ACM, pp 477–487
Long JD, Feng D, Cliff N (2003) Ordinal analysis of behavioral data. Wiley, New York
Martin W, Harman M, Jia Y, Sarro F, Zhang Y (2015) The app sampling problem for app store mining. In: Proceedings of the 12th working conference on mining software repositories, MSR '15. IEEE Press, Piscataway, pp 123–133
Martin W, Sarro F, Harman M (2016) Causal impact analysis for app releases in google play. In: Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering, FSE 2016. ACM, New York, pp 435–446. https://doi.org/10.1145/2950290.2950320
Martin W, Sarro F, Jia Y, Zhang Y, Harman M (2017) A survey of app store analysis for software engineering. IEEE Trans Softw Eng (TSE) PP(99):1–32
McIlroy S, Ali N, Khalid H, E Hassan A (2015) Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empir Softw Eng (EMSE) 21(3):1067–1106
McIlroy S, Ali N, Khalid H, E Hassan A (2016) Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empir Softw Eng (EMSE) 21(3):1067–1106
Mercado IT, Munaiah N, Meneely A (2016) The impact of cross-platform development approaches for mobile applications from the user's perspective. In: Proceedings of the international workshop on app market analytics (WAMA). ACM, pp 43–49
Morani L (2015) There are now more than 24,000 different Android devices. http://qz.com/472767/there-are-now-more-than-24000-different-Android-devices/, (last visited: May 30, 2017)
Mudambi SM, Schuff D (2010) What makes a helpful online review? a study of customer reviews on amazon.com. MIS Q 34(1):185–200
Nayebi M, Adams B, Ruhe G (2016) Release practices for mobile apps – what do users and developers think? In: 23Nd international conference on software analysis, evolution and reengineering (SANER), vol 1. IEEE, pp 552–562
Noei E, Syer MD, Zou Y, Hassan AE, Keivanloo I (2017) A study of the relation of mobile device attributes with the user-perceived quality of Android apps. Empir Softw Eng (EMSE) 22(6):3088–3116
Pagano D, Maalej W (2013) User feedback in the appstore: an empirical study. In: 21st international requirements engineering conference (RE). IEEE, pp 125–134
Palmieri M, Singh I, Cicchetti A (2012) Comparison of cross-platform mobile development tools. In: 16th international conference on intelligence in next generation networks (ICIN). IEEE, pp 179–186
Palomba F, Linares-Vásquez M, Bavota G, Oliveto R, Penta MD, Poshyvanyk D, Lucia AD (2015) User reviews matter! tracking crowdsourced reviews to support evolution of successful apps. In: 2015 IEEE international conference on software maintenance and evolution (ICSME), pp 291–300
Palomba F, Salza P, Ciurumelea A, Panichella S, Gall H, Ferrucci F, De Lucia A (2017) Recommending and localizing change requests for mobile apps based on user reviews. In: Proceedings of the 39th international conference on software engineering (ICSE). IEEE Press, Piscataway, pp 106–117

Panichella S, Sorbo AD, Guzman E, Visaggio CA, Canfora G, Gall HC (2015) How can I improve my app? Classifying user reviews for software maintenance and evolution. In: International conference on software maintenance and evolution (ICSME). IEEE, pp 281–290
Patterson B (2016) Blake's iOS device specifications grid. http://blakespot.com/ios_device_specifications_grid.html, (last visited: May 30, 2017)
Pettey C, Rob van der M (2012) Gartner says free apps will account for nearly 90 percent of total mobile app store downloads in 2012. http://www.gartner.com/newsroom/id/2153215, (last visited: May 30, 2017)
Porter MF (1997) An algorithm for suffix stripping. In: Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, pp 313–316
Poschenrieder M (2015) 77% Will not download a retail app rated lower than 3 stars. https://blog.testmunk.com/77-will-not-download-a-retail-app-rated-lower-than-3-stars/, (last visited: May 30, 2017)
Ramon L, Ryan R, Kathy N (2015a) Number of apps available in leading app stores as of july 2015. http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, (last visited: May 30, 2017)
Ramon L, Ryan R, Kathy N (2015b) Smartphone OS market share, 2015 q2. http://www.idc.com/prodserv/smartphone-os-market-share.jsp, (last visited: May 30, 2017)
Raphael J (2016) Android upgrade report card: grading the manufacturers on marshmallow. http://www.computerworld.com/article/3052937/android/android-upgrade-report-card-marshmallow.html, (last visited: May 30, 2017)
Romano J, Kromrey JD, Coraggio J, Skowronek J, Devine L (2006) Exploring methods for evaluating group differences on the NSSE and other surveys: are the t-test and Cohen's d indices the most appropriate choices. In: Annual meeting of the southern association for institutional research
Schick S (2014) Report: iOS app users are often richer than Android users. http://www.fiercewireless.com/developer/report-ios-app-users-are-often-richer-than-android-users, (last visited: May 30, 2017)
Shaw H, Ellis DA, Kendrick LR, Ziegler F, Wiseman R (2016) Predicting smartphone operating system from personality and individual differences. Cyberpsychol Behav Soc Netw 19(12):727–732
Syer MD, Nagappan M, Adams B, Hassan AE (2015) Studying the relationship between source code quality and mobile platform dependence. Software Quality Journal (SQJ) 23(3):485–508
Tian Y, Nagappan M, Lo D, Hassan AE (2015) What are the characteristics of high-rated apps? A case study on free Android applications. In: IEEE international conference on software maintenance and evolution (ICSME), pp 301–310
Villarroel L, Bavota G, Russo B, Oliveto R, Di Penta M (2016) Release planning of mobile apps based on user reviews. In: Proceedings of the 38th international conference on software engineering (ICSE). ACM, New York, pp 14–24
Yichuan M, Cuiyun G, Michael RL, Jiuchun J (2016) Experience report: understanding cross-platform app issues from user reviews. In: 27th international symposium on software reliability engineering (ISSRE). IEEE
Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng (TKDE) 12(3):372–390
Zhao WX, Jiang J, Weng J, He J, Lim EP, Yan H, Li X (2011) Comparing twitter and traditional media using topic models. Springer, Berlin, pp 338–349



Hanyang Hu is currently designing a machine learning-assisted software compliance system at BlackBerry QNX in Ottawa, Canada. His research interests include software defect prediction and text mining in a software context. Previously, he worked on reducing the false positives of static analysis tools in the BlackBerry Security R&D team. He studied computer science at the University of Waterloo and Queen's University in Canada, where he received his BCs (2015) and MSc (2017) respectively.

Cor-Paul Bezemer currently works as a postdoctoral research fellow in the Software Analysis and Intelligence Lab (SAIL) at Queen's University in Kingston, Canada. His research interests cover a wide variety of software engineering and performance engineering-related topics. His work has been published at premier software engineering venues such as the ESEC-FSE, ICSME and ICPE conferences and the TSE and EMSE journals. He is one of the vice-chairs of the SPEC research group on DevOps Performance. Before moving to Canada, he studied at Delft University of Technology in the Netherlands, where he received his BSc (2007), MSc (2009) and PhD (2014) degree in Computer Science. More about Cor-Paul can be read on his website: http://sailhome.cs.queensu.ca/~corpaul.

Ahmed E. Hassan is the Canada Research Chair (CRC) in Software Analytics, and the NSERC/BlackBerry Software Engineering Chair at the School of Computing at Queen's University, Canada. His research interests include mining software repositories, empirical software engineering, load testing, and log mining. Hassan received a PhD in Computer Science from the University of Waterloo. He spearheaded the creation of the Mining Software Repositories (MSR) conference and its research community. Hassan also serves on the editorial boards of IEEE Transactions on Software Engineering, Springer Journal of Empirical Software Engineering, Springer Journal of Computing, and PeerJ Computer Science. Contact ahmed@cs.queensu.ca. More information at: http://sail.cs.queensu.ca/