Empir Software Eng (2018) 23:3476–3502
https://doi.org/10.1007/s10664-018-9615-8

An empirical study of Android Wear user complaints

Suhaib Mujahid[1] · Giancarlo Sierra[1] · Rabe Abdalkareem[1] · Emad Shihab[1] · Weiyi Shang[2]

Published online: 23 March 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Wearable apps are becoming increasingly popular in recent years. Nevertheless, to date, very few studies have examined the issues that wearable apps face. Prior studies showed that user reviews contain a plethora of insights that can be used to understand quality issues and help developers build better quality mobile apps. Therefore, in this paper, we mine user reviews in order to understand the user complaints about wearable apps. We manually sample and categorize 2,667 reviews from 19 Android wearable apps. Additionally, we examine the replies posted by developers in response to user complaints. This allows us to determine the type of complaints that developers care about the most, and to identify problems that despite being important to users, do not receive a proper response from developers. Our findings indicate that the most frequent complaints are related to `Functional Errors`, `Cost`, and `Lack of Functionality`, whereas the most negatively impacting complaints are related to `Installation Problems`, `Device Compatibility`, and `Privacy & Ethical Issues`. We also find that developers

Communicated by: Romain Robbes

✉ Emad Shihab
eshihab@encs.concordia.ca

Suhaib Mujahid
s_mujahi@encs.concordia.ca

Giancarlo Sierra
g_sierr@encs.concordia.ca

Rabe Abdalkareem
rab_abdu@encs.concordia.ca

Weiyi Shang
shang@encs.concordia.ca

[1] Data-Driven Analysis of Software (DAS) Lab, Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada

[2] Software Engineering and System Engineering (SENSE) Lab, Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada

mostly reply to complaints related to `Privacy & Ethical Issues`, `Performance Issues`, and `notification` related issues. Furthermore, we observe that when developers reply, they tend to provide a solution, request more details, or let the user know that they are working on a solution. Lastly, we compare our findings on wearable apps with the study done by Khalid et al. (2015) on handheld devices. From this, we find that some complaint types that appear in handheld apps also appear in wearable apps; though wearable apps have unique issues related to `Lack of Functionality`, `Installation Problems`, `Connection & Sync`, `Spam Notifications`, and `Missing Notifications`. Our results highlight the issues that users of wearable apps face the most, and the issues to which developers should pay additional attention to due to their negative impact.

Keywords Wearable apps · Users' reviews · User complaints · Google Play Store · Empirical studies

1 Introduction

Mobile apps very popular and have been the focus of numerous studies in recent years (Nagappan and Shihab 2016; Martin et al. 2017). A fundamental change introduced by mobile apps is the way that they are released to users, which is through app stores. App stores allow users to directly provide feedback on the mobile apps through user reviews. Although these user reviews were meant to simply provide feedback about the apps, they proved to be much more useful (Harman et al. 2012; Galvis Carreño and Winbladh 2013; Pagano and Maalej 2013). For example, studies have shown that they can be used to prioritize devices to test (Khalid et al. 2014), prioritize feature improvements (Keertipati et al. 2016), and/or can be used to understand user problems so that developers can avoid low ratings, which can have a major impact on the app's user base, revenues and the success of the app in general (Finkelstein et al. 2017; Di Sorbo et al. 2016; Guzman and Maalej 2014).

Recently, wearables that complement handheld devices were introduced. Wearable devices i.e., smart watches and fitness trackers, are becoming increasingly popular and are expected to reach 101 million devices by 2020 (Chauhan et al. 2016). Wearable devices have unique characteristics that pose challenges when compared to other platforms or devices (Rawassizadeh et al. 2015). These devices provide their developers with access to a diverse set of sensors and features (e.g., physiological, biochemical, as well as motion sensing (Bonato 2010; Teng et al. 2008)) that can be used to enhance the user experience (Android Developers Documentation 2016a). As such, developers began to develop apps that are specifically designed to run on these wearable devices, called wearable apps. Wearable apps are different than handheld apps that run on mobile phones (Wright and Keith 2014), since they: 1) are often very lightweight (resource wise) (Park and Jayaraman 2003), 2) are meant to run on very small screens (Tehrani and Michael 2014), 3) have access to a different set of sensors (Do et al. 2017), and 4) heavily depend on a mobile device to perform most of the expensive processing (Chauhan et al. 2016; Wei 2014). To the best of our knowledge, very few studies have focused on wearable apps and their user reviews to this date.

Therefore, similar to the prior studies on (handheld) mobile app reviews (Khalid et al. 2015; Ha and Wagner 2013; Hoon et al. 2012; Vasa et al. 2012), we also investigate user complaints, however, we focus on reviews from wearable apps. Although the main goal of

this study is not to surface differences between complaint types from handheld and wearable apps, as we will show later, wearable apps share some common complaints and have their own unique complaints when compared with handheld apps.

To perform our study, we manually classify 2,667 reviews belonging to 19 wearable apps. The reviews were tagged by the first two authors of the paper and grouped into 15 different categories. For each category, we measured the frequency of the complaints and how negatively they are perceived by users. We measure this negative perception based on how low users rate complaints of a certain category. Since this negative perception is reflected into low user ratings, we rank the impact of each complaint category based on the ratio of 1-star rated reviews to 2-star rated reviews.

We also examine the developer replies to these complaints in order to better understand the areas that receive enough attention and areas that are important to the users, but not well attended by the developers. Our study concerns two main areas: I) examining user complaints and II) examining developer replies. For each area, we ask two research questions:

I.1 What do Wearable App Users Complain About?

Our findings indicate that `Functional Errors`, `Cost`, and `Lack of Functionality` are the three most frequent complaints.

I.2 What User Complaints are Most Negatively Impacting?

We find that `Installation Problems`, `Device Compatibility`, and `Privacy & Ethical Issues` are the most negatively perceived by users. Users that encounter `Installation Problems` of wearable apps are five times more likely to give a 1-star review than a 2-star review.

Our findings provide insight to the developer and research community as to what issues wearable app users face the most and which issues are most impactful.

II.1 What Types of Complaints Do Developers Reply To?

In addition, we also examined the developer replies to the user complaints. We find that developers are most likely to reply to complaints related to `Privacy & Ethical Issues`, `Performance Issues`, and `Spam Notifications`. We also contrast the complaints based on their impact and the developer replies and find that `Installation Problems`, `Device Compatibility`, and `Connection & Sync Issues` are most impacting, but have a low response rate from developers.

II.2 How Do Developers Reply to Complaints?

We find that when developers reply to user complaints, they often try to get more information or provide potential workarounds to solve the complaints.

Our results highlight areas that are of high importance to the users, but are not well addressed by the developers, and vice versa.

In addition, we compare our findings to the handheld user complains reported by Khalid et al. (2015). Our findings show that 10 of the 15 categories found in our research are common to both handheld and wearable apps, however, 5 of the complaint types are unique to wearable apps, namely—`Lack of Functionality`, `Connection & Sync`, `Spam Notifications`, `Missing Notifications`, and `Installation problem`. Moreover, we find that similar to their findings, approximately 12% of the complaints occur after an update. Our findings show that there is a need to ensure regression tests

are performed before wearable apps are updated. Furthermore, to enable future research and enable the replication of this work, we make our dataset publicly available.[1]

This paper extends and supplements our previous short paper (Mujahid et al. 2017), in which we manually analyzed 589 user reviews from 6 wearable apps and studied the types of issues that users complain about. Similar to our earlier work in Mujahid et al. (2017), we follow the study design and approach of Khalid et al. (2015), which is the closest study to ours since it studies the complaints from user reviews in the handheld app domain. This also allows us to perform a brief comparison of results with the findings of the user complaints in the mobile domain. In this paper, we extend our previous work by conducting a comprehensive study on user reviews of wearable apps that include: 1) 13 additional wearable apps from which we examine 2,667 user reviews that allow us to generalize our finding in the previous short paper; 2) we investigate the most impactful types of user complaints; 3) we further study the developer replies to different types of user complaints; and 4) we introduce a comparison between the types of complaints in wearable app and handheld mobile apps.

The rest of the paper is organized as follows. Section 2 presents and compares related work. Section 3 details our study design, including our collection and selection methodology. Section 4 and Section 5 present and discuss our results. Section 6 discusses the threats to validity of our study. Section 7 concludes the paper and outlines areas for future work.

2 Related Work

The work that is most related to our study falls into two main categories: work that leveraged mobile user reviews and work focusing on wearable apps.

2.1 Work Leveraging Mobile User Reviews

One of the first studies to leverage mobile app reviews was done by Harman et al. (2012) in 2012. In their paper, the authors studied the correlation of user reviews with key performance metrics such as the number of downloads. They found that there is a strong correlation between app ratings and its rank based on the number of downloads, suggesting that developers should pay close attention to their user ratings. More recently, Finkelstein et al. (2017) extended the work by Harman et al. (2012), which mined data from the Blackberry World App Store to analyze the correlation between: the customer rating of an app from user reviews, its price, popularity (based on downloads), and claimed features that extracted from each app's description with natural language processing (NLP) techniques. The authors found that there is a strong correlation between the customer rating of an app and its popularity, and a moderate correlation between price and the claimed features of an app.

Other studies mined user reviews to better understand the contents of these user reviews. Khalid et al. (2015) studied low-rated user reviews from 20 free iOS apps in order to help developers understand their nature. They exposed 12 types of complaints and found that feature requests, functional errors and, crashing apps were the most frequent reasons for negative reviews, while privacy and ethical concerns corresponded to the most impactful reviews that mostly lowered the rating of an app. Ha and Wagner (2013) manually analyzed the user reviews of 59 Android apps to examine the impact of privacy and ethical issues.

[1] https://github.com/suhaibtamimi/user_complaints_of_wearable_apps_dataset

They found that only around 1% of the apps contain complaints related to privacy and ethical issues. Hoon et al. (2012) and Vasa et al. (2012), reviewed the vocabulary of 8.7 million user reviews from the Apple App Store showing a link between the length of a review and its given rating.

In other work, Fu et al. (2013) automated the analysis of over 13 million reviews of more than a hundred thousand apps in the Google Play Store using Latent Dirichlet Allocation model (LDA). They uncovered 10 unique topics corresponding to user complaints; they also found that there is a significant difference between free and paid apps because paid apps often present a complaint topic of the involved pricing, absent in the user reviews of free ones. Similarly, Chen et al. (2014) created a framework to automatically extract the most informative reviews from a data set of mobile apps using NLP techniques. They found that frequently, the amount of reviews for an app can be too large for human reading or understanding, and that only 35.1% of the reviews actually contain valuable information that developers could use for app improvement. Therefore, their framework automates an approach to filter, group, rank and visualize the informative portions of the reviews only.

Other work by McIlroy et al. (2016) found that up to 30% of mobile app reviews can contain multiple topics of information and proposed an automated approach for labeling the user reviews, which reached a precision of 66% and 65% of recall while classifying them in 13 different categories.

Regarding categorization of developer replies to the low scored user reviews, a recent study by McIlroy et al. (2015) introduced the benefits of responding to app reviews, indicating that following a response users would increase the review rating 38.7% of the time by 20% of the previous score.

Panichella et al. (2015) studied the structure, sentiment and text features of mobile app reviews and proposed: 1) a taxonomy of 4 categories related to software maintenance and evolution tasks in which to classify app user reviews; and 2) an approach to automatically classify them using NLP, text analysis and sentiment analysis techniques. The authors combined these techniques using machine learning and empirically evaluated their classifiers, showing that their approach can aid developers to obtain the intention from user reviews. Later, Panichella et al. extended their work and implemented their approach from Panichella et al. (2015) in a tool named ARdoc (Panichella et al. 2016) that automates the classification of user reviews. The performance of the tool was validated by the developers of 3 real-world mobile apps and an external software engineer. ARdoc achieved promising results with precision, recall and F-Measure values ranging between 84% to 89%.

Di Sorbo et al. (2016) introduced a model to obtain the topics contained in user reviews from mobile apps, which they call URM (User Reviews Model). The model was combined with the approach presented in Panichella et al. (2015) to capture the intentions of user reviews in a new approach named SURF (Summarizer of user reviews Feedback). SURF generates summaries from sets of user reviews and clusters them considering both, the intention and topics found in user reviews to recommend software changes. The usefulness of this approach was validated on 17 mobile apps by 23 developers and researchers. As a follow-up, Di Sorbo et al. implemented and validated SURF as a tool to automate the processing of user reviews for developers (Di Sorbo et al. 2017).

More recently, Ciurumelea et al. (2017) manually analyzed 1,566 user reviews from 39 mobile apps and defined a multi-level taxonomy that is specific to the mobile domain. The authors introduced an approach, called URR (User Request Referencer) that not only automatically classifies user reviews in their multi-level taxonomy, but also points developers to the artifacts that need to be modified to address a particular user review. They showed that doing so reduces the time it takes to process user reviews manually by up to 75%.

With another perspective, Palomba et al. (2017) presented CHANGEADVISOR, an approach that clusters multiple user reviews that contain change requests to recommend developers which artifacts to modify in a mobile app to address user feedback. This approach uses NLP and clustering techniques to sort reviews based on their content, semantics and structure. A validation conducted with the developers of 10 mobile apps highlighted the usefulness of this approach when mining large numbers of user reviews, providing 81% of precision and 70% recall when recommending changes.

There are also a plethora of other works on mobile apps, that leverage users reviews for their techniques. In this section, we only discuss the most relevant studies, however, we refer the reader to a recent survey by Martin et al. (2017) for a more comprehensive list of studies on mobile apps.

Our work differs from the prior work since 1) we focus on the user reviews of wearable apps, 2) we triangulate two data sources user reviews and developers replies to understand the types of user complaints that developers care about.

2.2 Work Focusing on Wearable Apps

Very few studies have focused on the study of wearable apps, but many different paths are beginning to get explored in the domain. In our previous work (Mujahid et al. 2017), we studied the user complaints of wearable apps by analyzing 589 reviews from 6 Android wearable apps. Our main findings indicate that users complain mostly about `Functional Errors`, `Lack of Functionality`, and `Cost` of wearable apps.

Recently, Zhang and Rountev (2017) presented a formal semantics to statically model the notification mechanism of Android Wear, and contributed with the development of two domain-specific tools, one for test case execution and another for automated test generation. Ahola (2015) exposed three issues and limitations in Android Wear platform found during wearable app development that are better wear Internet connectivity, virtual button support for watch faces, and software configurable language support for voice input. On a different perspective, Lyons (2015) did a study on the user perceptions of functionality and design of smartwatches, including android wearable devices. Based on user feedback and contrast to traditional watches, possible features for future wearable app are suggested. Min et al. (2015) explored the battery usage of wearable apps and performed an online survey to get direct feedback and concerns from users. They found that most users do not complain about the battery usage of their wearable devices. Chauhan et al. (2016) did a previous categorization of smart watch apps from Samsung, Apple, and Android Wear. They used data from Android Wear Centre (2016) and GoKo (2016) as sources to get the wearable app identifiers for crawling their information; we applied the same approach to initialize our crawling phase.

Our work differs from prior work on wearable apps since, to the best of our knowledge, this study is the first comprehensive study that analyze wearable user complaints in depth. Moreover, we differ from previous work since we do not only investigate the complaint types, frequency and impact of low rated user reviews; but we also contrast our findings to similar ones in the domain of handheld apps. By doing this comparison, we are able to bring several implications for wearable app development into the community spotlight.

3 Study Design

The goal of our study is: 1) to determine the most frequent and negatively impacting user complaints of wearable apps and 2) to investigate the type of complaints that developers

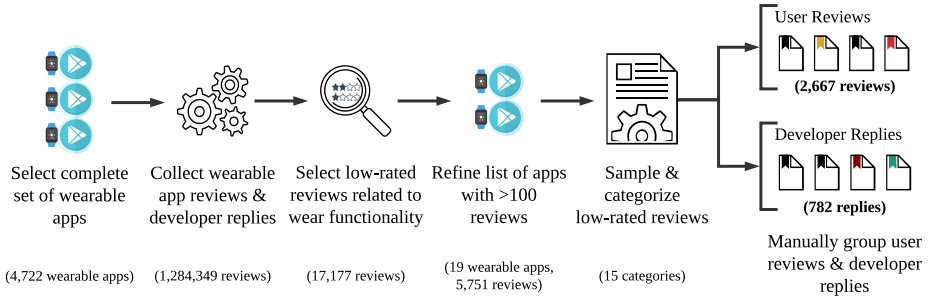


Fig. 1 Overview of the user review and developer reply classification process

reply to and the reply types. To do so, we mine the Google Play Store for the reviews of wear apps. Figure 1 provides an overview of our approach. In the following sections we describe our data collection and selection, as well as detail our manual classification of user reviews.

3.1 Data Collection and Selection

For the purpose of our study, we select a number of wearable apps that have negative user reviews. First, we obtained the available Android Wear apps on Google Play Store by collecting their identifiers from two alternative app markets: Android Wear Center (2016) and GoKo (2016). The two aforementioned sources have been used in prior work focusing on wearable apps (Chauhan et al. 2016). Then, we mined the wearable apps using a data scrapper that we wrote. The scrapper collected various information about each wear apps, including: the user reviews' text, its rating, the developer's reply to the review, if any, and the apps' overall rating. To enhance performance of the scrapper, it was deployed on a cluster of machines in order to distribute the requests sent.

In total, we mined the data of 4,722 wearable apps from 2,732 unique developers, which contained 1,284,349 user reviews. From the total number of mined apps, we found that 1,017 app did not contain any user review at all, i.e., 21.5% of apps. Since we are interested in wearable apps' user complaints, we selected only low-rated reviews that related to wearable apps (i.e., 1 and 2 stars rating). This was done following a prior study by Khalid et al. (2015), with the rationale that low-rated reviews are most likely to contain user complaints. We also noted, that 1,958 apps did not contain any 1 or 2 star rated user reviews, i.e., 41.5% of apps. Considering we need a reasonable amount of data to perform our analysis, we only selected apps with over 100 low-rated reviews. This left us with 5,751 low-rated user reviews from 19 wearable apps. Note that we include data from all available releases (up to the collection date) of the 19 wearable apps.

To ensure that we only examine wearable-related complaints, we discriminate between two type of apps: 1) apps that have their main functionality related to the wearable version e.g., watch faces. For these types of apps, we include all the low-rated reviews in the complaint data set since all of them are related to a wearable functionality; and 2) apps that have a full handheld app and an accompanying wearable version. For these apps, we only include reviews that contain keywords related to the wearable app i.e., '*wear*', '*watch*', and/or '*wrist*' and their variations. These keywords were selected based on a manual examination of wearable apps reviews. Due to our selection criteria, all apps in our study belong to the first category (i.e., apps whose main functionality is related to a wearable version).

Table 1 Statistics of studied Android Wearable Apps

Wear App name	Low rated reviews	Sampled reviews	Developers replies	Date span of reviews
ZenWatch Manager	201	132	40	26/11/2014–06/10/2016
WatchMaker Premium Watch Face	501	218	22	31/03/2015–07/10/2016
Odyssey Watch Face	125	94	20	10/12/2014–21/08/2016
Skymaster Pilot Watch Face	152	109	20	02/11/2014–28/09/2016
Ranger Military Watch Face	163	115	23	29/12/2014–20/09/2016
Wear Mini Launcher	133	99	44	21/08/2014–05/10/2016
Wear Face Collection	136	101	21	18/07/2014–20/09/2016
InstaWeather for Android Wear	141	103	61	18/12/2014–30/09/2016
Motorola Connect	213	137	20	07/09/2014–06/10/2016
Watch Faces for Android Wear	154	110	10	25/10/2014–30/09/2016
Facer Watch Faces Android Wear	926	272	134	01/08/2014–07/10/2016
Bits Watch Face	116	89	59	20/08/2015–01/10/2016
WatchMaster - Watch Face	124	94	85	24/07/2015–12/10/2016
Luxury Watch Faces for Wear	115	89	79	02/09/2014–07/10/2016
Android Wear - Smartwatch	1,531	307	79	29/09/2015–08/10/2016
Weather Watch Face	279	162	32	20/07/2014–21/09/2016
Web Browser for Android Wear	142	104	33	23/07/2014–04/10/2016
Plants vs. Zombies Watch Face	369	188	0	03/01/2015–30/09/2016
LG Call for Android Wear	230	144	0	28/04/2015–19/09/2016
Total	5,751	2,667	782	–

Since this is the first study to examine user complaints for wearable apps (in addition to our preliminary short study), we opt to perform our analysis of the user complaints manually. Given that this manual classification is a time and resource intensive task, we selected a random statistically representative sample of complaints from each wearable app. The sample sizes were selected to attain a 5% confidence interval and a 95% confidence level in the population being sampled. This random sampling process resulted in 2,667 total reviews varying from 89 to 307 reviews per app. The list of the studied wearable apps, number of low related reviews, the number of examined reviews, number of developers' replies, and date span of reviews are shown in Table 1. Our data was collected between 6th to 13th of October 2016.

3.2 Manual Classification of User Reviews

To perform our manual classification of user reviews, we need to come up with an initial set of categories that the reviews can be grouped into. To do so, once we obtained all of the reviews, we took a statistical significant random sample of 597 reviews from all the selected apps.[2] We manually inspected and classified the sampled reviews twice, (once by each of

[2]The random sample of 597 reviews was taken out of 5,751 low-rated reviews to achieve a confidence level of 99% and a confidence interval of 5%.

the first two authors of the paper) into different categories using an open coding approach (Seaman 1999; Usman et al. 2017).

Both classifications were done individually and independently. Each of the two classifiers classified the review into a certain category based on its content. Disagreements between the two classifiers were clarified through discussion. For both authors the categories were defined by the first half of the sampled reviews. By the end of this step, the authors defined 15 different initial categories. Note that throughout the paper we also refer to these categories as complaint types.

Once we defined the initial 15 complaint types, we proceeded to categorize our set of user reviews composed by samples of each studied wearable app (in total 2,667 reviews). To facilitate the categorization of the reviews, we built a web-based tool (shown in Fig. 2) that presented for each of the two people categorizing the review with all the review details and the respective developer reply, if a developer posted a reply to the review. The tool also has the option to add a new category in case a review belonged to a category that was not in our initial set. However, even though the tool had the option to add a new category in case a review belongs to a category that was not in our initial set, the authors did not come up with any new categories. Every review was tagged with all suitable categories, i.e., one review can have one or multiple tags based on its content. For example: if a user complaint mentions a battery drainage problem and also a connection issue, the review will be classified with the `Connection & Sync Issues` and `Battery Drainage` tags. In some instances, the user provided uninformative content in the review (e.g., "*Just nonsense, I hated this game...*"), in which case we put them in the 'Uninformative' category. The process to categorize all the user reviews took approximately 115 h in total.

As with any other human activity, there may be some disagreements when classifying the user reviews, and therefore, we applied a Cohen's Kappa to measure the level of agreement between the two individual classifications (Cohen 1960). The Cohen's Kappa coefficient has been commonly used to evaluate inter-rater agreement level for categorical scales, and provides the proportion of agreement corrected for chance. The resulting coefficient is scaled to range between -1 and $+1$, where a negative value means poorer than chance agreement, zero indicates exactly chance agreement, and a positive value indicates better than chance agreement (Fleiss and Cohen 1973). The closer the value is to $+1$, the stronger the agreement.

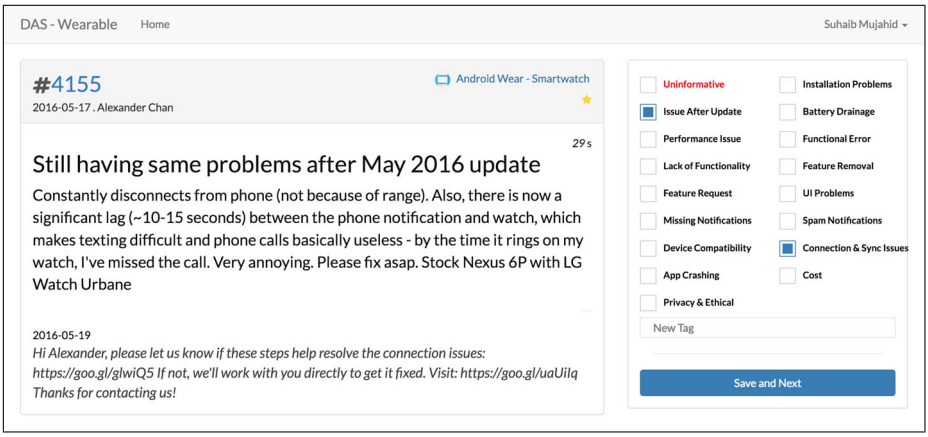


Fig. 2 Web-based tool for classifying wearable app user reviews

The level of agreement was +0.68, which is considered to be fair to good agreement (Fleiss and Cohen 1973). Out of the 2,667 classified reviews, 1,429 reviews had *full agreement* (i.e., both classifiers had the same selection while tagging the reviews). The remaining 1,238 reviews had a conflict in the classification, and for 710 reviews of them, the classification was a match for one or more of the categories but different in other(s). We examined all the reviews with a classification conflict and did a post agreement on the tags for those reviews; for example, if the first classifier tagged the review as `Device Compatibility` and the second one tagged it as `Functional Error`, we flag this review for discussion. The two classifying authors present their case as to why they classify a review in a certain category and reached agreement; this scenario solved most conflicts. When both authors could not agree, the third author was consulted to break the tie and reach a final classification.

4 Results

Once all the reviews in our dataset are categorized into the different complaint types, we proceed to answer our research questions, pertaining to two areas: user complaints and developer replies. In particular, we are interested in knowing what users complain about and what complaints tend to have the most negative impact. As for the developer replies, we examine the complaints that developers reply to and how they reply to them.

4.1 What Do Wearable App Users Complain About?

Since wearable apps are an emerging trend, our goal is to understand the types of user complaints so that developers can anticipate potential problems and plan their quality assurance efforts accordingly. Similar to prior studies on user complaints for handheld device apps (Khalid et al. 2015), we start by examining the different types of complaints based on the low-rated reviews of wearable apps.

To come up with the different complaint types, we manually categorized the wearable app reviews as mentioned earlier in Section 3.2. We then rank the various complaint types based on their frequency in the examined reviews.

Table 2 shows the 15 complaint types that we discovered from the wearable app reviews. For each category, we provide a brief description, an example review, and the percentage of reviews in each complaint type. It is important to note that one user review can present more than one complaint, hence, it can be mapped to more than one complaint type. Thus the percentage of reviews may sum up to more than 100%. From the table, we observe that many of the complaint types are related to the features provided by the wearable apps (e.g., `Feature Removal`, `Feature Request`), the behavior of the wearable apps (e.g., `App Crashing`, `Notifications`, `Battery Drainage`), and external factors (e.g., the `Cost` of the app, `Privacy & Ethical Issues`).

Next, to distinguish between the different complaint types, we measured the frequency of each complaint type. To do so, we follow the same approach used by Khalid et al. (2015), where we measure the percentage of reviews that belong to each complaint type on a per app basis. We calculate the percentage per app since different apps can have a different number of reviews, and if we do not normalize per app, then apps with more reviews could bias our results. Once we calculate the percentage of reviews for each complain type, we take the median percentage (from all the wearable apps) and assign it to the complaint type. Finally,

Table 2 User complaint types and the row percentage of reviews in each one

Complaint type	Description	Example review	%
App Crashing	The wear app stops completely, goes idle or restarts	"*This app always crash on my phone.*"	8.0
Battery Drainage	The wear app is draining the battery excessively	"*Worked less than half the time, and killed my Wear battery.*"	7.4
Connection & Sync	Problems in connectivity with the wearable	"*Watch faces don't sync to watch. Uninstalling until this is fixed.*"	18.1
Cost	Complaint about the wear app costs or business model	"*Have to purchase premium just to download anything.*"	5.6
Device Compatibility	The wear app is not compatible with a given device	"*Won't work, only for famous smart watch.*"	14.9
Feature Removal	A feature has been removed after an update	"*The latest update removed the watch battery state/graph. Why?*"	1.4
Feature Request	The user requires a specific new feature	"*Can we have a option to pick personal images for the top half of ...*"	3.0
Functional Error	A bug related to the functionality of the wear app	"*Dont buy...even the weather does not display correctly.*"	26.1
Installation Problems	Issue while pushing the wear app to the wear device	"*App not pushed to watch.*"	8.6
Lack of Functionality	Absence or deficiency of features in the wear app	"*Nothing special about this app and it's faces. They're barely acceptable...*"	11.4
Missing Notifications	The wear app lost or delayed notifications	"*Since I updated the app I get no notifications on either of my watches...*"	2.0
Performance Issue	The wear app slows or over use the resource	"*The app performs very poorly even after the 1.4 update.*"	2.1
Privacy & Ethical	Invasion of privacy or ethical concerns complaint	"*Oh joy, more permissions and information gathering.. Smh*"	0.9
Spam Notifications	The wear app generates many unwanted notifications	"*Keeps sending notifications to my watch telling me to download ...*"	2.6
UI Problems	Complaints about the interface design	"*Watch faces don't fit and are even off centre in compatibility mode.*"	6.0
Uninformative	User reviews that do not have any useful information	"*It would not even let me play what even is this garbage???*"	8.2

we rank all of the complaint types from 1–15, where 1 is the highest (i.e., most frequent rank) and 15 is the least ranked.

The first three columns of Table 3 show the complaint types, the rank, and median percentage of user reviews per complaint type. From the table we observe that complaints related to `Functional Errors` (i.e., bugs related to the functionality of the wearable app), `Cost` (i.e., issues related to the business model of the wearable app) and `Lack of Functionality` (i.e., deficiencies in the functionality of the app) are the most frequent complaints for wearable apps.

Table 3 User complaint types rank & median percentage for the most frequent and the most impactful complaints

Complaint type	Most frequent		Most impactful	
	Rank	Median (%)	Rank	Median (1:2 star)
Functional Error	1	30.10	10	1.21
Cost	2	14.55	5	2.17
Lack of Functionality	3	14.22	8	1.46
Connection & Sync	4	10.03	4	2.63
Device Compatibility	5	9.57	2	4.10
UI Problems	6	7.34	14	0.78
Battery Drainage	7	7.06	12	1.06
App Crashing	8	6.38	6	2.06
Installation Problems	9	4.26	1	5.71
Feature Request	10	3.29	13	0.80
Spam Notifications	11	2.38	7	2.00
Performance Issues	12	1.98	15	0.65
Missing Notifications	13	1.65	11	1.17
Privacy & Ethical	14	1.12	3	3.17
Feature Removal	15	1.06	9	1.25

Our results also highlight several new user complaint types that require attention from both, developers and software engineering researchers, such as: `Connection & Sync`, `Missing Notifications`, and `Device Compatibility` issues. It is important to acknowledge that wearable devices rely on handheld devices to perform expensive processing tasks, hence the connection and synchronization between them is critical. Our findings highlight areas where wearable apps need to address to ensure the high quality of apps. In particular, we recommend the development of tools and techniques that can assist developers with connection and sync issues and device compatibility issues. This seems particularly important for wearable apps, which typically require a mobile device for most useful features (e.g., sending out messages, or checking online resources). Additionally, developers should be careful when pricing/advertising their apps, since cost-related complaints are frequent for wearable apps.

The most frequent complaints from the wearable app users are related to `Functionality Errors`, `Cost`, and `Lack of Functionality`.

4.2 Which User Complaint Types are the Most Negatively Impacting?

In addition to examining the frequency of the complaint types, we would also like to examine their potential negative impact. We examine the impact of each complaint type since, as previous work showed, the most frequent complaints may not be the most negatively impacting on the users (Khalid et al. 2015). A negative impact can induce a snowball effect

that will reduce the success of an app on its marketplace over time. For example, a complaint type that is very frequent, but that does not impact the users so much, may be better than a less frequent complaint type that has a large negative impact on the users. To study the impact, once again, we follow the same methodology used by Khalid et al. (2015), where we measure the ratio of 1-to-2 star reviews for each complain type. Similar to the case when we calculate the frequency, we perform this calculation on a per app basis. Finally, we assign the median score from all the apps to the specific complaint type, and we rank them based on their median score from 1–15 where 1 is the highest impactful and 15 is the least impactful

Table 3 (columns 4 and 5) show the rank and median score of 1:2 star reviews for each complaint type. A 1:2 ratio of 1.21 shows that there are 21% more 1-star reviews assigned to this complaint type than 2-star reviews. A 1:2 ratio less than 1 indicates that there are more 2-star reviews assigned to the complaint type. Typically, higher ratio numbers can indicate a higher negative impact, and vice versa. From Table 3, we observe that the most impacting complaint types are the ones related to `Installation Problems`, achieving a 1:2 ratio of 5.71. In addition to `Installation Problems`, `Device Compatibility` issues, and `Privacy & Ethical Issues` also have a substantial negative impact on users. It is worth mentioning that compatibility to wearable devices is a challenge for developers to address, particularly since the app store does not provide a way to filter apps based on a specific wearable device; nor are developers able to provide multiple APKs based on the different wearable devices configurations (Android Developers Documentation 2016b).

Our findings show that the most frequent user complaints are not necessarily the most impactful ones. A similar observation was made in the study by Khalid et al. (2015), in their study on handheld apps. For example, the `Installation Problems` complaint type has been ranked ninth in terms of number of complaints, while it is the highest impactful user complaint type.

Our findings show that wearable developers need to carefully test their apps, particularly the Android wear apps since, there are many Android devices that these wear apps need to be compatible with. Hence, we suggest the development of techniques that can address these compatibility issues, in particular issues that may impact the installation of wearable apps.

> **`Installation Problems`, `Device Compatibility`, and `Privacy & Ethical` issues are the most negatively impacting complaints.**

4.3 What Types of Complaints Do Developers Reply to?

Thus far, our study has mainly focused on user complaints. However, the Google Play Store provides the ability for developers to reply to user reviews in the hope of providing some clarification or support. Therefore, we mined a set of developer replies in order to get an answer for which types of user complaints developers care about the most. Complementing our user complaint data with their respective developer replies gives us a two-dimensional view of the issues that both, users and developers tend to care about.

Table 4 (columns 2 and 3) shows the rank and median percentage value of developer replies for the different complaint types. From the table, we observe that `Privacy & Ethical Issues`, `Performance Issues` and `Spam Notifications` are the top three most replied-to complaint types. On the other hand, `Functional Errors` (which is the most frequent type of complaint) and `Installation Problems` (the

Table 4 Median and percentage values for developer replies and reply time in days per complain type

Complaint type	Developer replies		Reply time	
	Rank	Median (%)	Rank	Median (days)
Privacy & Ethical	1	75.00	1	1
Performance Issues	2	58.33	14	4.5
Spam Notifications	3	50.00	6	2
Feature Removal	4	45.83	1	1
Missing Notifications	5	42.95	15	5.5
Cost	6	40.96	5	1.5
App Crashing	7	37.50	1	1
Device Compatibility	8	37.40	1	1
Installation Problems	9	37.30	6	2
UI Problems	10	36.93	13	4
Feature Request	11	33.33	12	3.5
Connection & Sync	12	31.77	6	2
Battery Drainage	13	25.00	11	3
Functional Error	14	22.22	10	2.5
Lack of Functionality	15	20.20	6	2

most impacting) are not in the top most replied-to complaints. Columns 4 and 5 in Table 4 show the rank and median time (in days) it took developers to reply to the different user complaint types. From the median reply time, we see that there are types of complaints that developers take longer to reply to, such as `Performance Issues`, `Missing Notifications` and `Feature Requests`.

In addition to the results presented in Tables 4 and 3, we also use a Bubble plot to combine these three factors, i.e., complaint impact, frequency and developer replies; this is shown in Fig. 3. The y-axis of the plot shows the rank in terms of developer replies, the x-axis shows the rank in terms of impact of the complaint and the size of each bubble is used to represent frequency. The issues that have the most impact and receive the most replies are in the lower left quadrant, the issues that have an impact but do not get much developer attention are in the upper left corner, the issues that do not have a high impact but receive developer attention are in the lower right quadrant and finally, issues that do not have a high impact and do not receive much developer attention are shown in the upper right quadrant.

From Fig. 3, we see that `Privacy & Ethical Issues`, `Spam Notifications`, `Cost`, and `App Crashes` are important for both, users in terms of impact and receive replies from the developers. Complaints related to `Missing Notifications`, `Feature Removals`, and `Performance Issues` tend to receive replies, however, they do not tend to have a significant impact on users. Issues related to `Installation Problems`, `Device Compatibility`, `Connection & Sync Issues`, and `Lack of Functionality` negatively impact users, but, developers do not tend to reply to them often. Lastly, complaints to `Functional Errors`, `Battery Drainage`, `Feature Requests`, and `UI Problems` tend to be of low importance to both users (in terms of impact) and developers (in terms of replies).

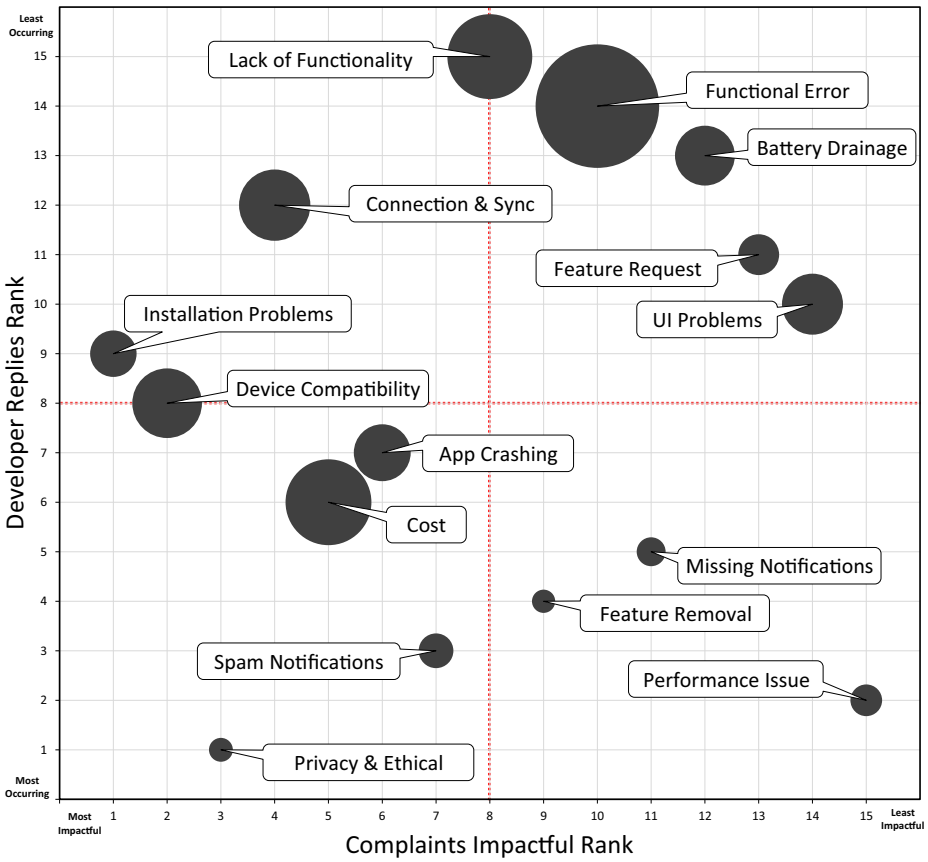


Fig. 3 Impactful complaint types vs. developer replies. The x-axis of the plot shows the rank of the most impactful complaint types. The y-axis shows the rank of developer replies, while the size of the bubble represents the frequency of each complaint

One take away for developers from this research question is that they need to pay closer attention to complaints related to the aforementioned issues (e.g., `Installation Problems` and `Device Compatibility`), since those are generating the most negative impact on users. This is particularly important since previous studies showed that responding to user reviews affects the app's success (Palomba et al. 2015; McIlroy et al. 2015). From our results, we observe that developers should put additional effort in replying to these negatively 'Impactful' complaint types to improve their app's ratings.

The most frequent types of user complaints that wearable app developers reply to are `Privacy & Ethical`, `Performance Issues`, and `Spam Notifications`. Furthermore, `Installation Problems`, `Device Compatibility`, and `Connection & Sync` issues have higher impact but are not replied-to by developers.

Table 5 Types of developer replies

Reply type	Description	Example reply
Request more details	The developer asking for more details	"*Can you record short video with this issue and send it to me [EMAIL]? Thank you.*"
Notify that Issue is Solved	The issue already solved in a newer version	"*We just released v1.6.1 that fixes the problem, please update it and let us know if the problem goes away, thank you.*"
Notify that a Solution is in Progress	Known issue and the developers work on it	"*I think I have identified the issue with crashes, should be fixed tomorrow.*"
Provide a Solution/ Workaround	Providing a solution to solve the issue	"*Hi ..., you can find steps for managing your notifications here: [WEBSITE] Let us know how it goes!*"
Offer Direct Support	The developer try to work directly on the case	"*Hello, please send me an email, I will help. Because of course it should work !*"
Offer Refund	The developer provide a refund offer	"*We will have a look at this. Let me know your order number to give you a refund.*"
Other	General replies	"*Thank you for your feedback.*"

4.4 How Do Developers Reply to Complaints?

In addition to quantifying the replies to the different complaint types, we also read through and classified the developer replies. In total, we had 782 replies. Similar to the case for the user reviews, we tagged each reply and added categories every time a reply did not fit into our existing categories. In the end, we ended up classifying the replies into 7 unique categories.

Table 5 shows the different reply categories, provides a brief description, and an example of each reply category. From the table, we observe that most replies try to provide a solution or gather more information about the complaints. In the case of paid apps, developers may also offer a refund.

Figure 4 shows the percentage of replies in each category. The percentage is simply measured as the number of replies in a category over all the 782 replies. We find that the majority of the replies provide a solution to solve the user complaint, followed by replies that request more details about the issue mentioned in the review and replies to notify the user that a solution is in progress. Based on Fig. 4, we see that the top four replies are related to the developers trying to get more information from the users, whereas, notification that a solution exists and offering a refund are the two least common replies.

Our results show that developers pay attention to the negative feedback given by users. When developers reply to low rated reviews, they do so to provide clarification or justification for missing features or problems in their wearable application. As previous studies have shown, developer replies tend to result in a positive update to the original low rating given by users (McIlroy et al. 2015). However, it is also important to consider that replying to the user is costly for developers. As we were able to observe from our dataset, the developer replies are manually generated; this problem needs to be addressed. A possible avenue for future work is to provide developers with a way to automatically respond to some of

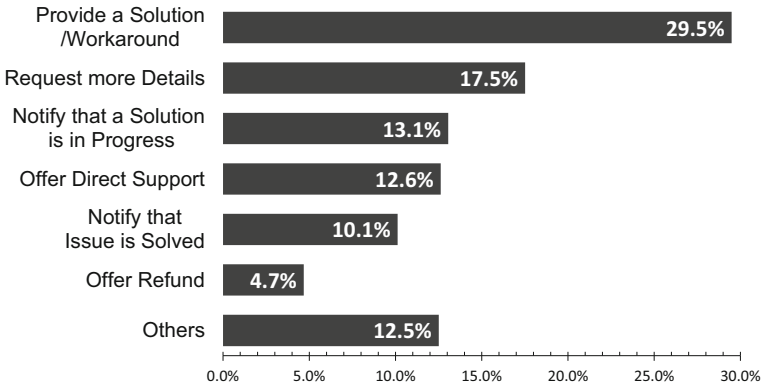


Fig. 4 Percentage of developer reply types

the most common complaints, which will lead to better reviews and minimal work for the developers.

> **Wearable app developers mostly reply to user complains to provide a solution/workaround, request more details and notify the user that a solution is in progress.**

5 Discussion

In this section, we examine the context of our findings. First, we discuss our findings with regards to comparing user complaint types of wearable and handheld apps. Then, we discuss the relation between user complaints and update of apps. Finally, we discuss the generalizability of our findings.

5.1 Comparing Wear and Handheld Device User Complaints

Thus far, we examined the user complaints for wearable apps. As mentioned earlier, prior work by Khalid et al. (2015) performed a similar study but for handheld apps. To determine the complaints that are specific to wearable apps and the complaints that are shared with handheld apps, we now contrast our findings of complaint types for wearable apps to complaint types of handheld apps.

Table 6 lists the complaint types found by our study and the ones mentioned in the study by Khalid et al. (2015). We compare the complaint types from Khalid et al.'s study with ours to better understand the unique challenges that wearable apps pose. We compare the complaint types in terms of their frequency and impact rank and refrain from comparing the percentage of reviews in each type, since the two studies are done on different sets of apps.

We observe from Table 6 that 10 of the 15 complaint types reported by our study are also mentioned in the study on handheld devices. However, 5 complaint types appear only for wearable apps (marked in bold in the table). Table 6 also shows that there are two complaint types appear only for handheld devices (marked in italics in the table).

Table 6 Comparison of complaint types for wear and handheld devices. Based on the findings reported by Khalid et al. (2015)

Complaint type	Frequency rank		Impact rank	
	Wearable	Handheld	Wearable	Handheld
Functional Error	1	1	10	7
Cost	2	7	5	2
Lack of Functionality	3	–	8	–
Connection & Sync	4	–	4	–
Device Compatibility	5	8	2	5
UI Problems	6	5	14	10
Battery Drainage	7	12	12	8
App Crashing	8	3	6	4
Installation Problems	9	–	1	–
Feature Request	10	2	13	12
Spam Notifications	11	–	7	–
Performance Issues	12	10	15	11
Missing Notifications	13	–	11	–
Privacy & Ethics	14	9	3	1
Feature Removal	15	6	9	3
Netwrok Problem	–	4	–	6
Uninteresting Content	–	11	–	9

We notice that complaints related to `Lack of Functionality` are mentioned, and with a high rank for wearable apps. Through our manual examination of the wearable reviews, we noticed that the lack of functionality is frequently mentioned in wearable app complaints due to the fact that wearable devices and apps are limited in what they can do, and heavily depend on the phone for any major features (Chauhan et al. 2016). Hence, users will often misunderstand what the app does, and quickly jump to a conclusion that the wearable app is useless since it does not provide more functionality compared to its handheld companion app. For example a user who was disappointed by the lack of functionality wrote: "*Useless app! Why bother downloading an app if you still have to take the phone out of your pocket. Not a smart move for a smartwatch...*" Hence, one recommendation based on our findings to wearable app developers is to ensure that the functionality of their wear apps are unique and add value to the user.

`Connection & Sync` problems are also highly ranked and only mentioned for wearable apps. This problem was very clear from the reviews that we read. As mentioned earlier, in many cases the wearable apps heavily depend on the handheld devices and due to the fact that there exist a variety of wearable devices and a wide variety of handheld Android devices (Khalid et al. 2014), these `Connection & Sync` problems are exacerbated. For example a user stated "*No longer lets my Watch stay connected to my phone...*" The problem is certainly magnified for wearable devices since they are limited to pretty much just displaying time and counting steps without a connection to the phone. One recommendation to developers based on our findings is for them to carefully select which devices they support and pay special attention to testing connection & sync features since they can render the wearable app useless and negatively impact the users.

Other issues related to notifications (i.e., `Spam Notification` and `Missing Notifications`) are also reported for wearable apps since this is one of the main ways that apps on the handheld device communicate with the user of the wearable app. As with any notification service, overdoing it causes users to complain. For example, a user mentions in one of their negative reviews "*No issues till it started popping up suggested watch faces in my notifications*". Based on our findings, one recommendation is for wear developers to not overuse notifications on their wearable apps since, clearly, it can get annoying for the users.

Also, many wearable app users complain about `Installation Problems`, contrary to the case for handheld apps. After closer examination of the reviews related to this complaint type, we find that the root cause of this complain type can be linked to the wearable app's distribution mechanism (Mujahid 2017). To distribute a wearable app, a developer can embed the APK of the wearable app inside its corresponding handheld APK. Then, when users want to install the wearable app they first install the handheld app and then the handheld device pushes the wearable app to the wearable device. This process is error prone, especially given the fact that there exist a plethora of handheld and wearable devices (Mujahid 2017). One example of this case is the review where the user mentions: "*Where is it?: I downloaded the app, opened watch faces on my Sony Smartwatch 3 and it's not there. I even opened my apps folder on my phone (LG G4) and it doesn't show. I know it is installed on the phone because when I find the app through Play store the only option it gives me is to uninstall.*" Since this behavior is specific to wearable apps only, it does not make sense that handheld app user's would complain about such a problem. More importantly, Table 6 shows that `Installation Problems` does not only frequently happens but that they also have the most negative impact on the wearable apps. Hence, to mitigate such issues, we highly recommend that wearable app developers carefully consider the list of compatible devices they support and properly test their wearable apps to ensure such installation problems are addressed.

On the other end of the spectrum, `Network Problem` and `Uninteresting Content` user complaints appear only for handheld apps. For `Network Problems`, handheld users complain about that the app having trouble with the network connection or is slow. Clearly, wearable app users do not face such problems since wearable apps often rely on the handheld to perform the network connection. For `Uninteresting Content`, it seems that wearable apps users do not complain about the content of the apps due to the fact that these wearable apps do not display much content anyways, especially given their limited screen size.

In Table 6, we also underline the cases where the ranks for the same complaint types have a clear difference for wearable and handheld devices. Interestingly, we find that although `Cost` and `Battery Drainage` have a high frequency rank for wearable apps, and they have a lower impact rank compared to handheld devices (note that a lower rank score indicates higher importance). Through our manual examination of the wearable app reviews, we did not find any complaints that are specific to the wearable apps. Thus, we conclude that the difference in ranks of the two aforementioned complaint types is due to the fact that wearable app users simply complain less about cost and battery drainage. Another possible reason for the difference in ranks is the difference in sampled apps (between our study and that by Khalid et al. 2015), which we discuss in detail in Section 6.

Additionally, we find that `Feature Requests`, `Feature Removals`, and `Privacy & Ethical` issues have a lower frequency and impact rank for wearable apps. We believe that due to the fact that wearable devices are often seen as an add-on, users expect less from them, hence they are less impacted when a feature is missing

or removed. Finally, Table 6 shows that `App Crashing`, `UI Problems`, `Device Compatibility`, `Functional Error`, and `Performance Issue` have similar or equal importance in both handheld and wearable apps.

Although we provided a detailed comparison of the complaint types above, we believe that such comparison can occupy an entire study on its own. In some cases, other information (e.g., hardware or API limitations) may need to be triangulated to enhance this analysis, which goes beyond the goals and scope of our study. Nevertheless, we do believe that our findings are the first to highlight such differences and open interesting questions regarding the differences between traditional handheld apps that run on phones and wearable apps that run on wearable devices.

5.2 Update-Related Complaints

A key observation presented in the paper by Khalid et al. (2015) is that many users posted complaints after an update. In handheld devices, update-related complaints account for approximately 11% of their studied reviews (Khalid et al. 2015). Similarly, we also noticed that many reviews mentioned problems after an update during our manual analysis. In fact, we found that approximately 12% of the examined wearable app complaints mentioned issues arising after an update. Our finding is similar to that reported by Khalid et al. (2015). A clear example of issues arising after an update are evident with the following review: "*Oh, my! My watch is completed useless again. Stop updating! Every time you fix a bug, you create many more!*"

Although the study on handheld devices reported that most complaints after an update were related to functional errors, the addition/removal of a feature, and hidden costs, we found that most of the complaints for wearable apps were related to `Connection & Sync Issues` (32.9% of the reviews that report a problem after an update), `Functional Errors` (30.5%) and `Battery Drainage` (23%). For example, the user in the review below complains about connection problems since the last update of the app. "*Constantly drops connection to watch since update*"

Our findings here draw attention to the importance of regression testing before an update is released. In particular, we suggest performing regression testing for `Connection & Sync` and `Battery Drainage Issues`.

5.3 Are the Complains Specific to the Studied Wearable Apps

To examine the wearable app users complaints, our experiment was conducted on a sample composed of 19 wearable apps. To do so, we start by collecting user reviews from all the free wearable apps published on Google Play Store. Then, we choose to manually classify complaint types from the wearable apps that have more than 100 negative rated reviews. However, based on suggestion from prior work (Martin et al. 2015), our results may be affected by our sampling process, hence our complaint types may not be representative of the complaints found in wearable apps in general. In other words, do the found users complaint types generalize or are they specific to the studied wearable apps?

To answer the posed question, we performed a sanity check on users' complaints from all wearable apps in our dataset. We first took a statistically significant sample from the 17,177 wearable complaints that were not included in the primary classification process (Section 3.2). The sample size was selected randomly to attain 5% confidence interval and a 95% confidence level. This sampling process resulted in 641 total complaints. Next, each of the

first two authors, separately, classified all selected complaints. Through the classification process, we kept an option to add new complaint types, as we did previously in our primary classification. Finally, we measured the Cohen's Kappa to measure the level of agreement between the two classifications (Cohen 1960). The level of agreement was +0.63, which is considered to be fair to good agreement (Fleiss and Cohen 1973).

Ultimately, the results of this experiment showed the same complaint types that were found in the primary classification. Thus, we conclude that the identified users' complaints of wearable apps are not specific to the sampled apps we studied in this paper. It is important to note that in this analysis we are interested only in examining how generalizable is the result of the primary classification. Hence, we did not consider the distribution of the complaints across the complaint types.

6 Threats to Validity

Our study is subject to a number of internal threats and external threats to validity.

6.1 Internal Validity

To identify wearable app user complains, we manually classify 2,667 reviews. Like any human activity, the manual classification is susceptible to human error. To mitigate this threat, two of the paper authors performed the manual classification. We also measured the agreement between the two annotators using Cohen's Kappa, which showed good agreement with value of +0.68.

Due to our manual classification phase being time consuming, we did not cover all of our data set, instead we took a sample of our dataset. This threat was addressed by taking a statistically representative sample with a 95% confidence level for each of the apps in our data set.

Our categorization is heavily dependent on the quality of the reviews provided by the users and their respective developer replies. As shown in prior studies, most user reviews contain useful information, however, in some cases different levels of details may lead to different complaint types.

Martin et al. (2015) studied a common problem of sampling bias when research work analyzes data mined from app stores. This problem exists because of an often-limited access to a full set of apps and their reviews to be studied. When studies are done only on subsets of data, they can be potentially biased and draw non-reasonable conclusions.

To understand the difference between wearable apps user complaints and handheld user complains, we contrast our findings with the results reported by Khalid et al. (2015). However, the two studies examine different set of apps which may affect our findings. To mitigate such an effect, we compare the complaint types in terms of their frequency and impact ranks (rather than percentages of reviews, for example).

In the scope of our work, despite our efforts in the data crawling phase, although we did not face the limitations for crawling described by Martin et al. (2015) in the Google App Store, we do not claim to have a guaranteed full set of reviews for each app. Furthermore, our work does target user complains only, which is already a given subset of user reviews; this is unavoidable for our purpose. However, to address the threat of bias, we took statistically significant random samples of the reviews for each app we study. These measures were taken precisely to remove bias from the study while following the similar approach previously used by Khalid et al. (2015)

It is important to note that throughout our study, we use the low ratings given by user reviews, i.e., 1 or 2 stars ratings, as a way to assess impact. We used this definition of impact, since it was used by Khalid et al. In their work. That said, we do believe that other definitions for impact are possible. For example, the messages of the reviews could be analyzed to determine the sentiment expressed by users; this can be done using a tool such as *Sentistrength* (Sentistrength 2017). In the future, we plan to explore other ways of measuring impact of a review.

6.2 External Validity

We found over 17,000 wearable app related user reviews but we filtered them down to 5,751, and hence, our data set can be considered small. This however, is because this platform is fairly new and we were only able to select the 19 wearable apps that had over 100 user reviews to make our findings from them relevant. On the same line of thought, the filtering phase for the wearable app related reviews may have discarded some useful information that did not match our filtering rules. Moreover, our study is performed on Android Wear apps, hence our findings may not generalize to wearable apps from other platforms.

7 Conclusion

Users provide direct feedback on their experience of mobile apps through user reviews. Prior work showed that user reviews can be mined to effectively determine user complaints to help developers understand the issues that users of handheld apps face the most, so they can be mitigated.

Given that wearable apps are a new trend that is only increasing in popularity, in this paper, we mine user reviews in order to understand the user complaints of wearable apps. We manually sample and categorize 2,667 reviews from 19 wearable apps. We find 15 unique complaint types that wearable users report in user reviews. We also examine the replies that developers post to some of the user complaints in order to determine complaints that developer care most about and identify areas that are important for users, but are not well replied-to by developers.

Our findings indicate that the most frequent complaints are related to `Functional Errors`, `Cost` and `Lack of Functionality`, whereas the most negatively impacting complaints are related to `Installation Problems`, `Device Compatibility`, and `Privacy & Ethical Issues`. On the other hand, we find that developers reply most to complaints related to `Privacy & Ethical Issues`, `Performance Issues` and notification-related issues. And, when developers reply they mostly do so to provide a solution, request more details or let the user know that they are working on solving the problem. We also compare our findings on wearable apps with the study by Khalid et al. (2015) on handheld apps and find that; 1) 10 of our 15 complaint categories are also reported for handheld apps; though wearable apps have unique issues related to `Lack of Functionality`, `Installation Problems`, `Connection & Sync`, `Spam Notifications`, and `Missing Notifications`. 2) Similar to the case of handheld apps, approximately 12% of complaints are mentioned after apps received an updated. To enable future research on the topic, we make our dataset publicly available.

References

Ahola J (2015) Challenges in android wear application development. In: Proceedings of the international conference on web engineering, ICWE '15. Springer, pp 601–604

Android Developers Documentation (2016a) Creating wearable apps. https://developer.android.com/training/wearables/apps/index.html. Accessed 2 Oct 2016

Android Developers Documentation (2016b) Filters on google play. https://developer.android.com/google/play/filters.html. Accessed 18 Dec 2016

Android Wear Center (2016) http://www.androidwearcenter.com. Accessed 9 Sep 2016

Bonato P (2010) Wearable sensors and systems. IEEE Eng Med Biol Mag 29(3):25–36

Chauhan J, Seneviratne S, Kaafar MA, Mahanti A, Seneviratne A (2016) Characterization of early smartwatch apps. In: Proceedings of the 2016 IEEE international conference on pervasive computing and communication workshops, PerCom '16. IEEE, pp 1–6

Chen N, Lin J, Hoi SC, 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 '14. ACM, pp 767–778

Ciurumelea A, Schaufelbühl A, Panichella S, Gall HC (2017) Analyzing reviews and code of mobile apps for better release planning. In: Proceedings of the 24th IEEE international conference on software analysis, evolution and reengineering, SANER '17. IEEE, pp 91–102

Cohen J (1960) A coefficient of agreement for nominal scale. Educ Psychol Meas 20:37–46

Di Sorbo A, Panichella S, Alexandru CV, Shimagaki J, Visaggio CA, Canfora G, Gall H (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 '16. ACM, pp 499–510

Di Sorbo A, Panichella S, Alexandru CV, Visaggio CA, Canfora G (2017) Surf: summarizer of user reviews feedback. In: Proceedings of the 39th international conference on software engineering companion, ICSE-C '17. IEEE Press, pp 55–58

Do Q, Martini B, Choo KKR (2017) Is the data on your wearable device secure? An android wear smartwatch case study. Softw: Pract Exp 47(3):391–403

Finkelstein A, Harman M, Jia Y, Martin W, Sarro F, Zhang Y (2017) Investigating the relationship between price, rating, and popularity in the blackberry world app store. Inf Softw Technol 87:119–139

Fleiss JL, Cohen J (1973) The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educ Psychol Meas 33:613–619

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 '13. ACM, pp 1276–1284

Galvis Carreño LV, Winbladh K (2013) Analysis of user comments: an approach for software requirements evolution. In: Proceedings of the 2013 international conference on software engineering, ICSE '13. IEEE Press, pp 582–591

Goko Store (2016) http://goko.me. Accessed 9 Sep 2016

Guzman E, Maalej W (2014) How do users like this feature? A fine grained sentiment analysis of app reviews. In: Proceedings of the 22nd IEEE international requirements engineering conference, RE '14. IEEE, pp 153–162

Ha E, Wagner D (2013) Do android users write about electric sheep? Examining consumer reviews in google play. In: Proceedings of the 10th IEEE consumer communications and networking conference, CCNC '13, pp 149–157

Harman M, Jia Y, Zhang Y (2012) App store mining and analysis: Msr for app stores. In: Proceedings of the 9th IEEE working conference on mining software repositories, MSR '12. IEEE Press, pp 108–111

Hoon L, Vasa R, Schneider JG, Mouzakis K (2012) A preliminary analysis of vocabulary in mobile app user reviews. In: Proceedings of the 24th Australian computer-human interaction conference, OzCHI '12. ACM, pp 245–248

Keertipati S, Savarimuthu BTR, Licorish SA (2016) Approaches for prioritizing feature improvements extracted from app reviews. In: Proceedings of the 20th international conference on evaluation and assessment in software engineering, EASE '16. ACM, pp 33:1–33:6

Khalid H, Nagappan M, Shihab E, Hassan A (2014) Prioritizing 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 '14. IEEE, pp 610–620

Khalid H, Shihab E, Nagappan M, Hassan A (2015) What do mobile app users complain about? IEEE Softw 32(3):70–77

Lyons K (2015) What can a dumb watch teach a smartwatch?: Informing the design of smartwatches. In: Proceedings of the ACM international symposium on wearable computers, UbiComp '15. ACM, pp 3–10

Martin W, Harman M, Jia Y, Sarro F, Zhang Y (2015) The app sampling problem for app store mining. In: Proceedings of the 12th IEEE/ACM working conference on mining software repositories, MSR '15. IEEE, pp 123–133

Martin W, Sarro F, Jia Y, Zhang Y, Harman M (2017) A survey of app store analysis for software engineering. IEEE Trans Softw Eng 43(9):817–847

McIlroy S, Shang W, Ali N, Hassan A (2015) Is it worth responding to reviews? A case study of the top free apps in the google play store. IEEE Softw 34:64–71

McIlroy S, Ali N, Khalid H, Hassan A (2016) Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empir Softw Eng 21(3):1067–1106

Min C, Kang S, Yoo C, Cha J, Choi S, Oh Y, Song J (2015) Exploring current practices for battery use and management of smartwatches. In: Proceedings of the ACM international symposium on wearable computers, UbiComp '15. ACM, pp 11–18

Mujahid S (2017) Determining and detecting permission issues of wearable apps. Master's thesis, Concordia University, Montreal

Mujahid S, Sierra G, Abdalkareem R, Shihab E, Shang W (2017) Examining user complaints of wearable apps: a case study on android wear. In: Proceedings of the 4th IEEE/ACM international conference on mobile software engineering and systems, MobileSoft '17. IEEE

Nagappan M, Shihab E (2016) Future trends in software engineering research for mobile apps. In: Proceedings of the 23rd IEEE international conference on software analysis, evolution, and reengineering, SANER '16. IEEE

Pagano D, Maalej W (2013) User feedback in the appstore: an empirical study. In: Proceedings of the 21st IEEE international requirements engineering conference, RE '13. IEEE Press, pp 125–134

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: Proceedings of the IEEE international conference on software maintenance and evolution, ICSME '15, 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 '17. IEEE Press, pp 106–117

Panichella S, Sorbo AD, Guzman E, Visaggio CA, Canfora G, Gall H (2015) How can i improve my app? classifying user reviews for software maintenance and evolution. In: Proceedings of the IEEE international conference on software maintenance and evolution, ICSME '15. IEEE, pp 281–290

Panichella S, Di Sorbo A, Guzman E, Visaggio CA, Canfora G, Gall H (2016) Ardoc: app reviews development oriented classifier. In: Proceedings of the 24th ACM SIGSOFT international symposium on foundations of software engineering, FSE '16. ACM, pp 1023–1027

Park S, Jayaraman S (2003) Smart textiles: wearable electronic systems. MRS Bull 28(8):585–591

Rawassizadeh R, Price BA, Petre M (2015) Wearables: has the age of smartwatches finally arrived? Commun ACM 58(1):45–47

Seaman CB (1999) Qualitative methods in empirical studies of software engineering. IEEE Trans Softw Eng (IST) 25(4):557–572

Sentistrength (2017) Sentiment strength detection. http://sentistrength.wlv.ac.uk/. Accessed 24 Aug 2017

Tehrani K, Michael A (2014) Wearable technology and wearable devices: everything you need to know. Wearable Devices Magazine. http://www.wearabledevices.com/what-is-a-wearable-device/. (Accessed 25 Aug 2017

Teng XF, Zhang YT, Poon CCY, Bonato P (2008) Wearable medical systems for p-health. IEEE Rev Biomed Eng 1:62–74

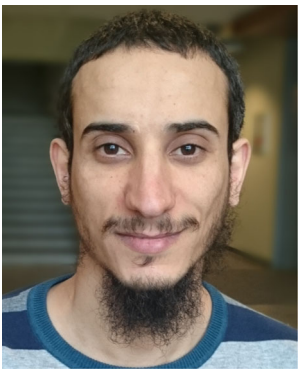
Usman M, Britto R, Börstler J, Mendes E (2017) Taxonomies in software engineering: a systematic mapping study and a revised taxonomy development method. Inf Softw Technol 85(Supplement C):43–59

Vasa R, Hoon L, Mouzakis K, Noguchi A (2012) A preliminary analysis of mobile app user reviews. In: Proceedings of the 24th Australian computer-human interaction conference, OzCHI '12. ACM, pp 241–244

Wei J (2014) How wearables intersect with the cloud and the internet of things: considerations for the developers of wearables. IEEE Consum Electron Mag 3(3):53–56
Wright R, Keith L (2014) Wearable technology: if the tech fits, wear it. J Electron Resour Med Libr 11(4):204–216
Zhang H, Rountev A (2017) Analysis and testing of notifications in android wear applications. In: Proceedings of the 39th international conference on software engineering, ICSE '17. IEEE Press

Suhaib Mujahid is a Ph.D. student in the Department of Computer Science and Software Engineering at Concordia University. He received his master's in Software Engineering from Concordia University (Canada) in 2017, where his work focused on detection and mitigation of permission-related issues facing wearable app developers. He did his Bachelors in Information Systems at Palestine Polytechnic University. His research interests include wearable applications, software quality assurance, mining software repositories and empirical software engineering. You can find more about him at http://users.encs.concordia.ca/~s_mujahi.

Giancarlo Sierra is a MASc. student in the Department of Computer Science and Software Engineering at Concordia University. He received his bachelor's degree in Computer Science in 2013 from SEK International University in Ecuador. With an industry background, his research interests include Technical Debt, Mining Software Repositories, and Mobile Software Engineering, among others. His work currently focuses on the detection and prioritization of Self-Admitted Technical Debt. You can reach him at g_sierr@encs.concordia.ca and find more about him at http://das.encs.concordia.ca/members/giancarlo-sierra.



Rabe Abdalkareem is a PhD candidate in the Department of Computer Science and Software Engineering at Concordia University, Montreal. His research investigates how the adoption of crowdsourced knowledge affects software development and maintenance. Abdalkareem received his master's in applied computer science from Concordia University. His work has been published at premier venues such as FSE, ICSME, MSR and MobileSoft, as well as in major journals such as IEEE Software and IST. Contact him at rab_abdu@encs.concordia.ca; http://users.encs.concordia.ca/~rab_abdu.

Emad Shihab is an Associate Professor in the Department of Computer Science and Software Engineering at Concordia University. He received his PhD from Queens University. Dr. Shihab's research interests are in Software Quality Assurance, Mining Software Repositories, Technical Debt, Mobile Applications and Software Architecture. He worked as a software research intern at Research in Motion in Waterloo, Ontario and Microsoft Research in Redmond, Washington. Dr. Shihab is a senior member of the IEEE and member of the ACM. More information can be found at http://das.encs.concordia.ca.

Weiyi Shang is an Assistant Professor and Concordia University Research Chair in Ultra-large-scale Systems at the Department of Computer Science and Software Engineering at Concordia University, Montreal. He has received his Ph.D. and M.Sc. degrees from Queen's University (Canada) and he obtained B.Eng. from Harbin Institute of Technology. His research interests include big data software engineering, software engineering for ultra-large–scale systems, software log mining, empirical software engineering, and software performance engineering. His work has been published at premier venues such as ICSE, FSE, ASE, ICSME, MSR and WCRE, as well as in major journals such as TSE, EMSE, JSS, JSEP and SCP. His work has won premium awards, such as SIGSOFT Distinguished paper award at ICSE 2013 and best paper award at WCRE 2011. His industrial experience includes helping improve quality and performance of ultra-large-scale systems in BlackBerry. Early tools and techniques developed by him are already integrated into products used by millions of users worldwide. Contact him at shang@encs.concordia.ca; http://users.encs.concordia.ca/~shang.