How to Report App Feedback? Analyzing Feedback Reporting Behavior

Jeungmin Oh
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
jminoh@kaist.ac.kr

Uichin Lee
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
uclee@kaist.ac.kr

SangJeong Lee
Software R&D Center, Samsung Electronics
Seoul, Republic of Korea
sj94.lee@samsung.com

CHI'16 Extended Abstracts, May 07-12, 2016, San Jose, CA, USA
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2892368

Abstract
Users face various usability issues when interacting with mobile apps. While collecting quality in-situ feedback is extremely valuable for the developers for UX improvement, our understanding of user feedback reporting behavior and interface design is very limited. In this paper, we conducted a user study to solicit feedback from five mobile apps using think-aloud protocol with video recordings. Using the collected data, we investigated the attributes of usability issues related to feedback reporting behavior. We identified four key attributes of usability issues related to feedback reporting behavior, i.e., usability issue topics, origin of usability issues, types of actions, and content/context dynamicity. Our analysis will provide a useful foundation for building a feedback reporting system for end users.

Author Keywords
Mobile apps; user feedback; usability issue;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

Introduction

User involvement in software engineering is considered to be very important for improving software quality and user experiences [11, 21]. Prior studies generally investigated its importance in the stage prior to deployment [2, 3, 24]. For instance, usability professionals and recruited testers examine working prototypes and provide user feedback [15]. In the case of mobile apps, development cycle is much shorter, and software updates are much easier. Listening to the customer voices and reflecting their feedback would ultimately improve their user experiences.

While collecting quality in-situ feedback is extremely valuable for the developers for UX improvement, our understanding of user feedback reporting behavior and interface design is very limited. This paper investigates what are the key properties of usability issues related to feedback reporting behavior. To the best of our knowledge, prior studies only focused on designing supporting tools for usability testing such as theoretical user models (e.g., Norman's action model [17], User Action Framework [1]) and usability issue classification methods [24]. These are suitable for specialists to perform high-level classification of usability issues, but they do not bring practical insights into what forms the properties of usability issues, which is expected to influence the choice of different feedback methods.

We studied how users report usability issues encountered while interacting mobile apps. In particular, we found that there are four key attributes of usability issues related to feedback reporting behavior, i.e., usability issue topics, origin of usability issues, types of actions, and content/context dynamicity. These attributes provided useful insights into how to systematically help users to provide quality feedback.

Background and Related Work

User involvement is a widely accepted principle in application design and development. Kujala [12] found that user involvements improve various aspects of software engineering such as requirement analysis. Prior studies [2, 3] confirmed that user involvements generally had positive effects on system success. However, this benefit often comes at the cost of time and budge concerns [14].

Usability engineering systematically considers user involvement in software development: e.g., understanding the users, setting usability goals, conducting parallel/participatory design, and performing iterative design **(**building**,** testing**,** improving prototypes**)**, and collecting user feedback in the field [24]. One common practice is to perform usability testing where usability specialists attempt to find usability issues based on usability guidelines. The intention of usability testing is either formative (to help improve the interface as part of an iterative design process) or summative (to assess the overall quality of an interface). In a typical usability test, users perform a set of usability tasks in the laboratory from which usability metrics such as task completion time are measured. Also, users are often asked to think out loud to collect informal comments of their usage experiences. After the tests, usability specialists review the measurement and self-report data to uncover usability problems. In most cases, usability testing happens in a usability laboratory, but remote testing can be done by holding online conferencing or by simply self-reporting the results after completing the

App (ver.)	Category	Description
Busan Bus (2.1.11)	Transpor-tation	A real-time bus schedule viewing app
Maniana (1.262)	Product-ivity	A to-do list app
How About (1.0.93)	Music	A Youtube music recommender app
Wiki-pedia (1.4 Beta14)	Books & Reference	An official Wikipedia mobile client
Gesture Way (4.2.25)	Tools	A gesture shortcut app

Table 1: Five apps used in the user

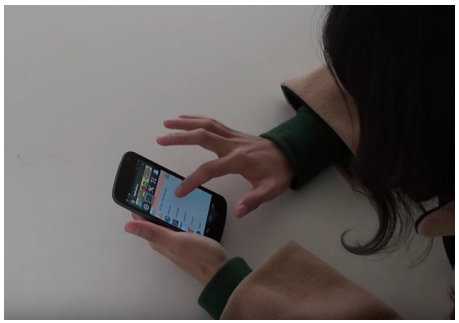

Figure 1: Video recording during think-aloud experiment

assigned usability tasks [6, 9]. However, formal usability testing often takes significant resources such as time and money [10], and simplified methodologies are often used among practitioners [16].

Beyond testing there are other usability assessment methods such as questionnaires/interviews, usage logging, and user feedback. Questionnaires and interviews are useful methods for studying how users use systems and what features they like or dislike. Analyzing logged data helps examine usage statistics; e.g., answering when users encountered the error most or under what context such as time and location users interacted with the software [7]. Additionally, user feedback is a valuable source of usability information since it is initiated by the users and shows their immediate and pressing concerns as well as any changes in their needs and usage circumstances. In the case of mobile apps, collecting in-situ user feedback will be very useful since mobile app usage can be context dependent. In recent years, mobile app development becomes more adaptive and agile in that rapid release of working software and user involvements are considered important for market success [19].

While many apps have in-app feedback pages, the most popular channels for user feedback are the app review sections in the app stores [8, 18]. According to Pagano et al. [21], a significant fraction of app reviews expressed usability issues (e.g., feature information / requests, bug reports, shortcomings), but the quality of user reviews is limited because most reviews are short in text. In terms of feedback reporting mechanisms, app stores support only text-based feedback. In addition to text, several popular apps such as Facebook allow users to attach the current screenshot. For those apps with a good user base, online forums are maintained to collect user feedback in a more organized way. A few open source projects even have bug tracking systems such as Bugzilla in Firefox Mobile where structured bug reporting forms are used—usage of bug tracking systems has been thoroughly investigated in software engineering literature [4, 5, 13, 22, 25]. Screen recording has been also used in previous studies for richer in-situ context sometimes with more apparatus [26, 27, 28]. However, its usage is still not popular in commercial apps except for few professional services.

Analyzing Feedback Reporting Behavior

Studies were conducted on usability problem classification such as Usability Problem Taxonomy [23] and User Action Framework [1]. The existing methods were designed to assist usability specialists to analyze problems in depth; however, these methods do not bring any notable insight into which attributes of usability issues are related to feedback reporting behavior, and how the normal user delivers the information. We endeavored to answer this question by establishing a controlled lab environment. Eight participants were recruited for a user study. Their ages ranged from 26 to 29 years (M = 27.5, SD = 1.41). Seven of the participants were Android users, and only one was an iPhone user. They were asked to use the mobile apps (see Table 1) and to provide any feedback

[1] https://github.com/kmshack/busanbus-android

[2] http://code.google.com/p/maniana

[3] https://github.com/recomio/howabout-android

[4] https://github.com/wikimedia/WikipediaMobile

[5] https://play.google.com/store/apps/details?id=sk.cep.gesturesway

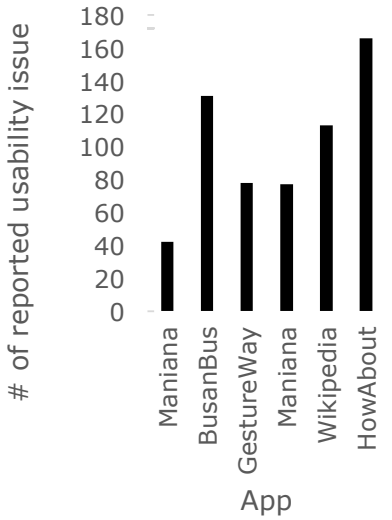

Figure 2: Number of reported issues by app

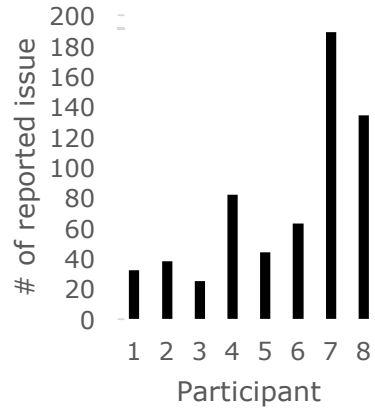

Figure 3: Number of reported issues by participant

which was worthwhile reporting to the developers, using a think-aloud protocol. As mentioned previously, the purpose of this study was not to test the reporting methods, but rather to analyze the attributes of usability issues as related to feedback reporting behavior. To this end objective, during the user study session, the participants were instructed to use five apps, and they spent approximately 15 minutes on each app. The entire sessions were video-taped and later transcribed for qualitative analysis.

With intention we selected relatively unpopular apps, that have been identified with a number of notable usability issues. The BusanBus provides bus schedules that display real-time bus location information. We expected that the users may encounter some issues while searching and navigating real-time sensor data. We chose Maniana because productivity apps such as a calendar or to-do list are among the most common apps that smartphone users install. Since users may be actively using similar scheduling software (e.g., Google Calendar), they would be more likely to encounter usability issues relatively easily. We chose HowAbout because one of app's main features is music streaming. Streaming applications are expected to potentially experience various usability issues regarding Internet connectivity, controls, and responsiveness. Wikipedia was selected because it is one of the most popular web pages, and it belongs to the general categories of online references. Lastly, GestureWay was chosen since it includes various gesture features that may lead to a number of usability issues related to gesture input.

We analyzed the recorded video of app usage, and 608 usability-related statements were identified by the usability experts. During the coding process, if two issues were stated in a single sentence, we separated this information into two issues. After removing similar/duplicate feedback reports in order, two authors then collaboratively performed affinity diagramming to cluster similar usability issues.

Attributes of Usability Feedback Reporting

Affinity diagramming resulted in four key attributes in usability issues related to feedback reporting behavior, i.e., usability issue topics, origins of usability issues, types of actions, and content and context dynamicity.

Usability issue topics

We identified three broader topics associated with usability issues, as shown in Figure 4; i.e., malfunction (functional errors with specific entities), suggestion (requests for adding/removing specific entities), and look & feel (dissatisfaction with UI elements).

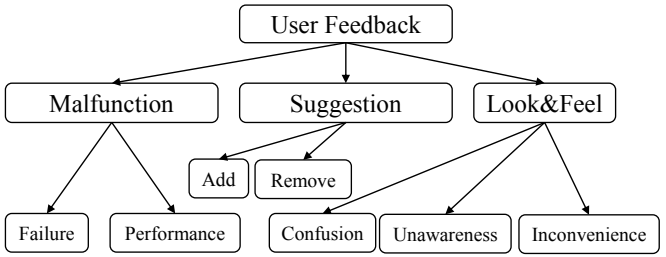

Figure 4: Usability issue topic categories

Malfunction includes two sub-categories: functional failures and performance problems. A failure issue is present when a certain entity such as UI Element or function does not work; for example, "Map doesn't load." A performance problem occurs when a certain entity does not work properly; for instance, "It seems that loading takes too much time…"

Suggestion consists of two sub-categories: adding a new entity or removing an undesirable entity; "[In a playlist screen] I wish that there is a repeat function."

Look & Feel issue has three sub-categories. Confusion may occur when the appearance of the UI element is not straightforward or confusing, for example "[Pointing at a trashcan-like icon] The recovery button looks like a delete button." Unawareness is a type of issue in which the meaning or the usage of the UI element is difficult to understand, for example "I don't know what this button is for." Inconvenience is related to the design of the UI element causing any discomfort, for example "It want the real time bus location screen to fit into a single screen."

Origin of usability issues

The first factor of usability issues is the origin. Four types of entities were related to the origin in the user reports; i.e., app, screen, UI element, and function. Function consists of three sub-categories: intra-screen function (function occurring within a screen such as a deleting function of an item), background function (a function running behind the screen) such as GPS and watching, and inter-screen function (function involving multiple screens, e.g. configuration of font size). The detail is depicted in Figure 5.

Regarding this function, a participant reported "Function for changing font doesn't work…" The UI element is a visual entity that controls interactions between the user and the app; for instance, "I don't understand what this color bar means." A typical screen has a collection of UI elements and functions. In the case of the app, one participant referred to an app as "This app is generally slow." describing an issue across the entire app. Identifying the origin of a usability issue in feedback reporting is very important in that it allows the developer to know where to investigate.

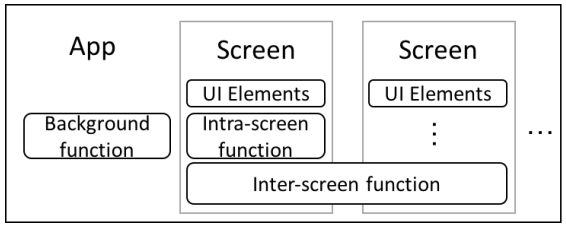

Figure 5: Origin of usability issues

Types of actions

Usability issues often involve user interactions with the entities. A user may perform a single interaction (e.g. touching a button) or a series of interactions (e.g., dragging followed by touching); one participants commented, "It would be nice if the letters are highlighted when I press the hyperlink." Some problems could occur without any user interaction, mainly in suggestion and look & feel areas. For instance, "There is no function for deleting playlist at once." would come up with no relevant user interaction. Action is a very useful means for reproducing an issue. We found that there are issues initiated without actions like "[Without any user's interaction] Songs automatically kept skipping to the next without playing current one." It should be noted that our experiment focused on touchscreen based interaction, which is the most common way of interacting with apps.

Content and context dynamicity

The fourth issue is a contextual factor of the reported trouble. In general, it can be persistent, content-dependent (e.g., web text, name of music), or context-

dependent (e.g., network connections, GPS availability). For a content-dependent case, one participant reported "[Trying to play specific music] If app doesn't play this song, it should have been deleted from the list." It is apparent that this factor is noticeable in a content providing app such as HowAbout or Wikipedia. As an example of a context-dependent case, a participant reported several context-dependent cases as "[Touching a play button in order to play music] the popup message says 'found'. But suddenly it says 'couldn't find' and it again turns to 'found' again!"

Discussion

The analysis that we have conducted will be a useful basis for building a feedback reporting system for end users. We expect that a feedback reporting system would provide optimized methods for reporting on each topic, so that users will be able to compose their feedback more efficiently based on the specific usability issue at hand. For example, issues related to malfunction might require a screen recording support in order to accurately capture the problematic context.

The usability issue topics would be particularly useful to to overcome a feedback overload problem which developers often have to deal with [18]. The feedback reporting system should utilize issue topics to classify and manage the reports so as not to overwhelm developers. For instance, issues belonging to the category of malfunction should be resolved in higher priority than other categories because these are often related to the availability of specific functionality.

The need to give users proper methods to identify the origins of usability issues and type of actions should be considered in designing a feedback reporting system because developers often have difficulty in reproducing a reported issue [14]. It becomes more important when it comes to end users' feedback reporting because they generally lack knowledge of technical terminology to adequately describe the issue. We observed that many users tend to use inexact descriptions such as unclear pronouns or incorrect words. For example, 'Notification bar' (a term for indicating row items located in top bar for Android platform) was described as 'button' by a user; such a vague descriptions or inaccuracies can confuse a developer.

Finally, we think that it would be be helpful to log a problem's content/context dynamicity information automatically so that an accurate description is readily available to create more useful reports. This is because issues containing this attributes are frequently related to a system's internal variable, which cannot make users hard to notice.

Conclusion and Future Work

Based on our findings and data, it is our objective to develop a feedback reporting tool for mobile apps to better support the inefficient transmission of user feedback to developers. In a follow-up experiment, we will study how and why users choose to use different reporting methods for various usability issues. Having knowledge of the reasons behind feedback reporting behaviors would be a useful basis for building and analyzing such improved reporting tools.

Acknowledgements

This research was supported by the Brain Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2015M3C7A1065859).

References

1. Andre, T.S., Rex Hartson, H., Belz, S.M., and Mccreary, F.A. 2001. The user action framework: a reliable foundation for usability engineering support tools. *International Journal of Human-Computer Studies 54,* 1, 107–136.
2. Bano, M. and Zowghi, D. 2013. User Involvement in Software Development and System Success: *A Systematic Literature Review*. *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering*, 125–130.
3. Bano, M. and Zowghi, D. 2013. Users' involvement in requirements engineering and system success. *International Workshop on Empirical Requirements Engineering* (EmpiRE '13), 24–31.
4. Bettenburg, N., Just, S., and Schröter, A. 2008. What makes a good bug report? *Proceedings of the ACM SIGSOFT symposium on Foundations of software engineering*, 308–318.
5. Bettenburg, N., Premraj, R., Zimmermann, T., and Kim, S. 2008. Extracting structural information from bug reports. *Proceedings of the 2008 international workshop on Mining software repositories (MSR '08)*, 27-30.
6. Castillo, C., Hartson, H.R., and Hix, D. 1998. Remote Usability Evaluation: Can Users Report Their Own Critical Incidents? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*, 253–254.
7. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., and Landay, J.A. 2007. MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones. *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys '07)*, 57–70.
8. Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., and Sadeh, N. 2013. Why People Hate Your App — Making Sense of User Feedback in a Mobile App Store. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*, 1276-1284
9. Hartson, H.R., Castillo, J.C., Kelso, J., Kamler, J., and Neale, W.C. 1996. Remote Evaluation: The Network as an Extension of the Usability Laboratory. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*, 228–235.
10. Jeffries, R. and Desurviret, H. 1992. Usability Testing vs. Heuristic Evaluation: Was there a contest? *SIGCHI Bulletin 24,* 4, 39–41.
11. Ko, a. J. and Myers, B. a. 2006. A Linguistic Analysis of How People Describe Software Problems. *Visual Languages and Human-Centric Computing (VL/HCC '06)*, 127–134.
12. Kujala, S. 2003. User involvement: a review of the bene its and challenges. *Behaviour & Information Technology 22,* 1, 1–16.
13. Lotufo, R., Malik, Z., and Czarnecki, K. 2012. Modelling the 'Hurried' bug report reading process to summarize bug reports. *IEEE International Conference on Software Maintenance (ICSM '12)*, 430–439.
14. Mao, B.J., Vredenburg, K., Smith, P.W., and Carey, T. 2005. User-centered design practice. *Communication of the ACM, 48, 3*, 105–109.
15. Mayhew, D.J. 1999. The usability engineering lifecycle. *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '99)*, 147-148.
16. Nielsen, Jakob. 1994. Usability engineering. *Elsevier*.
17. Norman, Donald A. 1986. Cognitive engineering. User centered system design, 31-61.
18. Oh, J., Kim, D., Lee, U., Lee, J., and Song, J. 2013. Facilitating Developer-User Interactions with Mobile App Review Digests. *Proceedings of the SIGCHI*

Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '13), 1809–1814.

19. Paetsch, F., Eberlein, a., and Maurer, F. 2003. Requirements engineering and agile software development. *Proceedings. IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 308–313.
20. Pagano, D. and Bruegge, B. 2013. User involvement in software evolution practice: A case study. *International Conference on Software Engineering (ICSE '13)*, 953–962.
21. Pagano, D. and Maalej, W. 2013. User feedback in the appstore: An empirical study. *IEEE International Requirements Engineering Conference (RE '13)*, 125–134.
22. Rastkar, S., Murphy, G.C., and Murray, G. 2010. Summarizing Software Artifacts: A Case Study of Bug Reports. *International Conference on Software Engineering (ICSE '10)*, 505–514.
23. Susan L. Keenan, Hartson, H.R., Kafura, D.G., and Schulman, R.S. 1999. The Usability Problem Taxonomy: A Framework for Classification and Analysis. *Empirical Software Engineering*, *4*, 71–104.
24. Vredenburg, K., Mao, J.-Y., Smith, P.W., and Carey, T. 2002. A survey of user-centered design practice. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, 471-478.
25. Wilson, C. and Coyne, K.P. 2001. Tracking Usability Issues: To Bug or Not to Bug? *Interactions*, 15–19.
26. Schusteritsch, R., Wei, C. Y., and LaRosa, M. 2007. Towards the perfect infrastructure for usability testing on mobile devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI EA '07)*, 1839-1844.
27. Kjeldskov, J., Skov, M. B., Als, B. S., & Høegh, R. T. 2004. Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. *Mobile Human-Computer Interaction – MobileHCI 2004,* 3160, 61-73.
28. Barry Brown, Moira McGregor, and Donald McMillan. 2014. 100 days of iPhone use: understanding the details of mobile device use. *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14)*, 223-232.