How Can Software Testing be Improved by Analytics to Deliver Better Apps?

Conference or Workshop Item

How Can Software Testing be Improved by Analytics to Deliver Better Apps?

Julian Harty
Computing and Communications, Open University
Supervisors: *Arosha Bandara & Yijun Yu*
Milton Keynes, United Kingdom
ORCID 0000-0003-4052-0054

***Abstract*—Many consider software testing to be necessary yet given the nature of testing and practical project constraints it cannot be comprehensive or complete. The resulting software has bugs including those that affect some users. Analytics of usage of apps may help illuminate testing that has been performed on existing releases and also inspire improvements to future testing. The Android ecosystem provides unusually rich analytics tools for developers of apps released in Google Play so my research focuses on this ecosystem to evaluate several analytics tools including Google Play Console, Android Vitals, which are integrated into the platform and the operating system, together with additional mobile analytics offerings from Google and Microsoft.**

***Index Terms*—Android, Android Vitals, Apps, Crashlytics, Firebase, Mobile, Software Analytics, Software Testing**

I. Introduction

My research aims to understand the relationship between and impact of different information sources on how developers and testers understand issues with mobile apps. It includes connecting and comparing data from these sources and ways information from one source can usefully inform the development and testing to deliver better apps, where ‘better’ includes reliability and performance of the apps in pre-launch testing and in use by end users.

Testing is one of several ways people can obtain information about qualities of software, there are other sources including automated tools and feedback from production usage. Software testing can be measured in various ways including the progress that was made (*e.g.* what did we manage to test?) and the results that were obtained (*e.g.* bugs found using exploratory testing of Android apps [1]). There are relationships between software testing and assessing the quality of software.

There are ongoing debates in industry and academic research that tries to compare the effectiveness of testers and techniques [2], for instance asking *”Is Carmen Better than George?”* [3], [4], and ongoing research to improve test processes [5]. There does not appear to be any commonly agreed measures to assess the efficacy of software testing in practice.

App Stores introduced another paradigm to software development practices by connecting users and development teams [6]. With Android, Google collects usage data including several quality metrics under the banner of stability metrics, which include crashes, non-responsive behaviour, known as Application Not Responding (ANR), battery usage and other performance-related data. This data provides analytics from the platform perspective. They describe this as *”Android Vitals”* and integrated it in the Google Play Console, which is aimed at the developers of a given app. Google confirms they use the results they calculate to assess the quality of Android apps and that poor results materially affect the app’s discoverability, get more 1 star ratings, *etc.* [7]

Software analytics provides insight into software development practices [8], including testing [9] and usage. These insights may help improve these practices and improve the software being developed if it can provide relevant answers to key questions asked by practitioners: for instance for some of the 145 questions for data scientists in software engineering (top categories were, development practices: 28, testing practices: 20, and evaluating quality:16) [9]; and to address the information needs for software development analytics [10].

II. Problems to be addressed

My research aims to address five related problems:

1) To provide actionable data, using analytics, that can be used to assess the testing of apps in order to illuminate areas where the testing can be productively improved. 2) To apply usage analytics as a source of information to inspire testing based on patterns of usage by end users. 3) To evaluate how well testing can reproduce and localise issues, when quality problems are reported by analytics tools. 4) To connect and join data from testing by the development team (including software testers) to usage and quality-related metrics from end user usage of the apps in ways that respect privacy objectives such as GDPR regulations in Europe. 5) As my research has identified flaws both within and between various analytics tools (particularly those provided by Google) to test analytics tools and report inconsistencies, bugs and flaws to enable development teams to be forewarned and tool providers to consider improving their tools.

III. Research hypothesis

Analytics of data pertaining to usage of apps can help development teams to improve their testing (their process) and the product they create (the app) which can lead to greater user satisfaction.

Some issues may be detected by several sources (figure 1), comparing and contrasting these sources may also help the teams to choose the most appropriate information source for particular types of flaw. It may be viable to build on reliability engineering such as [11] [12]. Null Hypothesis: software testing does not need analytics to improve apps.

IV. Expected contributions of the research

The primary contribution of my research is to provide an understanding of how software analytics can be used to complement testing activities to improve the quality of mobile applications. We chose the Android ecosystem as it is extremely popular, very diverse as a platform, with a rich seam of analytics tools; but the findings can be extended to other environments.

Secondary objectives include several case studies, open-source projects where changes and results can be analysed in further research [13], [14], and open-source utilities that automatically download and preserve otherwise transient data imparted by the GUI of analytics tools to facilitate comparisons, bug analysis and reporting, and so on [15], [16].

V. Research approach

To obtain analytics data to measure the perceived quality of Android apps using information available to the development and testing teams of those apps, we use trusted, freely available professional analytics tools from several sources including Google and Microsoft, and deploy Empirical Software Engineering approaches.

Our research spans a variety of popular open-source Android apps: eighteen for the Kiwix project [17], two (PocketCode and PocketPaint) for the Catrobat project [18], and a VPN client for the eduVPN project [19]. We have also created several Android apps to help evaluate Android Vitals and Google Play Console under controlled conditions.

- **Internal Developer** For the Kiwix project I am an integrated participant in the engineering team, the aim was to perform Action Research where worked directly as part of the development team to find, test and fix issues reported using Android Vitals.
- **Internal Tester** The Catrobat team actively uses a complete set of tools, including static analysis, sophisticated continuous builds and automated testing, Android Vitals and [Google] Fabric's Crashlytics analytics library. This project offers the scope to compare testing, analytics and static analysis data to determine how this differs for a given set of bugs. For the Catrobat project I coach the engineering team and assisted in some of the bug investigation and analysis, however I do not directly work on the code.
- **External Contributor** EduVPN is a recent collection of apps, where my focus is on improving the use of software analytics and establishing end-to-end trustworthy automated testing and continuous builds.
- **External Observer** I also obtain reports, data while interviewing developers of a variety of popular Android apps in several categories of the Google Play store.

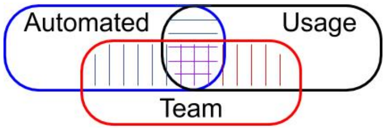

Fig. 1: Venn diagram of information from various sources

VI. Summary of results to date

The results have proven to enable material improvements in the reported qualities of various Android apps, developed by several otherwise unconnected development teams. Aspects of the research have been published in 2019 [20] [21]. The crash rate was reduced for Kiwix by a factor of 10+ over a series of releases; for the PocketCode app has already been reduced by a factor of 2 (work started more recently for this project and further improvements are expected).

Only a minority of crashes reported in the analytics reports could be reproduced during testing by the development team, yet they were able to fix most of these as confirmed by analytics reports for releases incorporating those fixes. This may be an example where debugging without testing applies in some cases [22]. I intend to investigate the reasons why our testing could not reproduce various crashes, as others were able to reproduce reported crashes automatically in other apps [23]. I have requested access to the CrashScope tool used in that research [24].

As part of early sharing of the results, in 2015 I was the lead author of *The Mobile Analytics Playbook: A practical guide to better testing* [25]. HP sponsored several print runs (totaling 5,000+ copies) and two editions of the book.

In 2019, Google's engineering team for Android Vitals reviewed some more recent findings and accepted the validity of various bugs. They then requested a complete report of the findings to date so they can investigate flaws and bugs in that project. While they have acknowledged some of the bugs they stated they are highly unlikely to provide complete feedback or acknowledge the work as they make changes and improvements to their product offering.

VII. Evaluation and dissemination plan

From a practical perspective, the evaluation is through the quality improvements for Android apps who have used the concepts identified in my research. My work focus on addressing the following questions: Will the improvements be sustained and the quality maintained throughout various releases? Will they address degradations in quality quickly and effectively? From a research perspective, will additional researchers get involved in the field and build on and extend the work in these areas?

The dissemination includes my PhD thesis, of course, and possibly the authorship of another book aimed at software developers and testers to complement and improve on several books I have co-authored so far. The research will also be shared at academic conferences and workshops, and potentially in one or more journal papers. It will also be presented at industry focused venues.

References

[1] M. Souza, I. K. Villanes, A. C. Dias-Neto, and A. T. Endo, "On the exploratory testing of mobile apps," in *Proceedings of the IV Brazilian Symposium on Systematic and Automated Software Testing*, 2019, pp. 42–51.

[2] J. Itkonen, M. V. Mantyla, and C. Lassenius, "Defect detection efficiency: Test case based vs. exploratory testing," in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. IEEE, 2007, pp. 61–70.

[3] A. Borg, C. Porter, and M. Micallef, "Poster: Is Carmen better than George? testing the exploratory tester using HCI techniques," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2. IEEE, 2015, pp. 815–816.

[4] M. Micallef, C. Porter, and A. Borg, "Do exploratory testers need formal training? an investigation using HCI techniques," in *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2016, pp. 305–314.

[5] W. Afzal, S. Alone, K. Glocksien, and R. Torkar, "Software test process improvement approaches: A systematic literature review and an industrial case study," *Journal of Systems and Software*, vol. 111, pp. 1–33, 2016.

[6] A. AlSubaihin, F. Sarro, S. Black, L. Capra, and M. Harman, "App store effects on software engineering practices," *IEEE Transactions on Software Engineering*, 2019.

[7] (2019) Improve your app and game quality with android vitals (google i/o'19). [Online]. Available: https://www.youtube.com/watch?v=RZSR4a1SBCY

[8] R. P. Buse and T. Zimmermann, "Analytics for software development," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, 2010, pp. 77–80.

[9] A. Begel and T. Zimmermann, "Analyze this! 145 questions for data scientists in software engineering," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 12–23.

[10] R. P. Buse and T. Zimmermann, "Information needs for software development analytics," in *Proceedings of the 34th international conference on software engineering*. IEEE Press, 2012, pp. 987–996.

[11] P. Frankl, D. Hamlet, B. Littlewood, and L. Strigini, "Choosing a testing method to deliver reliability," in *Proceedings of the 19th international conference on Software engineering*, 1997, pp. 68–78.

[12] P. G. Frankl, R. G. Hamlet, B. Littlewood, and L. Strigini, "Evaluating testing methods by delivered reliability [software]," *IEEE Transactions on Software Engineering*, vol. 24, no. 8, pp. 586–601, 1998.

[13] Commercetest Limited. (2019) Androidcrashdummy github project. Last checked on 2020-02-03. [Online]. Available: https://github.com/ISNIT0/AndroidCrashDummy

[14] (2019) Zipternet github project. Last checked on 2020-02-03. [Online]. Available: https://github.com/ISNIT0/zipternet/

[15] Commercetest Limited. (2019) Android stability analysis source code on github. Last checked on 2020-02-03. [Online]. Available: https://github.com/commercetest/android-stability-analysis

[16] ——. (2019) Vitals scraper source code on github. Last checked on 2020-02-03. [Online]. Available: https://github.com/commercetest/vitals-scraper

[17] (2019) Kiwix lets you access free knowledge – even offline. [Online]. Available: https://www.kiwix.org/

[18] (2019) Catrobat: Free educational apps for children and teenagers. [Online]. Available: https://www.catrobat.org/

[19] (2019) eduvpn - home. [Online]. Available: https://www.eduvpn.org/

[20] J. M. A. Harty, "Google play console: Insightful development using android vitals and pre-launch reports," in *MOBILESoft 2019*, IEEE. Montreal, QC, Canada: IEEE, 2019, pp. 62 – 65. [Online]. Available: http://oro.open.ac.uk/61066/

[21] J. Harty and M. Müller, "Better android apps using android vitals," in *Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics*, 2019, pp. 26–32.

[22] W. Ghardallou, N. Diallo, A. Mili, and M. F. Frias, "Debugging without testing," in *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2016, pp. 113–123.

[23] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk, "Automatically discovering, reporting and reproducing android application crashes," in *2016 IEEE international conference on software testing, verification and validation (icst)*. IEEE, 2016, pp. 33–44.

[24] ——, "Crashscope: A practical tool for automated testing of android applications," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2017, pp. 15–18.

[25] J. Harty and A. Aymer, *The Mobile Analytics Playbook: A Practical Guide to Better Testing*. Hewlett Packard Enterprise, 2015. [Online]. Available: https://books.google.co.uk/books?id=Z1_ujgEACAAJ