An Industry Case Study: Methodology Application to the Reviewing Process on Android Releases Homologation

Pedro Ivo Pereira Lancellotta[1], **Heryck Michael Dos Santos Barbosa**[1], **João Gabriel Castro Santos**[1], **Klirssia Matos Isaac Sahdo**[1], **Janislley O. De Sousa**[1]

[1]Sidia Institute of Science and Technology - SIDIA, Manaus - AM, Brazil

{pedro.lancellotta,heryck.barbosa,joao.castro}@sidia.com

{klirssia.sahdo,janislley.sousa}@sidia.com

Abstract. *Quality is the main factor at any process, in the software context is not different. The Android Operational System has rigid quality standards. Device producers who choose to use Android, must comply with rules and contract requirements. The quality assurance concerning these rules on Android release homologation process is challenging, due to the great amount of test artifacts to be analyzed at the end of process. In this paper, we propose a methodology to improve the review step for Android releases homologation. As a preliminary result, we identified 52,84% of improvement using the proposed approach. In advance of this research, we intend to implement the automation and evaluate it's impact in a context of software quality product in our company.*

1. Introduction

The Quality process is the main factor at every scenario and provided service. In the software quality context is no different. The quality performs a fundamental role assuring that a product is stable and safe for use [Mishra and Otaiwi 2020]. The test reviewing process is a main step for software quality, the adoption of this method presents gains due to error removal in previous phases and time reduction in future maintenance processes [Hassan et al. 2019]. The appliance of a quality process on huge software is still a challenge. The Android Operational System, which is one of the biggest software of the world, has an adhesion about 71,93% on Mobile devices in January 2021 [Laricchia 2022], it has some rigid quality standards to be attended. In this scenario, the Original Equipment Manufacturers (OEMs) develops their private models on its own industries [Possemato et al. 2021]. In case these manufacturers choose to use Android and services as Google Mobile Service (GMS) on their devices, they must comply with an amount of rules and requirements defined in contract, which is called Mobile Application Distribution Agreement (MADA). Thus, OEMs perform contracts with Google, in order to stipulate and attend to necessary conditions to maintain Android updates.

By manufacturing mobile solutions using Google's operational system, the companies mass product devices or they offer them to carriers, so on their side, they decide regarding customization according to region or market niche, whose build the system firmware which can be applied to the models, also known as builds or releases [Alure and Puri 2021]. To apply these particular customizations, carriers also have to validate their quality in order to guarantee their demand and public acceptance, and by doing so, during the steps of homologating this software product, a great amount of files artifact (over 500 files) are generated, demanding significant efforts to validation and analysis. In accordance, this homologation process will be the study object of this paper, with the main objective to make this process more agile and assertive.

This paper is structured as follows: section 2 introduces related works; section 3, describes our approach and technical details; and finally, section 4, presents conclusions and future works suggestion.

2. Related Works

There are not many researches related to Software Engineering covering the steps of homologating releases, even less about Android system. Amongst the existent studies, we would like to highlight two: An automation to detect problems on Android Builds using Screenshot analysis, replacing the old manual task performed by testers [de Figueiredo et al. 2022]; and, a proposal for homologation improvement of Android releases, implementing a new process called Internal Review, its goal was to reduce errors on test results [Bernardon et al. 2022]. Both papers are referring to the quality assurance in a complex result scope, when a great amount of test artifacts are generated for review, which is a typical scenario for software quality professionals. Our proposal aim to expand these solutions, without limiting only by screenshot analysis, but covering all the test artifacts and improving the proposed Internal Review process by a new methodology.

3. Methodology to Android Releases Homologation

3.1. Test Artifacts

As soon as the product is ready and the Android operational system embedded, a great variety of preset tests by Google are required. These tests are all based on Tradefed - Trade Federation, consisting in a continuous framework developed to do tests on Android devices (Table 1). We consider these test suites as the automated parte of the scope. There are some manual test scope defined and predetermined by specific OEMs accordingly.

Table 1. Android mobile homologation process tests

Test Suites [1]	Validation
CTS	Compatibility Test Suite
CTS Verifier	Complement to CTS with Manual APIs
VTS	Reliability and System Conformity Improvement
GTS	GMS Applications and Google contractual presets
STS	Security Test suite for CTS
Black Box Test	Specific tests for Exclusive Applications to the Producer

3.2. Results Inconsistency

After generating testing results, they are checked in order to find out nonconformities that goes against specified rules and attributes defined for them. In this verification, one or more nonconformities may be found and need to be fixed. Whenever this happen, we

[1] CTS - Compatibility Tests Suite; VTS - Vendor Teste Suite; GTS - GMS Test Suite; STS - Security Teste Suite

receive what is called Feedback where it is requested to fix or explain these discordances. If there is no fix or explanation for them, Google will reject that software as it does not comply to the requirements. Feedback process may cause delays on the software homologation and in order to avoid compromising schedule predefined contractually with carriers and Google, which negatively affects all involved companies and deteriorating user's experience regarding the product quality and expectations.

Due to urgencies, deadline for new requirements and other scheduled demands it is most likely to occur inconsistencies that might pass by unnoticed by engineers. The Figure 1 presents the numbers of our company regarding the quantity of homologated releases on feedback perspective at the first half of 2022 without using our methodology.

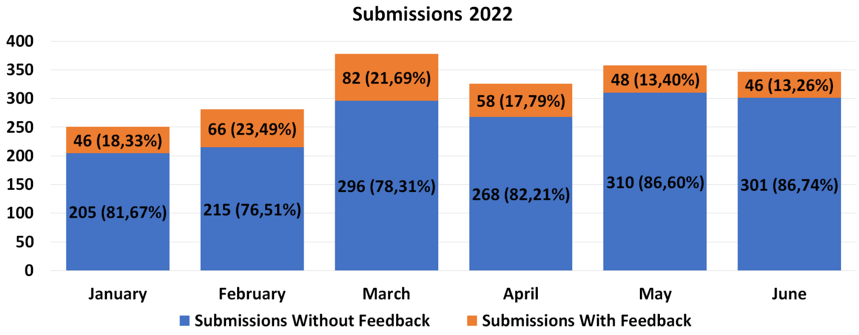

Figure 1. Submissions with and without feedback - first half 2022.

3.3. Our Approach: An Automation for Test Reviewing

On the presented context, related to the test artifacts inconsistencies recurrence observed along 2022, we proposed in this paper an improvement on the Android releases homologation process: an automation to review test artifacts and compare current values with expected values. In synthesis, the automation will have a set of rules to validate informations contained on test artifacts files, warning the tester through a report, any identified inconsistencies. Once the nonconformities are informed, it is expected that the tester to solve every incongruity and in order to have them corrected, aiming to guarantee that all files are able to be homologated as soon as they are submitted. The Figure 2 represents a visual flow of the Methodology proposed in this paper. This methodology was applied during a reviewing process on android releases homologation. It was compared two cases: (a) manual process and (b) methodology approach. The results shown that a set of testers in case (a) found an average of 5,66 wrong information on test artifacts from a total of 12 introduced errors. In case (b), we had an improvement, reaching the totality of errors, with 52.84% more chance to find the wrong information.

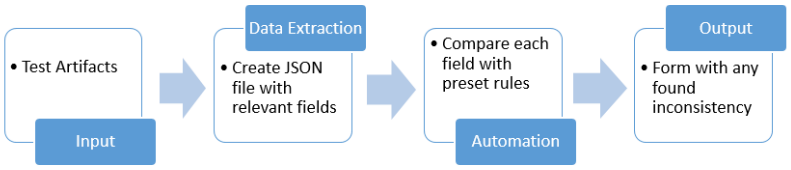
Test Artifacts
Input
Data Extraction
Create JSON file with relevant fields
Compare each field with preset rules
Automation
Output
Form with any found inconsistency

Figure 2. Methodology process.

4. Conclusion and Future Works

This paper presented a new methodology approach to improve the test artifacts reviewing in the end of homologation process on Android releases. Currently, there is no automated mechanism in our company that fulfills this scope, so the proposed methodology becomes a support agent in order to mitigate observed inconsistencies. As a preliminary result of our approach evaluation, we identified an improvement of 52,84% chance to detect inconsistencies by applying our methodology. We noticed that with the previous identification of these feedbacks, we gained approximately 1 day on the approval step of these projects, culminating in a higher quality and higher efficiency. In other words, this solution have helped to reduce costs by collaborating with strategic points, considering to maximize quality by decreasing human efforts. As future work, we intend to improve our methodology approach by creating an automate tool, evaluating its implementation in our company environment.

5. Acknowledgment

The authors would like to thank the Sidia Institute of Science and Technology which made the execution of this research possible through their laboratory facilities and equipments. This work was partially supported by Samsung Eletrônica da Amazônia Ltda., under the auspice of the Informatic Law no 8.387/91

References

Alure, S. and Puri, R. (2021). Firmware designing for android mobile. *INTERNATIONAL JOURNAL*, 5(12).

Bernardon, C., Lemos, D., Garcia, M., Souto, T., and Bonifacio, B. (2022). Improving the software homologation process through peer review: An experience report on android development environment. *International Journal of Computer and Systems Engineering*, 16(3):54–60.

de Figueiredo, L. P., Gomes, J. P., de S. Santos, F., Queiroz, G. M., Giuntini, F. T., and Sales, J. E. (2022). An automatic approach to detect problems in android builds through screenshot analysis. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 926–932.

Hassan, M., Hussain, M., Irfan, M., et al. (2019). A policy recommendations framework to resolve global software development issues. In *2019 International Conference on Innovative Computing (ICIC)*, pages 1–10. IEEE.

Laricchia, F. (2022). Market share of mobile operating systems worldwide 2012-2022. *Statista*.

Mishra, A. and Otaiwi, Z. (2020). Devops and software quality: A systematic mapping. *Computer Science Review*, 38:100308.

Possemato, A., Aonzo, S., Balzarotti, D., and Fratantonio, Y. (2021). Trust, but verify: A longitudinal analysis of android oem compliance and customization. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 87–102. IEEE.