Release Practices for iOS and Android Apps

Daniel Domínguez-Álvarez
IMDEA Software Institute, Spain
University of Verona, Italy
daniel.dominguez@imdea.org

Alessandra Gorla
IMDEA Software Institute, Spain
alessandra.gorla@imdea.org

ABSTRACT

We conduct a preliminary study on the practices of releasing apps for the two major mobile platforms: iOS and Android. We select the most popular applications on the official stores, and we retrieve all the releases of such apps to understand how often developers make releases on each platform. Our study aims to highlight possible differences on the release practices for the same application on iOS and Android. We observe that developers tend to publish new releases more often on the Android platform than on iOS, and most of the times the development on the two platforms is not aligned at all.

CCS CONCEPTS

• **Software and its engineering → Software version control**; *Software libraries and repositories.*

KEYWORDS

Android apps, iOS apps, Release engineering

ACM Reference Format:
Daniel Domínguez-Álvarez and Alessandra Gorla. 2019. Release Practices for iOS and Android Apps. In *Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics (WAMA '19), August 27, 2019, Tallinn, Estonia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340496.3342762

1 INTRODUCTION

Mobile apps are more and more popular, to the point that there are now more users of mobile devices than of desktops. In the mobile scene, Android is by far the operating system with the largest share of the market (over 80% in 2018). Despite its smaller market share, most publishers offer their apps for the iOS platform as well. This is because in some markets (e.g. Europe and North America) iOS has a larger share of the market, despite not being as popular as Android.

Most research work on mobile applications focused on Android. There is little work on the analysis of iOS, mostly because it is a much harder platform to analyze. Our research agenda for the future is to analyze apps on both platforms, as we believe that findings on one platform cannot be generalized to the other one.

In this paper we focus on the release practices that developers follow on Android and iOS. We aim to understand if there are significant differences in these platforms. In order to study how apps differ, we need a dataset of iOS and Android apps. None of the official mobile stores allow to retrieve older releases. We therefore rely on two existing datasets: ApkPure, which is an alternative store that shares both iOS and Android apps, and Tacyt, a service from Telefonica that collects million apps for analytics. We select the top 50 apps on the Android apps, and we analyze the data of all the available releases. We first study how practices differ in Android and iOS in terms of release frequency. We observe that Android users experience more updates than iOS users. We then analyze if there is a big variance in the release frequency or rather if developers follow regular release schedules. We observe that most apps, even if they are very popular, do not follow regular release practices. Only few of them follow regular releases, mostly weekly bases. We also analyze if releases are somehow aligned between the two platforms, in terms of release number. This is not the case most of the times, and the few apps that follow this practice are the ones that also regularly release software.

The remainder of the paper is structured as follows: Section 2 describes the dataset and how we obtained it. Section 3 presents the observations on our dataset regarding the release practices on Android and on iOS. Section 4 presents the related work, and Section 5 concludes the paper and discusses the related work.

2 DATASET COLLECTION

Neither Apple nor Google officially allow users to download older releases of the applications listed in their official stores. We therefore resort to alternative services that regularly crawl these stores and keep older releases. ApkPure [1] is essentially an alternative store for Android and iOS apps, and allows users to download any release of the application. Tacyt [2] is a service from Telefonica that collects million apps for data analytics.

We select our dataset as follows: We first identify the 50 most popular applications on the Spanish market for Android in June 2019. We then crawl both Apkpure and Tacyt aiming to download all the available releases for both Android and iOS in the last two years. We use the package name (e.g. com.whatsapp) to retrieve Android apps, while we use the application name (e.g. WhatsApp Messanger) to retrieve iOS apps. This is because most of the times the same app does not have the same package name on both platforms (for instance, WhatsApp is net.whatsapp.whatsapp on iOS). In order to crawl ApkPure and Tacyt, we developed our own crawlers, and we later manually validated the quality of the data.

[1] http://apkpure.com
[2] http://tacyt.elevenpaths.com

WAMA '19, August 27, 2019, Tallinn, Estonia

ACM ISBN 978-1-4503-6858-2/19/08…$15.00
https://doi.org/10.1145/3340496.3342762

Some apps are only available on one platform (Android and not iOS), or may be actively maintained only on one platform. We discard apps that do not have releases in both iOS and Android in the last two years. This process leaves us with 35 applications and 2928 releases in total.

3 PRELIMINARY OBSERVATIONS

The goal of our preliminary study is to see whether there are significant differences in the release practices of the same applications on the two most popular mobile app platforms, i.e. Android and iOS platform. The long term plan is to assess whether the same or different teams take care of the development of the same app on different platforms, and whether somehow the development follows the same path on iOS and Android. We start investigating these research questions looking at metrics that are easy to compare automatically, for instance the release frequency.

3.1 Release Frequency

Figure 1 plots the number of releases for each app on the two different platforms. We see that on average for the apps in our dataset the Android platform seems to offer more releases that the iOS platform. Most apps produce between 25 and 50 releases on Android, while these numbers are lower on iOS.

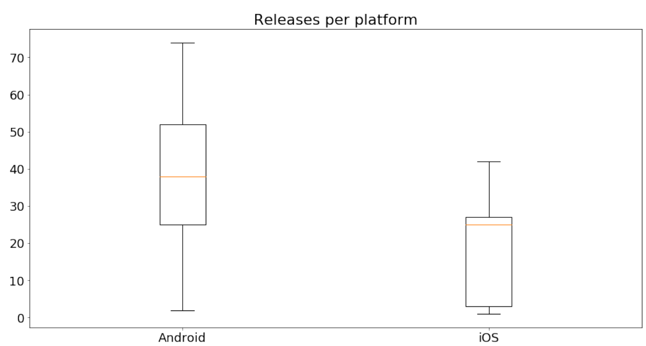

Figure 1: Number of releases per app per Android (left boxplot) and iOS (right boxplot).

Figure 2 plots the same data, but on a monthly timeline starting in January 2018. The blue bars are the cumulative number of releases for Android, while the orange bars are the cumulative number of releases for iOS. This plot confirms the observations of the previous one: developers tend to release Android apps more often than iOS apps. The aggregated numbers show the general trends on a specific platform. However, we also want to compare the release frequency per application, and see if we notice any remarkable difference between the two platforms. Figure 3 shows the number of days between two subsequent releases for an app on Android. Figure 4 shows the same information for the iOS platform.

We see that indeed on Android the release cycles tend to be mostly weekly or monthly based most of the times. Apps with longer release cycles are applications that have limited competition, such as banks (CaixaBank and Bankia), or apps that are periodically very popular, but may be unused most times during the rest of the year (e.g. Agencia Tributaria is the app used for tax declaration in

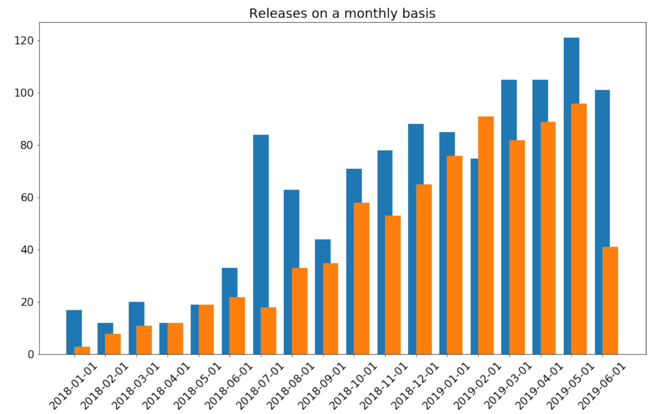

Figure 2: Releases of Android apps (blue bars) and iOS apps (orange bars) on a monthly bases.

Spain, and it is thus heavily used only once a year). It is surprising on the other hand that HBO has similarly long release cycles on Android, especially given that its competitor Netflix has a much shorter release cycle.

On iOS we see more variance in the release cycle between apps. Some of them are regularly updated every few days (e.g Instagram, Netflix, Pinterest, TikTok and Outlook), but others are released less often. Apps that have frequent release cycles on one platform have frequent release cycles also on the other one. Given this insight, we investigate if development for these apps is aligned across the two platforms, or rather it is completely independent.

3.2 Release Alignment in Android and iOS

All the apps we analyzed seems to use semantic versioning. We manually compared release numbers and dates for each app on the two platform and we found that:

- Apps that have regular weekly release cycles (e.g. Pinterest, Instagram, TikTok) are pretty aligned in release numbers and dates.
- We found that some apps are completely aligned (e.g. Mi Fit) and we believe it might be because developers use common framework to build hybrid apps.
- Some apps, e.g. Booking, follow the same version practice, but the release numbers differ of a few months between the two platforms.
- Most apps are completely misaligned. The most extreme case is of Amazon apps (Amazon Prime Video and Shopping). In iOS there are several changes in the major releases, while the Android releases are stuck with the same major and minor versions for over two years.

4 RELATED WORK

This paper relates to two categories of research works: 1) studies on different mobile platforms (especially iOS and Android), and 2) studies on release engineering.

Studies on Android and iOS. While most work on the analysis of mobile apps focuses on Android, there are a few related work

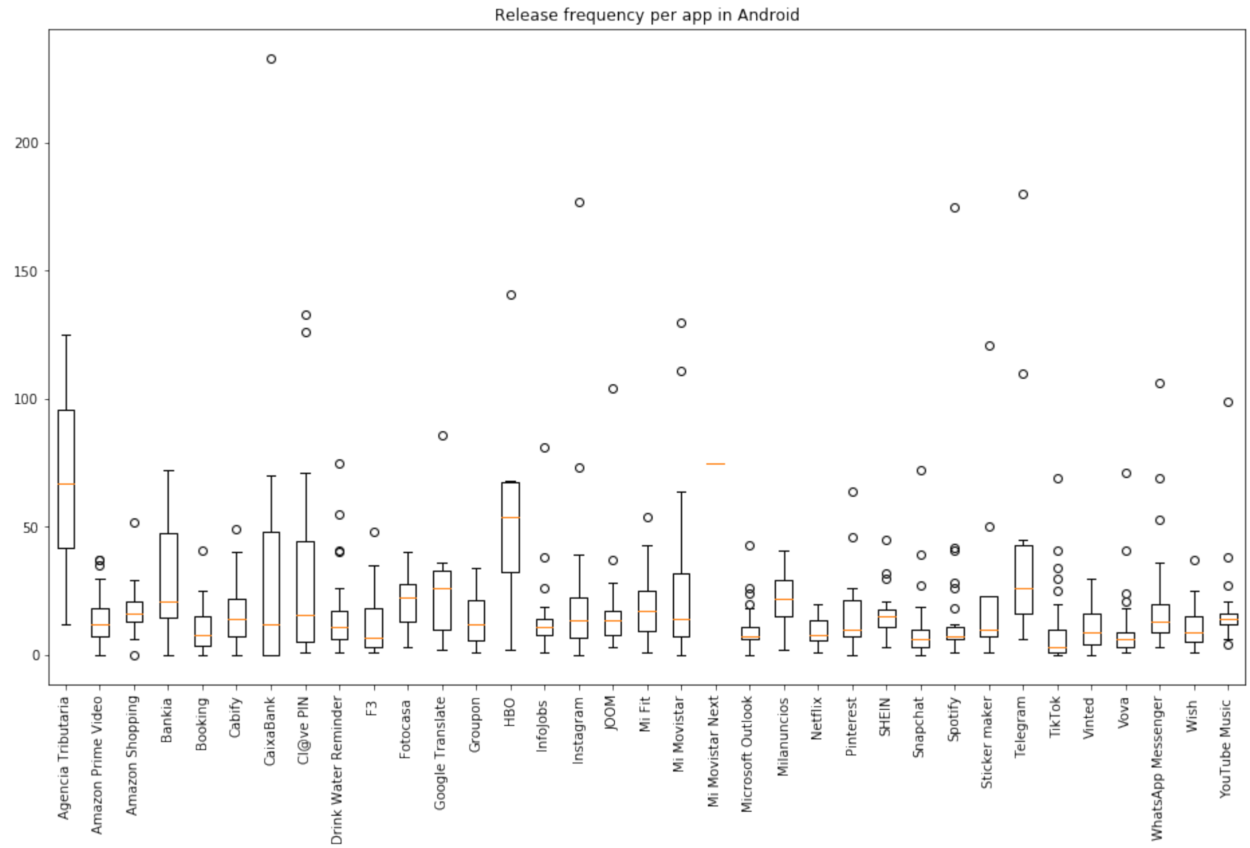

Figure 3: Release frequency, in number of days between each release, of each app on Android.

that compare Android and iOS along several aspects. Gronli et al. compare the Android and iOS mobile platforms according to their features, aiming to assess their strengths and weaknesses for development [8]. Similarly, Goadrich et al. compare different mobile development platform looking into their main features, but their goal is to assess which one is best to use for teaching [7].

Other researchers focus on several quality aspects of the two platforms. Zhou et al. analyze the differences in bugs and bug-fixes in Android and iOS apps [17]. Wang et al., instead, study StackOverflow on API-related discussions for the iOS and Android platforms to highlight respective usage difficulties [15]. Ali et al. focus instead on users, and study how the same app differs in rating, prices and reviews in different mobile platforms [1]. Researchers also look at non functional properties of the iOS and Android platforms. Liu et al. analyze and compare the performance of Android and iOS devices when accessing Internet streaming services [11]. Other researchers compare the security on each platform, looking at possible attacks to Web components [12], assessing the feasibility of man in the middle attacks [9], or identifying potentially harmful libraries in Android and iOS apps [5]. Other works compare the two platforms in terms of privacy and protection of minors. Chen et al. propose a technique to systematically uncover the extent and severity of unreliable maturity ratings for mobile apps [6], while Benenson et al. survey more than 700 students to assess their awareness in terms of privacy violations when using apps on Android and iOS [2]. None of the mentioned works study release engineering on Android and iOS platform as we do in this paper.

Studies on Release Engineering. Many researchers studies release engineering, but they focused on a single platform. Calciati et al., for instance, study how behavior changes across different releases of the same Android app [3, 4]. Martin et al. analyze a large number of app releases and their corresponding reviews from users. They observe that over one third of the releases cause a change in user ratings [13]. Xia et al. instead, use machine learning techniques to effectively predict mobile app releases that are more likely to crash [16]. Last but not least, Khomh et al. show how shorter release cycles lead to better quality perceived by users [10]. This study, however, does not consider any mobile platform. The closest work to ours is by Nayebi et al., as they also aim to study the release practices in mobile development. Their contribution complements ours, as their analysis is based on surveys of developers and users rather than on actual data retrieved from app stores [14].

5 CONCLUSIONS AND FUTURE WORK

In this paper we presented our preliminary study on the practices of release engineering on the two main mobile platforms, Android and iOS. Looking at the release frequency of the most popular apps,

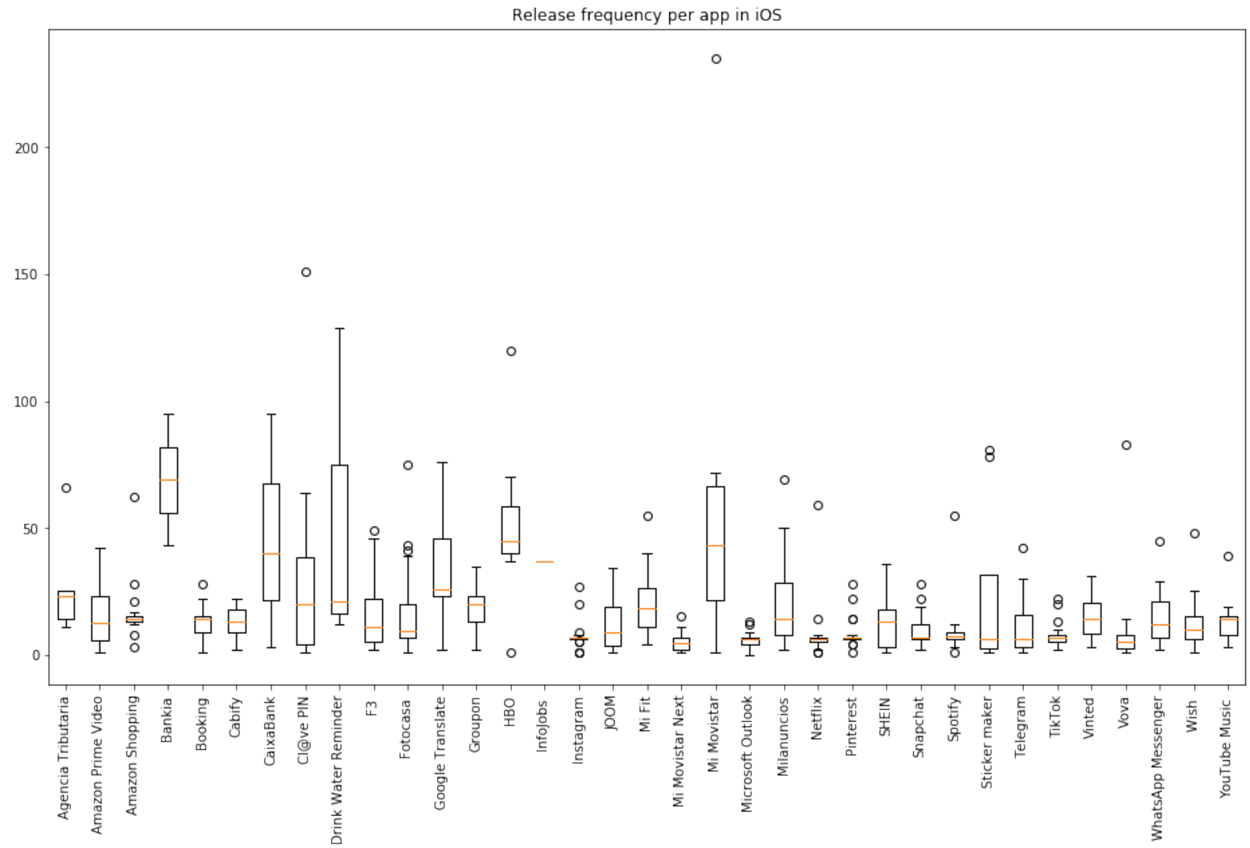

Figure 4: Release frequency, in number of days between each release, of each app on iOS.

we observe that developers tend to create more releases on the Android platform than on iOS. We also observe that most apps have completely independent development processes, as our manual analysis could not easily align releases of the same app on the two platforms. For the future we plan to continue analyzing apps on these two ecosystems and focus on their differences, both with respect to software engineering practices, but mostly on security and privacy aspects. Our long term plan is to assess whether the same app, released on two platforms, behaves the same, as most users would expect.

ACKNOWLEDGMENTS

This work was supported by the italian project ATEN by Fondazione Cariverona, by the Spanish Government through the SCUM grant RTI2018-102043-B-I00 and the project DEDETIS, and by the Madrid Regional projects N-Greens Software (n. S2013/ICE- 2731), BLOQUES and MadridFlightOnChip.

REFERENCES

[1] Mohamed Ali, Mona Erfani Joorabchi, and Ali Mesbah. 2017. Same app, different app stores: A comparative study. In *MobileSoft 2017*. 79–90.
[2] Zinaida Benenson, Freya Gassmann, and Lena Reinfelder. 2013. Android and iOS users' differences concerning security and privacy. In *CHI 2013*. 817–822.
[3] Paolo Calciati and Alessandra Gorla. 2017. How Do Apps Evolve in Their Permission Requests? A Preliminary Study. In *MSR 2017*. 37–41.
[4] Paolo Calciati, Konstantin Kuznetsov, Bai Xue, and Alessandra Gorla. 2018. What did Really Change with the new Release of the App?. In *MSR 2018*. 142–152.
[5] Kai Chen, Xueqiang Wang, Yi Chen, Peng Wang, Yeonjoon Lee, XiaoFeng Wang, Bin Ma, Aohui Wang, Yingjun Zhang, and Wei Zou. 2016. Following devil's footprints: Cross-platform analysis of potentially harmful libraries on android and ios. In *IEEE S&P*. 357–376.
[6] Ying Chen, Heng Xu, Yilu Zhou, and Sencun Zhu. 2013. Is this app safe for children?: a comparison study of maturity ratings on Android and iOS applications. In *WWW 2013*. 201–212.
[7] Mark H Goadrich and Michael P Rogers. 2011. Smart smartphone development: iOS versus Android. In *SIGCSE 2011*. 607–612.
[8] Tor-Morten Grønli, Jarle Hansen, Gheorghita Ghinea, and Muhammad Younas. 2014. Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS. In *AINA 2014*. 635–641.
[9] John Hubbard, Ken Weimer, and Yu Chen. 2014. A study of SSL proxy attacks on Android and iOS mobile applications. In *CCNC 2014*. 86–91.
[10] Foutse Khomh, Tejinder Dhaliwal, Ying Zou, and Bram Adams. 2012. Do Faster Releases Improve Software Quality?: An Empirical Case Study of Mozilla Firefox. In *MSR 2012*. 179–188.
[11] Yao Liu, Fei Li, Lei Guo, Bo Shen, and Songqing Chen. 2013. A comparative study of android and iOS for accessing internet streaming services. In *PAM 2013*. 104–114.
[12] Tongbo Luo, Xing Jin, Ajai Ananthanarayanan, and Wenliang Du. 2012. Touchjacking attacks on web in android, ios, and windows phone. In *FPS 2012*. 227–243.
[13] William Martin, Federica Sarro, and Mark Harman. 2016. Causal Impact Analysis for app Releases in Google Play. In *FSE 2016*. 435–446.
[14] Maleknaz Nayebi, Bram Adams, and Guenther Ruhe. 2016. Release Practices in Mobile Apps -– Users and Developers Perception. In *SANER 2016*. 552–562.
[15] Wei Wang and Michael W Godfrey. 2013. Detecting api usage obstacles: A study of ios and android developer questions. In *MSR 2013*. 61–64.
[16] Xin Xia, Emad Shihab, Yasutaka Kamei, David Lo, and Xinyu Wang. 2016. Predicting crashing releases of mobile applications. In *ESEM 2016*. 29:1–29:10.
[17] Bo Zhou, Iulian Neamtiu, and Rajiv Gupta. 2015. A cross-platform analysis of bugs and bug-fixing in open source projects: desktop vs. Android vs. iOS. In *EASE 2015*.