Flutter and iOS Application Comparison

An Empirical Metric Analysis of Performance and User Experience

Bachelor Thesis

submitted by:	Philip Krück
Date of Birth:	04.11.1998
Matriculation Number:	3938
Company Supervisor:	Jan Jelschen
First Reviewer:	Dr. Oliver Becker
Word Count:	< 12.000 (text + footnotes)
Degree Program:	B.Sc. Business Informatics (A Track 2018)
University:	Hamburg School of Business Administration
Submission Date:	09.04.2021
Partner Company:	apploft GmbH

HSBA HAMBURG SCHOOL OF
BUSINESS ADMINISTRATION

Contents

Chapter 1

Introduction

Mobile platforms are dominated by two players - Apple and Google with their respective operating systems iOS and Android. Cumulatively, they form a duopoly in the smartphone operating systems market with a combined usage share of 15.2% for iOS and 84.8% for Android in 2020 according to to the International Data Corporation (IDC 2021).

To develop a mobile software application (app) for both target platforms, the corresponding development environments and technologies are utilized for each platform. This leads to a doubling of cost, development time, and the need for knowledge of two different application development paradigms. As a result cross-platform frameworks such as Xamarin (Microsoft Corp. 2021), React Native (Facebook 2021), and Ionic (Ionic 2021) have been created.

The fundamental principle behind these frameworks is a specific tech stack operating on a single code base leading to increased development speed while also providing the ability to deploy for both mobile operating systems.

Generally, cross-platform frameworks utilize webviews or a software bridge to communicate with the underlying host platform plugging into native interfaces. In both cases, transmission delays may occur during app runtime leading to reduced execution speed. In addition, cross-platform frameworks deliver abstract interfaces over multiple native interfaces leading to a decreased subset of functionality and especially impaired UI customizability. These inherent architecture attributes (further see Section 2.1) explain both impeded performance (Ebone et al. 2018 and Corbalán 2019) and user experience (Mercado et al. 2016 and Angulo and Ferre 2014) compared to native technologies.

However, over the past 2 years one particular cross-platform technology with certain distinct attributes has risen strongly in popularity (Statista 2021). Flutter (Google Inc. 2020) is a newly developed open-source UI toolkit developed by Google. It has a nonconventional approach to cross-platform development in that every app ships with the frameworks rendering engine (Section ...). Thereby, Flutter bypasses host platform communication in terms of UI rendering and delivers a natively compiled binary which can be executed by the underlying device (see Section 2.2). Furthermore, the framework provides a *"[...] collection of visual, structural, platform, and interactive widgets"* (Google Inc. (2021b)) for UI customization. It seems that Flutter addresses the exact same issues that are generally criticized about cross-platform solutions.

1.1 Motivation

As a digital agency specialized on native iOS and Android development, apploft GmbH (the partnering company of this thesis) is highly interested in Flutter. The implications of using this framework could be wide ranging including the extension of the services portfolio to clients with lower budgets. For example, startups which are unsure about product/market fit (Andreesen 2007) and held by tight budget constraints are especially keen on reaching the maximum number of potential customers with their app. Flutter could be utilized for a fast iteration of a product deployable to both mobile platforms. Worth noting are the benefits of serving smaller customers like startups which include higher growth potential for long-term cooperation and risk diversification in apploft's client portfolio.

The potential of Flutter exceeds the acquisition of small clients. Instead of focusing on platform customization with native tooling, more effort could be directed into developing unique custom features. Furthermore, infrastructure setup, package development and app updates would only be necessary for one codebase.

Since the aforemntioned economic incentives aren't necessarily company-specific, they are relevant for mobile application developers at large. Scientifically, this thesis may be the basis for future work on Flutters architecture attributes and their contribution to runtime performance and UI rendering of frontend toolkits generally.

1.2 Problem Statement

Unfortunately, since Flutter was first released in March 2018 (Google Inc. 2021a), there are no peer reviewed articles comparing the performance or usability to native apps[1]. This leaves

[1] A search for relevant articles has been conducted using Google Scholar, Sci-hub and IEEE Xplore.

an especially interesting gap in the literature, since both aspects are the topmost perceived challenges of cross-platform frameworks considered from an industry-perspective (Biørn-Hansen 2019).

1.3 Thesis Objective

The aim of this thesis is derived based on the initially stated problems with cross-platform frameworks and the current lack of research on Flutters lofty marketing claims to solve those drawbacks. Specifically, Google's assertion that Flutter can match "native performance" and the framework can be utilized for building "expressive and flexible UI" (Google Inc. 2020) will both serve as inductively derived hypotheses that shall be empirically verified or falsified individually by this thesis:

H_P: The Flutter framework yields comparable performance to native mobile frameworks for app development.

H_U: The Flutter framework can be utilized to produce comparable user experiences as native UI toolkits for mobile devices.

1.4 Methods

An architypal native mobile app has been chosen as a case study for the evaluation of both hypotheses H_P and H_U. Based on typical mobile application features (as explained in Section 3.1), *Kickdown* (Kickdown GmbH 2021) - an online car auction app - was chosen for this thesis. The app has already been developed for iOS by apploft and released to the App Store in February of 2021 (link to appstore). For the purposes of comparison, a Flutter equivalent has been developed mimicking the relevant subset of UI and functionality of the original app (see Section 3.1). Both the original and Flutter replica app are used for subsequent hypotheses testing.

1.4.1 Performance comparison

The assesment of the performance hypothesis H_P has been conducted by profiling specific performance metrics including CPU, GPU and memory usage for particular use case flows in the original and Flutter application. The profiling results have been analyzed using common statistical techniques. A further explaination of the measurement process is detailled in Section

3.2.

1.4.2 Usability comparison

The analysis of the usability hypothesis H_U has been evaluated by the means of semi-structured expert interviews. Specifically, study participants have been be asked to evaluate both the Kickdown iOS and Flutter application clone along various metrics.

A detailed explaination of the interview process is given in section 3.3

1.5 Scope & Limitations

The feature set of the implemented app is representative for most, but not every type of app (see Section 3.1). Therefore the results cannot be generalized to every type of possible app. However, they should be seen as indicators of Flutters value as a cross-platform framework for the archetypal mobile app. Furthermore, the deductively chosen methodology yields the potential of finding adjacent hypothesis which may be further explored by other researchers.

Additionally, the usability study doesn't provide statistical significance due to its qualitative nature. Nevertheless, the depth of detail in expert interviews is much greater compared quantitative methods, and unthought of considerations may be suggested by the interviewees.

This research attribute is especially compelling given the current research state on the Flutter framework as mentioned above.

1.6 Thesis Structure

- TODO: Describe structure of thesis and summarize each chapter. Do this once the other chapters are actually written.

Bibliography

Andreesen, Marc. 2007. "Product/Market Fit." `https://web.stanford.edu/class/ee204/ProductMarketFit.html`.

Angulo, Esteban, and Xavier Ferre. 2014. "A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX." *Proceedings of the XV International Conference on Human Computer Interaction.* `https://dl.acm.org/doi/10.1145/2662253.2662280`.

Biørn-Hansen, Andreas. 2019. "An Empirical Study of Cross-Platform Mobile Development in Industry." *Wireless Communications and Mobile Computing.* `https://www.hindawi.com/journals/wcmc/2019/5743892/#conclusion`.

Corbalán, L. 2019. "A study of non-functional requirements in apps for mobile devices. In: Cloud computing and big data." *Springer International Publishing.* `https://www.researchgate.net/publication/334718481_A_Study_of_Non-functional_Requirements_in_Apps_for_Mobile_Devices`.

Ebone, A., Y. Tan, and X. Jia. 2018. "A Performance Evaluation of Cross-Platform Mobile Application Development Approaches." *IEEE.*

Facebook. 2021. *React Native.* `https://reactnative.dev/`.

Google Inc. 2020. *Flutter Homepage.* `https://flutter.dev/`.

———. 2021a. *Flutter SDK Releases.* `https://flutter.dev/docs/development/tools/sdk/releases`.

———. 2021b. *Widget Catalog.* `https://flutter.dev/docs/development/ui/widgets`.

IDC. 2021. *Smartphone Market Share,* December. https://www.idc.com/promo/smartphone-market-share/os.

Ionic. 2021. *Ionic Framework.* https://ionicframework.com/.

Kickdown GmbH. 2021. *Kickdown.* https://www.kickdown.com/.

Mercado, I. T., N. Munaiah, and A. Meneely. 2016. "The impact of cross-platform development approaches for mobile applications from the user's perspective." *Association for Computing Machinery Journal.* https://dl.acm.org/doi/abs/10.1145/2993259.2993268.

Microsoft Corp. 2021. *Xamarin.* https://dotnet.microsoft.com/apps/xamarin.

Statista. 2021. "Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020." https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/.