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

apploft.

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 IDC (2021).
To develop a mobile application 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. This has resulted in the creation of cross-platform (CP) frameworks such as Xamarin (Microsoft Corp. 2021), React Native (Facebook 2021) and Ionic (Ionic 2021).

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

However, CP frameworks lack in terms of performance and usability when compared to native technologies as shown by Mercado et al. (2016) and Ebone et al. (2018). This insufficiency in both aspects can be explained by their inherent architectures (see section 2.1.1). Generally, these frameworks utilize a software bridge to communicate with the underlying host platform leading to a transmission delay and thus impaired app performance. In addition, CP frameworks provide abstract interfaces over multiple native interfaces leading to a decreased subset of functionality and especially UI customizability.

Flutter (Google Inc. 2020) is a newly developed open-source UI toolkit developed by Google. It has a nonconventional approach to CP development in that every app ships with the Flutter rendering engine. Thereby, Flutter bypasses host platform communication in terms of UI rendering and deliveres a natively compiled binary which can be executed by the underlying device (see section 2.1.2). Furthermore, the framework delivers a *"[...] collection of visual, structural, platform, and interactive widgets"* (Google Inc. (2021b)) for UI customization. It seems that Flutter addresses the very issues that are generally criticized about CP solutions.

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]

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. The services portfolio of apploft could be extended to clients with lower budgets while not being tied to a specific operating system. For example, startups which are unsure about product market-fit 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 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.

Furthermore, infrastructure setup, package development and app updates would only need to be done for one codebase.

1.2 Thesis Goal

Based on the above stated problems with CP frameworks and the potential business opportunities, the goal of this thesis is to evaluate whether Flutter's claims on performance and usability hold up in practice.

1.3 Methods

To properly compare Flutter and native, an application will be developed which has typical mobile app features including the interaction with a remote API, different means of navigation between screens, user authentication and authorization. The feature selection process is further explained in section 3.1.

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

Based on these characteristics, *Kickdown* (Kickdown GmbH 2021) - an online car auction app - was chosen. The app has already been developed for iOS by apploft. To verify laid out claims of the Flutter framework, an app with specific matching functionality and UX design is reproduced.8

1.3.1 Performance comparison

To evaluate performance the typical measures of CPU, GPU and memory usage are chosen in this thesis. On the one hand, these metrics are the underlying causes of more ephemeral metrics such as page load speed. On the other hand, they can be easily measured using software tools (see section 3.2).

1.3.2 Usability comparison

Expert interviews with employees of apploft are conducted to compare the user experience of the developed Flutter app with the iOS application.

1.4 Scope & Limitations

The feature set of the implemented app is representative for most, but not every type of app. Therefore the results cannot be generalized to every type of app.
The usability study doesn't have a statistical significance due to its qualitative nature.
TODO: elaborate on these points.

1.5 Plan of Attack

The following is a list of subgoals of this thesis including accompanying deadlines.

1.5.1 Section Writing

- ~~write draft of *Introduction* section and discuss - 05.02.21~~
- write draft of *Flutter* section and discuss - 19.02.21
- write draft of *Study Design* section and discuss - 05.03.21
- write draft of *Application Design* section and discuss - 19.03.21

- write draft of *Performance Comparison* section and discuss - 02.04.21
- write draft of *UX Comparison* section and discuss - 02.04.21
- write draft of *Summary* section and discuss - 02.04.21
- write draft of *Abstract* section and discuss - 02.04.21

1.5.2 Performance Measurement

- prepare tooling and conceptualize measurement process - 26.02.
- execute a first iteration of measurement process - 05.03.
- execute a final iteration of measurement process - 26.03.

1.5.3 UX Measurement

- prepare interview process - 26.02.
- conduct a minimum of 3 interviews - 31.03.
- evaluate interviews - 01.04.

1.5.4 Submission

- topic submission - 15.02.
- send out thesis for proof reading - 05.04.21
- submit to examination office + register for colloquium - 09.04.

*The minimum requirement is to complete building out the *offerings* screen. This is the most complex screen of the app and constitutes the main feature. It is sufficient for performance comparison as well as a usability study. However, if time permits, more of the app will be developed and comparatively evaluated.

1.6 Following Chapter Summary

- TODO: Describe structure of thesis and summarize each chapter. This should be somehow interwoven into introduction.

Bibliography

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`.