

# Vergleichbares App Performance Testing unter Android und iOS mit Flutter-Modulen

## Table of Contents

1. Kontext / Problemstellung .....	1
1.1. Verschiedene App-Architekturen .....	1
1.2. Softwarequalität für mobile Applikationen .....	2
1.3. Ziel .....	4
2. Grundlagen .....	4
2.1. Performance Profiling .....	4
2.2. Instrumented Tests, UI Tests .....	5
2.3. Einbindung von Flutter-Modulen .....	5
3. Testumgebung .....	5
3.1. Implementierung .....	5
3.2. Ergebnisse .....	5
3.3. Ausblick .....	5

## 1. Kontext / Problemstellung

Android iOS

### 1.1. Verschiedene App-Architekturen

traditionell native android / ios apps doppelter Aufwand für nahezu gleiches ergebnis verschiedene cross platform optionen (paper, welches generell mal alles angeschaut hat)

normalerweise schon app vorhanden

Eine komplette Neuentwicklung einer bestehenden App ist aber nicht nur Zeit- und kostenaufwändig, zudem müssen die existierenden Apps parallel weiterhin unterstützt werden, um die bestehenden Benutzer nicht vom Produkt zu vertreiben. Weiterhin ist es ein großes Risiko, die Funktionalität einer bestehenden App in einer neuen Architektur zu kopieren und im schlimmsten Fall ein schlechteres Ergebnis als die ursprüngliche App zu bekommen. Eine schrittweise Aktualisierung einzelner Features erscheint da sinnvoller.

Mit der Version 1.12.13 des Flutter SDKs in 2019 wurde unter anderem das sogenannte Add-to-App Feature vorgestellt <sup>[1]</sup>. Mit diesem ist es möglich, ein Flutter-Modul in eine bestehende Android- oder iOS App zu integrieren. Somit ist eine schrittweise Aktualisierung der Apps von der Ersetzung einzelner Features hin zu einer reinen Flutter-App prinzipiell technisch möglich. Es stellt sich

jedoch die Frage, für welche Apps sich ein Umstieg zu Flutter lohnt und wo die Schnittstellen zwischen Flutter- und nativem Quellcode am sinnvollsten sind.

## 1.2. Softwarequalität für mobile Applikationen

wie kann man die Qualität messen? Um zu entscheiden, ob sich generell der Einsatz eines anderen Frameworks wie bspw. Flutter im Vergleich zur nativen Implementierung lohnt, stellt sich zunächst das Hindernis, wie die App-Qualität untersucht werden kann. Dazu werden die offiziellen Richtlinien von Android und iOS in den Kontext von zwei weiteren Qualitätsmodellen für mobile Applikationen gesetzt.

Table 1. Teilgebiete der Softwarequalität für mobile Applikationen

	<b>Android Guidelines</b> <sup>[2]</sup>	<b>iOS Best Practices</b> <sup>[3]</sup>	<b>Key Attributes for a Quality Mobile Application</b> <sup>[4]</sup>	<b>An Efficient and Effective New Generation Objective Quality Model for Mobile Applications</b> <sup>[5]</sup>
(1) App benutzt Systemressourcen in sinnvollem Maße	Performance & Stability Functionality	Optimize Performance	Productivity	Efficiency (battery power, memory)
(2) Benutzung nativer UIs, Bedienkonzepte, Services	Functionality	Assemble your UI from standard views Human Interface Guidelines	Convenience Ease of Use	Adaptability Usability
(3) Daten werden vor unauthorisiertem Zugriff geschützt	Security	Secure Data	Security	Security
(4) Für eine Aufgabe werden die sinnvollsten und simpelsten Methoden verwendet	Functionality	Human Interface Guidelines	Ease of Use Productivity	Suitability Usability
(5) App lässt sich über App Stores aktualisieren, hält dessen Standards ein	Google Play	App Store Guidelines	Adaptability Convenience	Extensibility Maintainability

	<b>Android Guidelines</b> <sup>[2]</sup>	<b>iOS Best Practices</b> <sup>[3]</sup>	<b>Key Attributes for a Quality Mobile Application</b> <sup>[4]</sup>	<b>An Efficient and Effective New Generation Objective Quality Model for Mobile Applications</b> <sup>[5]</sup>
(6) Daten überleben Pause, Neustart, Rotation der App	Visual Experience	-	Information Perisistence	Data Integrity
(7) Unterstützt mehrere Gerätemodelle, auch in Zukunft	-	Build for multiple platforms Automatic Layout	Flexibility	Portability Maintainability
(8) Anpassung an verschiedene Regionen, Kulturen, Einschränkungen	Visual Experience	Accessibility Inclusion Internationalize	-	-
(9) Privatsphäre der Nutzerdaten	Privacy	Protect Privacy	-	-
(10) Debugging und Testen für robusten Code	(Checkliste abarbeiten)	Test	-	-
(11) App erfüllt die Anforderungen der User	-	-	-	Functionality
(12) Nutzen von verschiedenen Geschäftsmodelle n	-	Choose Business model	-	-

Es konnten 12 verschiedene Teilgebiete der Qualität ermittelt werden, von denen die ersten 5 in allen vier Quellen eine Relevanz haben.

(Übergang nur zu Performance)

Es existieren bereits einige Vergleiche zwischen der Performance von nativen Apps und Cross-Plattform-Apps mit verschiedenen Technologien.

Es gibt bereits einige paper die flutter einbinden, hier ergebnisse

*A Comparison of Performance and Looks Between Flutter and Native Applications* <sup>[6]</sup> vergleichen Android, iOS und Flutter in Bezug auf die App Performance, Die Komplexität des Quellcodes und das Look&Feel der resultierenden Apps. Dazu wurde ein App-Konzept jeweils nativ in Kotlin (Android), Swift (iOS) und Flutter implementiert und die resultierenden Apps auf den jeweiligen

Plattformen miteinander verglichen. Die CPU-Auslastung wurde manuell auf beiden Plattformen gemessen und dann die jeweilige Flutter-App mit der nativen App verglichen. Es konnte kein nennenswerter Unterschied in der Performance festgestellt werden. Die Qualität des UIs in den beiden Android-Apps wurde aus einer Umfrage ermittelt, die iOS Apps wurden nicht verglichen. Die Automatisierung von Performance- und UI-Tests wurde für die potentielle Verbesserung der Messergebnisse vorgeschlagen.

<sup>[7]</sup> <sup>[8]</sup> das aber nicht mit modulen zusammen, wie sieht es da aus?

performance tests nur manuell, nicht automatisch

automatisierung auch in xref1 thema

Im folgenden wird sich um verschiedene Möglichkeiten der Integration von Flutter Modulen zu untersuchen sollen die Performance-footprints zwischen nativ und flutter verglichen werden

## 1.3. Ziel

Eine wichtige Entscheidung, ob sich die Integration eines Flutter-Moduls lohnt, ist die resultierende Performance der App. Da Mobilgeräte sowohl durch die Akkulaufzeit und die Temperatur (für angenehme Bedienung in der Hand) strenge Grenzen gesetzt sind

integration tests für aufrufe native → flutter

ui tests für vergleich von native ui mit flutter ui

### RQ1

Welche Tools und Methoden existieren zum Performance Profiling auf den Plattformen Android und iOS und für das Flutter-Framework?

### RQ2

Welche Performance-Metriken können auf beiden Plattformen und im Kontext von Flutter-Modulen erfasst werden?

### RQ3

Wie können die Daten aufbereitet und in ein einheitliches Format zur Weiterverarbeitung überführt werden?

die vorliegende Arbeit soll den Grundstein legen für Performance Profiling von Software

## 2. Grundlagen

### 2.1. Performance Profiling

#### 2.1.1. Profiler und deren Metriken

#### 2.1.2. macro / microbenchmarks

## 2.2. Instrumented Tests, UI Tests

## 2.3. Einbindung von Flutter-Modulen

# 3. Testumgebung

Die Nutzbarkeit der gefundenen Methoden zum Performance Profiling zwei apps erstellt flutter modul eingebunden

## 3.1. Implementierung

### 3.1.1. Flutter Modul

kommunikaton zwischen Flutter und native

### 3.1.2. Android App

### 3.1.3. iOS App

learnings bei ios

## 3.2. Ergebnisse

## 3.3. Ausblick

nicht nur Performance wichtig, sondern auch andere wichtige Metriken

wie kann softwarequalität sinnvoll und objetiv gemessen werden? ISO 9126

**Änderbarkeit / Wartbarkeit**

**Benutzbarkeit**

**Änderbarkeit**

Testbarkeit

**Effizienz**

Verbrauchsverhalten

**Übertragbarkeit**

**Zuverlässigkeit**

**Funktionalität**

welche Tools/Methoden existieren zur Performance-Messung auf den Plattformen Android und iOS wenn mehrere Methoden: welche Vor- und Nachteile haben diese?

Wie können die Daten aufbereitet und in ein einheitliches Format zur Weiterverarbeitung überführt werden? Gibt es nutzbare Standards?

### *Feeding the Werewolves*

While werewolves are hardy community members, keep in mind the following dietary concerns:

#### **IMPORTANT**

1. They are allergic to cinnamon.
2. More than two glasses of orange juice in 24 hours makes them howl in harmony with alarms and sirens.
3. Celery makes them sad.

Sidebars are used to visually separate auxiliary bits of content that supplement the main text.

sdf

[1] Flutter 1.12.13 release notes, Add-to-App, <https://docs.flutter.dev/development/tools/sdk/release-notes/release-notes-1.12.13#add-to-app-feature>, Abgerufen 2023-03-06

[2] Android Core App quality checklist, <https://developer.android.com/docs/quality-guidelines/core-app-quality>, Abgerufen 2023-03-06

[3] Planning your iOS App, best practices, <https://developer.apple.com/ios/planning/#adopt-best-practices>, Abgerufen 2023-03-06

<a id="\_footnotedef\_4"></a>[<a anchor="\_footnoteref\_4">4</a>] Key Attributes for a Quality Mobile Application, <a href="https://ieeexplore.ieee.org/document/9058278/" class="bare">https://ieeexplore.ieee.org/document/9058278/"</a>, Abgerufen 2023-03-05

<a id="\_footnotedef\_5"></a>[<a anchor="\_footnoteref\_5">5</a>] An Efficient and Effective New Generation Objective Quality Model for Mobile Applications, <a href="https://www.researchgate.net/profile/Ali-Javed/publication/274048560\_An\_Efficient\_and\_Effective\_New\_Generation\_Objective\_Quality\_Model\_for\_Mobile\_Applications/links/56a8dc1d08aeea2a20497e7e/An-Efficient-and-Effective-New-Generation-Objective-Quality-Model-for-Mobile-Applications.pdf" class="bare">https://www.researchgate.net/profile/Ali-Javed/publication/274048560\_An\_Efficient\_and\_Effective\_New\_Generation\_Objective\_Quality\_Model\_for\_Mobile\_Applications/links/56a8dc1d08aeea2a20497e7e/An-Efficient-and-Effective-New-Generation-Objective-Quality-Model-for-Mobile-Applications.pdf"</a>, Abgerufen 2023-03-05

[6] A Comparison of Performance and Looks Between Flutter and Native Applications, <https://www.diva-portal.org/smash/get/diva2:1442804/FULLTEXT01.pdf>, Abgerufen 2023-02-12

<a id="\_footnotedef\_7"></a>[<a anchor="\_footnoteref\_7">7</a>] Animations in Cross-Platform Mobile Applications: An Evaluation of Tools, Metrics and Performance, <a href="https://www.mdpi.com/1424-8220/19/9/2081" class="bare">https://www.mdpi.com/1424-8220/19/9/2081"</a>, Abgerufen 2023-02-25

[8] Flutter Versus Other Mobile Development Frameworks: A UI And Performance Experiment. Part 2, <https://web.archive.org/web/20221005043739/https://blog.codemagic.io/flutter-vs-android-ios-xamarin-reactnative/>, Abgerufen 2023-02-12