

Kapitel 1

Über dieses Buch

»Geduld und Mut schaffen sich ihr eigenes Glück.«
– Publilius Syrus, Römischer Mimen-Dichter

Wenn Sie die plattformübergreifende Entwicklung in den letzten Jahren verfolgt haben, sind Ihnen bestimmt viele Technologien aufgefallen, die mit dem Versprechen angetreten sind, das Zeitalter von Plattform-Teams zu beenden und eine Ära anzubrechen, in der *ein* Team zum Entwickeln, Testen und Managen ausreicht, um eine App auf alle vorstellbaren Geräte zu bringen.

Wenn Sie sich in den Medien oder möglicherweise sogar in Ihrem Unternehmen umsehen, zeigt sich, dass die neuen Sprachen und Werkzeuge zwar eingesetzt werden und zahlreiche Apps mithilfe der plattformübergreifenden Entwicklung programmiert wurden. Doch sind die Platzhirsche wie Swift/UIKit und Kotlin/Android-API dadurch nicht etwa zurückgedrängt worden, sondern weiden, ganz überwiegend unbeindruckt, auf den saftigsten Wiesen weiter.

Die Herausforderer schafften es nicht, genügend überzeugte Entwicklerinnen und Entwickler hinter sich zu scharen. Manch ein Werkzeug stellte sich als zu umständlich heraus. Andere boten einfach nicht genug Mehrwert oder machten es wegen der mangelnden Verbreitung schwer, (bezahlbares) Personal zu finden, das die jeweils geforderte Programmiersprache beherrscht.

Und was ist mit Flutter? Flutter unterscheidet sich unter anderem von seinen Konkurrenten darin, dass es nicht versucht, die Plattform-Views wie die von Android und iOS für die Programmiersprache Dart *verfügbar* zu machen, sondern gleich die gesamte Oberfläche selbst zeichnet. Flutter-Views sind somit keine UIViews oder Android-Views. Das schafft eine enorme Flexibilität, mit der Sie sogar den kompletten Designstil auswechseln können – noch während Ihre App läuft! Diese Eigenschaft bieten Ihnen zwar auch andere Technologien, doch fußen diese dann auf HTML, welches in einer WebView auf dem Gerät angezeigt wird. Flutter läuft dahingegen auf einem hochperformanten und gänzlich nativen Framework. Ihre Bits müssen also nicht drei Mal umsteigen, bevor sie ans Ziel gelangen, sondern reisen direkt von A nach B!

Weil Flutter in Dart geschrieben wurde, können Sie die Änderungen an Ihrer App jederzeit während der Entwicklung per *Hot-Reload* ausprobieren (mehr dazu in Kapitel 3, »Warum man mit Dart voll ins Schwarze trifft«). Das Entwickeln mit Flutter

ist schnell (ein Entwurfsziel des Frameworks) und unkompliziert. Im Gegensatz zu anderen Technologien entwickelte sich Flutter blitzartig zu einer der beliebtesten und angesagtesten Technologien.

Seit dem Jahr 2012 arbeite ich in Projekten, die mehrere Plattformen und Geräte ansprechen, und kann auf Erfahrungen mit Windows UWP und Xamarin zurückblicken. Doch dieses Gefühl, dass ein plattformübergreifendes Framework richtig durchstarten wird, stellte sich erst ein, als ich 2019 begann, erste Projekte mit Flutter zu realisieren. Besonders deutlich wurde mir dies, als mir ein Engineer der Qualitätssicherung eines Unternehmens von seinem Portfolio aus Flutter-Apps berichtete, die er ohne vorherige Erfahrung in der App-Entwicklung geschrieben und veröffentlicht hatte. Flutter schien also eine Technologie zu sein, die jeden dazu befähigt, den Traum von einer eigenen App anzugehen!

Interessanterweise verbreitet sich diese Technologie genauso in Start-ups wie auch in »Grown-ups« und schreckt dabei selbst vor komplexeren Anwendungsbereichen nicht zurück. Viele solche Apps habe ich vom ersten Grundgerüst bis zur Veröffentlichung im App Store begleiten dürfen. Nicht zuletzt während dieser nicht unerheblichen Anzahl von Stunden habe ich Flutter zu schätzen gelernt und mich von seiner Flexibilität und Wandlungsfähigkeit, aber auch Stabilität gänzlich überzeugen lassen.

Ich freue mich daher ganz besonders, Ihnen in diesem Buch die Sprache Dart und das Framework Flutter vorstellen zu können und hoffe, dass Sie genauso viel Spaß beim Schreiben Ihrer Apps haben werden!

1.1 Was Sie in diesem Buch lernen werden

Mit diesem Buch möchte ich Ihnen einen umfassenden Überblick zu Flutter geben. Da das Framework nur in Verbindung mit der Programmiersprache Dart benutzt werden kann (und darin geschrieben wurde), stelle ich Ihnen diese Sprache zu Beginn vor. Dabei führe ich Sie vom Einsatz von Variablen über die Benutzung von Klassen bis hin zu mächtigen Werkzeugen wie `async/await`, `Mixins` und `Null-Safety`. Sie benötigen hierzu keinerlei Vorkenntnisse in anderen Programmiersprachen.

Falls Sie bereits mit der Programmiersprache vertraut sind, können Sie auch mit Kapitel 4, »Pubs: Abhängigkeiten komfortabel verwalten«, starten und in Kapitel 5, »Widgets«, in die Welt von Flutter eintauchen. Dort angekommen, erfahren Sie, was Widgets eigentlich sind, wie Sie Oberflächen daraus zusammensetzen und wie Sie eine responsive und dynamische App aufbauen.

Anschließend führt Sie das Kapitel 6, »Layouting«, durch die vielseitigen Möglichkeiten, um Widgets zu positionieren und zu scrollen. Nachdem Sie diese Grundlagen beherrschen, behandeln wir die Navigation zwischen Pages, wie und wo Sie Ihre Geschäftslogik schreiben und wie Sie gängige Elemente wie Buttons, Textfelder usw.

einsetzen. Wie einfach Sie eigene Inhalte zeichnen und animieren können, erlernen Sie im mittleren Teil, der sich ebenfalls damit befasst, wie Sie Ihre App lokalisieren und für Barrierefreiheit sorgen.

Von der Oberfläche geht es weiter zu den anderen Komponenten einer modernen App wie HTTP-Requests an APIs, Datenbanken und Sensoren. Zudem heben wir den Schleier, hinter dem Flutter über sogenannte *Method Channels* mit der Plattform kommuniziert und es Ihnen ermöglicht, auf die APIs etwa von iOS oder Android zuzugreifen.

Im letzten Teil befassen wir uns mit Qualitätsmaßnahmen wie dem Logging, der Integration von Analytics, dem Debuggen und Profilen Ihrer App und dem Schreiben von automatischen Tests. Abschließend möchte ich Ihnen außerdem einige Informationen zum Thema Release mit auf den Weg geben, sodass Sie wissen, wie Sie Ihre Flutter-App fit für die App Stores bekommen oder Ihr eigenes Plugin für die Flutter-Entwicklung anderen zur Verfügung stellen können.

1.2 Null-Safety als Default-Einstellung ab Flutter 2.2

Technologien, die sich schnell entwickeln, sind eine besondere Herausforderung für gedruckte Bücher und ihre Autoren. Ähnlich wie in der Softwareentwicklung sollte es für dieses Buch ein »Code Freeze« geben, bevor der Text »eingefroren« wurde. Dabei konnte es aber nicht bleiben:

Mit Dart 2.12 wurde die *Null-Safety* ausgerollt, die ich in Abschnitt 3.16 beschreibe. Erst mit Flutter 2.2 aber, das nach dem »Freeze« veröffentlicht wurde, wird Null-Safety beim Erstellen eines Projekts eingeschaltet. Die Codebeispiele in diesem Buch setzen, so wie sie abgedruckt sind, voraus, dass Null-Safety nicht aktiviert ist. Deshalb stelle ich Ihnen bei den Materialien zum Buch zusätzlich den Code so zur Verfügung, wie er zu Flutter 2.2 bei aktivierter Null-Safety passt.

Wenn Sie mit Flutter 2.2 oder höher arbeiten, haben Sie also zwei Möglichkeiten:

► Sie deaktivieren die Null-Safety für Ihr Projekt.

Ändern Sie dazu in Ihrer Konfigurationsdatei `pubspec.yaml` (siehe Kapitel 4) die Angabe zu `sdk`:

```
environment:
  sdk: ">=2.11.0 <3.0.0"
```

Dann funktionieren alle Codebeispiele so, wie sie im Buch abgedruckt sind. Wählen Sie im Downloadmaterial bei den Materialien zum Buch CODEBEISPIELE (ORIGINALVERSION).

► **Sie verwenden die aktualisierten Codebeispiele aus dem Downloadmaterial.**

Um mit aktivierter Null-Safety zu arbeiten, sind Änderungen im Quelltext nötig, die aber nicht umfangreich sind. Sie werden sie in Kapitel 3 ab Abschnitt 3.16 kennenlernen. Wählen Sie bei den Materialien zum Buch die Codebeispiele für Flutter 2.2 oder höher aus, wenn Sie von vornherein mit dieser Version und aktivierter Null-Safety arbeiten möchten.

1.3 Danksagung

Mein ganz besonderer Dank gebührt Frau Almut Poll für Ihre Hilfe und Unterstützung sowie dem Rheinwerk Verlag. Außerdem danke ich Uwe Post dafür, dass er seinen aufmerksamen Blick auf das Fachliche gerichtet hat und mir zahlreiche Hinweise und Empfehlungen lieferte. Viele Tipps und Ratschläge meines Kollegen Nino Wagen-sonner rundeten die Inhalte zusätzlich ab.

Letztlich hätte ich dieses Buch aber nicht ohne die Unterstützung meiner Frau Anne vollenden können und kann das entgegengebrachte Verständnis auch unserer Tochter Thalea kaum genug würdigen!

Marc Marburger

Berlin, Juni 2021