

.NET BIBLIOTHEK

hrsg. von Dr. Holger SCHWICHTENBERG

André KRÄMER



Cross-Plattform-Apps mit **Xamarin.Forms** entwickeln



MIT C# FÜR
ANDROID UND iOS
PROGRAMMIEREN

HANSER

Krämer

Cross-Plattform-Apps mit Xamarin.Forms entwickeln



Bleiben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

www.hanser-fachbuch.de/newsletter



André Krämer

Cross-Plattform-Apps mit Xamarin.Forms entwickeln

Mit C# für Android und iOS
programmieren

HANSER

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2021 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Walter Saumweber, Ratingen

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Max Kostopoulos

Titelmotiv: © Stephan Rönigk, unter Verwendung von Grafiken von © fotolia.com/Dora Vision

Satz: Eberl & Koesel Studio GmbH, Altusried-Krugzell

Druck und Bindung: CPI books GmbH, Leck

Printed in Germany

Print-ISBN: 978-3-446-45155-1

E-Book-ISBN: 978-3-446-45366-1

E-Pub-ISBN: 978-3-446-47204-4

Inhalt

Vorwort	XIII
Der Autor	XVII
Hinweise zum Buch	XIX
1 Einführung	1
1.1 Xamarin und die Xamarin-Plattform	2
1.2 Überblick über die Xamarin-Plattform	3
1.3 Architektur der Xamarin-Plattform	6
1.3.1 Mono und das .NET Framework	6
1.3.2 Mono als Grundlage der Xamarin-Plattform	8
1.3.2.1 Xamarin.Android	8
1.3.2.2 Xamarin.iOS	9
1.4 Entwicklungswerkzeuge	10
1.5 Was Sie in diesem Kapitel gelernt haben	13
2 Schnelleinstieg: Ihre erste App mit Xamarin.Forms in einer Stunde	15
2.1 Das Konzept der App	16
2.1.1 Die besondere Stärke der App	16
2.1.2 Auswahl der App-Funktionen	17
2.1.3 Skizze der Benutzeroberfläche	18
2.2 Anlegen des Projekts	21
2.3 Umsetzen der Oberfläche	22
2.4 Implementieren des Quellcodes	30
2.5 Die fertige App	39
2.6 Was Sie in diesem Kapitel gelernt haben	40

3	Überblick über Android und Xamarin.Android	41
3.1	Das Betriebssystem Android	41
3.2	Architektur von Xamarin.Android	43
3.3	Software Development Kits	44
3.3.1	Android-API-Level	46
3.4	Eine einfache Android-App mit Visual Studio anlegen	47
3.4.1	Ordnerstruktur einer einfachen App	48
3.4.2	Standarddateien einer einfachen App	52
3.5	Apps testen und debuggen	57
3.5.1	Test im Emulator	57
3.5.1.1	Android-Emulatoren	58
3.5.2	Debugging auf einem echten Gerät	63
3.6	Was Sie in diesem Kapitel gelernt haben	65
4	Überblick über iOS und Xamarin.iOS	67
4.1	Das Betriebssystem iOS	67
4.1.1	Verbreitung und Versionen	67
4.2	Architektur von Xamarin.iOS	68
4.3	Software Development Kits und Systemvoraussetzungen	70
4.3.1	Xcode und das iOS SDK installieren	70
4.3.2	Mono, Xamarin.iOS und Visual Studio für Mac installieren	71
4.3.3	Integration in Visual Studio für Windows installieren	72
4.4	Eine Verbindung zum Mac herstellen	73
4.5	Eine einfache iOS-App mit Visual Studio anlegen	77
4.5.1	Ordner- und Dateistruktur einer einfachen App	78
4.6	Apps testen und debuggen	85
4.6.1	Tests im Simulator	85
4.6.2	Einschränkungen des Simulators	87
4.6.3	Debugging auf einem echten Gerät	87
4.7	Was Sie in diesem Kapitel gelernt haben	92
5	Überblick über Xamarin.Forms	93
5.1	Der Xamarin.Forms-Ansatz	93
5.2	Funktionsumfang	99

5.2.1	Views, Layouts, Pages und Cells	100
5.2.2	GestureRecognizer	100
5.2.3	Navigation	101
5.2.4	MessagingCenter	101
5.2.5	DependencyService	102
5.2.6	App Lifecycle	102
5.3	Architektur von Xamarin.Forms	103
5.4	Eine einfache Xamarin.Forms-App in Visual Studio anlegen	105
5.4.1	Projekt-, Ordner- und Dateistruktur einer Xamarin.Forms-App	106
5.4.1.1	Das Android-Projekt	108
5.4.1.2	Das iOS-Projekt	109
5.4.1.3	Das plattformübergreifend geteilte Projekt	110
5.5	Grenzen von Xamarin.Forms	113
5.6	Was Sie in diesem Kapitel gelernt haben	114
6	Architektur von Cross-Plattform-Apps	115
6.1	Plattformübergreifende Wiederverwendung	115
6.2	Freigegebene Projekte und Präprozessoranweisungen	116
6.2.1	Ein freigegebenes Projekt in Visual Studio anlegen	118
6.2.2	Vorteile von freigegebenen Projekten	121
6.2.3	Nachteile von freigegebenen Projekten	121
6.2.4	Fazit zu freigegebenen Projekten und Präprozessoranweisungen	124
6.3	.NET-Standard-Klassenbibliotheken für plattformübergreifenden Quellcode nutzen	124
6.3.1	Ein .NET-Standard-Projekt in Visual Studio anlegen	129
6.3.2	Vorteile von .NET-Standard-Projekten	129
6.3.3	Nachteile von .NET-Standard-Projekten	130
6.3.4	Fazit zu .NET-Standard-Projekten	130
6.4	Abhängigkeitsmanagement	130
6.4.1	Der Xamarin.Forms DependencyService	131
6.4.2	Dependency Injection mit dem Inversion of Control Container Microsoft.Extensions.DependencyInjection	134
6.5	Was Sie in diesem Kapitel gelernt haben	145

7	Oberflächendesign mit Layoutcontainern	147
7.1	Das Xamarin.Forms-StackLayout	147
7.2	Das Xamarin.Forms-Grid	154
7.3	Das Xamarin.Forms-AbsoluteLayout	159
7.4	Das Xamarin.Forms-FlexLayout	161
7.5	ScrollView	165
7.6	Was Sie in diesem Kapitel gelernt haben	167
8	Eingabeformulare anlegen	169
8.1	Grundlegende Steuerelemente	169
8.1.1	Übergreifende Konzepte zu Steuerelementen	169
8.1.1.1	Größe und Position von Views	170
8.1.1.2	Zugriff auf Views im Code-Behind	170
8.1.1.3	Plattformspezifische Eigenschaftswerte vergeben	171
8.1.2	Views zur Darstellung von Inhalten	171
8.1.2.1	Label	171
8.1.2.2	WebView	173
8.1.2.3	ActivityIndicator und Progressbar	175
8.1.3	Texteingaben mit Entry und Editor	176
8.1.4	Button	178
8.1.5	Steuerelemente zur Auswahl	179
8.1.5.1	RadioButton	179
8.1.5.2	CheckBox	181
8.1.5.3	Switch	182
8.1.5.4	Picker	183
8.1.5.5	Datum und Uhrzeit mit DatePicker und TimePicker auswählen	185
8.2	Das Model-View-ViewModel (MVVM)-Muster und Datenbindung	187
8.2.1	Das MVVM-Muster	187
8.2.2	Datenbindung	188
8.2.2.1	Einfache Datenbindung	189
8.2.2.2	Command-Binding	191
8.2.2.3	Performanceschub durch kompilierte Bindung	193
8.2.3	Fazit zur Datenbindung und dem MVVM-Muster	194

8.3	Steuerelemente in der Beispiel-App	194
8.3.1	Das Model anlegen	195
8.3.2	Das ViewModel	197
8.3.3	Die View	200
8.3.4	Aufruf der Profilseite und Registrierung der Abhängigkeiten ..	202
8.3.5	Ein Wort zur Strukturierung der App	203
8.4	Was Sie in diesem Kapitel gelernt haben	204
9	Navigation und die Xamarin.Forms Shell	205
9.1	Navigation in mobilen Apps	205
9.1.1	Registerkarten	205
9.1.2	Hierarchische Navigation	208
9.1.3	Seitenleiste	210
9.2	Die Xamarin.Forms Shell	214
9.2.1	Überblick über die Shell	214
9.2.2	Eine Navigationsstruktur mit der Shell definieren	216
9.2.2.1	Eine Registerkartennavigation mit der Shell anlegen ..	216
9.2.2.2	Eine seitliche Navigationsleiste mit der Shell	218
9.2.2.3	Die seitliche Navigation mit Registerkarten kombinieren	219
9.2.2.4	Menüeinträge als Aktionselemente nutzen	221
9.2.3	Kopf- und Fußzeile der Shell definieren	223
9.2.4	Routenbasierte Navigation	224
9.3	Dependency Injection und die Xamarin.Forms Shell	227
9.4	Navigation in der Beispiel-App	228
9.4.1	Die Shell anlegen	229
9.4.2	Änderungen an der Dependency Injection	230
9.4.3	Das Navigationsframework der Shell abstrahieren	231
9.5	Was Sie in diesem Kapitel gelernt haben	232
10	Listen	233
10.1	Listen unter Xamarin.Forms	233
10.2	Die Xamarin.Forms-CollectionView	234
10.2.1	Einträge selektieren	236

10.2.2	Kontextmenü	240
10.2.3	Daten aktualisieren	242
10.2.4	Gruppierte Einträge	244
10.3	Listen in der Beispiel-App	246
10.3.1	Daten in der Beispiel-App	246
10.3.2	Die Startseite der App	249
10.3.3	Die Speisekarte	253
10.3.4	Die Detailseite	259
10.4	Was Sie in diesem Kapitel gelernt haben	263
11	Bilder, Icons und Schriftarten	265
11.1	Bilder darstellen	265
11.1.1	Plattformspezifische Bilder	265
11.1.2	Eingebettete Bilder	270
11.2	Schriftarten	272
11.3	Icon-Schriftarten verwenden	275
11.4	Bilder und Schriftarten in der Beispiel-App	279
11.4.1	Startseite und Menü	281
11.4.2	Kategorie- und Detailseite	284
11.5	Was Sie in diesem Kapitel gelernt haben	286
12	Styles und Themes in Xamarin.Forms	289
12.1	Styles in Xamarin.Forms	289
12.1.1	Implizite und explizite Stildefinitionen	291
12.1.2	Mehrere Formatvorlagen mit Style-Classes anwenden	294
12.2	Styling mit Cascading Style Sheets (CSS)	297
12.3	Themes	299
12.3.1	Hell- und Dunkelmodus des Betriebssystems unterstützen	303
12.4	Styles und Dunkelmodus in der Beispiel-App	308
12.5	Was Sie in diesem Kapitel gelernt haben	311
13	Aufruf von Webservices	313
13.1	Webservices mit dem „HttpClient“ aufrufen	314
13.1.1	Daten mit GET lesen	315

13.1.2	Daten mit POST anlegen	318
13.1.3	Daten mit PUT ändern	319
13.1.4	Daten mit DELETE löschen	320
13.1.5	Umgang mit dem Offline-Fall	320
13.2	Webservices in der Beispiel-App	323
13.3	Was Sie in diesem Kapitel gelernt haben	326
14	Lokale Datenhaltung	327
14.1	Zugriff auf das lokale Dateisystem	327
14.1.1	Die Android-Verzeichnisstruktur	328
14.1.2	Die iOS-Verzeichnisstruktur	329
14.1.3	Dateizugriff über die Klassen File und Directory aus dem Namensraum System.IO	330
14.2	Lokale Datenhaltung mit einer SQLite-Datenbank	333
14.3	Was Sie in diesem Kapitel gelernt haben	338
15	Zugriff auf native Gerätefunktionen mit Xamarin.Essentials ..	339
15.1	Funktionsüberblick über Xamarin.Essentials	339
15.2	Xamarin.Essentials im eigenen Projekt einbinden	342
15.3	Xamarin.Essentials in der Beispiel-App	347
15.4	Was Sie in diesem Kapitel gelernt haben	352
16	Nachwort	353
	Register	355

Vorwort

Im Frühjahr 2013 hörte ich auf der Fahrt zu einem Kunden die Tablet Show, einen früheren Podcast von Carl Franklin und Richard Campbell zum Thema Mobile Development. Mein Hauptinteresse galt damals, wie bei so vielen .NET Entwicklern, der Entwicklung von Windows-8-Apps. Gemeinsam mit einem Bekannten hatte ich wenige Monate zuvor eine App für Windows 8 veröffentlicht, die zu dieser Zeit gerade die Schallmauer von 1 000 000 Downloads durchbrach.

In früheren Episoden der Tablet Show hatte ich bereits einige nützliche Tipps erhalten, die uns bei der Optimierung unserer App halfen, und so war ich sehr neugierig, was diese Folge Neues bringen würde.

In dieser Folge berichtete Miguel de Icaza, einer der beiden Gründer der Firma Xamarin, dass es eine kostenfreie, eingeschränkte Einsteigerversion von Xamarin geben wird sowie eine günstige Variante für einzelne Entwickler. Außerdem kündigte er an, dass die Entwicklung von Xamarin-Apps von nun an auch in Visual Studio möglich war.

Ab diesem Zeitpunkt war ich von Xamarin fasziniert. Die Tatsache, dass ich mit meinen C#, .NET- und Visual-Studio-Kenntnissen dank Xamarin in der Lage war, Apps für Android und iOS zu schreiben, ohne Java oder Objective-C lernen zu müssen, begeisterte mich, und diese Begeisterung hält bis heute an.

Kurz nachdem ich den Podcast gehört hatte, installierte ich die Xamarin-Werkzeuge auf meinem Rechner und legte mit den ersten Experimenten los. Da ich keinen Mac hatte, konnte ich am Anfang nur für Android programmieren. Da mir das nicht genug war, kaufte ich mir einige Monate später einen günstigen Mac und ein günstiges iPhone, um auch für iOS entwickeln zu können.

Ab diesem Zeitpunkt wuchs mein Wunsch, mein Wissen über diese großartige Technologie zu teilen. Ich startete damit, Schulungen und Workshops zum Thema Xamarin zu halten, Fachartikel darüber zu schreiben, auf Konferenzen und User Group Meetings darüber zu sprechen und Videokurse für LinkedIn Learning aufzuzeichnen und Kunden bei der Umsetzung ihrer Projekte zu unterstützen.

Mein Enthusiasmus für das Thema Xamarin und insbesondere das Thema Xamarin.Forms ist seit 2013 ungebrochen. Noch immer freue ich mich darüber, wenn ich Entwicklern etwas über Xamarin beibringen und mein Wissen somit weitergeben kann.

Von der Idee zum Buch

Ein Buch zu schreiben ist harte Arbeit. Härter als ich es erwartet hatte, als ich im Sommer 2016 die Anfrage des Carl Hanser Verlags erhielt, ob ich ein Buch über Xamarin schreiben möchte. Zunächst zögerte ich, denn ich konnte zu diesem Zeitpunkt bereits auf zwei gescheiterte Buchprojekte zurückblicken, die beide nach 130 Seiten abgebrochen wurden. Am Ende siegte jedoch meine Leidenschaft für das Thema Xamarin und nach Rücksprache mit meiner Familie und meiner Lektorin sagte ich zu.

Vom Gedanken, ein Buch zu schreiben, bis zum fertigen Buch dauerte es fast fünf Jahre. Als Außenstehender mag man da vielleicht den Kopf schütteln und sich fragen, warum es so lange dauerte. Natürlich könnte ich Ihnen jetzt erzählen, dass gute Texte ähnlich wie guter Wein eine lange Zeit reifen müssen, aber das wäre einfach Unsinn.

Die wirklichen Ursachen sind eine Kombination verschiedener Gründe. Der Hauptgrund war sicherlich, dass ich noch nie ein Buch (zu Ende) geschrieben hatte und schlichtweg nicht wusste, wie man das macht, ein Buch schreiben. Als Ergebnis machte ich das, was in der Entwicklung von Individualsoftware normalerweise der Auftraggeber macht: Ich wollte einfach zu viel. Meine Vision war es, ein Buch zu schreiben, das Xamarin.Android, Xamarin.iOS, Xamarin.Forms und die Entwicklung von UWP Apps von der Idee bis zum Deployment behandelt.

Die ersten 3 ½ Jahre hielt ich an diesem Plan fest und scheiterte auf voller Linie damit. Der mangelnde Fokus kombiniert mit der Tatsache, dass es eigentlich vier Bücher in einem gewesen wären, der Situation, dass dies mein erstes Buch war und dass ständig, wenn ich ein Kapitel fertig hatte, sich etwas Grundlegendes seitens Microsoft, Apple oder Google änderte, führte dazu, dass ich große Teile des Buches oder der Beispielcodes immer wieder neu schrieb und somit auf der Stelle trat.

Im Frühjahr 2020, als ich das Buch schon fast aufgeben wollte, einigte ich mich mit meiner Lektorin beim Carl Hanser Verlag darauf, das Thema des Buches etwas enger zu fassen und mich auf die Entwicklung von Xamarin.Forms-Apps zu konzentrieren. In diesem Bereich liegt sicherlich auch das Hauptinteresse der Leser, zumindest wenn meine Schulungs- und Beratungsanfragen sowie die Abrufe meiner Videokurse repräsentativ sind. Seit längerer Zeit steigt hier das Interesse an Xamarin.Forms, während Xamarin.Android und Xamarin.iOS so gut wie gar nicht mehr nachgefragt werden.

Als Microsoft auf der Build-Konferenz im Mai 2020 dann .NET MAUI als Evolution von Xamarin.Forms ankündigte und kaum ein Wort über Xamarin.Android oder Xamarin.iOS verlor, war die Strategie klar: Dieses Buch muss sich auf Xamarin.Forms konzentrieren. Also startete ich im Mai 2020 von vorne mit dem Buch und schrieb große Teile der bis dahin 130 geschriebenen Seiten neu.

Mangelnder Fokus und eine zu große Vision waren aber nur eine Ursache für den langen Weg bis zum fertigen Buch. Eine andere war, dass das Schreiben dieses Buches nicht meine Haupttätigkeit war. Stattdessen verdiente ich mein Geld in der Anfangsphase hauptsächlich als Trainer und Berater, ehe ich 2018 die Quality Bytes GmbH, eine Softwarefirma spezialisiert auf die Entwicklung von Apps und Cloud Solutions, gründete und aufbaute, was aufwendiger war als ich dachte. Und das Leben besteht nicht nur aus Arbeit, sondern auch aus Familie. Meine wundervolle Frau Ana und unsere drei großartigen Kinder wollten natürlich auch Aufmerksamkeit von mir und Zeit mit mir verbringen. Das Endergebnis war, dass das Schreiben eine von vielen Tätigkeiten war. Im Laufe der Jahre experimentierte ich mit ver-

schiedensten Techniken, um den Schreibprozess zu optimieren und somit schneller fertig zu werden. Der Durchbruch kam in den letzten zwei Monaten. Nachdem ich nach neun Monaten wieder nur 130 Seiten fertig hatte, schrieb ich den Rest des Buches in zwei Monaten. Die angewandte Optimierungstechnik lässt sich übrigens sehr einfach zusammenfassen: „Einfach machen“.

Falls also einer der Leser selbst mit dem Gedanken spielt, sein erstes Buch zu schreiben, dann kann ich ihn oder sie nur herzlich dazu motivieren und folgende Tipps mit auf den Weg geben:

Es ist einfacher, den Umfang des Buches klein zu planen. Nach meiner Erfahrung wächst der Umfang von ganz allein während des Schreibprozesses. Außerdem hilft es ungemein, Gewohnheiten zu schaffen und zum Beispiel jeden Tag ein wenig zu schreiben als sich z. B. vorzunehmen in einer Woche das halbe Buch fertigzustellen.

Danksagung

Wie Sie gerade erfahren haben, war der Weg zu diesem Buch etwas „holprig“. Vermutlich wäre es nie erschienen, wenn es nicht zahlreiche Menschen gegeben hätte, die mir geholfen haben und denen ich meinen Dank aussprechen möchte.

Mein größter Dank gilt meiner Frau Ana, die mich während der ganzen Zeit unterstützt, ermutigt und an mich geglaubt hat. Während der letzten fünf Jahre musste sie nicht nur wegen der Arbeit an diesem Buch, sondern auch wegen meiner anderen beruflichen Verpflichtungen häufig auf meine Gesellschaft verzichten. Sie war dabei stets verständnisvoll und spornte mich an, wenn meine Motivation sank. Dass eine Frau ihrem Mann so viele Freiräume für seine persönlichen Projekte einräumt, ist nicht selbstverständlich und ich bin froh, eine so wundervolle Partnerin gefunden zu haben.

Bedanken möchte ich mich auch bei meinen Kindern Joel und Lyandra und bei meinem Stiefsohn Raúl. Ihr wart stets geduldig und verständnisvoll, wenn ich keine Zeit für euch hatte, da ich an diesem Buch arbeitete.

Mein Dank gilt außerdem meiner Mutter Sylvia, die mir im Grundschulalter einen Atari 800XL zusammen mit dem Buch „Spielend Programmieren lernen“ kaufte und damit die Grundlage für eine erfolgreiche IT-Berufslaufbahn geschaffen hat. Außerdem las sie jedes Kapitel meiner fehlgeschlagenen Buchprojekte sowie meiner Diplom- und Masterarbeit Korrektur und gab mir wertvolle Tipps, um meinen Schreibstil zu verbessern.

Ein besonderer Dank gilt auch meiner Lektorin Sylvia Hasselbach beim Carl Hanser Verlag, die trotz meiner vorherigen missglückten Buchversuche das notwendige Vertrauen in mich hatte und dieses Buchprojekt mit mir startete. Besonders möchte ich ihre Geduld bei den vielen E-Mails und Telefonaten hervorheben, bei denen ich neue Terminverschiebungen ankündigte. Weiter möchte ich mich bei Walter Saumweber bedanken, der dieses Buch Korrektur las und mir in geduldigen Telefonaten freundlich und gut gelaunt wertvolle Tipps zu meinen Texten gab. Darüber hinaus möchte ich mich auch bei Kristin Rothe und Irene Weilhart vom Carl Hanser Verlag für die gute Zusammenarbeit bedanken.

Außerdem möchte ich meinem Informatiklehrer Gerd Larscheid danken, der mein Interesse an der Programmierung Mitte der 90er Jahre, als ich Schüler in seiner Klasse war, förderte. Er ist der beste Informatiklehrer, den ich jemals kennengelernt habe. Von ihm lernte ich nicht nur programmieren, sondern auch, dass es beim Programmieren nicht nur um das

Schreiben von Quellcode, sondern um das Lösen von (Geschäfts-)Problemen mit der Hilfe von Software geht. Diese Erkenntnis war für meine berufliche Laufbahn unverzichtbar.

Bedanken möchte ich mich auch bei meinem guten Freund Oliver, der dieses Buch direkt nach der Ankündigung vorbestellte und mich in den letzten Jahren immer wieder motivierte, es zu Ende zu schreiben.

Selbstverständlich gibt es nicht nur Menschen im persönlichen Umfeld, denen mein Dank gilt, sondern auch Wegbegleiter aus der Entwickler-Community.

Hier gibt es so viele – aktuelle und ehemalige Kollegen, Sprecher und Aussteller auf Konferenzen –, dass es schwerfällt, jemanden besonders hervorzuheben. Trotzdem möchte ich nicht nur all denjenigen, die zur vorherigen Gruppe gehören, meinen pauschalen Dank aussprechen, sondern einige namentlich nennen. Zum einen wäre da Thomas Claudius Huber, mit dem ich mir während des Microsoft MVP Summits 2014, als ich an meinem Windows-8-Buch schrieb und daran verzweifelte, ein Zimmer teilte. Thomas erzählte mir seine inspirierende Geschichte darüber, wie er sein dreimal so dickes WPF-Buch geschrieben hatte. Seine Erzählungen waren so motivierend, dass ich auch heute noch häufig daran denke.

Weiter wäre da Manfred Steyer, den ich seit vielen Jahren von verschiedenen Konferenzen kenne und der ein erfahrener Autor ist. 2019 hielt er in meiner Firma eine Angular-Schulung. Während des Abendessens an einem der Schulungstage fragte ich ihn nach Tipps und er sagte mir, dass ich am besten erst einmal die Beispiele fertig programmiere. Während der Entwicklung der Beispiele würde mir klarwerden, was ich detailliert beschreiben muss, und was nicht. Mit diesem Hinweis hatte er eindeutig recht und es war dieser Hinweis, dessen Umsetzung mir in den letzten zwei Monaten dabei half, Fahrt aufzunehmen.

Mein Dank gilt weiterhin dem Xamarin-Team für das herausragende Produkt sowie den vielen Freiwilligen in der Community, die auf der ganzen Welt ihr Wissen in der Form von Blog-Beiträgen und Vorträgen teilen, oder ihre Freizeit in Open-Source-Projekte stecken.

Außerdem möchte ich auch Dr. Holger Schwichtenberg für die großartige Zusammenarbeit in den letzten Jahren danken. Er war es, der mich als Autor beim Carl Hanser Verlag ins Gespräch brachte und ohne den es dieses Buch wohl niemals geben würde.

Schlussendlich gilt mein Dank natürlich auch **Ihnen!** Vielen Dank, dass Sie sich für dieses Buch entschieden haben. Wenn Sie zu denen gehören, die mehr oder weniger geduldig auf das Erscheinen dieses Buches gewartet haben, dann möchte ich mich an dieser Stelle bei Ihnen für die lange Wartezeit entschuldigen und hoffe, dass der Inhalt die Wartezeit wieder wettmacht.

André Krämer

Mai 2021

■ Der Autor



André Krämer startete seine berufliche Laufbahn 1997 mit einer Ausbildung zum Fachinformatiker für Anwendungsentwicklung, nachdem er zuvor auf der höheren Berufsfachschule für Datenverarbeitung den Abschluss als staatlich geprüfter kaufmännischer Assistent für Datenverarbeitung machte.

Nach seiner Ausbildung bei einem großen IT-Systemhaus arbeitete er zunächst als Entwickler und später als technischer Teamleiter in seinem Ausbildungsbetrieb. Bereits damals fokussierte er sich auf Microsoft-Technologien.

2002 wechselte er zu einem international tätigen Softwarehaus für Software im industriellen Qualitätsmanagement und sammelte dort umfangreiche Erfahrungen mit Microsoft .NET ab der Version 1.0. Während dieser Zeit studierte er nebenberuflich an der FH Köln Wirtschaftsinformatik und erlangte seinen Abschluss als Diplom-Informatiker (FH).

2008 bis 2011 arbeitete er für ein internationales Beratungsunternehmen als Senior Application Architect und absolvierte währenddessen ein nebenberufliches Studium zum Master of Science Wirtschaftsinformatik an der TH Kön.

2012 machte er sich selbständig, wurde Partner im www.IT-Visions.de-Expertennetzwerk, und arbeitete bis Sommer 2018 als Trainer und Berater. Während dieser Zeit führte er überwiegend Individualschulungen für Softwareentwickler durch und schulte vor allem die Themen App-Entwicklung mit Xamarin, Web-Entwicklung mit ASP.NET und DevOps mit Azure DevOps. Im Rahmen seiner Tätigkeit veröffentlichte er über 20 Videokurse für LinkedIn Learning, schrieb zahlreiche Fachartikel und sprach regelmäßig auf Entwicklerkonferenzen.

2013 erhielt er für sein Fachwissen und Community-Engagement von Microsoft die Auszeichnung zum Microsoft Most Valuable Professional (MVP), und wurde seitdem jedes Jahr erneut mit dem Titel ausgezeichnet.



QUALITYBYTES

2018 wagte er den nächsten Schritt und gründete die Quality Bytes GmbH, ein Softwarehaus in Bad Breisig am Rhein zwischen Bonn und Koblenz, und übernahm

dort die Geschäftsführung. Mit seinem mittlerweile 15-köpfigen Team schreibt er dort Individualsoftware auf Projektbasis für seine Kunden.

Der Fokus der Quality Bytes GmbH liegt auf der Entwicklung von Mobile Apps für Android, iOS und Windows mit Xamarin, der Entwicklung von Webportalen für die Microsoft Azure Cloud oder das Selbsthosting auf der Basis von Angular und ASP.NET Core sowie der Implementierung von SAP-Integrationslösungen.

Wenn Sie Projektunterstützung in einem der genannten Bereiche benötigen, dann freuen sich André Krämer und sein Team über Ihre Nachricht unter info@qualitybytes.de oder den Besuch des Internetauftritts der Quality Bytes GmbH, den Sie unter <https://qualitybytes.de> finden.

Sein aktuelles Interesse im Bereich der Softwareentwicklung gilt neben Xamarin, ASP.NET und Azure DevOps den Themen Azure und Docker sowie agilen Entwicklungsmethoden.

Seine Freizeit verbringt André Krämer am liebsten mit seiner Familie. Nachdem die beiden ältesten Kinder bereits erwachsen und ausgezogen sind, beobachten er und seine Frau mit großer Freude die fußballerischen Aktivitäten ihres jüngsten Sohns. Alle drei sind daher häufig auf den Fußballplätzen rund um Koblenz anzutreffen.

Hinweise zum Buch

Dieses Buch besteht aus 15 Kapiteln, die Sie Schritt für Schritt auf Ihrem Weg zum App-Entwickler begleiten. Hier eine kurze Beschreibung des Inhalts der einzelnen Kapitel:

- **Kapitel 1, Einführung**

Im ersten Kapitel erhalten Sie grundlegende Informationen über die Xamarin-Plattform und das zugrunde liegende Ökosystem.

- **Kapitel 2, Schnelleinstieg: Ihre erste App mit Xamarin.Forms in einer Stunde**

Das zweite Kapitel führt Sie Schritt für Schritt von der Idee über die Planung zur ersten App und das alles in einer Stunde. Das Ziel dieses Kapitels ist es, Ihnen Appetit auf mehr zu machen und zu zeigen, wie leistungsfähig Xamarin ist.

- **Kapitel 3, Überblick über Android und Xamarin.Android**

Obwohl Xamarin.Forms die zugrunde liegenden Betriebssysteme und Programmierframeworks weitestgehend abstrahiert, schadet ein grundlegendes Verständnis der Zielplattformen nicht. In diesem Kapitel sehen wir uns daher Grundlegendes zu Android und Xamarin.Android an. Außerdem werfen wir einen Blick darauf, wie Sie Ihr System für die Entwicklung mit Android konfigurieren müssen.

- **Kapitel 4, Überblick über iOS und Xamarin.iOS**

Dieses Kapitel ist das Gegenstück zu Kapitel 3, nur dass es dieses Mal um iOS und nicht um Android geht.

- **Kapitel 5, Überblick über Xamarin.Forms**

Nachdem in den Kapiteln 3 und 4 die spezifischen Plattformen behandelt wurden, geht es in Kapitel 5 um die Grundlagen zu Xamarin.Forms. Wir beleuchten in diesem Kapitel die Xamarin.Forms-Architektur und sehen uns den Aufbau von Xamarin.Forms-Projekten an.

- **Kapitel 6, Architektur von Cross-Plattform-Apps**

In diesem Kapitel sehen wir uns verschiedene Strategien zum Teilen von Quellcode zwischen den verschiedenen Plattformen an. Unter anderem erfahren Sie, wie Sie durch den Einsatz von `Microsoft.Extensions.DependencyInjection` plattformübergreifenden und plattformspezifischen Quellcode separieren können.

- **Kapitel 7, Oberflächendesign mit Layoutcontainern**

Im siebten Kapitel dreht sich alles darum, wie Sie Oberflächenelemente auf dem Bildschirm mit Layoutcontainern wie dem `StackLayout` oder dem `Grid` positionieren können.

■ Kapitel 8, Eingabeformulare anlegen

In Kapitel 7 haben Sie gelernt, wie Sie Steuerelemente auf dem Bildschirm positionieren können, in Kapitel 8 lernen Sie, welche Steuerelemente dies überhaupt sind. Wir werfen in diesem Kapitel einen Blick auf die gängigen Steuerelemente und ihre Einsatzmöglichkeiten.

■ Kapitel 9, Navigation und die Xamarin.Forms-Shell

Die Xamarin.Forms-Shell gehört zu den neueren Steuerelementen von Xamarin.Forms. Ihr Zweck ist es, eine einheitliche, routenbasierte Navigationsstruktur in Apps zu ermöglichen. In Kapitel 9 werfen wir einen Blick auf die Shell und sehen uns gemeinsam an, wie Sie mit der Shell eine seitliche Navigationsleiste, auch Burger-Menü genannt, oder eine Navigation über Tabs realisieren können.

■ Kapitel 10, Listen

Viele mobile Anwendungen bestehen zu großen Teilen aus Listen. Daher ist es nicht verwunderlich, dass Listen ein eigenes Kapitel erhalten und nicht nur ein kleiner Teil von Kapitel 8 sind. In Kapitel 10 sehen wir uns die Klasse `CollectionView` näher an, die die Klasse `Listview` zur Darstellung von Listen abgelöst hat.

■ Kapitel 11, Bilder, Icons und Schriftarten

Nutzer von mobilen Apps haben viel höhere Erwartungen an schöne Benutzeroberflächen als zum Beispiel Anwender klassischer Desktopsoftware. Bilder, Icons und Schriftarten sind drei Elemente, mit denen Sie diese Erwartungen erfüllen können. Ihren praktischen Einsatz sehen wir uns in Kapitel 11 an.

■ Kapitel 12, Styles und Themes in Xamarin.Forms

Styles werden in Xamarin.Forms genutzt, um Steuerelementen ein einheitliches Erscheinungsbild zu geben. Dieses Kapitel gibt Ihnen einen Einblick in die Nutzung von Styles.

■ Kapitel 13, Aufruf von Webservices

Daten von entfernten Webservices spielen für viele Apps eine große Rolle. In diesem Kapitel beschäftigen wir uns daher damit, wie wir Daten von Webservices abrufen, sie effizient zwischenspeichern und Daten an Webservices für schreibende Zugriffe übergeben.

■ Kapitel 14, Lokale Datenhaltung

Neben externen Daten sind selbstverständlich auch interne Daten wichtig. In Kapitel 14 sehen wir uns detailliert an, wie Sie Daten lokal im Dateisystem oder in einer lokalen SQLite-Datenbank speichern können.

■ Kapitel 15, Zugriff auf native Gerätefunktionen mit Xamarin.Essentials

In den Kapiteln 1 bis 14 haben wir uns überwiegend mit den plattformübergreifenden Aspekten der App-Entwicklung beschäftigt. Das Besondere einer App ist allerdings nicht nur, dass Sie sie überall hin mitnehmen können, sondern dass sie Sensoren und Funktionen des Mobilgeräts oder des mobilen Betriebssystems nutzen können. In diesem Kapitel werfen wir einen Blick auf Xamarin.Essentials, eine Bibliothek, die uns den plattformübergreifenden Zugriff auf Gerätefunktionen ermöglicht.

■ An wen richtet sich dieses Buch und was sollten Sie als Leser mitbringen?

Dieses Buch ist für Softwareentwickler gedacht, die plattformübergreifende Apps mit **Xamarin.Forms** für Android und iOS schreiben möchten. Die primäre Zielgruppe sind Einsteiger und Fortgeschrittene, aber auch für Experten ist das ein oder andere Kapitel dabei.

Für die Lektüre dieses Buches sollten Sie bereits Erfahrung mit C#, Visual Studio und dem .NET Framework haben. Sie müssen kein Experte sein, aber ohne Vorwissen im Bereich der .NET-Entwicklung werden Sie wenig Freude an diesem Buch haben. Sollten Sie in diesem Umfeld noch keine Erfahrung haben, dann empfehle ich Ihnen, sich zunächst C#- und .NET-Basiswissen über Fachbücher oder eines der vielen Tutorials aus dem Internet anzueignen. Anschließend können Sie mit diesem Buch loslegen.

Wie Sie das Buch am besten lesen

Der Inhalt des Buches ist so aufgebaut, dass Sie Schritt für Schritt das notwendige Wissen erlangen, um Apps mit Xamarin.Forms zu entwickeln. Daher empfehle ich Ihnen, das Buch von vorne nach hinten zu lesen. Jedes Kapitel startet mit einem kurzen Überblick und endet mit einer Zusammenfassung, anhand deren Sie kontrollieren können, ob Sie die Kernaussagen des Kapitels nachvollziehen konnten.

Wenn Sie bereits Erfahrung mit Xamarin.Forms haben, dann spricht natürlich nichts dagegen, einzelne Kapitel zu überspringen oder sich themenbezogen auf einzelne Kapitel zu konzentrieren.

An dieser Stelle möchte ich Sie jedoch darauf hinweisen, dass die Kapitel und vor allem der Quellcode der durchgehenden Beispiel-App jeweils auf den Inhalten vorheriger Kapitel aufbauen. Wenn Ihnen beim Sprung in ein Kapitel auffällt, dass Ihnen Grundlagen fehlen, dann empfehle ich Ihnen einen Blick in die vorherigen Kapitel.

Abschließend ist es mir noch wichtig anzumerken, dass dieses Buch kein Ersatz für die offizielle Xamarin-Online-Dokumentation (<https://docs.microsoft.com/de-de/xamarin>) ist. Dafür ist diese schlichtweg zu umfangreich. Dieses Buch legt den Fokus auf den praktischen Einsatz von Xamarin und kann dementsprechend nicht jede verfügbare Funktion beschreiben.

■ Die Codebeispiele

Zu fast jedem Kapitel gibt es Codebeispiele. Diese können Sie unter

<https://github.com/andrekraemer/xamarin-buch>

entweder vollständig als ZIP-Datei herunterladen oder über einen beliebigen Git-Client klonen.

Je Kapitel gibt es einen Ordner mit dem Namen *Kap<Nr>*, also zum Beispiel *Kap02* für das Kapitel 2 oder *Kap15* für das Kapitel 15. Außerdem gibt es auf oberster Ebene noch den

Ordner *Beispieldateien*. In diesem Ordner finden Sie Bilddateien, Schriftarten und Daten, die Sie zur Bearbeitung der durchgängigen Beispielanwendung benötigen.

Die durchgängige Beispiel-App

Dieses Buch verfolgt zwei Strategien für die Codebeispiele. In jedem Kapitel-Ordner gibt es Unterordner, die auf dem Wort *Demo* enden. In diesen Unterordnern finden Sie die Beispielfiles zu den Listings der entsprechenden Kapitel.

Ab Kapitel 5 gibt es außerdem in den meisten Kapitel-Ordnern den Unterordner *ElVegetarianoFurio*. Dabei handelt es sich um den Quellcode der Beispiel-App, die wir Schritt für Schritt in diesem Buch entwickeln werden. Der Quellcode der Beispiel-App in den jeweiligen Kapitelordnern entspricht dem Fortschritt zum Ende des Kapitels.

Inhalt der Beispiel-App ist die Umsetzung einer Speisekarten-App für das fiktive vegetarische, spanische Restaurant *El Vegetariano Furio* (zu Deutsch: Der wütende Vegetarier).

Bild 1.1 zeigt die Startseite der fertigen Beispiel-App, Bild 1.2 zeigt die Speisekarte in der Übersicht sowie die Detailseite einer Speise.

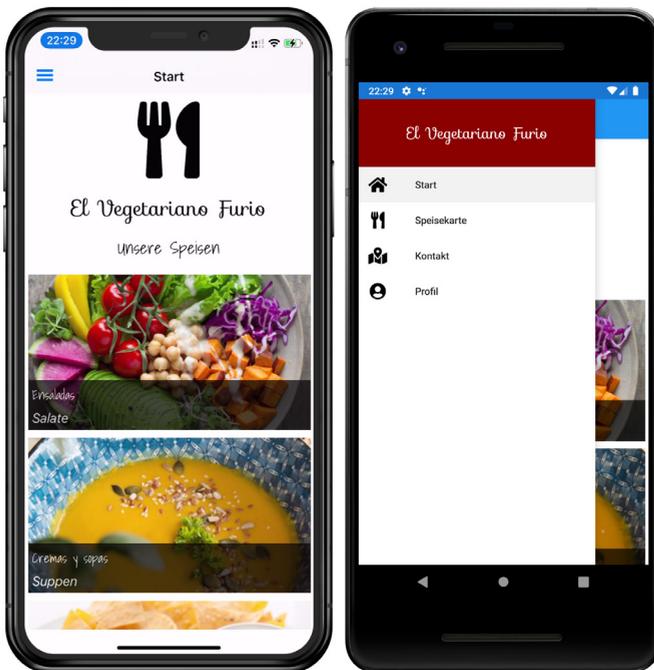


Bild 1.1 Die Startseite der Beispiel-App

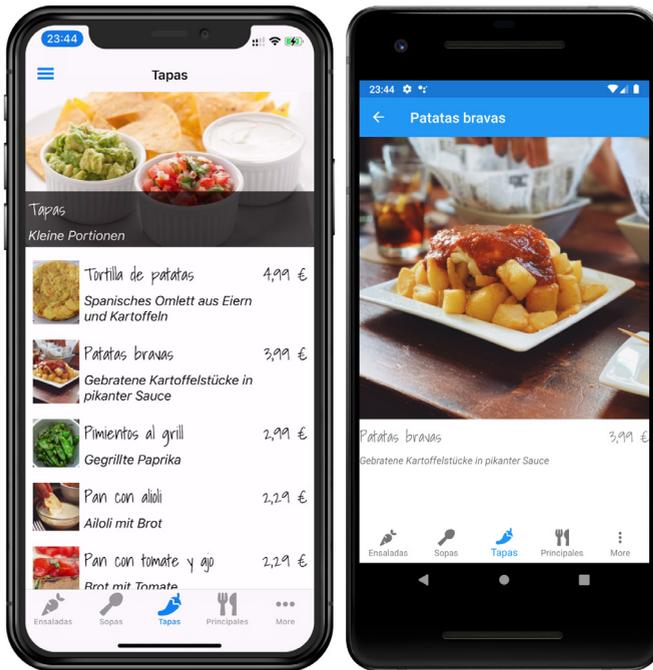


Bild 1.2 Übersicht über die Speisekarte und die Detailseite der Beispiel-App

Den vollständigen Quellcode der App werden wir Schritt für Schritt gemeinsam in den einzelnen Kapiteln schreiben. Es wird nichts ausgelassen oder übersprungen, sodass Sie in der Lage sind, mit zu programmieren. Die Musterlösung der einzelnen Kapitel finden Sie, wie bereits gesagt, im Ordner *ElVegetarianoFurio* der einzelnen Unterkapitel.

Wenn Sie das Beispiel nicht durchgängig entwickeln, sondern mittendrin einsteigen möchten, dann ist dies natürlich kein Problem. Kopieren Sie einfach meine Musterlösung vom vorherigen Kapitel und legen Sie los.

■ Fehler, Aktualisierungen und Kontakt zum Autor

Kein Buch ist fehlerfrei. Zumindest liest man dies in allen Fachbüchern, die man am Markt erhält. Wenn also bereits alle Autoren vor mir eingeräumt haben, dass ihr Buch Fehler hat, dann sollte ich wohl davon ausgehen, dass sich auch in meinem Buch trotz sehr gründlicher Prüfung – schließlich habe ich mir fünf Jahre Zeit genommen – Fehler eingeschlichen haben.

Korrekturen zu fachlichen Fehlern im Text sowie Rechtschreibkorrekturen finden Sie auf der Webseite zu diesem Buch, die Sie unter

<https://andrekraemer.de/xamarin-buch>

erreichen.

Eventuelle Fehler in den Codebeispielen werde ich kontinuierlich auf dem GitHub-Repository zu diesem Buch unter

<https://github.com/andrekraemer/xamarin-buch>

korrigieren. Wenn Sie in den Beispielen einen Fehler finden sollten, der noch nicht korrigiert ist, dann zögern Sie nicht, einen *Issue* über das GitHub-Repository anzulegen.

Dieses Buch lebt, wie alle Bücher, von Rückmeldungen und Kritiken der Leser. Ich lade Sie also herzlich ein, Rezensionen über das Buch, positiv wie negativ (am liebsten natürlich positiv) auf den gängigen Internetplattformen zu hinterlassen.

Scheuen Sie sich auch nicht, mich bei Fragen oder Anmerkungen zum Buch direkt per E-Mail unter andre@andrekraemer.de oder auf Twitter unter <https://twitter.com/codemurai> anzuschreiben. Bitte haben Sie aber Verständnis dafür, dass ich unter der E-Mail-Adresse nur Fragen, die sich direkt auf den Inhalt des Buches beziehen, beantworten kann.

Darüber hinausgehende Fragen kläre ich jedoch gerne in Schulungen und Workshops, die Sie auch unter dieser E-Mail-Adresse anfragen können.

■ Etwas zu Sprache, Anglizismen und „Denglisch“

Während des Schreibens dieses Buches habe ich mir Mühe gegeben, verständliche deutsche Sätze zu schreiben. Ich hoffe dies durchgehend geschafft zu haben und setze große Hoffnungen auf das Lektorat, falls dem nicht so sein sollte.

Überall, wo es sinnvoll war, habe ich außerdem deutsche statt englische Fachbegriffe genutzt. So werden Sie in diesem Buch häufiger Begriffe wie *Schaltfläche* statt *Button* oder *Ereignisbehandlungsroutine* statt *Event-Handler* finden. Aufgrund der Erfahrung aus meinen Schulungen und Workshops habe ich auch die deutschen Spracheinstellungen für Visual Studio für dieses Buch genutzt, da ein nicht unerheblicher Teil der Teilnehmer meiner Schulungen Visual Studio auf Deutsch benutzen. Daher finden Sie auch Begriffe wie *Projektmappen-Explorer* statt *Solution-Explorer* in diesem Buch.

Die deutsche Variante habe ich allerdings nur für etablierte Begriffe genutzt. Die krampfhaftige Übersetzung von Begriffen, die durchgängig jeder noch so puristische Entwickler nutzt, den ich kenne, habe ich mir gespart. Statt Sie also rätseln zu lassen, was es mit einem Wegbestimmungssystem auf sich haben könnte, habe ich lieber den etablierten Ausdruck Routing-System genutzt.

In dem Wissen, dass man es niemals allen recht machen kann, nehme ich auch hier gerne Ihre Rückmeldung direkt per E-Mail unter andre@andrekraemer.de oder auf Twitter unter <https://twitter.com/codemurai> entgegen.

■ Ein Wort zu .NET MAUI

Im November 2021 wird .NET MAUI, der Nachfolger von Xamarin.Forms, erscheinen. Natürlich stellt sich nun die berechnete Frage, ob es sich trotzdem noch lohnt, dieses Buch zu lesen. Wenn Sie dieses Buch vor dem Erscheinen von .NET MAUI lesen, dann ist die Antwort auf diese Frage ganz klar „ja!“. Ein Großteil des Wissens, dass Sie sich bei der Lektüre dieses Buches aneignen, wird auf .NET MAUI übertragbar sein. Mit der Hilfe dieses Buches können Sie also das notwendige Vorwissen erlangen, um mit .NET MAUI ab November durchstarten zu können.

Während des Verfassens dieses Buches habe ich auch darauf geachtet, Themen, die unter .NET MAUI vollständig anders gelöst werden, wie zum Beispiel `Renderer`, auszulassen. Außerdem habe ich statt des `Xamarin.Forms.DependencyService`, den es unter .NET MAUI nicht mehr geben wird, den Einsatz von `Microsoft.Extensions.DependencyInjection` für das Thema Dependency Injection erläutert, da dies auch bei .NET MAUI Anwendung finden wird.

Selbst nach der Veröffentlichung von .NET MAUI kann es sinnvoll sein, dieses Buch zu lesen. Zum einen für den Fall, dass Sie eine bestehende Anwendung pflegen müssen, die noch nicht zu .NET MAUI migriert wird, zum anderen falls es noch kein Buch zu .NET MAUI gibt. Denn auch für diesen Fall ist das vorliegende Buch besser als gar keines, schließlich ist die Ähnlichkeit beider Frameworks sehr hoch.

Ein Buch zum Thema .NET MAUI habe ich übrigens bereits in Planung. Dieses Mal bin ich auch guter Dinge, dass es in kürzerer Zeit, relativ nah am Erscheinungstermin von .NET MAUI, fertig sein wird.

In diesem Sinne wünsche ich Ihnen viel Spaß und Erfolg bei der Lektüre dieses Buches und freue mich auf ein Wiedersehen beim nächsten Buch.

1

Einführung

There's no chance that the iPhone is going to get any significant market share. No chance.

Steve Ballmer

Unzutreffender hätte Steve Ballmer, der ehemalige CEO von Microsoft, im April 2007 die Zukunftsaussichten für das iPhone wohl kaum einschätzen können.

Wenige Wochen vor dessen Veröffentlichung ging er in einem Interview mit der USA Today davon aus, dass der Marktanteil des im Januar 2007 angekündigten iPhones ungefähr zwei bis drei Prozent ausmachen wird und Microsoft mit seinem Betriebssystem den Mobiltelefonmarkt mit einer Verbreitung von ca. 60 bis 80 % beherrschen wird.

Heute wissen wir, dass Ballmer nicht nur vollkommen falsch mit seiner Einschätzung lag, sondern dass Apple mit der Einführung des iPhones den Wendepunkt im Markt für Mobiltelefone einläutete. Während zuvor meist sogenannte Feature Phones, also Mobiltelefone im klassischen Sinne, die weder über einen Touchscreen noch einen Internetzugang oder App Store verfügten, verkauft wurden, lag der Marktanteil der Smartphones bereits im dritten Quartal 2014 laut einer Studie des amerikanischen IT-Marktforschungsunternehmens Gartner bei 66 Prozent. In der Studie wurde davon ausgegangen, dass bis 2018 neun von zehn Mobiltelefonen ein Smartphone sind.

Gemeinsam mit dem Siegeszug der Smartphones stieg auch der Bedarf an sogenannten Enterprise Mobile Apps, also solchen Apps, die Mitarbeiter, Partner oder Kunden von Unternehmen bei ihrer täglichen Arbeit unterstützen. Beispiele für solche Apps wären unter anderem eine Zeiterfassungs-App, eine Reisekosten-App, eine Customer-Relationship-Management (CRM)-App, eine Support-/Ticket-App, oder eine mobile Variante einer anderen internen Software. Gartner geht davon aus, dass der Bedarf an solchen Enterprise Mobile Apps die verfügbaren Entwicklungskapazitäten um den Faktor fünf übersteigen wird.

Verschärft wird diese Problematik dadurch, dass eine Mobile App häufig nicht nur für ein Betriebssystem entwickelt werden muss, sondern für zwei oder drei. Dies liegt daran, dass es für Unternehmen in der Regel nicht möglich ist, ihren Kunden die Nutzung eines bestimmten mobilen Betriebssystems, zum Beispiel iOS, vorzuschreiben. Selbst innerhalb des Unternehmens ist diese Möglichkeit nicht immer gegeben, da immer mehr Firmen auf eine Strategie setzen, die sich Bring Your Own Device (BYOD) nennt, bei der die Angestellten ihre privaten Mobiltelefone, Tablets usw. mit ins Unternehmen einbringen. Auch in diesem Fall kann das Unternehmen die Verwendung eines bestimmten mobilen Betriebssystems meist nicht vorschreiben, da es sich um die privat angeschafften Geräte der Mitarbeiter handelt.

Viele Unternehmen stehen somit vor der Herausforderung, dass sie eine oder mehrere Apps für zwei oder mehr mobile Betriebssysteme entwickeln müssen. Dabei kann es sich

um eine interne App zur Optimierung der eigenen Unternehmensabläufe, um eine externe App, die zur Kommunikation mit den Kunden genutzt wird, oder – im Fall von Softwarehäusern – um eine mobile Variante eines bereits bestehenden Produkts handeln.

Natürlich stellt sich nun die Frage, wie diese Herausforderung möglichst wirtschaftlich gemeistert werden kann. So stehen neben der Variante, für jedes mobile Betriebssystem eine eigene App mit den plattformspezifischen Werkzeugen zu entwickeln, weitere Alternativen zur Verfügung, die es ermöglichen, Apps für mehrere Plattformen auf einer gemeinsamen Code-Basis zu erzeugen. Zu den bekanntesten Alternativen gehören React Native, Flutter, Native Script, Apache Cordova und Xamarin.

In diesem Buch werden Sie lernen, wie Sie Apps für Android und iOS auf Basis der Xamarin-Plattform unter der Verwendung von Xamarin.Forms entwickeln können. Das Buch richtet sich dabei hauptsächlich an Entwickler, die aktuell Software auf Basis der Microsoft .NET-Plattform unter der Verwendung von Visual Studio entwickeln und eine Enterprise bzw. Line-Of-Business (LOB)-App entwickeln möchten. Die Entwicklung von Spielen für mobile Endgeräte ist somit explizit ausgeklammert.

■ 1.1 Xamarin und die Xamarin-Plattform

Wenn man von Xamarin spricht, dann meint man in der Regel die *Xamarin-Plattform*, ein Produkt der Firma Microsoft. Sie ermöglicht die Entwicklung von Apps für iOS und Android in Microsofts Programmiersprache C#. Eine spezielle Version für Windows-Apps gibt es nicht, da Windows-Apps bereits von Haus aus in C# entwickelt werden.

Grundlage für die Xamarin-Plattform ist das Mono Framework. Mono ist eine quelloffene Implementierung des Microsoft .NET Framework für Linux, Mac OSX und Windows, die unter einer MIT-Lizenz steht. Das Projekt wurde im Dezember 2000 durch Miguel de Icaza gestartet. Im Mai 2011 gründete er gemeinsam mit einem Partner die Firma Xamarin, die im März 2016 von Microsoft aufgekauft wurde.

In diesem Zuge wurde das Software Development Kit (SDK) der Xamarin-Plattform unter eine MIT-Open-Source-Lizenz gestellt. Die Integration in die Entwicklungsumgebung Visual Studio bleibt jedoch zunächst proprietär.



Xamarin, Xamarin-Plattform und .NET Multi-platform App UI (MAUI)

Die *Xamarin-Plattform* ist ein Produkt von Microsoft, welches ursprünglich von der Firma Xamarin entwickelt wurde. *Xamarin* ist somit der Name einer durch Microsoft aufgekauften Firma, der Begriff Xamarin-Plattform bzw. auf Englisch Xamarin Platform steht für das Produkt zur Entwicklung plattformübergreifender Lösungen.

Viele Entwickler sagen lediglich Xamarin, wenn sie von der Xamarin-Plattform sprechen. Zur Steigerung des Leseflusses halte ich es auch in diesem Buch so und werde statt Xamarin-Plattform an den meisten Stellen einfach Xamarin schreiben. Selbstverständlich ist dann stets das Produkt und nicht die Firma gemeint.