

Content-Verwertungsmodelle und ihre Umsetzung in mobilen Systemen

Apps – Entwurf und Programmierung



4FriendsOnly.com
Internet Technologies AG

Vorlesung im Sommersemester an der
Technischen Universität Ilmenau von
Privatdozent Dr.-Ing. habil. Jürgen Nützel,
Vorstand der 4FriendsOnly.com Internet Technologies AG (4FO AG)
JN (at) 4FO (dot) DE

*Diese Folien und weitere Informationen unter:
www.juergen-nuetzel.de/content_verwertungsmodelle_mobile_umsetzung.html*

Überblick

- Was ist eine App?
- Verschiedene Grundtypen
 - Native App, Web-App und hybride App
- User-Interface (UI) Prototype Design
- Unterschiedliche Wege der Programmierung und ausgewählte Entwicklungssysteme
 - Native, für ein Betriebssystem angepasst
 - Native, für mehrere Betriebssysteme
 - Hybride

Was ist eine App?

App steht für den englischen Begriff application (bzw. Applikation). Und bezeichnet damit eine Anwendungssoftware (bzw. ein Anwendungsprogramm). Anwendungsprogramme (wie z.B. Textverarbeitung oder Computerspiele) dienen der Lösung von Anwenderproblemen. Anwendungssoftware grenzt sich von Systemsoftware und systemnaher Unterstützungssoftware (wie z.B. Betriebssysteme oder Compiler) ab.

Im Deutschen wird die Abkürzung App seit dem Erscheinen des Apple App Stores (2008) fast ausschließlich mit mobiler App gleichgesetzt, also Anwendungssoftware für Mobilgeräte wie Smartphones und Tablet-Computer. Apple hat diese Kurzform von Application eingeführt.

Sicherlich auch mit bewusster Anspielung auf den Firmennamen Apple. Inzwischen versucht Microsoft unter Windows 8/10 auch Desktop-Anwendungen als Apps zu bezeichnen.

Hier steht App weiterhin nur für Anwendungssoftware für Mobilgeräte.

<http://de.wikipedia.org/wiki/Anwendungssoftware>

Zwei App-Grundtypen

Native App

Eine App, die speziell für ein einzelnes Betriebssystem ausschließlich mit dem vom Betriebssystem-Anbieter hierfür bereitgestellten SDK (System Development Kit) entwickelt wurde. Wird über App-Stores verbreitet und auf dem Endgerät installiert.

Vorteile:

Eine native App kann die über das jeweilige Betriebssystem angebotenen Hardware-Ressourcen optimal nutzen. Native Apps können daher sehr performant und Ressourcen-sparend sein.

Nachteile:

Native Apps müssen für jedes Betriebssystem getrennt entwickelt werden. Die Entwicklungskosten sind daher sehr hoch.

Web-App

Eine Web-App läuft komplett im Browser ab. Eine Web-App wird als Web-Seite vom Web-Server geladen. Neben HTML wird CSS und JavaScript genutzt.

Vorteile:

Da HTML, CSS und JavaScript auf allen Betriebssystemen standardisiert sind, laufen Web-Apps auf allen Systemen. Dies reduziert die Entwicklungskosten.

Nachteile:

HTML, CSS und JavaScript müssen vom Browser interpretiert werden. Daher ist die Performanz gering. Der Browser kann nicht auf alle Hardware-Ressourcen (wie z.B. Kamera und Datei-System) zugreifen. Kann nicht über App-Stores verbreitet werden.

Hybride Apps als Mischung

Hybride App

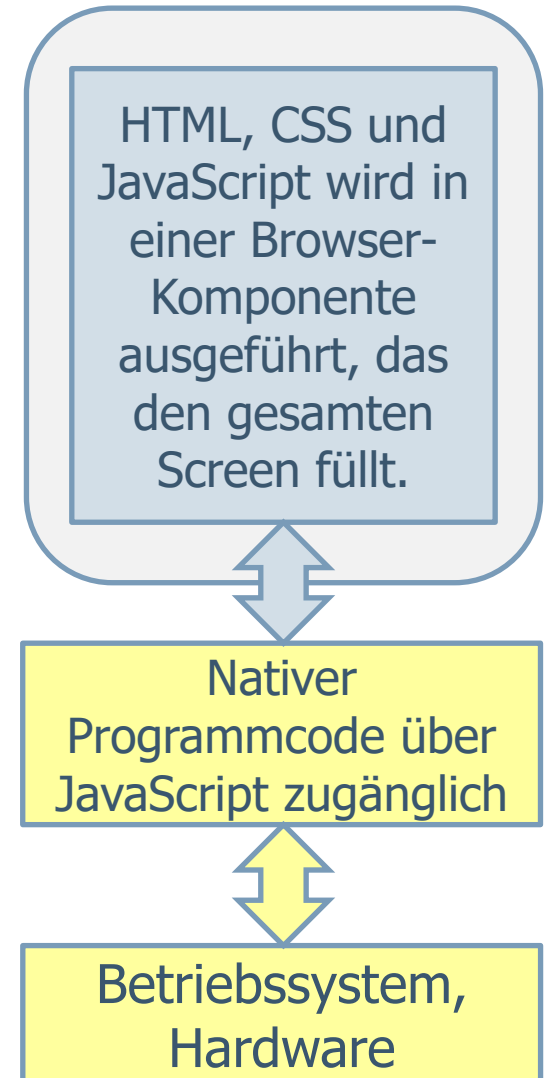
Eine hybride App, ist eine native App, die für die Darstellung der Oberfläche eine Browser-Komponente nutzt. Der größte Teil der App ist in HTML, CSS und JavaScript erstellt. Nativer Code, der über JavaScript der Oberfläche zu gänglich ist, ermöglicht den Zugriff auf native Gerätefunktionen.

Vorteile:

Als native App, kann die hybride App, alle über das jeweilige Betriebssystem angebotenen Hardware-Ressourcen nutzen. Sie kann über App-Stores verbreitet werden. Die Oberfläche und die Interaktionslogik sind für alle Betriebssysteme gleich. Dies reduziert die Entwicklungskosten.

Nachteile:

HTML, CSS und JavaScript müssen vom Browser interpretiert werden. Daher ist die Performanz/Reaktionszeit der Oberfläche oft zu gering.



User-Interface (UI) Prototype Design

Wireframe / Mock-Up

Wireframes (Drahtgerüst) oder Mock-Ups (Attrape) werden dazu benutzt, um einen sehr frühen konzeptuellen (nicht funktionsfähigen) Prototypen eines App-Frontends darzustellen. Navigation und Nutzerführung und wesentliche Inhaltsbereiche sollten Teil dieses Skeletts sein.

Im Gegensatz zu statischen Wireframes ermöglichen dynamische Wireframes die Navigation zwischen den einzelnen Ansichten.

What is a Wireframe?

www.youtube.com/v/T0vt3nLZKks

Wireframing-Tools

www.mashable.com/2013/04/02/wireframing-tools-mobile

Proto.io: www.proto.io

Proto.io - Silly-fast mobile prototype

www.youtube.com/v/6t2YjjvKZR8



Moqups 2: <https://app.moqups.com>

Balsamiq: www.balsamiq.com

UXPin: www.uxpin.com

Pencil: <http://pencil.evolus.vn/>

Integrierte Entwicklungsumgebungen

- IDEs für native Apps
 - Xcode von Apple für iOS: <https://developer.apple.com/xcode/>
 - Android Studio: <http://developer.android.com/sdk>
 - Visual Studio Community 2017 für Windows:
<http://www.visualstudio.com/>
- Cross-Plattform
 - Xamarin: www.xamarin.com
- Hybrid
 - XDK von Intel: <http://xdk-software.intel.com/>
- Ganz anders (cool, online grafisch):
 - App Inventor: <http://appinventor.mit.edu/explore/>

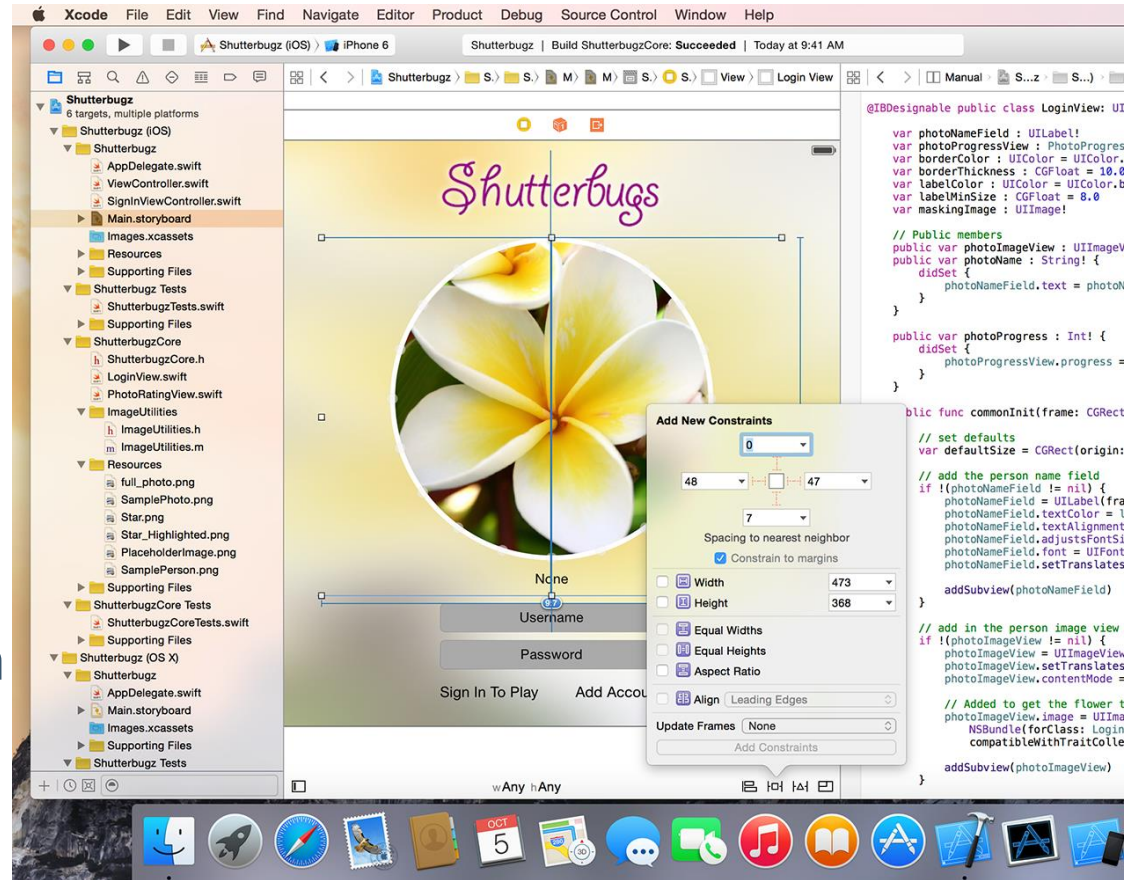
Xcode (Version 8)

IDE von Apple zur Erstellung nativer Apps für sämtliche iOS Geräte. Leider nur für Mac verfügbar.

Xcode hat inzwischen eine sehr gute Unterstützung bei der Erstellung des UI und der Verknüpfung mit dem Code.

Xcode ist kostenlos.

Die Programmierung erfolgt in Objective C. Ab Version 6 zusätzlich in Swift.

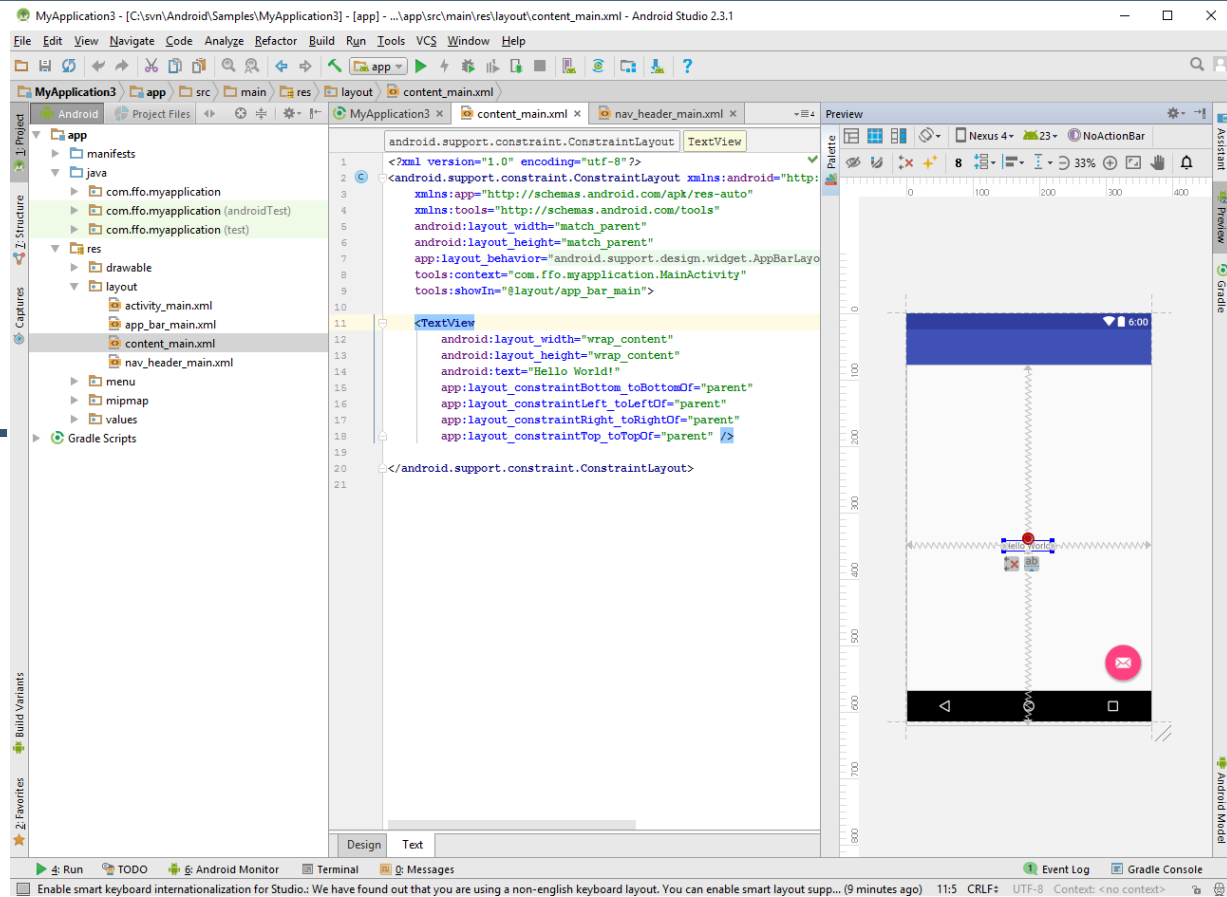


Swift Xcode 6 Tutorial – Hello World App

www.youtube.com/v/K4XycF10_1A

Android Studio 2.x

Google hat auf Basis von IntelliJ das Android Studio entwickelt. Android Studio löste die für Java spezialisierte OpenSource IDE Eclipse ab. Das Android SDK beinhaltet auch einen Emulator, der flexibel für verschiedene Geräte angepasst werden kann.



Android Studio Tutorial for Beginners (Step by Step tutorial)

https://www.youtube.com/playlist?list=PLS1QuWo1RIbb1cYyzZpLFCKvdYV_yJ-E

Xamarin

Die Entwickler von Mono (quelloffene Implementierung des .NET Framework) gründeten im Jahr 2011 die Firma Xamarin. Xamarin wurde im Februar 2016 von Microsoft übernommen. Xamarin ist nun Bestandteil von Visual Studio.

Mit der Programmiersprache C# können Entwickler iOS-, Android- und Windows-Apps schreiben. Der Quellcode kann für alle Plattformen gemeinsam benutzt werden.

<https://de.wikipedia.org/wiki/Xamarin>

Xamarin Platform

Mit den Komponenten Xamarin.iOS und Xamarin.Android ist es möglich, native Apps für iOS, Android und Windows in C# entweder mit **Xamarin Studio** oder Visual Studio zu entwickeln.

Xamarin.Forms erlaubt portable Bedienelemente zu verwenden (über XAML), welche die eigentlichen Bedienelemente von Android, iOS und Windows Phone aufrufen.

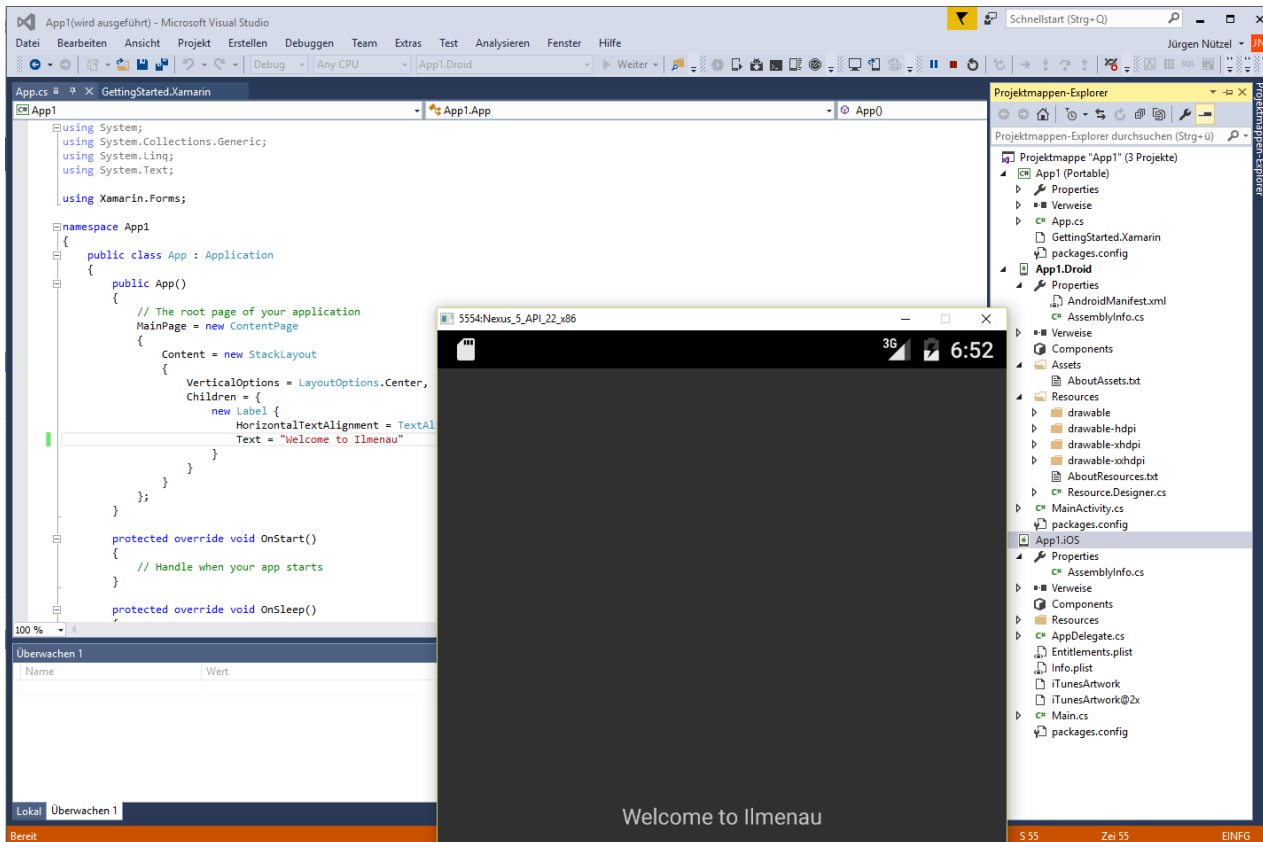
Your first Xamarin.Forms App (2014)

www.youtube.com/v/xWyOcEpNyXQ

Visual Studio Community 2017

(mit Xamarin.forms)

Visual Studio und Xamarin Studio nutzen das Android SDK, um native Android Apps zu erstellen. Native iOS Apps können mit Xamarin Studio für den Mac oder Visual Studio erstellt werden. Visual Studio benötigt dazu einen Mac (mit Xcode und Xamarin) im lokalen Netzwerk.

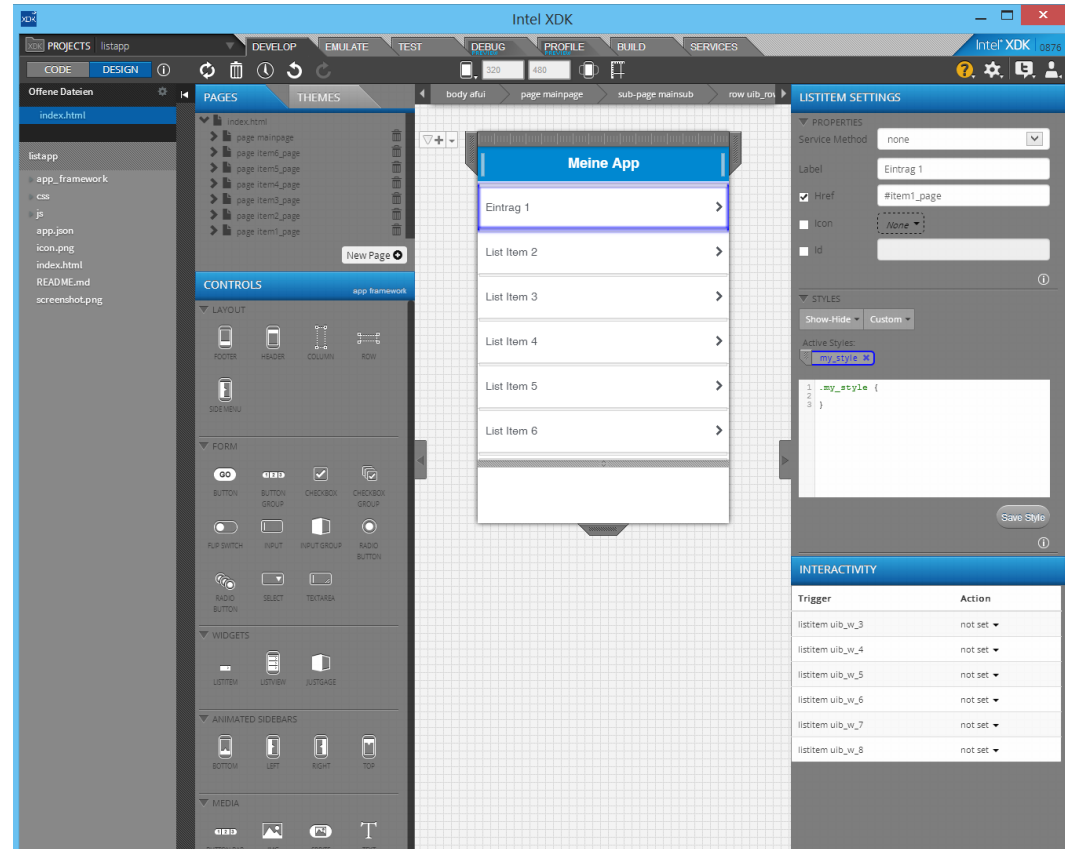


Intel XDK

Plattformübergreifende HTML5 Lösung, mit der Entwickler Web- und Hybrid-Apps einmal programmieren und dann über viele App-Stores und für viele Geräteformate vertreiben können.

Schlanker Workflow von Design bis App-Store.

Intel XDK gibt es als kostenlosen Download für Windows 7 und 8, Apple OS X und Ubuntu. Linux.



Getting Started With Mobile Apps Video

<https://software.intel.com/en-us/xdk/videos/getting-started-with-the-intel-xdk>

Weitere Informationen

- Anwendungssoftware (Apps):
<http://de.wikipedia.org/wiki/Anwendungssoftware>
- iOS Human Interface Guidelines
<https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>
- Wireframeing-Tools:
<http://mashable.com/2013/04/02/wireframing-tools-mobile/>
- 22 Good Prototype and Wireframe Tools:
<http://www.onextrapixel.com/2013/03/01/22-good-prototype-and-wireframe-tools-for-mobile-and-web-design/>
- Objective C: <http://de.wikipedia.org/wiki/Objective-C>
- Swift: [http://de.wikipedia.org/wiki/Swift %28Programmiersprache%29](http://de.wikipedia.org/wiki/Swift_%28Programmiersprache%29)
- Download von Visual Studio Community:
www.visualstudio.com/en-us/products/visual-studio-community-vs