

























































## 4 Auswahl Rahmenwerk

### 4.1 Vergleich der einzelnen Rahmenwerke

Für die App-Programmierung stehen viele verschiedene Rahmenwerke bereits zur Verfügung. Dieser Abschnitt beschäftigt sich mit dem Vergleich der einzelnen Möglichkeiten, sowie einer endgültigen Auswahl eines der Rahmenwerke zur Verwendung im Diplomprojekt.

**Microsoft Xamarin** ist ein Rahmenwerk, mit dem man Multiplatform-Applikationen für unter Anderem Android, iOS, UWP und noch viele weitere Plattformen erstellen kann. Es basiert auf Microsoft's .NET-Rahmenwerk und dem Mono-Projekt, welches sich als Ziel gesetzt hat, das .NET-Rahmenwerk auf andere Plattformen zu portieren. Xamarin bietet mehrere Projekttypen an, darunter Xamarin.Android und Xamarin.iOS für native App-Entwicklung und Xamarin.Forms für großteils plattformunabhängige Entwicklung. Eine Xamarin.Forms-Solution besteht daher aus drei Teilen: einem Portable/.NET-Standard-Projekt und je ein natives Xamarin-Projekt für jede Zielplattform. Der Vorteil Xamarin's ist, dass der Großteil des geschriebenen Codes im plattformunabhängigen Projekt bleibt und nur für wenige Funktionen auf native Programmierung zurückgegriffen wird, wie zum Beispiel für hardwarenahe Audio-Aufnahme. Außerdem ist das Xamarin-Projekt Open Source, das bedeutet jeder kann den Quellcode betrachten und unter Umständen Verbesserungen vorbringen.

**Apache Cordova** ist ein Appentwicklungs-Rahmenwerk welches ursprünglich von der Firma Nitobi entwickelt wurde. Im Jahr 2011 wurde Nitobi von Adobe aufgekauft und das Rahmenwerk wurde zu PhoneGap umgetauft. Später wurde eine quelloffene Version unter dem ursprünglichen Namen veröffentlicht. Apps werden mittels gängiger Webtechnologie, wie zum Beispiel JavaScript, CSS und Ähnlichen, entwickelt und realisiert, weshalb das Rahmenwerk alle gängigen Plattformen unterstützt. Der Vorteil von Cordova liegt im enormen Support dieser Webtechnologien, jedoch sind Sprachen wie JavaScript eher weniger für Back-End-Programmierung geeignet. Noch dazu kommt, dass an der HTL Anichstraße großteils die Programmierung nur in C und C# gelehrt wird, weshalb die Entwicklung mit JavaScript nicht realistisch ist.

**Andere** Neben Apache Cordova existieren noch viele weitere Rahmenwerke, die wie Cordova auf Webtechnologie aufbauen, um so die plattformunabhängigkeit garantieren zu können. Allerdings bringen diese alle dieselben Probleme wie Cordova mit sich.

## 4.2 Build-Umgebung

Aufgrund obiger Aufstellung wurde Visual Studio 2019 der Firma Microsoft als Entwicklungsumgebung gewählt. Diese steht für Schüler und Private gratis auf der Seite <https://visualstudio.microsoft.com/> zum Download zur Verfügung. Visual Studio ist das offizielle Werkzeug für die Programmierung mit dem Xamarin-Rahmenwerk.

### 4.2.1 Installation Visual Studio

Die minimalen Hardware-, bzw. Software-Anforderungen können der **Microsoft-Dokumentation des Visual Studio** entnommen werden:

Unterstützte Betriebssysteme	<ul style="list-style-type: none"> <li>- Windows 10, Version 1703 oder höher (LTSC und S werden nicht unterstützt)</li> <li>- Windows Server 2019</li> <li>- Windows Server 2016</li> <li>- Windows 8.1 (mit Update 2919355)</li> <li>- Windows Server 2012 R2 (mit Update 2919355)</li> <li>- Windows 7 SP1 (mit neuesten Windows-Updates)</li> </ul>
Hardware	<ul style="list-style-type: none"> <li>- 1,8-GHz-Prozessor oder schneller; Quad-Core oder besser empfohlen</li> <li>- 2 GB RAM; 8 GB RAM empfohlen</li> <li>- Speicherplatz auf der Festplatte: Mindestens 800 MB, je nach installierten Features bis zu 210 GB des verfügbaren Speicherplatzes. Eine typische Installation erfordert 20–50 GB freien Speicherplatz.</li> <li>- Festplattengeschwindigkeit: Zur Verbesserung der Leistung installieren Sie Windows und Visual Studio auf einem Festkörperlaufwerk (SSD).</li> <li>- Grafikkarte, die eine Auflösung von mindestens 720p (1280 x 720) unterstützt. Visual Studio funktioniert am besten mit einer Auflösung von WXGA (1366 x 768) oder höher.</li> </ul>

Abbildung 4.1: Soft- und Hardware-Anforderungen Visual Studio

Auf der oben genannten Website steht der Visual Studio 2019 Installer zum Download bereit. Hier ist bereits die gewünschte Version, in diesem Fall die Community-Edition, auszuwählen.

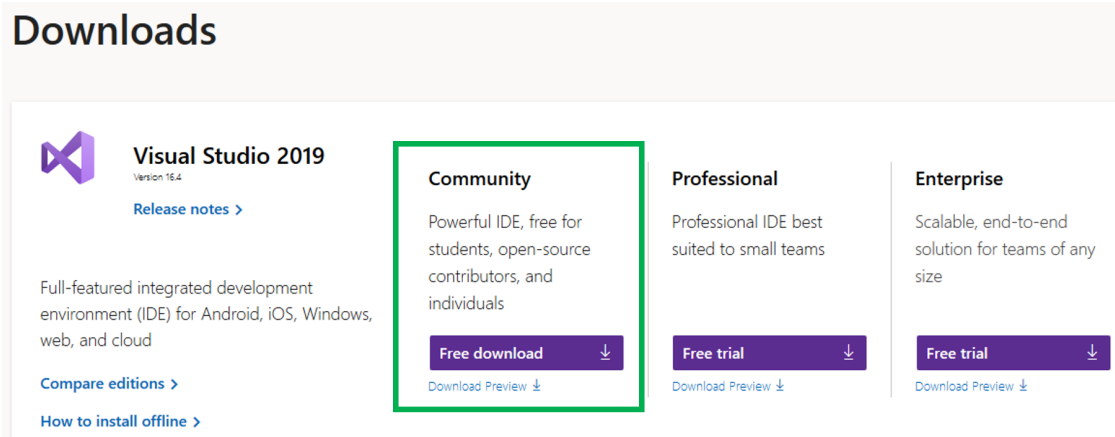


Abbildung 4.2: Auswahl der Visual-Studio-Edition

Nach dem Download und anschließender Installation können die gewünschten Programmteile ausgewählt werden. Für die Entwicklung von Xamarin-Applikationen ist das Mobile development with .NET™-Paket erforderlich.

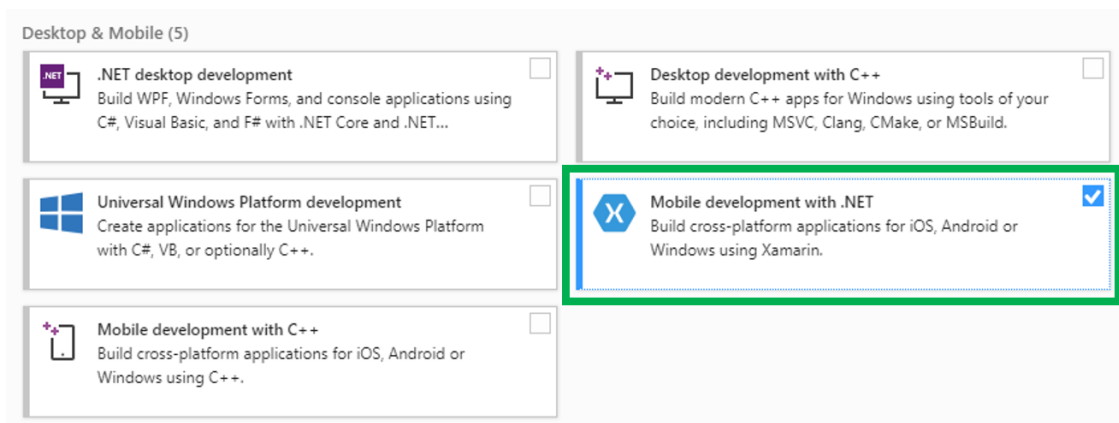


Abbildung 4.3: Auswahl des Xamarin-Rahmenwerks bei der Installation

Keine weiteren Einstellungen oder Auswahlen sind zwingend erforderlich, allerdings empfiehlt es sich noch zusätzlich das „.NET desktop development“-Paket zu installieren. Mit diesem kann man neue Technologien des .NET Frameworks in einer einfacheren Konsolenanwendung ausprobieren und kennenlernen.

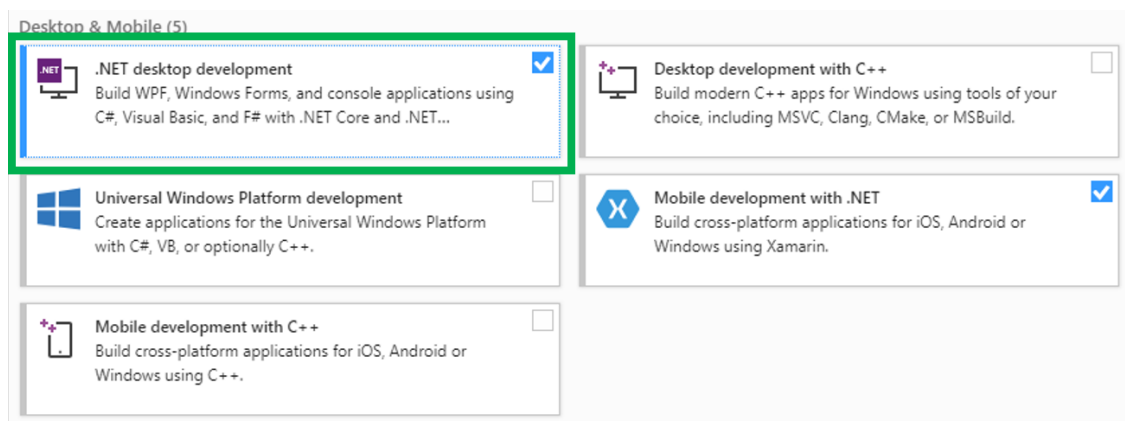


Abbildung 4.4: Zusatzpaket für Desktop-Entwicklung mit .NET

Nach Fertigstellung der Installation von Visual Studio 2019 erscheint dieses am Desktop und im Startmenü als neues Icon.

### 4.2.2 Installation Android SDK

Das Xamarin-Rahmenwerk benötigt zum Entwickeln von Android-Apps noch zusätzlich mehrere Teile des Android SDK der Firma Google. Mit diesem wird während dem Kompilieren das Programm in ein installierbares Android-Paket, kurz APK, umgewandelt. Damit dies richtig funktioniert, müssen alle Komponenten, auf die im Programm zugegriffen wird, über den Android-SDK-Manager installiert werden.

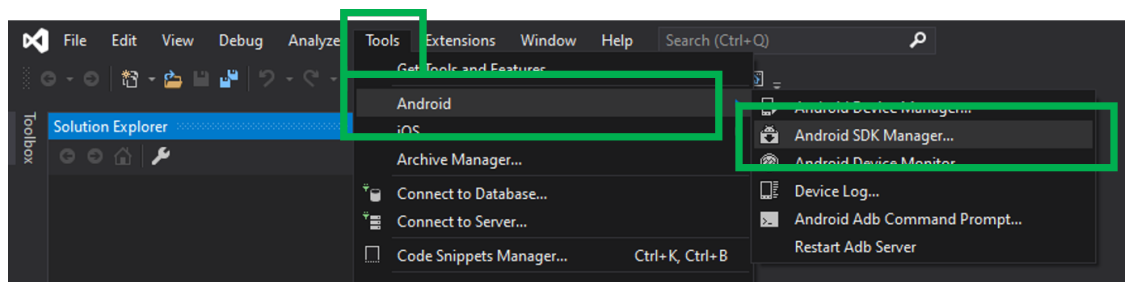


Abbildung 4.5: Android SDK Manager

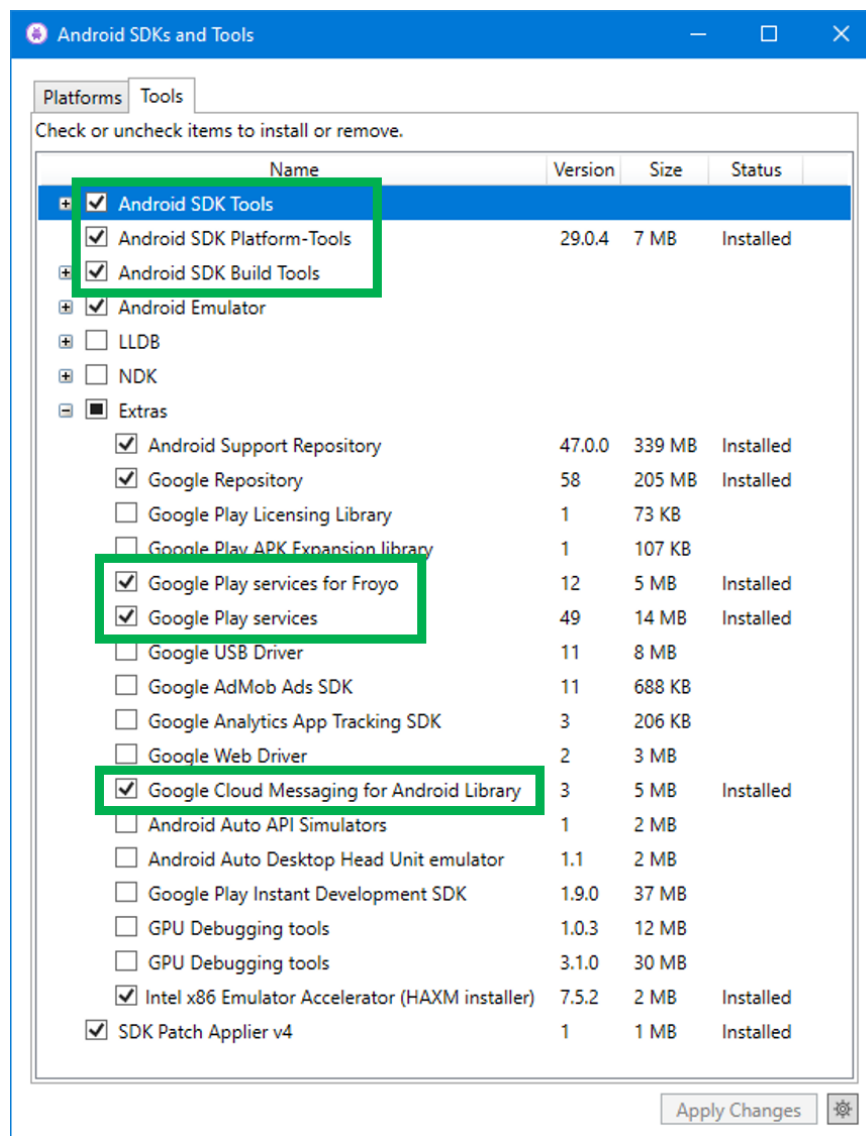


Abbildung 4.6: Notwendige Teile des Android SDK

Besonders wichtig sind hier die Pakete der Google Play Services und die Google Cloud Messaging Library, welche für das Empfangen von Push-Benachrichtung zwingend erforderlich sind. Diese Benachrichtigung wird in einem eigenen Kapitel näher erklärt. Die Android SDK Tools sind für jede Art von Android-Applikation mit Xamarin erforderlich. Bei Bedarf kann über dieses Menü auch der offizielle Android-Emulator installiert werden.

Nach dem Start von Visual Studio muss eine neue „Solution“ angelegt werden. Diese beinhaltet alle Programmteile, welche für die Xamarin-Entwicklung benötigt werden.

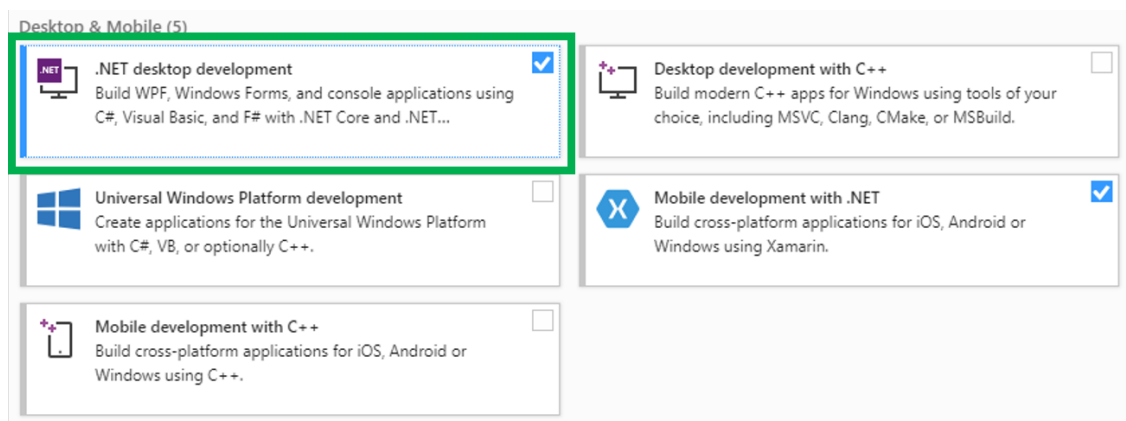


Abbildung 4.7: Anlegen einer neuen Solution

## 4.3 Gedanken

# 5 Programm-Dokumentation

## 5.1 Übersicht

Grundsätzlich besteht das Software-Projekt aus drei Teilen:

**PiBell** ist das portable Porjekt, auch .NET-Standard-Projekt genannt. Dieses beinhaltet den ganzen plattformunabhängigen Code, sowie die Definition der UI-Oberfläche mittels XAML-Dateien.

**PiBell.Android** beinhaltet allen Code, der plattformspezifisch für Android geschrieben wurde. Unter anderem ist hier enthalten die Android-Implementation des Mikrofon-Aufnahme-Services, sowie das Berechtigungs-Management.

**PiBell.iOS** beinhaltet wie PiBell.Android den plattformspezifischen Code für die iOS-Plattform. Aufgrund mangelnder Entwicklungswerkzeuge wurde dieser Teil nicht großartig behandelt.

Der immer wieder auftretende Name PiBell ist dabei eine freigeistliche Erfindung. Der Begriff setzt sich zusammen aus RaspberryPi und dem englischen Wort für Glocke oder Klingel (Bell).

## 5.2 Portable Project

## 5.3 PiBell.Android

## 5.4 PiBell.iOS

Abbildung 5.1: Aufbau der Solution























# 10 Testen

## **Teil III**

## **Server**





# **11 Grundlagen**

## **11.1 GStreamer**



## 12 Live555 Proxy



## **Teil IV**

## **Appendix**





# **A Verwendete Entwicklungswerkzeuge**



## **B Lastenheft**



## **C Zeitplanung**





## **D Kostenübersicht**



# **E Fertigungsdokumentation**

## **Abkürzungen**

**RPi** RaspberryPi