Innovative Mobile Applications

Übung 2

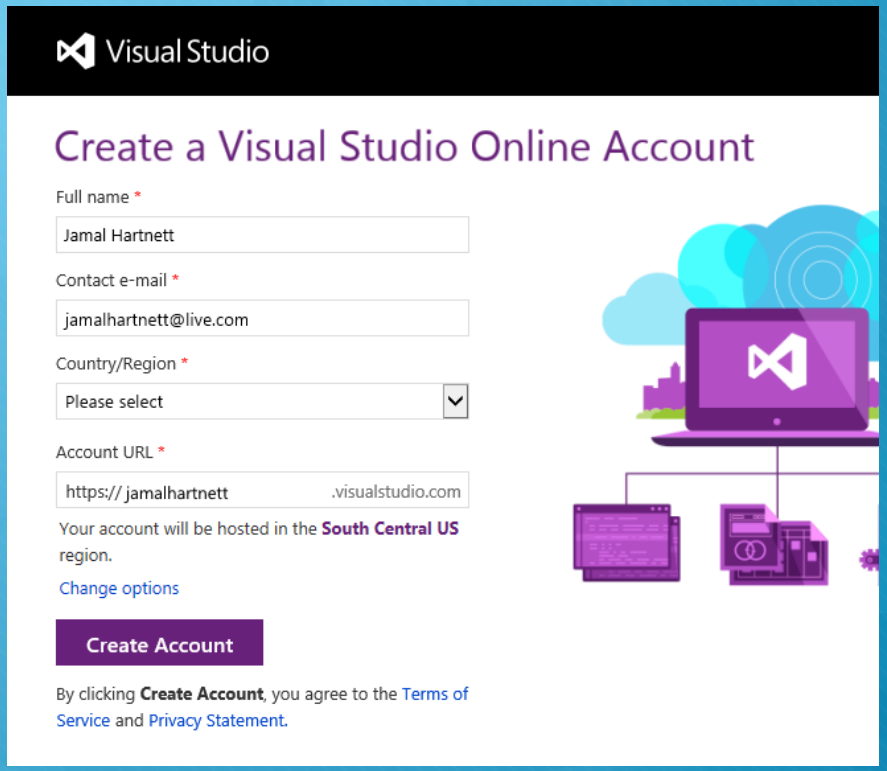
Agenda

- Organisation mit Visual Studio Online
- Projekt – Funktionalität und Code
- C# im Vergleich zu Java
- Native Funktionalität mittels Xamarin

Visual Studio Online

Setup:

- Anmeldung mit Live-ID
- Account-URL erstellen
- Team-Mitglieder zufügen



Visual Studio Online

Backlog:

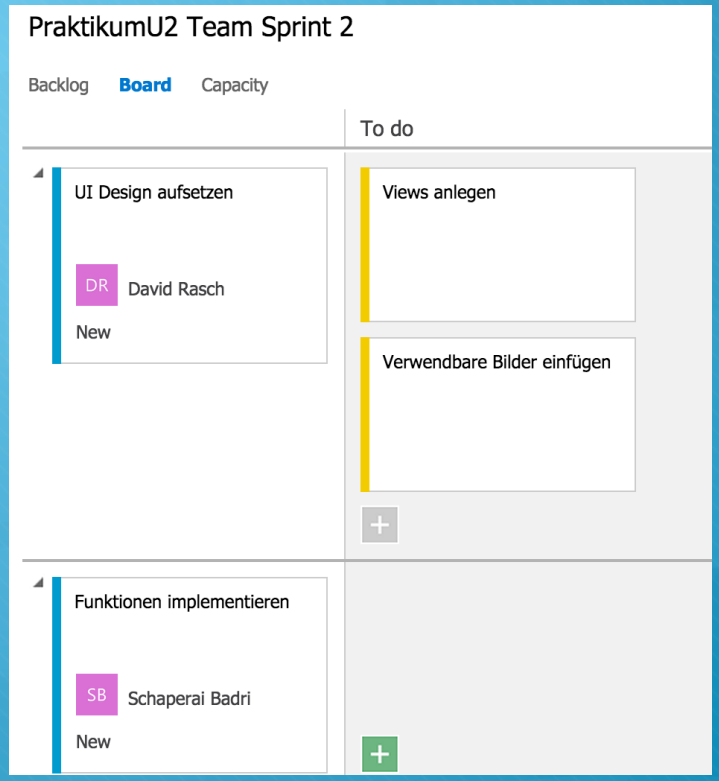
- Ort an dem eine Prioritätenliste der Anforderungen gehalten wird
- Repository aller Informationen

Order	Work Item Type	Title	State	Effort	Iteration Path
1	Product Backlo...	UI Design aufsetzen	New	6	PraktikumU2\Release 1\Sprint 2
2	Product Backlo...	Funktionen implementieren	New	6	PraktikumU2\Release 1\Sprint 2
3	Product Backlo...	Tests	New	1	PraktikumU2\Release 1\Sprint 3
4	Product Backlo...	Präsentationen anfertigen	New	1	PraktikumU2\Release 1\Sprint 3

Visual Studio Online

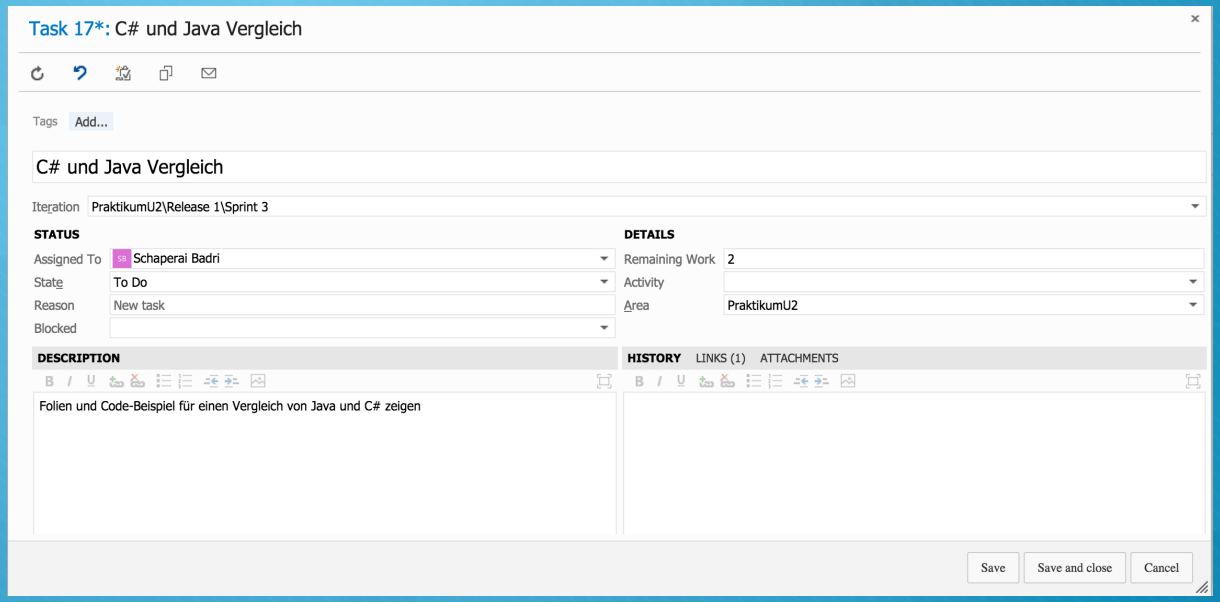
Scrum:

- Sprints mit gefilterten Anforderungen für die Dauer eines Arbeitsschrittes



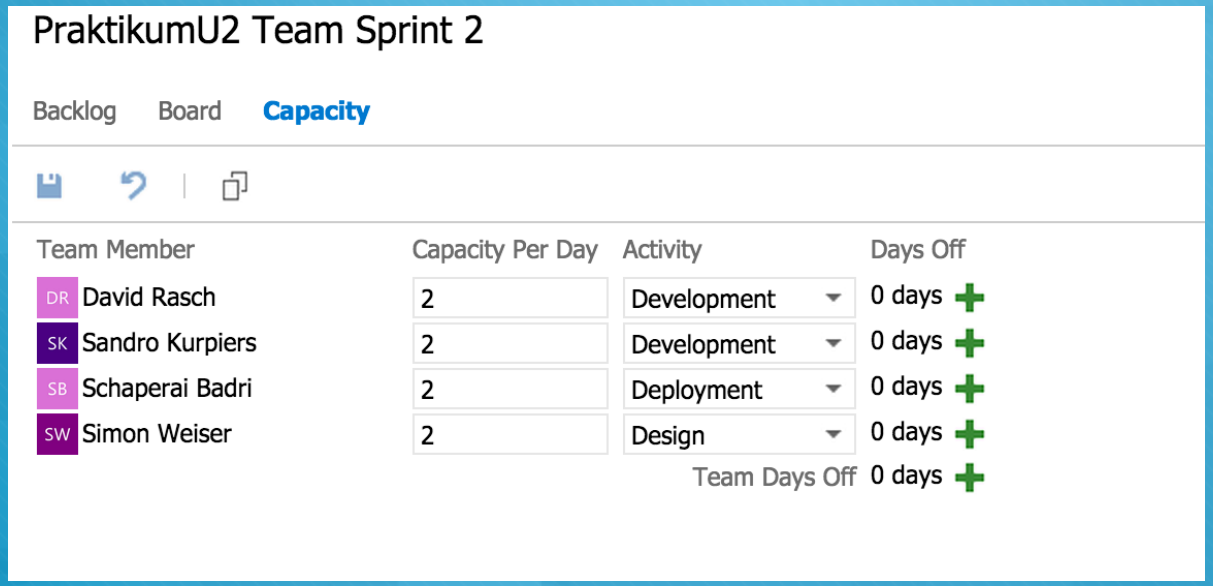
Visual Studio Online

To-Do auf Board anlegen:

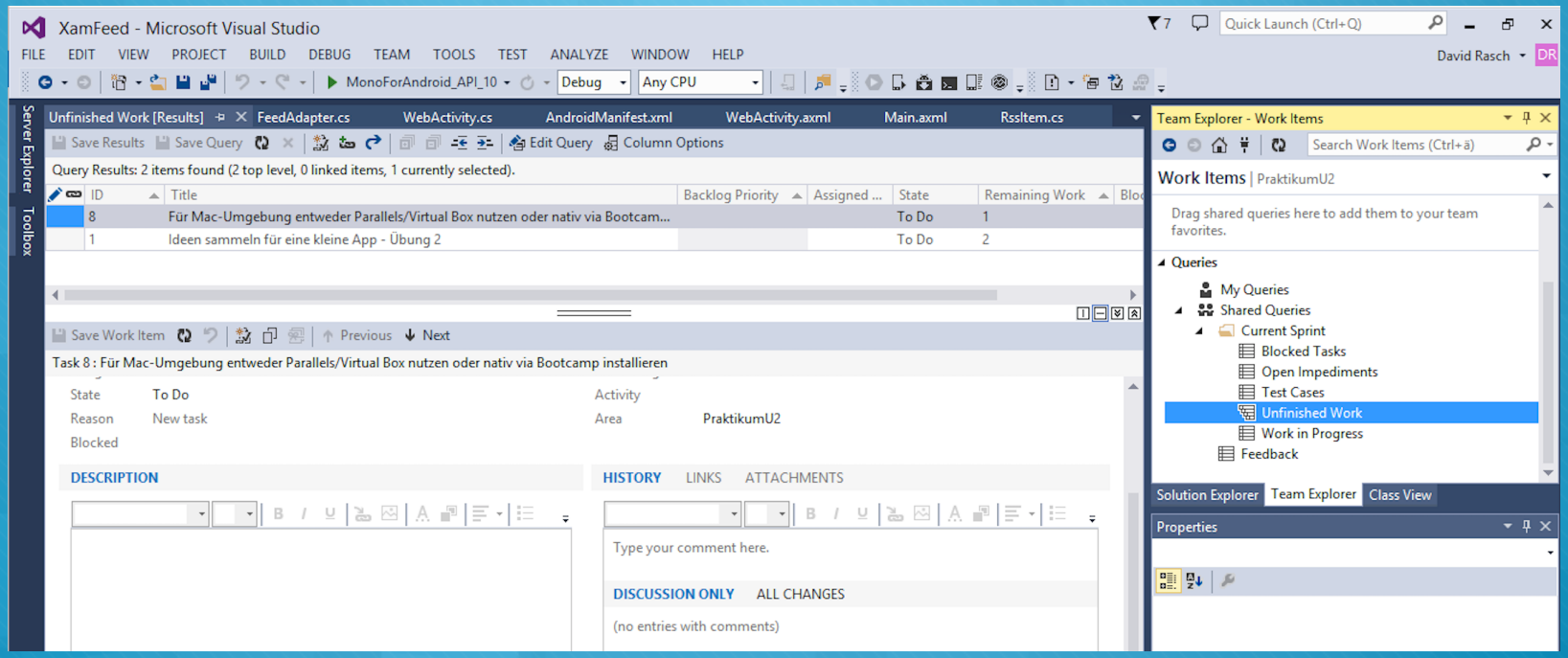


Visual Studio Online

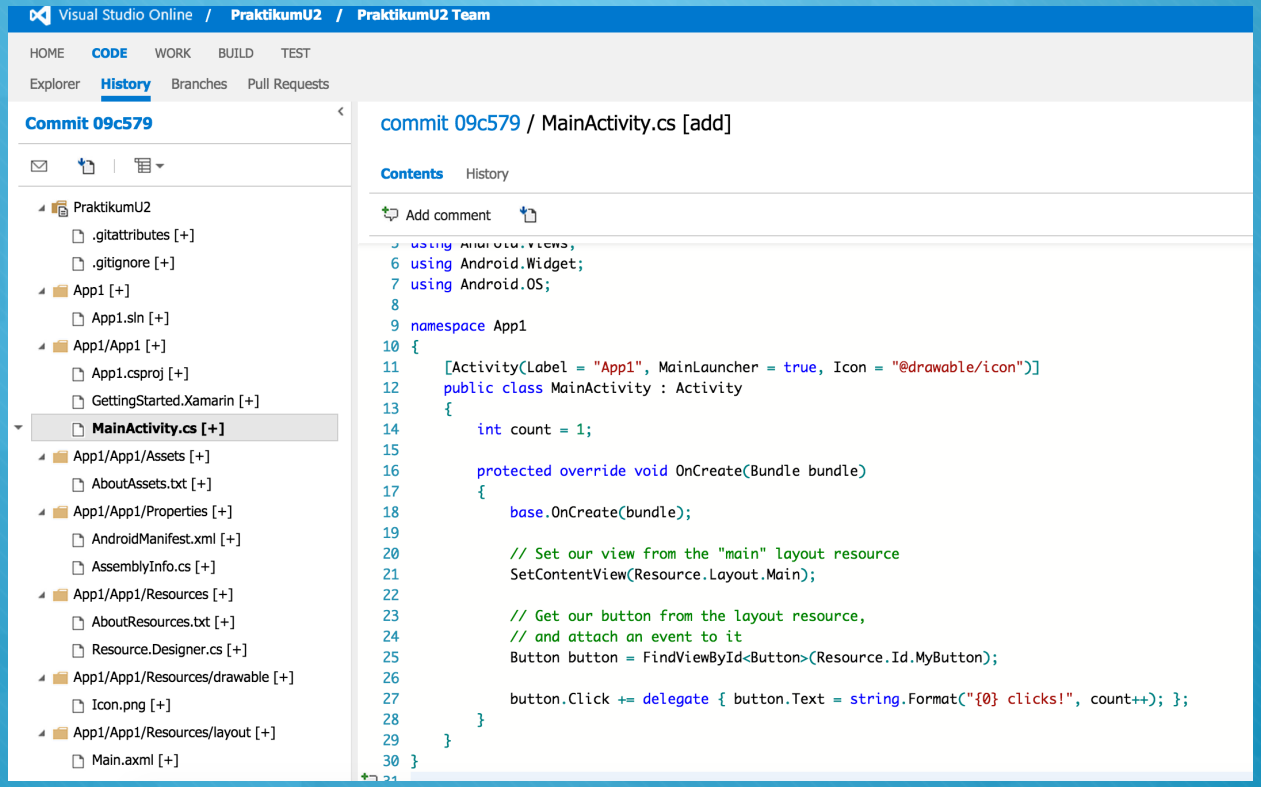
Kapazitäten zuweisen:



Visual Studio Einbindung

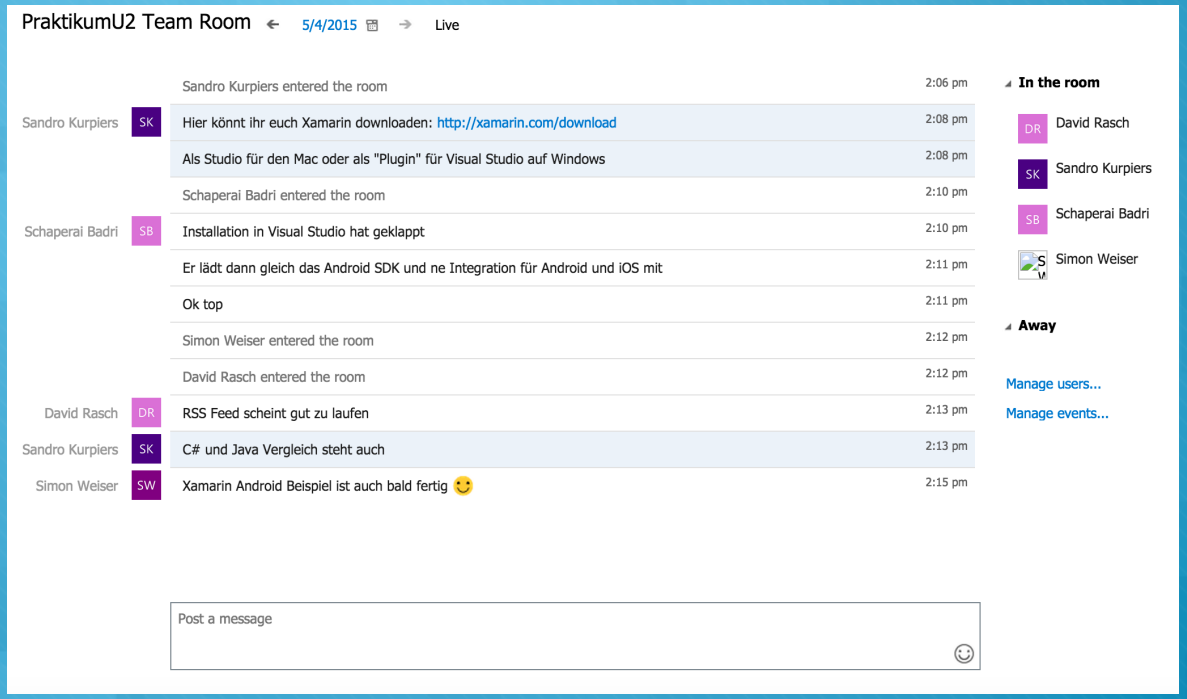


Visual Studio Online



Visual Studio Online

Chat-Room:



Code-Demo

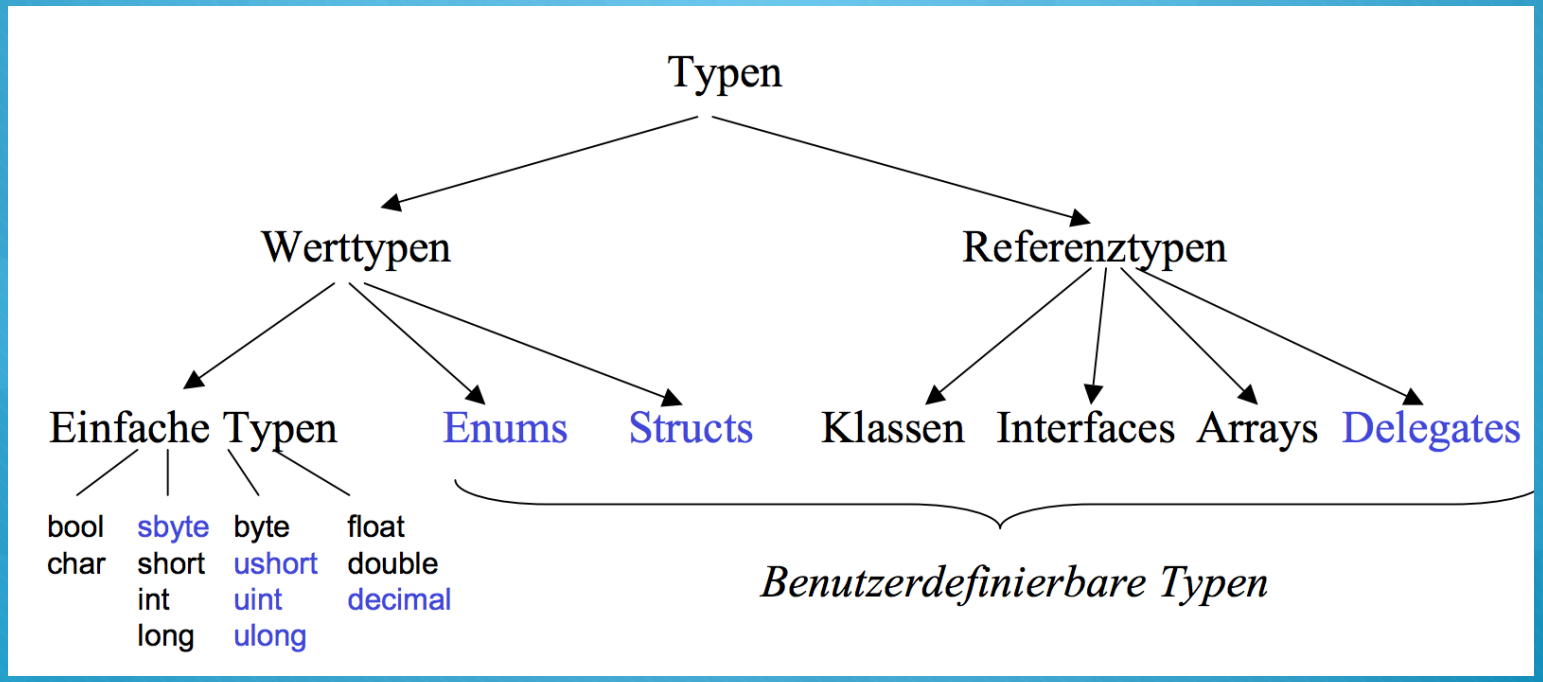
C# und Java im Vergleich

	C#	Java
Entwickler:	Microsoft	Sun Microsystems
Art:	objektorientiert	objektorientiert
Laufzeit-umgebung	.NET (CIL)	JRE (Bytecode)

C# und Java im Vergleich

	C#	Java
Typisierung:	stark, statisch, klassenbasiert	stark, statisch, klassenbasiert
Pointer	ja	nein
Versionierung	ja	nein

C# Typsysteme



Quelle: Hanspeter Mössenböck – Universität Potsdam

C# und Java Codebeispiel

Fibonacci Sequenz in Java:

```java
// Fibonacci

class Fibonacci implements Iterable<Integer> {

    public Iterator<Integer> iterator() {
        return new Iterator<Integer>() {
            private int a = 0;
            private int b = 1;

            @Override
            public boolean hasNext() {
                return true;
            }

            @Override
            public Integer next() {
                int tmp = a;
                a = b;
                b = a + tmp;
                return tmp;
            }

            @Override
            public void remove() {
                throw new UnsupportedOperationException(
                        "Keine Fibonacci Sequenz");
            }
        };
    }
}
```

C# und Java Codebeispiel

Fibonacci Sequenz in C# :

```
1  // Fibonacci
2  public static IEnumerable<int> Fibonacci() {
3      int a = 0;
4      int b = 1;
5
6      while (true) {
7          yield return a;
8          a += b;
9          yield return b;
10         b += a;
11     }
12 }
13 }
14
```

Zusammenfassung

Pro C#:

- Mächtiger (Structs, Referenzparameter)
- Bequemer (Indexer, foreach)
- Flexibler (erlaubt Systemprogrammierung)
- Besser unter Windows

Zusammenfassung

Pro Java:

- Kleiner und einfacher
- Größere Verbreitung
- Strikter (kein Unsafe Code)
- Portabler

IDE für C#

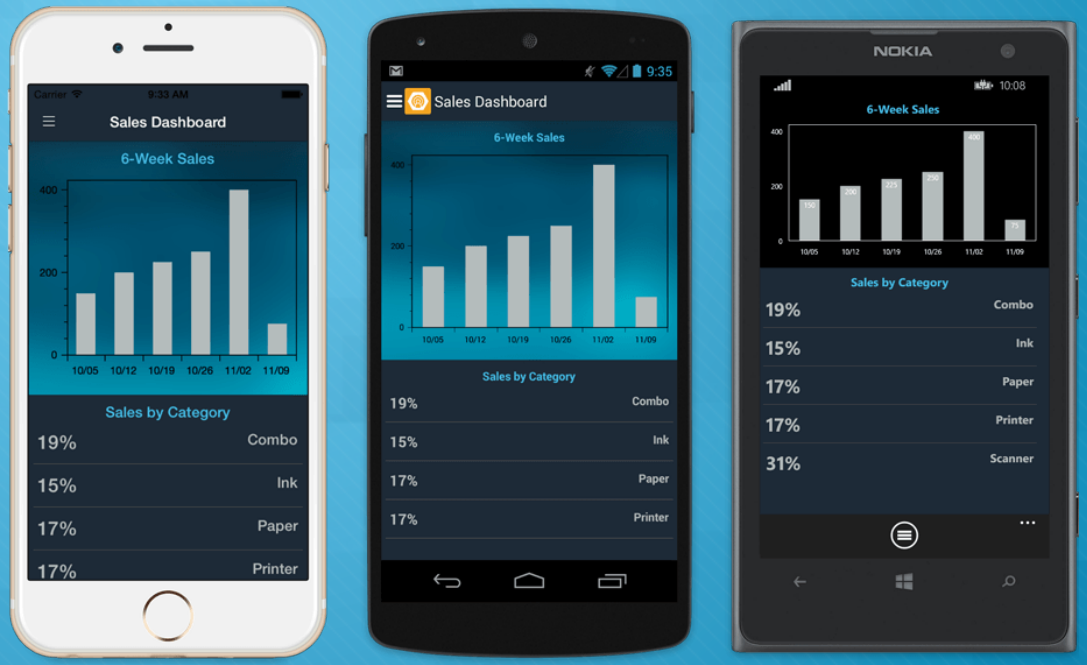

Android
Studio
Powered by the IntelliJ Platform


XCODE


Visual Studio

Xamarin

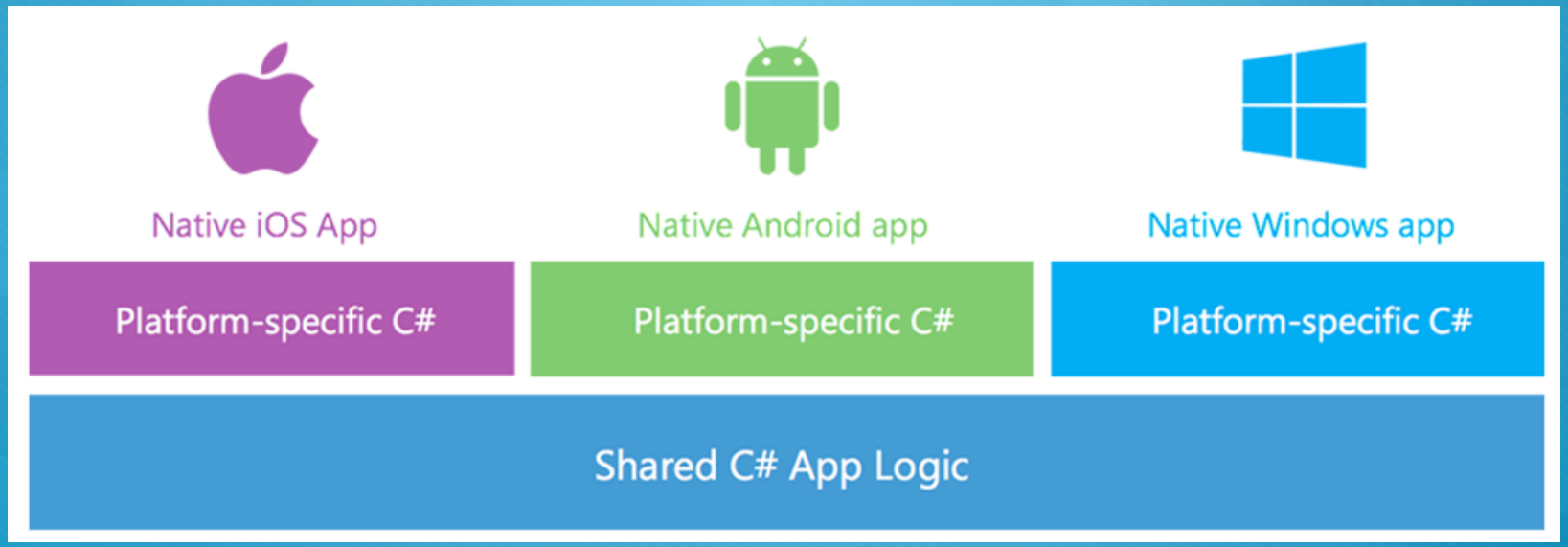
Native Programmierung
für Android, iOS, Windows
und Mac mit C#



Xamarin Studio

	
Standalone IDE für mobile App-Entwicklung (erhältlich für Windows und Mac)	Mächtige IDE für zahlreiche Hochsprachen (nur Windows)
Enthält Xamarin.Android, Xamarin.iOS und Xamarin.Mac (nicht Windows Phone)	In Kombination mit Xamarin for VS möglich Apps für Android, iOS und Windows Phone in C# zu schreiben

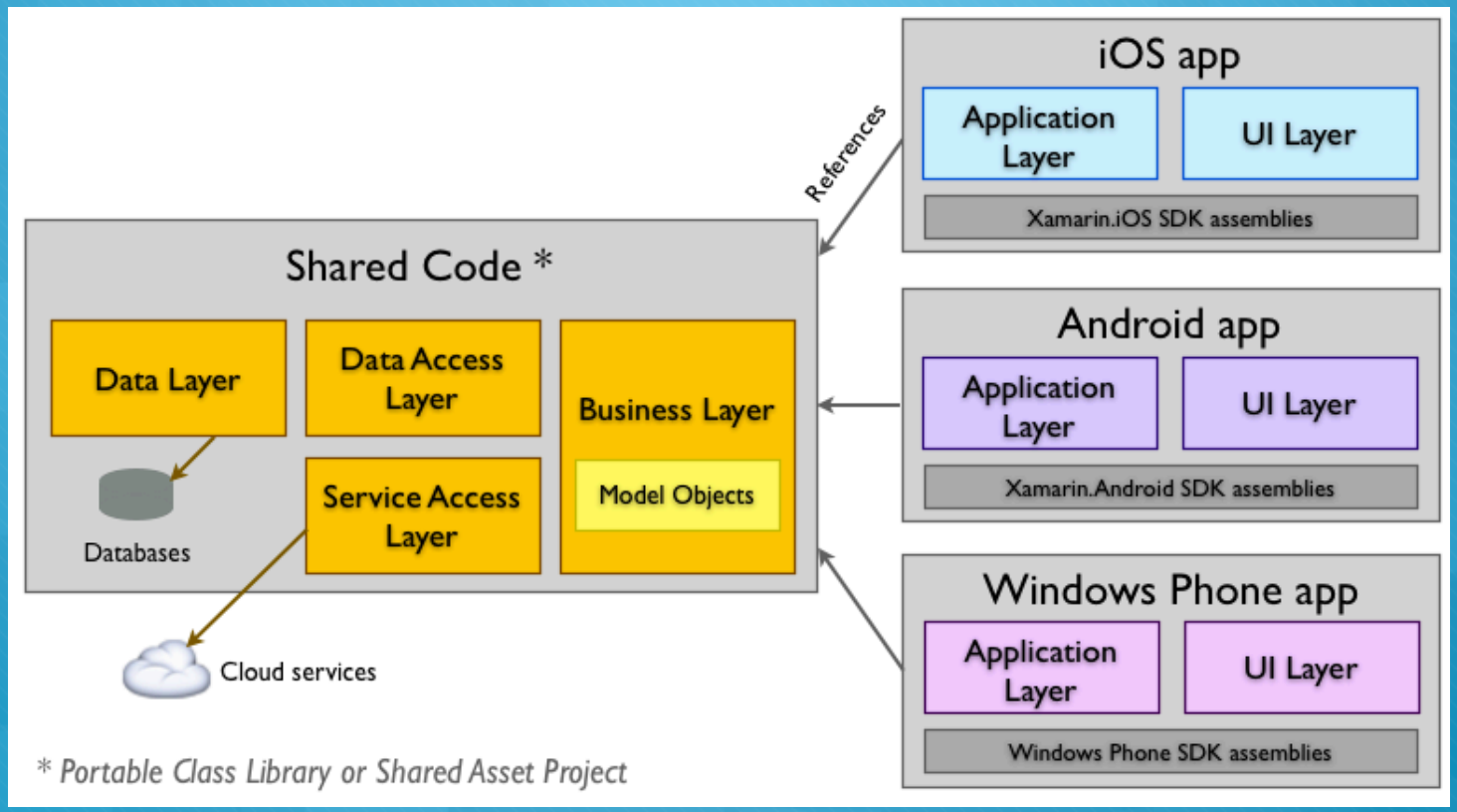
Xamarin


Native iOS App
Native Android app
Native Windows app
Platform-specific C#
Platform-specific C#
Platform-specific C#
Shared C# App Logic

Xamarin.Forms

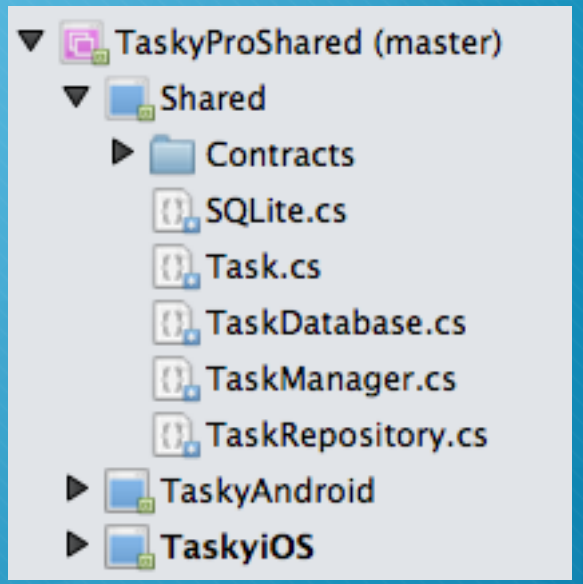
- Applikation (inkl. GUI) muss nur 1x entwickelt werden
- Xamarin.Forms.Button
  - → UIButton (iOS)
  - → AndroidButton (Android)
  - → WindowsPhoneButton (Windows)
- Unterschiede der Plattformen per Default eingebaut

Shared Code


iOS app
Application Layer
UI Layer
Xamarin.iOS SDK assemblies
References
Shared Code *
Data Layer
Data Access Layer
Business Layer
Model Objects
Service Access Layer
Databases
Cloud services
Android app
Application Layer
UI Layer
Xamarin.Android SDK assemblies
Windows Phone app
Application Layer
UI Layer
Windows Phone SDK assemblies
* Portable Class Library or Shared Asset Project

Shared Project

- **Shared** – Shared Project enthält Code den alle Projekte gemeinsam haben
- **TaskyAndroid** – Xamarin.Android application project
- **TaskyiOS** – Xamarin.iOS application project



Compiler Directives

- Sollten Unterscheidungen für eine spezielle Plattform notwendig sein, wird häufig eine Kompiler-Direktive verwendet

```
using System;
using System.Collections.Generic;
using System.Text;

#if __IOS__
using MonoTouch.Accounts;
#elif __ANDROID__
using Android.Accounts;
#endif

namespace MyApps.Shared
{
    public class MyFirstClass
    {
```

Danke für die Aufmerksamkeit

Praktikum Innovative Mobile Applications, Übung 2
Schaperai Badri, Sandro Kurpiers, David Rasch, Simon Weiser