Microsoft

Visual Studio Microsoft Azure

前進的力量

創新應用開發大會

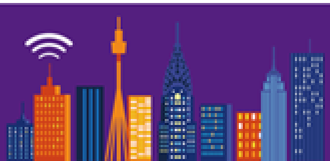
Agenda

- Architecting Mobile App
- Apache Cordova
- Xamarin
- Cordova vs. Xamarin

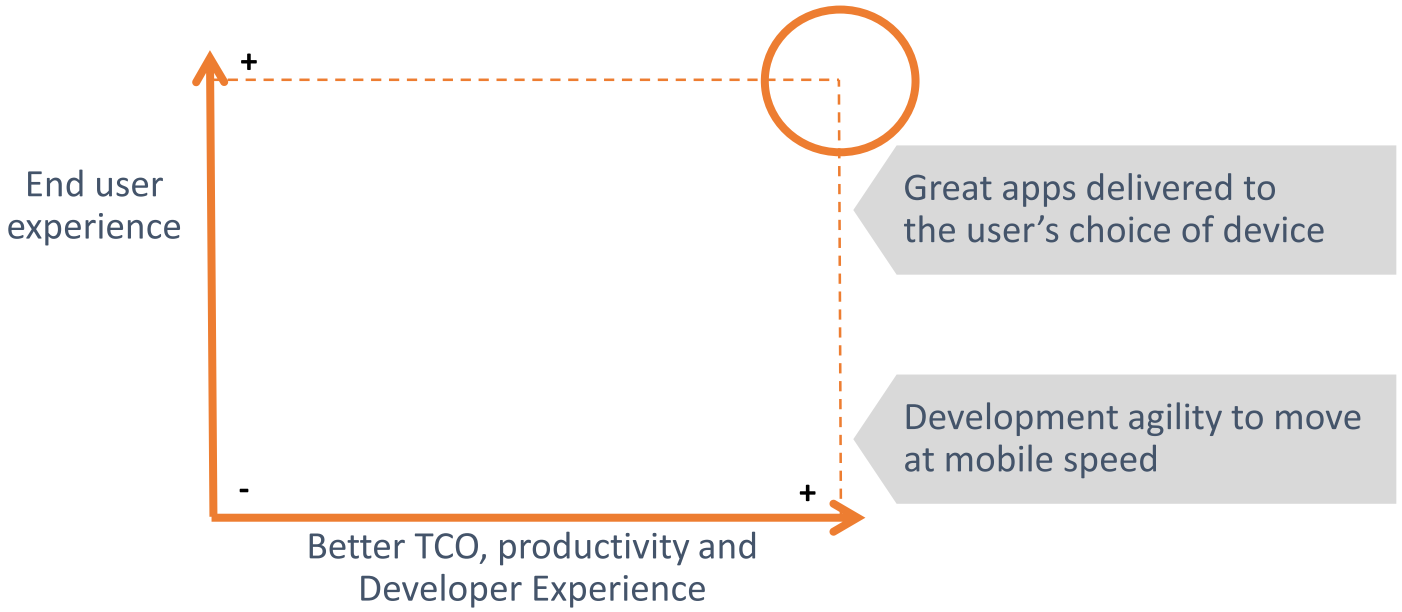
Mobile represents the single largest technology innovation of all time

Enterprises are going mobile on multiple platforms

By 2016, 70% of the mobile workforce will
have a smartphone,
and 90% of enterprises will have two or more
platforms to support



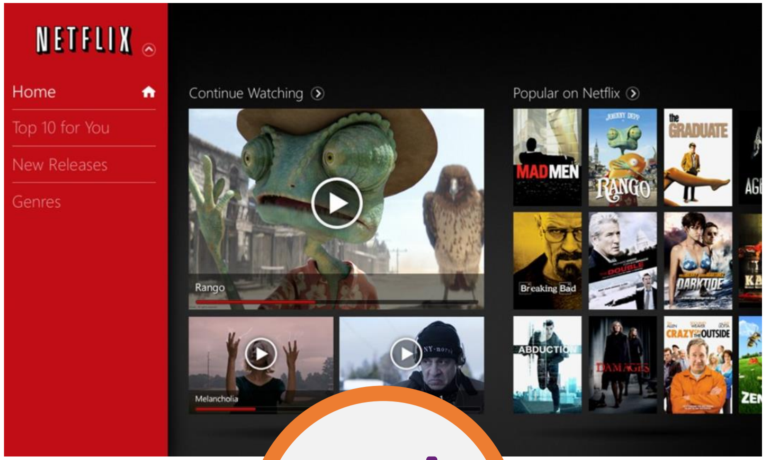
Sweet spot for the successful mobile enterprise

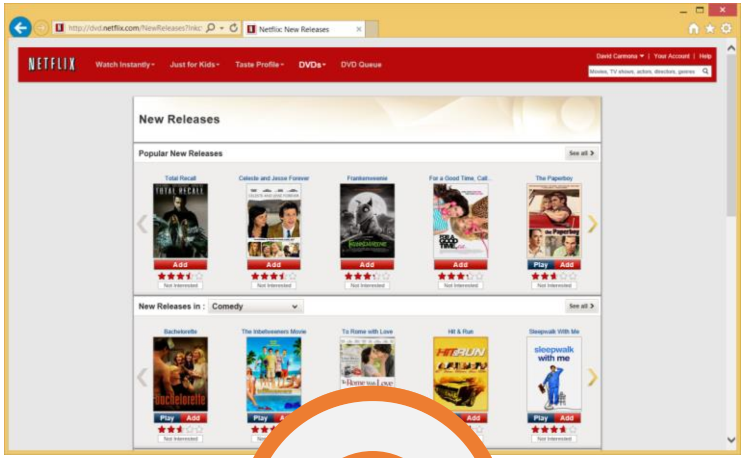


Great apps delivered to the user's choice of device

Development agility to move at mobile speed

Client development trends

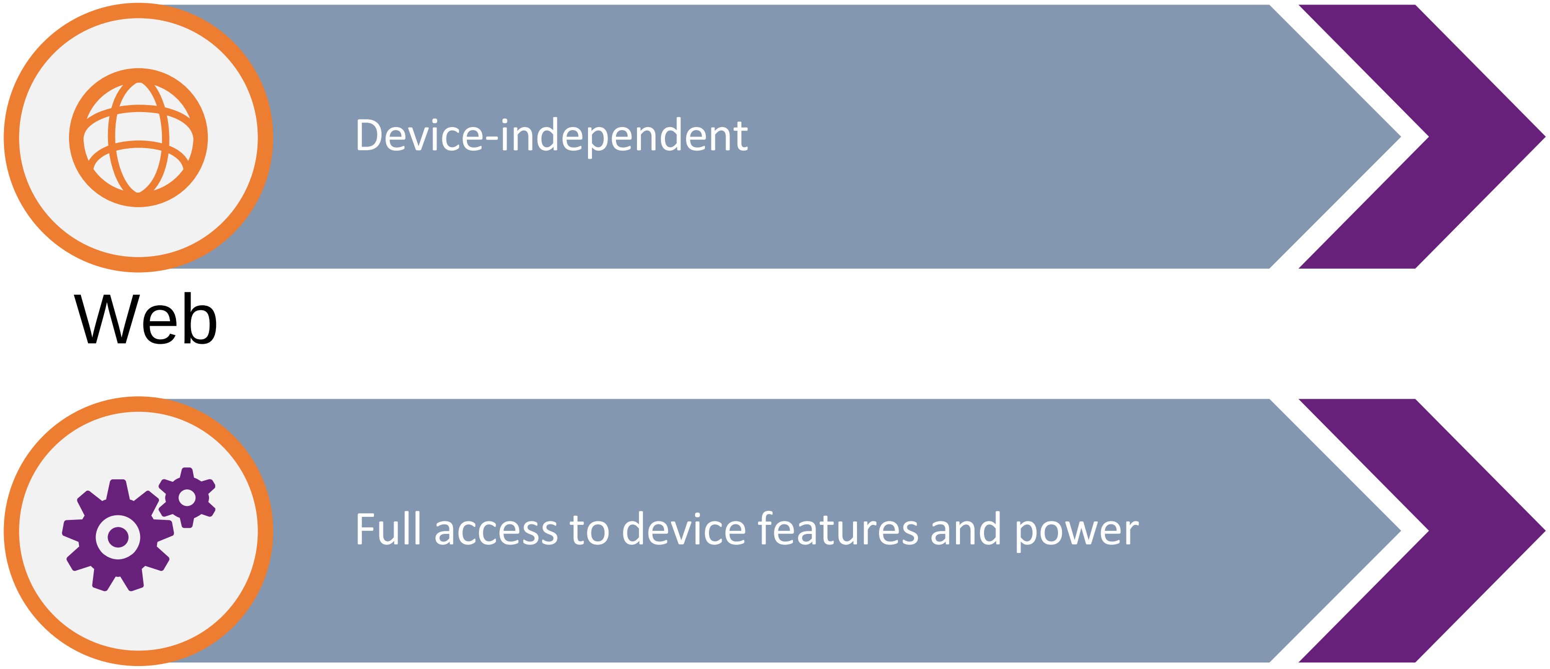




Native

Web

Client development trends



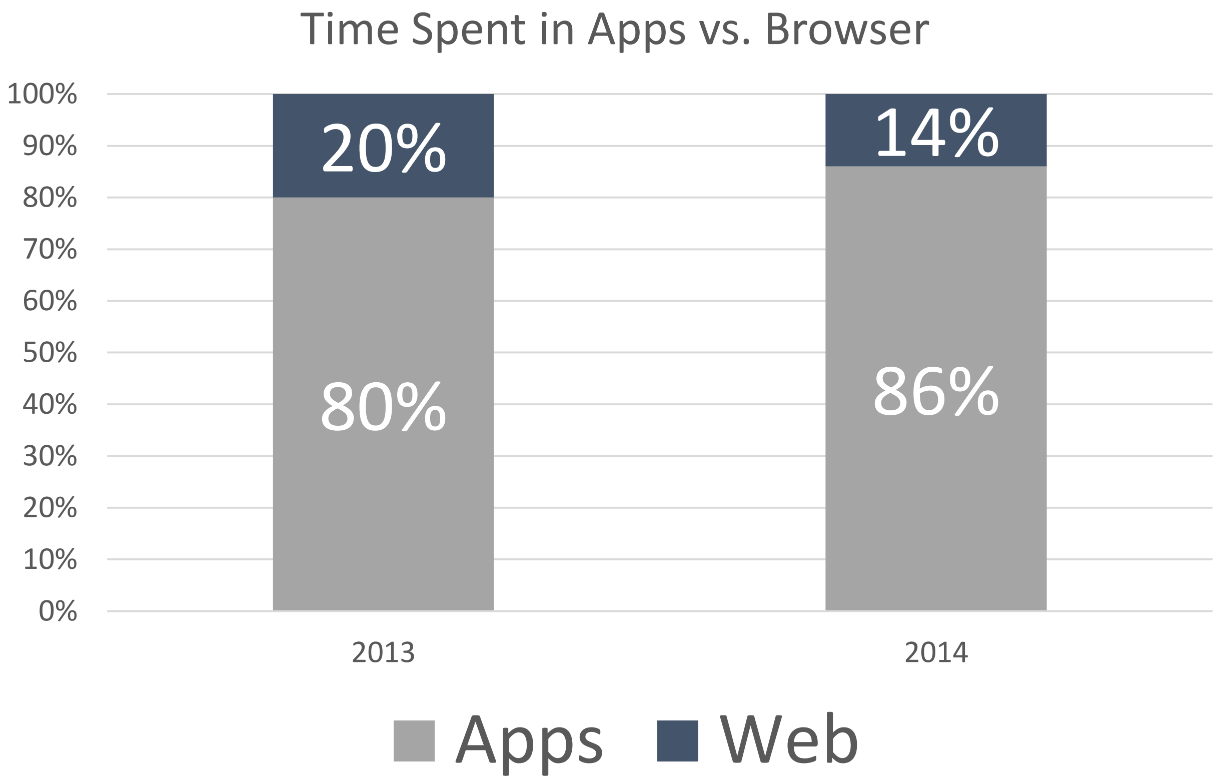
Device-independent

Web

Full access to device features and power

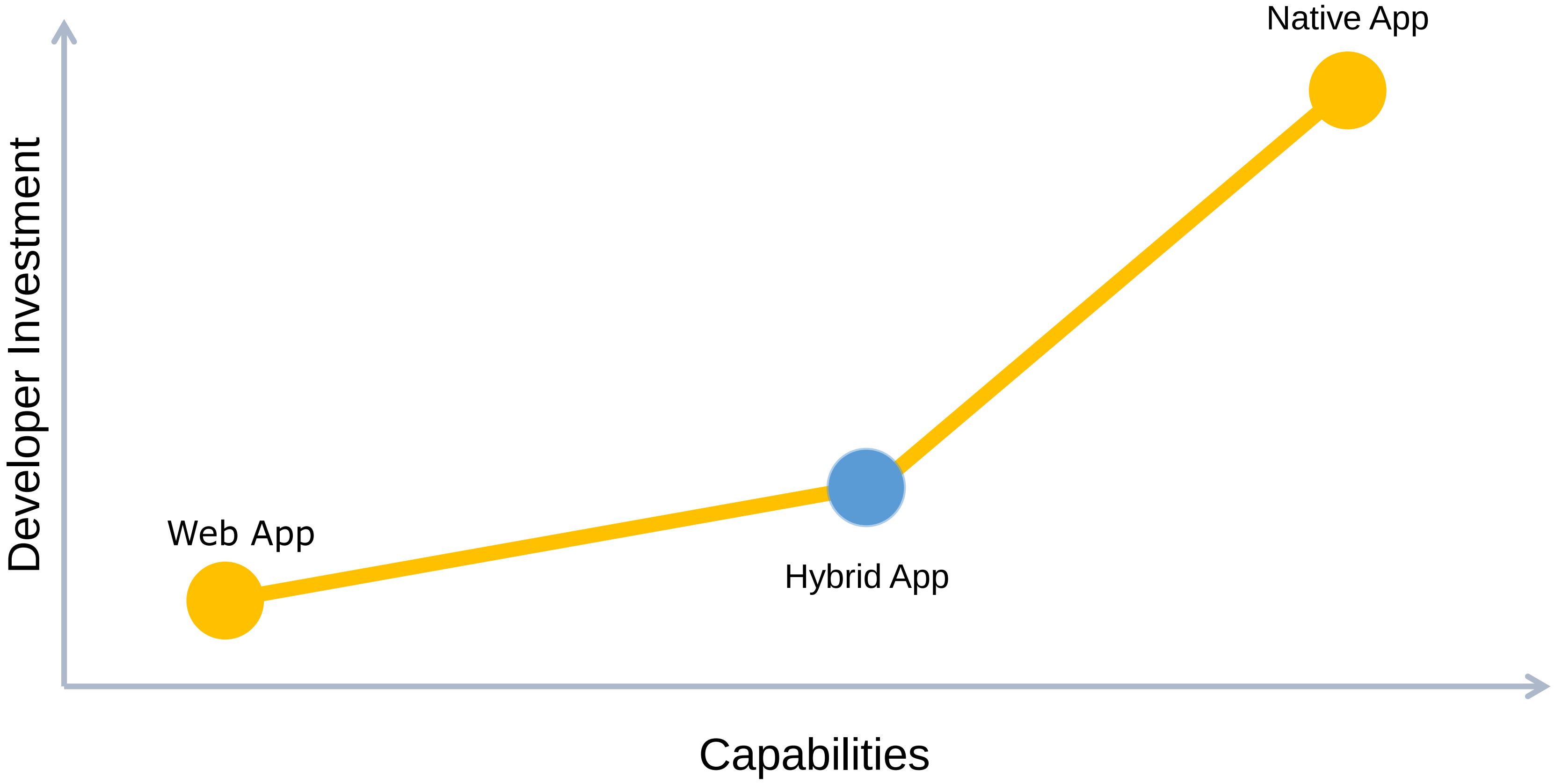
Native

Apps dominate the mobile web



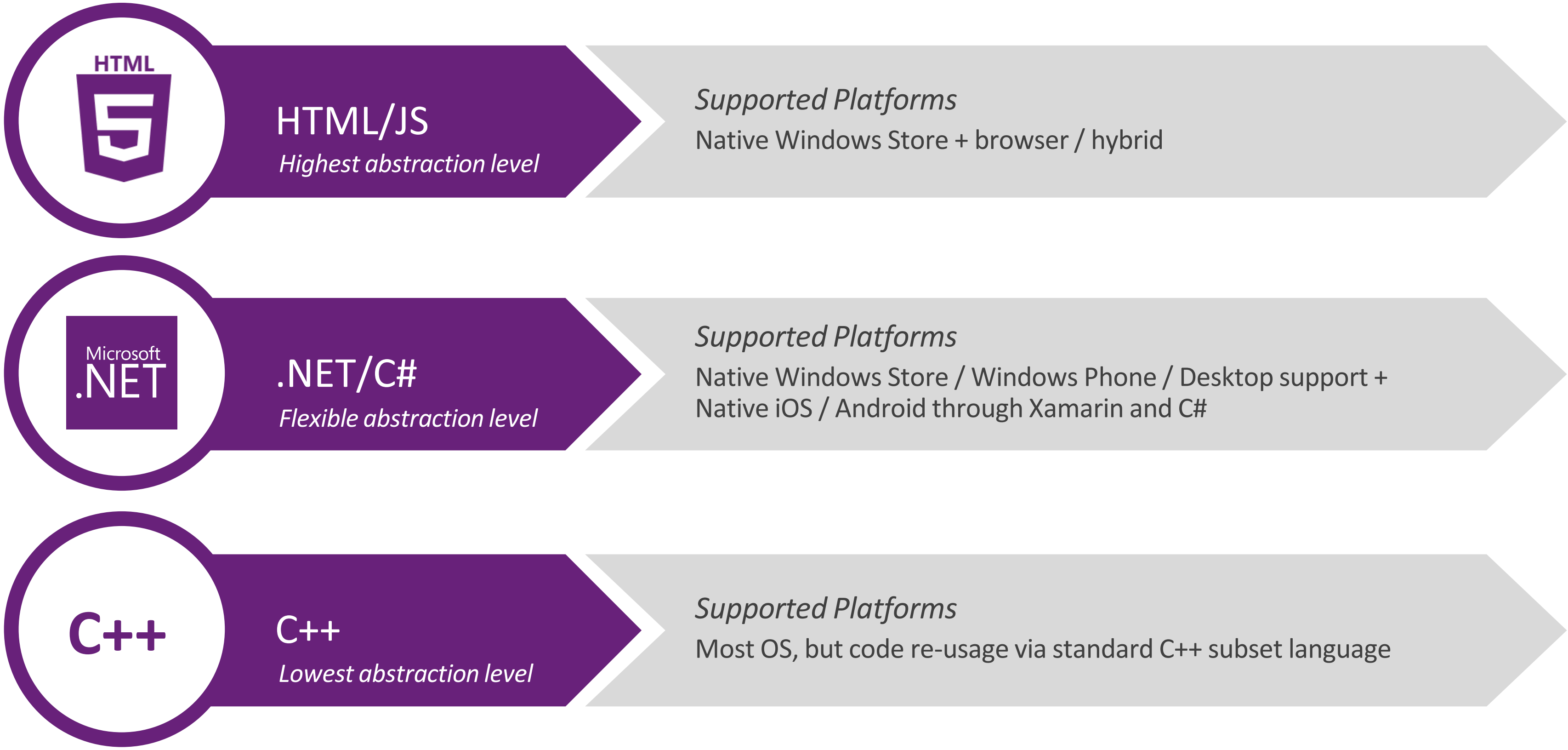
Source:
Flurry Analytics

Low investment for more capabilities



Native Client Apps

Microsoft Technology choices



HTML

HTML/JS

Highest abstraction level

Supported Platforms

Native Windows Store + browser / hybrid

Microsoft .NET

.NET/C#

Flexible abstraction level

Supported Platforms

Native Windows Store / Windows Phone / Desktop support +
Native iOS / Android through Xamarin and C#

C++

C++

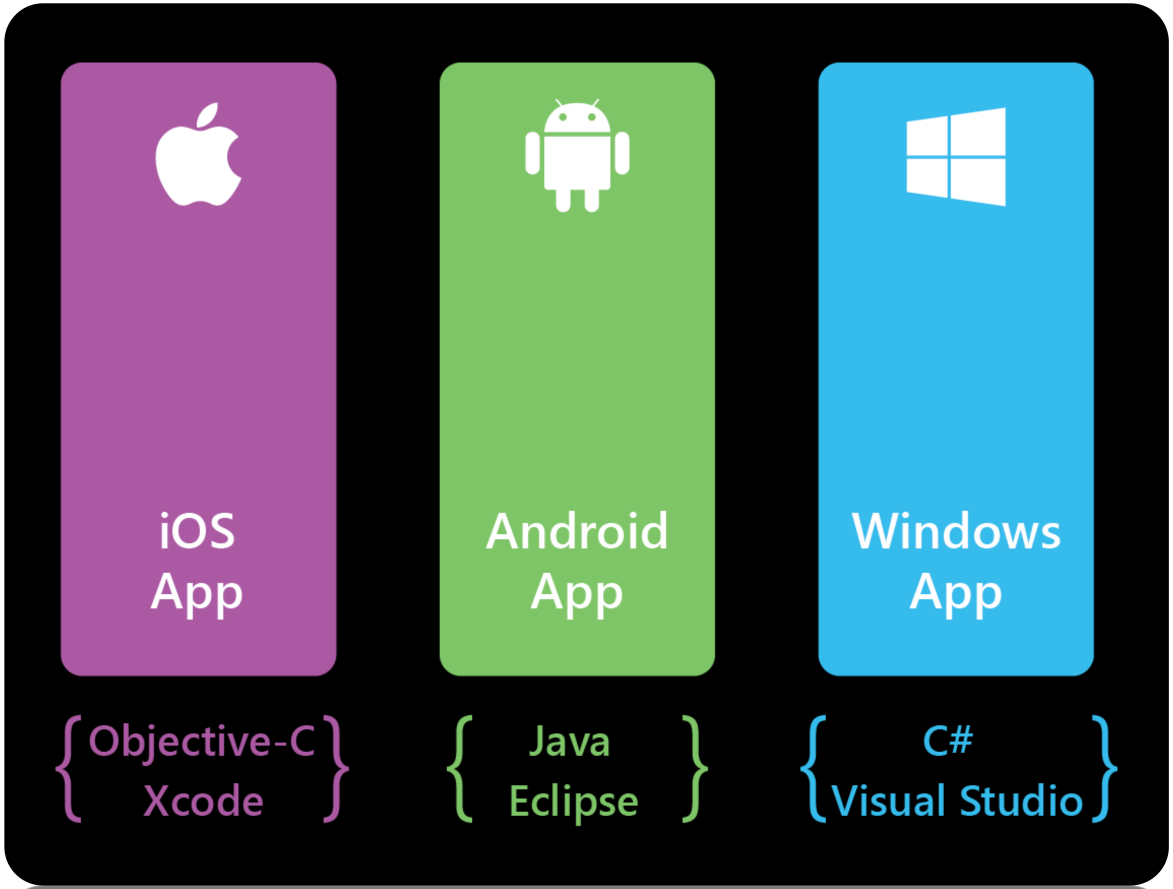
Lowest abstraction level

Supported Platforms

Most OS, but code re-usage via standard C++ subset language

Silo Approach:

Build the Same Apps Multiple Times



iOS App

{ Objective-C Xcode }

Android App

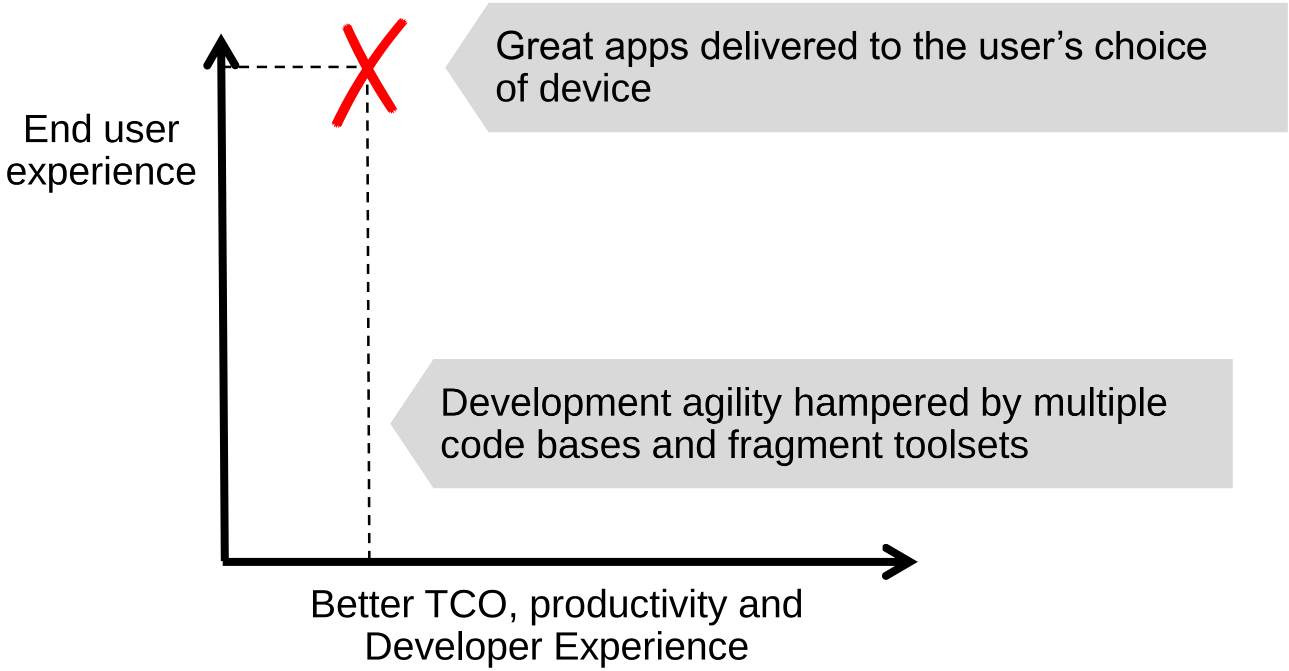
{ Java Eclipse }

Windows App

{ C# Visual Studio }

The Silo approach: Build native apps multiple times

Multiple teams and multiple code bases are expensive and slow



Great apps delivered to the user's choice of device

End user experience

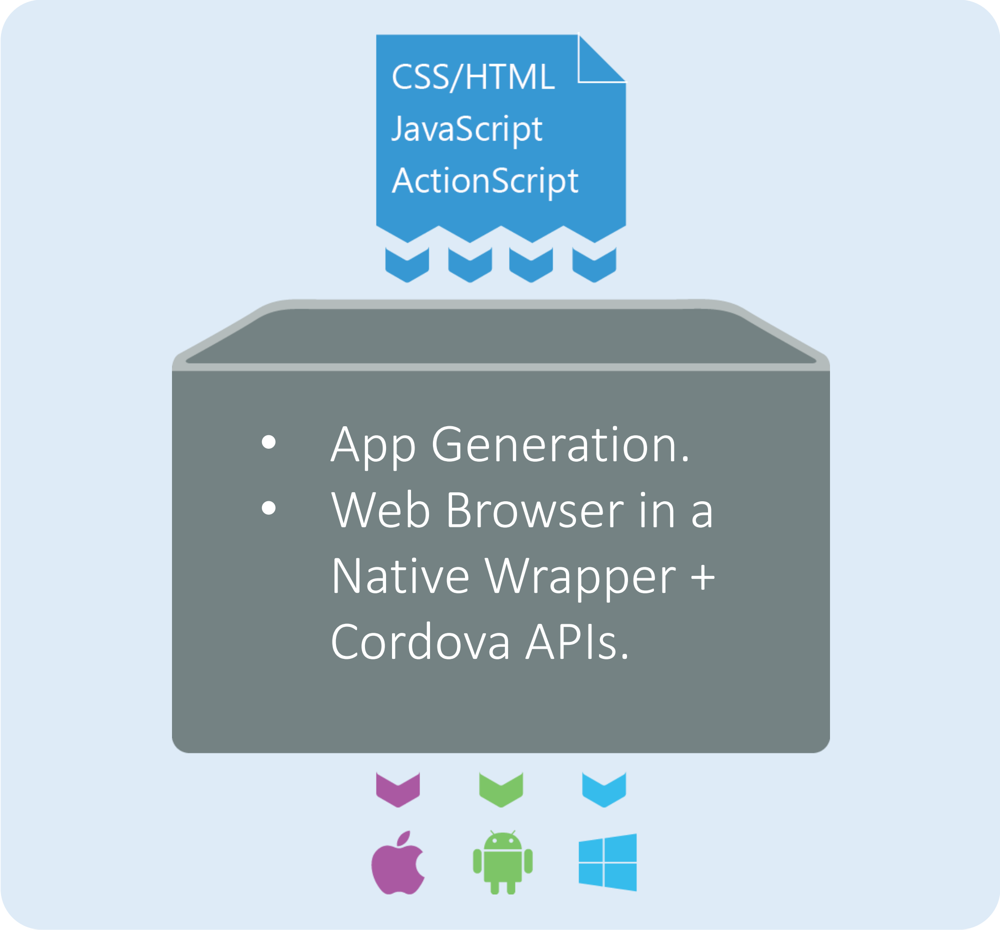
Development agility hampered by multiple code bases and fragment toolsets

Better TCO, productivity and Developer Experience

Write Once,
Run Anywhere
Approach

Lowest
Common
Denominator

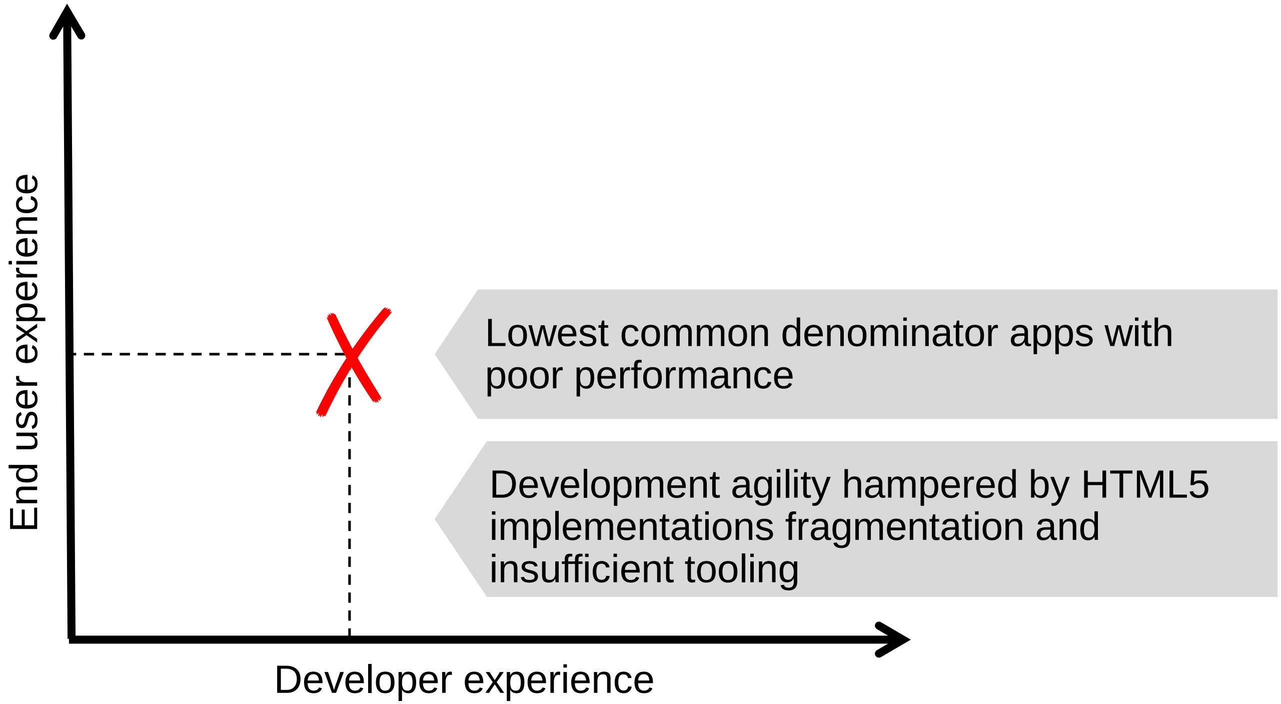
Browser
Fragmentation



- App Generation.
- Web Browser in a Native Wrapper + Cordova APIs.

The write-once-run-anywhere approach

HTML Hybrid scenarios (Semi-native apps) like **PhoneGap (i.e. Cordova)**



Lowest common denominator apps with poor performance

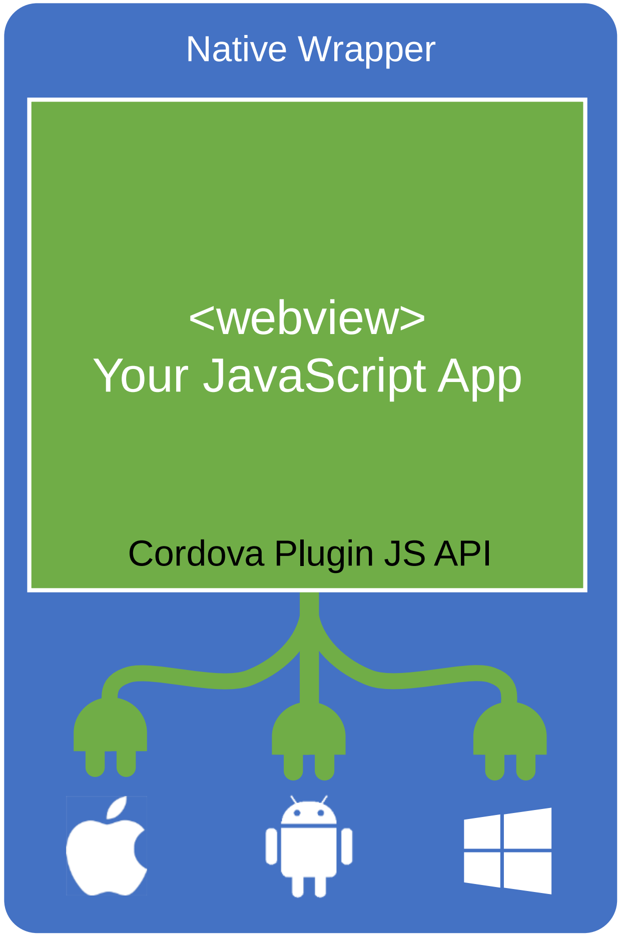
Development agility hampered by HTML5 implementations fragmentation and insufficient tooling

Benefits of cross platform development

- Improved reuse of code.
- Less duplicated work.
- Easier to maintain code base.
- Better consistency in implementations.
- Leveraging of existing skills on different platforms.
- Lower development cost.
- Quicker to market for multiple platforms.

What is Apache Cordova?

- Open-source framework
- Hosted webview
- Single, shared codebase deployed to all targets
- Plugins provide a common JavaScript API to access device capabilities



Native Wrapper

<webview>
Your JavaScript App

Cordova Plugin JS API

Cordova @ MS Open Tech

- MS Open Tech devs contribute to Cordova
  - We released the Visual Studio Tools for Apache Cordova
  - Included in Cordova 4.0.0: npm install –g cordova
- Updated the ‘windows’ platform
  - Support now for Universal Apps
    - Windows Phone 8.1 & Windows 8.1
    - Native JS/HTML5 development
    - does NOT use Webview
    - Local Security context vs. Web security context
      - JavaScript Dynamic Content Shim for Windows apps
      - https://github.com/MsopenTech/winstore-jscompat
  - Windows Phone 8
    - Webview control

Visual Studio Tools for Apache Cordova

- Extension to Visual Studio 2013 and 2015
- Optimized for local build
  - All parts of the tool chain installed
  - Generally updated within a month of new releases
- Integrated Dev Environment
  - Code editing, plugin acquisition build, debug, package & publish
- Debug against any* target

Supported Platforms

Android

- Uses Android SDK installed locally
- Android emulator, Ripple, real device

iOS

- Uses agent running on a MAC with Xcode tools installed
- Ripple

Windows Store

- Device Visual Studio is running on
- Simulator
- Remote debugging

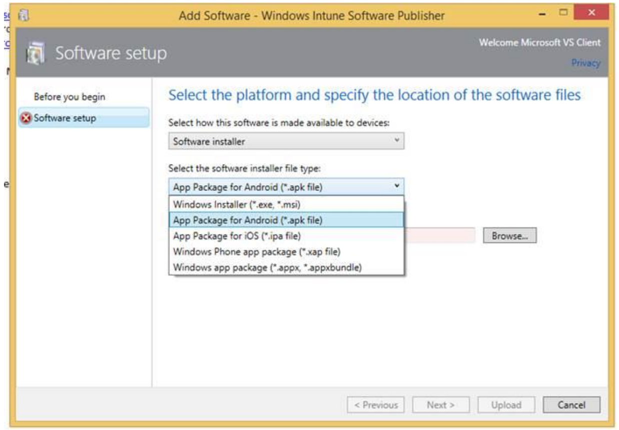
Windows Phone

- Emulator (Hyper-V based)
- Real device

Publish & Manage Your App: Pick your deployment!

Deploy to your enterprise via Windows Intune

Manage distribution & version updates



Add Software - Windows Intune Software Publisher

Software setup

Welcome Microsoft VS Client

Privacy

- Before you begin
- Software setup

Select the platform and specify the location of the software files

Select how this software is made available to devices:

Software installer

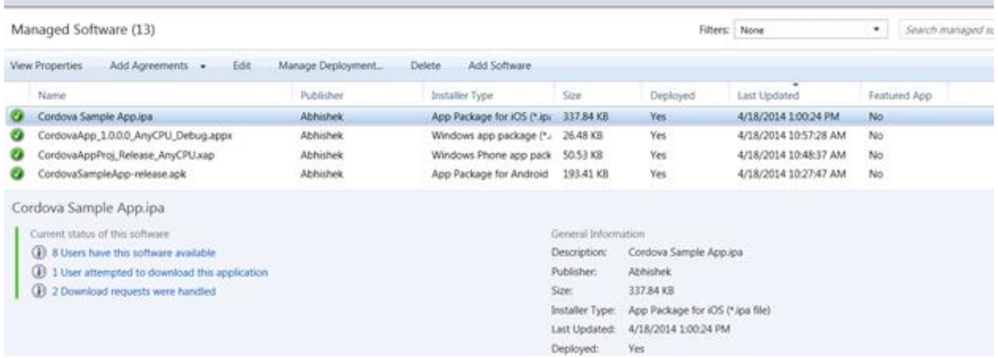
Select the software installer file type:

App Package for Android (*.apk file)

- Windows Installer (*.exe, *.msi)
- App Package for Android (*.apk file)
- App Package for iOS (*.ipa file)
- Windows Phone app package (*.xap file)
- Windows app package (*.appx, *.appxbundle)

Deploy to the public via Windows Store, Google Play or the Apple Store

Reach the widest audience possible



Managed Software (13)

Filters: None

	Name	Publisher	Installer Type	Size	Deployed	Last Updated	Featured App
✓	Cordova Sample App.ipa	Abhishek	App Package for iOS (*.ip	337.84 KB	Yes	4/18/2014 1:00:24 PM	No
✓	CordovaApp_1.0.0.0_AnyCPU_Debug.appx	Abhishek	Windows app package (*.	26.48 KB	Yes	4/18/2014 10:57:28 AM	No
✓	CordovaAppProj_Release_AnyCPU.xap	Abhishek	Windows Phone app pack	50.53 KB	Yes	4/18/2014 10:48:37 AM	No
✓	CordovaSampleApp-release.apk	Abhishek	App Package for Android	193.41 KB	Yes	4/18/2014 10:27:47 AM	No

Cordova Sample App.ipa

Current status of this software

- 8 Users have this software available
- 1 User attempted to download this application
- 2 Download requests were handled

General Information

Description:	Cordova Sample App.ipa
Publisher:	Abhishek
Size:	337.84 KB
Installer Type:	App Package for iOS (*.ipa file)
Last Updated:	4/18/2014 1:00:24 PM
Deployed:	Yes

Where are Microsoft going with this?

- Cordova contributions
  - Improve the support matrix for most popular plugins
  - Support for Windows Platform
- Stay in the browser longer
  - Cordova Browser Platform
  - Evolve the Ripple emulator
- Interoperability with other tools
  - CLI
  - Frameworks (e.g. Ionic)
  - Grunt, Bower, etc.
- Better coding & debug experiences
  - Intellisense for popular frameworks
  - TypeScript

Demo

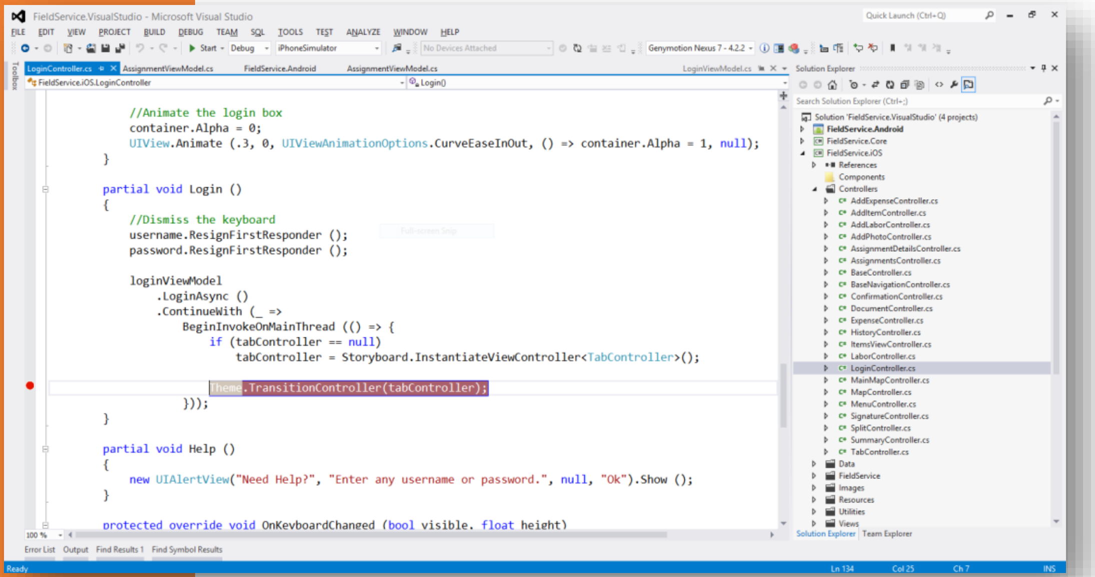
What is Xamarin?

Extension to Visual Studio

- iOS, Android apps entirely within Visual Studio
- Compiles .NET/C# code to native platforms
- Visual Studio ALM and IDE capabilities fully available

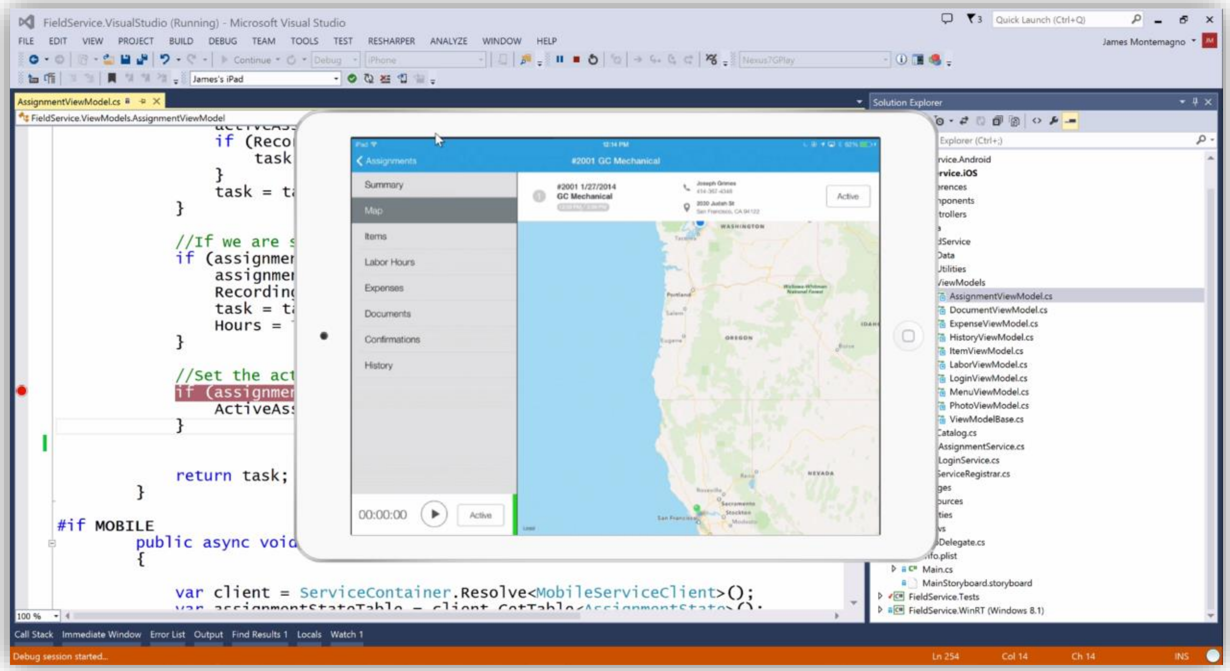
Build apps faster

- Leverage skills
- Reuse code and binaries with flexibility
- Fully native user interfaces


FieldService.VisualStudio - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP
Start Debug iPhoneSimulator No Devices Attached Genymotion Nexus 7 - 4.2.2
LoginController.cs AssignmentViewModel.cs FieldService.Android AssignmentViewModel.cs LoginViewModel.cs
FieldService.iOS.LoginController
Login()
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'FieldService.VisualStudio' (4 projects)
FieldService.Android
FieldService.Core
FieldService.iOS
References
Components
Controllers
AddExpenseController.cs
AddItemController.cs
AddLaborController.cs
AddPhotoController.cs
AssignmentDetailsController.cs
AssignmentsController.cs
BaseController.cs
BaseNavigationController.cs
ConfirmationController.cs
DocumentController.cs
ExpenseController.cs
HistoryController.cs
ItemsViewController.cs
LaborController.cs
LoginController.cs
MainMapController.cs
MapController.cs
MenuController.cs
SignatureController.cs
SplitController.cs
SummaryController.cs
TabController.cs
Data
FieldService
Images
Resources
Utilities
Views
Solution Explorer Team Explorer
100 %
Error List Output Find Results 1 Find Symbol Results
Ready Ln 134 Col 25 Ch 7 INS

```
            //Animate the login box
            container.Alpha = 0;
            UIView.Animate (.3, 0, UIViewAnimationOptions.CurveEaseInOut, () => container.Alpha = 1, null);
        }

        partial void Login ()
        {
            //Dismiss the keyboard
            username.ResignFirstResponder ();
            password.ResignFirstResponder ();

            loginViewModel
                .LoginAsync ()
                .ContinueWith (_ =>
                    BeginInvokeOnMainThread (() => {
                        if (tabController == null)
                            tabController = Storyboard.InstantiateViewController<TabController>();

                        Theme.TransitionController(tabController);
                    }));
        }

        partial void Help ()
        {
            new UIAlertView("Need Help?", "Enter any username or password.", null, "Ok").Show ();
        }

        protected override void OnKeyboardChanged (bool visible, float height)
```

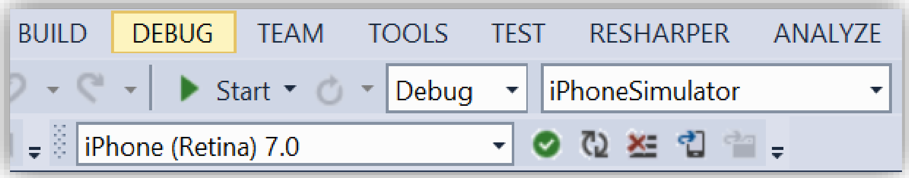
Visual Studio Integration

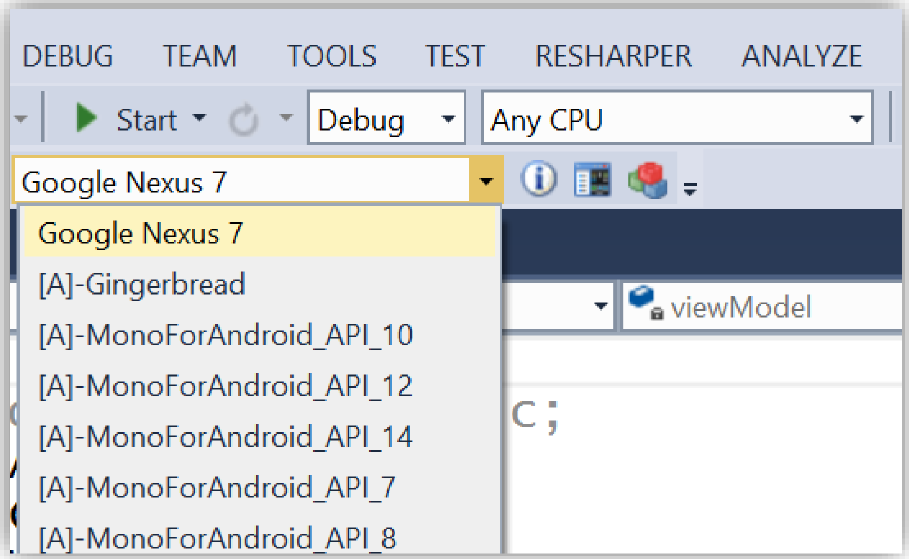


```
                    if (Reco
                        task
                    }
                    task = ta
                }

                //If we are s
                if (assignmer
                    assignmer
                    Recordin
                    task = ta
                    Hours = 
                }

                //Set the act
                if (assignmer
                    ActiveAs
                }

                return task;
            }

#if MOBILE
            public async voi
            {

                var client = ServiceContainer.Resolve<MobileServiceClient>();
```

- A single solution:
  - iOS
  - Android
  - Windows Phone
  - Windows Store
- Leverage the entire Microsoft ecosystem:
  - ReSharper
  - Team Foundation Server
  - Your favorite code coverage and profiling tools

Visual Studio Integration

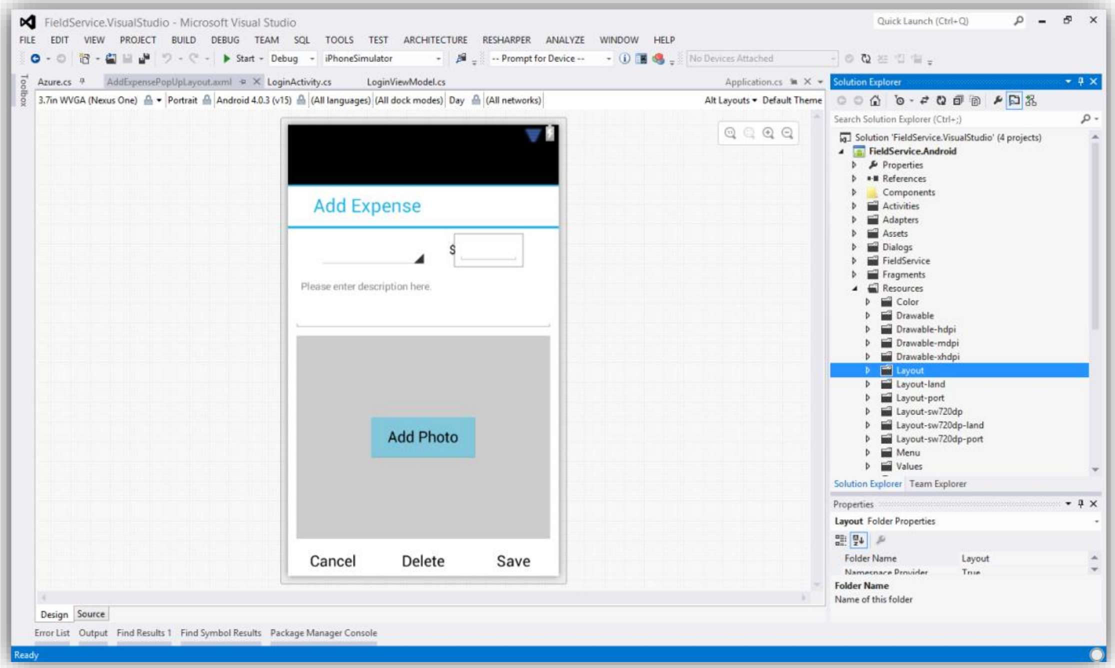




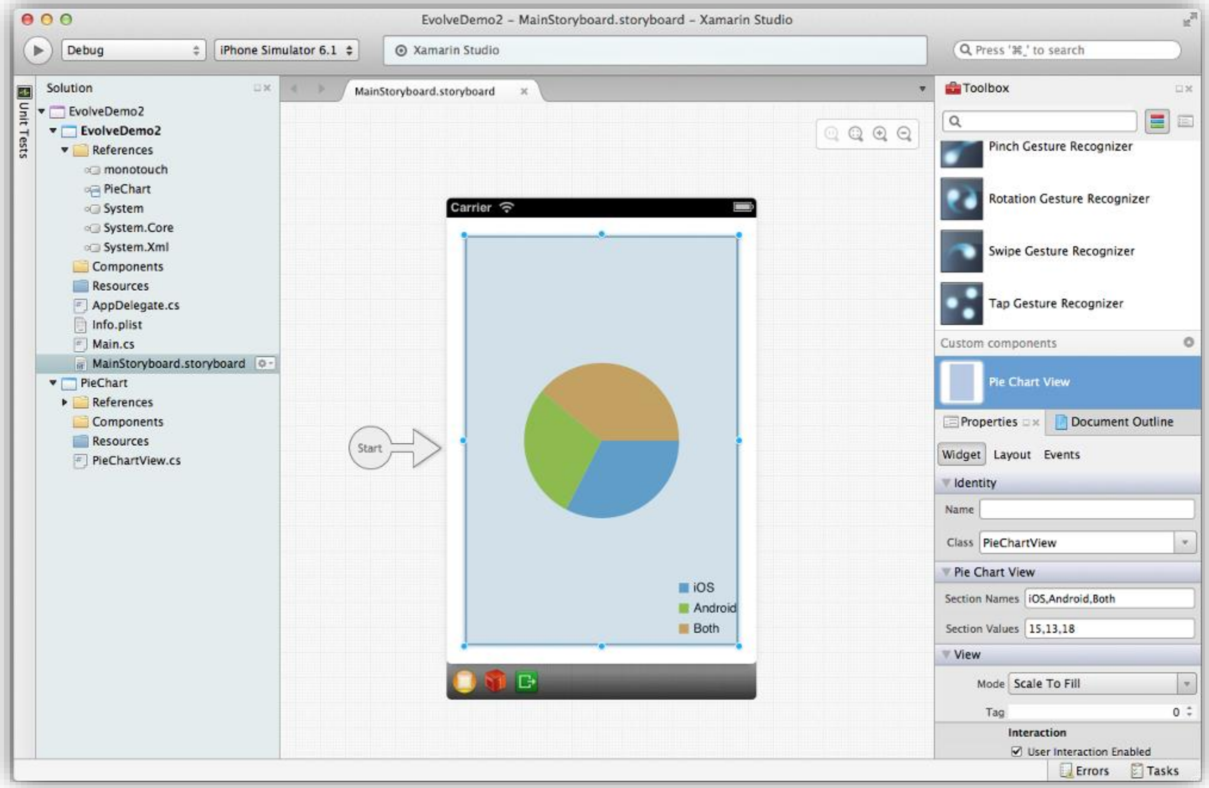
- Debug to:
  - Emulators
  - Devices
- Integrated into toolbar
  - Status
  - Logs
  - List of devices
- Just Click Start Debugging!

Android Designer

- Full-featured Android designer
- Available in:
  - Xamarin Studio
  - Visual Studio
- Create UI with drag & drop simplicity
- Target multiple screen sizes, resolutions and Android versions
- Layouts saved in standard Android XML files



iOS Designer

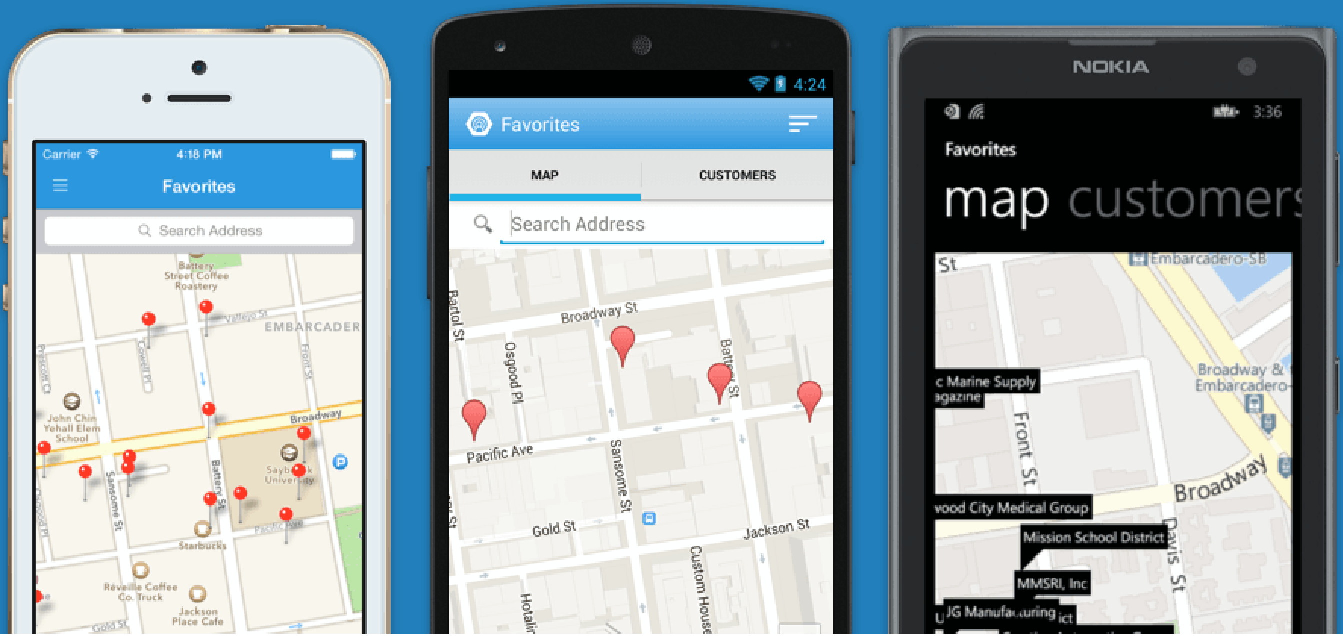


Name:
Class: PieChartView
Section Names: iOS,Android,Both
Section Values: 15,13,18
Mode: Scale To Fill
Tag: 0
☑ User Interaction Enabled

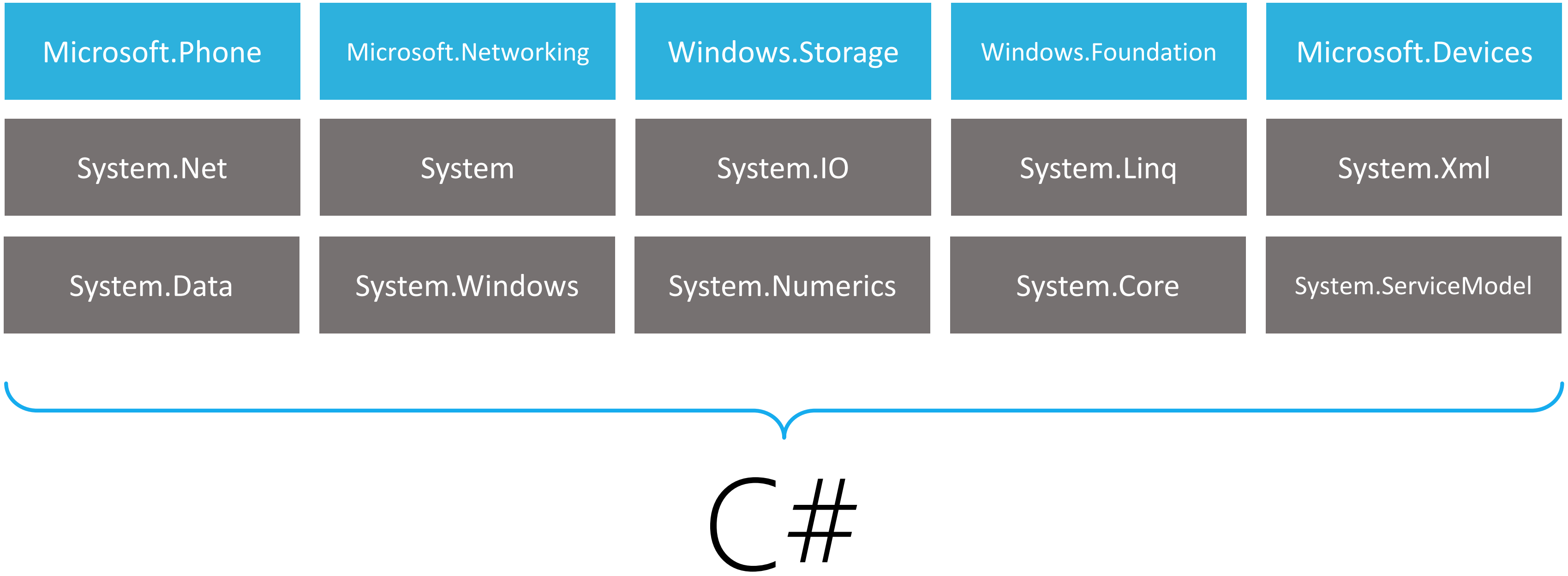
- New iOS Designer available in Xamarin Studio (and soon Visual Studio)
- Follows familiar Visual Studio designer idioms
- Supports all UIKit elements
- Edit custom and 3rd party components
- Live preview of changes to properties

Meet Xamarin.Forms

Build native UIs for iOS, Android and Windows Phone from a single, shared C# codebase.



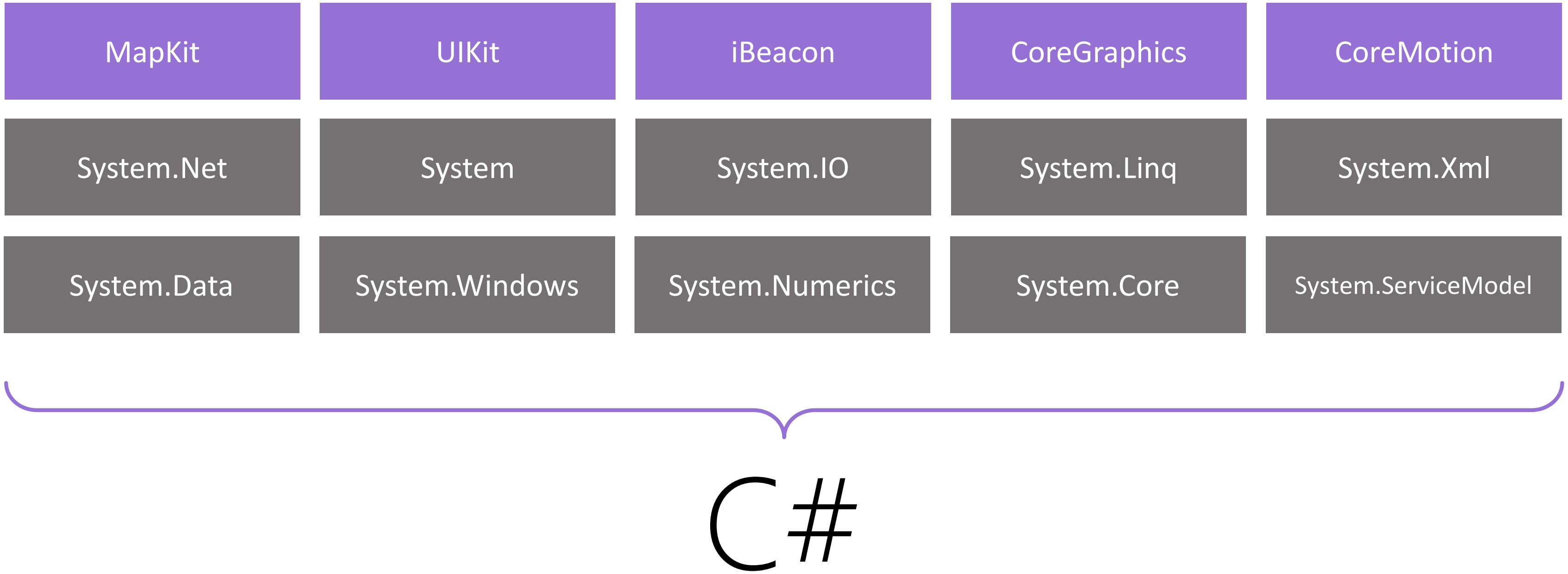
Windows APIs



Microsoft.Phone	Microsoft.Networking	Windows.Storage	Windows.Foundation	Microsoft.Devices
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

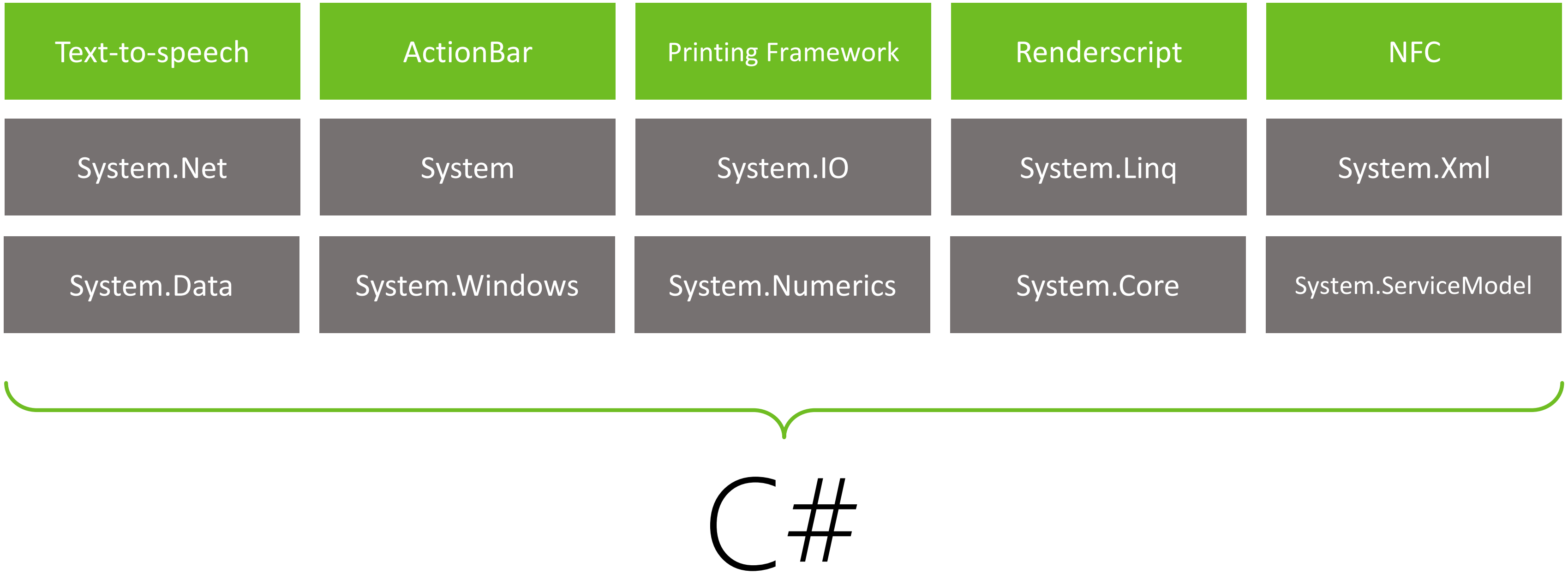
iOS – 100% API Coverage



MapKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

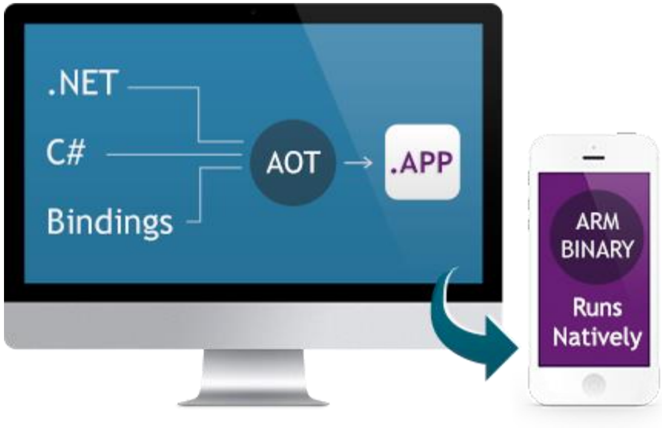
Android – 100% API Coverage

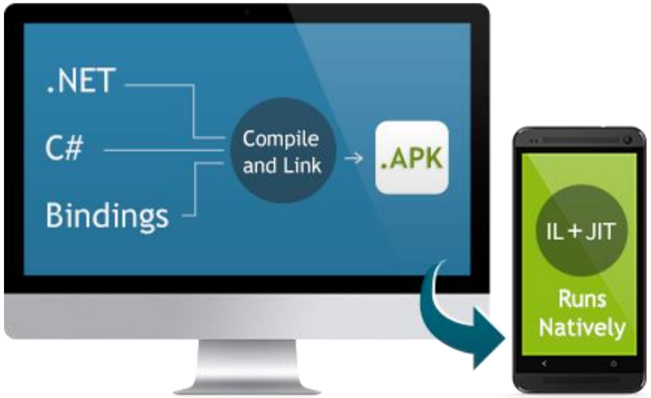


Text-to-speech	ActionBar	Printing Framework	Renderscript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

Native Performance: How it works

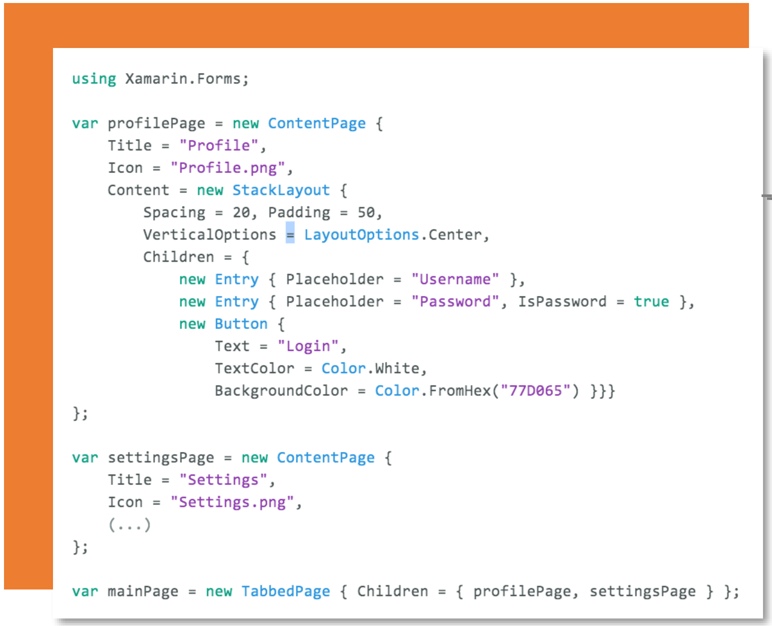




- Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.

- Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

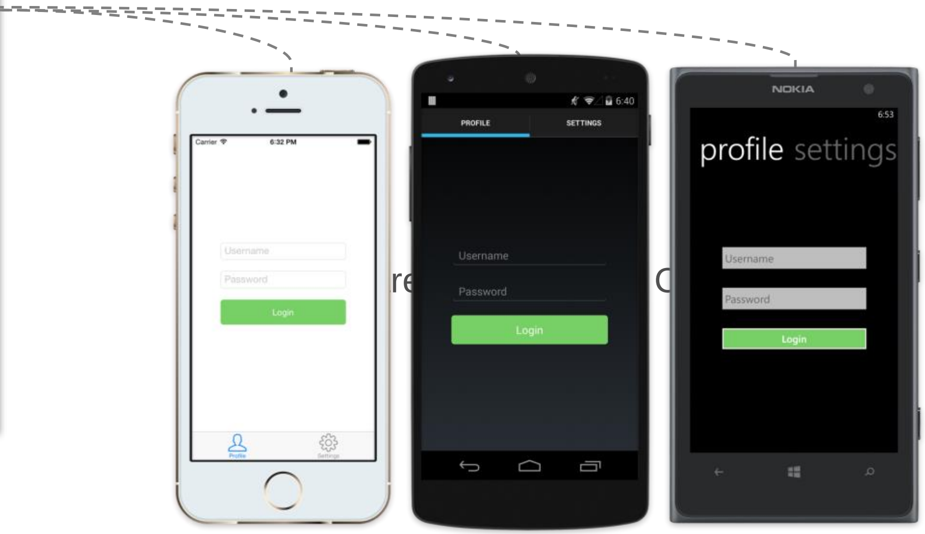
How Xamarin.Forms works



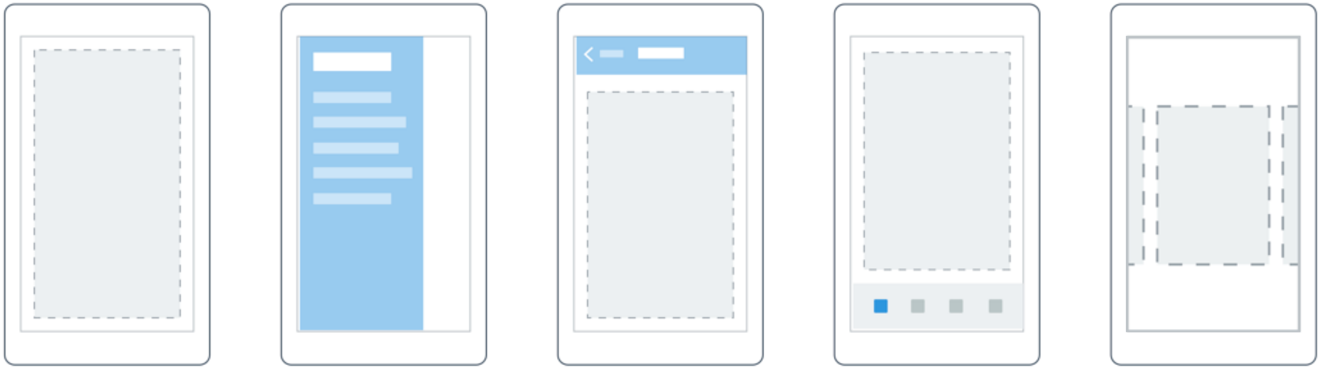
```
using Xamarin.Forms;

var profilePage = new ContentPage {
    Title = "Profile",
    Icon = "Profile.png",
    Content = new StackLayout {
        Spacing = 20, Padding = 50,
        VerticalOptions = LayoutOptions.Center,
        Children = {
            new Entry { Placeholder = "Username" },
            new Entry { Placeholder = "Password", IsPassword = true },
            new Button {
                Text = "Login",
                TextColor = Color.White,
                BackgroundColor = Color.FromHex("77D065") }}}
};

var settingsPage = new ContentPage {
    Title = "Settings",
    Icon = "Settings.png",
    (...)
};

var mainPage = new TabbedPage { Children = { profilePage, settingsPage } };
```

Use a single API to generate native, platform-specific user interfaces

At **runtime**, each Xamarin.Forms page and its controls are mapped to platform-specific native user interface elements



Pages



Content

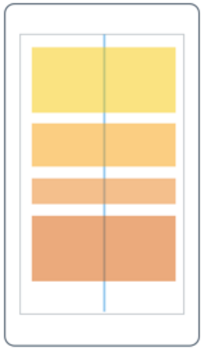
MasterDetail

Navigation

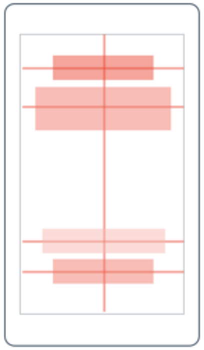
Tabbed

Carousel

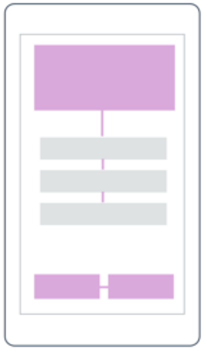
Layouts



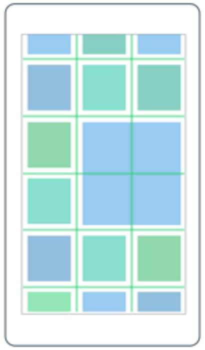
Stack



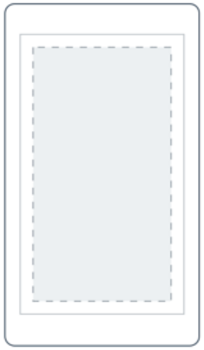
Absolute



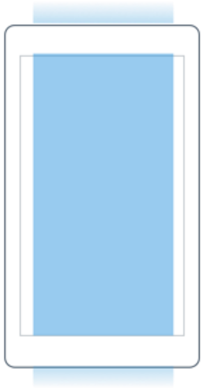
Relative



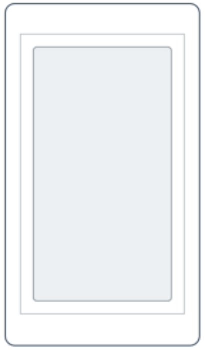
Grid



ContentView



ScrollView

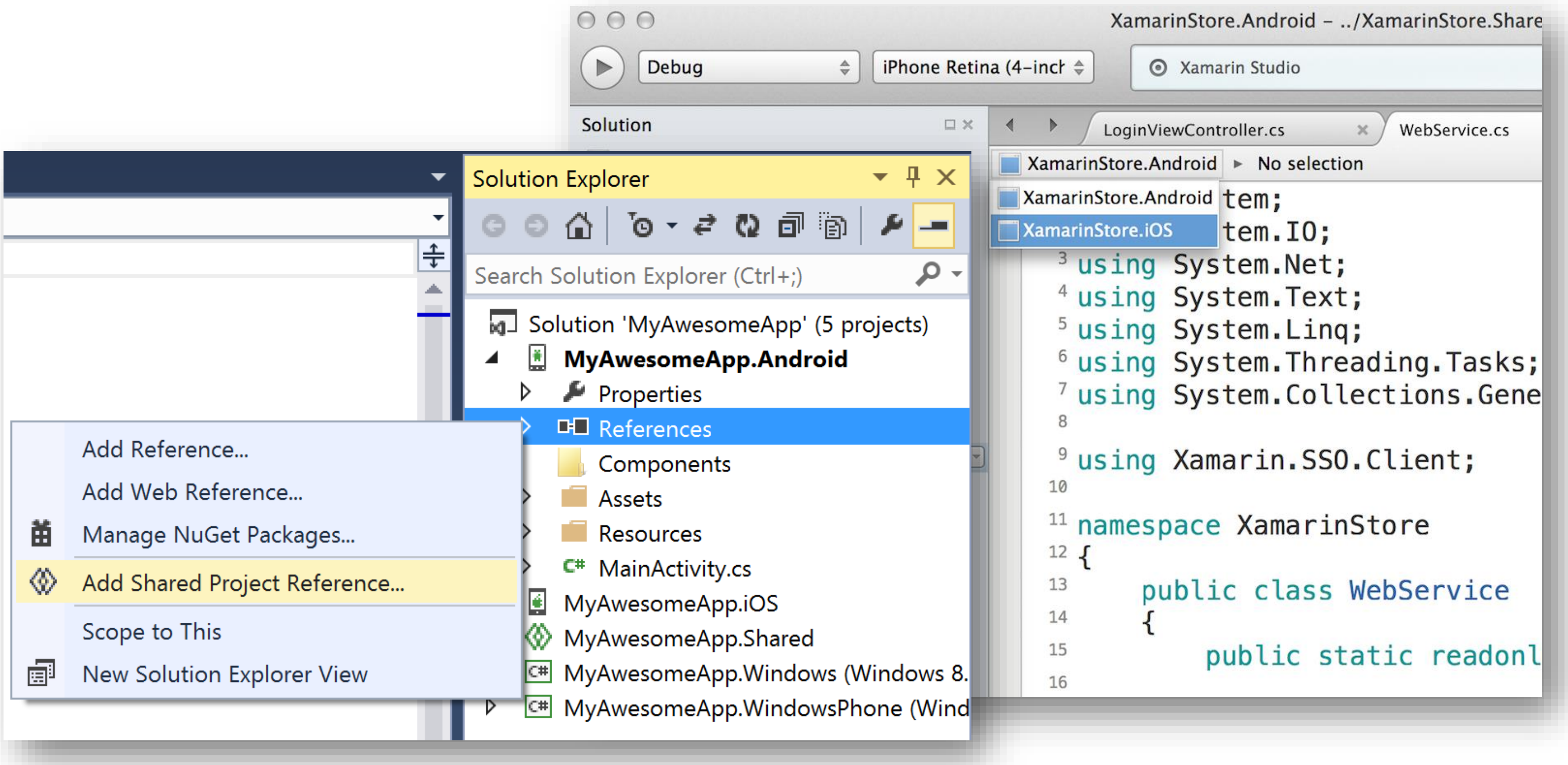


Frame

Controls

- ActivityIndicator
- BoxView
- Button
- DatePicker
- Editor
- Entry
- Image
- Label
- ListView
- Map
- OpenGLView
- Picker
- ProgressBar
- SearchBar
- Slider
- Stepper
- TableView
- TimePicker
- WebView
- EntryCell
- ImageCell
- SwitchCell
- TextCell
- ViewCell

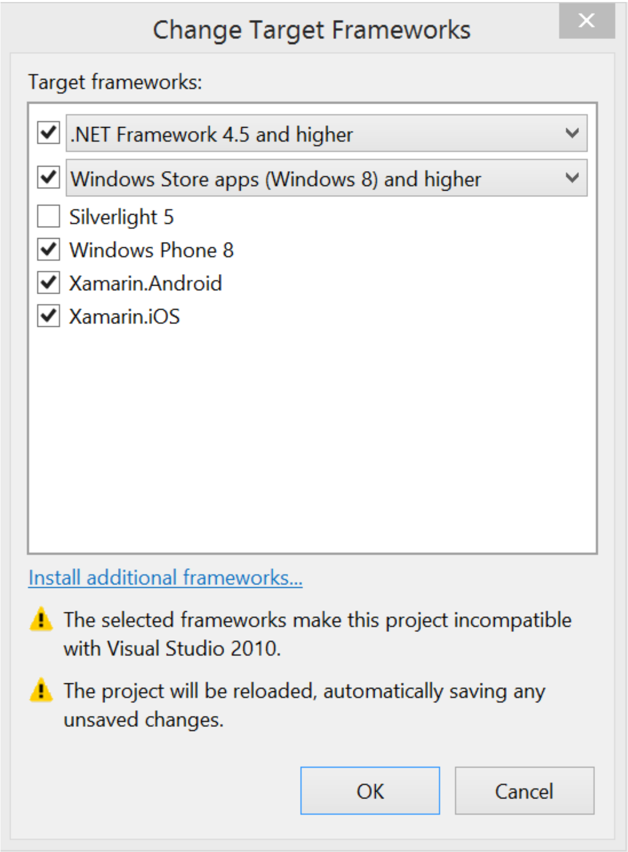
Shared Projects



```
tem;
tem.IO;
using System.Net;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Collections.Gene

using Xamarin.SSO.Client;

namespace XamarinStore
{
    public class WebService
    {
        public static readonl
```

Choosing a Code Sharing Option

- Shared Projects / File Linking
  - Reuse the same files in multiple projects
  - Can use conditional compilation (i.e. #if)
- Portable Class Libraries
  - Reuse your past investments in .NET code
  - Common Denominator only
  - Common code only, no #if conditionals
- Combo Approach
  - Not mutually exclusive, use both in same solution
  - e.g. Shared Project with Xamarin.Forms + UI code, combined with shared non-UI code in PCL



Change Target Frameworks

Target frameworks:

☑ .NET Framework 4.5 and higher
☑ Windows Store apps (Windows 8) and higher
□ Silverlight 5
☑ Windows Phone 8
☑ Xamarin.Android
☑ Xamarin.iOS

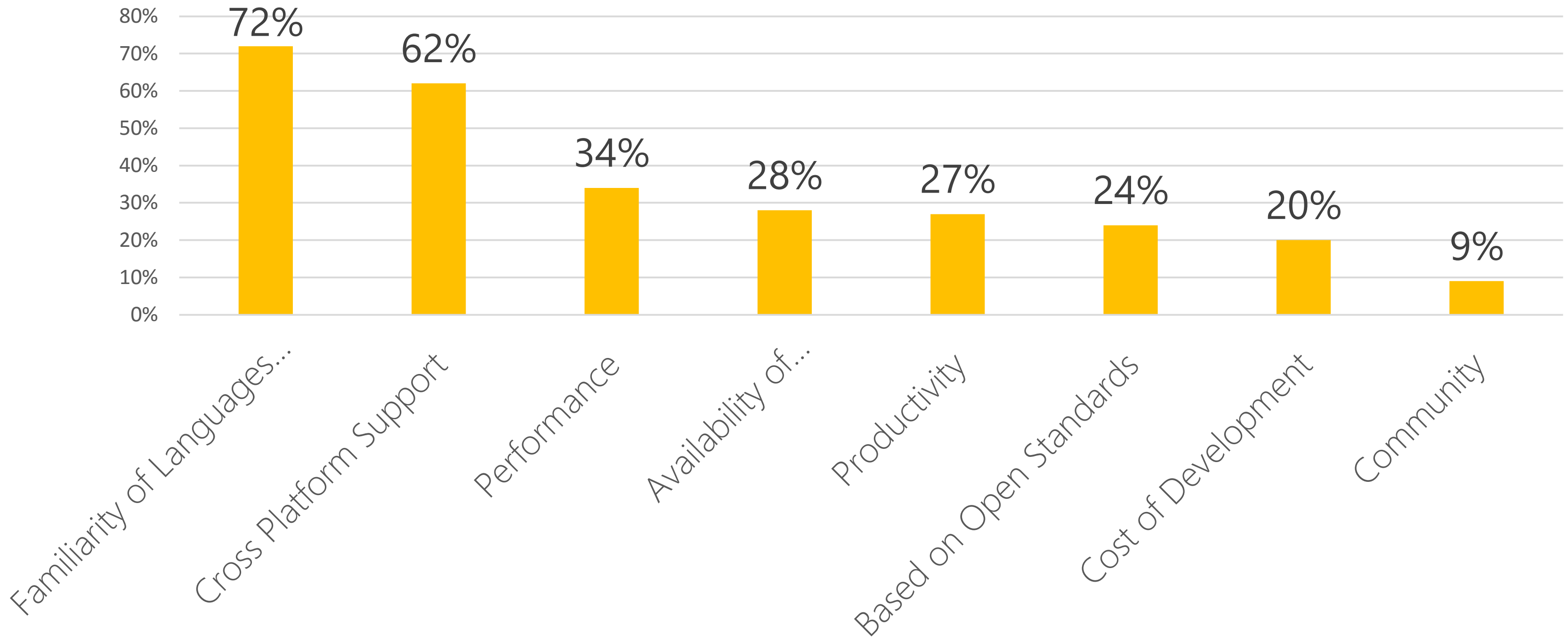
Install additional frameworks...

⚠ The selected frameworks make this project incompatible with Visual Studio 2010.

⚠ The project will be reloaded, automatically saving any unsaved changes.

Demo

Why Cordova?



Source: Kendo UI Developer Survey 2013

Why Xamarin?

- High performance application that can access native API
- Strong typed and object oriented development to support you flexible architecture
- Drag and Drop user interface development. If you are Windows Developer, no need to learn JS or HTML. You can build using drag and drop style for IOS or Android.
- Xaml UI and adaptive layout.
- Based on Mono runtime and totally open source.

Picking between Xamarin and Apache Cordova

Xamarin is generally better when you:

- Xamarin.Forms can help you implement device specified UI
- Have .Net developers especially UWP developers
- Need a high performing, native looking app like a game

Cordova with Visual Studio is generally better when you:

- Want to share more UI between platforms, with fewer native looking elements.
- Share UI assets with a mobile website
- Have HTML / JS developers
- Are OK with a somewhat less smooth UI on older Android and iOS devices, or don’t plan on supporting older Android

No matter what's your choice.
Visual Studio 2015 offers both and FREE!

Thanks