



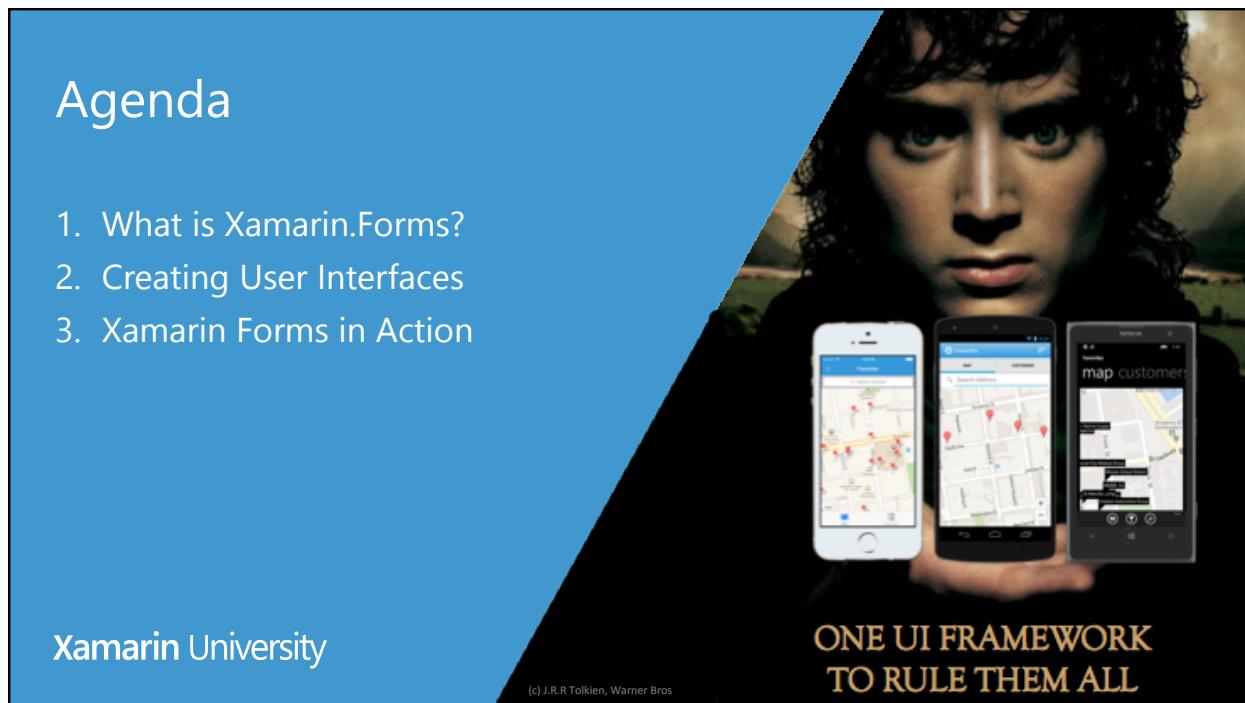
Xamarin Evolve 2014

Introduction to Xamarin.Forms



 Rob Gibbens
rob.gibbens@xamarin.com
@RobGibbens

 Xamarin University



Agenda

1. What is Xamarin.Forms?
2. Creating User Interfaces
3. Xamarin Forms in Action





ONE UI FRAMEWORK
TO RULE THEM ALL

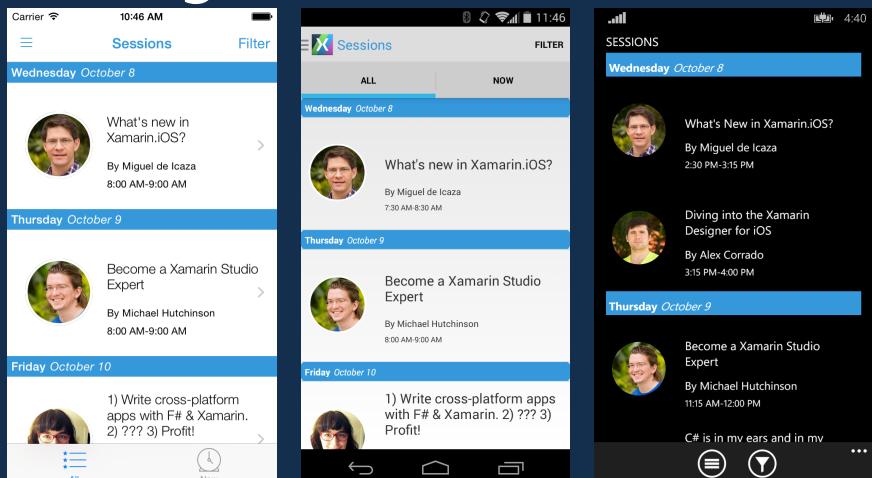
(c) J.R.R Tolkien, Warner Bros

Xamarin University

Xamarin.Forms?

Xamarin University

You've got to see it to believe it!



Xamarin University

Xamarin.Forms

- Shared Business Logic
- Platform specific UI code
- Shared UI Layer
- UI through code or XAML

The diagram illustrates the architecture of Xamarin.Forms. At the top, there are icons for Android (green robot), iOS (apple), and Windows (blue square). Below these are two main horizontal layers. The upper layer is orange and labeled "Shared UI Layer (C#, XAML)". The lower layer is blue and labeled "Shared App Logic (C#)". Within the "Shared App Logic" layer, there are two sub-layers: "Shared Project" and "Portable Class Library". Below the "Shared App Logic" layer are three separate colored boxes corresponding to the platforms: green for Android, grey for iOS, and blue for Windows.

Xamarin University

Features

- RAD
- Android 4.0+
- iOS 6.1+
- Windows Phone 8 (Silverlight)

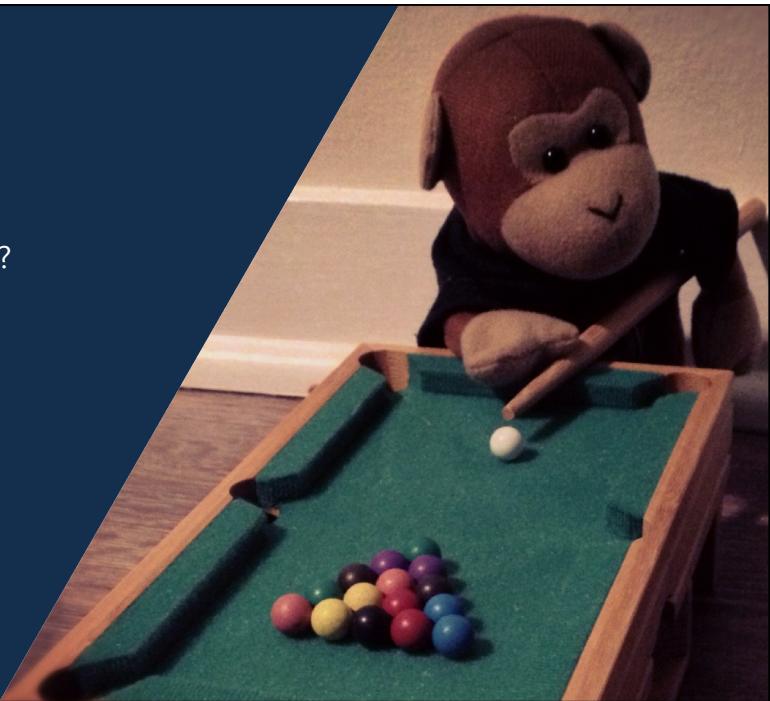
The diagram shows seven interconnected features represented as hexagons. In the center is a purple hexagon labeled "Multi-Page Navigation". Surrounding it are six other hexagons: "Standard Controls" (green, top-left), "Flexible Layout" (red, top-right), "Custom Controls" (grey, right), "Data Binding Engine" (grey, bottom-right), "XAML" (grey, bottom-left), and another grey hexagon (left). The features are interconnected by lines forming a network.

Xamarin University

Considerations

- Not for all types of apps
- Is your app very customized?
- Great for data driven apps
- Utility apps
- Also supports maps

Xamarin University



Traditional Xamarin development

- Native look and feel
- Native performance
- Code sharing/re-use
- Direct access to native SDKs



Layout

UIKit

XAML

Shared App Logic (C#)

Shared Project

Portable Class Library

Xamarin University

Xamarin.Forms

- Native look and feel
- Native performance
- Code sharing/re-use
- Native SDKs via Custom Renderers and DependencyService

The diagram illustrates the Xamarin.Forms architecture. At the top, there are icons for Android (green robot), iOS (apple), and Windows (blue square). Below these are two stacked orange boxes: 'Shared UI Layer (C#, XAML)' and 'Shared App Logic (C#)'. The 'Shared UI Layer' box contains a green bar (Android), a grey bar (iOS), and a blue bar (Windows). Below 'Shared App Logic' are two grey boxes: 'Shared Project' and 'Portable Class Library'.

Xamarin University

Project Structure

Platform-specific projects act as "host" to create native application

Portable Class Library or Shared Project used to hold shared code that defines UI and logic

Xamarin University

Demo

Xamarin University

Creating User Interfaces

Xamarin University

Controls

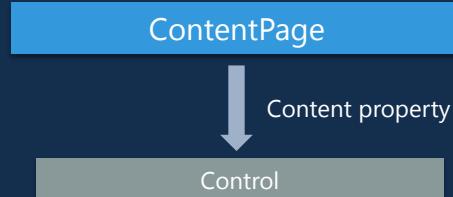
- All standard controls are there
- Button
- Picker
- ListView
- WebView
- Many more
- Create your own



Xamarin University

Pages

- Single screen of content
- ContentPage holds one visual element



Xamarin University

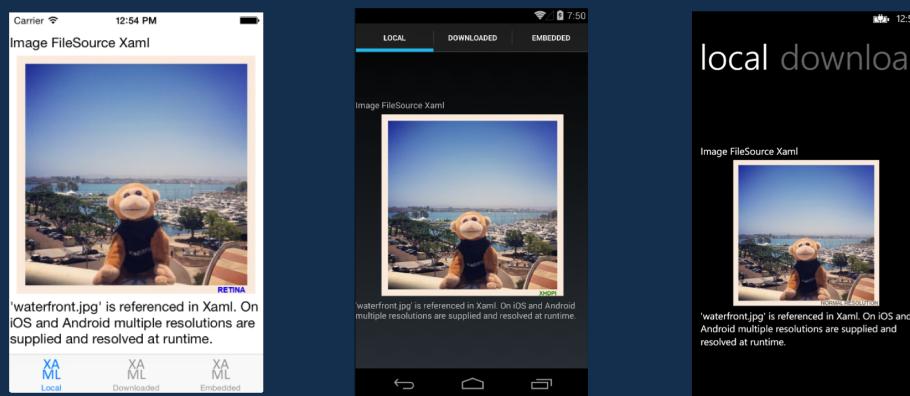
Other Page Types

- MasterDetailPage
- NavigationPage
- TabbedPage
- CarouselPage



Xamarin University

Example: TabbedPage



Xamarin University

Layout

- Organize Child Elements
- StackLayout
- Grid
- RelativeLayout
- Create your own!

Xamarin University

Layout

- Layouts handle child elements
- Layouts come in two types: managed and unmanaged

```
graph TD; ContentPage[ContentPage] --> Content[Content property]; Content --> Control1[Control]; Control1 --> Children1[Children]; Children1 --> Control2[Control]; Children1 --> Control3[Control]; Children1 --> Control4[Control]
```

Xamarin University

Demo

Xamarin University

Providing Behavior

Controls expose *properties* to alter visualization

```
var entry = new Entry
{
    Placeholder = "Enter text",
    Keyboard = Keyboard.Email
};

entry.TextChanged += (sender, e) => {
    // Input changed
};
```

Controls expose *events* to provide interactive behavior

Xamarin University

Demo

Xamarin University

Summary

- Cross platform apps the easy way
- Powerful layout functionality
- Many built in controls
- Extensible

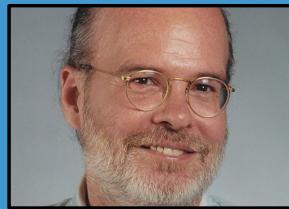
Xamarin University

Related Sessions

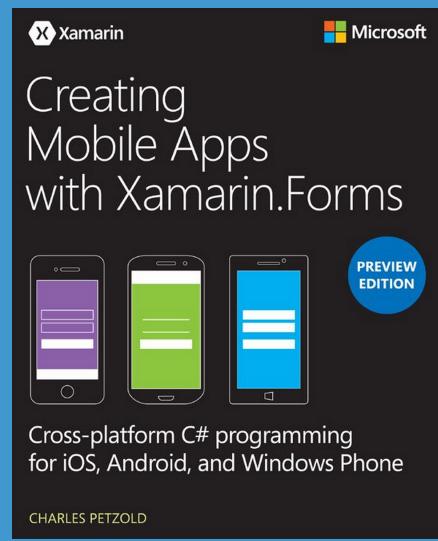
- **Your First Xamarin.Forms App**
Wednesday, 9:00 - Craig Dunn
- **Building Cross-Platform Applications with Xamarin, Xamarin.Forms and MVVM Light**
Wednesday, 5:00 - Laurent Bugnion
- **XAML for Xamarin.Forms**
Thursday, 10:00 - Charles Petzold
- **Extending Xamarin.Forms with your own controls and layouts**
Thursday, 3:15 - Jason Smith
- **Xamarin.Forms is Even Cooler than You Think**
Friday, 9:00 - Charles Petzold

Xamarin University

Read the book



Charles Petzold
Xamarin



Xamarin University

Questions?

Xamarin University

Xamarin Evolve 2014



Introduction to
Xamarin.Forms



Rob Gibbens
rob.gibbens@xamarin.com
@RobGibbens



Xamarin
University