

# Xamarin 研讨会

2012年4月19日

# Agenda

## 跨平台移动开发



Greg Shackles

高级软件工程师

OLO, 在线咨询

[greg@gregshackles.com](mailto:greg@gregshackles.com)

gregshackles.com

@gshackles

github.com/gshackles

Xamarin

# 简介

本次研讨会旨在讨论如何利用C#为iOS、Android以及Windows Phone开发应用程序。

我们的议题包括：

- 为什么要使用C#?
- 代码共享技术及模式

常用库

Xamarin

# 本地平台语言



Xamarin

# 一次编写，平台通吃？



!=

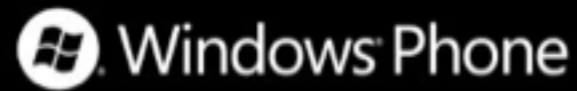


!=



Xamarin

# C# 前来拜访!



Microsoft®  
**.NET**



**MonoTouch**



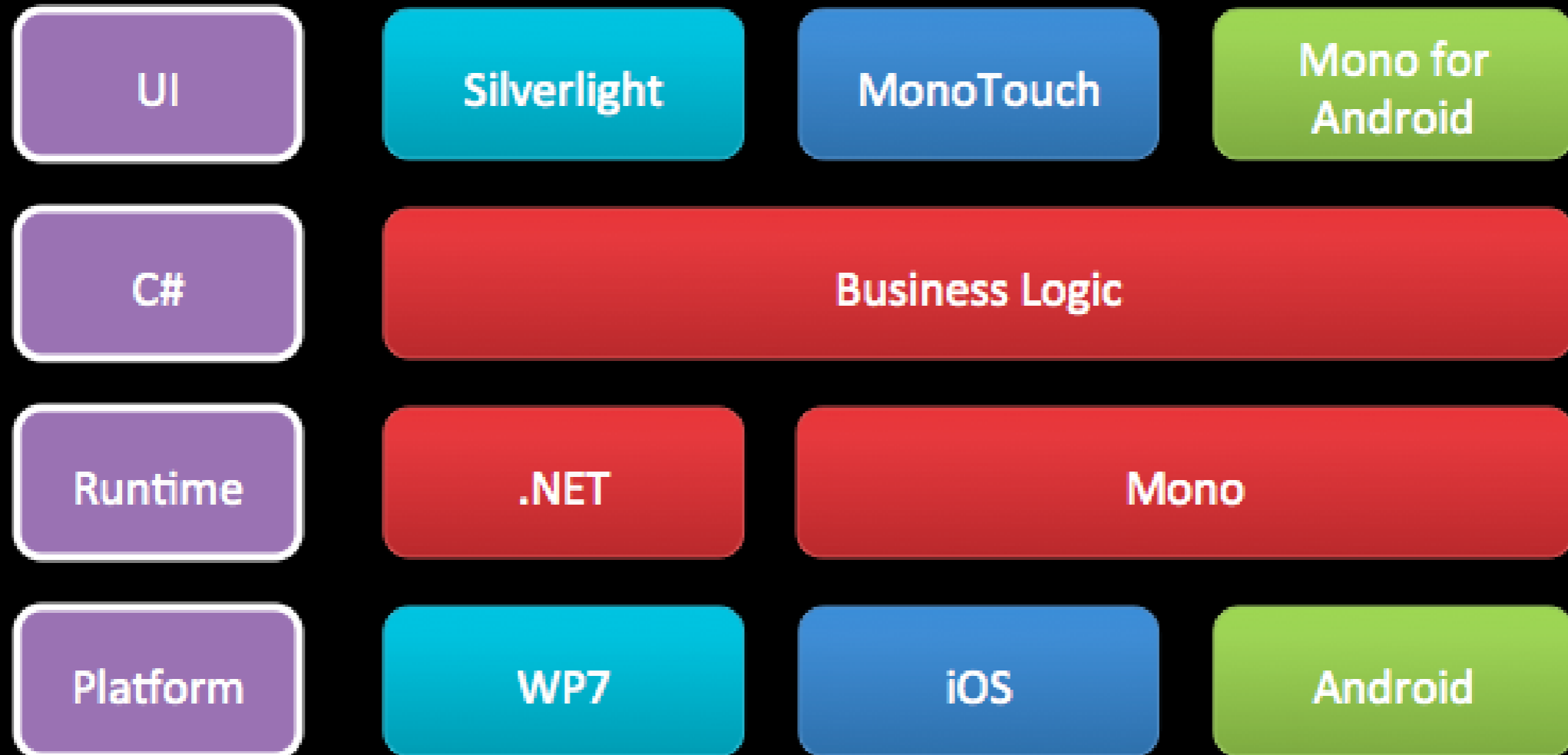
**mono**  
for Android™

Xamarin

# 优势

- C# 与 .NET 成熟而强大
- 编码技能可有效作用于各平台
- 应用程序仍保持本机特色
- 代码可在跨平台时重新使用（甚至包括非移动平台！）

# 架构





# 哪些代码能够共享？

- 大多数非UI或平台代码
- 核心应用逻辑
- 字符实体
- LINQ (对象, XML)
- 网络访问(System.Net)

# 必备工具

- MonoTouch

- Mac OS X

- MonoDevelop

- Mono for Android

- Mac OS X 或 Windows

- MonoDevelop 或 Visual Studio 2010

- Windows Phone

- Windows

- Visual Studio 2010

# 文件连接

- 各配置文件并不完全相同
- 单独文件副本
- 编译时间验证
- VS2010项目连接器扩展

<http://msdn.microsoft.com/en-us/library/ff648745>

# 抽象

- 从UI中提取通用逻辑
- 接口、基本类等

```
public interface IContactManager
{
    IList<Person> GetContacts();
}
```

```
public class AndroidContactManager : IContactManager
{
    public IList<Person> GetContacts()
    {
        return null;
    }
}
```

# 库: Xamarin.Mobile

- 平台API上的抽象层
- 通讯簿
- 地理位置
- 照相机
- ...

<http://xamarin.com/mobileapi>

# 观察者模式

- 从业务逻辑层中提取UI
- 业务层能够发布更新
- UI层接收更新内容

```
public EventHandler<EventArgs> MessageReceived;  
  
MessageReceived += (sender, args) =>  
{  
    Console.WriteLine("Message received");  
};
```

# 库: TinyMessenger

- 事件整合器/提示器
- 发布/接收
- 单独文件
- 支持 iOS, Android与Windows Phone

<https://github.com/grumpydev/TinyMessenger>

# 局部类与局部方法

```
public partial class MyClass
{
    // partial methods are private, and must return void
    partial void Foo();

    public void Bar()
    {
        Foo();
    }
}
```

```
public partial class MyClass
{
    partial void Foo()
    {
        Console.WriteLine("Foo");
    }
}
```



# 条件编译

```
#if __ANDROID__  
    Console.WriteLine("Only on Android");  
#elif WINDOWS_PHONE  
    Console.WriteLine("Only on Windows Phone");  
#else  
    Console.WriteLine("Everything else");  
#endif
```

# 条件编译

- MonoTouch
- No default symbols
- Mono for Android
  - `__ANDROID__`
  - `__ANDROID_1__`
  - `__ANDROID_2__`
  - ...
- Windows Phone
  - `WINDOWS_PHONE`
  - `SILVERLIGHT`

Xamarin

# 文件访问

- 正向文件访问
  - `System.IO`
  - 适用于iOS 及 Android
  - Windows Phone 不提供支持
  - 路径与iOS及Android不同

```
File.WriteAllText(  
    Environment.GetFolderPath(Environment.SpecialFolder.Personal),  
    "Writing directly to a file");
```

# 文件访问

- 独立存储
  - 适用于iOS, Android以及Windows Phone
  - 更高级别的API
  - 不必考虑文件路径

```
using (var store = IsolatedStorageFile.GetUserStoreForApplication())  
using (var writer = store.OpenFile("MyFile", FileMode.Create))  
{  
    // write to the file  
}
```

# 数据库访问

- iOS 与 Android
  - SQLite
  - 本地 APIs 或 ADO.NET
- Windows Phone
- SQL Server CE
- LINQ to SQL

# 库: C#-SQLite

- SQLite数据库的C#端口
- 基础文件
- 适用于Windows Phone及Silverlight

<http://code.google.com/p/csharp-sqlite>

# 库: sqlite-net

- 层位于SQLite数据库之上
- 强类型查询
- 单独文件
- 支持 iOS, Android以及Windows Phone

<http://code.google.com/p/sqlite-net>

# 库: TinyIoC

- 反向控制容器
- 单独文件
- 支持 iOS, Android 以及 Windows Phone

<https://github.com/grumpydev/TinyIoC>



# 库: MonoCross

- 跨平台 MVC 框架
- 共享模型及控制器
- 平台专用视图
- 支持iOS, Android以及Windows Phone
- 基于MonoTouch与Mono for Android

<http://code.google.com/p/monocross/>

Xamarin

# 更多链接

MWC 2012 App <https://github.com/xamarin/mobile-samples/tree/master/MWC>

NYC Code Camp 6 App <https://github.com/gshackles/NycCodeCamp6> Mobile Development in C# <http://amzn.com/1449320236>

Xamarin

# Xamarin 研讨会

我们希望听到您的反馈

<http://bit.ly/xamfeedback>

在Twitter上关注我们  
**@XamarinHQ**

2012年4月19日