

Microsoft Tech Summit 2017

微软技术暨生态大会



Microsoft Tech Summit 2017

微软技术暨生态大会

Xamarin企业跨平台 移动应用的DevOps实践

DEV206

朱永光

MVP、技术顾问



定义



企业跨平台移动应用

- 基于Xamarin Forms，一套代码实现Android和iOS的UI和交互逻辑
- 面向内部用户，整合内部业务系统
- 内部分发
 - 使用iOS In-House证书
 - 通过内部网站来下载

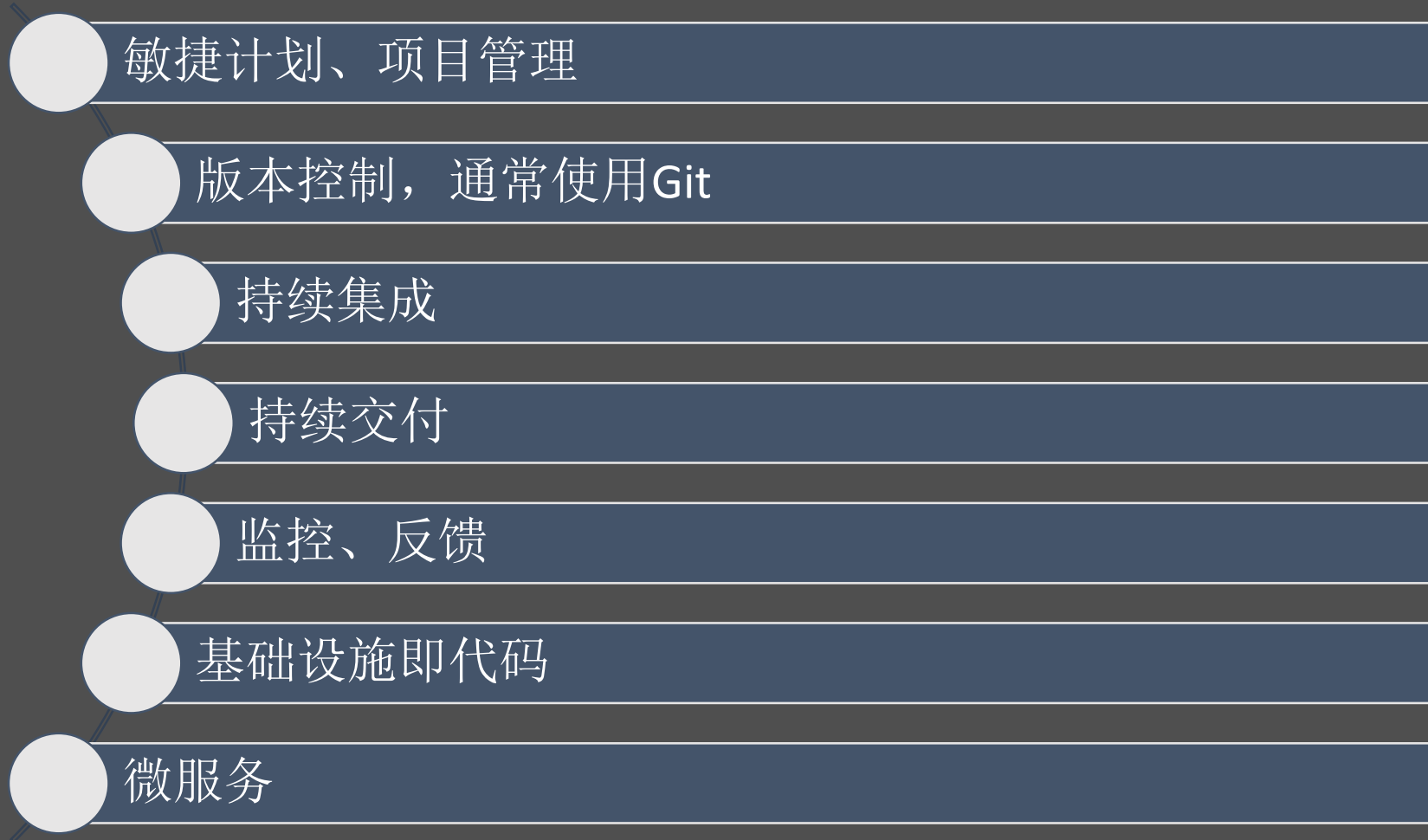
铺垫



微软对DevOps的理解

- 把人、过程和产品联合起来，能够把价值持续交付给最终用户
- 让团队更快、更低成本的交付更加安全和高质量的解决方案

DevOps涉及一系列最佳实践



目标



轻松实现Xamarin APP的DevOps

我将带领大家一步一步实践如下 “套路”

- 敏捷开发
- 版本控制
- 持续集成
- 持续交付
- 持续监控



对其他技术开发的APP同样适用

磨刀不费砍柴功



准备使用的工具或服务：

- VSTS/TFS
- Build Agent安装技巧
 - Windows
 - Mac (with PowerShell)
- 企业内部的分发站点
- Mobile Center

Windows Build Agent安装

- Windows/VS2017 Community
- Nodejs/Android SDK/Xamarin
- 下载Agent安装包
 - 解压
 - 配置：<https://docs.microsoft.com/zh-cn/vsts/build-release/actions/agents/v2-windows>
 - Windows Services运行
 - 手动添加没有自动检测到的能力（AndroidSDK、Xamarin.Android等）

Demo : 安装 Windows Build Agent



Mac Build Agent安装

- macOS 10.10+ / XCode 8+ / git 2.9.0+ / .NET Core 2.0 / PowerShell 6.0
- VS Community for Mac / Xamarin
- 下载Agent安装包
 - 解压
 - 配置 : <https://docs.microsoft.com/zh-cn/vsts/build-release/actions/agents/v2-osx>
 - console运行

敏捷开发



毫无悬念的Scrum+Kanban

- 选择Scrum模板创建团队项目
- 基于用户故事来描述前后端需求
- 灵活运用需求看板的列和泳道：
 - 列跟踪状态：新提需求和Bug、分析、开发、测试、UAT、上线
 - 泳道跟踪关注：生产Bug.....

版本控制



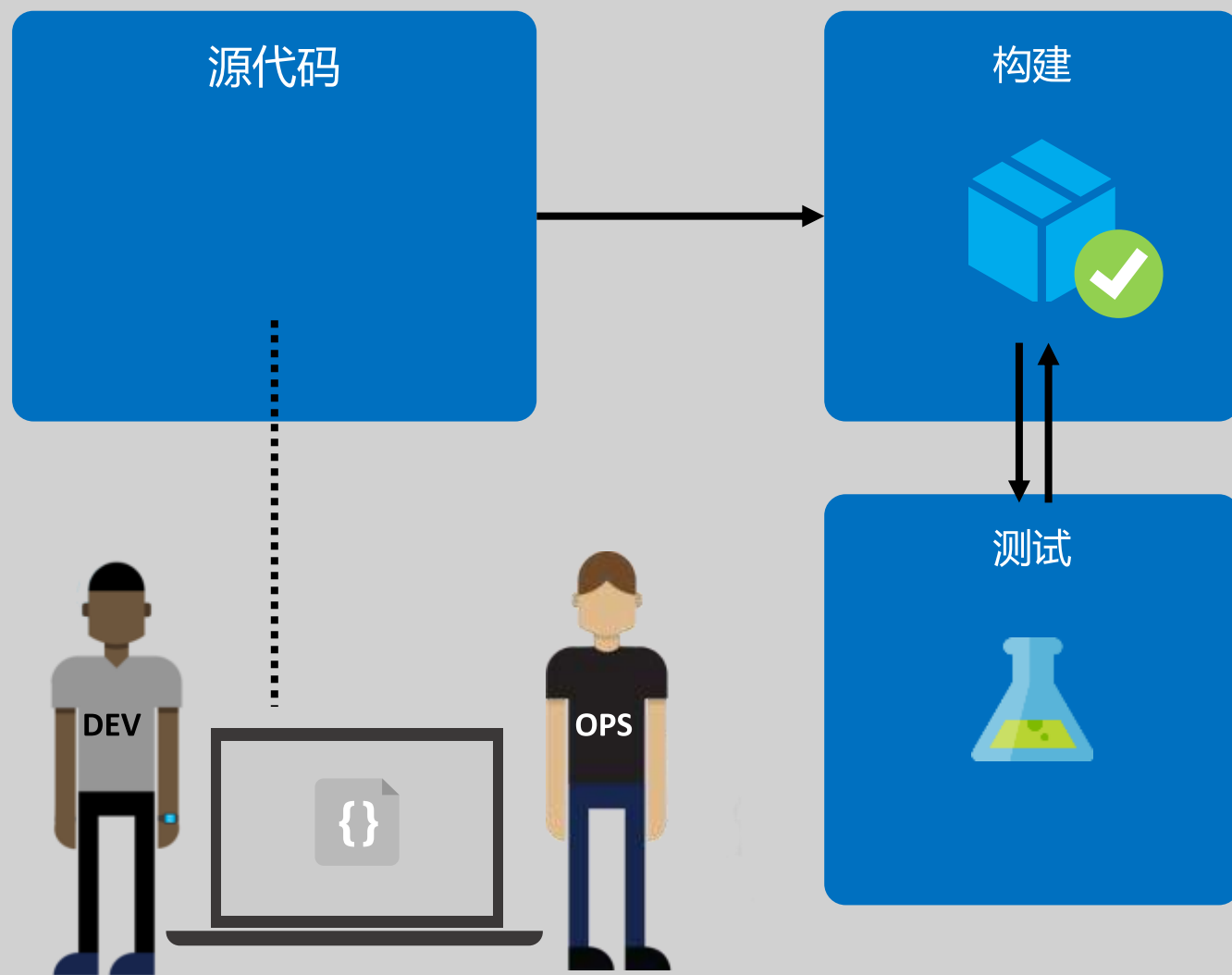
你不用Git难道用TFVC？

- 在一个团队项目中建立多个Repository分别存储前后端代码
- 对于移动端代码采用多分支模式而非单主干模式，Why？

持续集成



什么是持续集成



价值

- 加速交付
- 可重复
- 优化资源

度量

- 部署耗时
- MTTR
- MTTD

使用VSTS进行持续集成

- 真正的跨平台（ nodejs、 PowerShell ）
- 方便自定义
- 丰富的扩展（ 362 ）
- 提供了覆盖大部分应用的模板
 - Xamarin Android和Xamarin iOS

MobileCore-Test-Android



✓ 2017.10.09

MobileCore-UAT-Android



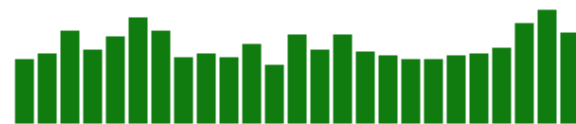
✓ 2017.10.09

MobileCore-Prod-Android



✓ 2017.09.26

BackendConsole-CI



✓ 2017.10.10

MobileCore-Test-iOS



✓ 2017.10.09

MobileCore-UAT-iOS



✓ 2017.10.09

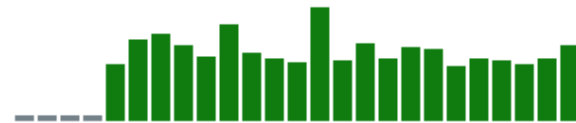
MobileCore-Prod-iOS



✓ 2017.09.26

Configure W

BackendService-CI



✓ 2017.09.29

MobileCore-Prod-iOS-AppStore



✓ 2017.09.26

DemoApi-CI



✓ 2017.09.23



VSTS的Build配置

任务

- 过程
- 阶段
 - 步骤

变量

- 过程变量
- 预定义变量

触发器

- CI触发
- 定时触发

选项

- 构建属性:
构建版本格式
- 任务配置

保留

- 保留策略
- 保留配置

历史


- 配置版本

Xamarin Android的CI



选择内置的Xamarin Android模板

Select a template

 Search

Or start with an  Empty process

requires Maven to be installed on the build agent.



NodeJS With Grunt (PREVIEW)

Build NodeJS applications using Grunt task runner



NodeJS With Gulp (PREVIEW)

Build NodeJS applications using Gulp task runner



Xamarin.Android

Build an Android app and Xamarin.UITest assembly. Test with Xamarin Test Cloud.

Apply

默认编译步骤



~~Xamarin component restore~~ ~~***.sln~~

Disabled: Xamarin Component Restore



Use NuGet 4.3.0

PREVIEW

NuGet Tool Installer



NuGet restore

NuGet



Build Xamarin.Android Project ~~**/*Droid*.csp...~~

Xamarin.Android



Build solution ~~**/*test*.csproj~~

MSBuild



~~Test \$(build.binariesdirectory)/\$(BuildConfig...~~

Disabled: Xamarin Test Cloud



Signing and aligning APK file(s) \$(build.binari...

Android Signing



Publish Artifact: drop

Publish Build Artifacts

Demo : Android 编译定义设置



我的编译步骤



PowerShell Script: Set AssemblyInfo

PowerShell



PowerShell Script: Change Constant

PowerShell



NuGet restore ***.sln

NuGet



Build Xamarin.Android Project **/*Droid*.csp...

Xamarin.Android



Signing and aligning APK file(s) \$(build.binari...

Android Signing



Copy Apk Files to: \$(Build.ArtifactStagingDire...

Copy Files



Run copy: Create Deploy Apk

Command Line



FTP Upload: AppPackage

FTP Upload

PowerShell Script: Set AssemblyInfo

为什么会有这个步骤？

设置代码版本

设置编译版本

A screenshot of a PowerShell script named 'SetAssemblyInfo.ps1' in a Visual Studio Code editor. The script is written in a dark theme with syntax highlighting. It defines two parameters, \$codeVersion and \$buildNumber, and then uses the Get-Content, Replace, and Out-File cmdlets to update the AssemblyInfo.cs file in a project directory.

```
1 Param(  
2     [string]$codeVersion,  
3     [string]$buildNumber  
4 )  
5  
6 $content = Get-Content(".\src\PCLProject\Properties\AssemblyInfo.cs")  
7 $content = $content.Replace("__CODEVERSION__", $codeVersion)  
8 $content = $content.Replace("__BUILDNUMBER__", $buildNumber)  
9 $content | Out-File ".\src\PCLProject\Properties\AssemblyInfo.cs"
```

```
[AttributeUsage(AttributeTargets.Assembly)]
public class AssemblyCodeVersionAttribute : Attribute
{
    public AssemblyCodeVersionAttribute(string codeVersion)
    {
        CodeVersion = codeVersion;
    }

    public string CodeVersion { get; }

    public static string GetCodeVersion()
    {
        var assembly = typeof(AssemblyCodeVersionAttribute).GetTypeInfo().Assembly;
        var attribute =
CustomAttributeExtensions.GetCustomAttribute<AssemblyCodeVersionAttribute>(assembly);
        return attribute.CodeVersion;
    }
}
```

Colin's ALM Corner Build & Release Tools

- Version Assemblies
- Replace Tokens
- Tokenizer
- Tag Build / Release



The image shows a dark-themed extension card for 'Colin's ALM Corner Build & Release Tools'. On the left is a circular profile picture of a man in a green and white jersey. To the right, the title 'Colin's ALM Corner Build & Release Tools' is displayed in white. Below the title, it says 'Colin's ALM Corner' followed by a download icon, '2,972 installs', five yellow stars, and '(19)'. A line of text reads 'Build extensions from Colin's ALM Corner to make life easier!'. At the bottom are two green buttons: 'Install' and 'Download'. A footer note states: 'Click **Install** for Visual Studio Team Services and **Download** for Team Foundation Server.'

Colin's ALM Corner Build & Release Tools
Colin's ALM Corner | 📄 2,972 installs | ★★★★★ (19)

Build extensions from Colin's ALM Corner to make life easier!

[Install](#) [Download](#)

Click **Install** for Visual Studio Team Services and **Download** for Team Foundation Server.


Xamarin iOS的CI



选择内置的Xamarin iOS模板

Select a template

Or start with an  **Empty process**

 Search

requires Maven to be installed on the build agent.



NodeJS With Grunt (PREVIEW)

Build NodeJS applications using Grunt task runner



NodeJS With Gulp (PREVIEW)

Build NodeJS applications using Gulp task runner



Xamarin.Android

Build an Android app and Xamarin.UITest assembly. Test with Xamarin Test Cloud.



Xamarin.iOS

Build a Xamarin.iOS app and Xamarin.UITest assembly. Test with Xamarin Test Cloud. This template requires a Mac OS build agent.

Apply

默认编译步骤



~~Xamarin component restore **/*.sln~~

Disabled: Xamarin Component Restore



Build Xamarin.iOS solution **/*.sln

Xamarin.iOS



~~Test **/*.ipa with Xamarin.UITest in Xamarin T...~~

Disabled: Xamarin Test Cloud



Copy Files to: \$(build.artifactstagingdirectory)

Copy Files



Publish Artifact: drop

Publish Build Artifacts

Demo : iOS编译 定义配置

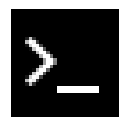


我的编译步骤



Run PowerShell script: Set AssemblyInfo

Command Line



Run PowerShell script: Change Constant

Command Line



NuGet restore ***.sln

NuGet



Build Xamarin.iOS solution

Xamarin.iOS



Copy Ipa Files to: \$(Build.ArtifactStagingDirec...

Copy Files



Run cp: Create Deploy Ipa

Command Line



FTP Upload: AppPackage

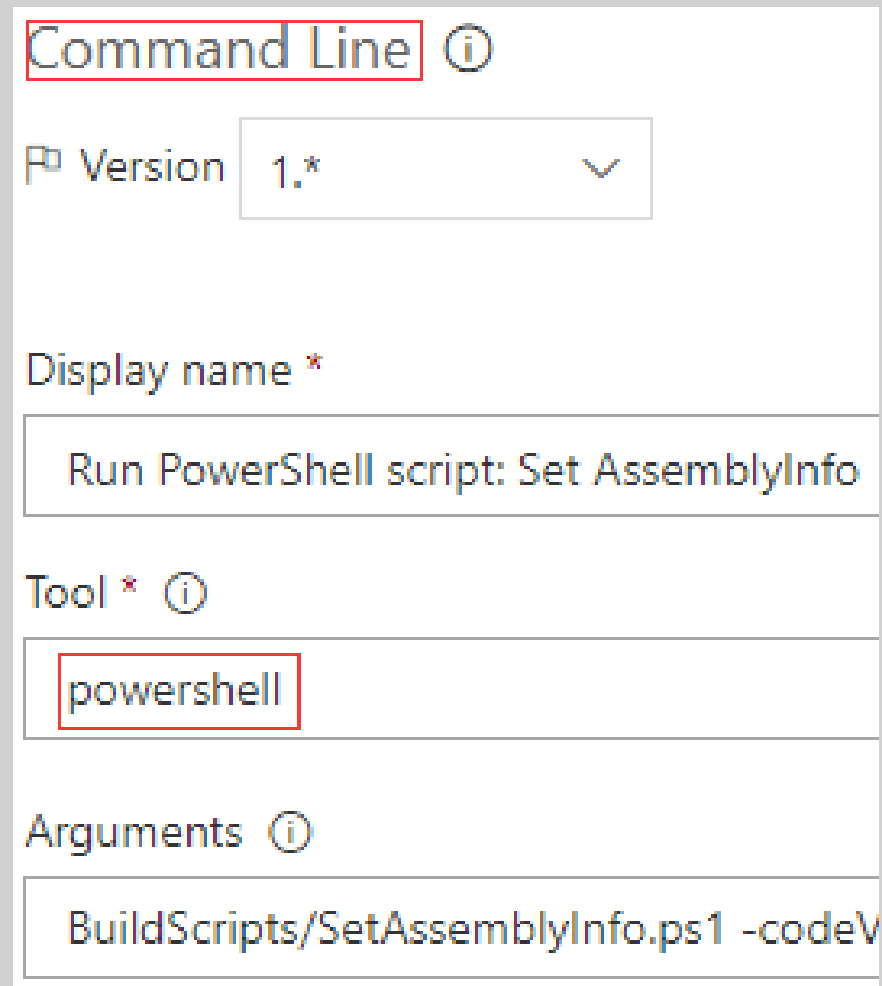
FTP Upload

如何写一份脚本让两个平台的CI共享

诀窍就在开源跨平台的
PowerShell 6.0

在macOS上的运行技巧：

- Command Line
- powershell



The image shows a configuration form for a CI job. The 'Command Line' field is highlighted with a red box and contains an information icon. Below it is a 'Version' dropdown menu set to '1.*'. The 'Display name' field contains the text 'Run PowerShell script: Set AssemblyInfo'. The 'Tool' field is highlighted with a red box and contains the text 'powershell', with an information icon to its right. The 'Arguments' field contains the text 'BuildScripts/SetAssemblyInfo.ps1 -codeV'.

Command Line ⓘ

Version 1.* ▼

Display name *

Run PowerShell script: Set AssemblyInfo

Tool * ⓘ

powershell

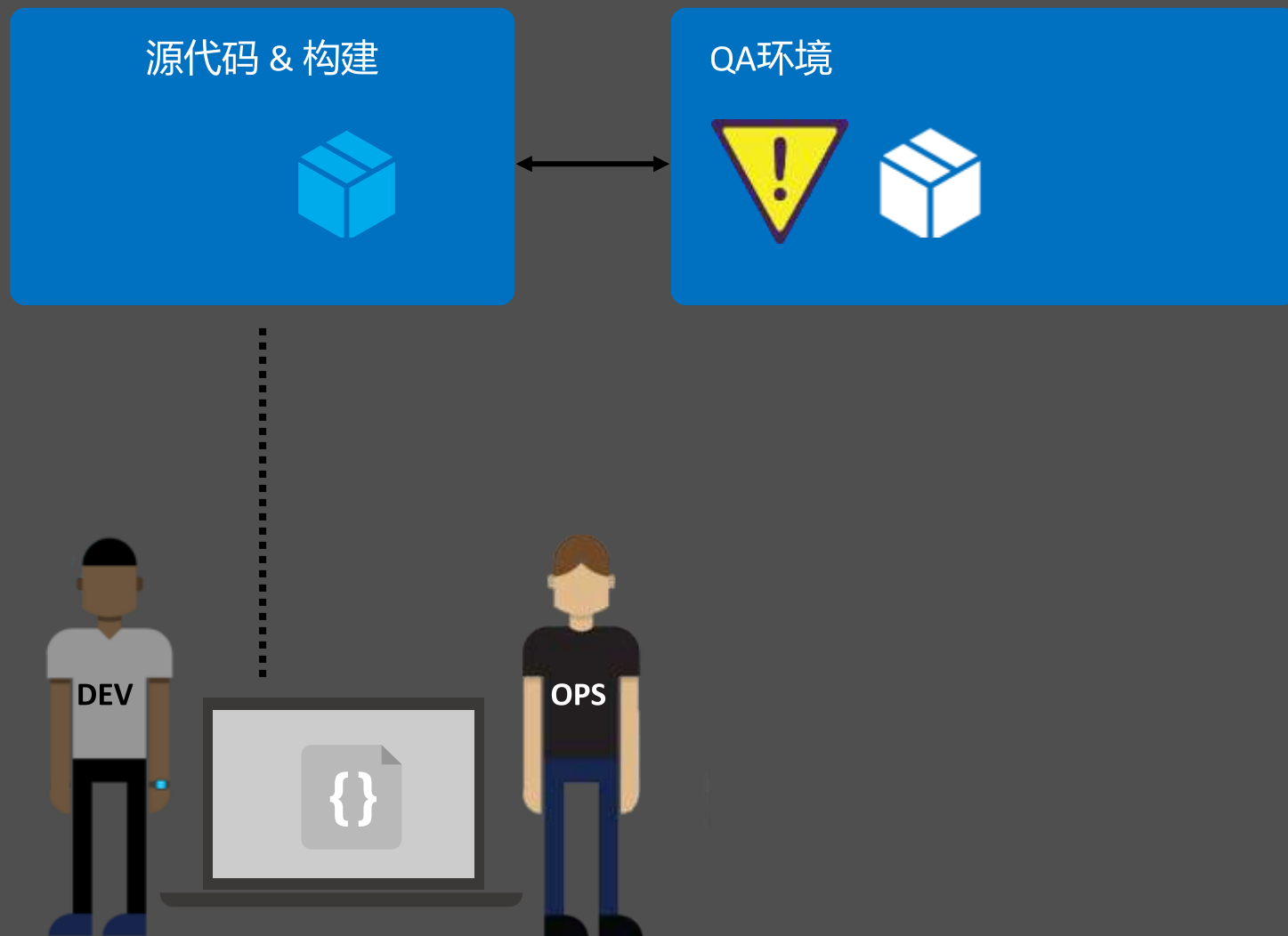
Arguments ⓘ

BuildScripts/SetAssemblyInfo.ps1 -codeV

持续交付 与 持续部署



持续交付



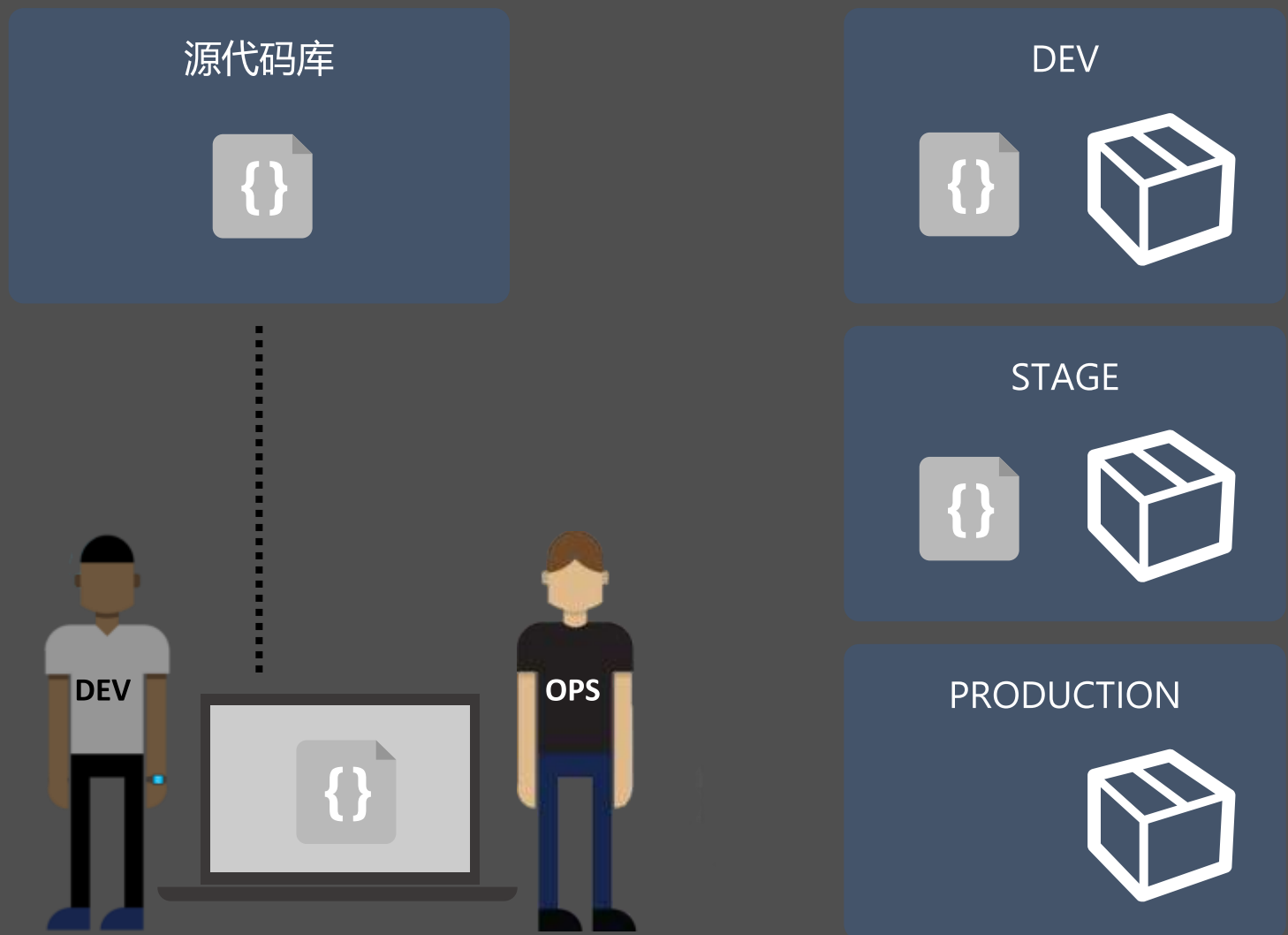
价值

- 优化资源
- 加速交付

度量

- 部署频率
- MTTR
- 可用性

持续部署



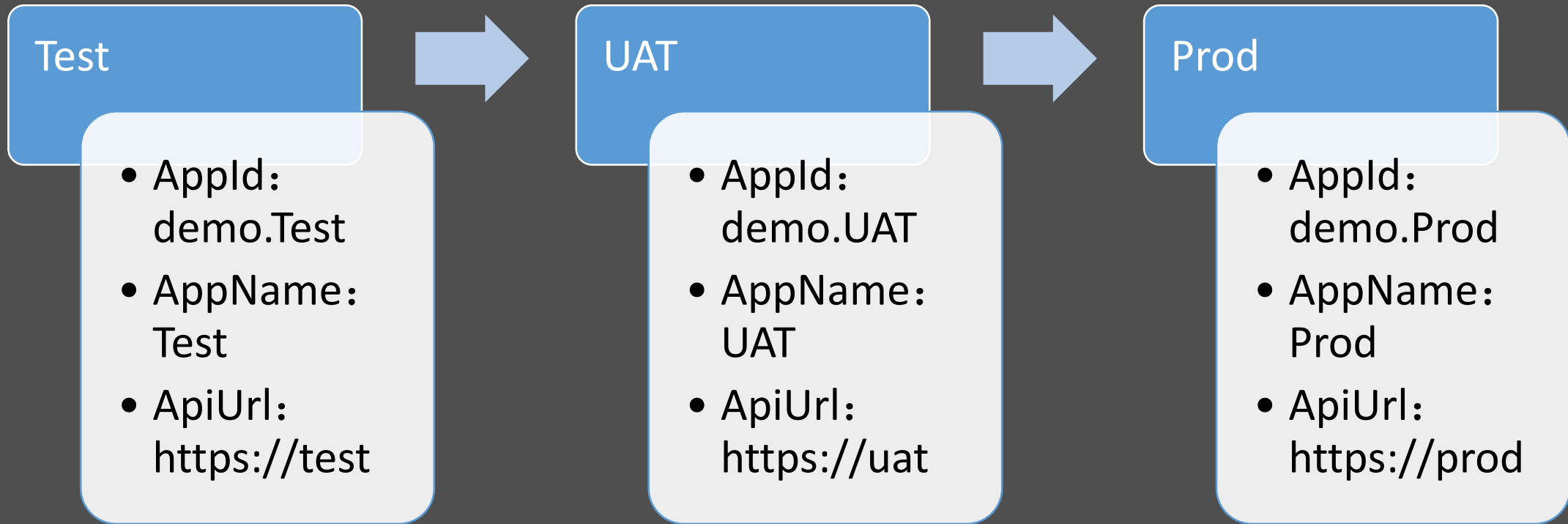
价值

- 优化资源
- 加速交付

度量

- 部署频率
- MTTR
- 可用性

APP的特殊要求



应对方法

采用持续交付而非持续部署

因为我们无法把一个Artifact（即APK或IPA）同时用于多个环境

多分支驱动的多环境APP持续交付

Test

- 分支: master
- 部署到Test分发站点

UAT

- 分支: uat
- 部署到UAT分发站点

Prod

- 分支: release
- 部署到Prod分发站点

在CI中添加特定编译步骤

- Android : ChangeAndroidManifest.ps1
- iOS : ChangeInfoPlist.sh
- Change Constant : -buildSymbol UAT
- FTP Upload : ftp://uat/

ChangeAndroidManifest.ps1

```
Param(  
    [string]$appId,  
    [string]$appName  
)
```

```
$xml = [xml](Get-Content -Encoding UTF8 ".\src\PCLProject\Properties\AndroidManifest.xml")
```

```
$xml.manifest.package = $appId
```

```
$xml.manifest.application.SetAttribute("label", "http://schemas.android.com/apk/res/android",  
$appName)
```

```
$xml.Save(".\src\PCLProject\Properties\AndroidManifest.xml")
```

ChangeInfoPlist.sh

```
plutil -replace CFBundleIdentifier -string $1 "../src/iOSProject/Info.plist"  
plutil -replace CFBundleDisplayName -string $2 "../src/iOSProject/Info.plist"
```

Demo : CI/CD应用安装包



持续测试



Visual Studio Mobile Center

Mission control for mobile apps



Build



Test



Analytics



Push Notifications



Distribute



A/B Testing



Crash Reporting



Remote Config.



Identity



Tables



Live Update

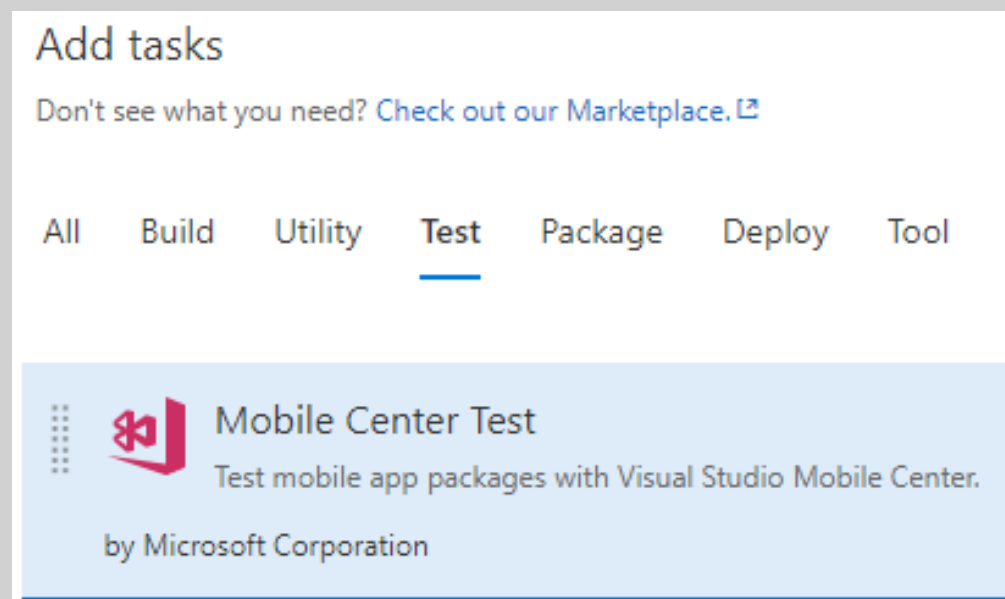


Storage



在CI的时候添加测试步骤

- 建议使用Mobile Center进行测试，基于Xamarin UI Test框架
 - 为不同环境创建不同的APP
- 当然也可以使用自己熟悉的测试服务（编写自定义插件）



持续监控



利用Mobile Center收集崩溃和使用日志

- 分别为不同平台不同环境创建APP，
得到： $2 \times 3 = 6$ 个AppKey
- 把AppKey保存为常量，通过编译条件
切换不同环境
- 在CI的Change Constant步骤中设置
编译条件

Mobile Center的SDK配置

- 添加Nuget :
 - Microsoft.Azure.Mobile.Analytics
 - Microsoft.Azure.Mobile.Crashes
- 在App.xaml.cs的构造器中启动SDK :

```
MobileCenter.Start("android={Your Android App secret here};" +  
    "uwp={Your UWP App secret here};" +  
    "ios={Your iOS App secret here}",  
    typeof(Analytics), typeof(Crashes));
```

其他问题与考虑

- 对安装包进行签名、重命名
 - 多环境的iOS签名MobileProvision
- 分发的时候更新版本，实现In-App Update
- iOS的CI的时候上传Symbol文件

演示用源代码地址

- 移动端：
<https://github.com/heavenwing/XamarinDevOpsMobile>
- 分发网站：
<https://github.com/heavenwing/XamarinDevOpsWeb>

Q&A

