

Continuous Delivery for iOS, Android & Windows Apps using Xamarin, VSTS & HockeyApp

Nish Anil
Senior PM – Xamarin @ Microsoft
@nishanil

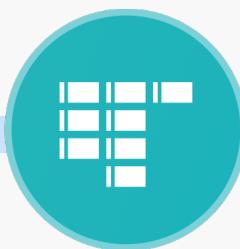
Microsoft – Your Complete Mobile Solution



Develop



Test



Build



Distribute



Monitor

Xamarin test cloud › MonkeyChat › master › May 24, 2016 7:37 PM New Test Run Support Docs James ▾

RECODER TEST
ScrollToEvent[AppView: Class=Xamarin.TestRec...]

Overview Filter devices

ALL RESULTS

Recorder Test

New Test 8

Tapped on view with class: AppCompatButton	6
Tapped on view with class: EditText	1
Tapped on view with class: AppCompatButton	1
Tapped on view with class:	1

HTC One A9 Huawei Nexus 6P LG Nexus 5X Samsung Galaxy S6 HTC One M9 Samsung Galaxy S6 Ed... Samsung Galaxy S5 Samsung Galaxy Note ...

Android 6.0.1 Android 6.0.1 Android 6.0.1 Android 5.1.1 Android 5.0.2 Android 5.0.2 Android 5.0 Android 4.3

The screenshot shows the Xamarin Test Cloud interface. At the top, it displays the navigation path: Xamarin test cloud > MonkeyChat > master > May 24, 2016 7:37 PM. Below this is a search bar and a 'New Test Run' button. The main area is titled 'RECODER TEST' with the subtitle 'ScrollToEvent[AppView: Class=Xamarin.TestRec...]' in red. On the left, there's an 'Overview' button and a 'Filter devices' dropdown. The results section is titled 'ALL RESULTS' and shows a table for the 'Recorder Test'. The table has two columns: 'Action' and 'Count'. The first row shows 'Tapped on view with class: AppCompatButton' with a count of 6 (red icon). The second row shows 'Tapped on view with class: EditText' with a count of 1 (green icon). The third row shows 'Tapped on view with class: AppCompatButton' with a count of 1 (green icon). The fourth row shows 'Tapped on view with class: ' with a count of 1 (red icon). To the right of the table, there's a list of devices with their names and Android versions: HTC One A9 (Android 6.0.1), Huawei Nexus 6P (Android 6.0.1), LG Nexus 5X (Android 6.0.1), Samsung Galaxy S6 (Android 5.1.1), HTC One M9 (Android 5.0.2), Samsung Galaxy S6 Ed... (Android 5.0.2), Samsung Galaxy S5 (Android 5.0), and Samsung Galaxy Note ... (Android 4.3). Each device is represented by a small image of a smartphone.



Test

Challenges

- Device Fragmentation
- App Complexity
- Fast Release Cycles
- Short Sessions
- High Mobile User Expectations

On Mobile, Quality is Hard

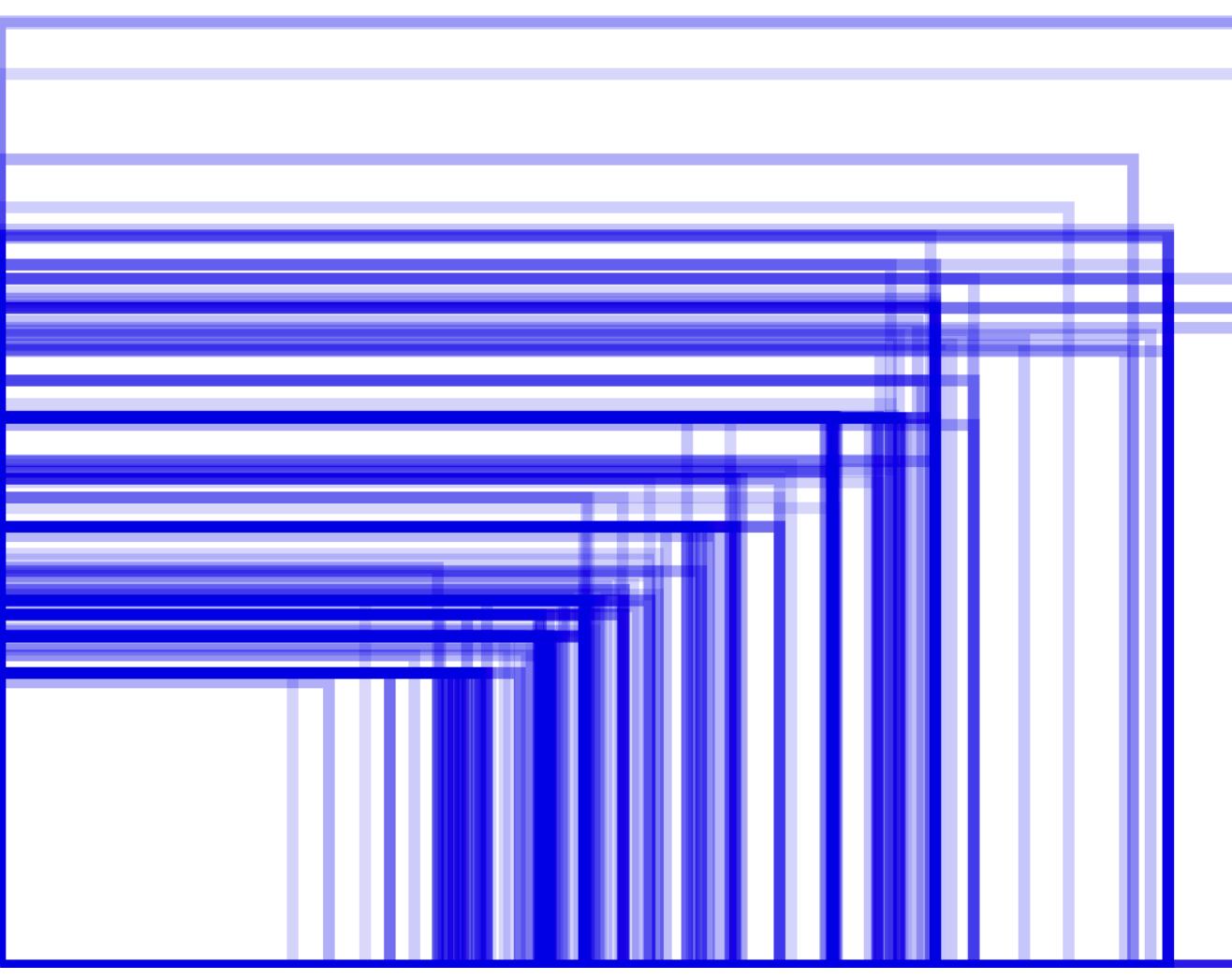


{ 5 OS versions
20 Devices
20 Languages
35 Locales
6 Screen sizes



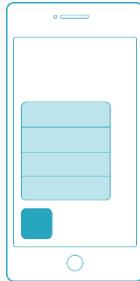
{ 9 OS versions
24K+ Distinct devices
39 Languages
57 Locales
27 Screen sizes
1,294 Brands
6 Screen configurations



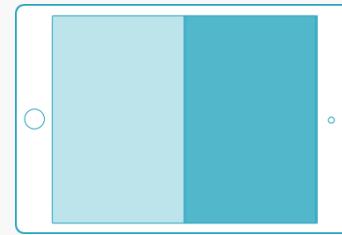
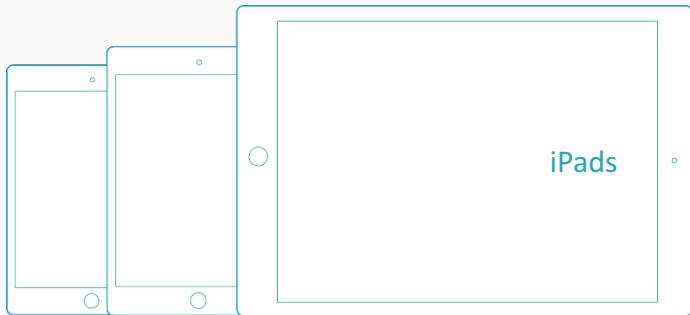


OpenSignal

iOS Fragmentation: It does exist!



Apple pencil
iPad Pro only



Multi Tasking
Only select iPads

Form factors

iOS 9 features



Tap



Scroll



Swipe



Pinch



Multi Finger



Text Entry



Rotation



GPS



Language

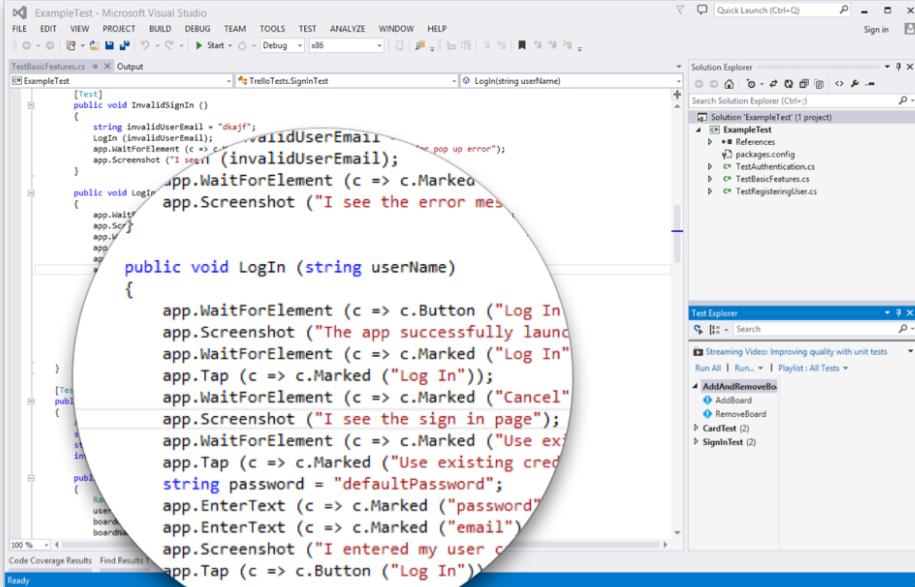


Currency



Network Condition

Introducing Xamarin.UITest



Create Automated User Interface tests all in C#

Run directly from Visual Studio or Xamarin Studio

FREE to use on a Simulator or Device

Works on ANY app:

- Native
- Hybrid
- Xamarin

Every App, Every Language

- Native Objective-C, Swift, Java
- Native C# with Xamarin
- Hybrid: PhoneGap, Cordova, etc
- Anything Else!

Let's build our first Xamarin.UI Test!

Test Recorder for Mac

- Record UITests for iOS and Android apps
- Replay or ship to Test Cloud

The image shows the Xamarin Test Recorder application running on a Mac. On the left, the recorder interface displays a sequence of recorded interactions:

- Record button (red)
- Run button (play)
- Stop button (black square)
- Open dropdown: Device: Nexus 5 (Lollipop) - 5.1, App: Simple Credit Car...
- Tap on EditText [creditCardNumberText]
- Type text: 123456789012345
- Tap on Button "Validate Credit Card"
- Tap on TextView "Credit card number..."

The recorded code in the editor pane is:

```
public void NewTest ()  
{  
  
    app.Tap(x => x.Class("EditText").Id("creditCardNumberText"));  
  
    app.EnterText(x => x.Class("EditText").Id("creditCardNumberText"),  
    "123456789012345");  
  
    app.Tap(x => x.Class("Button").Id("validateButton").Text("Validate  
    Credit Card"));  
  
    app.Tap(x =>  
    x.Class("TextView").Id("errorMessagesText").Text("Credit card number  
    is too short."));  
}
```

On the right, the Xamarin Android Player shows the "Simple Credit Card" app running on a Nexus 5 (Lollipop). The screen displays:

Enter Credit Card Number

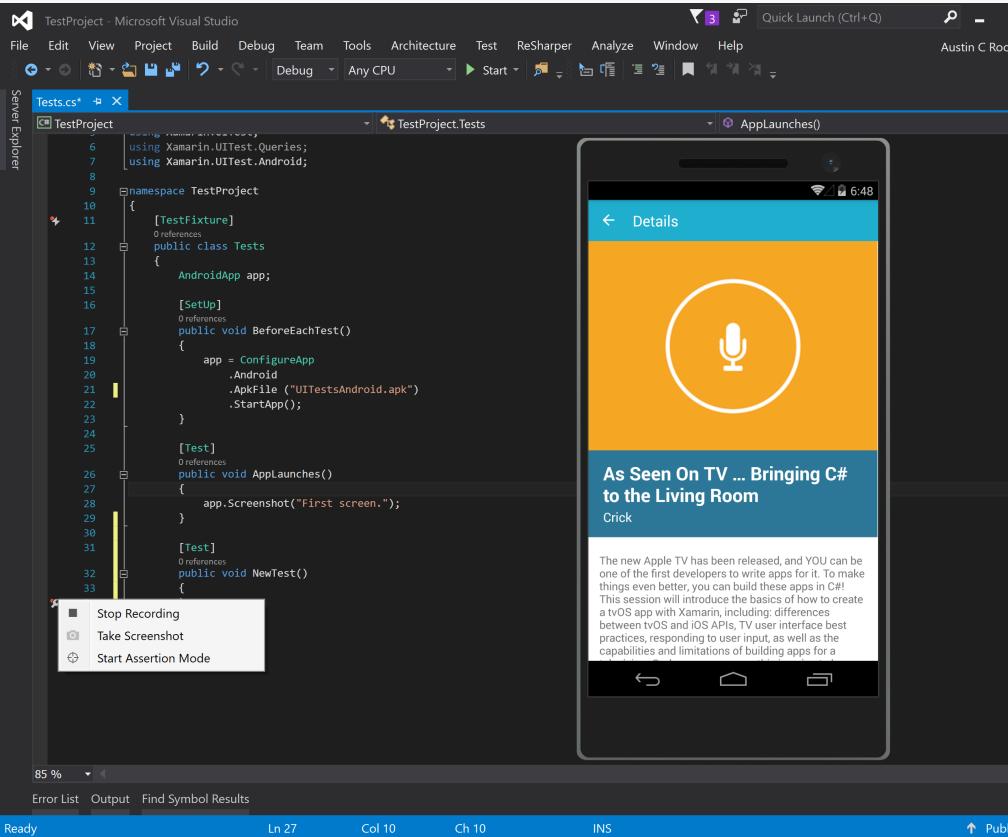
123456789012345

Validate Credit Card

Credit card number is too short.

Test Recorder for Visual Studio

- Record UITests from Visual Studio
- Replay or ship to Test Cloud
- Use Touch Screen on Android Emulator





Xamarin Test Cloud

xamarin.com/testcloud

Screenshot of the Xamarin Test Cloud interface showing a grid of mobile device screenshots and a sidebar with test results.

The sidebar on the left lists test results under "Customers tests" and "Sales tests".

- Customers tests:**
 - Check Customer Details (✓)
 - Check Customer Navigation (✓)
 - Check Customer Phone (✓)
 - Investigate Customer Page (✓)
- Sales tests:**
 - Add an item (⚡)

A detailed view of the "Add an item" test is shown in a modal window:

Then I tap 'Sales'

- First I launch the app
- Then I tap 'Sales'
- Then I tap 'Add'
- Then I choose the first result
- Then I set the title and description
- Then I tap 'Save'
- Then I go back

The main area displays a grid of 21 mobile device screenshots, each showing the "Sales" screen of the Xamarin CRM application. The devices listed are:

- Apple iPhone 5C (iOS 8.2)
- Apple iPhone 5 (iOS 8.3)
- Apple iPhone 5C (iOS 8.3)
- Apple iPhone 5S (iOS 8.3)
- Apple iPhone 6 (iOS 8.3)
- Apple iPhone 6 Plus (iOS 8.3)
- Apple iPhone 5S (iOS 8.2)
- Apple iPhone 6 (iOS 8.2)
- Apple iPhone 5S (iOS 8.1.3)
- Apple iPhone 6 (iOS 8.1.3)
- Apple iPhone 5 (iOS 7.4)
- Apple iPhone 5C (iOS 7.4)
- Apple iPhone 5S (iOS 7.4)
- Apple iPhone 6 (iOS 7.4)
- Apple iPhone 5S (iOS 7.4)
- Apple iPhone 6 (iOS 7.4)
- Apple iPhone 5 (iOS 7.4)
- Apple iPhone 5S (iOS 7.4)
- Apple iPhone 6 (iOS 7.4)
- Apple iPhone 5S (iOS 7.4)
- Apple iPhone 6 (iOS 7.4)

Devices on Demand

Which devices do you want to run?

Top 20 Top 10 Tablets Top 10 Phones Select None X

Sort By

Market share

Name

OS version

Form Factor

Phone

Tablet

Manufacturer

Acer

Amazon

Asus

HP

90% worldwide market share

Samsung Galaxy S III
Android 4.3
Market share: 2.492%

Samsung Galaxy S4
Android 4.4.2
Market share: 2.310%

Samsung Galaxy Y
Android 2.3.6
Market share: 1.409%

Samsung Galaxy S III Mini
Android 4.1.2
Market share: 1.324%

Samsung Galaxy S4
Android 4.3
Market share: 1.132%

Samsung Galaxy Note II
Android 4.3
Market share: 1.005%

LG Nexus 5
Android 4.4.2
Market share: 0.980%

Samsung Galaxy S II
Android 4.1.2
Market share: 0.954%

Samsung Galaxy S III
Android 4.1.2
Market share: 0.924%

Samsung Galaxy S Duos
Android 4.0.4
Market share: 0.834%

Samsung Galaxy Core
Android 4.1.2
Market share: 0.827%

Samsung Galaxy Grand Duos
Android 4.2.2
Market share: 0.824%

Back Select 20 devices

Multiple Supported Frameworks

Xamarin.UITest

- C#
- Run with Nunit
- XS & VS
- SpecFlow support

Calabash

- Ruby
- BDD, run with cucumber
- RubyMine or CLI

Appium

- Selenium or Webdriver
- Java, Python, JS

To The Test Cloud



Build

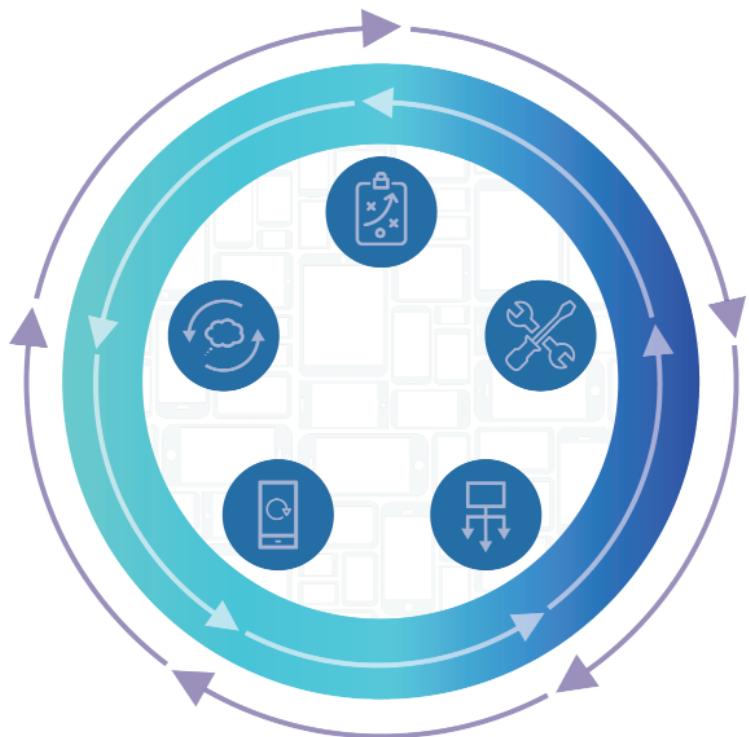


Distribute



Monitor

What is DevOps ?



“DevOps is the union of people, process,
and products to enable continuous
delivery of value to our end users.”

- Donovan Brown

Senior DevOps Program Manager,

Microsoft

Mobile DevOps

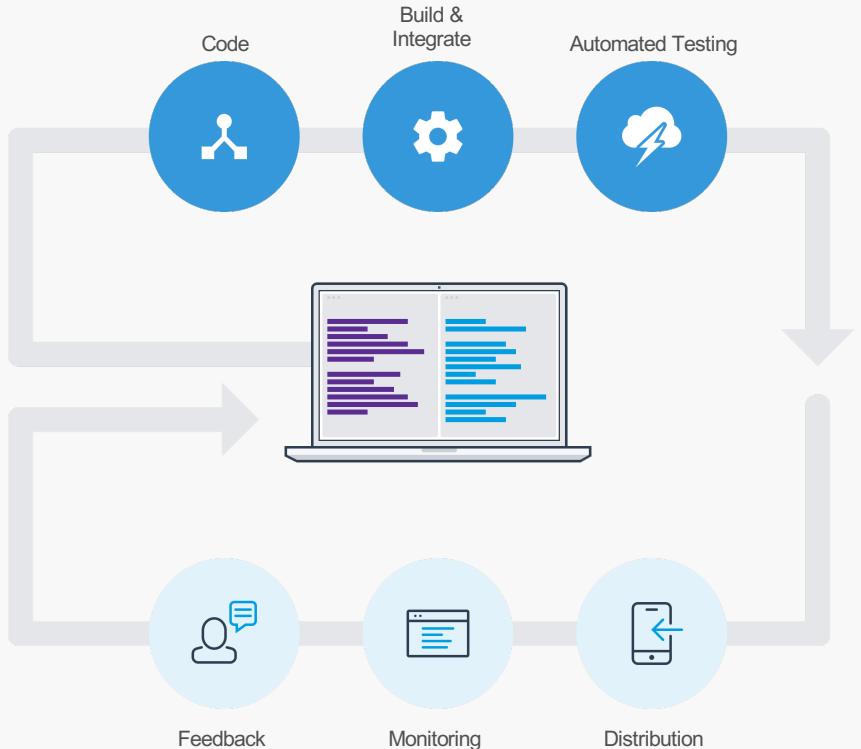
“Mobile” DevOps

■ Streamline

- Frequently releasing value to users
- Constantly maintaining quality
- Monitoring app health & engagement in real-time

■ Automating the process

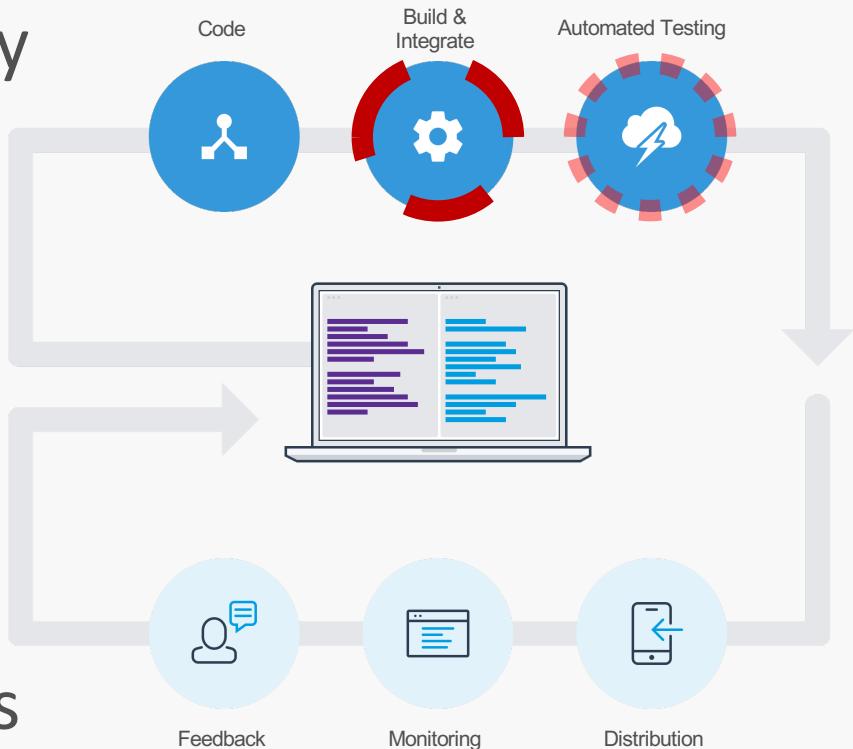
- Continuous Integration (CI)
- Continuous Deployment (CD)
- Continuous Monitoring (CM)



Continuous Integration

Continuous Integration

- Every Commit...Continuously
- Build
- Integrate
 - Version Bumping
 - Replacing Private Variables
 - Testing
 - Unit Tests
 - UI Tests
 - Other SDKs
- Sign, Package, Drop Artifacts



Visual Studio Team Services (VSTS) / TFS

Version Control

- [Git and Team Foundation Version Control \(TFVC\)](#)
- [Unlimited private repos](#)

Agile Planning

- [Work item tracking](#)
- [Kanban, backlogs, task boards](#)

Continuous Integration

- [Cloud-hosted build agents](#)
- [Build support for all mobile platforms](#)
- [Integration with test clouds](#)

Pull Request 4: Merge users/raisa/instructors_new to master

Jamal Hartnett commented on the file [InstructorController.cs](#)

//TODO, implement support for null courses
if (courseID != null)
{
 ViewBag.CourseID = courseID.Value;
 // Lazy loading
 //viewModel.Enrollments = viewModel.Courses.Where(
 // x => x.CourseID == courseID).Single().Enrollments;

Why didn't you enable the lazy loading?
Jamal Hartnett - 2 weeks ago

I haven't had a chance to do the perf tuning yet. I want to get this unblocked for the upcoming release, then we can implement lazy loading.

Status: Active

Master.CI / Build 20150216.1

Build 20150216.1
Ran for 16 seconds (default), completed 4 days ago

Build details

Definition	Master.CI (edit)
Source branch	refs/pull/31/merge
Source version	0945638f1c78656a28ea5a82b95585def9a7cdcb
Started	Mon Feb 16 2015 12:50:31 GMT-0500 (Eastern Standard Time)
Finished	Mon Feb 16 2015 12:50:47 GMT-0500 (Eastern Standard Time)

Visual Studio Team Services - Build

Team Services / MyDriving

James Montemagno | 🌐 🛡️ 🚀 🤖 ?

HOME CODE WORK **BUILD** TEST RELEASE

Search work items

Explorer

Definitions / MyDriving.Xamarin.Android-Feature | Builds

Build Options Repository ! Variables Triggers General Retention History

Save Queue build... Undo

+ Add build step...

- Replace tokens for BingMaps **Replace Tokens**
- NuGet restore src/MobileApps/MyDriving.XS.sln **NuGet Installer**
- Update Version Name **Version Assemblies**
- Update Version Code **Version Assemblies**
- Download keystore **Command Line**
- Activate Xamarin license **Xamarin License**
- Build and Sign Android Project **Xamarin.Android**
- Deactivate Xamarin license **Xamarin License**
- Build tests **MSBuild**
- Test in Xamarin Test Cloud **Xamarin Test Cloud**

Copy Files to: \$(build.artifactstagingdirectory)
Copy Files

↑ Publish Artifact: drop **Publish Build Artifacts**

Replace tokens for BingMaps

Source Path: src/MobileApps/MyDriving/MyDriving.Utils/
Target File Pattern: Logger.cs

Advanced

Control Options

Enabled:
Continue on error:
Always run:

More Information

My favorites

Team favorites

- MyDriving.Xamarin.Android **Completed 5 hours ago**
- MyDriving.Xamarin.iOS **Completed 5 hours ago**
- MyDriving.Xamarin.UWP **Completed 5 hours ago**

All build definitions

- All build definitions
- MyDriving.Services
- MyDriving.Xamarin.Android
- **MyDriving.Xamarin.Android-Feature**
- MyDriving.Xamarin.Android-Regression
- MyDriving.Xamarin.iOS
- MyDriving.Xamarin.iOS-Evolve
- MyDriving.Xamarin.iOS-Feature
- MyDriving.Xamarin.iOS-Regression
- MyDriving.Xamarin.UWP
- MyDriving.Xamarin.UWP-Feature
- MyDriving.Xamarin.UWP-Regression

All XAML definitions

Demo – VSTS end to end

Continuous Deployment

Continuous Deployment – Alpha/Beta

- Development Builds
- Send to Testers
- Release to App Store
- Promote App Store Channel
 - Alpha -> Beta -> Production





HOCKEYAPP

HockeyApp

Distribution

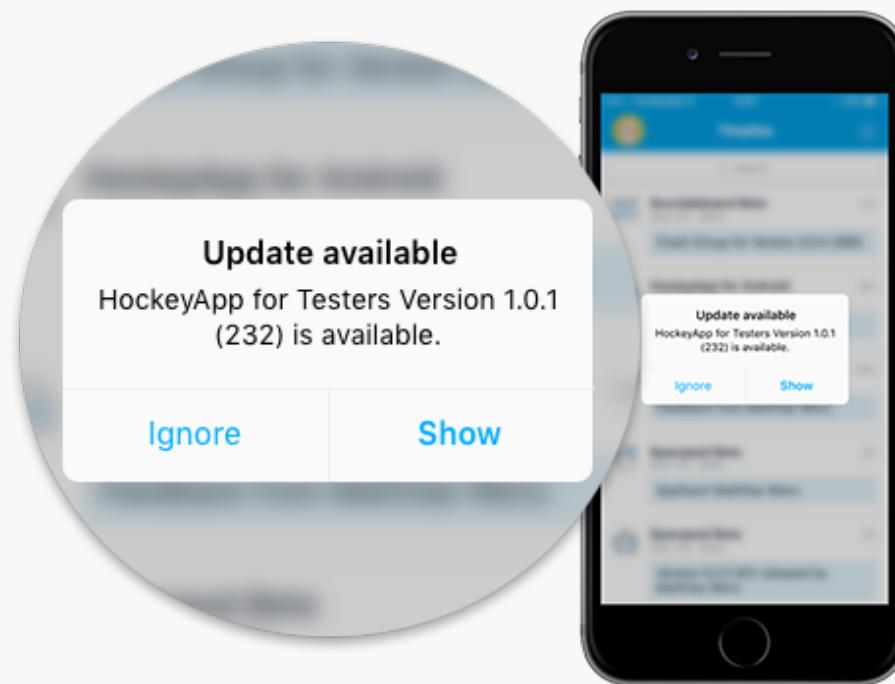
- In-house app store
- Notify testers of updates

Crash Reporting

- Real-time collection
- “Symbolicated” stacks

Feedback

- Gather in-app feedback
- Respond directly to testers

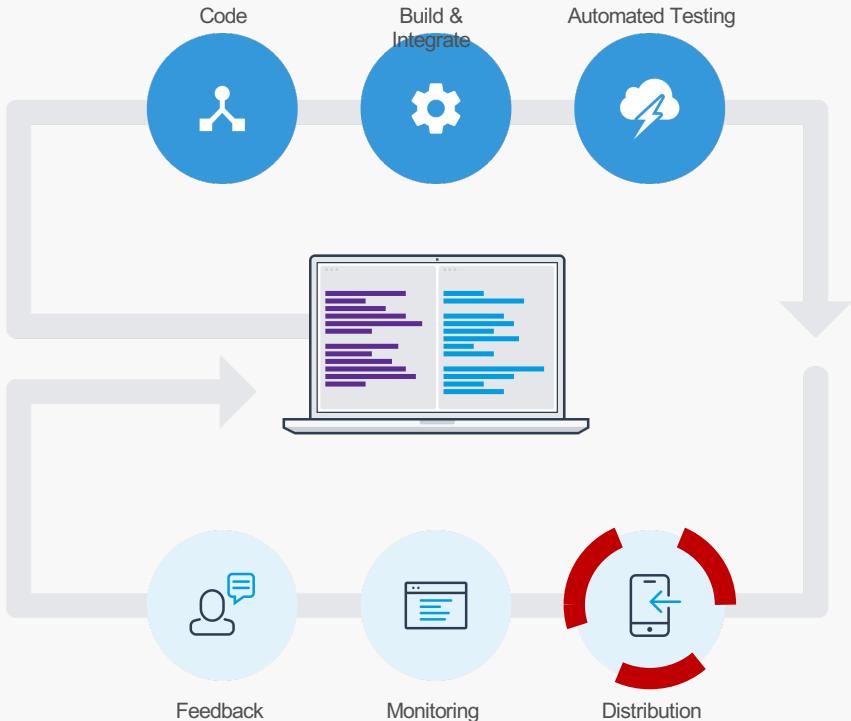


Continuous Deployment - Production

How can you automate your production deployments once a release is signed-off?

Recommendations

1. Setup a multi-environment release process (e.g. beta and production)
2. Assign the necessary stakeholders to sign-off on beta/staging builds
3. Automate as much of this workflow as possible



VSTS – Release Management

Continuous Delivery

Define per-environment release pipeline

Trigger manually or via successful builds

Approval Policies

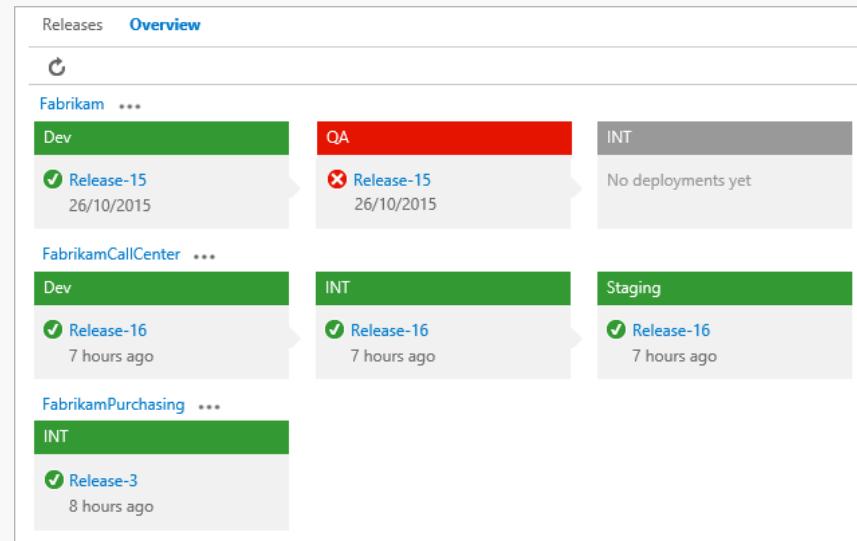
Configure per-environment sign-offs

Manual validation requirements

Release Visibility

View per-environment release status

Track release back to commits



Demo – HockeyApp



=>
Get Started Today
testcloud.xamarin.com
visualstudio.com
hockeyapp.net

More!



aka.ms/vsandxamarin



aka.ms/gettingstartedwithxamarin



github.com/xamarin



twitter.com/msdevindia



facebook.com/MicrosoftDeveloper.India/



Booth # 10

Thank you. Questions?

Nish Anil
Senior PM– Xamarin @ Microsoft

nish@microsoft.com

@nishanil