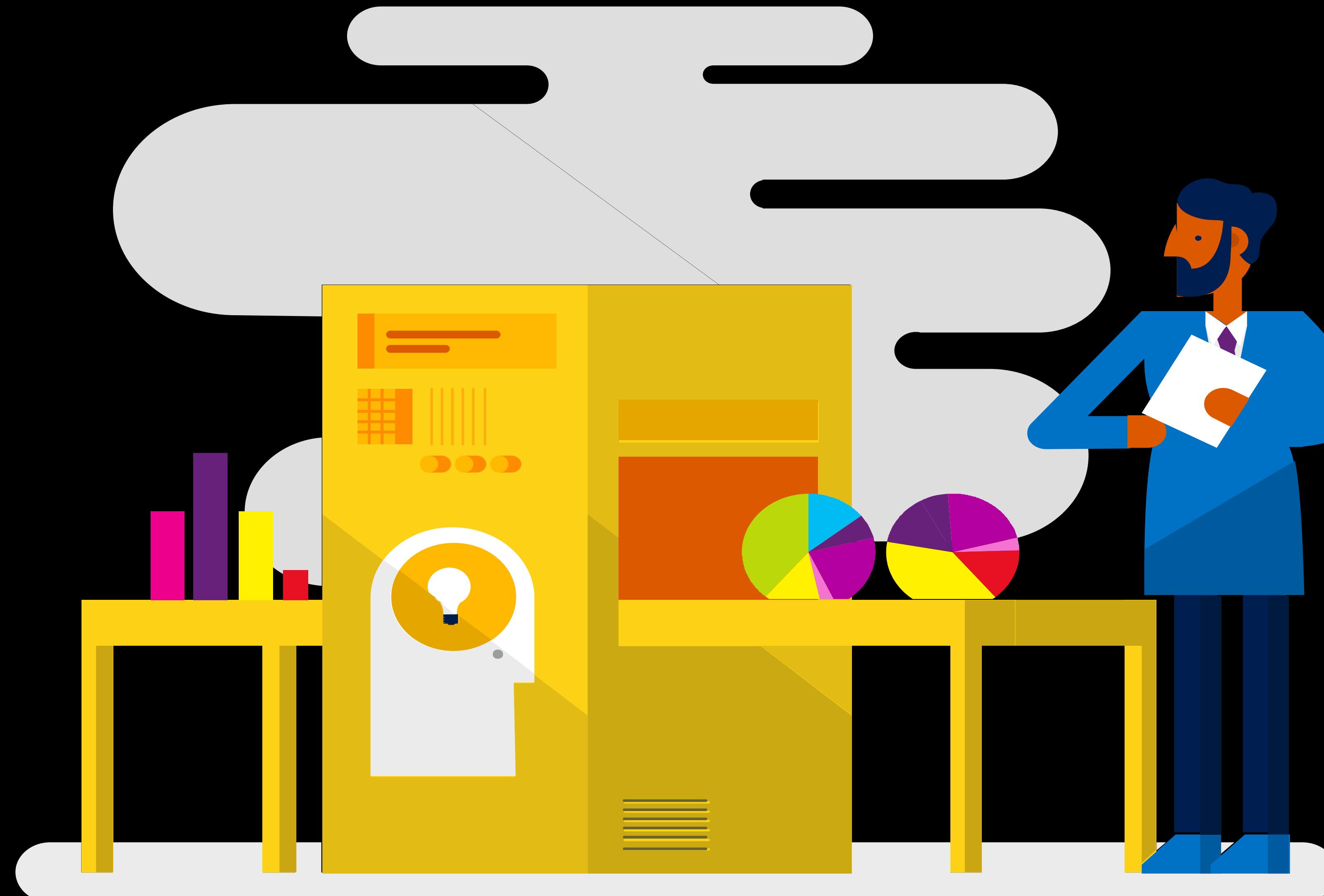


AUTOMATE THE LIFECYCLE OF YOUR APPS WITH



VISUAL STUDIO APP CENTER

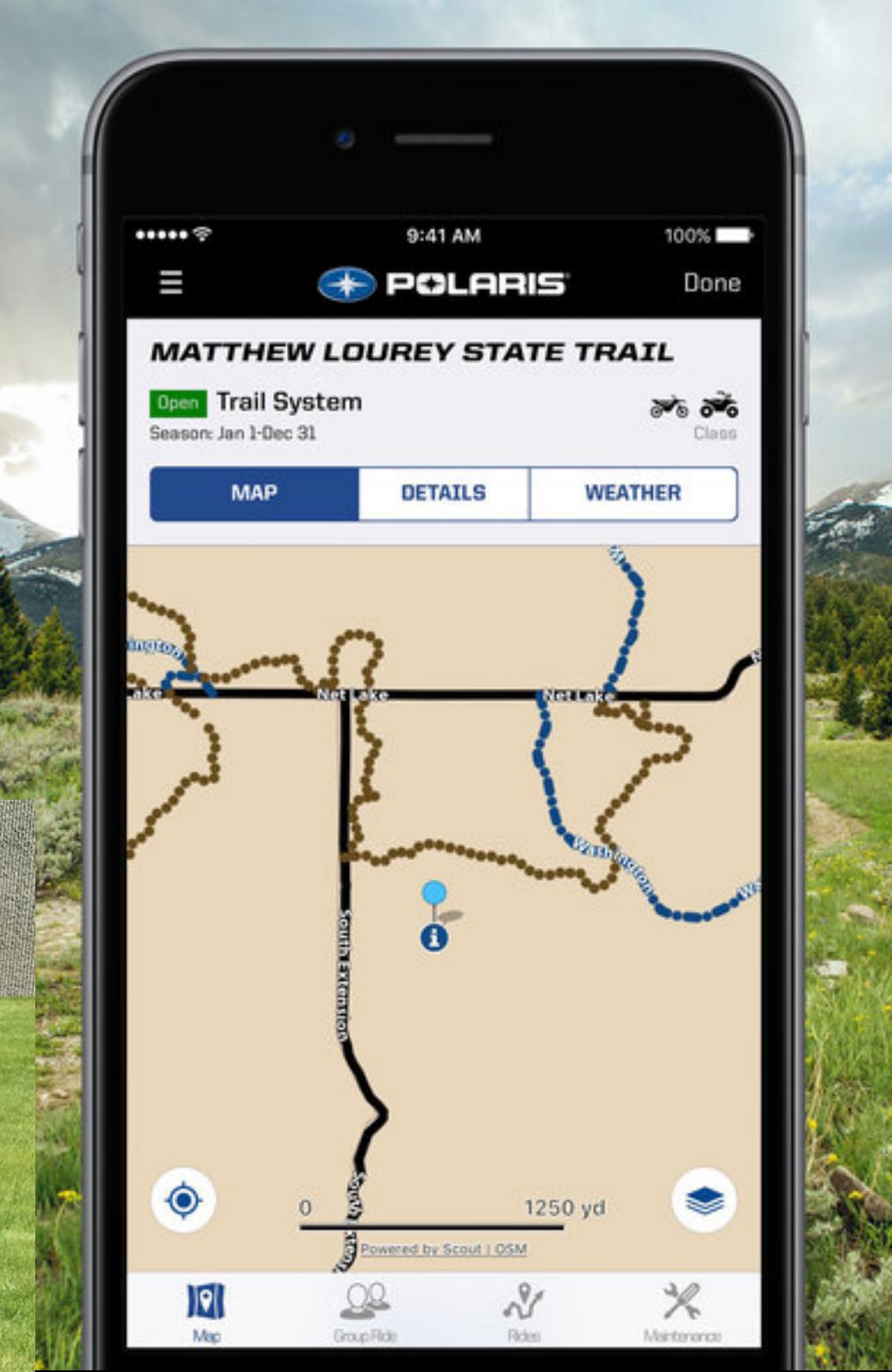
ABOUT ME

Sr. Mobile Developer @ Polaris Industries, Ride Command Software

Xamarin.Forms enthusiast, DevOps hobbyist, Machine learning beginner,
4 year XCMD

Blog: <https://tomsoderling.github.io>

Twitter: @**tom**soderling



XAMARIN UNIVERSITY

New
Aug 28th 2017



Unlimited training, 1-on-1 office hours

FREE!



Xamarin Certified Mobile Professional

To achieve the Mobile Professional certification, complete the list of required courses in green below and pass the Professional Certification exam.

XAM101 - Getting Started with Xamarin

AND101 - Introduction to Xamarin.Android

IOS102 - Introduction to the Xamarin Designer for iOS

AND102 - Activities and Intents

IOS101 - Introduction to Xamarin.iOS

XAM110 - Introduction to Cross-Platform Mobile Development

XAM150 - Consuming REST-based Web Services

XAM160 - SQLite and Mobile Data

XAM220 - Preparing for Publishing

XAM120 - Introduction to Xamarin.Forms

XAM130 - XAML in Xamarin.Forms

XAM140 - Resources and Styles in Xamarin.Forms

XAM135 - Layout in Xamarin.Forms

AND110 - ListViews and Adapters in Android

XAM250 - Patterns for Cross Platform Mobile Development

IOS110 - Fundamentals of TableViews

XAM270 - Data Binding in Xamarin.Forms

XAM280 - Using ListView in Xamarin.Forms

IOS115 - Customizing TableViews

AND205 - Android Navigation

IOS205 - Navigation Patterns

XAM370 - Diagnosing Memory Management Issues

XAM330 - Xamarin.Forms Effects

AND180 - Toolbar and App Bar

XAM301 - Mobile Application Architecture

Xamarin Certified Mobile Developer

To achieve the Mobile Developer certification, complete all of the required courses listed above and pass both the Professional Certification and Developer Certification exams. Note: You must be a Xamarin Certified Mobile Professional prior to gaining eligibility to take the Developer Certification Exam.



THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- Support
- Resources

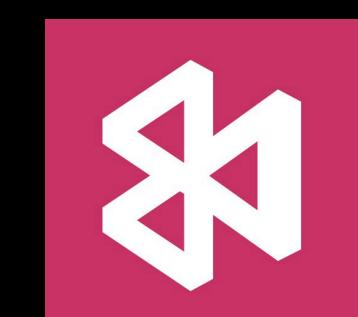
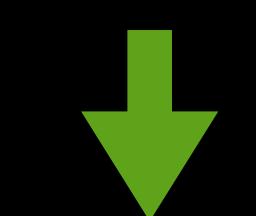
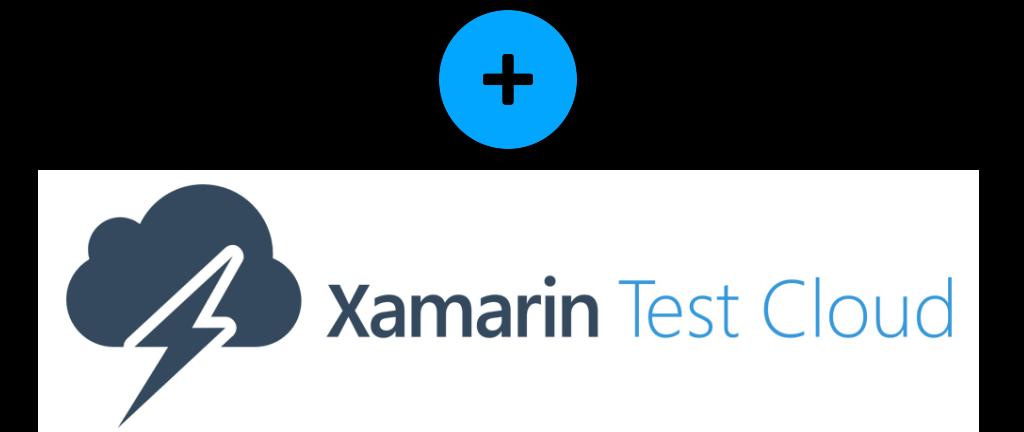
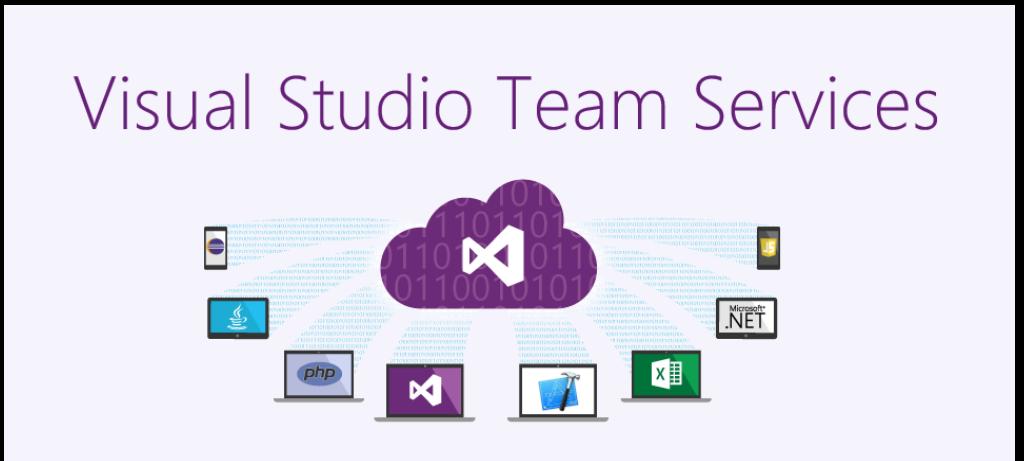


Use Visual Studio App Center to release **more frequently**,
with higher quality, and **have more fun** by spending
time on coding features instead of managing overhead

INTRODUCTION

- Continuous integration, delivery, and learning solution for Android, iOS, Windows, and macOS apps
- First went into public preview at Connect(); 2016 as Visual Studio **Mobile** Center
- General availability* at Connect(); 2017 (Nov. 15) as Visual Studio **App** Center
 - * Push notifications still in preview

PSA: fully retired on March 31st 2018



INTRODUCTION

- Pricing
- Distribute, Analytics, and Crashes are completely free
- Build and Push offer generous free tiers
- Test has an unlimited 30-day free trial

free tier

free trial

free

free

free

free tier

Build Build Objective-C, Swift, Java, Xamarin (C#), and React Native in the cloud		240 build minutes per month Up to 30-min per build
Test Automate UI tests on thousands of real devices		Free 30 day trial
Distribute Send your apps to your beta testers and app stores instantly		Unlimited distributions Unlimited users
Analytics Gain insights into your audience and app usage		All features included
Crash reporting Monitor your app health with real-time crash reports		All features included
Push notifications Engage your users by sending targeted messages		Up to 5 audience segments

THE PLAN

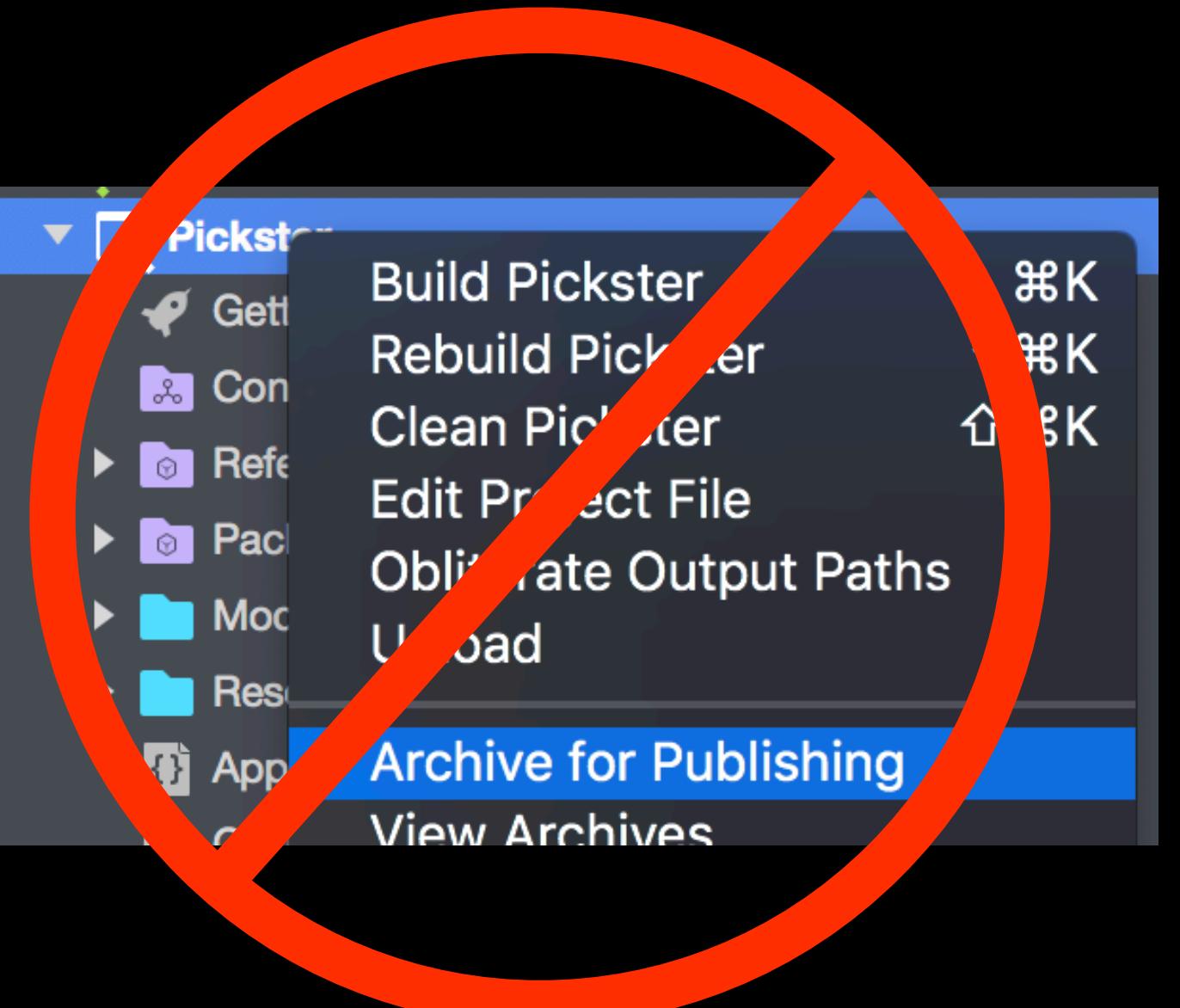
- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. **Build**
 2. Test
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- Support
- Resources

1 . BUILD

- Don't be the "build guy"
 1. Machines do it better. Many small details to forget/mess up
 - Version numbers, signing, misc. build config switch, dSYM files, what am I forgetting?, I can't even remember...
 2. You become the bottleneck
 3. In App Center, setting up builds is **delightful**



- "Friends don't let friends right-click, archive" - DB
 - Automate this as well
 - DEMO



THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. **Test**
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- Support
- Resources

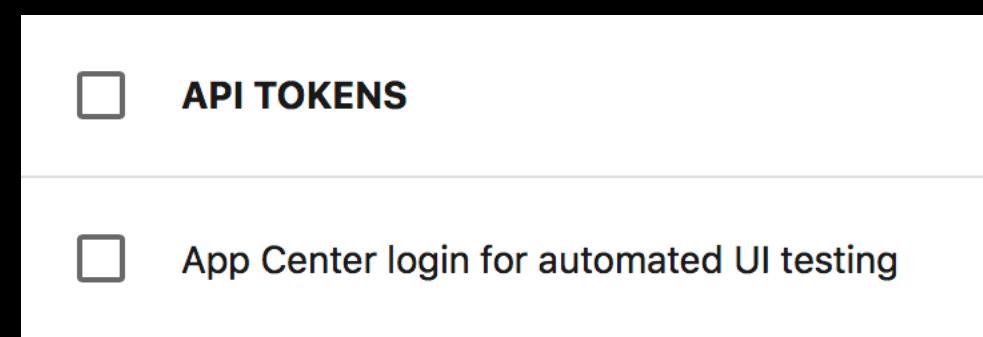
2. TEST

- 6 options for UI testing frameworks: Appium, Calabash, Espresso, XCUITest, Xamarin.UITest, UWP tests ("shortly")
- DEMO

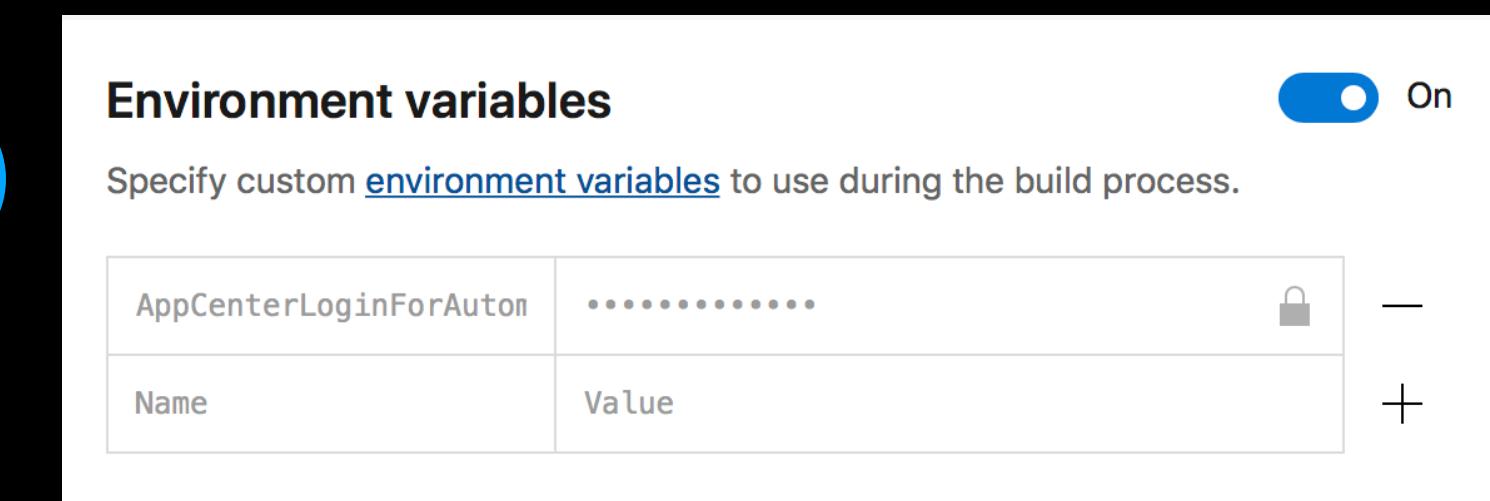
2. TEST

- Once your Xamarin UI Tests are written and working locally, you need 5 pieces:

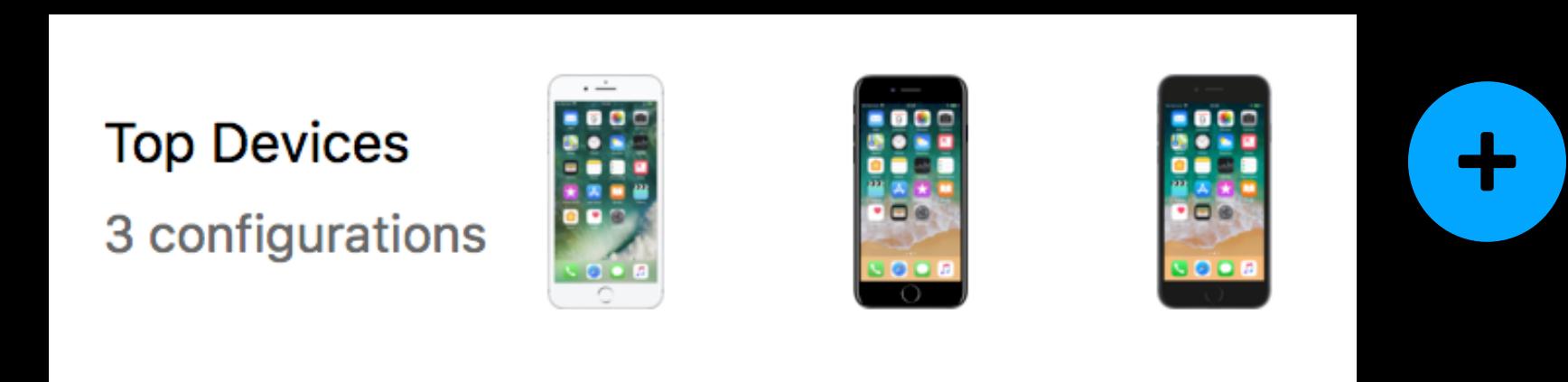
1. App Center API token



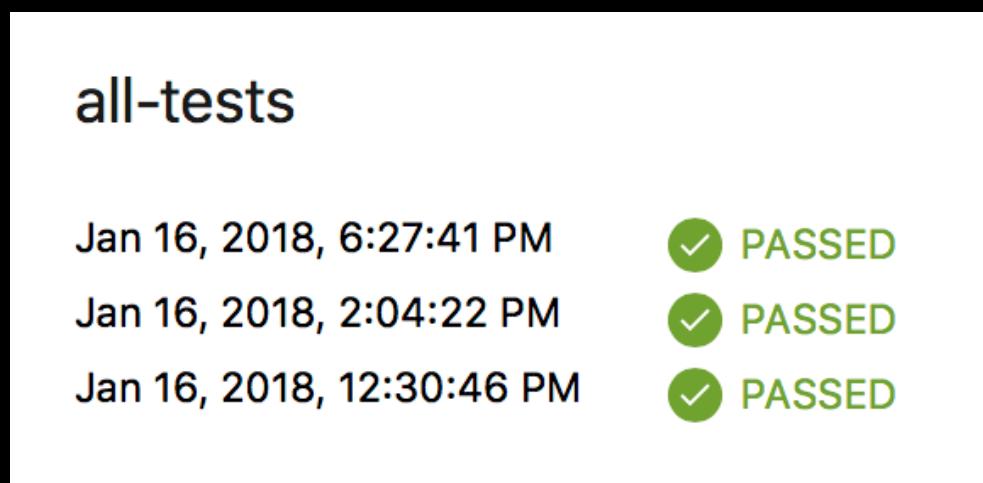
2. As build env. variable



3. Test Cloud device set



4. Test series name



5. Post-build bash script

```
echo "> Run UI test command"
appcenter test run uitest --app $appName --devices $deviceSetName --app-path $APPCENTER_OUTPUT_DIRECTORY/Pickster.ipa
--test-series $testSeriesName --locale "en_US" --build-dir $APPCENTER_SOURCE_DIRECTORY/Pickster.UITests/bin/Debug
--uitest-tools-dir $APPCENTER_SOURCE_DIRECTORY/packages/Xamarin.UITest.*/tools --token $appCenterLoginApiToken
```

- How to set this up the EASY way

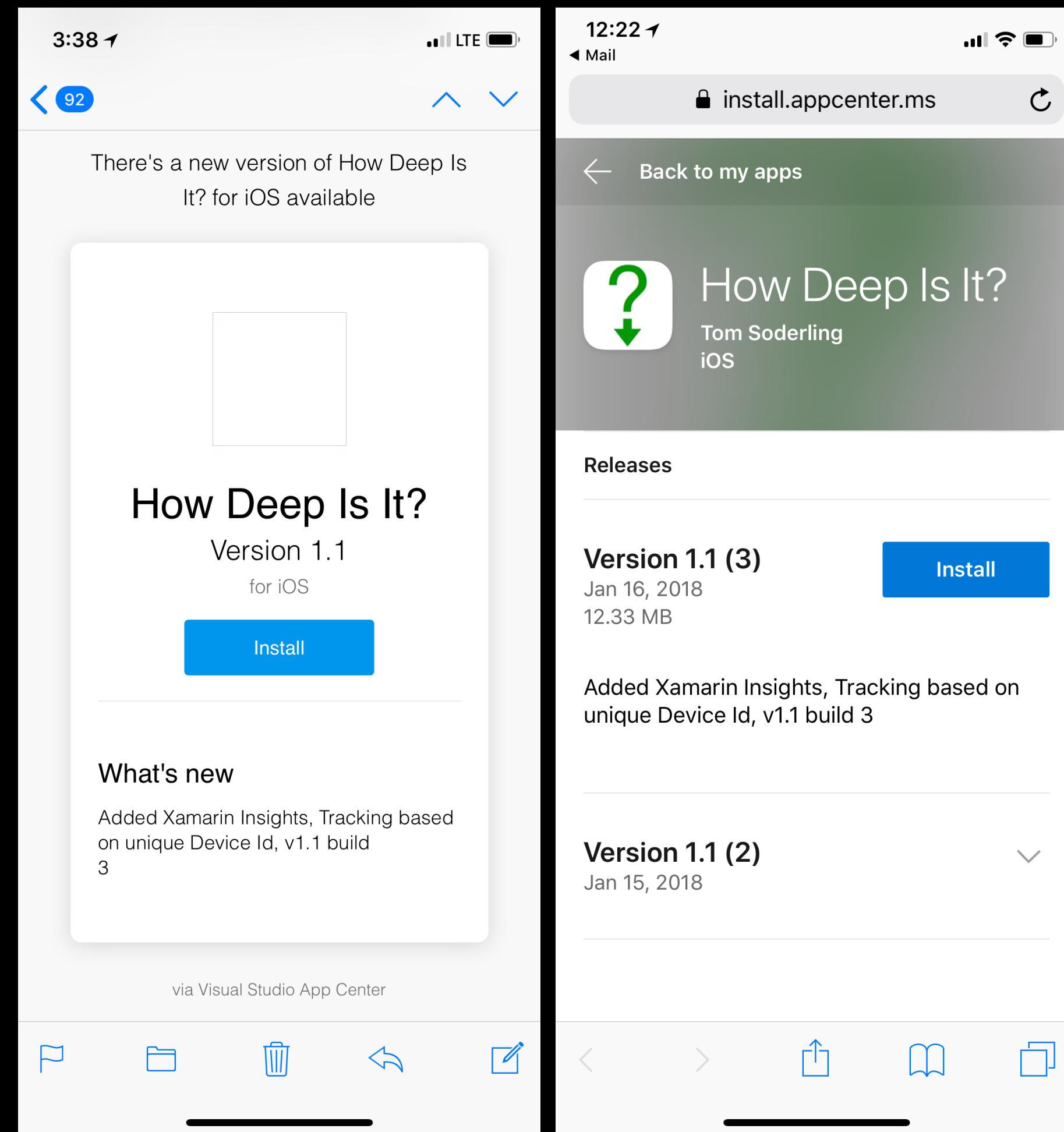
- <https://tomsoderling.github.io/AppCenter-Automated-UI-tests-on-build>

THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. **Distribute**
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- Support
- Resources

3. DISTRIBUTE

- 2 Types
 - Groups (alpha/beta testers)
 - App Stores + Alpha/Beta, TestFlight
- DEMO
- HockeyApp side by side experience
 - Log in to App Center using your HockeyApp credentials. Your app data has already been synced - allowing you to explore App Center while continuing to use HockeyApp as normal.



THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. **Crashes**
 5. Analytics
 6. Push Notifications
- Customize
- Support
- Resources

4. CRASHES

- dSYM files for symbolication of crashes on iOS
 - Your App Center builds do it automatically!
- Automatically create tickets in your bug tracker with integrations for Jira, VSTS, or GitHub

THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Crashes
 5. **Analytics**
 6. Push Notifications
- Customize
- Support
- Resources

5. ANALYTICS

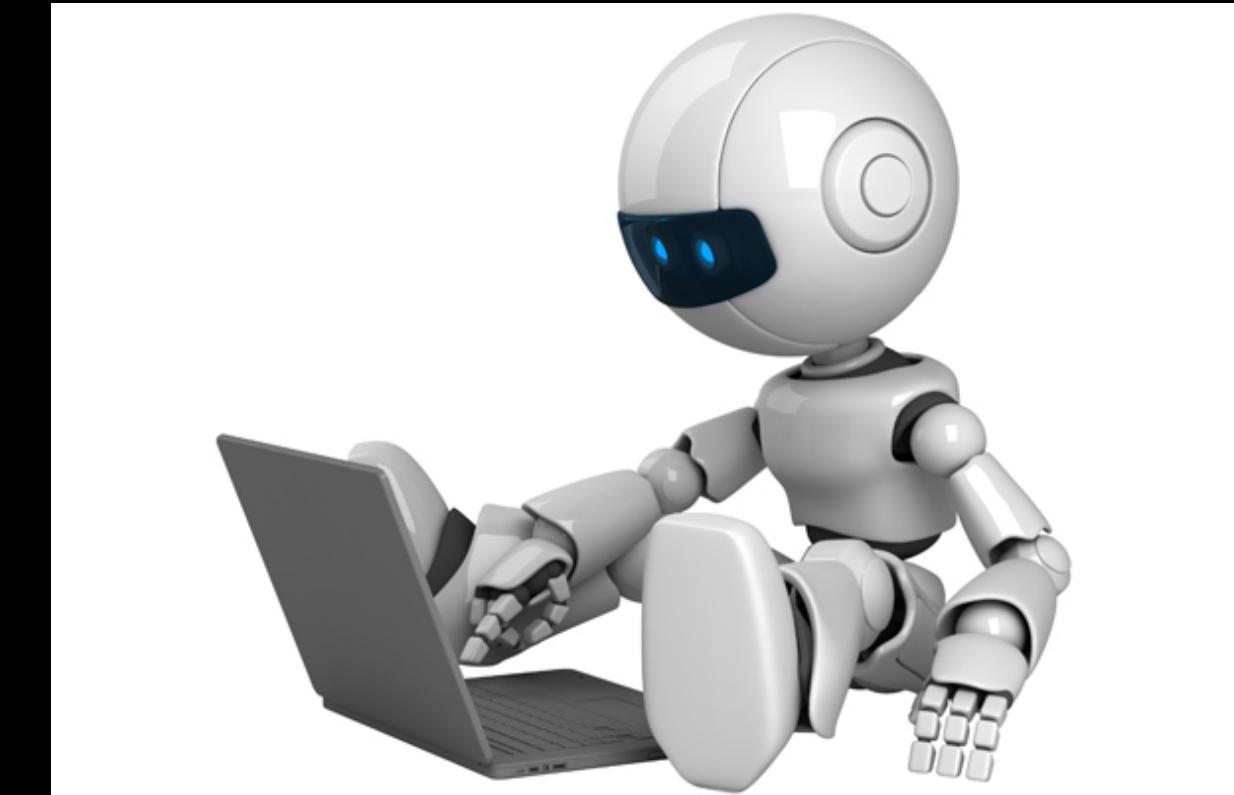
- Fascinating to watch!
- Use the analytics you've collected to see the top devices and OS versions users are running your app on. You can then create this specific device set for UI testing
- DEMO

THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- **Customize**
- Support
- Resources

CUSTOMIZE

- Classic DevOps question: What to do with API keys?
 - **Option 1:** Check into source control
 - Snatch and grab bitcoin mining
 - Jerk move: bots are scanning public GitHub repos
 - Amazon AWS S3 keys checked in for **5 min = 140** EC2 servers mining for bitcoins overnight to the tune of **\$2,375**
 - **Option 2:** Not that ^
 - Use build environment variables



CUSTOMIZE

- Method using --assume-unchanged

```
namespace Spaniel
{
    // Changes to this file are ignored so API keys won't be checked in to github. Used command:
    // git update-index --assume-unchanged [path to file]
    // To undo:
    // git update-index --no-assume-unchanged [path to file]
    public static class ApiKeys
    {
        public const string AppCenterAppSecret_iOS = "[your iOS App Center secret goes here]";
        public const string AppCenterAppSecret_Android = "[your Android App Center secret goes here]";
    }
}
```

Check in this state

- Run this command --assume-unchanged[filename]
- Make changes to the file:

```
namespace Spaniel
{
    // Changes to this file are ignored so API keys won't be checked in to github. Used command:
    // git update-index --assume-unchanged [path to file]
    // To undo:
    // git update-index --no-assume-unchanged [path to file]
    public static class ApiKeys
    {
        public const string AppCenterAppSecret_iOS = "02342342-sdf0234ksadf-234msd";
        public const string AppCenterAppSecret_Android = "0458123123-dfksdf0234-asdfsdf324534";
    }
}
```

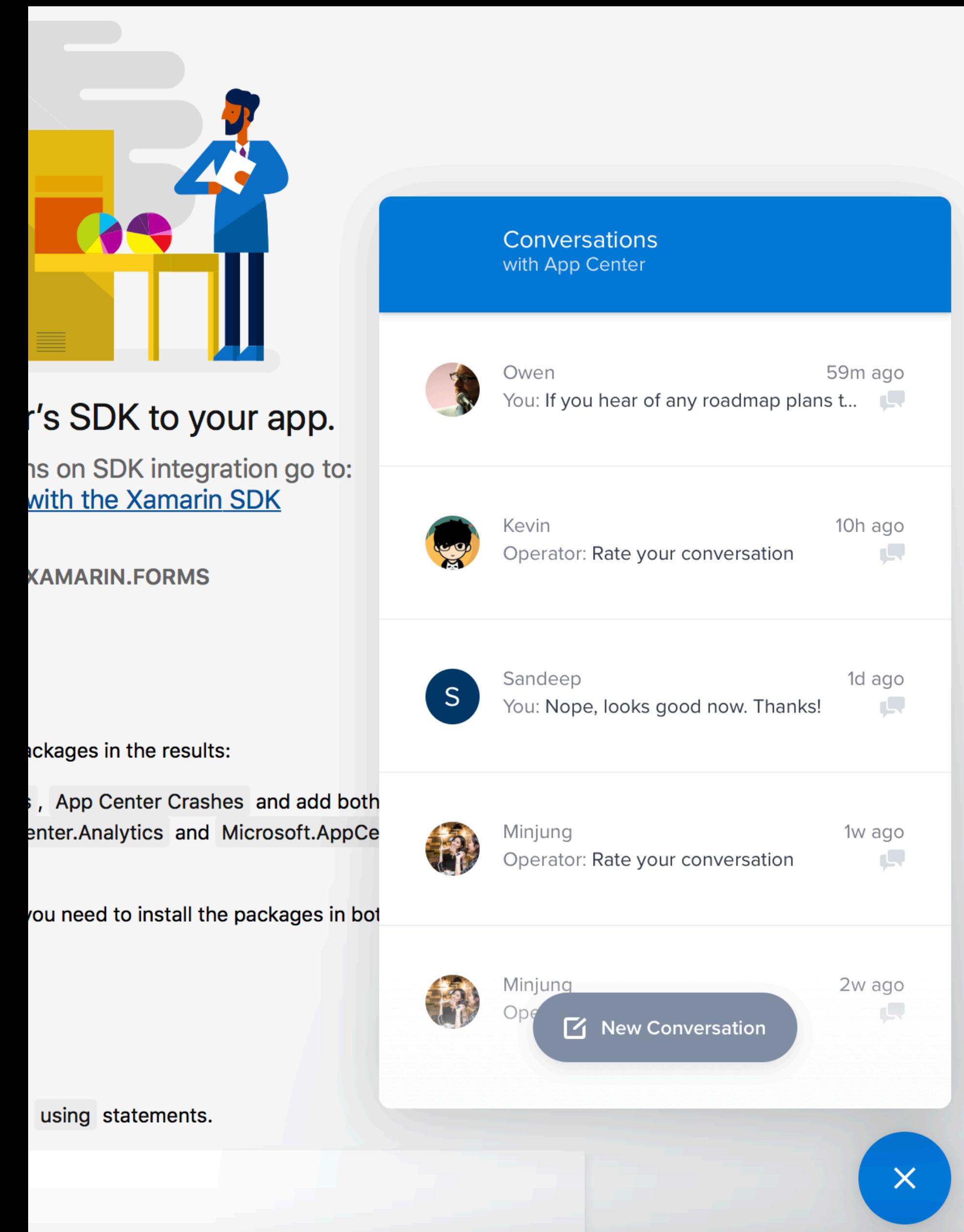
Useful locally in this state

THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- **Support**
- Resources

SUPPORT

- Stepped up in App Center
 - Nice on-page chat
 - Paste in screenshots
 - Usually quick response (live even)



THE PLAN

- Introduction: what's this all about?
- 6 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Crashes
 5. Analytics
 6. Push Notifications
- Customize
- Support
- **Resources**

RESOURCES

- App Center blog: <https://blogs.msdn.microsoft.com/vsappcenter>
- Xamarin webinar next week (Jan 25th): xmni.io/2EtGxQR
- Overview post with links to deep dive videos on Channel 9
- <https://blogs.msdn.microsoft.com/vsappcenter/introducing-visual-studio-app-center>
- Product roadmap: <https://docs.microsoft.com/en-us/appcenter/general/roadmap>
- What's changed: <https://docs.microsoft.com/en-us/appcenter/general/changelog>
- Adding Test Cloud UI tests to your build pipeline the EASY way
- <https://tomsoderling.github.io/AppCenter-Automated-UI-tests-on-build/>
- Azure Application Insights
- <https://blogs.msdn.microsoft.com/vsappcenter/better-decisions-through-better-analytics-visual-studio-app-center-with-azure-application-insights>