

HAVE MORE **FUN** BY AUTOMATING THE LIFECYCLE OF YOUR APPS WITH



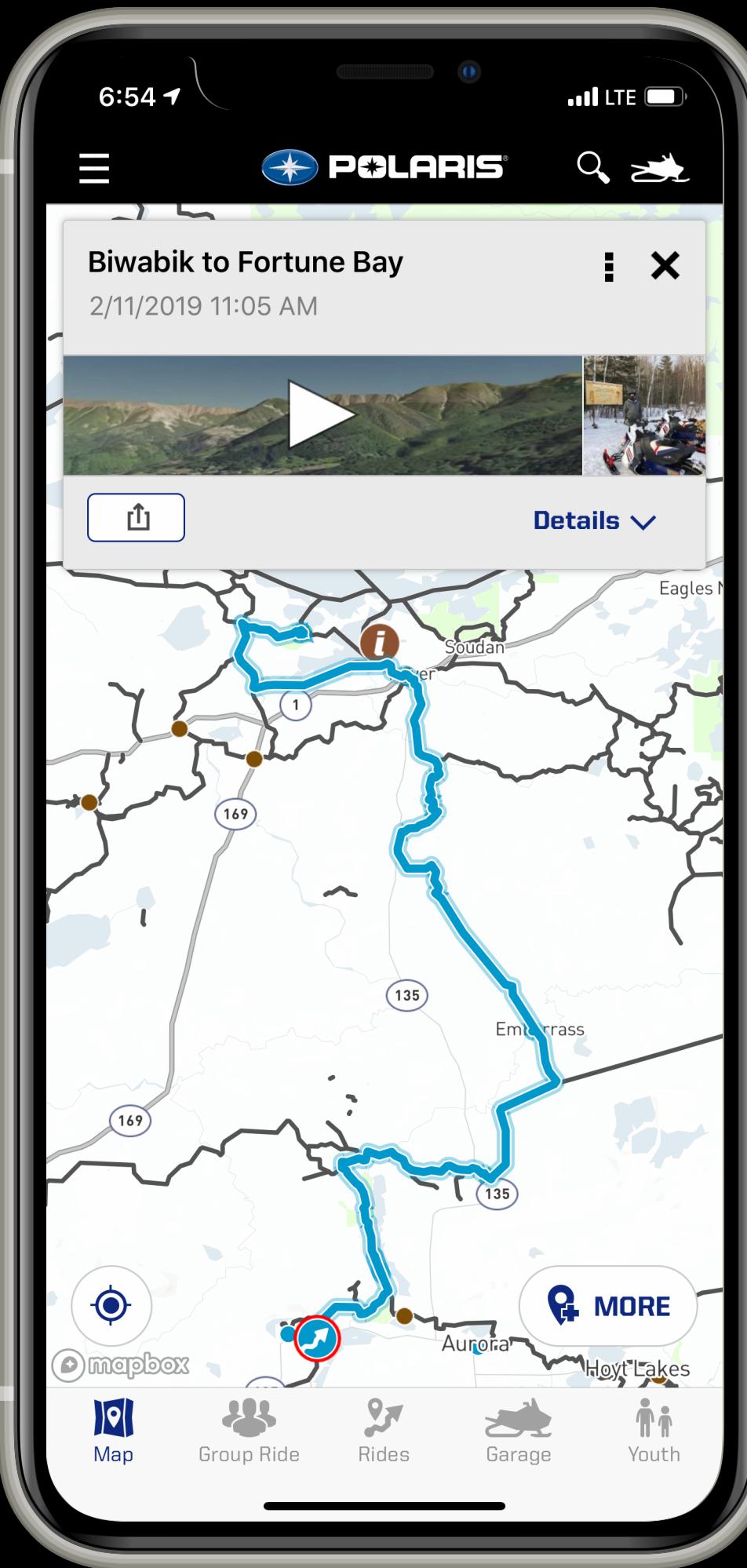
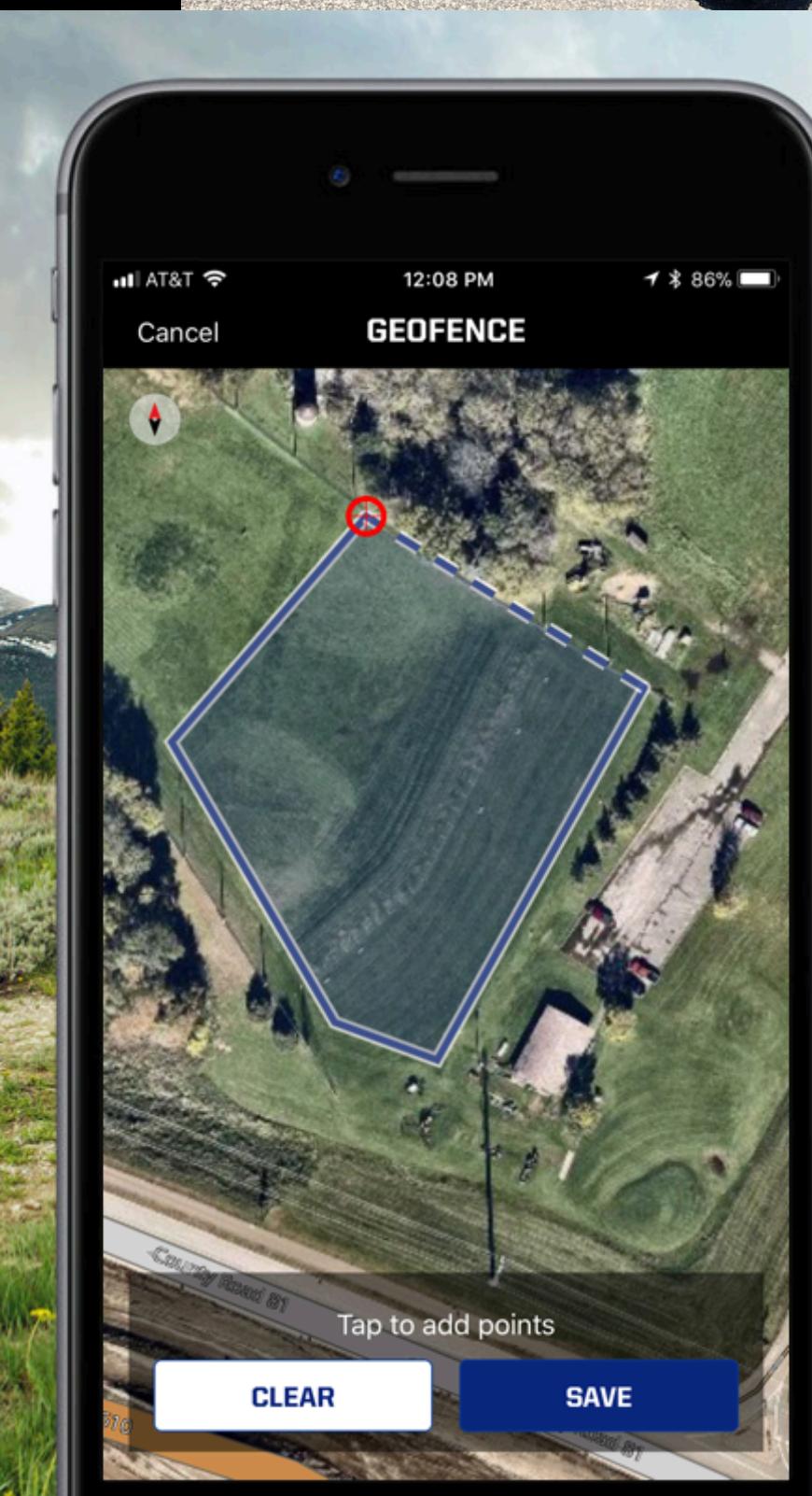
VISUAL STUDIO APP CENTER

ABOUT ME

Tom Soderling

Sr. Mobile Apps Developer @ Polaris Industries; Ride Command
Xamarin.Forms enthusiast & DevOps hobbyist
5-year Xamarin Certified Mobile Developer

Blog: <https://tomsoderling.github.io>
GitHub: <https://github.com/TomSoderling>
Twitter: [@tomsoderling](https://twitter.com/tomsoderling)



THE PLAN

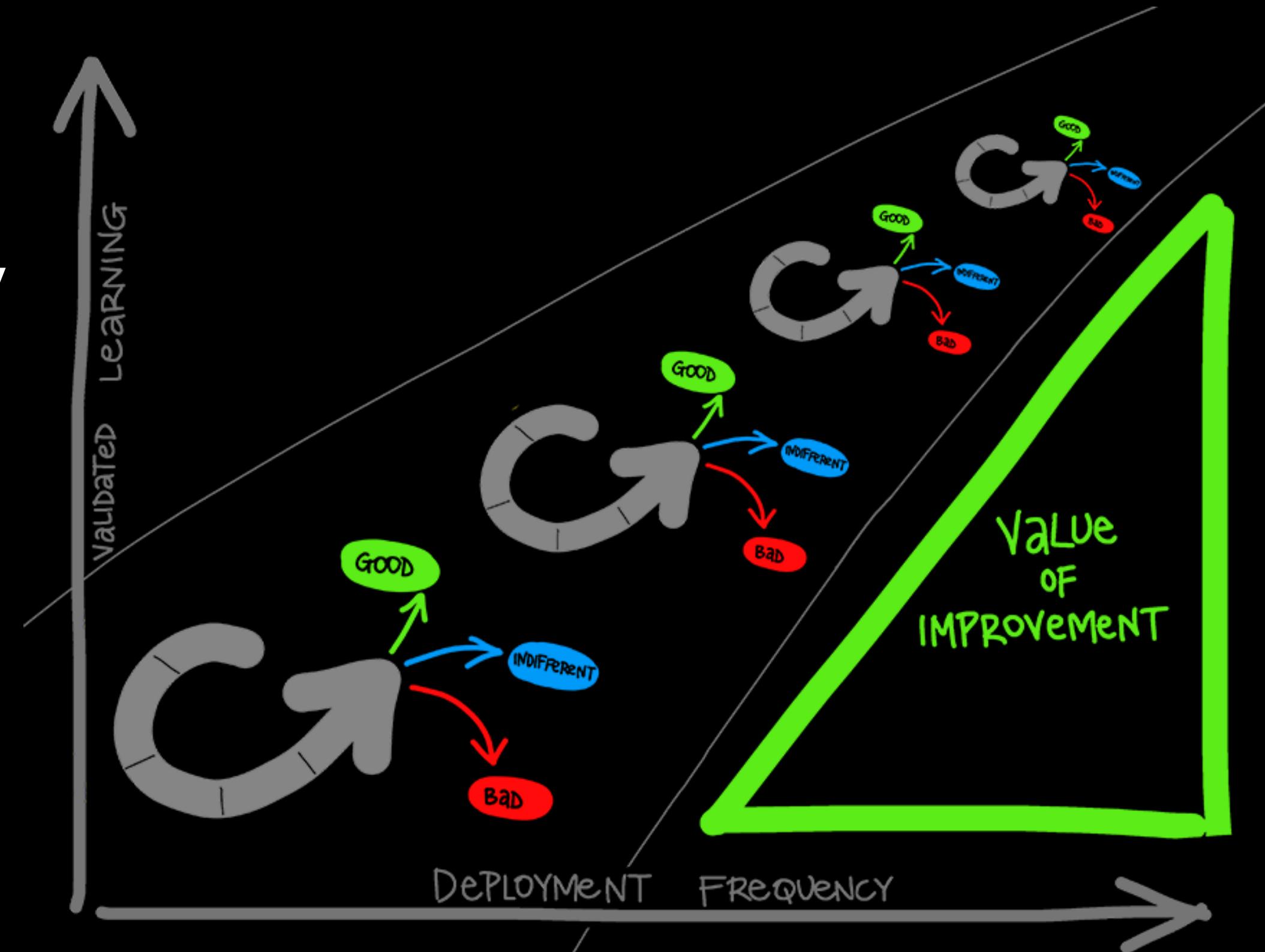
- **Introduction: what's this all about?**
- 5 Major Functional Areas:
 1. Build
 2. Test
 3. Distribute
 4. Diagnostics
 5. Analytics
- Customize
- Support
- Resources



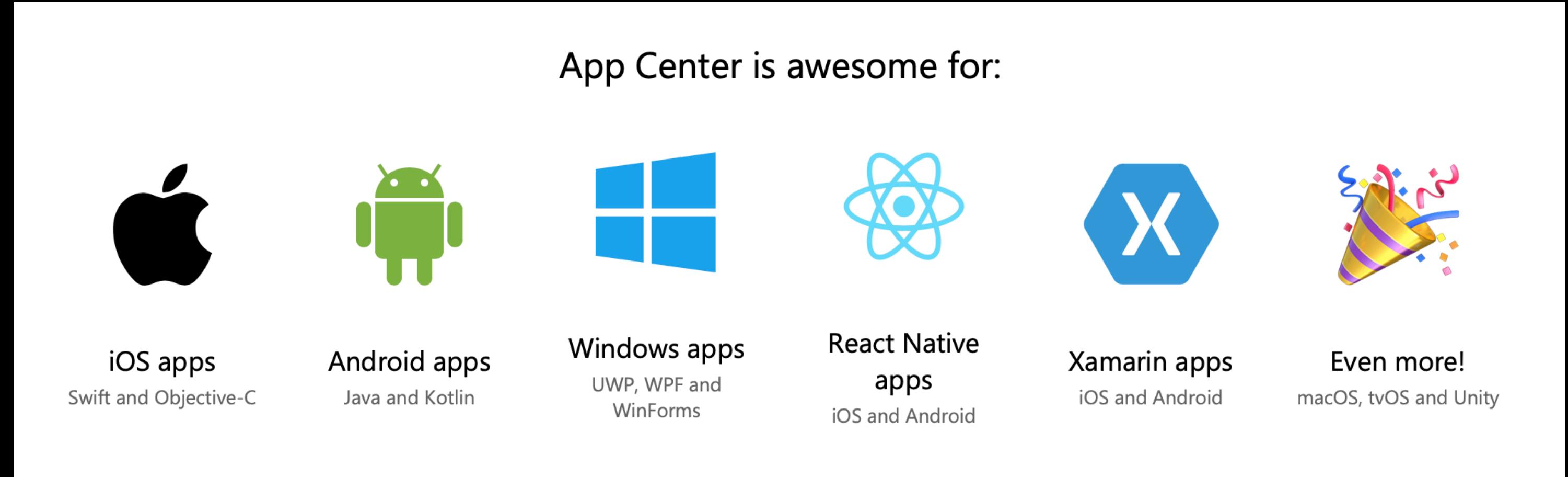
Use Visual Studio App Center to release **more frequently**,
with higher quality, and **have more fun** by spending
time on coding features instead of managing overhead

INTRODUCTION

- Continuous integration, delivery, and learning solution for Android, iOS, Windows, macOS, Unity, and tvOS apps
- A.K.A. “DevOps”
- “DevOps is the union of people, process, and products to enable **continuous delivery of value** to our end users.”



INTRODUCTION



- First went into public preview at Connect(); 2016 as Visual Studio **Mobile** Center
- General availability at Connect(); 2017 (Nov. 15) renamed to Visual Studio **App** Center



INTRODUCTION

App Center Mission Statement:
“Make app development 10x easier”

INTRODUCTION

- Pricing
- Distribute, Analytics, and Crashes are completely free
- Build offers generous free tier
- Test has free launch tests on build, and an **unlimited** 30-day free trial

free tier

free trial

free

free

free

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

THE PLAN

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

1 . BUILD

- Before
 - Works on **my** machine
 - “Build day” anxiety
- Now
 - Continuous integration
- App Center provides cloud-hosted build environments

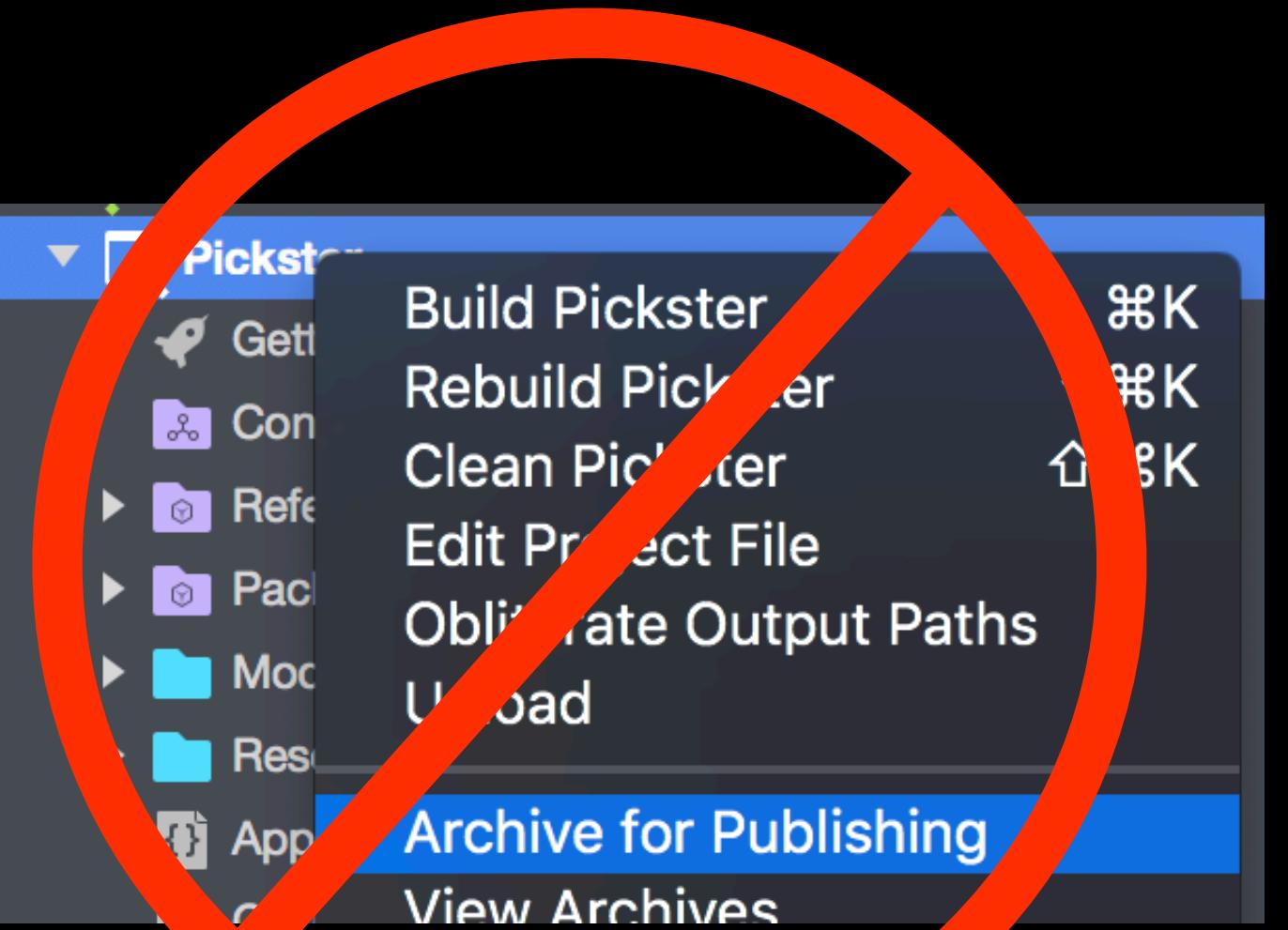


1 . BUILD

- Don't be the "build person"
 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 read me....
 2. You become a bottleneck
 3. In App Center, setting up builds is **delightful**



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



1 . BUILD

- Failures
 - Everyone's favorite kind of email
 - Can set up Slack integration so **everyone** knows



Visual Studio App Center

BUILD REPORT
Pickster 2
master
Build #343 failed
Duration: 14 minutes 3 seconds

[View build](#)

Change the type and frequency of the notifications in your app [notifications settings](#).

Visual Studio App Center

BUILD REPORT
Pickster 2
master
Build #344 fixed
Duration: 5 minutes 36 seconds

[View build](#)

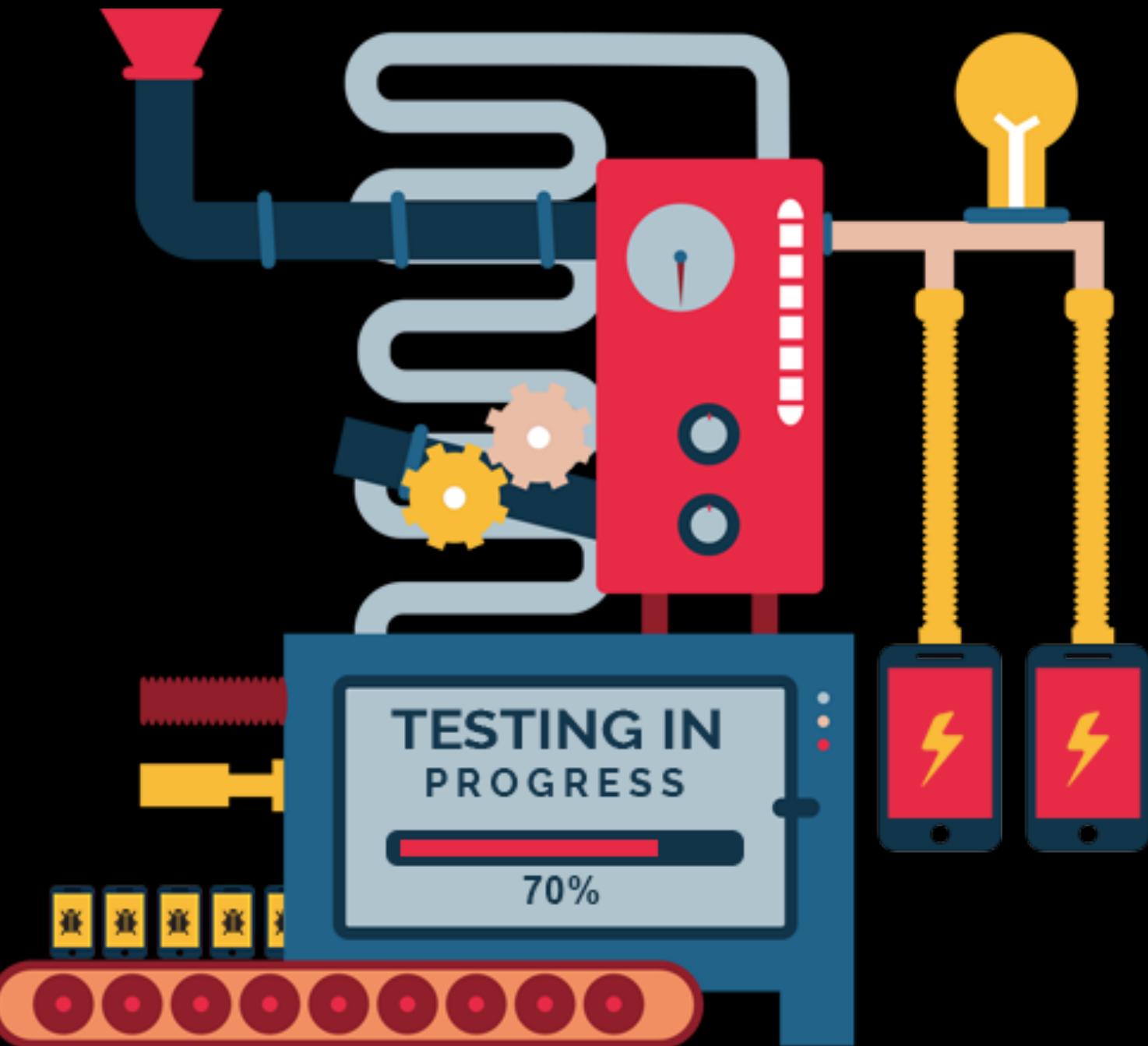
Change the type and frequency of the notifications in your app [notifications settings](#).

THE PLAN

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

2. TEST

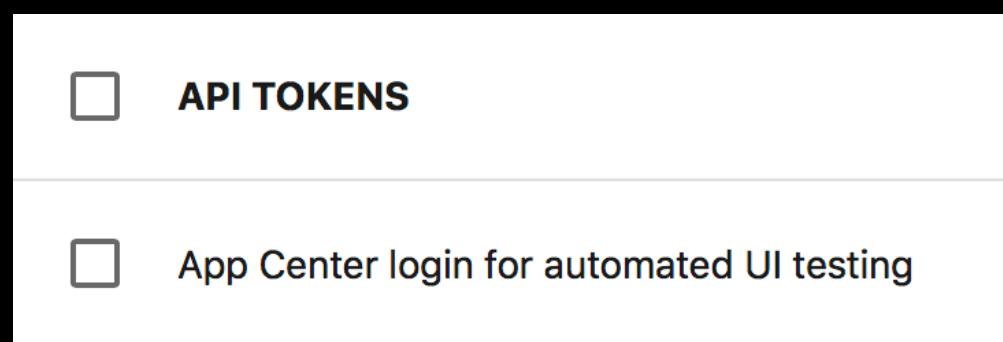
- 5 options for automated UI testing frameworks:
 - **Appium** (open source, iOS & Android, Windows & Mac apps)
 - **Calabash** (open source by Xamarin, iOS & Android apps)
 - **Espresso** (open source by Google, Android)
 - **XCUITest** (by Apple, iOS)
 - **Xamarin.UITest** (NUnit-based, iOS & Android)
 - **UWP** tests (TBD)
- Tip: Use the Page-Model Pattern to write UI tests
(link in resources)
- DEMO



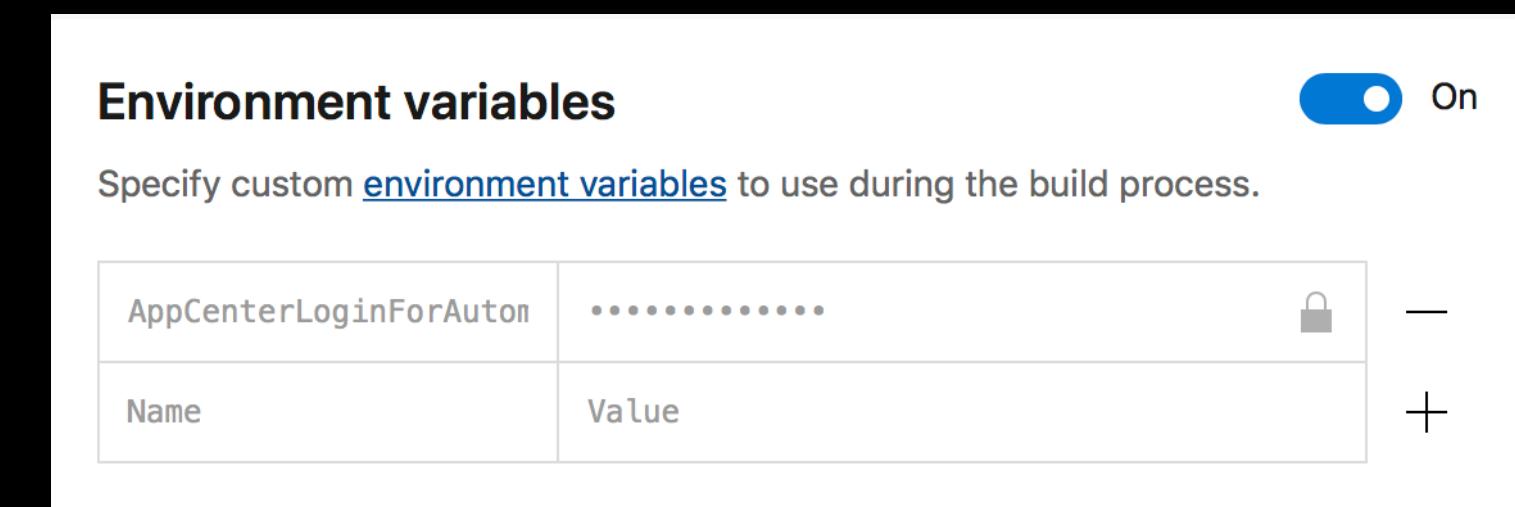
2. TEST

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

1. App Center API token



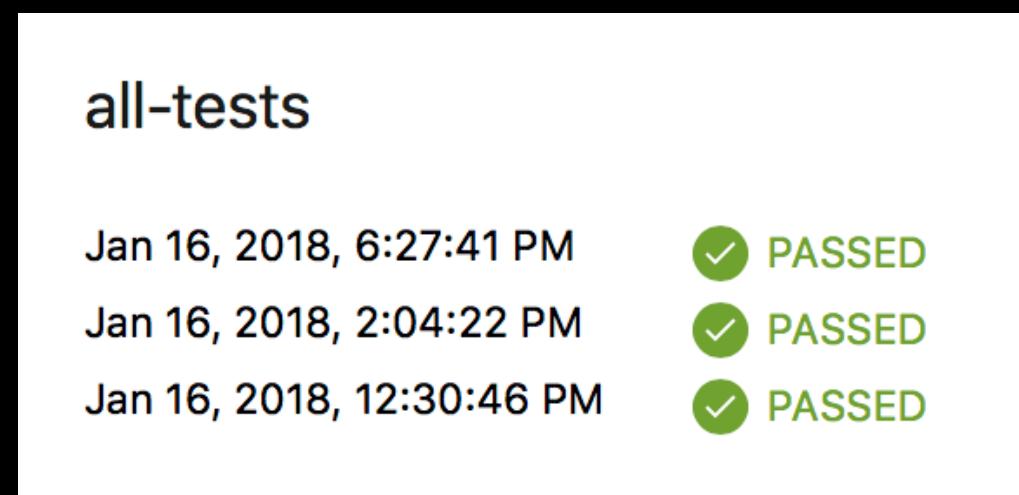
2. As build env. variable



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 $appCenterLoginApiKey
```

- How to set this up <https://tomsoderling.github.io/AppCenter-Automated-UI-tests-on-build>

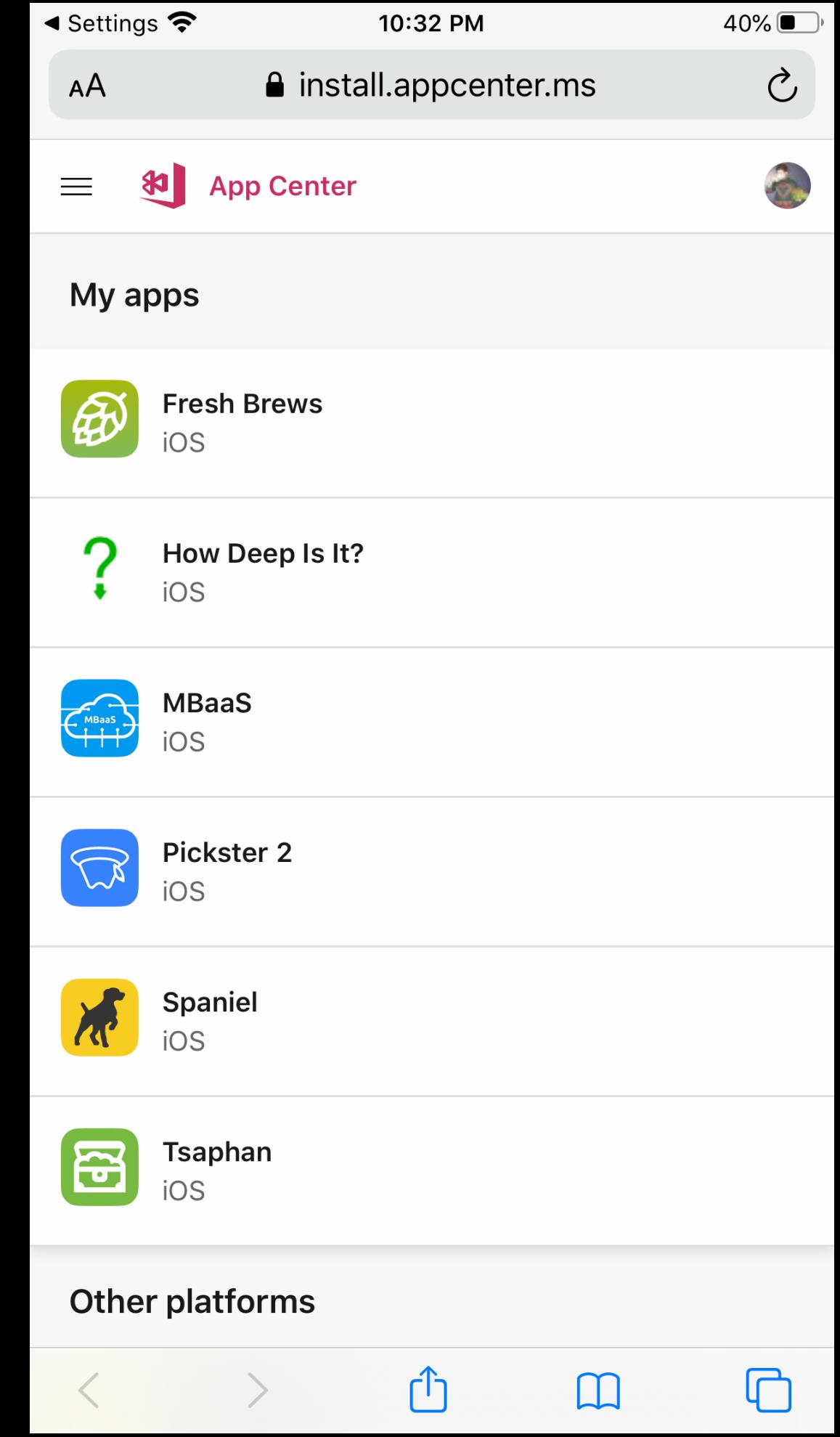
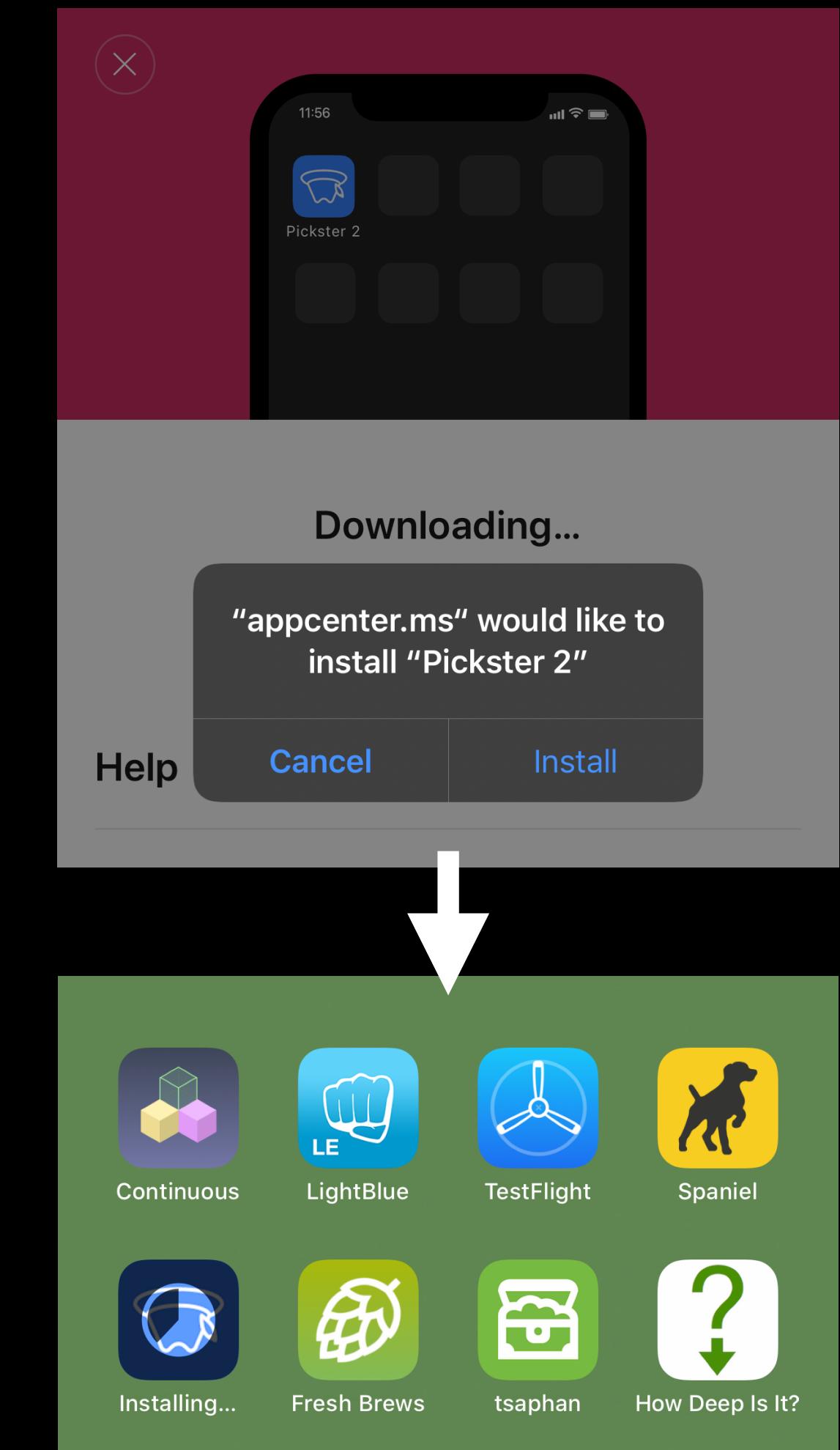
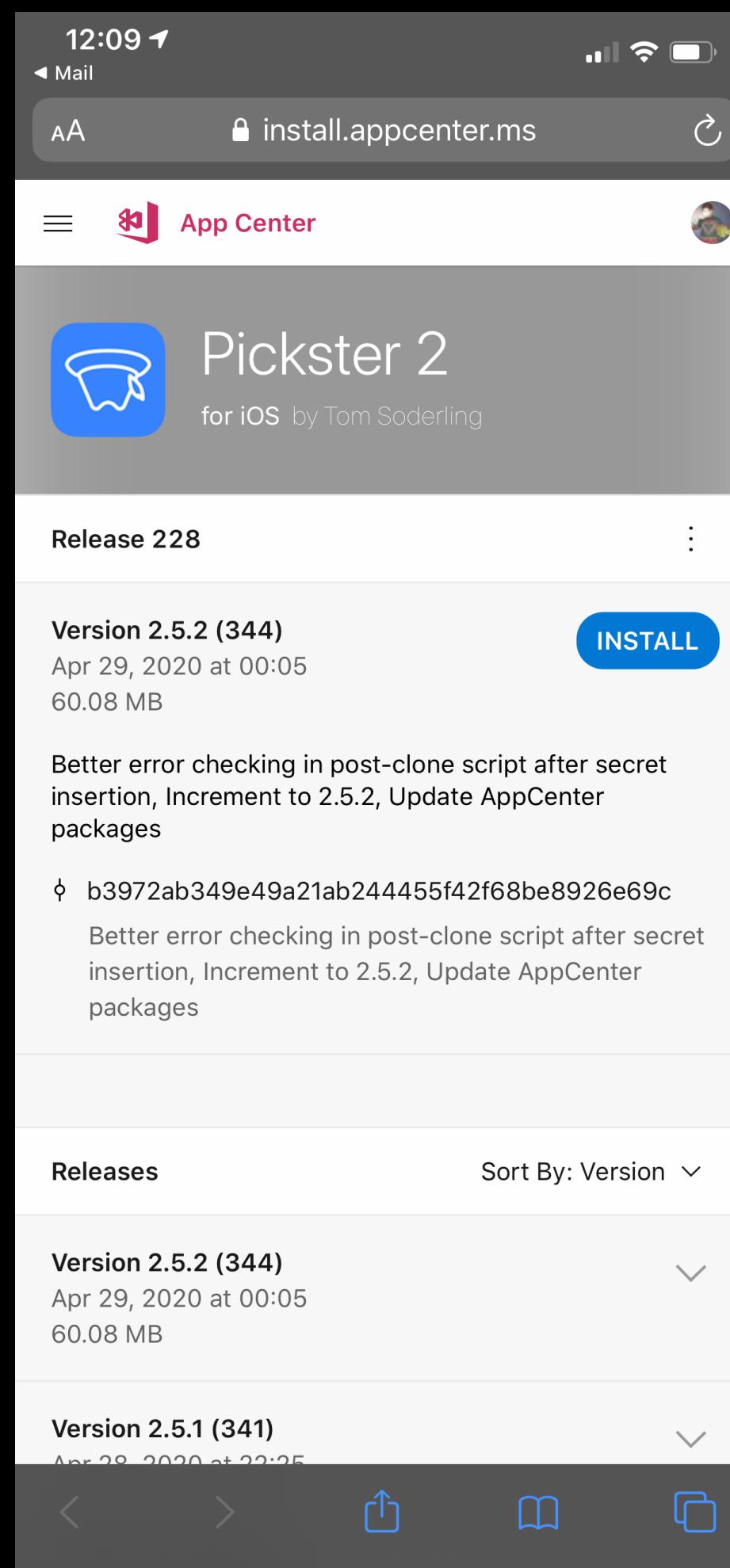
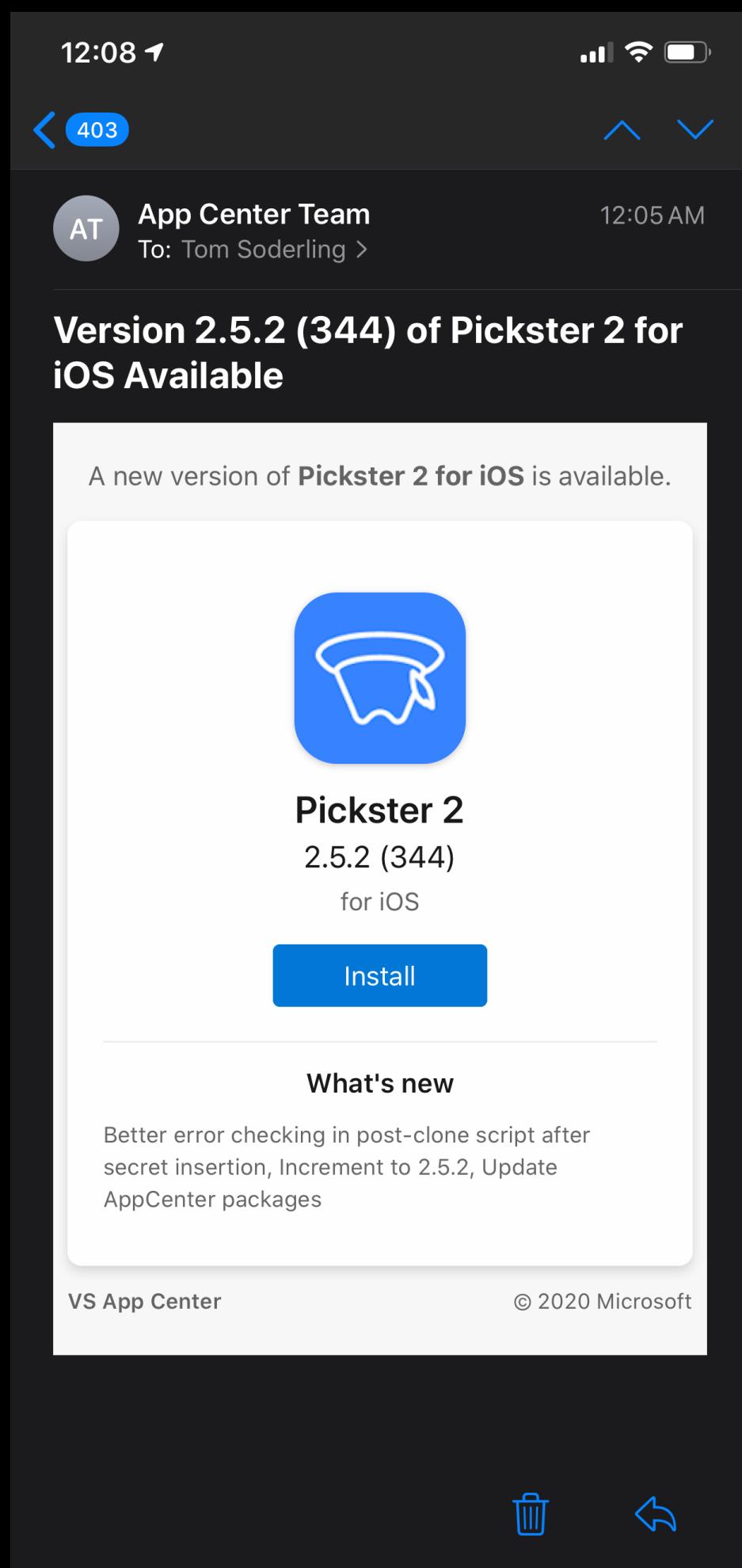
THE PLAN

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

3. DISTRIBUTE

- Two Types

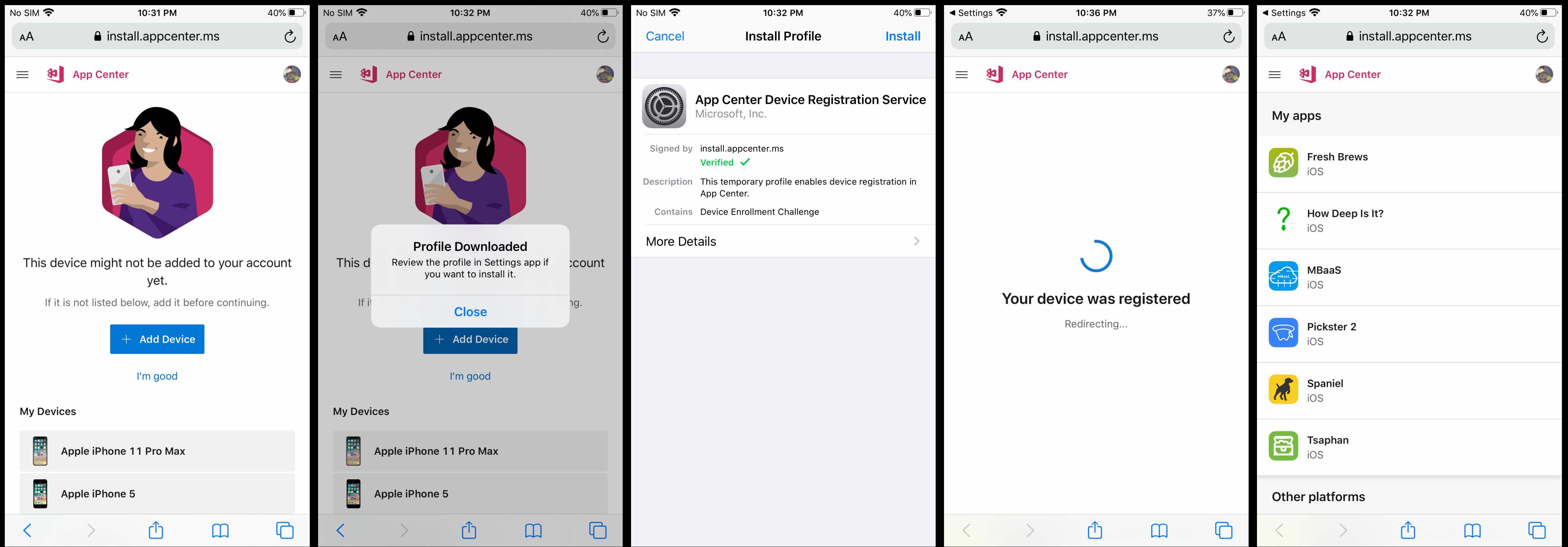
1. Groups (alpha/beta testers)



All builds available
on demand

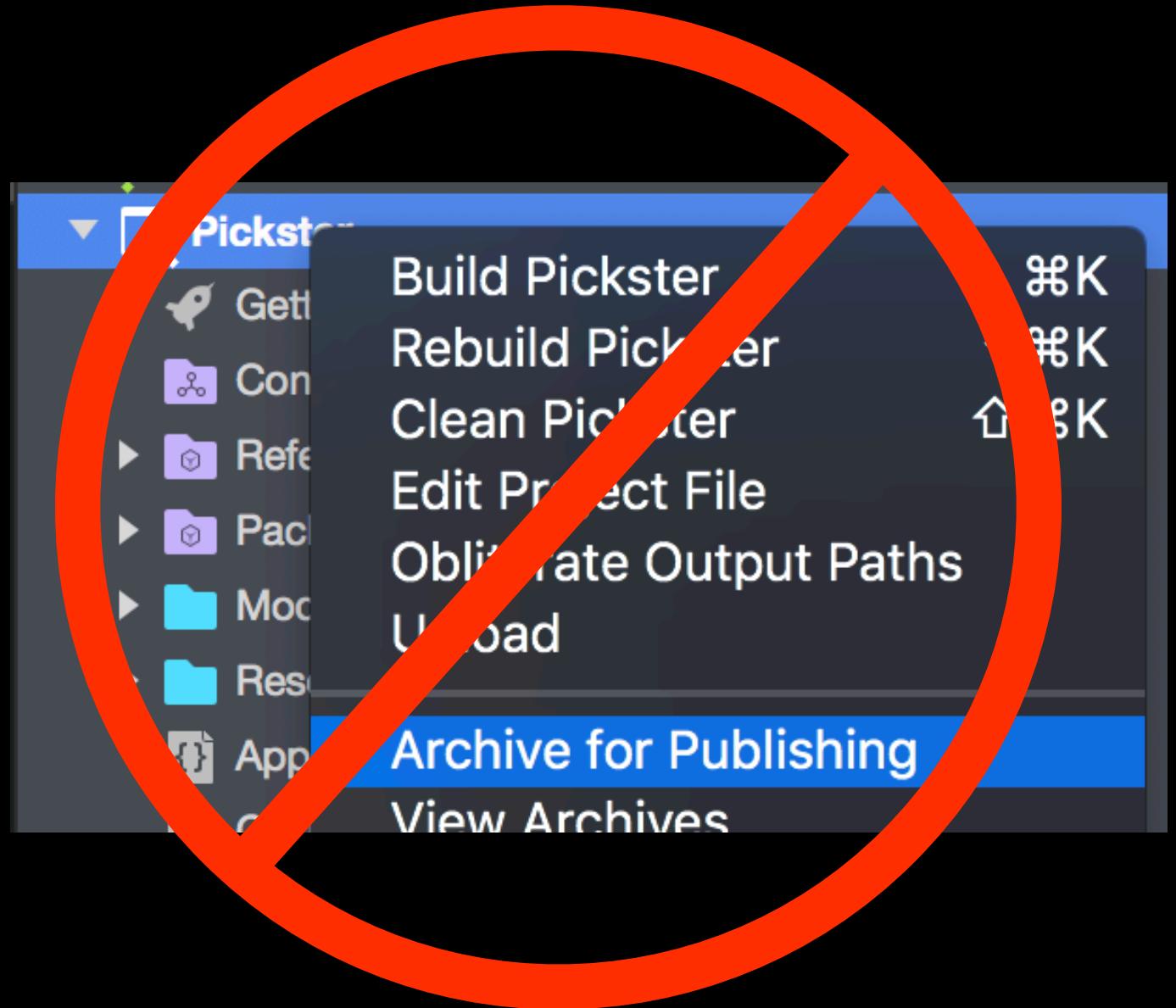
3. DISTRIBUTE

- Two Types
 1. Groups (alpha/beta testers)
Onboarding new devices/users



3. DISTRIBUTE

- Two Types
 1. Groups (alpha/beta testers)
 2. App Stores + Alpha/Beta, TestFlight
- DEMO

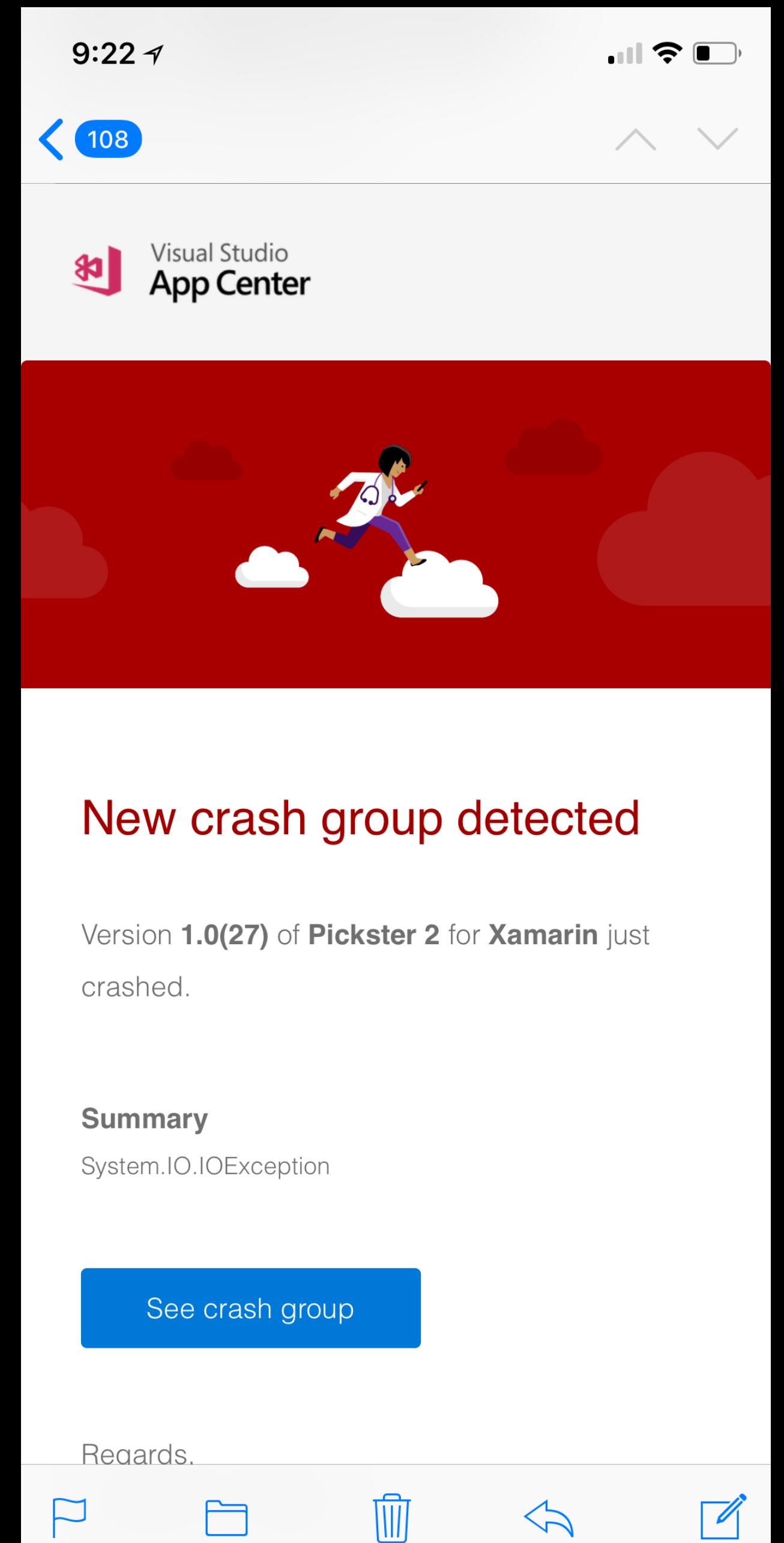


THE PLAN

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

4. DIAGNOSTICS

- Two Types:
 1. Crashes (unhandled exceptions)
 2. Errors (caught exceptions, unexpected behavior)
- How? Simply by adding the VS App Center SDK
- 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
- DEMO



THE PLAN

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

5. ANALYTICS

- Fascinating to watch!
- Lots of uses:
 - Does anyone care about new feature X? Monetize v2.0?
 - What feature does nobody care about at all?
 - Did the # of crashes/errors spike? Decrease?
 - Hone UI testing effort: top devices and OS versions
 - Custom events: drive decisions on real user data
 - Is it worth it to internationalize my app?
- DEMO



5. ANALYTICS

- Want to go deeper?
 - Export to Azure: Continuously export your raw data into Azure for extended retention and query capabilities.
 - Azure Application Insights
 - Smart detection
 - Usage analysis
 - Dashboards
 - Power BI



THE PLAN

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

CUSTOMIZE

- What are my customization options?
 - Three spots for bash shell scripts:

1. Post-clone

The post-clone script runs immediately after the repository was cloned but before we do anything else on our end.

2. Pre-build

The pre-build script runs before the actual build starts, but after we have installed dependencies from e.g. NuGet, CocoaPods or Carthage.

3. Post-build

The post-build script runs after the build has finished and we have copied all the necessary artifacts to the output directory.

Build scripts:

Post-clone

Pre-build

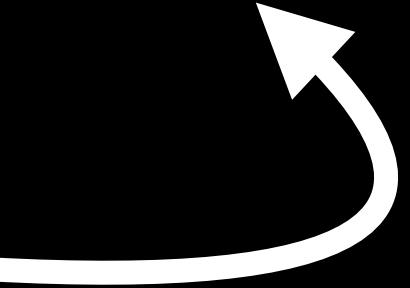
Post-build

Learn more about [custom build scripts](#)

CUSTOMIZE

- What would I use these for?
 - Insert API keys/secrets (script in resources)
 - Run unit tests (script in resources)
 - Kick off automated tests in Test Cloud, or some other device farm
 - Kobiton, SauceLabs, AWS Device Farm
 - Report build status: email, Slack, next to GitHub commit, etc.
 - Change app constants, config, resource dict.
 - Upload build somewhere via SFTP
- Microsoft repo of build script examples:
 - <https://github.com/microsoft/appcenter/tree/master/sample-build-scripts>

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:** Anything but 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

CUSTOMIZE

- What about at build time?
 - Pair with post-clone build script to slide in keys
 - <https://github.com/TomSoderling/VSAppCenter>
- =====Task : Shell ScriptDescription : Run a shell script using bashVersion : 2.1.3Author : Microsoft CorporationHelp : [More Information] (<https://go.microsoft.com/fwlink/?LinkId=613738>)=====

```
[command]/bin/bash /Users/vsts/agent/2.131.0/work/1/s/Pickster/appcenter-post-clone.sh
```

App Center Secret Inserter

Working directory: /Users/vsts/agent/2.131.0/work/1/s/Pickster
Secret from env variables: *****
Target file: /Users/vsts/agent/2.131.0/work/1/s/Pickster/..../Pickster/Pickster.Shared/ApiKeys.cs
Text to replace: \[your iOS App Center secret goes here\]

Target file found
Load file: /Users/vsts/agent/2.131.0/work/1/s/Pickster/..../Pickster/Pickster.Shared/ApiKeys.cs
Line found
App secret inserted

Script complete

THE PLAN

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

SUPPORT

- Really stepped it up in App Center
- Form to fill out on the page
- 8 business hour response time or less
- Product roadmap
 - Designed in the open
 - Feature requests

Go to docs

?

📡 System status

▶ Roadmap

★ Top feature requests

➤ Submit feature request

👤 Contact support

THE PLAN

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

RESOURCES

- <https://www.theurlist.com/vsappcenter>

