Can we deploy native mobile apps multiple times per day with confidence?

Karl Krukow
Principal Engineering Manager, Microsoft
@karlkrukow on Twitter
Can we deploy native mobile apps multiple times per day with confidence?

Karl Krukow

Principal Engineering Manager, Microsoft
@karlkrukow on Twitter
Never forget a special occasion.

Confetti is a simple app that reminds you of birthdays, anniversaries, Mother's Day and Father's Day for the people you care about most.

https://github.com/krukow/confetti
WHY?
Key Benefits of Continuous Delivery

- Reduced lead-time
- Faster feedback
- Higher-Quality
- Reliable releases
The emotional cycle of manual delivery:

- **Plateau of Obliviousness**
- **Cliff of Urgency**
- **Target Ship Date**
- **Actual Ship Date**
- **Ascent of Celebration**
- **Peak of Jubilation**
- **Slope of Relief**
- **Valley of Complacency**

**Abyss of Apple**

The business case for continuous delivery, Atlassian blog
What’s Stopping Us?
What are all the steps needed to ship?

What else?
Tests
Distribute beta
Sign for app-store distribution
Release notes / metadata
Generate Updated Screenshots
iTunes Connect
Submit for review
Apple
Publish
User updates
Feedback
New Ideas
Experiments
What are all the steps needed to ship?

Experiment
Idea
Change

Change  Build  Test  Distrib  Resign  Upload  Sub  Apple®  Pub  User

Feedback
New Ideas
Experiments
August 15, 2015

Setting Up A Continuous Build Environment For Xamarin: Part 1 - Jenkins

Continuous integration is the core foundation of the DevOps lifecycle, as it allows tests to be run and builds to be created every time a member of the development team checks in new code. This allows the team to quickly know if the latest changes have ‘broken the build’ and depending on how the continuous integration is configured, for that check in or commit to be rejected (essentially allowing the team to pre-emptively avoid major issues from ever making it into the build).

Setting up a continuous integration environment for Xamarin is a non-trivial task because of the number of dependencies that Xamarin requires to build packages (a lot of this is thanks to Apple’s requirements that you can only build iOS apps on a Mac). So in this 2 part tutorial series, I’ll walk through configuring two different continuous integration solutions for Xamarin. Part 1 will be built using a combination of Jenkins (for Xamarin.iOS and Xamarin.Android) and a hosted TFS build controller (for Windows/Windows Phone). Part 2 will be built using the new ‘Build.vNext’ tools that are included Visual Studio Online and the newly released Team Foundation Server 2015.

SAMPLE SOLUTION

To give you a sample solution to work with for building these continuous integration environments, I’ve taken the TaskyPortable solution from the Xamarin Samples GitHub account, made some modifications and uploaded it to my GitHub account. So you just need to download this solution and check it in/commit it to your VSO or TFS Team Foundation Server.
Quality & QA cycle time
Form Factors

http://opensignal.com/reports/2015/08/android-fragmentation/
Want to build your own device lab?
DAUNTING
<table>
<thead>
<tr>
<th>Release Cadence</th>
<th>Avg. rating</th>
<th># Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>bi-weekly</td>
<td>4.5</td>
<td>155,344</td>
</tr>
<tr>
<td>weekly+</td>
<td>3.5</td>
<td>2,975,737</td>
</tr>
<tr>
<td>weekly+</td>
<td>4.2</td>
<td>71,968</td>
</tr>
<tr>
<td>Weekly-</td>
<td>3.5</td>
<td>354,415</td>
</tr>
</tbody>
</table>

sources: https://www.applyzer.com/ & App Store
* = approximation based on current frequency
Ratings as of June 11, 2017
Apple review cycle time

“I’ll get straight to the point, I’ve been thinking about shutting down the App Review Times site for quite a while now. I just don’t think it serves a useful purpose in 2018 now that review times have been stable at “About a Day” on both stores for just less than two years.”

Dave Verwer, Mar 22, 2018
https://medium.com/@daveverwer/the-future-of-app-review-times-a4c35ff30129
Less than 2h to get a brand new iOS App validated! I think this is my new personal record. Kudos for AppStore review teams! #iosreviewtime
Automated UI Testing

- Interact with UI controls in your app using gestures
- Declarative query language to identify views on screen
- Wait for events to occur (e.g., no spinner visible)
- App-lifecycle APIs (start/stop, reset, etc)
- Generate screenshots for test reports
Demo: XCUITest for iOS
Some UI Test Automation Frameworks

- **Native/Platform provider solutions**
  - Apple/iOS: XCUITests
  - Google/Android: Espresso
  - Xamarin: Xamarin.UITest

- **Cross platform frameworks:**
  - Xamarin.UITest
  - Appium
  - Detox (React Native)
  - Calabash (deprecated)

- **Others:**
  - EarlGrey (iOS, made by Google)
  - KIF 2.0 (iOS, open source)
  - UIAutomator (Android, made by Google)

- **What you choose depends on your organisation**
  - Platform (e.g. native iOS, Android, React or Xamarin)
  - Cross platform teams or siloed experts?
  - Developers writing test or dev-in-test or outsourced
  - Experience with Selenium
  - Programming language experience (e.g. C#)

What you choose depends on your organisation:

- Platform (e.g. native iOS, Android, React or Xamarin)
- Cross platform teams or siloed experts?
- Developers writing test or dev-in-test or outsourced
- Experience with Selenium
- Programming language experience (e.g. C#)
Cloud CI providers can help you accelerate
Cloud hosted mobile CI services

Experiment
Idea
Change

Feedback
New Ideas
Experiments

Change  Build  Test  Distrib  Resign  Upload  Sub  Apple  Publish
Demo: Build-Test-Distribute

Visual Studio App Center

https://appcenter.ms
# Cloud Mobile CI - 10,000 foot view

<table>
<thead>
<tr>
<th>Feature</th>
<th>App Center</th>
<th>Firebase</th>
<th>AWS</th>
<th>bitrise</th>
<th>BuddyBuild</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosted macOS</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>real-device testing</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>distribution</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>analytics</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>crash analytics</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>engagement / push</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>remote config</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Cloud Mobile DevOps

• How much are you looking for?

• Mix-and-max à la carte, e.g.,
  • bitrise for build + App Center Test
  • BuddyBuild + AWS Device Farm
  • Visual Studio Team Services + Test and Distribute via App Center
  • DIY: Jenkins or Xcode server (self-hosted) build & test + Firebase analytics and remote config + fastlane tools
  • …
Release More Often!

yearly → quarterly → monthly → bi-weekly → weekly

Daily → ???
Now you’ve got new problems...

(but they’re better problems!)
Release note overload

- Users don't read release notes if you’re shipping every week or they have automatic updates.

- The “What’s new screen”:
  - From time to time include Help Screens to inform of changes
  - they should be light and fast.
Version management

- Add a 'kill switch' / 'expiry' to manage the complexity of multiple versions
  - Be careful: should the app be unusable if offline?
  - On the other hand: users might switch on airplane mode to not have to update
Managing App Store Ratings

- We can't be certain that we've prevented all bugs or unhappy users
- Provide in-app feedback before asking user to rate the app
  - DIY or
  - e.g. http://www.apptentive.com/
- Once in-app rating is high, direct to AppStore!
- One trick early on: publish your app under a different developer name to prevent association with your brand.
A/B Testing and Feature Toggles

- App Store distribution is Pull not Push so users control which version they run.
- "View JSON": different view-types, server controlled
  - Backend is able to direct some set of users to have different experiences
  - Roll out certain views/features slowly
  - Cut off, or reduce load from a new feature if there are problems
- Firebase Remote config & A/B Testing

https://engineering.linkedin.com/mobile/mobile-ab-testing-linkedin-how-members-shape-our-apps
Can we deploy native mobile apps multiple times per day with confidence?

@karlkrukow
karl.krukow@microsoft.com