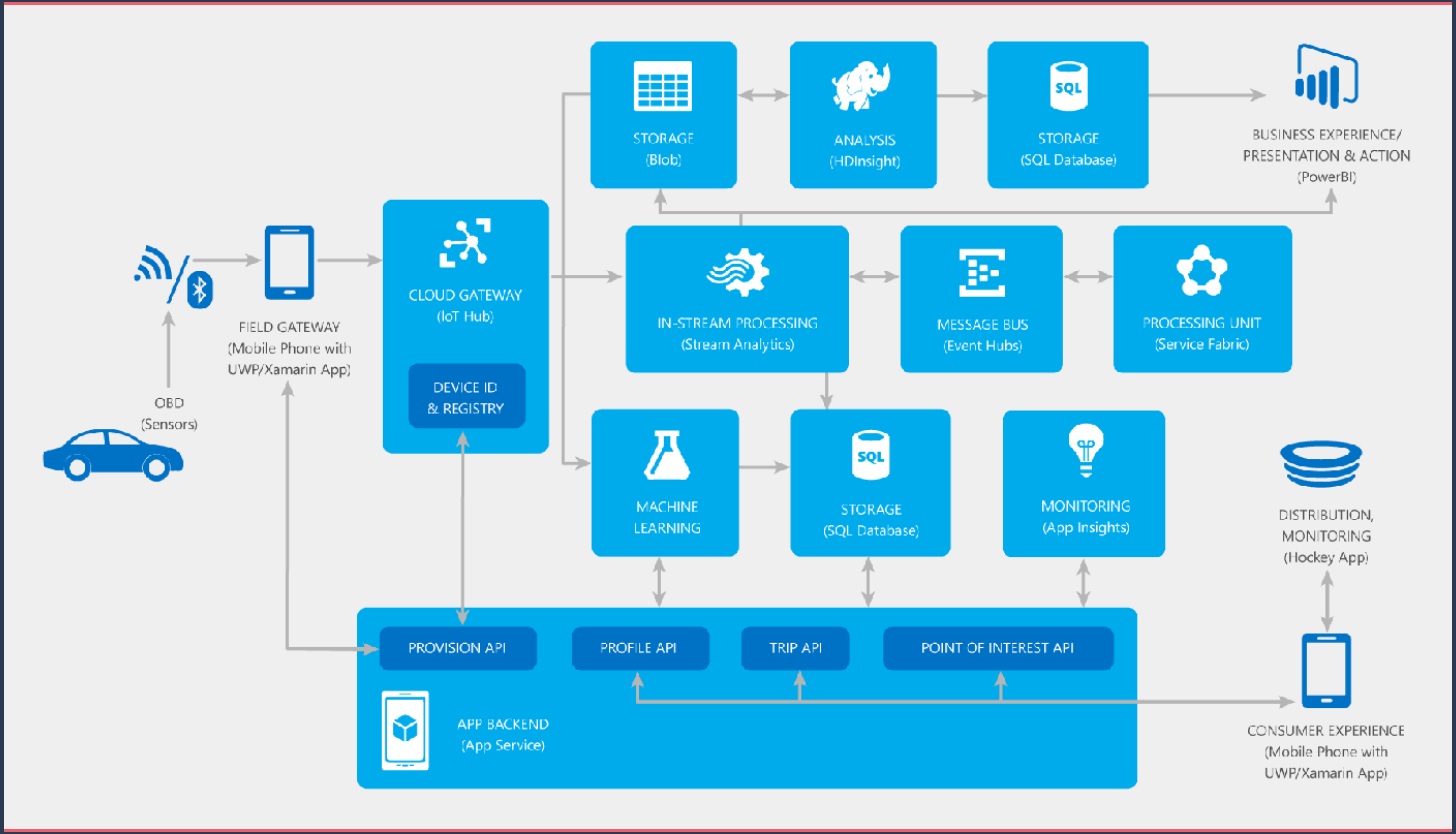
goto; copenhagen

GOTO Copenhagen 2017
Conference Oct. 1-3, 2017

Mobile Continuous Delivery is closer, easier (and more fun) than you think!

Karl Krukow

MyDriving Mobile & Azure IOT


STORAGE (Blob)
ANALYSIS (HDInsight)
STORAGE (SQL Database)
BUSINESS EXPERIENCE/ PRESENTATION & ACTION (PowerBI)
FIELD GATEWAY (Mobile Phone with UWP/Xamarin App)
CLOUD GATEWAY (IoT Hub)
DEVICE ID & REGISTRY
IN-STREAM PROCESSING (Stream Analytics)
MESSAGE BUS (Event Hubs)
PROCESSING UNIT (Service Fabric)
OBD (Sensors)
MACHINE LEARNING
STORAGE (SQL Database)
MONITORING (App Insights)
DISTRIBUTION, MONITORING (Hockey App)
PROVISION API
PROFILE API
TRIP API
POINT OF INTEREST API
APP BACKEND (App Service)
CONSUMER EXPERIENCE (Mobile Phone with UWP/Xamarin App)

Release More Often!

$$\begin{tikzcd}
\text{yearly} \arrow[r, bend left] & \text{quarterly} \arrow[r, bend left] & \text{monthly} \arrow[r, bend left] & \text{bi-weekly} \arrow[r, bend left] & \text{weekly}
\end{tikzcd}$$

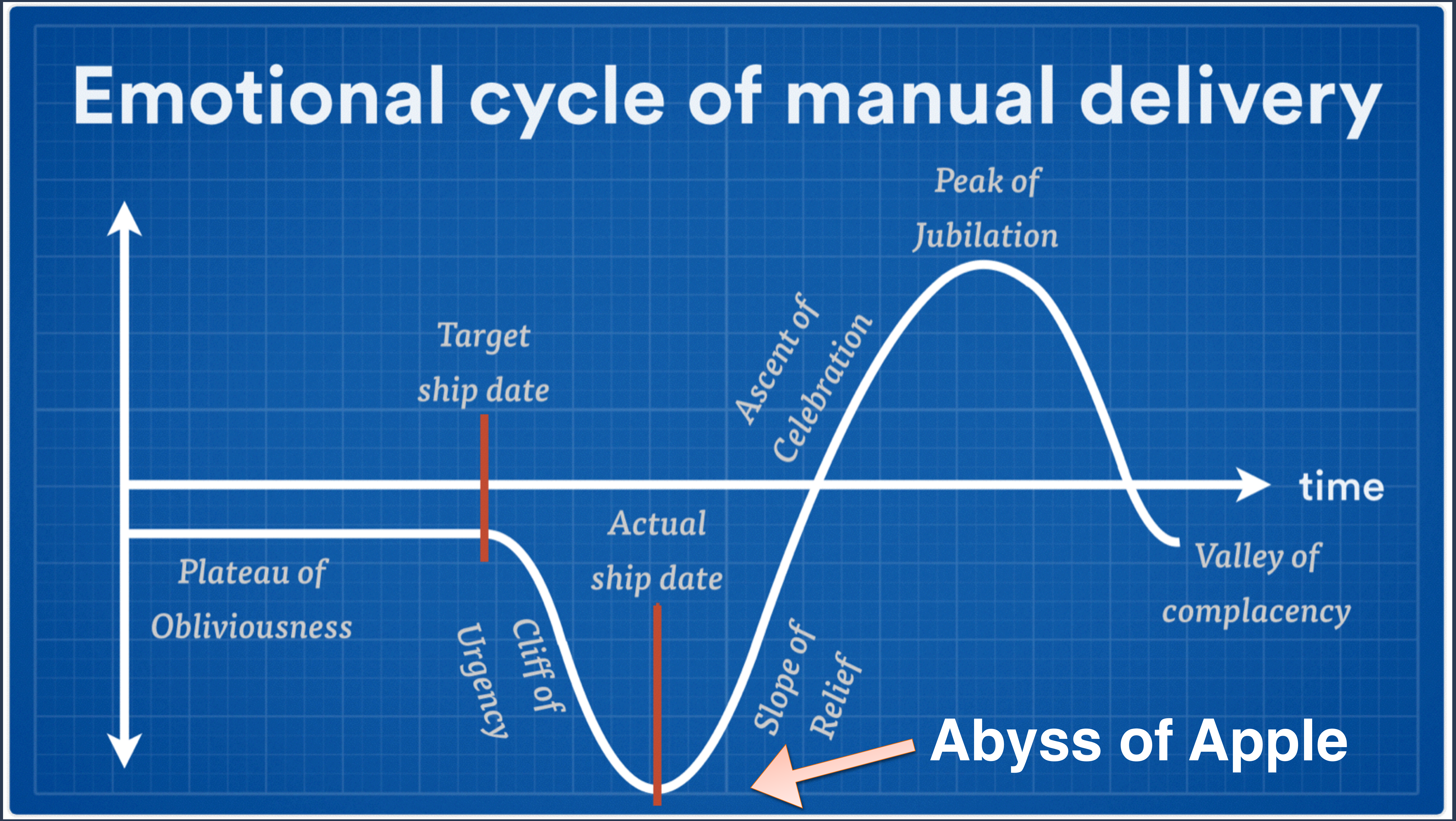
Why?

Key Benefits of Continuous Delivery

Reduced lead-time

Faster feedback

Higher-Quality



The business case for continuous delivery, Atlassian blog

I/S
eyewear

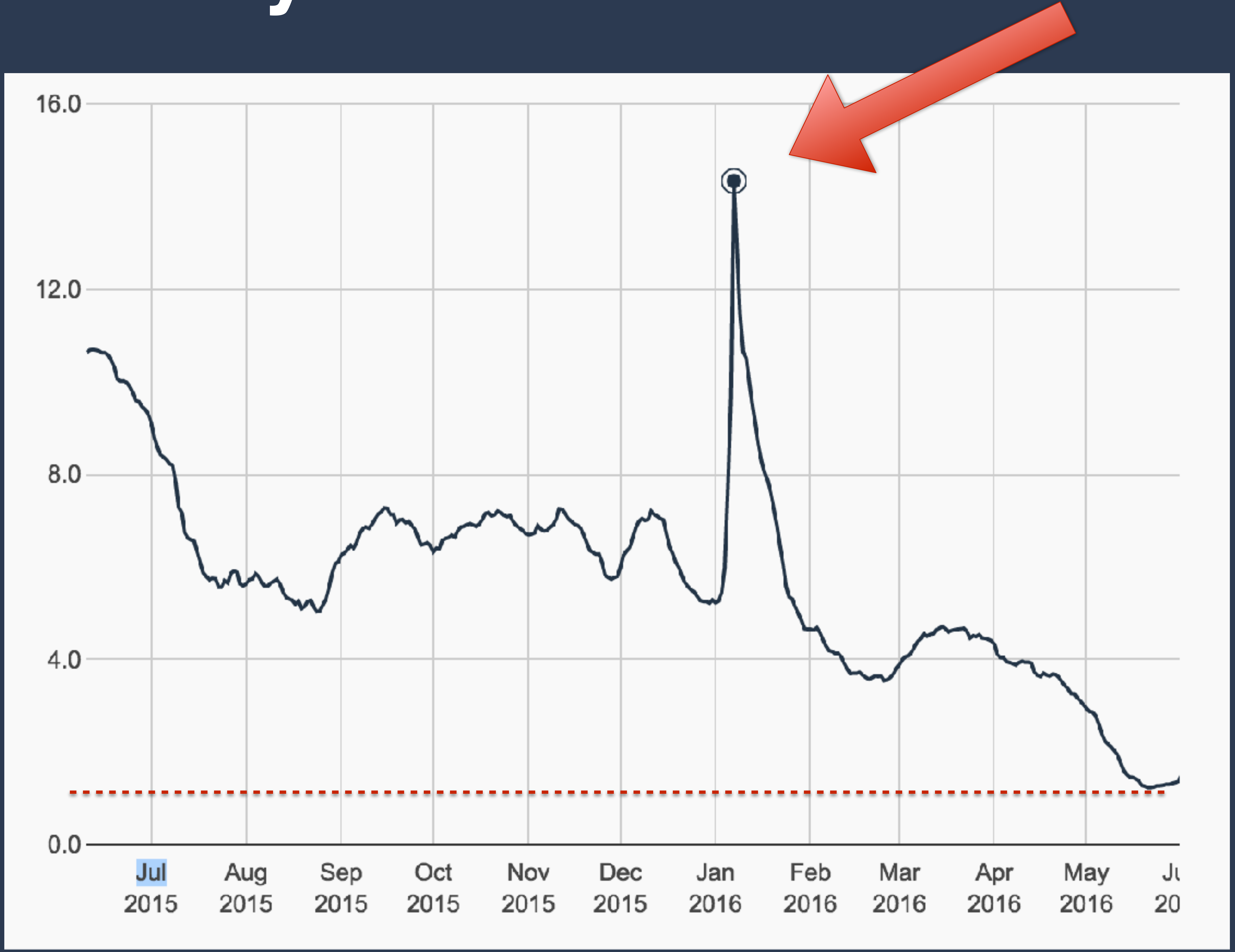
Release Cadence	**bi-weekly**	**weekly**	**weekly**	**weekly**
Avg. rating	**4.5**	**3.5**	**4.2**	**3.5**
# Ratings	**155,344**	**2,975,737**	**71,968**	**354,415**

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

What’s Stopping Us?

®

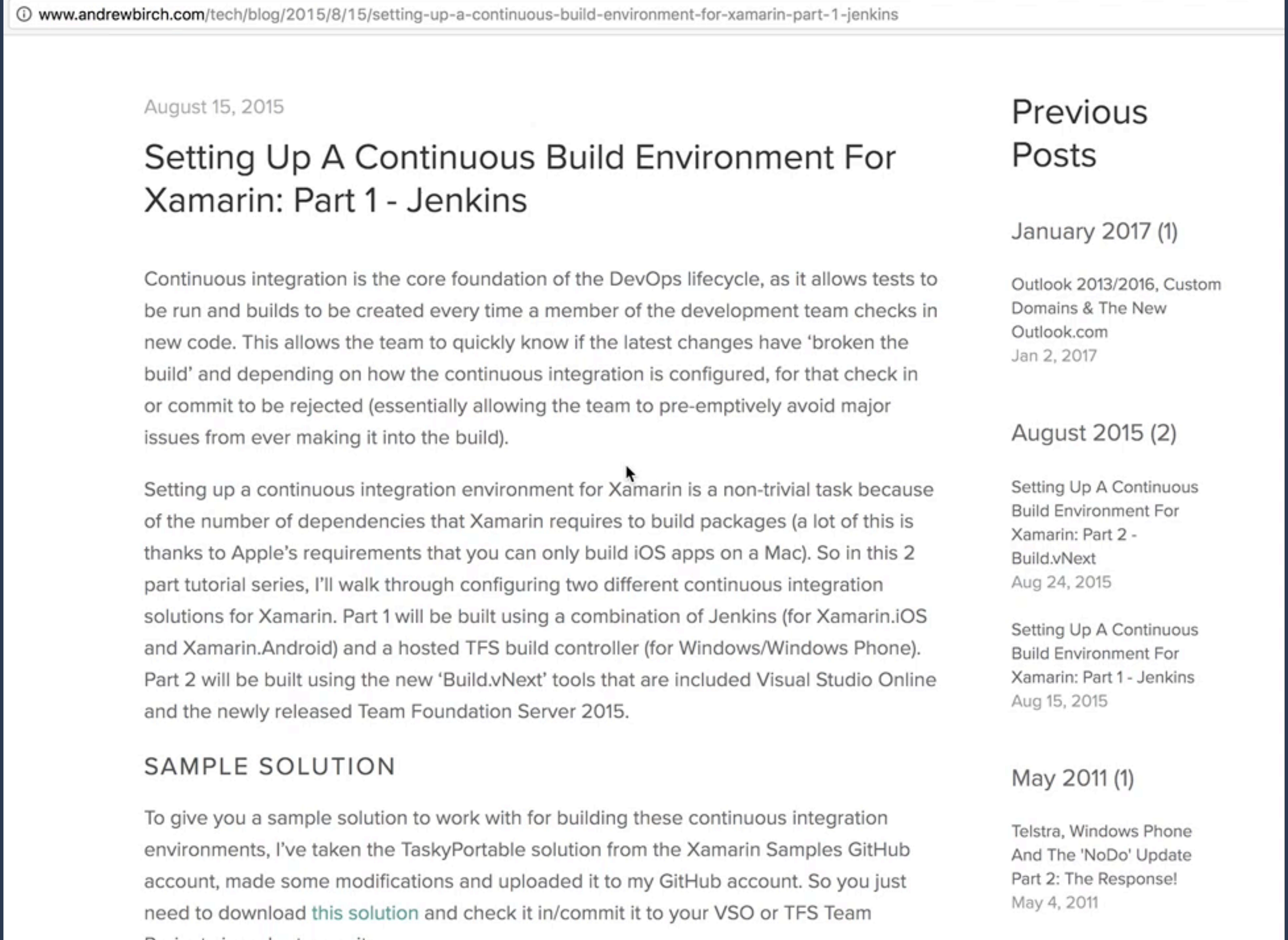
Apple review cycle time



http://appreviewtimes.com/

https://www.macstories.net/news/apple-shortening-app-review-times/ http://daringfireball.net/2016/06/the_new_app_store

Continuous Integration



www.andrewbirch.com/tech/blog/2015/8/15/setting-up-a-continuous-build-environment-for-xamarin-part-1-jenkins

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

Previous Posts

January 2017 (1)

Outlook 2013/2016, Custom Domains & The New Outlook.com
Jan 2, 2017

August 2015 (2)

Setting Up A Continuous Build Environment For Xamarin: Part 2 - Build.vNext
Aug 24, 2015

Setting Up A Continuous Build Environment For Xamarin: Part 1 - Jenkins
Aug 15, 2015

May 2011 (1)

Telstra, Windows Phone And The 'NoDo' Update Part 2: The Response!
May 4, 2011

http://www.andrewbirch.com/tech/blog/2015/8/15/setting-up-a-continuous-build-environment-for-xamarin-part-1-jenkins

CI-As-A-Service?

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

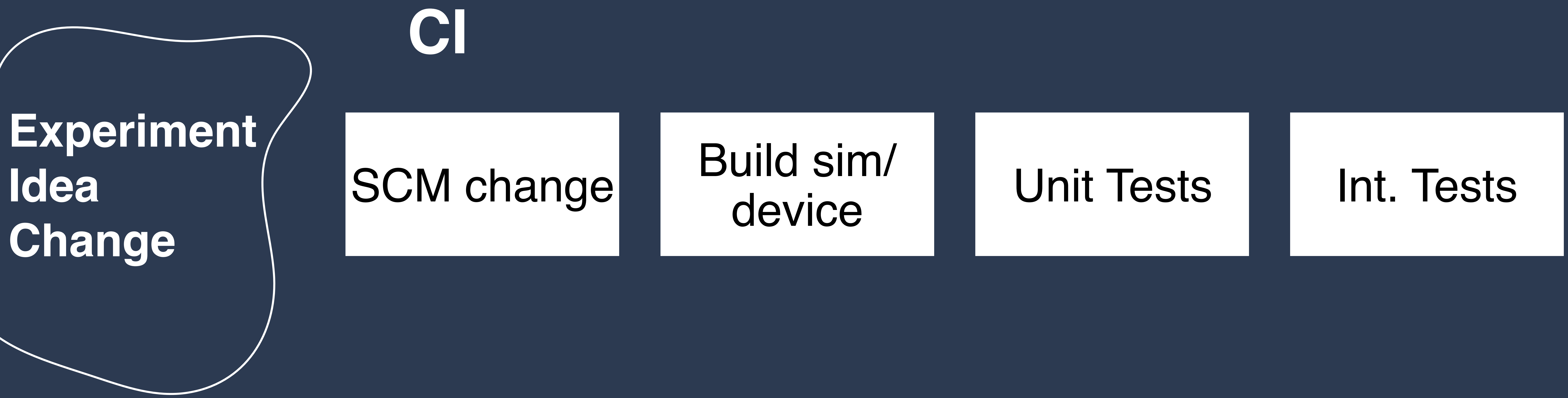
```
//Example in C#
app.Tap ("Help");
app.Tap (e => e.Id ("history-btn"));
app.WaitForElement (e => e.Text ("Ink"));
app.Screenshot ("View the purchasing history");
```

Demo: Xamarin.UITest

(yes, it works with all apps, not just Xamarin :)

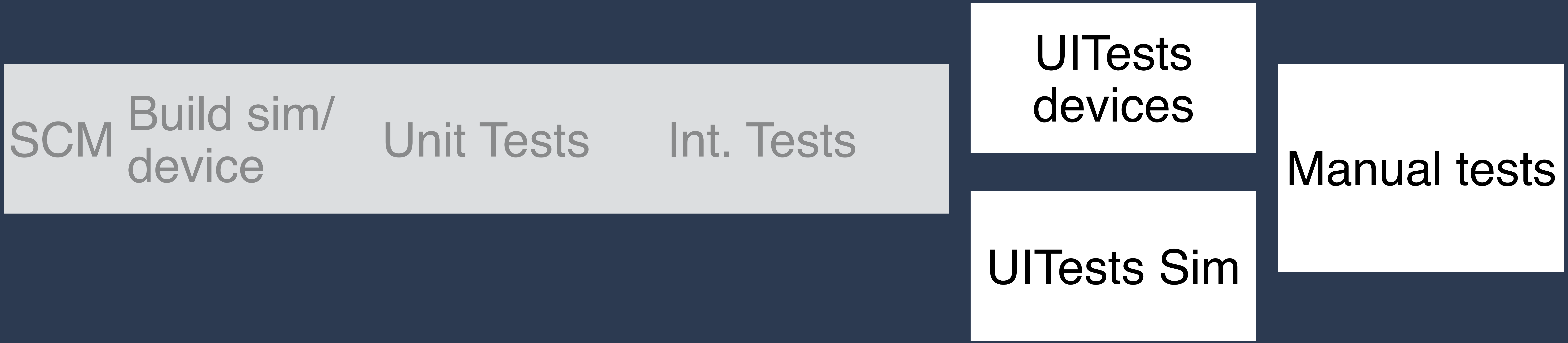
How?

What are all the steps needed to ship?



What else?

Continuous Delivery: Anatomy of the Deployment Pipeline
http://www.informit.com/articles/article.aspx?p=1621865&seqNum=8


SCM
Build sim/ device
Unit Tests
Int. Tests
UITests devices
UITests Sim
Manual tests

What else?

UITests, devices
UITests, sims
Manual Tests
Sign for app-store distribution
Generate Updated Screenshots
Release notes / metadata
iTunes Connect
Submit for review
Apple
Publish
User updates
Feedback
New Ideas
Experiments

What are all the steps needed to ship?

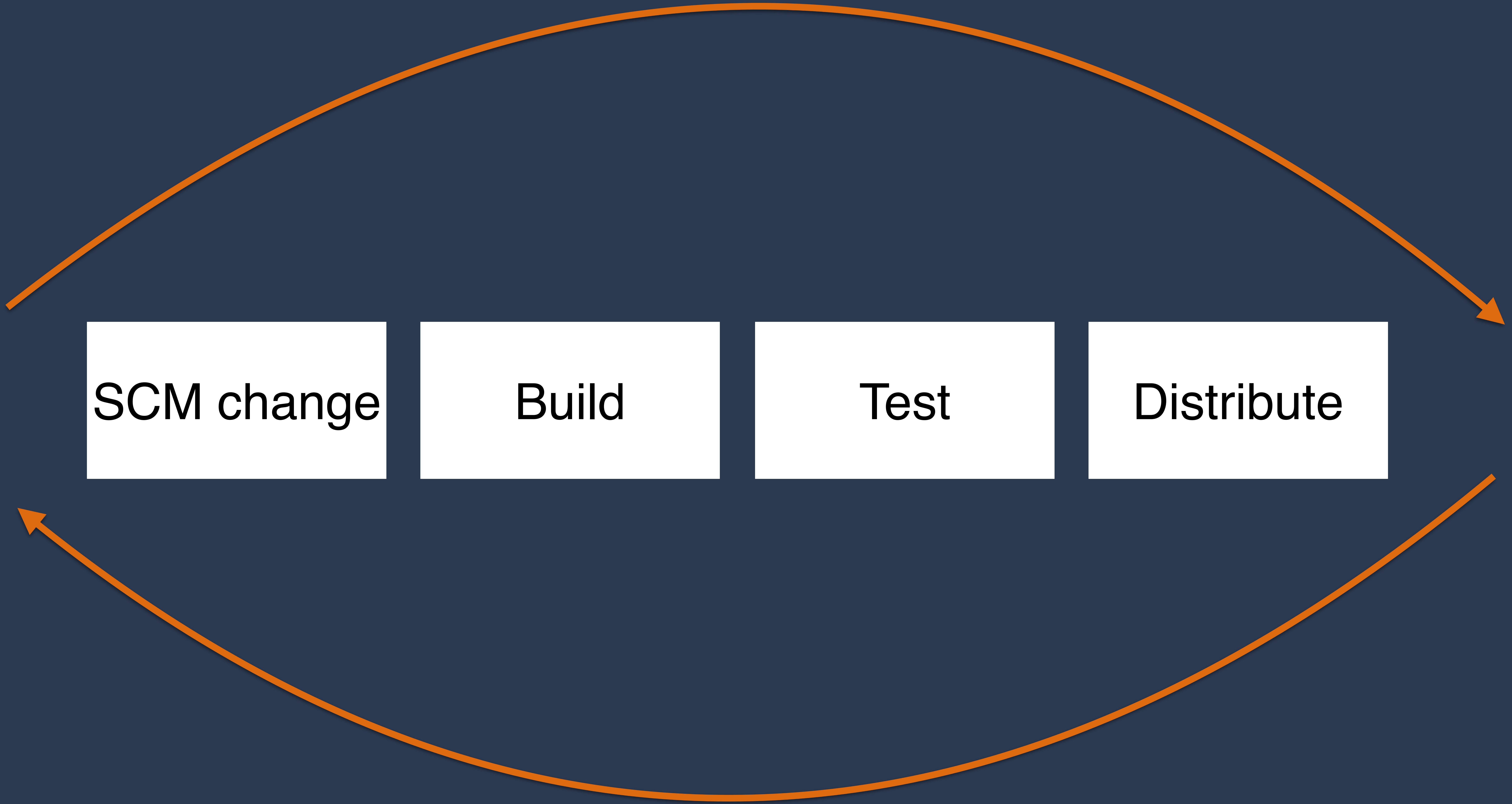
Experiment
Idea
Change
SCM
Build
Unit
Int.
UI: Sim
UI: Dev
Manual
Resig
img
meta
Upl
Sub
Appl
Publish
User
Feedback
New Ideas
Experiments

DAUNTING

What are all the steps needed to ship?

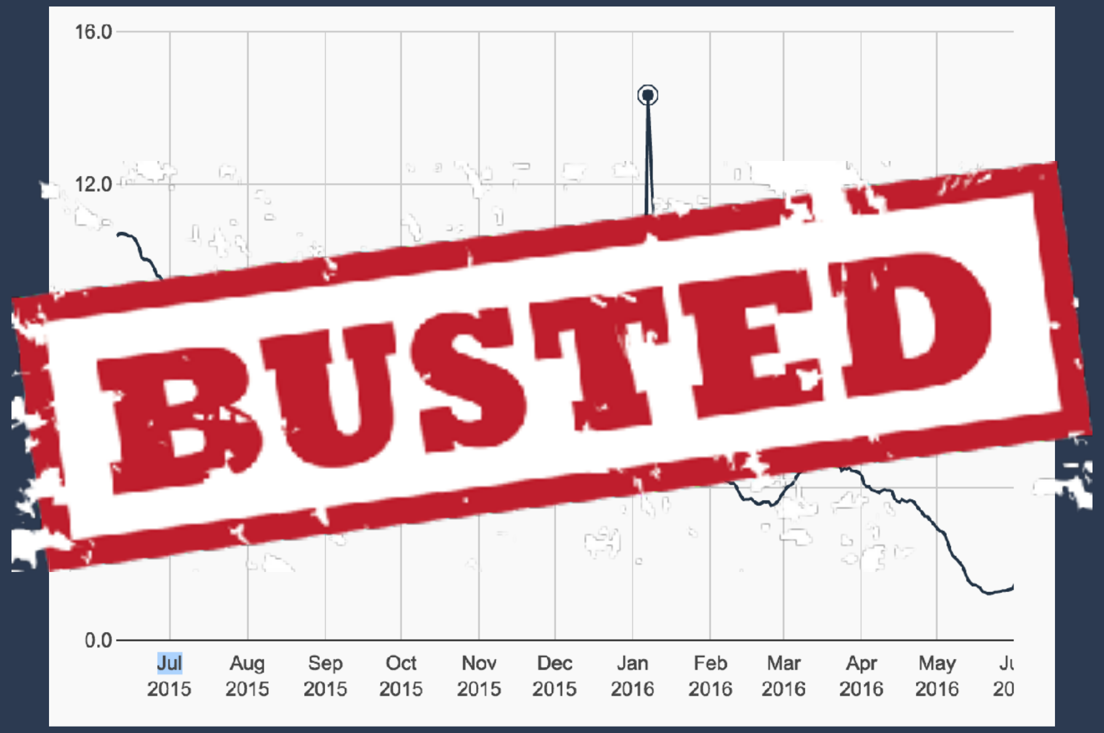
Experiment
Idea
Change
SCM
Build
Unit
Int.
UI: Sim
UI: Dev
Manual
Resig
img
meta
Upl
Sub
Appl
Publish
User
Feedback
New Ideas
Experiments

Build-Test-Distribute with Mobile Center



Mobile Center

https://aka.ms/mobilecenter



www.andrewbirch.com/tech/blog/2015/8/15/setting-up-a-continuous-build-environment-for-xamarin-part-1-jenkins

August 15, 2015

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

Continuous integration is the core foundation of the DevOps lifecycle
be run and builds to be created every
new code.

Windows/Windows Phone).
Build.vNext' tools that are included Visual Studio Online
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

Previous Posts

Setting Up A Continuous Build Environment For Xamarin: Part 1 - Jenkins
Aug 15, 2015

May 2011 (1)

Telstra, Windows Phone And The 'NoDo' Update Part 2: The Response!
May 4, 2011

Release More Often!

$$\begin{tikzcd}
\text{yearly} \arrow[r, bend left] & \text{quarterly} \arrow[r, bend left] & \text{monthly} \arrow[r, bend left] & \text{bi-weekly} \arrow[r, bend left] & \text{weekly}
\end{tikzcd}$$

https://aka.ms/mobilecenter

goto; copenhagen

GOTO Copenhagen 2017

Conference Oct. 1-3, 2017

Please

Remember to rate this session

Thank you!

Follow us @gotocph