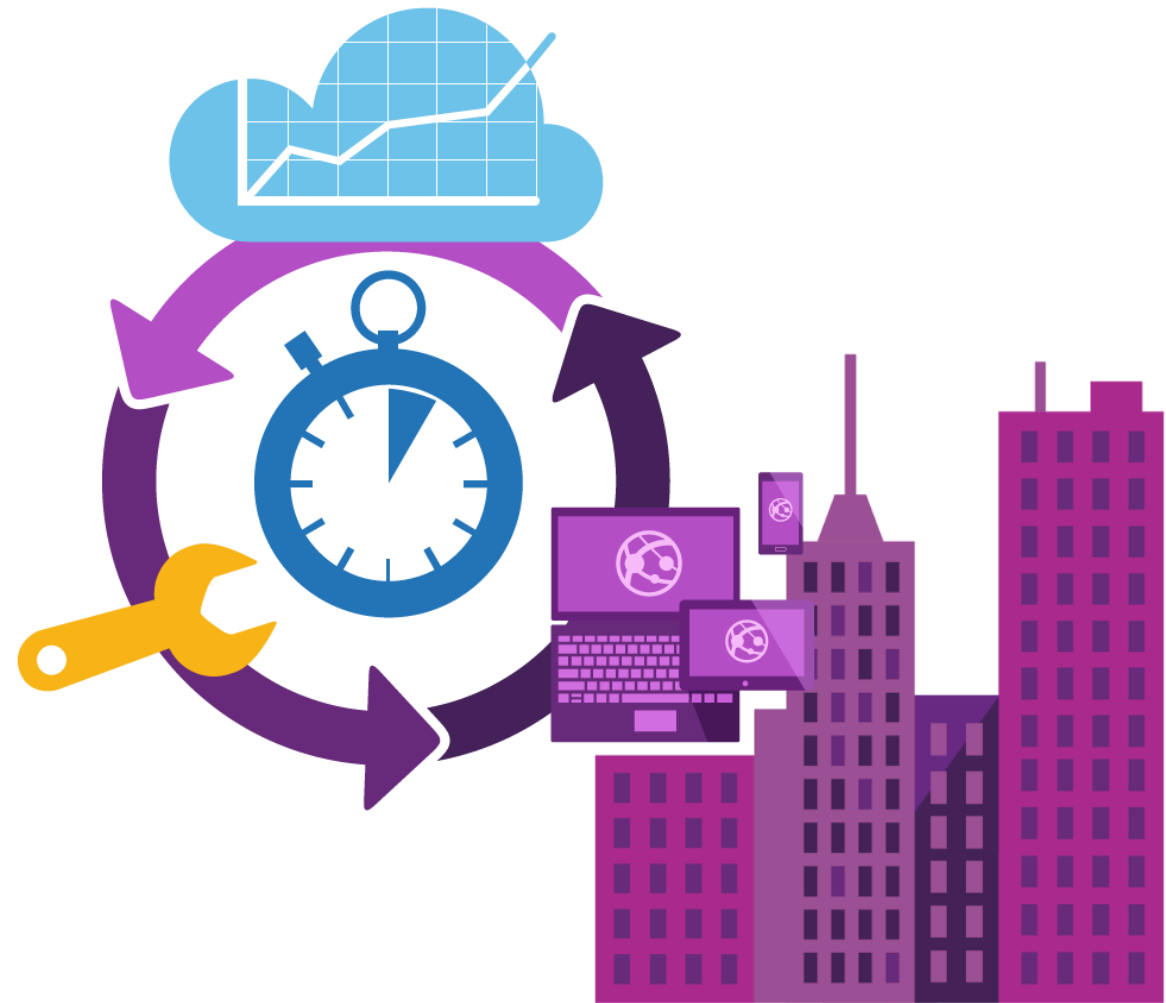


DevOps at Scale: A True Story

Davide Benvegnù

Microsoft MVP – Visual Studio ALM

Founder, Director & CTO – DBTek Ltd



@bout Me

Daide Benvegnù

Director & CTO – DBTek Ltd

Former International Development Manager – Aruba.it

Microsoft **MVP** in Visual Studio ALM

Speaker - Trainer

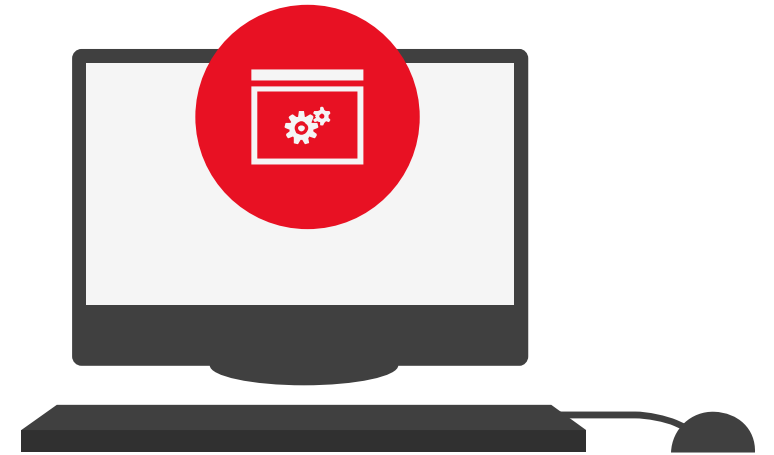
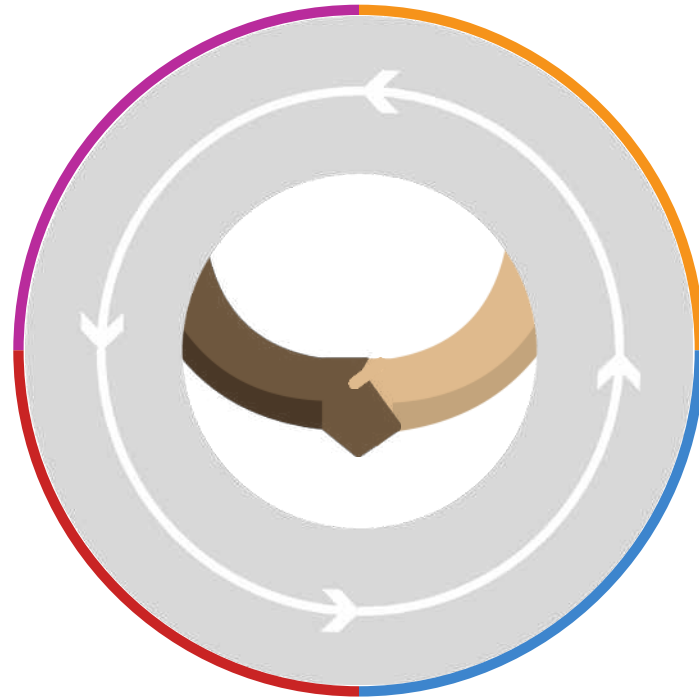
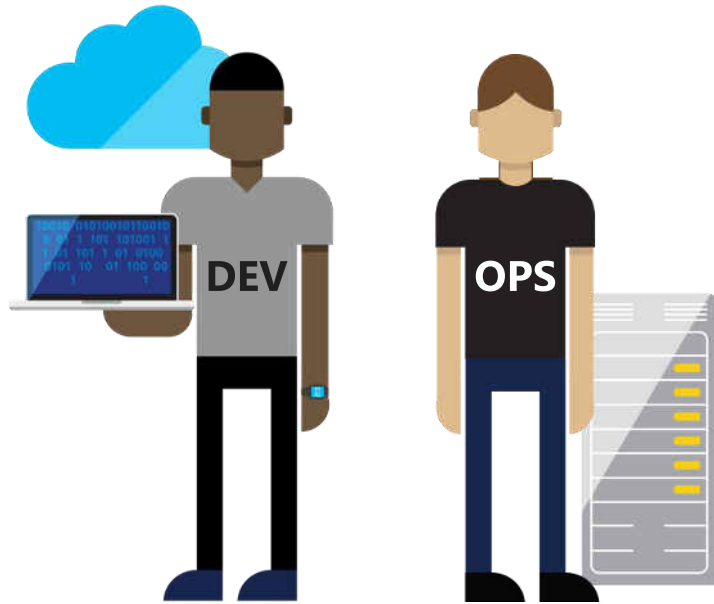
Community Contributor (1 in HK, 2 in Italy)

13+ years experience in IT, 10+ years experience in ALM / DevOps

@davidebenvegnu – www.dbtek.com.hk – davide@dbtek.com.hk



DevOps: the three stage conversation

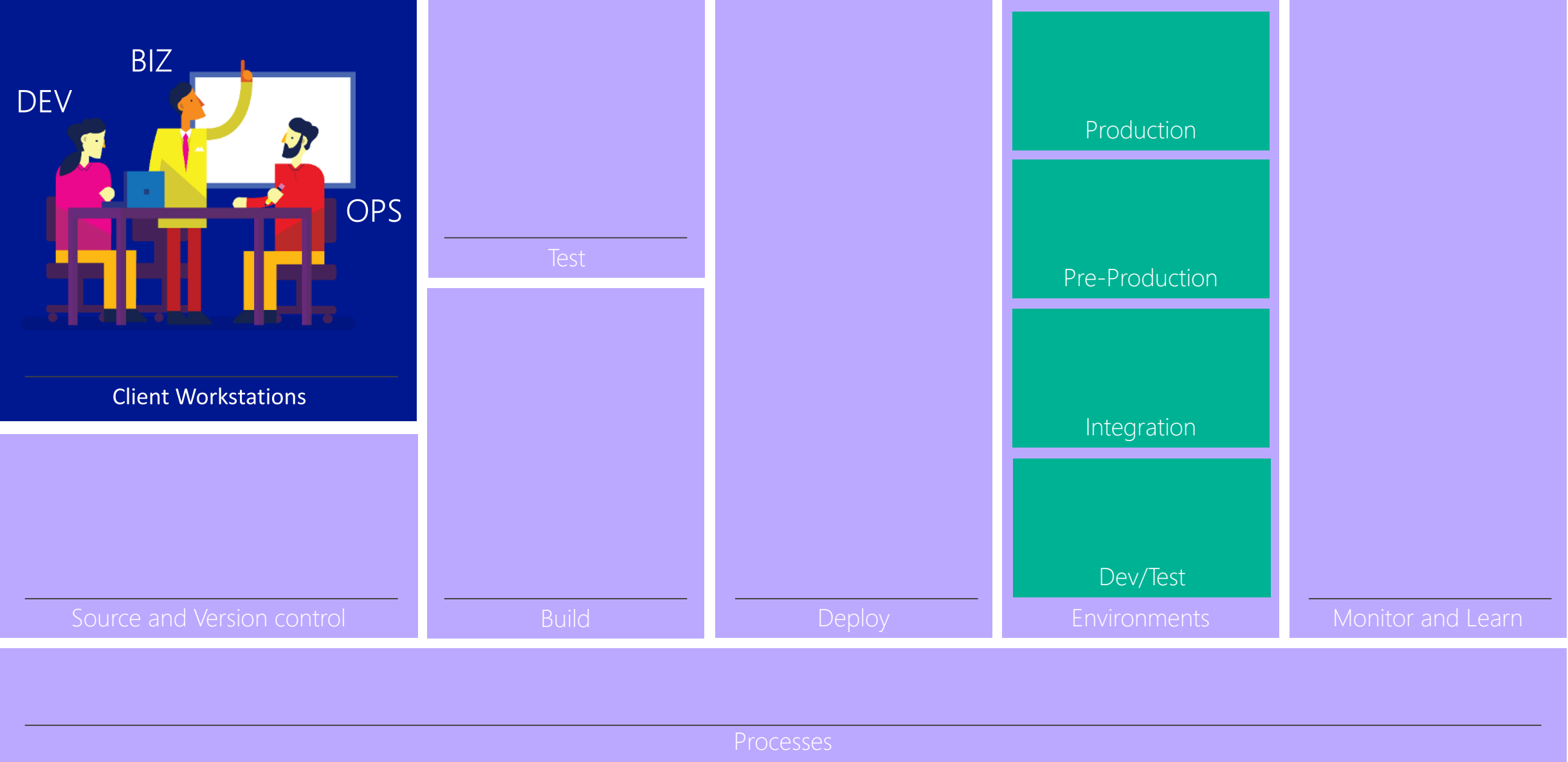


1 People

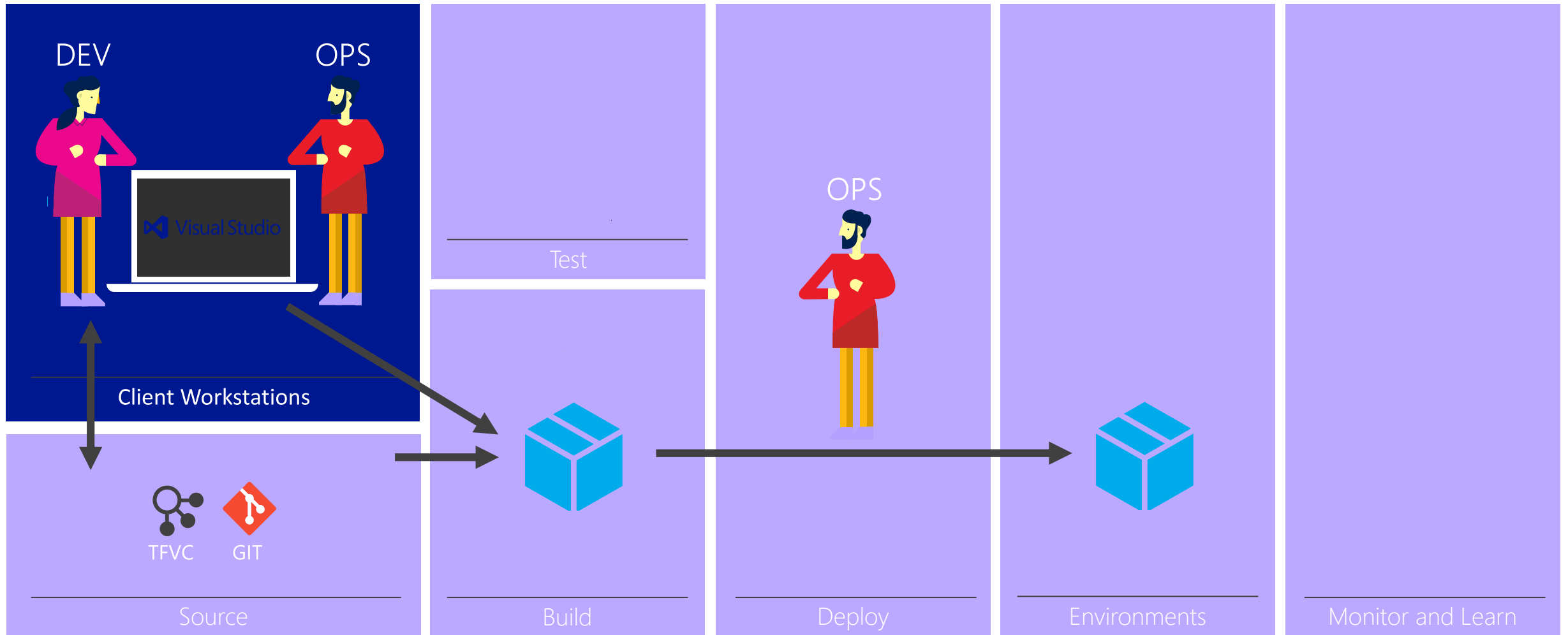
2 Process

3 Tools

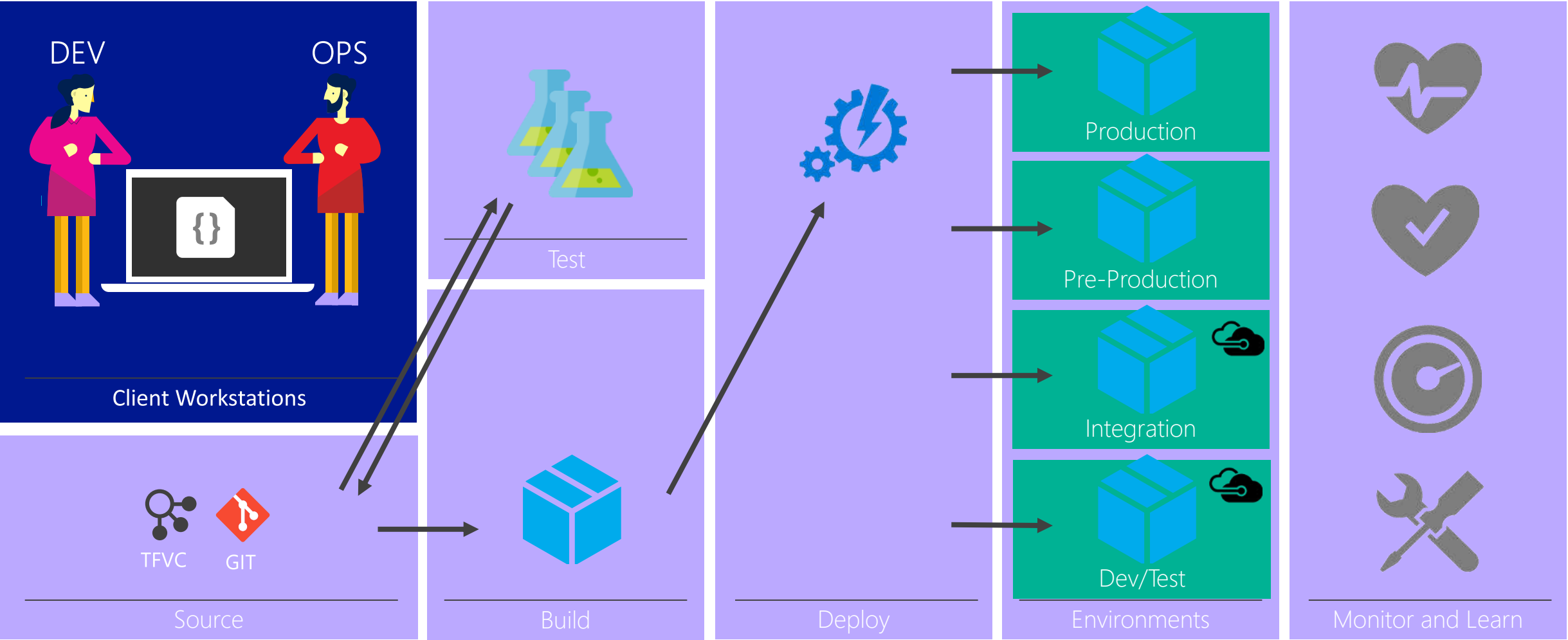
DevOps Framework



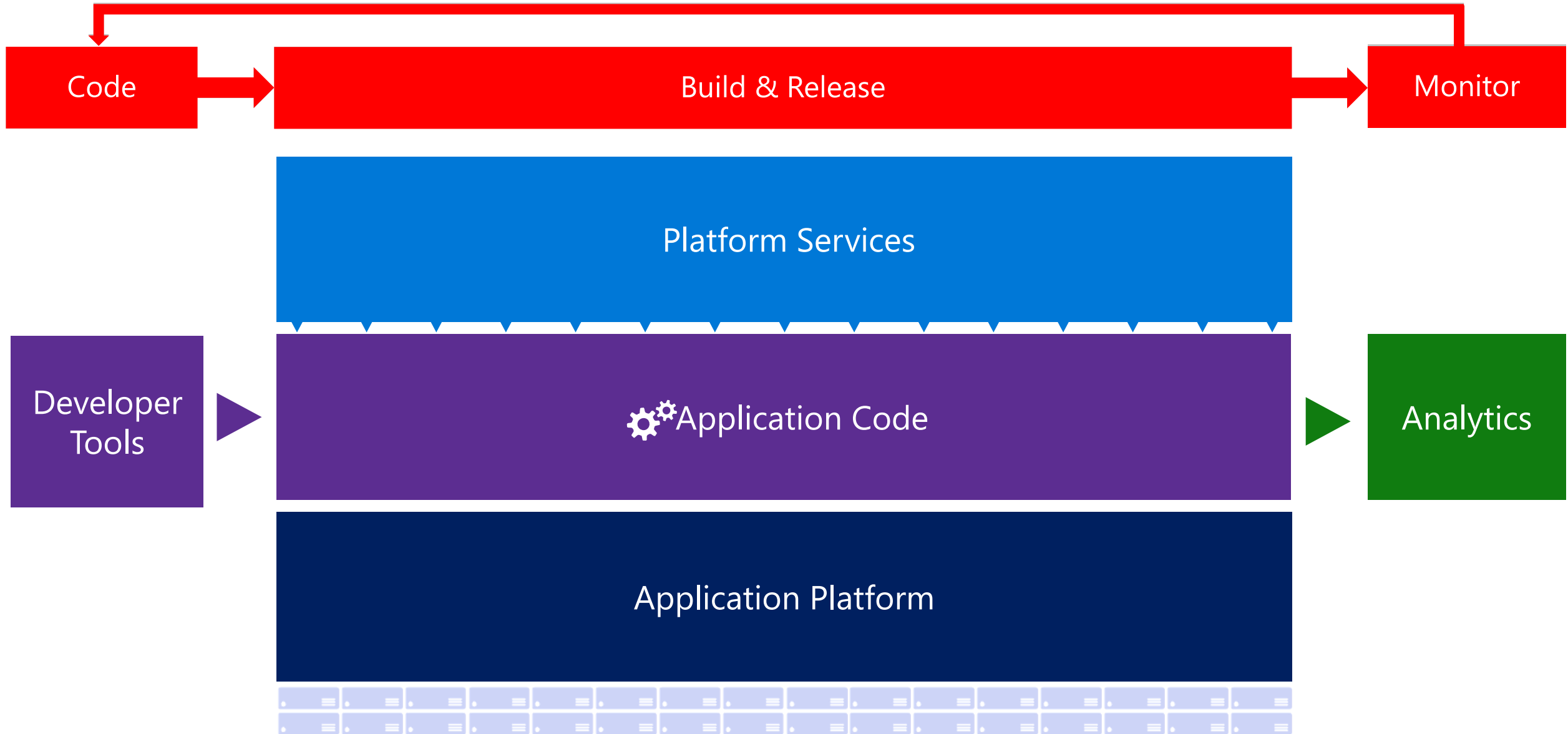
Manual Builds and Deploy



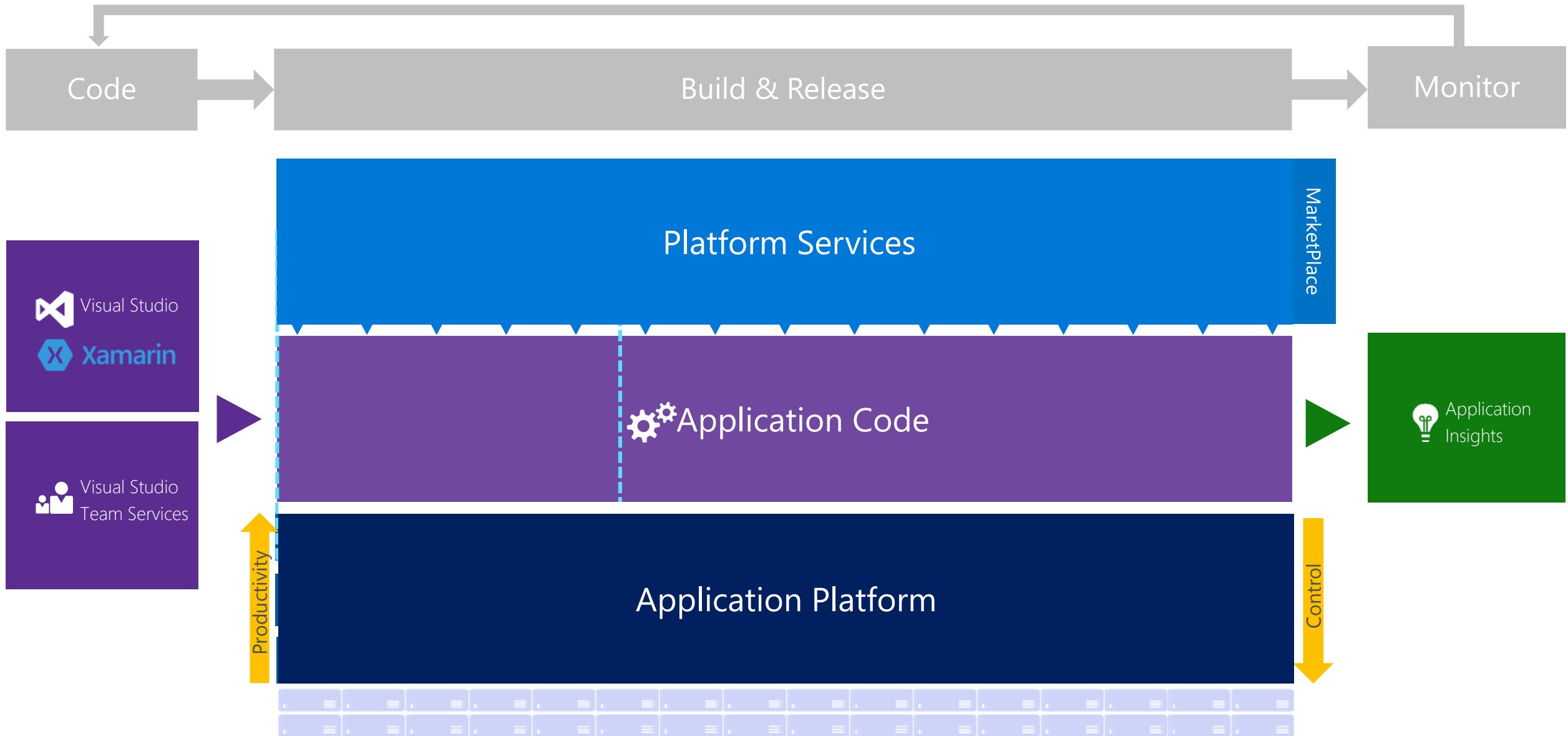
Continuous Integration and Deployment



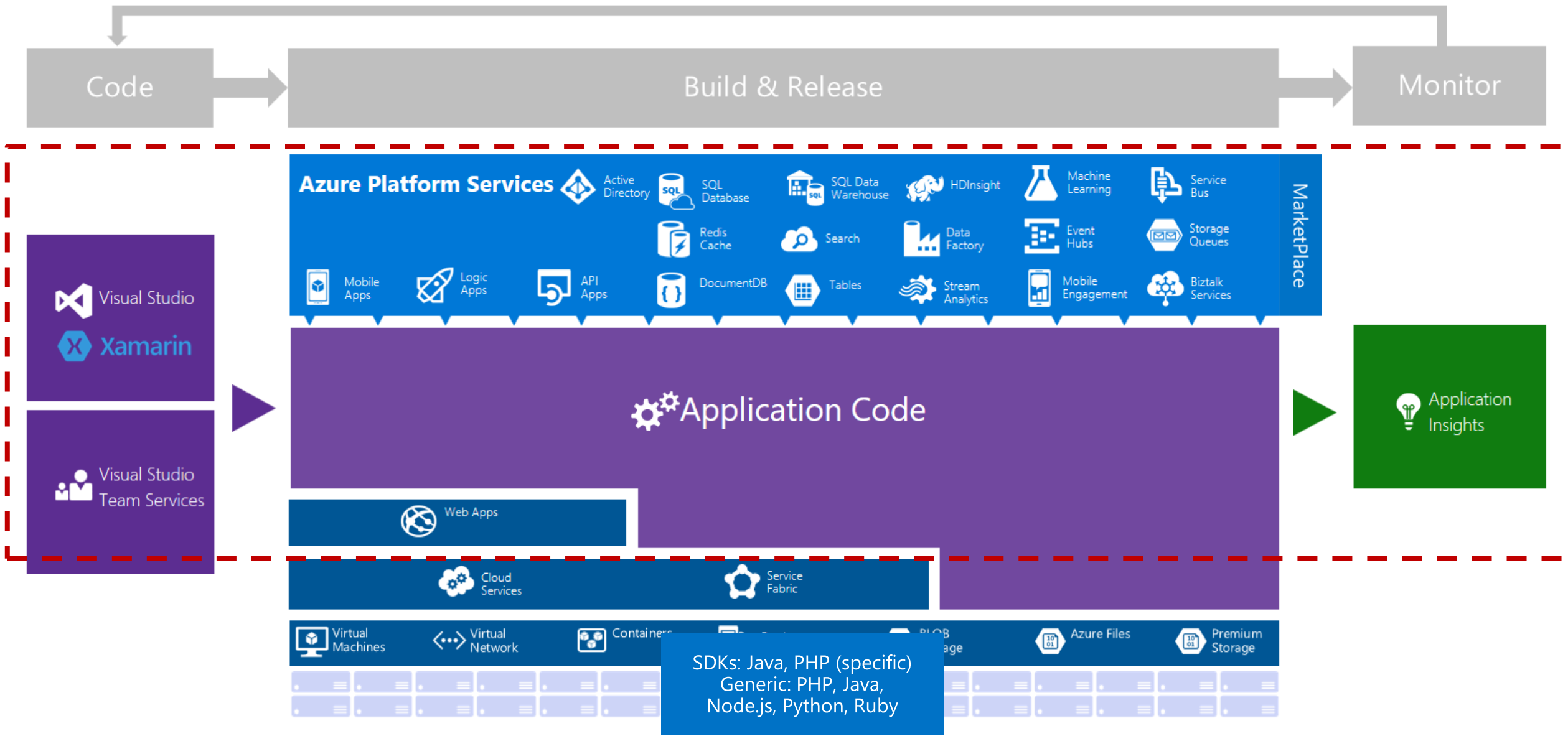
Microsoft's Cloud Development Platform



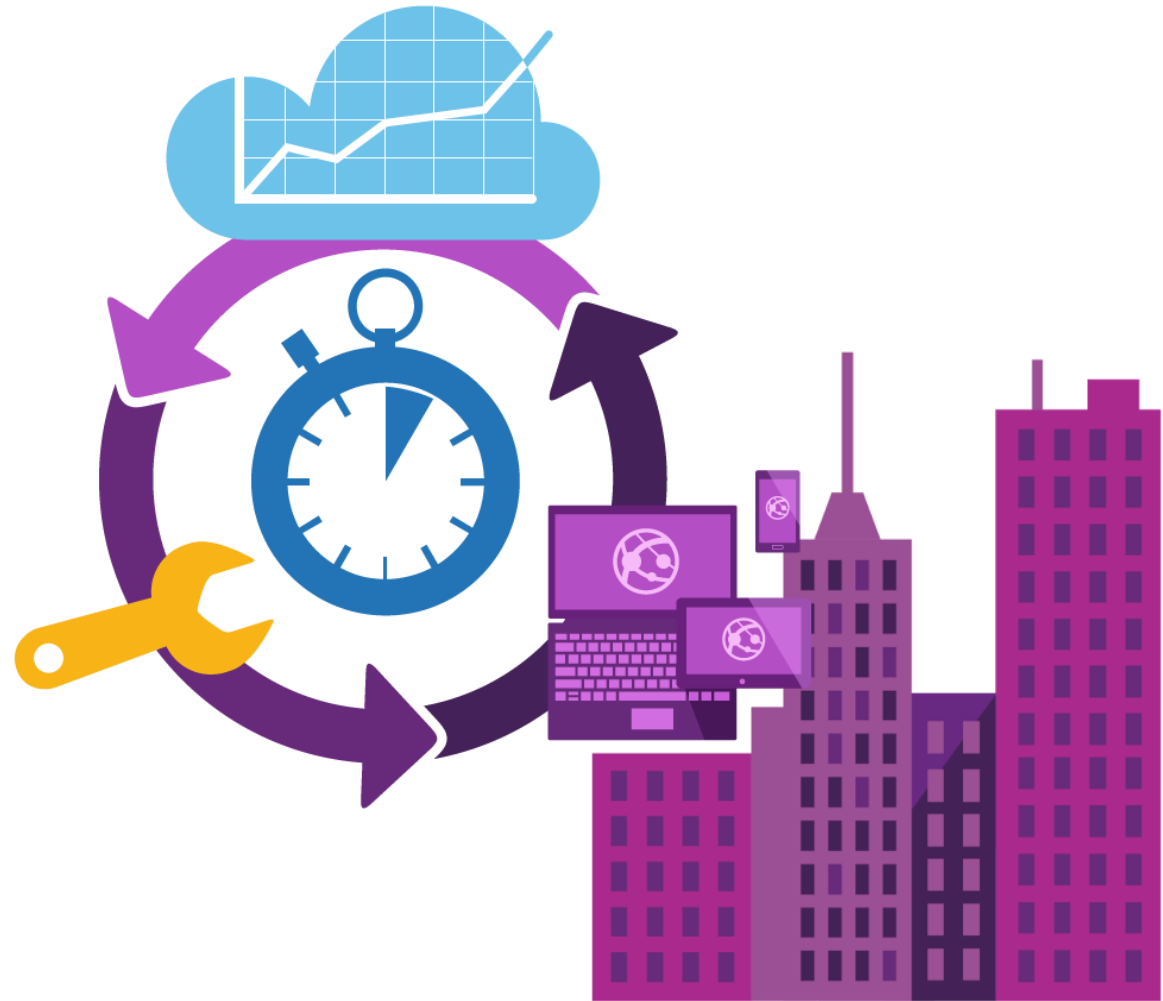
Microsoft's Cloud Development Platform



Enterprise Web & Mobile Dev Platform



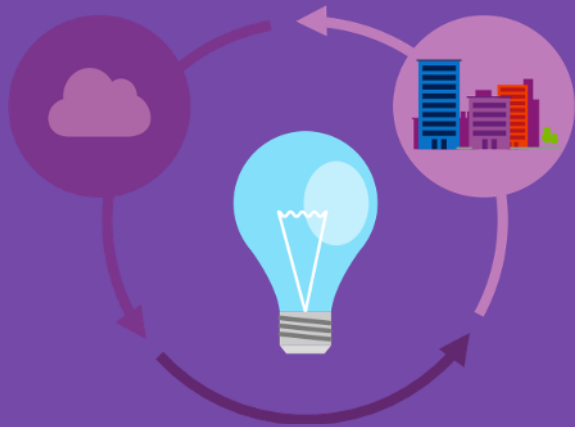
The Microsoft DevOps solution Overview



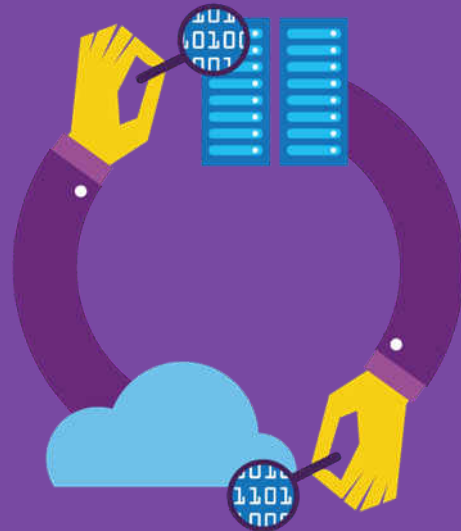
The Microsoft DevOps solution

An integrated, end-to-end solution for teams of any size to design, build and manage enterprise solutions and cross-platform mobile business apps.

Shorten cycle times
and deliver value faster



Improve quality
and availability



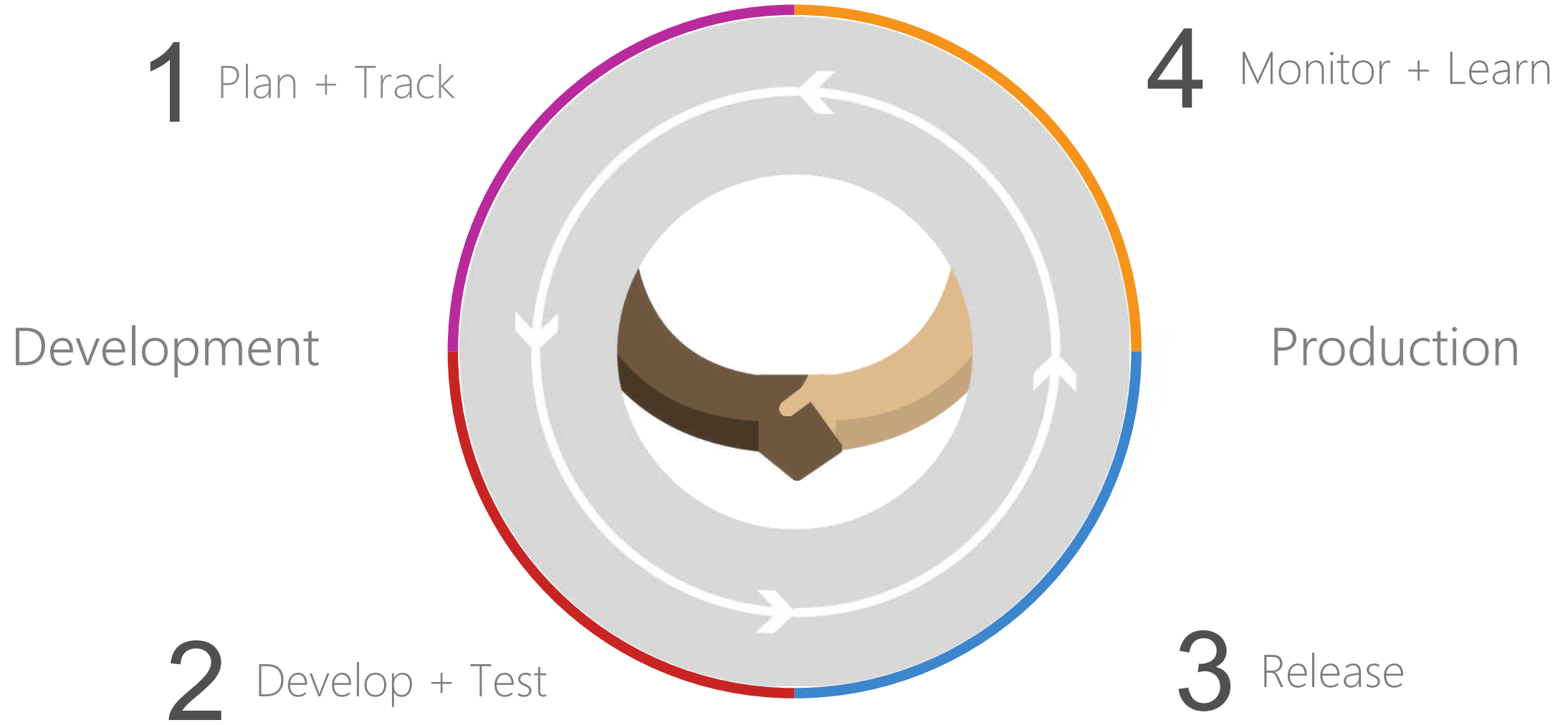
Optimize resources
and eliminate waste



Deliver mobile apps with
digital-era velocity



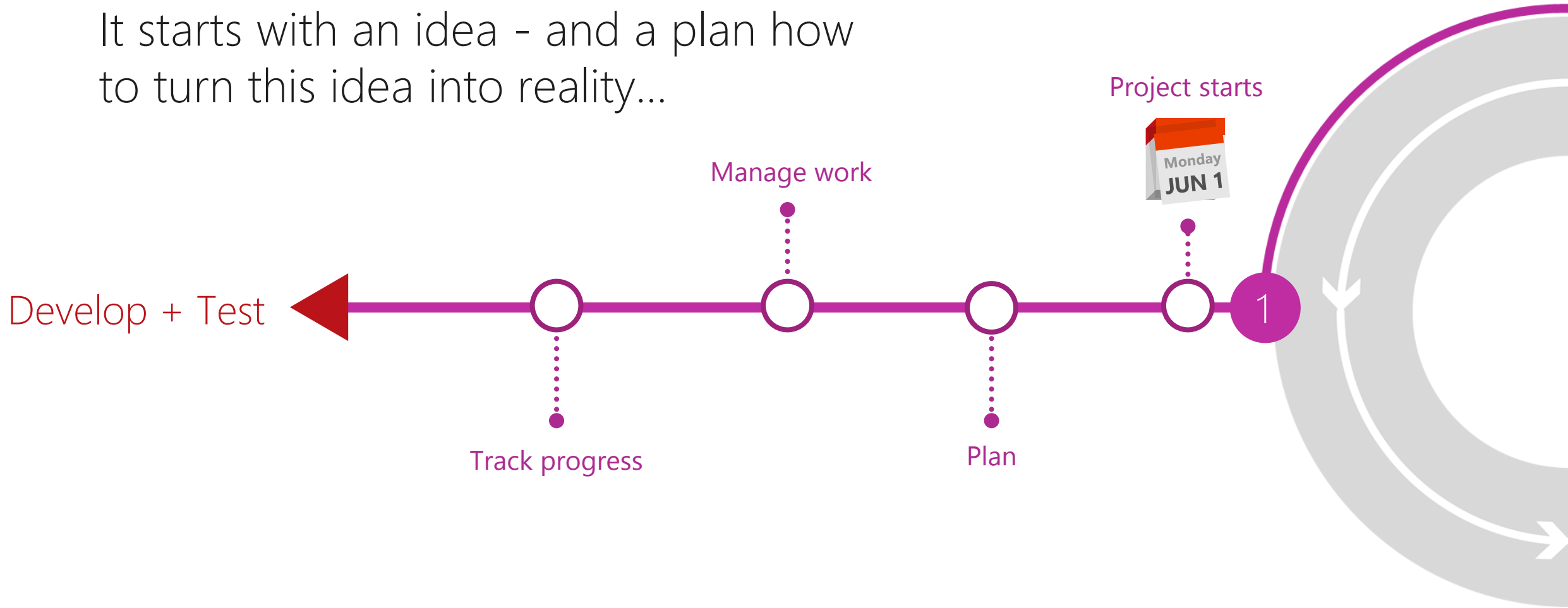
End-to-end DevOps





Plan + Track

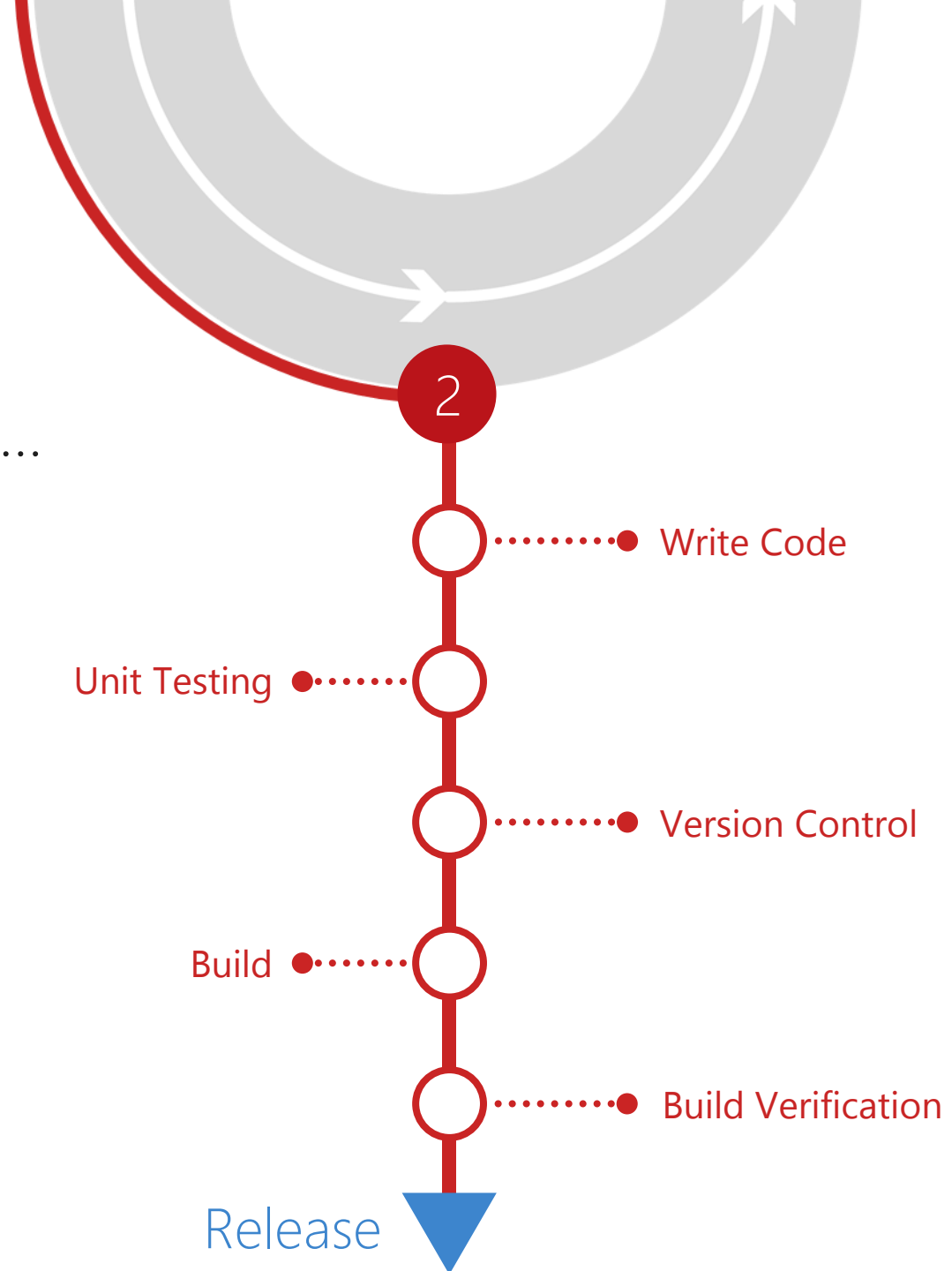
It starts with an idea - and a plan how to turn this idea into reality...





Develop + Test

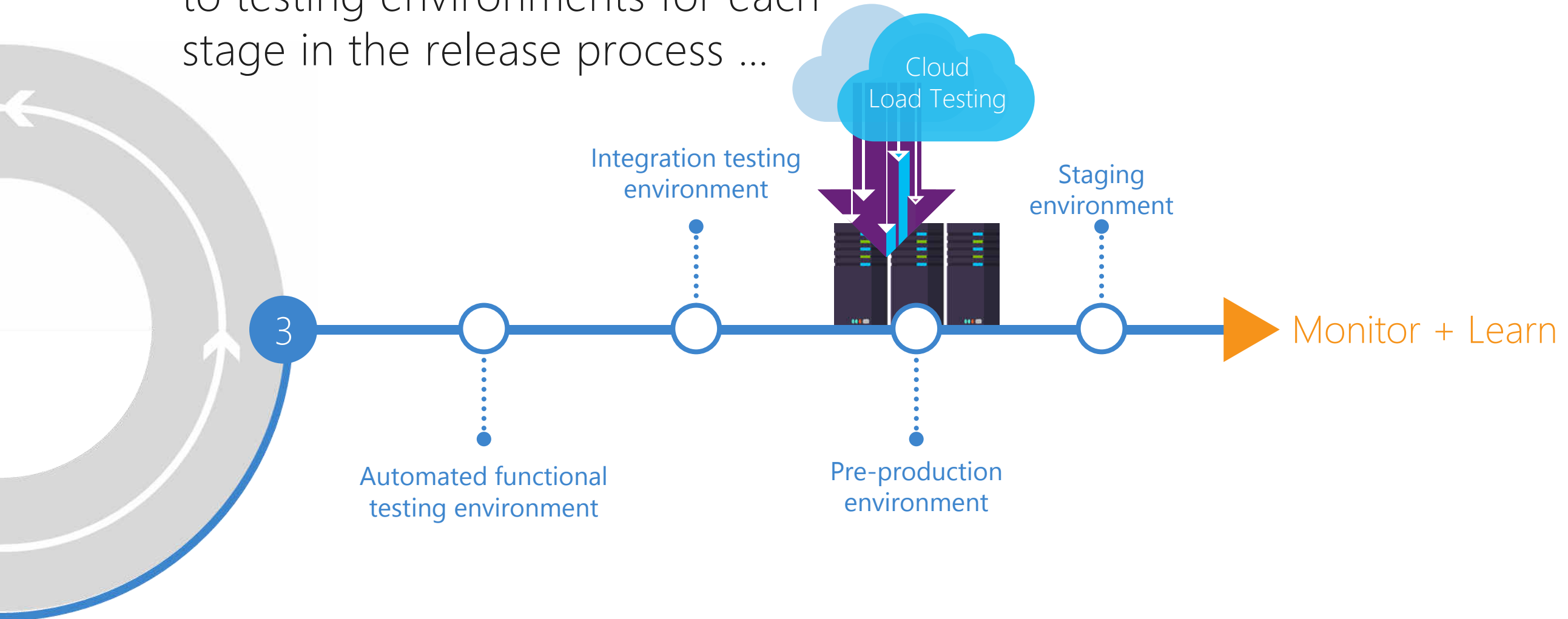
After the iteration starts, developers turn great ideas into features and functionality ...





Release

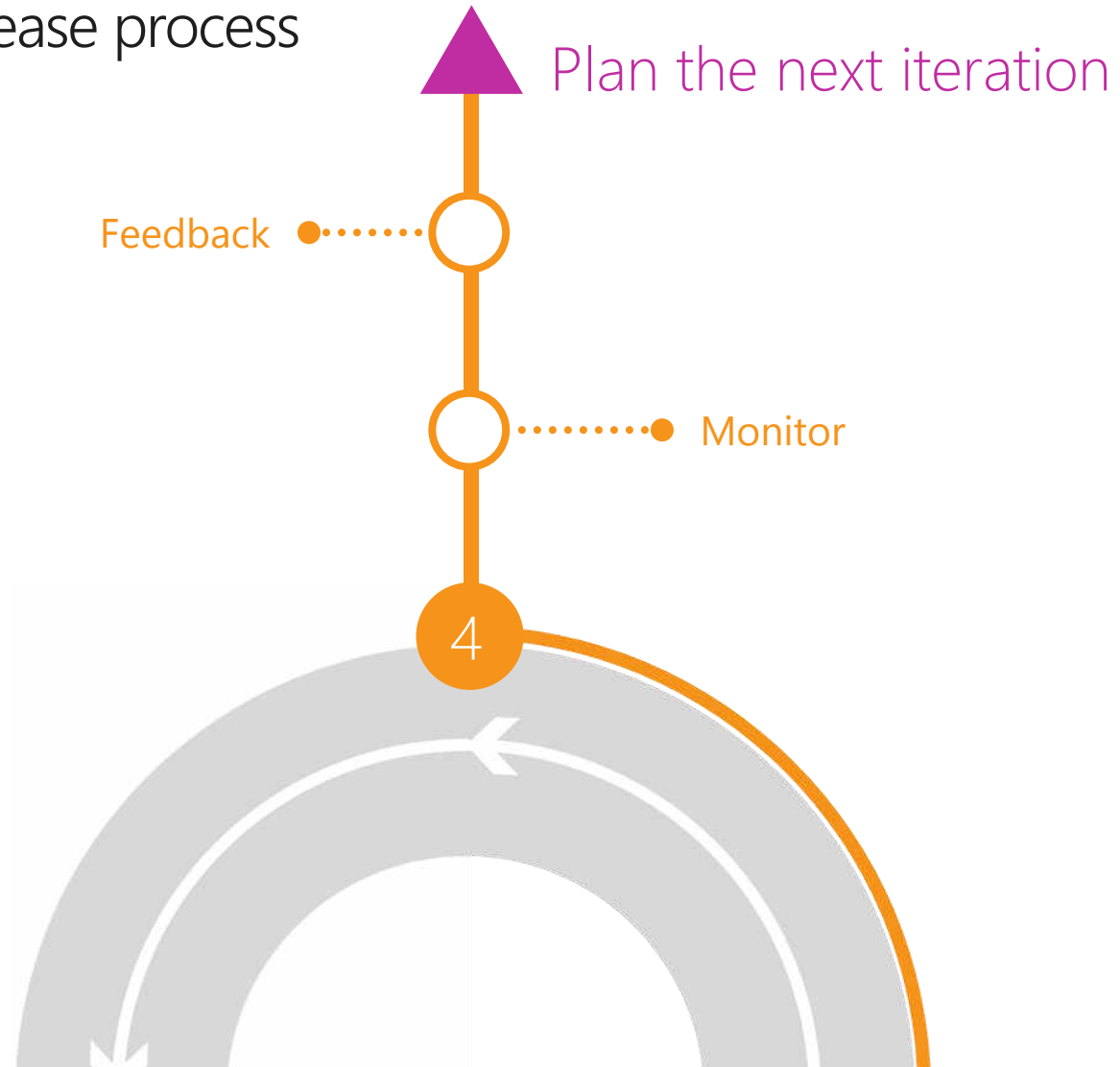
When all tests pass, the build is deployed to testing environments for each stage in the release process ...





Monitor + Learn

When all tests pass, the build is deployed to testing environments for each stage in the release process

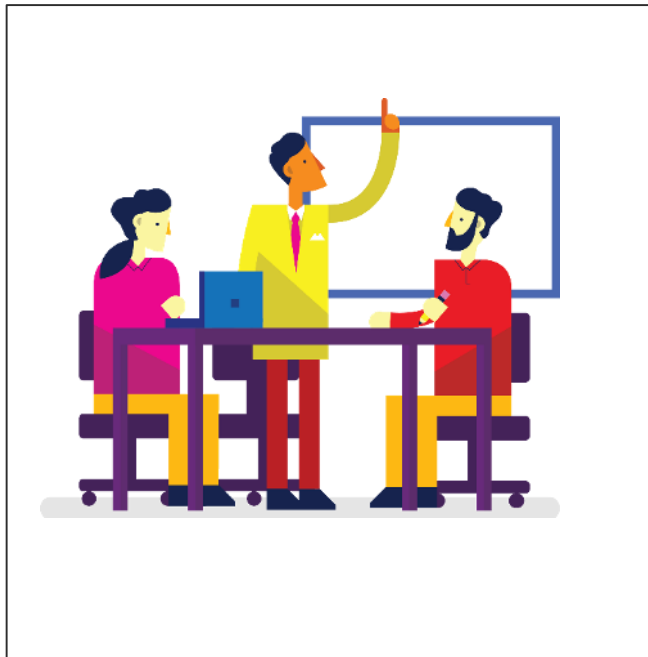


Real example: *Aruba.it*

- 6 developers teams: Italy, Czech Rep., Poland, Ukraine
- 1 QA team
- 4 main products: Cloud, Hosting, Email, VPS

	<u>Pre MS DevOps</u>	<u>Post MS DevOps</u>
Avg project time:	~ 3 months	~ 1 month
Avg simultaneous projects:	5	10
Successful deploys to Prod:	43%	88%
Number of testers:	12 (QA)	12 (Sprint + QA)
Number of «deployers»:	6	2

Real Example: HOW?



Auto UI test +



Test Manager

Test

Continuous
Integration

TFS



Build

Release
Management



Continuous
Deployment



Deploy

Production

Staging

Integration

Dev/Test

Environments



App Insights:
Health &
Performance
Analysis

Monitor and Learn

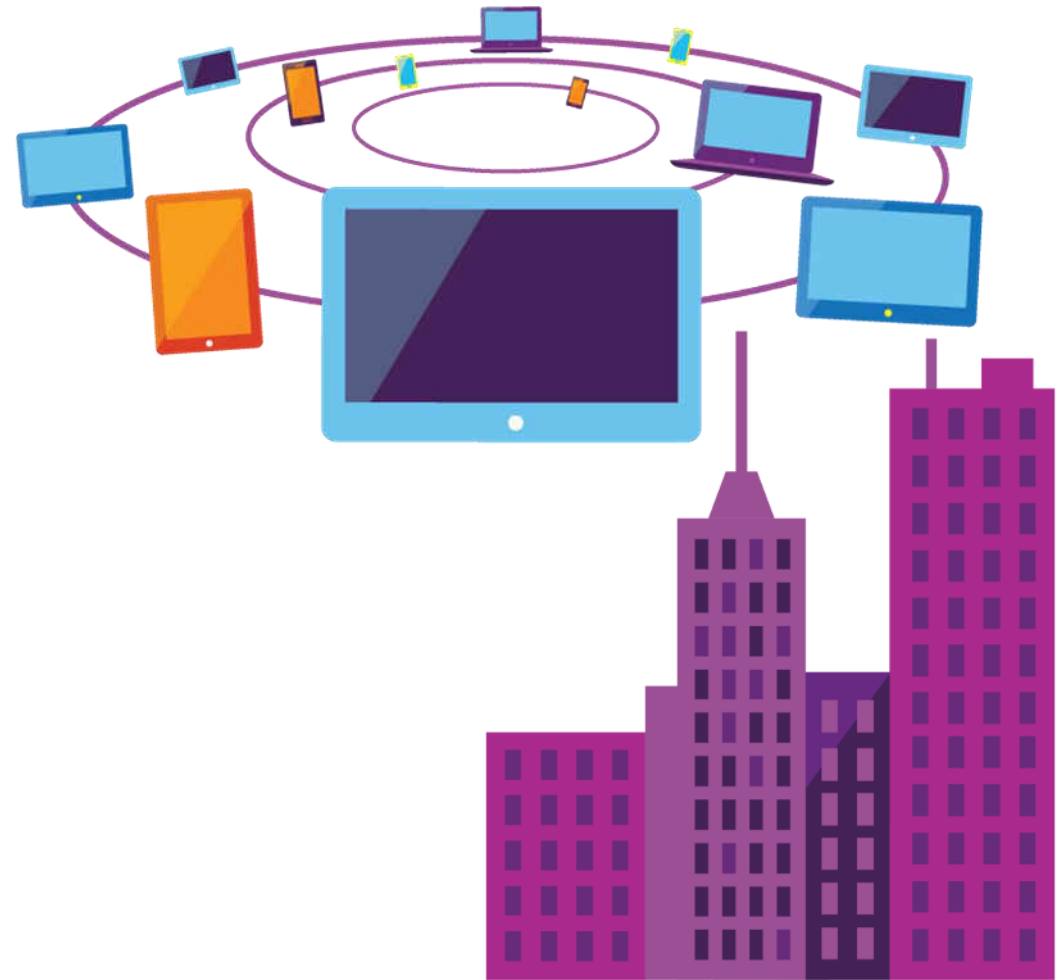
SVN → TFVC

Source and Version control

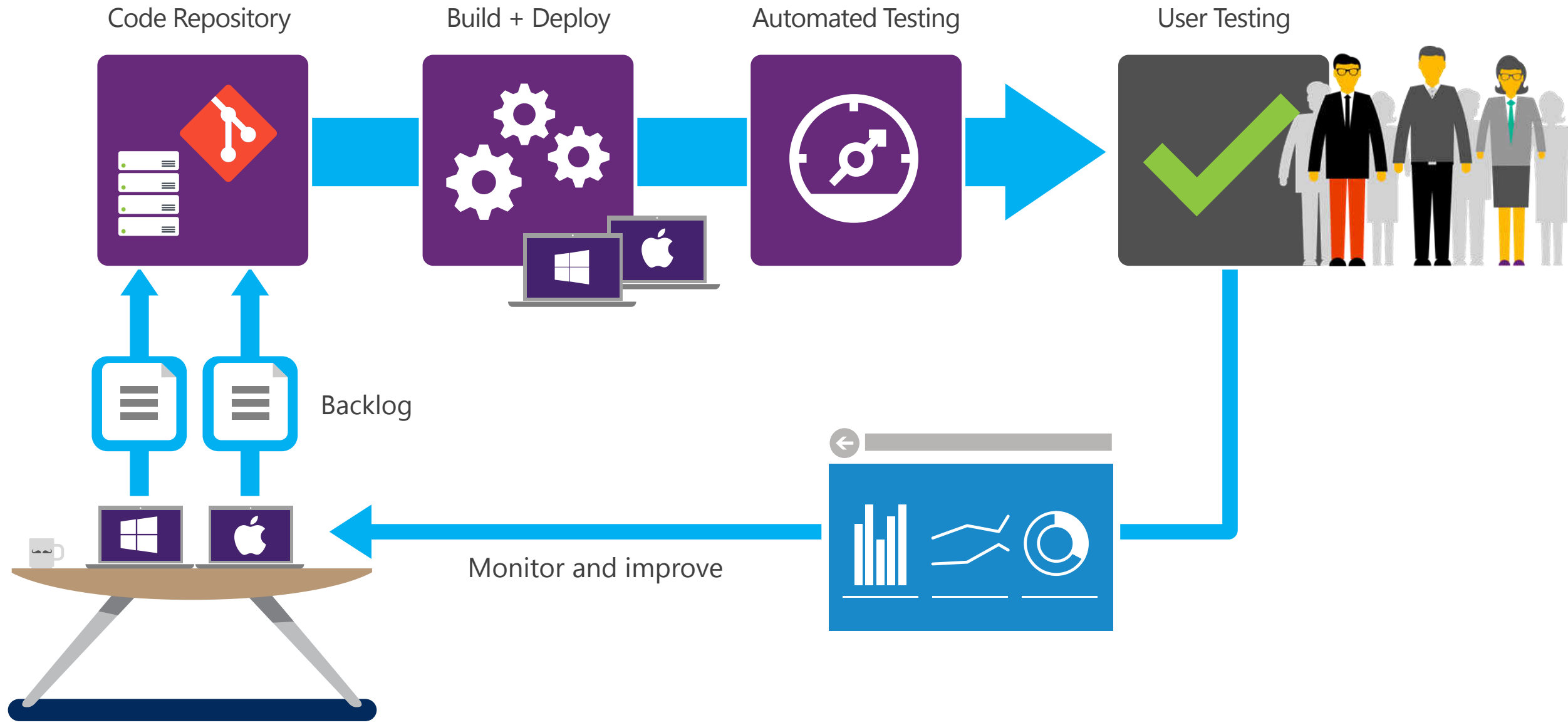
Agile management with TFS (Scrum, Kanban, Task)

Processes

The Microsoft DevOps Solution
DevOps for the digital era



Mobile app CI and CD



Xamarin Test Cloud

2,000+ devices

Automate your app testing and run them on over 2,000 different real devices. Test everything users do, as well as any performance problems with step-by-step memory and performance tracking.

Real devices, real quality

- Ensure higher quality by testing on real devices
- Automate app testing on over 2,000 real devices
- Use C#, Ruby, or Cucumber to build automated tests
- Integrates with any continuous integration system

The screenshot displays the Xamarin Test Cloud interface. At the top, the browser address bar shows 'testcloud.xamarin.com'. The page header includes 'Xamarin test cloud', navigation links for 'Xamarin CRM', 'master', and a timestamp 'Aug 23, 2015 10:03:59 PM'. The main content area is divided into two sections. On the left, an 'Overview' sidebar lists test results with green checkmarks for 'Customers tests' (Check Customer Details, Check Customer Navigation, Check Customer Phone, Investigate Customer Page) and 'Pages tests' (Add an item). A detailed view of the 'Add an item' test is shown, listing steps: 'First I launch the app', 'Then I tap 'Sales'' (highlighted), 'Then I tap 'Add'', 'Then I choose the first result', 'Then I set the title and description', 'Then I tap 'Save'', and 'Then I go back'. On the right, a grid of 12 device thumbnails is displayed, each with a label: 'Apple iPhone 5C iOS 8.2', 'Apple iPhone 5 iOS 8.3', 'Apple iPhone 5C iOS 8.3', 'Apple iPhone 6 iOS 8.2', 'Apple iPhone 5S iOS 8.1.3', 'Apple iPhone 6 iOS 8.1.3', 'Apple iPhone 5 iOS 7.1.1', 'Apple iPhone 5C iOS 7.1.1', 'Apple iPhone 5S iOS 7.1.1', and three more iPhone 5S devices on iOS 7.1.1. A red lightning bolt icon and the text 'ADD AN ITEM Then I tap 'Sales'' are visible at the top of the device grid.

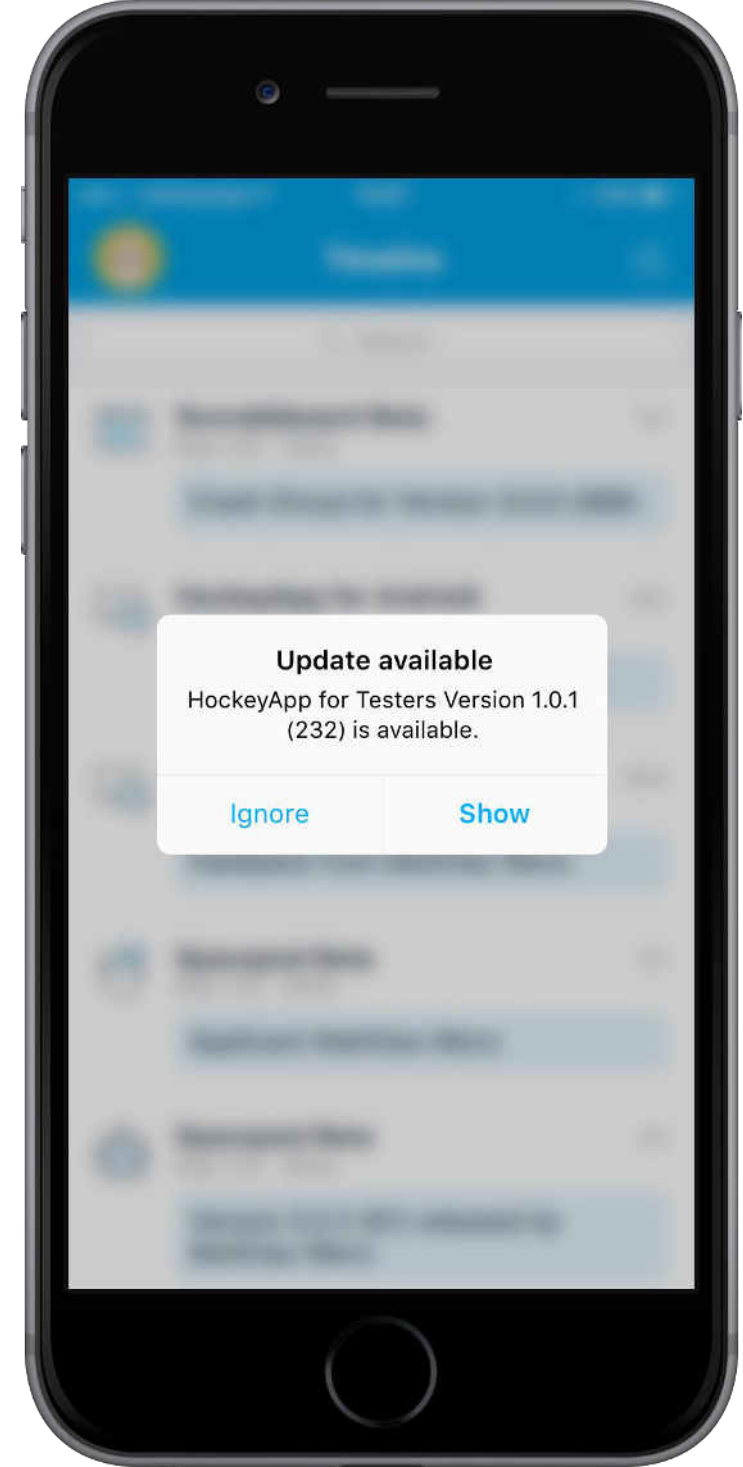
HockeyApp Beta Distribution

In-house App Store for testers

Upload beta versions of your application to the HockeyApp store to allow testers to install and test beta versions on actual devices.

Deployment made easy

HockeyApp's desktop application automatically tracks all necessary information about your latest build in order to make uploading beta versions easy.



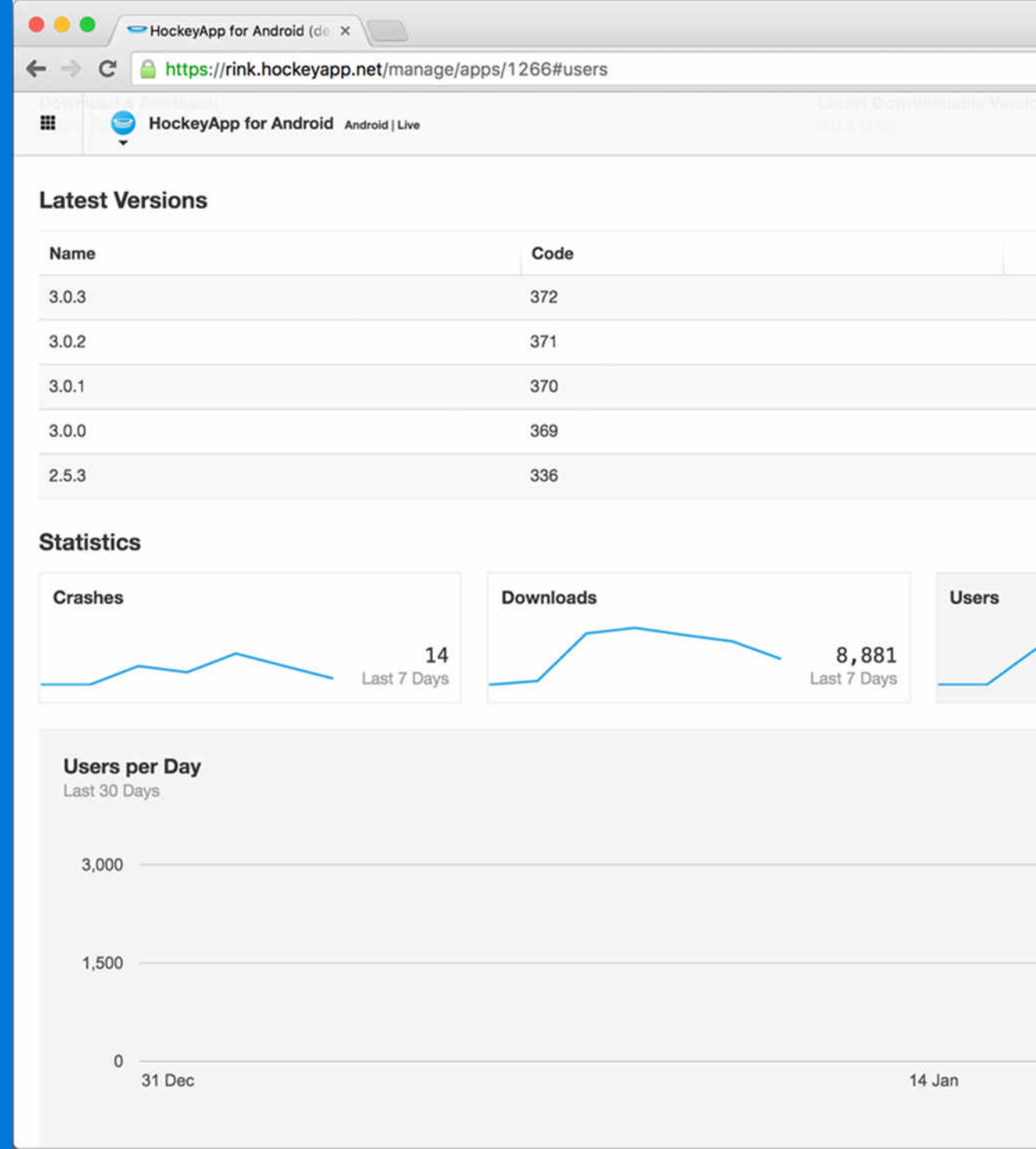
Beta test coverage

Real usage matters

Advanced metrics to see which devices were used, how long the app was used for, and which language was tested. No additional setup.

Detailed charts

HockeyApp features both raw data from analytics as well as live, interactive charts for the most important metrics.



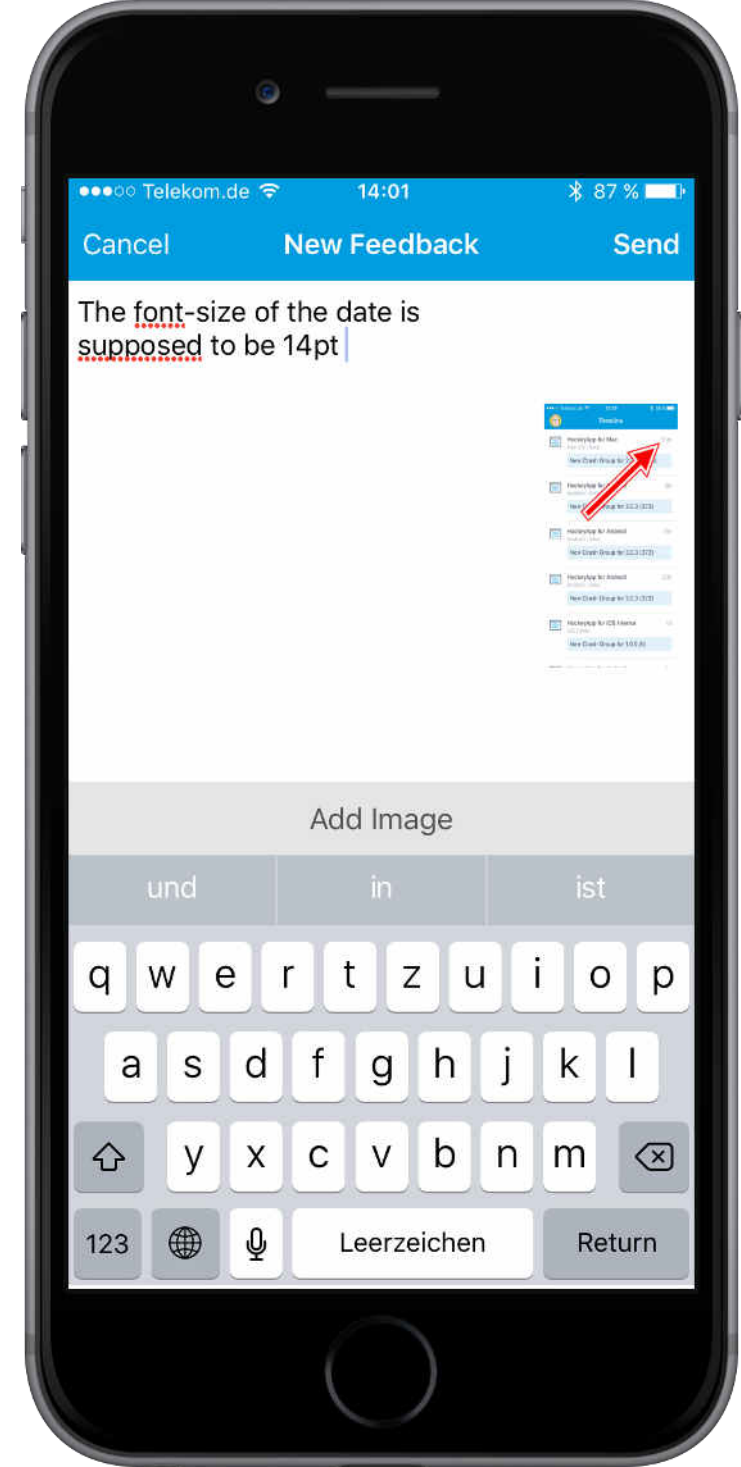
User feedback management

A conversation with your users

All feedback is handled as discussions. You can manage discussions in the web interface or through email. Search discussions, mark them as completed or create work items based on them. Give your users a voice outside of the app store.

For all development phases

You can use the feedback feature for all builds of your app or just beta versions. Either way, HockeyApp makes it easy for users to tell you what's what. Let them post ideas right from within your app.



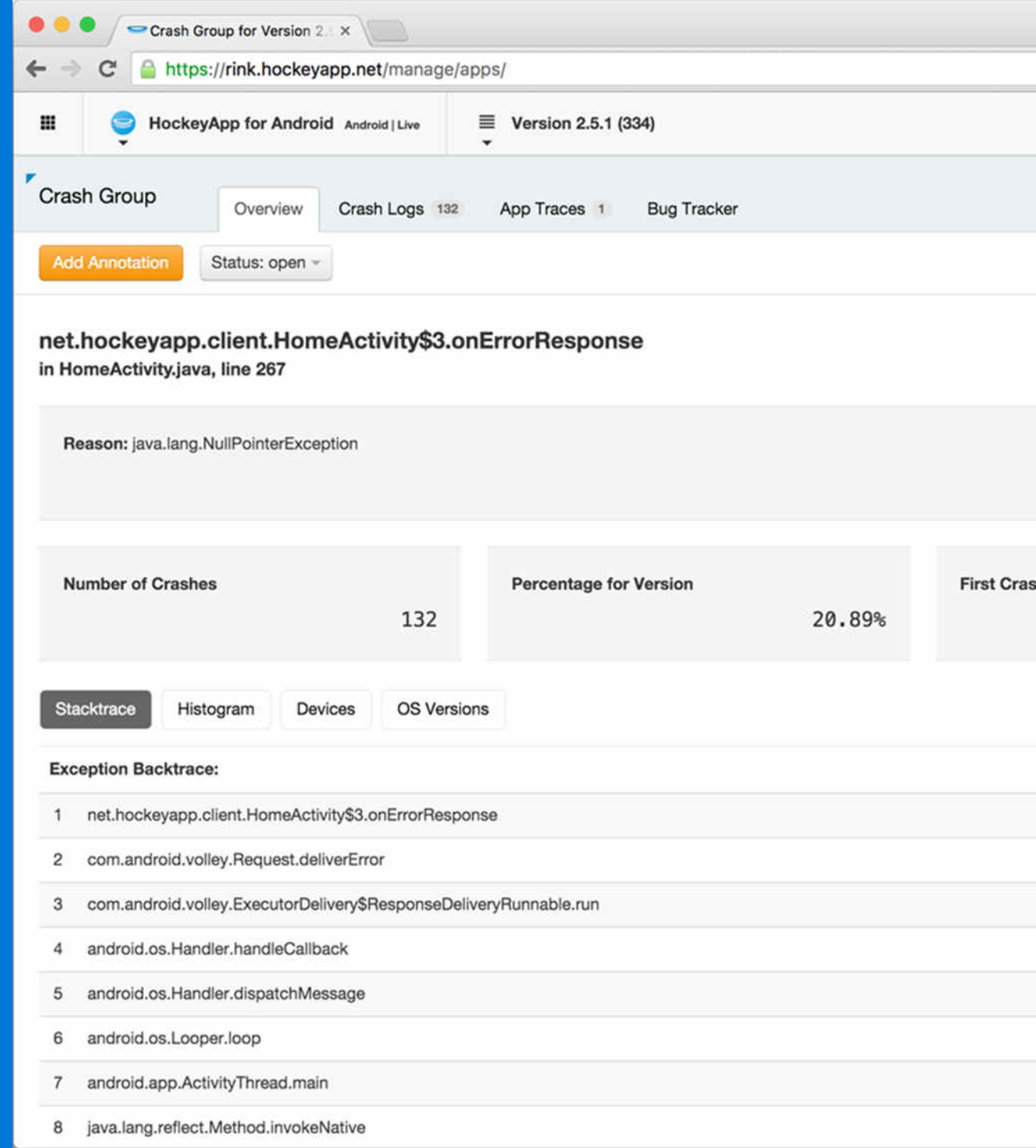
HockeyApp Crash Analytics

No additional code

By integrating HockeyApp's open source SDK for Android, iOS, Mac, and Windows your apps can send crash reports directly to HockeyApp, no additional code needed.

Powerful crash analytics

HockeyApp processes and symbolicates all crash reports. This gives you meaningful stack traces with friendly class names, methods, and accurate line numbers.



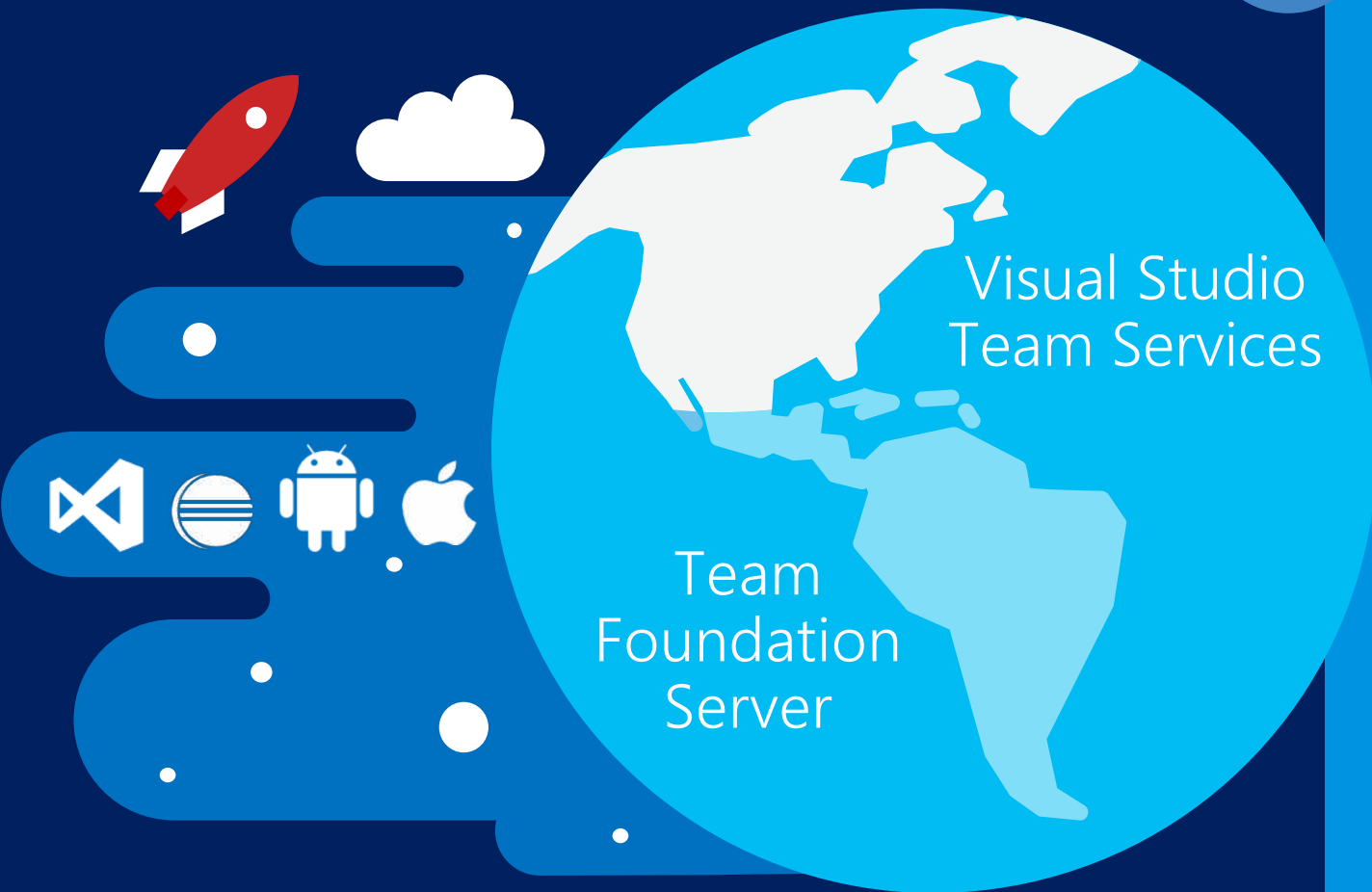
The screenshot shows the HockeyApp web interface for a specific crash group. The browser address bar displays `https://rink.hockeyapp.net/manage/apps/`. The page title is "Crash Group for Version 2.5.1 (334)". The main content area shows the crash details for a `NullPointerException` in `net.hockeyapp.client.HomeActivity$3.onErrorResponse` at line 267 of `HomeActivity.java`. The reason for the crash is `java.lang.NullPointerException`. The interface also displays summary statistics: 132 total crashes and a 20.89% percentage for this version. Below these statistics are tabs for "Stacktrace", "Histogram", "Devices", and "OS Versions". The "Stacktrace" tab is selected, showing an 8-line exception backtrace starting from `net.hockeyapp.client.HomeActivity$3.onErrorResponse` and ending with `java.lang.reflect.Method.invokeNative`.

Number of Crashes	Percentage for Version	First Cras
132	20.89%	

Exception Backtrace:

- 1 net.hockeyapp.client.HomeActivity\$3.onErrorResponse
- 2 com.android.volley.Request.deliverError
- 3 com.android.volley.ExecutorDelivery\$ResponseDeliveryRunnable.run
- 4 android.os.Handler.handleCallback
- 5 android.os.Handler.dispatchMessage
- 6 android.os.Looper.loop
- 7 android.app.ActivityThread.main
- 8 java.lang.reflect.Method.invokeNative

Summary



Open, flexible and extensible cross-platform DevOps tools

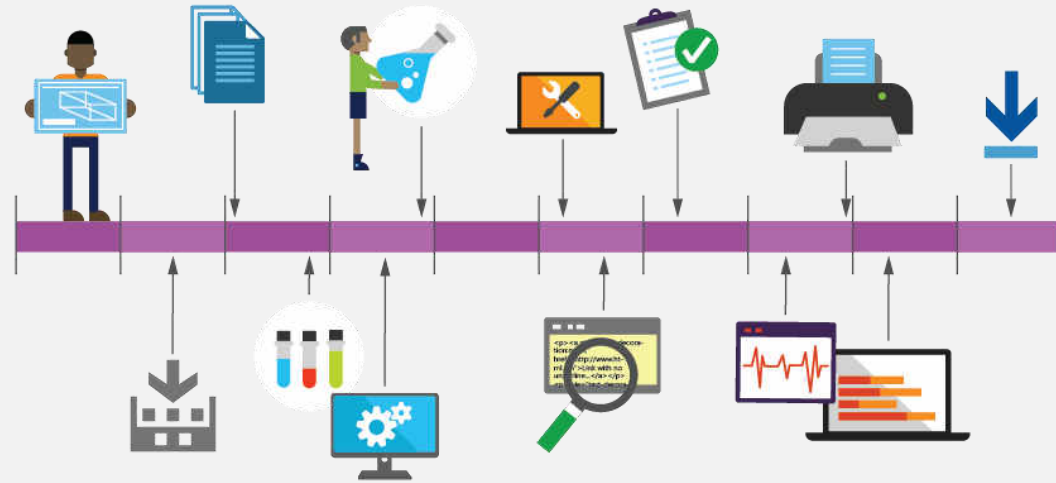
Enterprise Ready

- *SLA*
 - *Security*
-

Enables Mobile DevOps

- *Xamarin Test Cloud*
- *HockeyApp*

Thanks!



DevOps at Scale: A True Story

Davide Benvegnù

Microsoft MVP Visual Studio ALM

