

Mobile DevOps

Taavi Kõosaar, VS ALM MVP & DevOps
Consultant, taavik@solidify.se, Solidify AB
Simon Jäger, Technical Evangelist, Microsoft



Agenda:

Day 1

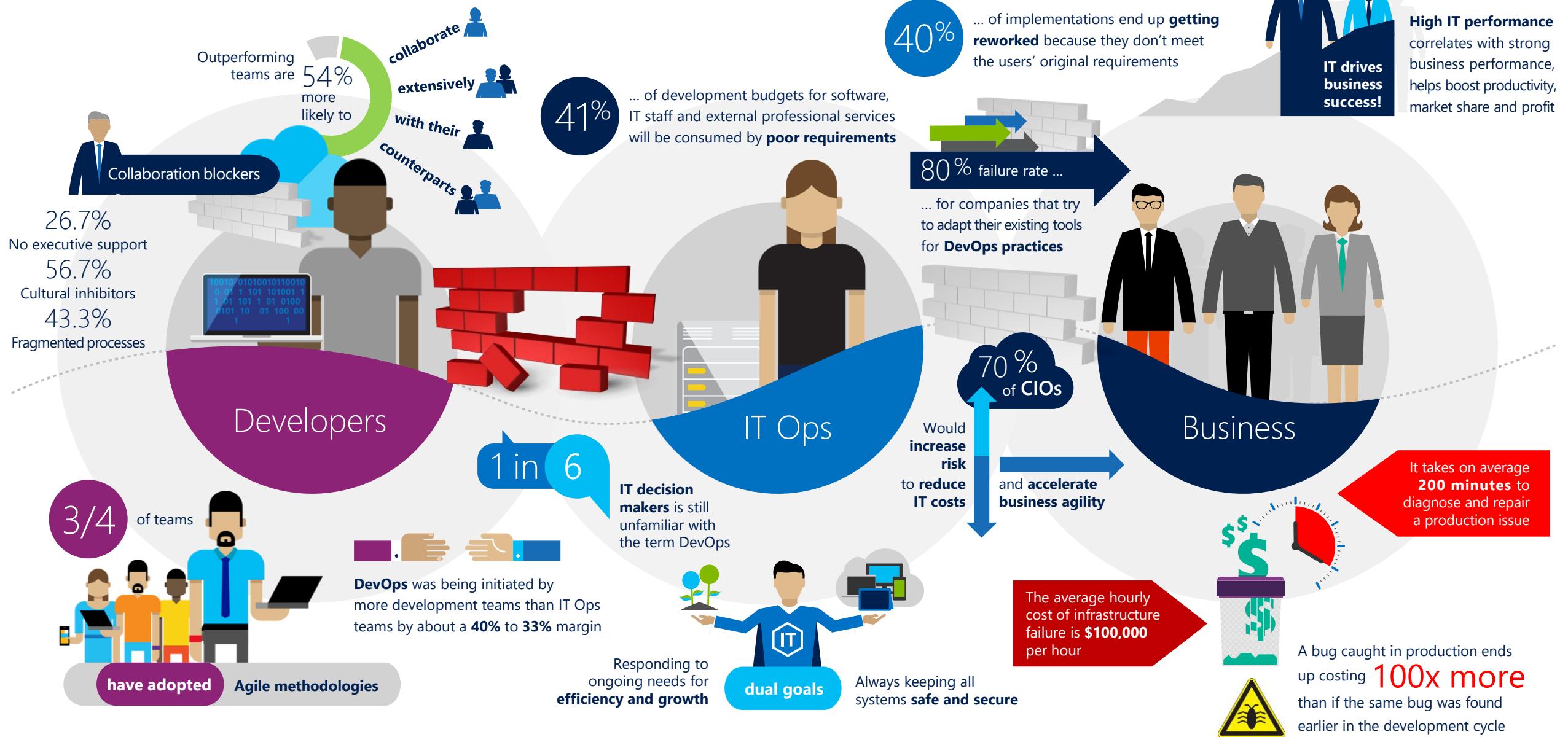
0	Welcome, introduction, organizational matters, brief overview of the workshop	15
0	DevOps Introduction	45
1	Microsoft Developer Platform and Mobile Development	75
2	Creating a Xamarin Sample Application	75
	<i>HOL time for participants / Q&A / Discussion</i>	30
3	Intro VSTS: Setting up Build, Release Management	120
	<i>HOL time for participants / Q&A / Discussion</i>	45
4	Importance of testing your projects	60
	<i>HOL time for participants / Q&A / Discussion</i>	20
5	Automated Testing using Xamarin Test Cloud	60
	<i>HOL time for participants / Q&A / Discussion</i>	30
6	Gathering User Feedback using HockeyApp	75
	<i>HOL time for participants / Q&A / Discussion</i>	30
7	Enriching your Mobile Application with Services	90
	<i>HOL time for participants / Q&A / Discussion</i>	30

Day 2

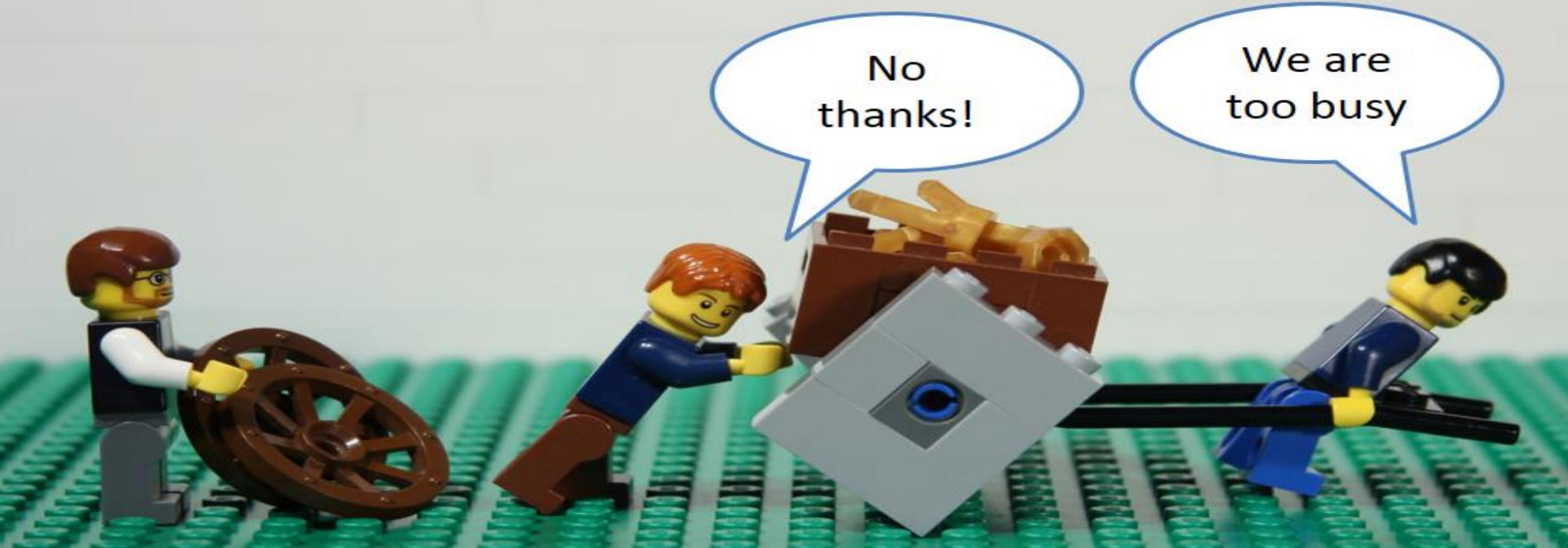
DevOps Introduction



Value delivery challenges



Are you too busy to improve?



Håkan Forss @hakanforss <http://hakanforss.wordpress.com>

This illustration is inspired by and in part derived from the work by Scott Simmerman, "The Square Wheels Guy" <http://www.performancemanagementcompany.com/>

Good & Green Guides

The world's largest taxi company, owns no vehicles

Facebook

The most valuable retailer, has no inventory

Airbnb

The leading sustainable travel guide specialist, has no guides

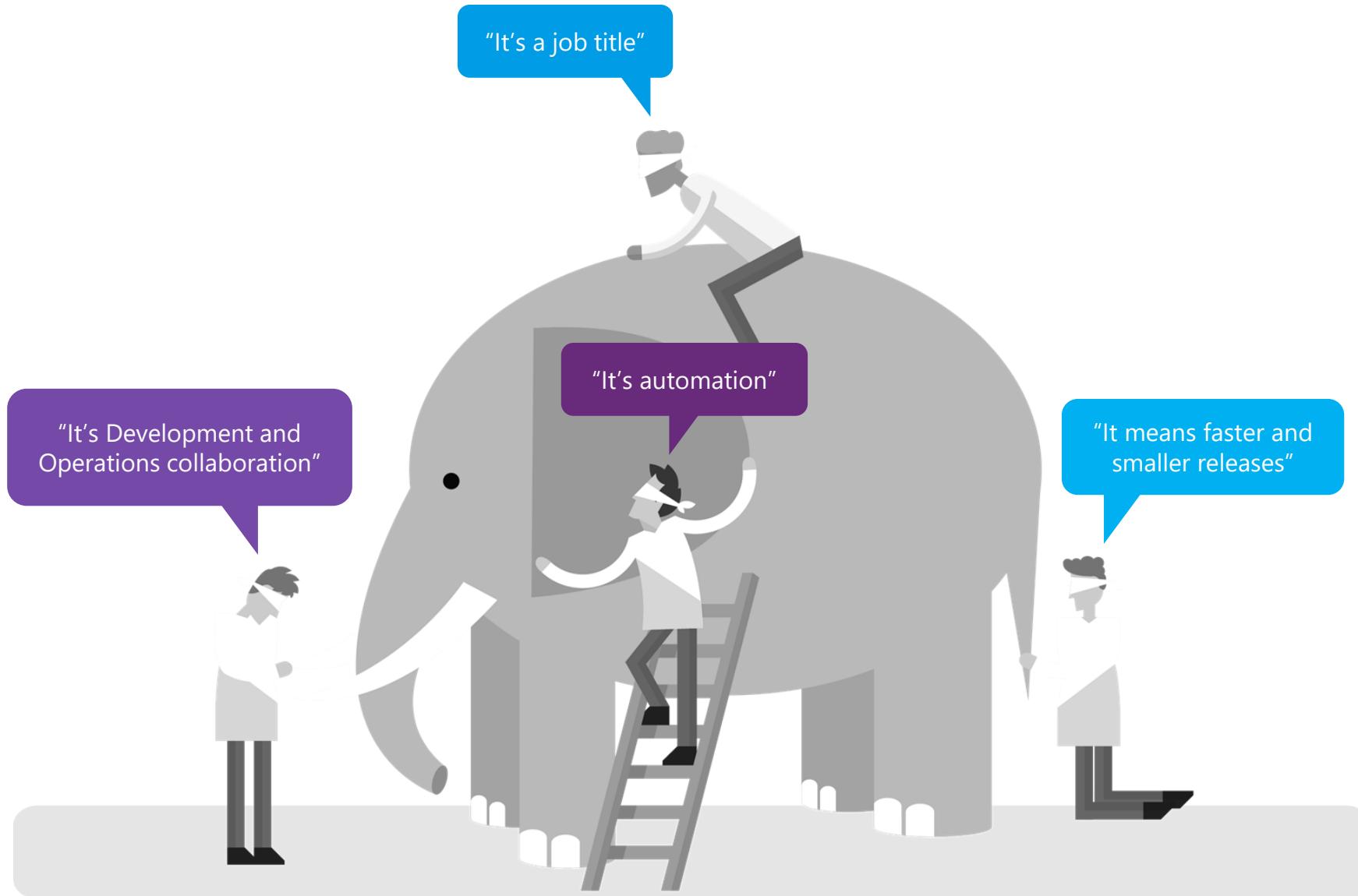
Uber

The world's most popular media owner, creates no content.

Alibaba

The world's largest accomodation provider, owns no real estate.

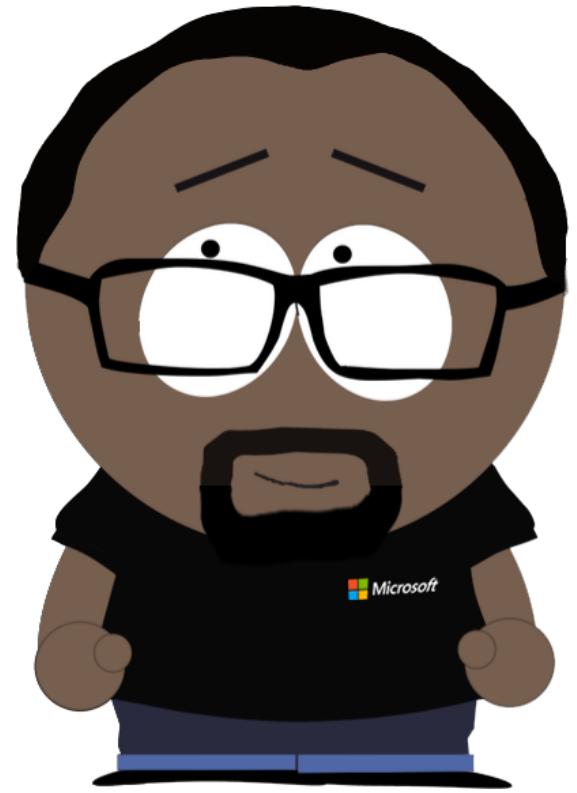
What is DevOps?



“DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.”

- Donovan Brown

<http://bit.ly/WhatIs-DevOps>



The converged DevOps lifecycle



The DevOps conversation



PEOPLE

Collaborate more

Share common goals

Focus on improvement

BRINGING PEOPLE TOGETHER



PROCESS

Eliminate waste

Increase efficiency

Streamline feedback

DELIVERING VALUE FASTER



TOOLS

Enhance productivity

Enable collaboration

Facilitate experimentation

EXECUTING A DEVOPS STRATEGY

DevOps habits and practices

PRACTICES

- Automated Testing
- Continuous Integration
- Continuous Deployment
- Release Management



PRACTICES

- Enterprise Agile
- Continuous Integration
- Continuous Deployment
- Release Management

PRACTICES

- Usage Monitoring
- Telemetry Collection
- Testing in Production
- Stakeholder Feedback



EVIDENCE gathered in PRODUCTION

PRACTICES

- Testing in Production
- Usage Monitoring
- User Telemetry
- Stakeholder feedback
- Feature flags

PRACTICES

- Code Reviews
- Automated Testing
- Continuous Measurement



PRACTICES

- Application Performance Management
- Infrastructure as Code
- Continuous Delivery
- Release Management
- Configuration Management
- Automated Recovery

PRACTICES

- Application Performance Management
- Infrastructure as Code
- Continuous Deployment
- Release Management
- Configuration Management
- Automated Recovery



The shift to DevOps

OLD WORLD

Focus on planning
Compete, not collaborate
Static hierarchies
Individual productivity
Efficiency of process
Assumptions, not data

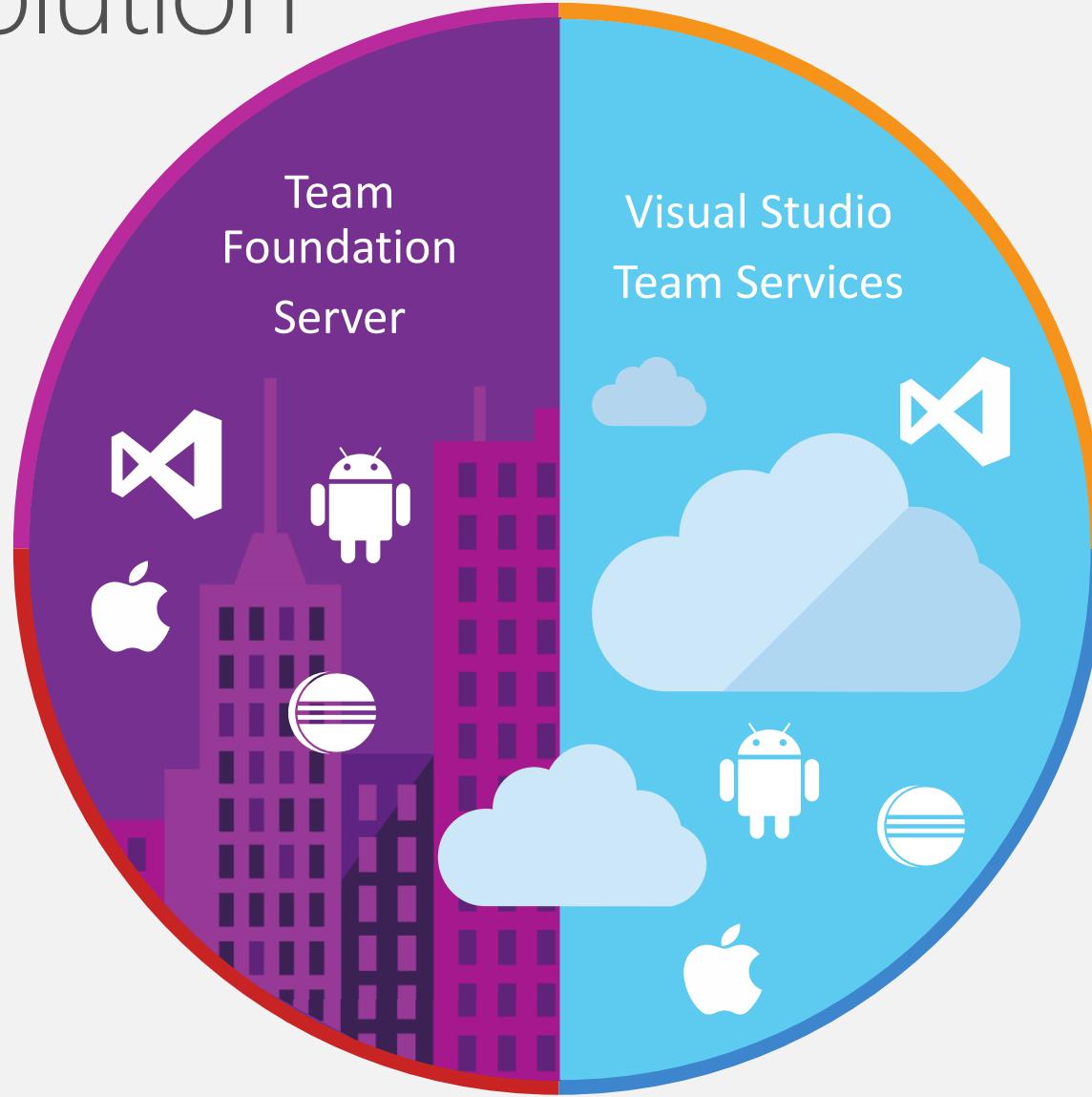
NEW WORLD

Focus on delivering
Collaborate to win
Fluent and flexible teams
Collective value creation
Effectiveness of outcomes
Experiment, learn and respond

Manage the full stack

Monitor/Learn	Telemetry	App Insights HockeyApp
	Diagnostics	App Insights HockeyApp
	Performance	App Insights HockeyApp
Approve	Collect Approvals	Release Management
	Notify Approvers	Release Management
Test	Manual Tests	Microsoft Test Manager
	Automated Tests	UI Test, Calabash
	Create Test Data	BacPac with SSDT
App	Configure Application	Tokenization
	Install Application	NuGet, WebDeploy, etc
Infra	Configure Environment	Desired State Configuration
	Provision Environment	Azure Resource Manager
Plan	Tracking Work	TFS / VSTS
	Planning Work	TFS / VSTS

Microsoft's solution

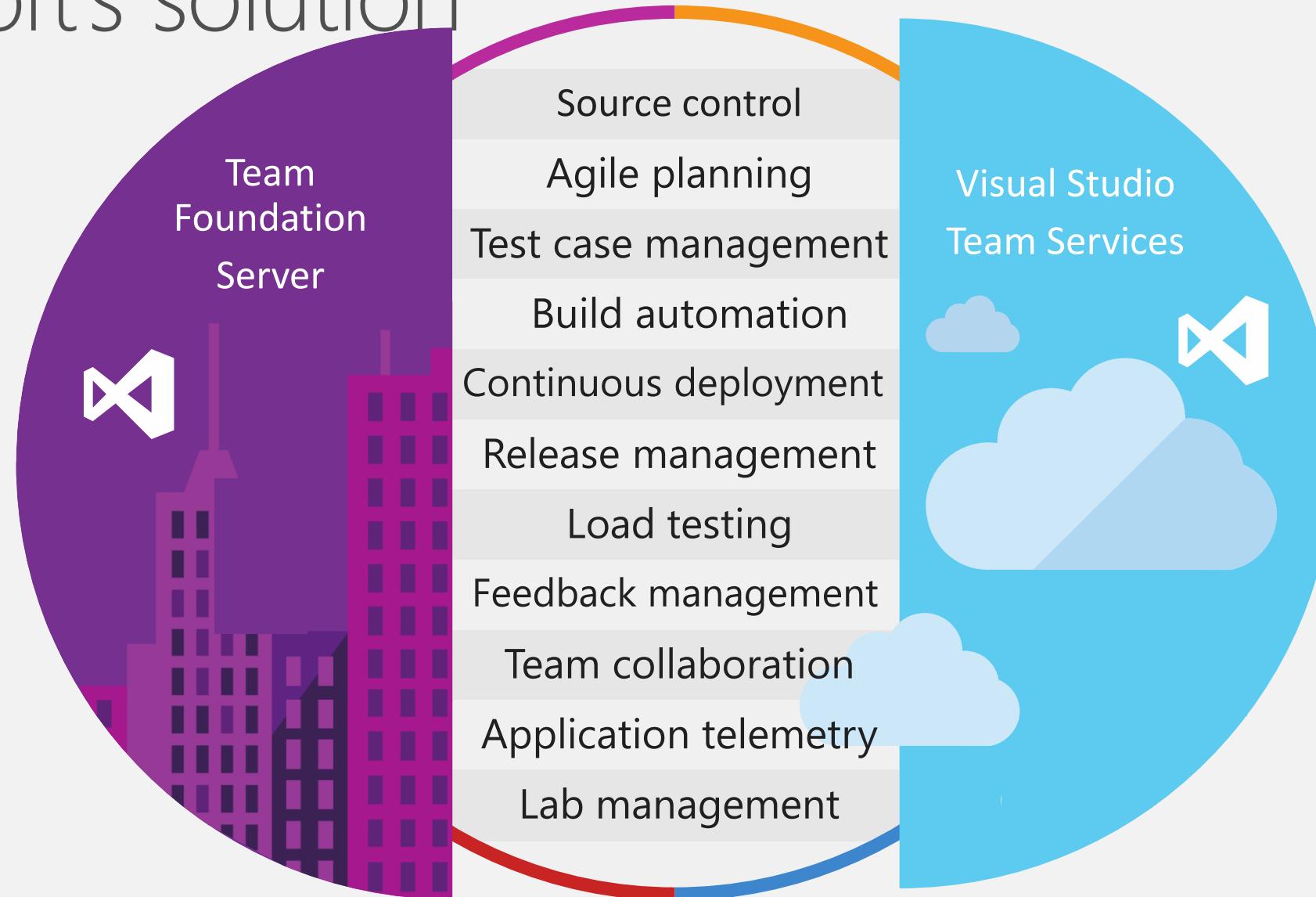


Office Project Server

Microsoft System Center

Microsoft Azure SharePoint

Microsoft's solution



Office Project Server

Microsoft System Center

Microsoft Azure SharePoint

Our solution for mobile DevOps

Tools for continuous integration, beta deployment, crash analytics and monitoring of mobile applications that are distributed through app stores.

Automate and orchestrate
build, test and release

Friction-free mobile app
deployment and beta testing

Close the loop between
Development and Operations

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 shows the Xamarin Test Cloud interface. At the top, it displays the URL "testcloud.xamarin.com" and the project "Xamarin CRM" with the branch "master" and the date "Aug 23, 2015 10:03:59 PM". The main area is divided into two sections: "OVERVIEW" on the left and "TEST RESULTS" on the right.

OVERVIEW: Shows a summary of test results for the "Customer" feature. It includes a table with columns for "Test Name", "Status", and "Last Run". The table shows several green checkmarks indicating successful tests.

TEST RESULTS: This section lists various tests grouped under categories like "Customer tests" and "Order tests". Each test has a status indicator (green checkmark or red lightning bolt) and a link to view the results. A specific test, "Then I tap 'Sales'", is highlighted with a blue background and a red lightning bolt icon, indicating a failure.

Device Screenshots: On the right side, there are 12 small screenshots of an iPhone displaying a bar chart titled "8 Week Sales" with values ranging from \$2,126.00 to \$2,126.00. The devices shown are Apple iPhone 5C, Apple iPhone 5, and Apple iPhone 6, running iOS versions 8.2, 8.3, 8.1.3, and 7.1.1.

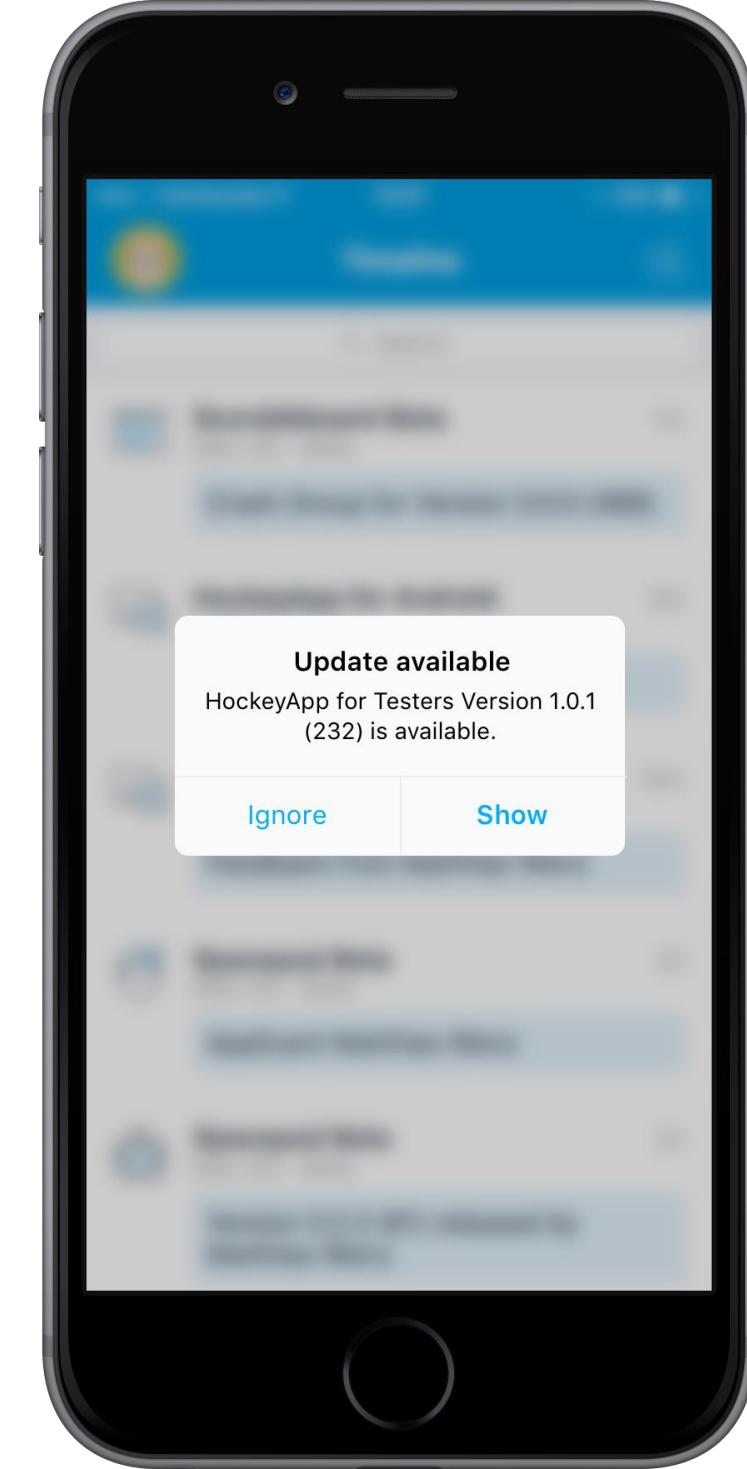
HockeyApp Beta Distribution

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.



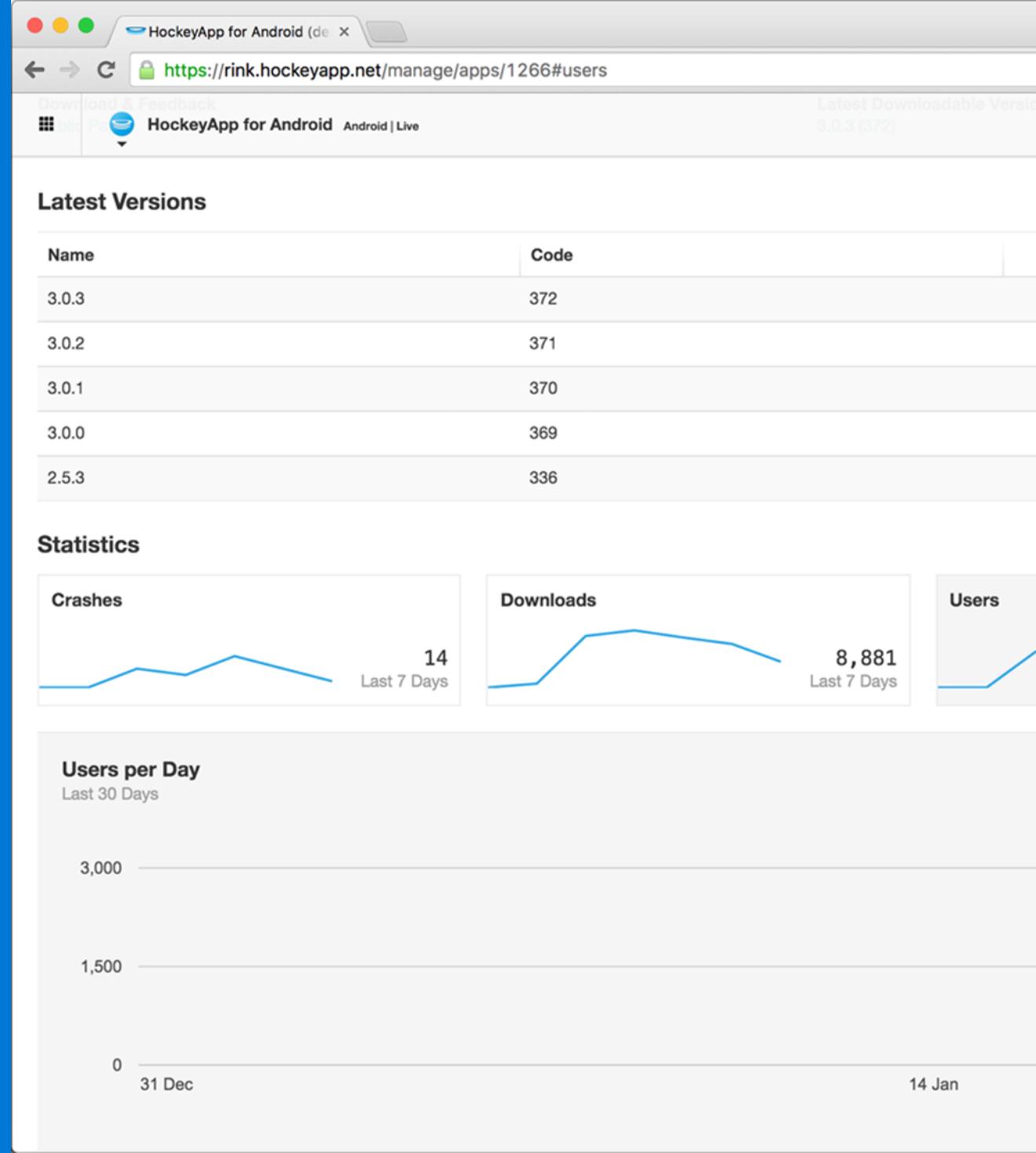
Beta test coverage

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.



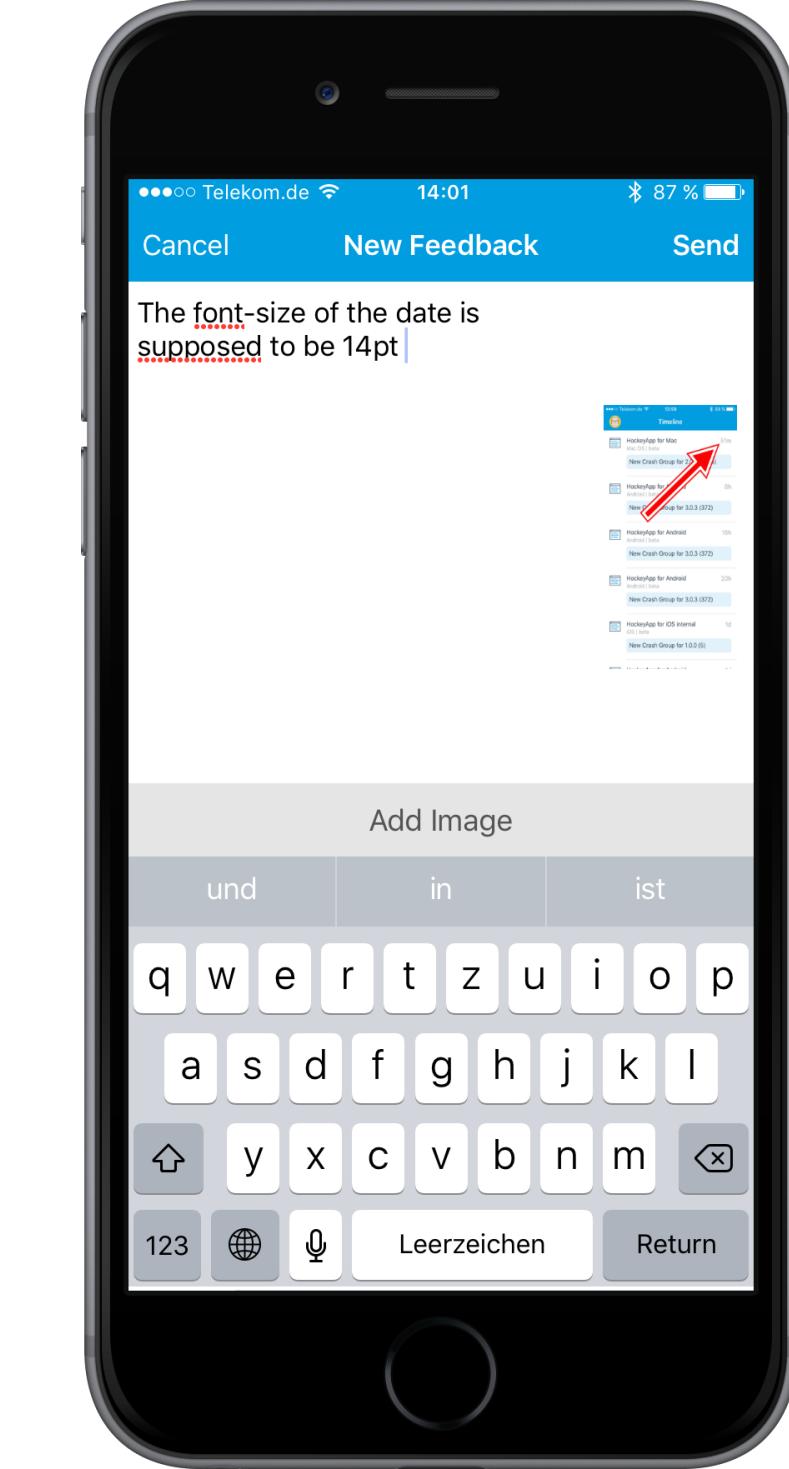
User feedback management

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 at <https://rink.hockeyapp.net/manage/apps/>. The main navigation bar includes links for Overview, Crash Logs (132), App Traces (1), and Bug Tracker. Below the navigation is a search bar and a status dropdown set to 'open'. The main content area displays a crash group for 'HockeyApp for Android' (Android | Live) and Version 2.5.1 (334). A specific crash entry is highlighted, showing the reason as 'java.lang.NullPointerException' occurring in 'HomeActivity.java' at line 267. The 'Stacktrace' tab is selected, displaying the following 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
```

Below the stacktrace, there are summary statistics: 'Number of Crashes' (132), 'Percentage for Version' (20.89%), and 'First Crash' (not visible in the screenshot). Other tabs available include Histogram, Devices, and OS Versions.

DevOps in practice

- Continuous Delivery & Automation
 - Release Management & Package Management
 - Semantic Versioning
- Short lived topic branches and Trunk based development
 - Pull Requests
- Separating deployment from feature releases
 - Feature Toggling
- Measure and Learn
 - Hypothesis Based Development & Experimentation

Demo



Microsoft Tooling



OSS Tooling

Develop

Developer IDE



Team Collaboration

GitHub
CodePlex



People | Process | Tools

Build + Test

Build/CI

gradle
GRUNT
Jenkins
Hudson

Test

gradle
GRUNT



On-Premises | Hybrid | Cloud

Deploy

Configuration

puppet labs **CHEF**

Release

gradle
GRUNT
Jenkins
Hudson

VAGRANT

Monitor + Learn

Monitor

Nagios
ZABBIX





Visual Studio
Team Services



Team Foundation
Server



Open, flexible and extensible
cross-platform DevOps tools

Unlocks a greater ecosystem and
works with what you already use

More agility and flexibility for
continuous value delivery

Hosted by Microsoft
Visual Studio Team Services



Cloud services for teams

Hosted in your data center
Team Foundation Server



Team collaboration server

Getting Visual Studio Team Services

Completely free for small teams

FREE: 5 users + unlimited stakeholders

FREE: 240 minutes cloud build time/month

FREE: 20k virtual user minutes for load testing/month

Pricing for additional team members

No. of users:	5	10	20	50	100	200	1000
Cost per month:	FREE	\$30	\$110	\$350	\$750	\$1,150	\$4,350

Each Team Services paid user also gets a Team Foundation Server CAL (Client Access License)

Free for MSDN and Visual Studio subscribers

Users with the following subscriptions can be added to an unlimited number of VSTS accounts for free, with "Basic" feature access:

- Visual Studio Enterprise - annual
- Visual Studio Enterprise - monthly
- Visual Studio Enterprise with MSDN - including subscriptions offered through BizSpark and the Microsoft Partner Network
- Visual Studio Professional - annual
- Visual Studio Professional - monthly
- Visual Studio Professional with MSDN
- Visual Studio Test Professional with MSDN
- MSDN Platforms

Additional services per account

Cloud Build

FREE: 240 minutes/month on a hosted build agent with 30 mins max runtime for a single build

FREE: 1 private agent (new build system) or XAML controller (legacy build system)

THEREAFTER:

Hosted build agent for \$40/month* (you can upgrade the free agent or buy additional agents)

Private agent for \$15/month

Cloud-based Load Testing*

FREE: 20,000 virtual user minutes per month

THEREAFTER:

\$0.004/virtual user minute for 20,001-2M VUM

\$0.002/VUM for 2,000,001-10M VUM

\$0.001/VUM for usage above 10M VUM/month

* Requires Visual Studio Enterprise

- Services purchased via Microsoft Azure based on actual usage
- Additional services are accessible to all users on the account and shared among them
- Build time is actual computing time required to run the build, and does not include queuing time (if any)
- Load test runs specify the number of virtual users and duration in minutes
Virtual users x minutes = virtual user minutes (VUM)

Getting Team Foundation Server

Completely free for individuals and small teams

Team Foundation Server Express is free for individual developers and small teams of five or less and can easily be installed on a personal desktop or laptop without a dedicated server.

Pricing for additional team members

With **Team Foundation Server**, you license the server and the users connecting to the server. Users who only need access to work items are free, and other users need a user CAL (client access license) for access to most features in Team Foundation Server.

Free for MSDN and Visual Studio subscribers

Users with the following subscriptions get both a Team Foundation Server license and a Team Foundation Server User CAL:

- Visual Studio Enterprise - annual
- Visual Studio Enterprise - monthly
- Visual Studio Enterprise with MSDN - including subscriptions offered through BizSpark and the Microsoft Partner Network
- Visual Studio Professional - annual
- Visual Studio Professional - monthly
- Visual Studio Professional with MSDN
- Visual Studio Test Professional with MSDN
- MSDN Platforms

Feature comparison

	Visual Studio Team Services	Team Foundation Server
Plan & Track Work	■	■
Source Code Management	■	■
Package Management	■	■
Test Management	■	■
Cross-Platform Build	■	■
Continuous Deployment	■	■
Release Management	■	■
Application Telemetry	Application Insights/HockeyApp	System Center/PreEmptive Analytics
Reporting	Limited	■
Extend, Customize & Integrate	■	■
Team Structure	1 Collection – Unlimited Projects/Teams	Unlimited Collections/Projects/Teams
Identity & Permissions	Azure Active Directory Microsoft Accounts (Live ID)	Active Directory Integration
Languages	English Only	Multilanguage Support
Data Location	US, Europe, Australia DCs - India (coming soon)	Data Stored On Premises
Backup/Data Migration	Limited support, TFS Import (coming soon)	Backup/Restore Tools Available
Support	Phone, Online and Forums, Engineering Excellence	Phone, Online, Forums, Microsoft Premier Support
Updates	Frequent Updates (~3 weeks) Automatically Upgraded	Periodic Updates (~3 or 4 months) Planning Required

Feature comparison

	Visual Studio Team Services	Team Foundation Server
Version Control		
Team Foundation Version Control	■	■
Distributed Version Control with Git	■	■
Agile Planning & Collaboration		
Scrum, Agile, CMMI	■	■
Custom Process Templates & WIT Customization	Limited	■
Work Item Tagging, Kanban Board, Feedback	■	■
SharePoint, Project Server, System Center Integration		■
Build		
Automated Builds On-Premises with CI	Support for hybrid scenarios	■
Hosted Build Service with Continuous Integration	■	
Application/Platform Support	Many Microsoft platforms and growing (Build Server Software)	Fully customizable
Continuous Deployment to Azure	■	■
Testing		
Test Case Management	■	■
Cloud-Based Load Testing Service	■	
Reporting	Work Item Queries and Favorites Work Item Charting, Power BI	Work Item Queries and Favorites/Work Item Charting/Business Intelligence Warehouse

Microsoft Developer Platform



.NET

Any app, any developer, any platform



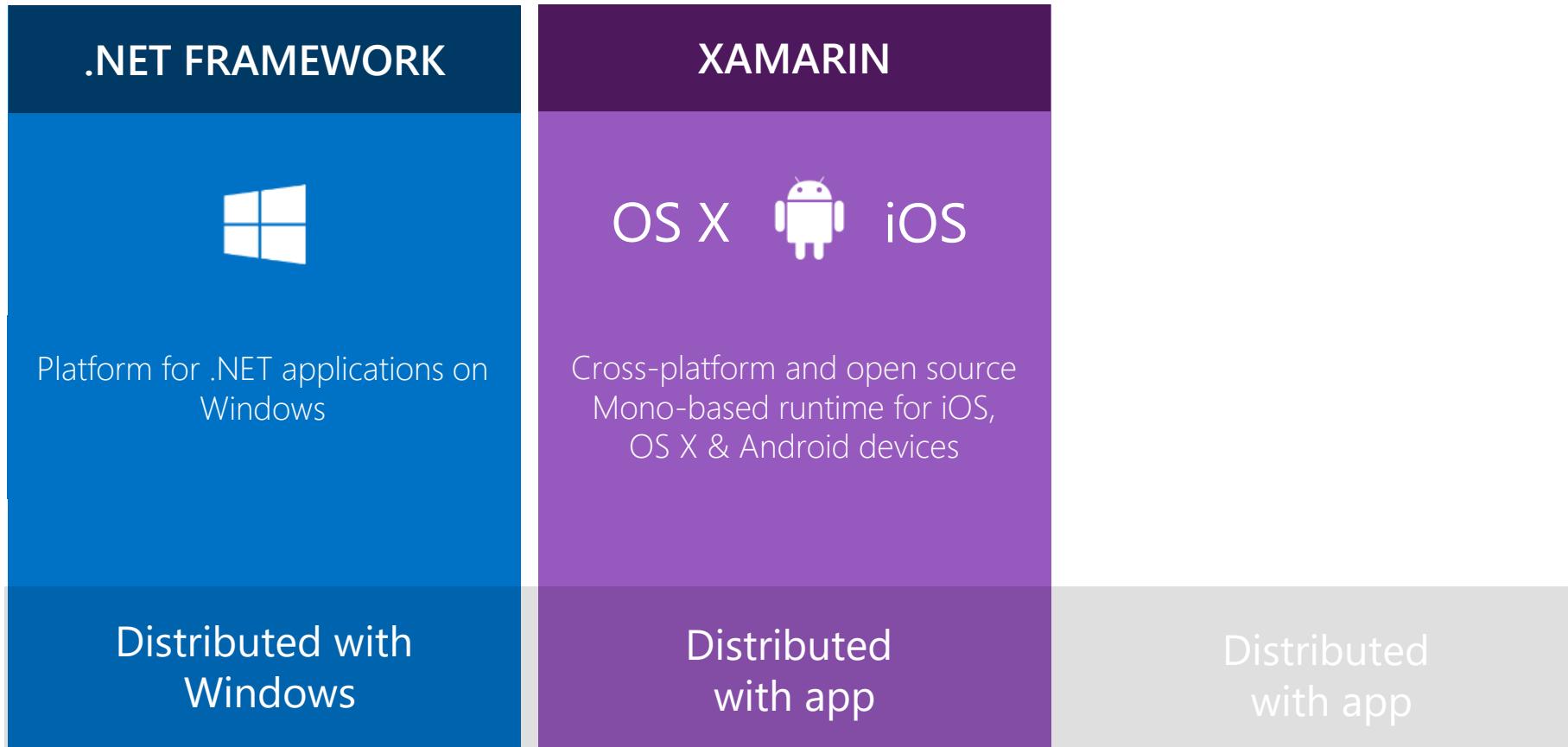
iOS



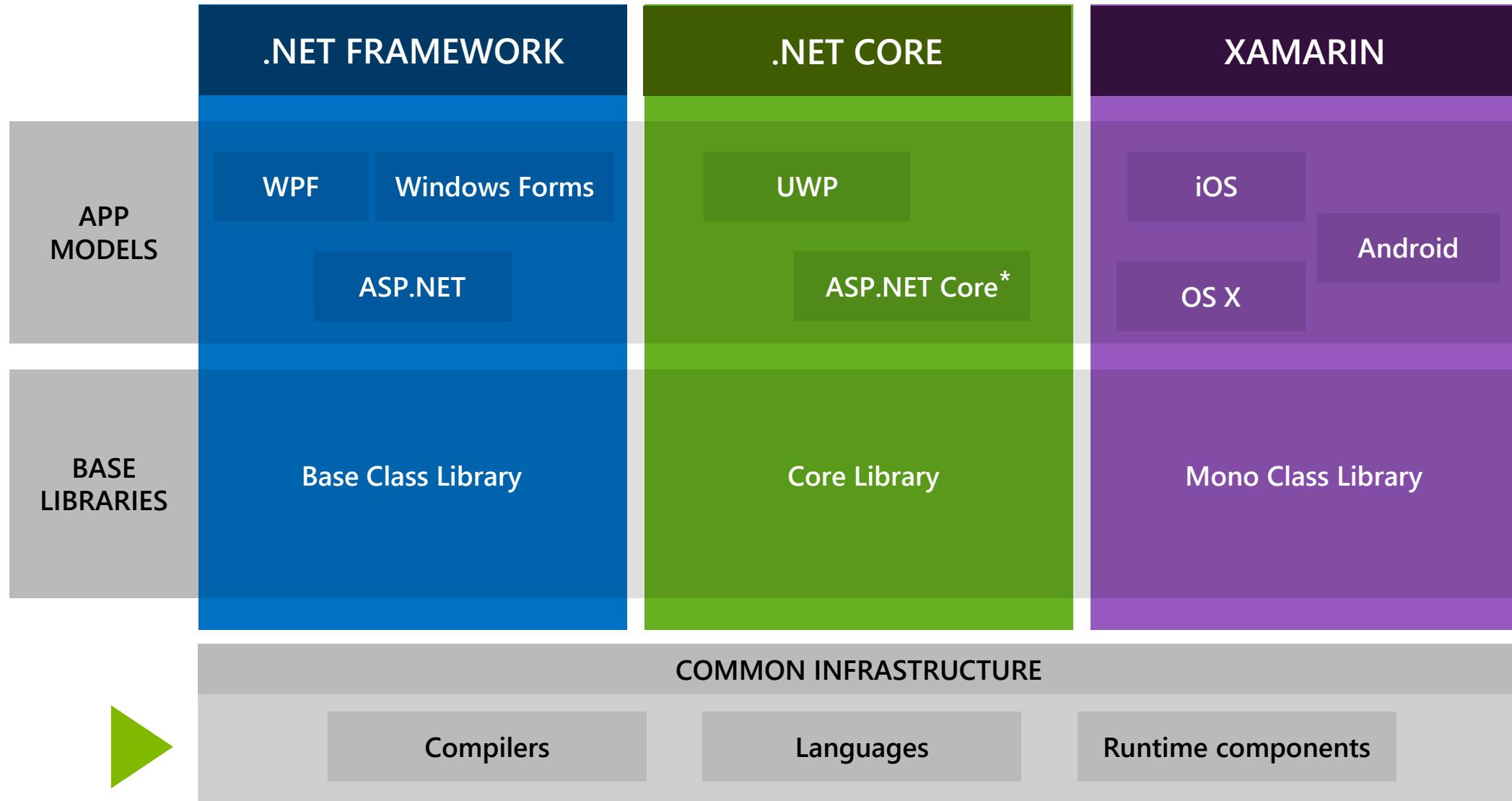
OS X



.NET Today - The family gets bigger



.NET Today



Mobile is transforming how we live and work ...

Mobile represents the single largest technology innovation of all time.

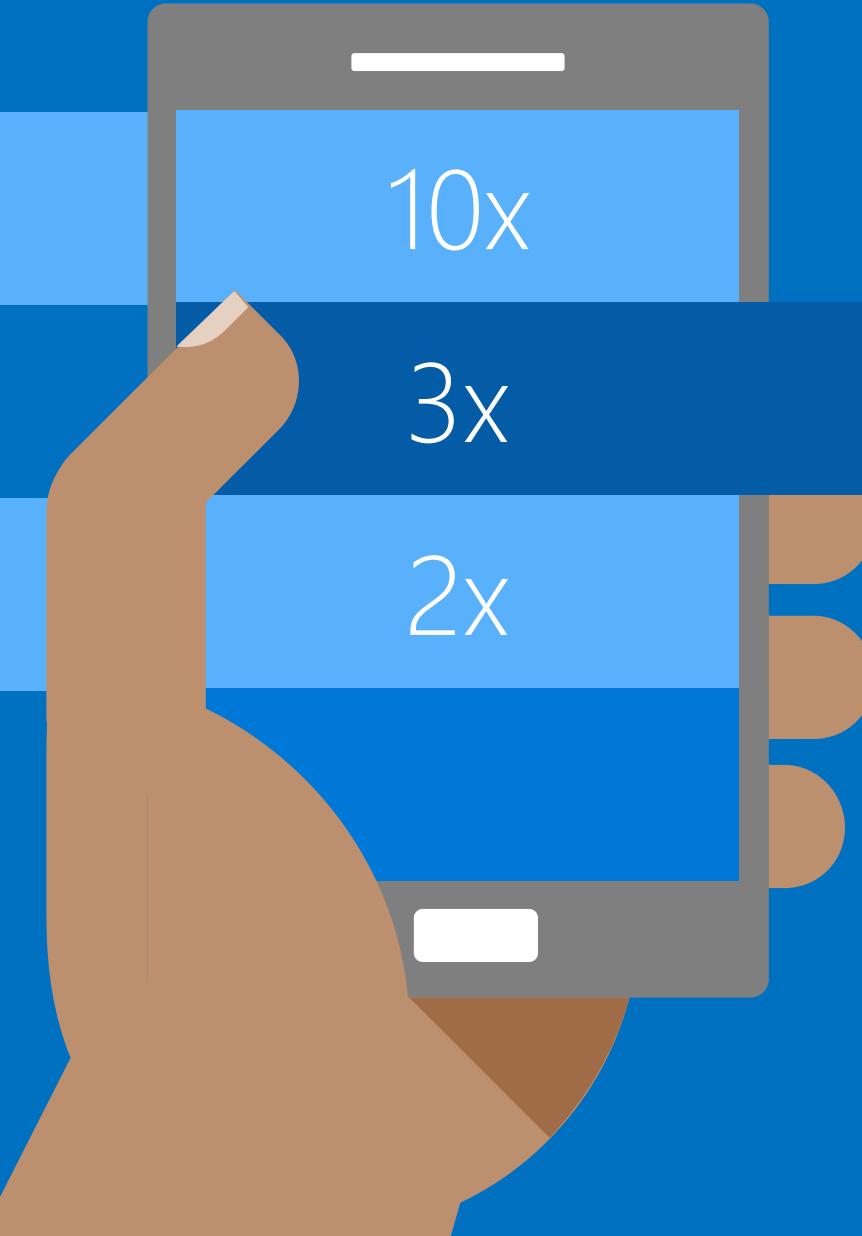


Mobile has happened ...

Faster than the
PC boom of the 80s

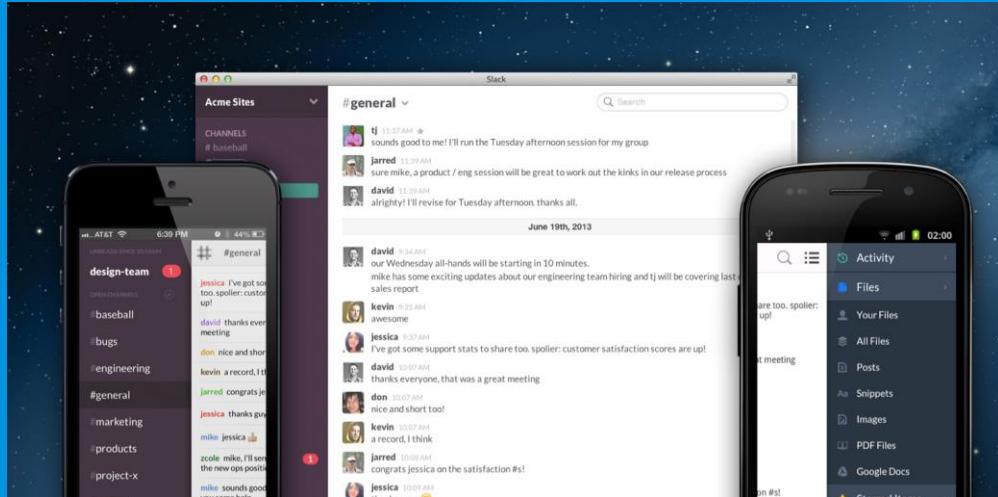
Faster than the
Internet boom of the 90s

Faster than the recent
social networking explosion

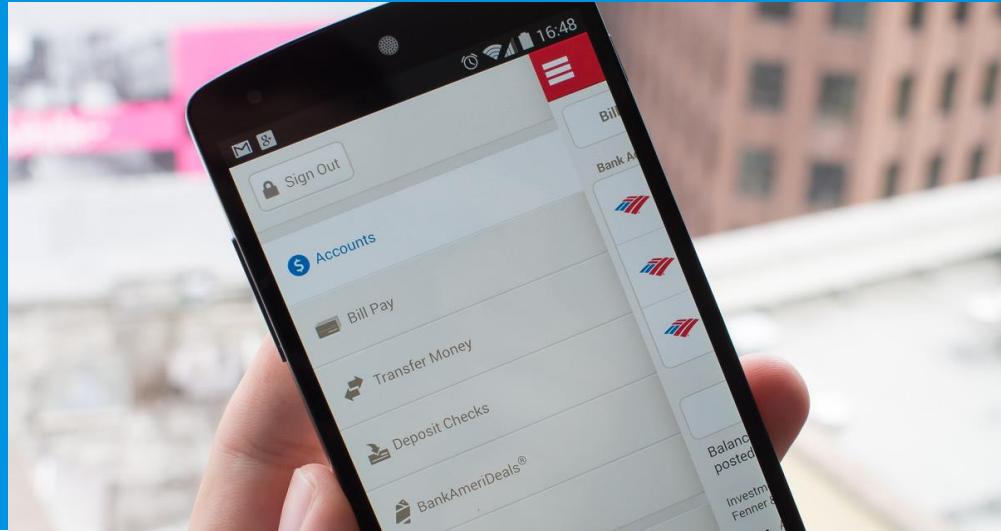
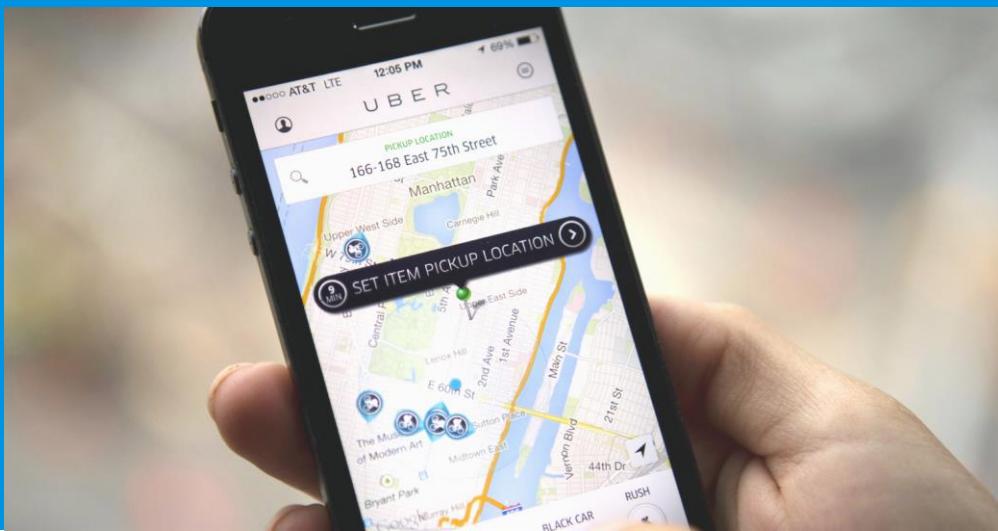
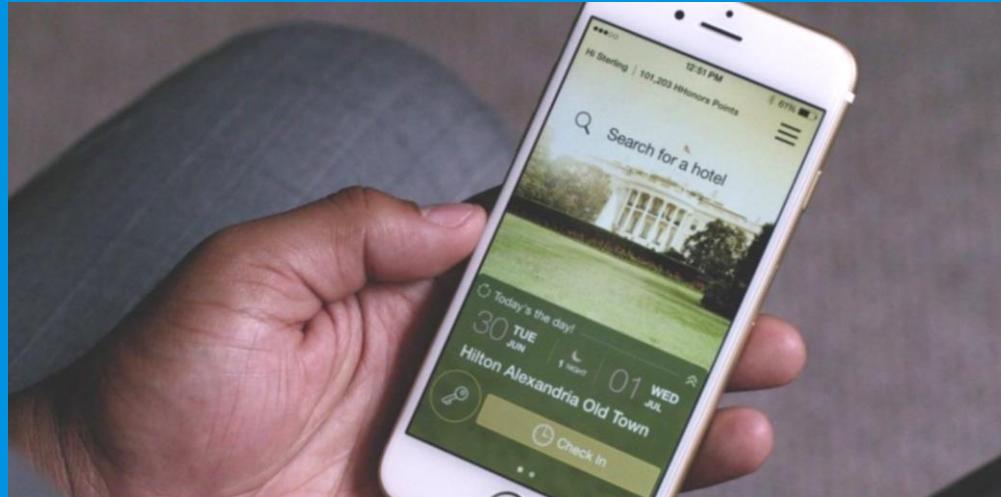


Every business is a mobile business

Entrepreneurs



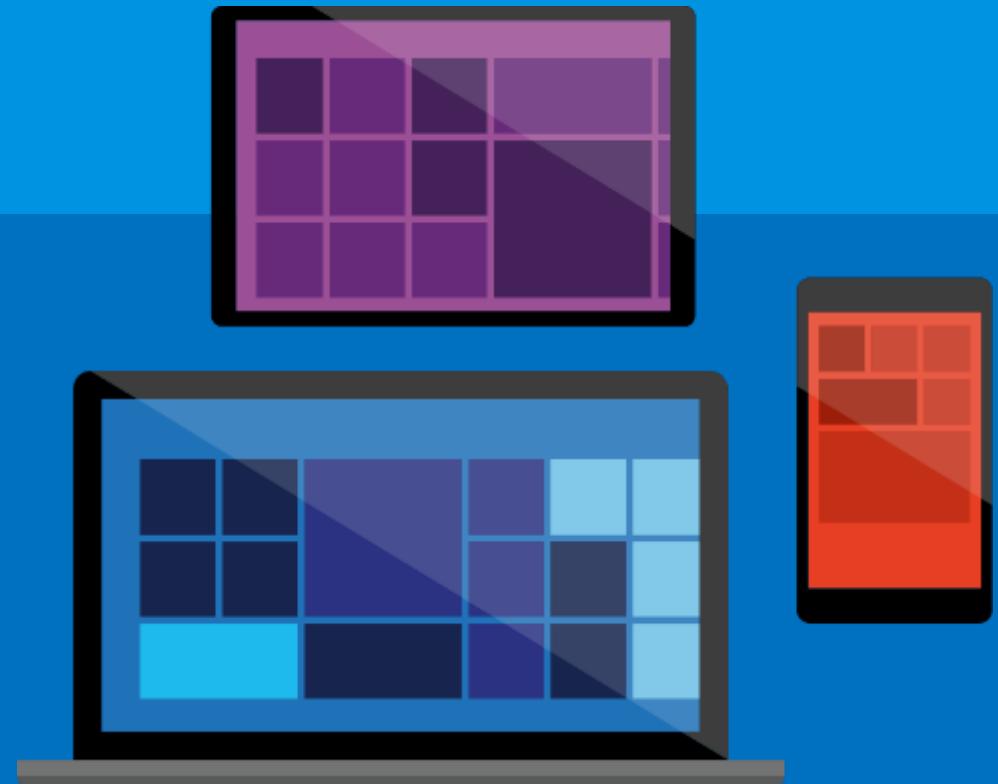
Enterprises



Mobile apps are a part of larger modern applications

Mobile app strategy

A mobile app strategy is not sufficient if you want to be successful in your future mobile development efforts. You need to think of mobile apps as one component of larger, modern, omnichannel application architecture ...



The *appification* of modern business

There's an app for that ...

Consumer apps are everywhere. Now, the same thing is happening across every aspect of the modern business.

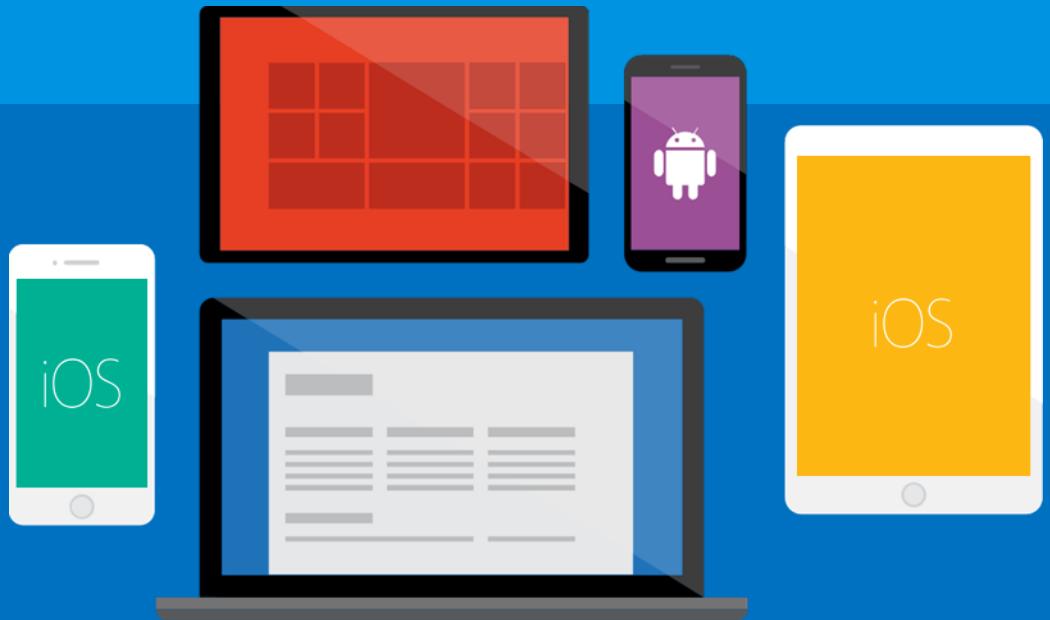
In order to be successful and stay valuable to their users – customers, employees and partners – businesses will have to support a multi-app mobile strategy and provide dedicated apps for different purposes and user groups.



Enterprises are going mobile on multiple platforms

In 2016

Employees use an average of **3 different devices** in their daily routine today, and this will increase to 5 or 6 devices as new technologies like IoT and wearables become more mainstream.



The mobile app dilemma

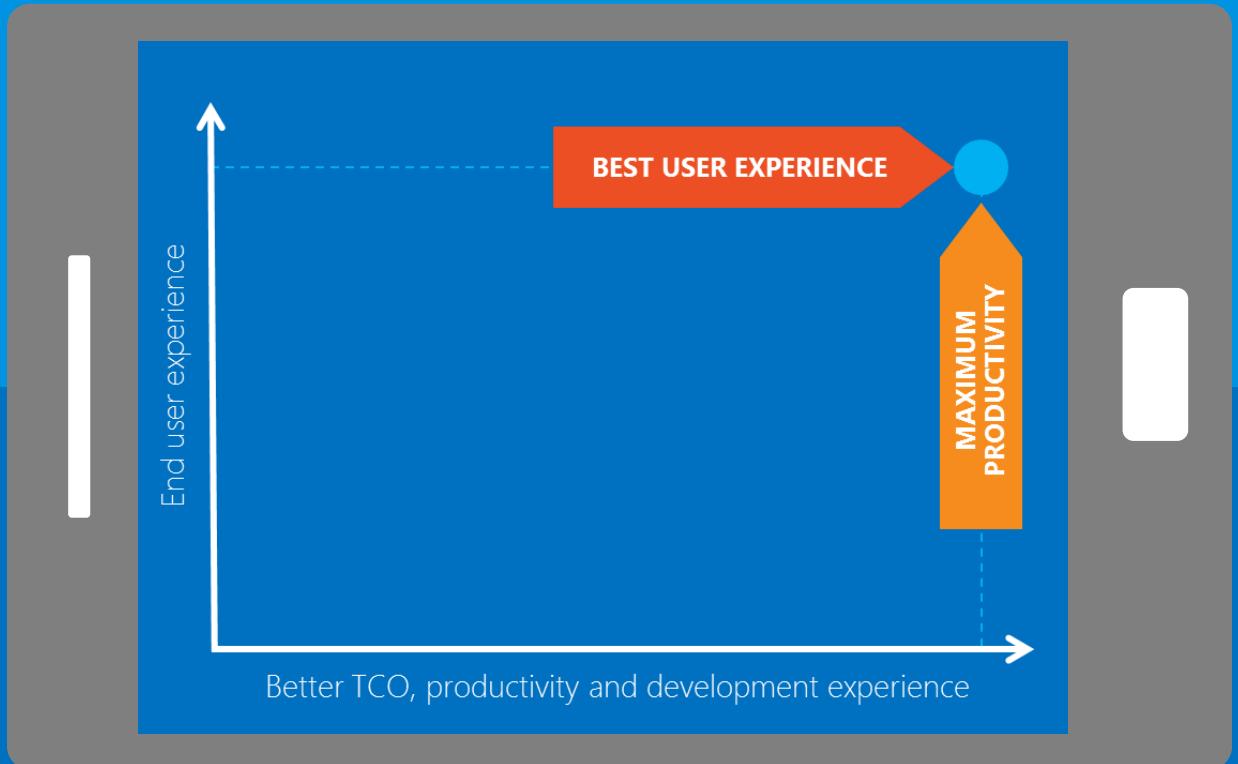
Trade-off

noun

*noun: **tradeoff***

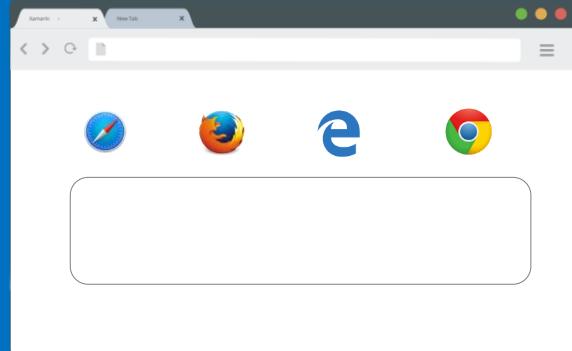
A balance achieved between two desirable but incompatible features; a compromise.

"a trade-off between objectivity and relevance"



Mobile approaches

Hybrid and Web



Platform Silos



Cross-Platform Native



User Experience



Non-native UX
Performance compromises



Native experience
Maximum performance



Fully native UI and performance
High fidelity API access

Developer Productivity



Build once, iterate easily



Multiple code bases
Higher TCO and maintenance



Only one code base, one team

Tools



Mature tooling, variety of tools



Little or no lifecycle management
tool support from platform vendors



Full-featured end-to-end lifecycle
management and DevOps tools

Platform silos



iOS Objective-C + Xcode



Java + Eclipse



.NET + Visual Studio

End User

- ✓ Platform-native UX
- ✓ Full performance
- ✓ Native service integration

Developer

- ✗ Only services (server-side) can be re-used
- ✗ Client development and developer experience is different for each platform and device type
- ✗ Higher TCO due to multiple code bases and requirements for developer skills

Platform silos

Native applications (x3)

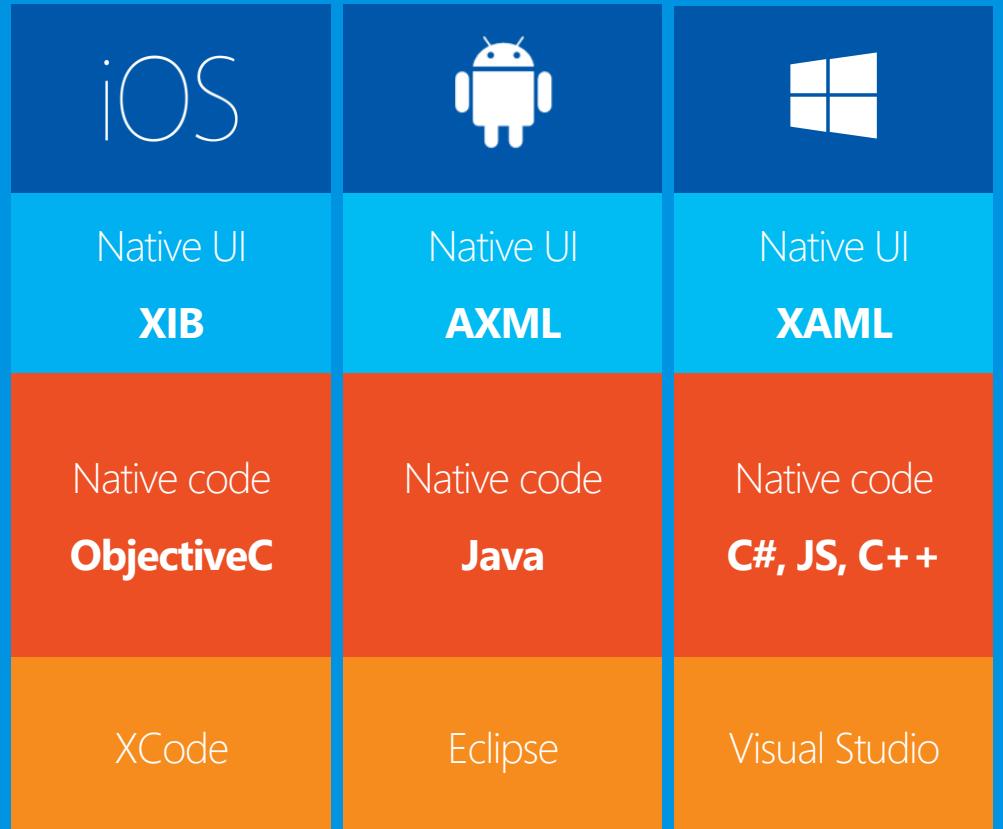
Individual native apps built for each platform

Native user experience but lowest code re-use

Only services (server-side) can be re-used, with custom code required for consuming

Client development and developer experience is different for each platform and device type

Higher TCO due to multiple code bases and developer knowledge



Write once, run anywhere



End User

- ✖ Non-native UX
- ✖ Performance tradeoff
- ✓ Works across devices

Developer

- ✓ Great for teams with HTML/JavaScript skills
- ✖ Scaling to enterprise complexity challenging
- ✓ Developer productivity with Visual Studio
- ✓ Mature tooling

Write once, run anywhere

Hybrid applications

Presentation layer written in HTML/CSS and logic written in JavaScript/TypeScript

Highest code reuse ratio but non-native UX

Overall performance depends on container and platform technology/UI frameworks

Web applications

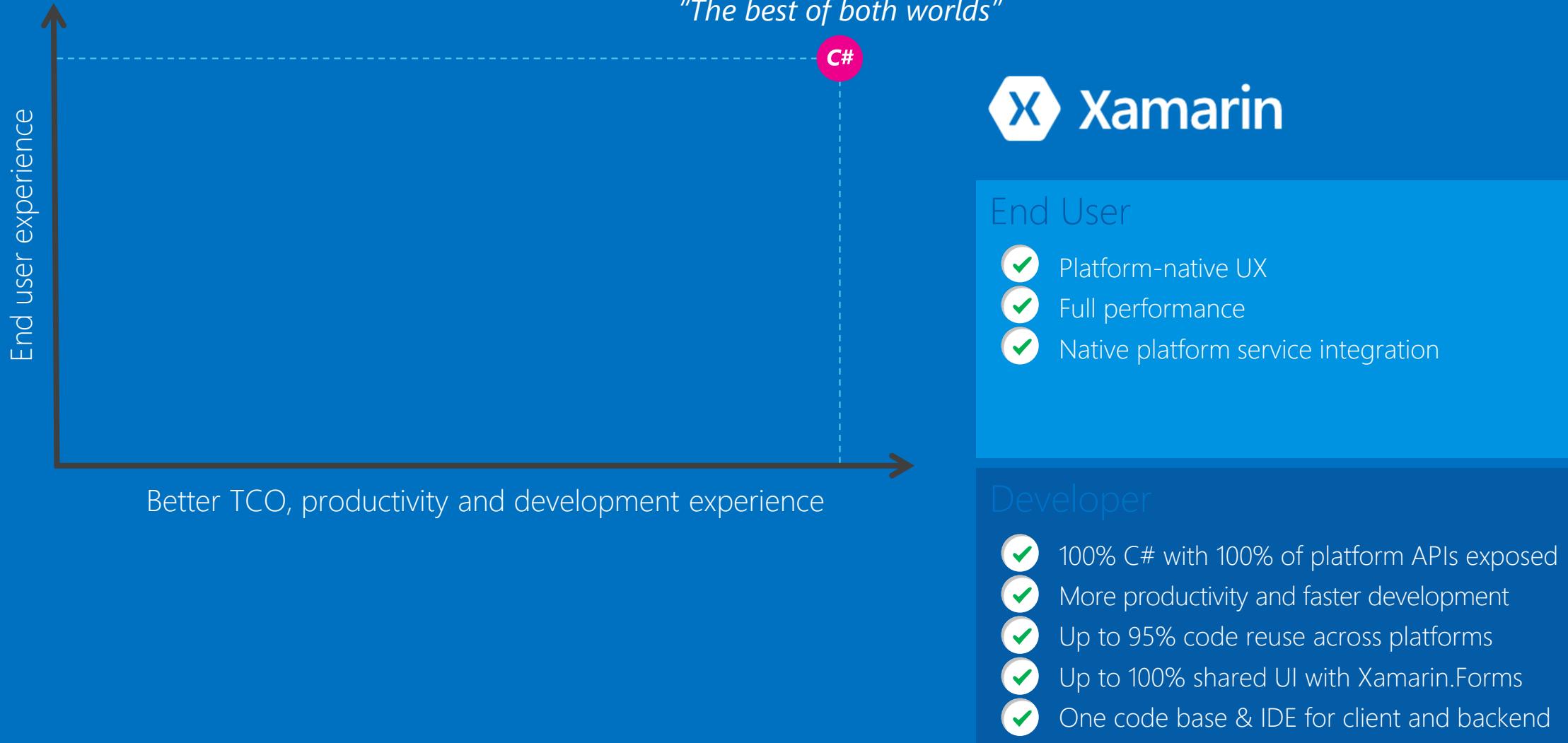
Browser-based web apps and mobile sites

Cross-platform: write once, run everywhere

Best updateability but non-native UX



Cross-platform native with Xamarin



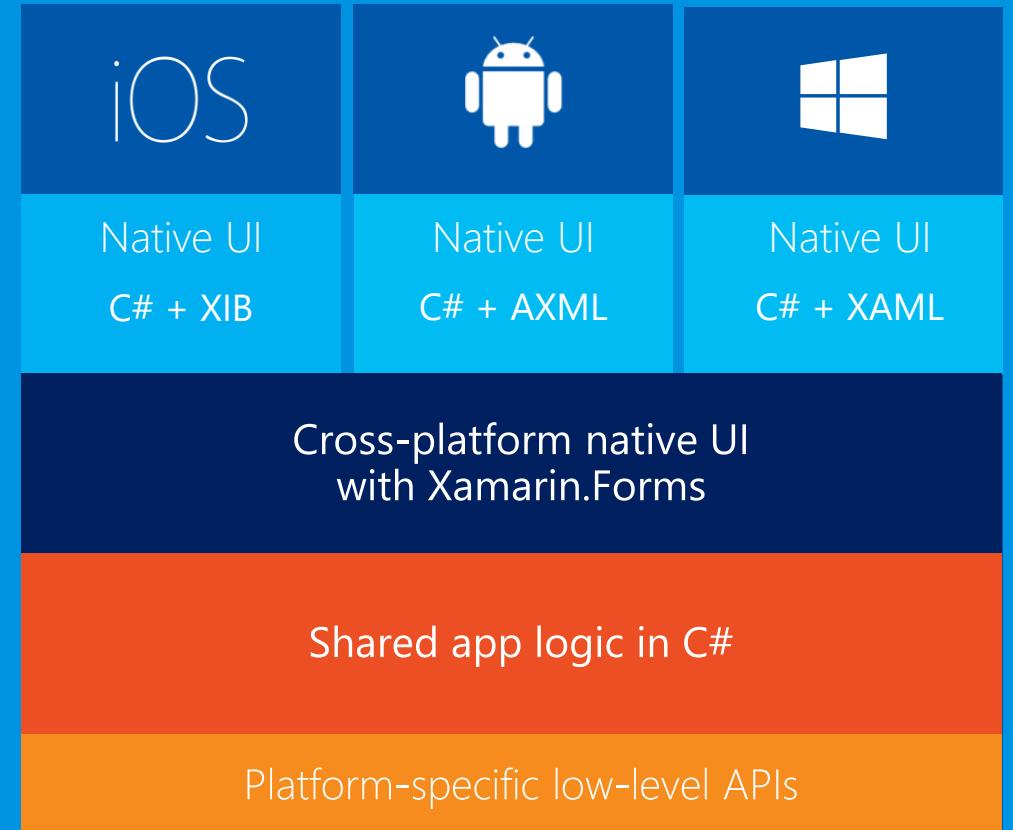
Cross-platform native

Anything you can do in
Objective-C, Swift or Java,
you can do with Xamarin

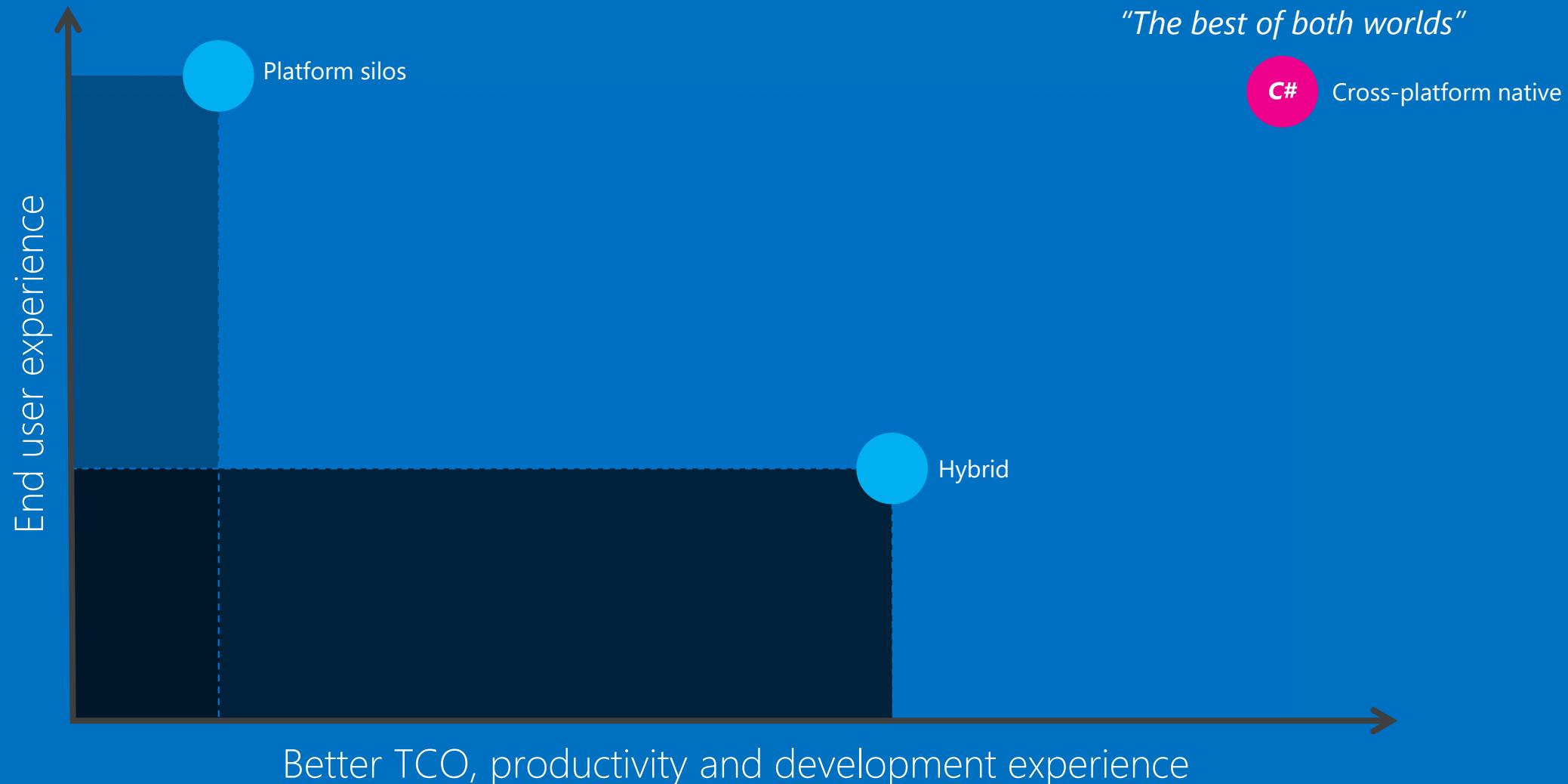
Native user interface

Native performance

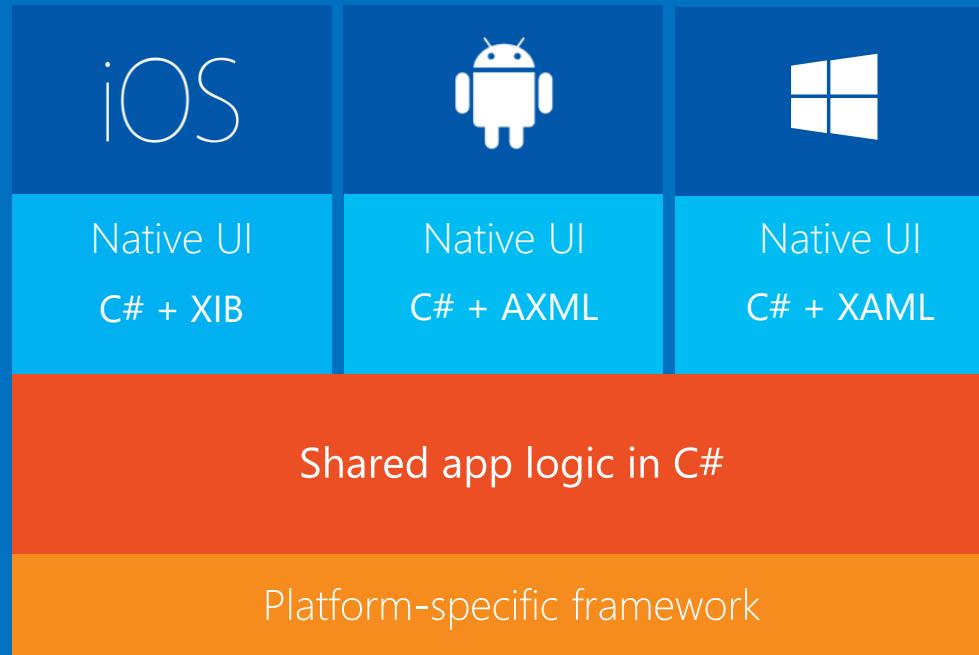
High-fidelity API access



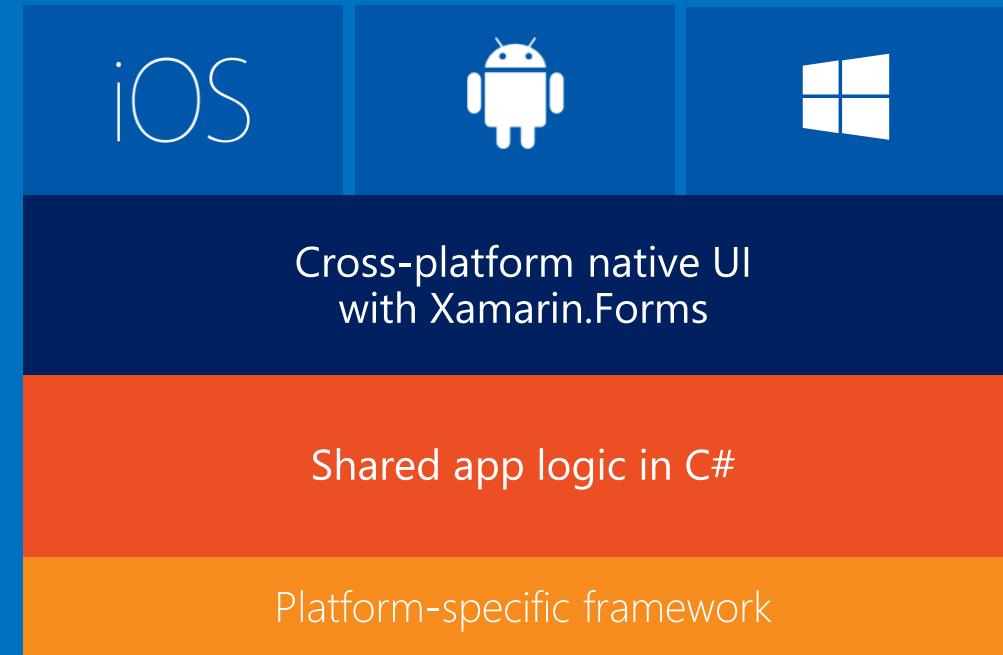
Summary: mobile client approaches



Xamarin + Xamarin.Forms



Traditional Xamarin approach:
Native UI & shared app logic



With Xamarin.Forms:
More code-sharing, all native

Cross-platform shared UI with Xamarin.Forms

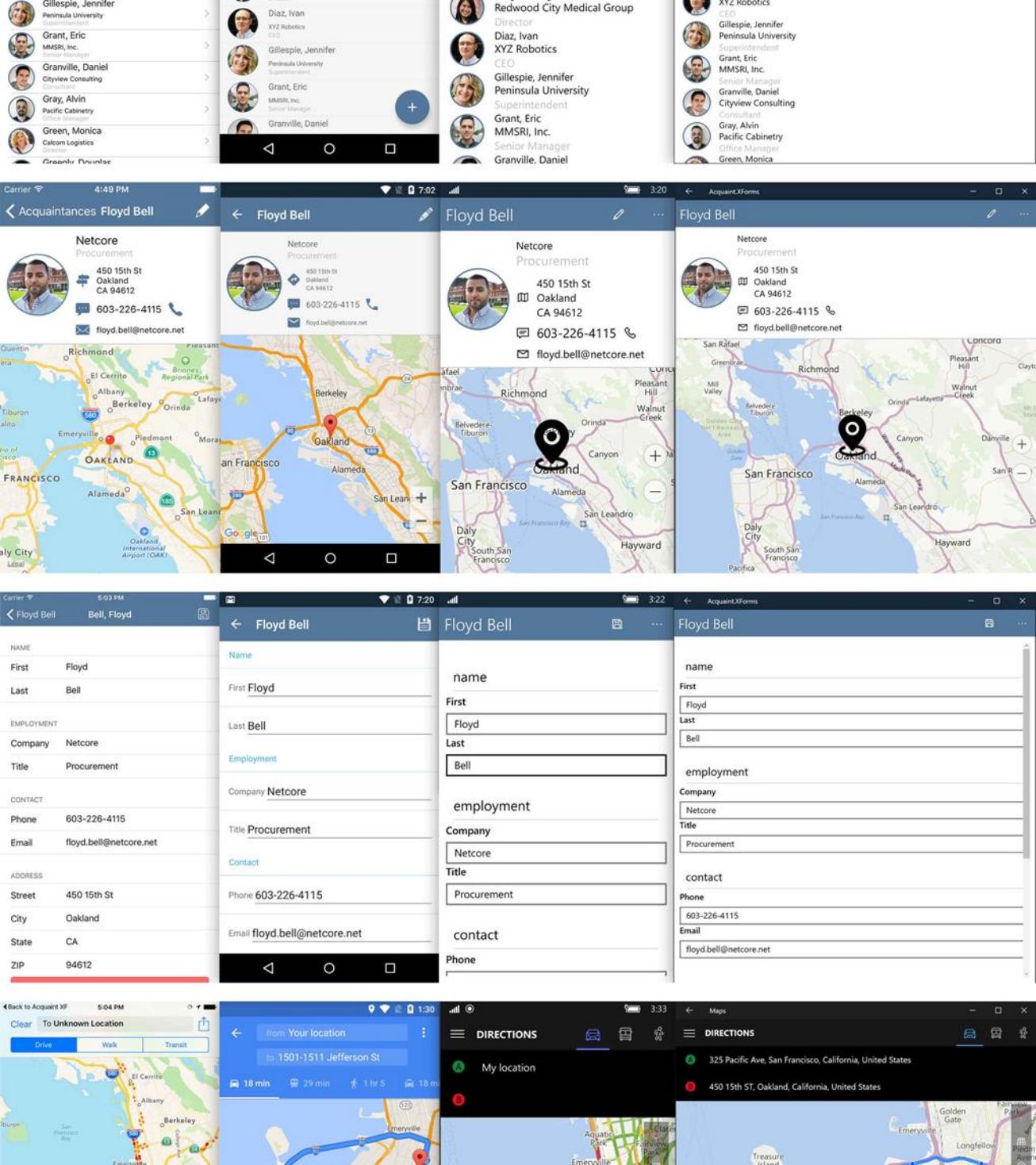
Native UI, cross-platform

Xamarin.Forms is a cross-platform UI toolkit that allows developers to easily create native user interface layouts that can be shared across iOS, Android, and Windows

Build native user interfaces for iOS, Android and Windows from a single, shared C# codebase

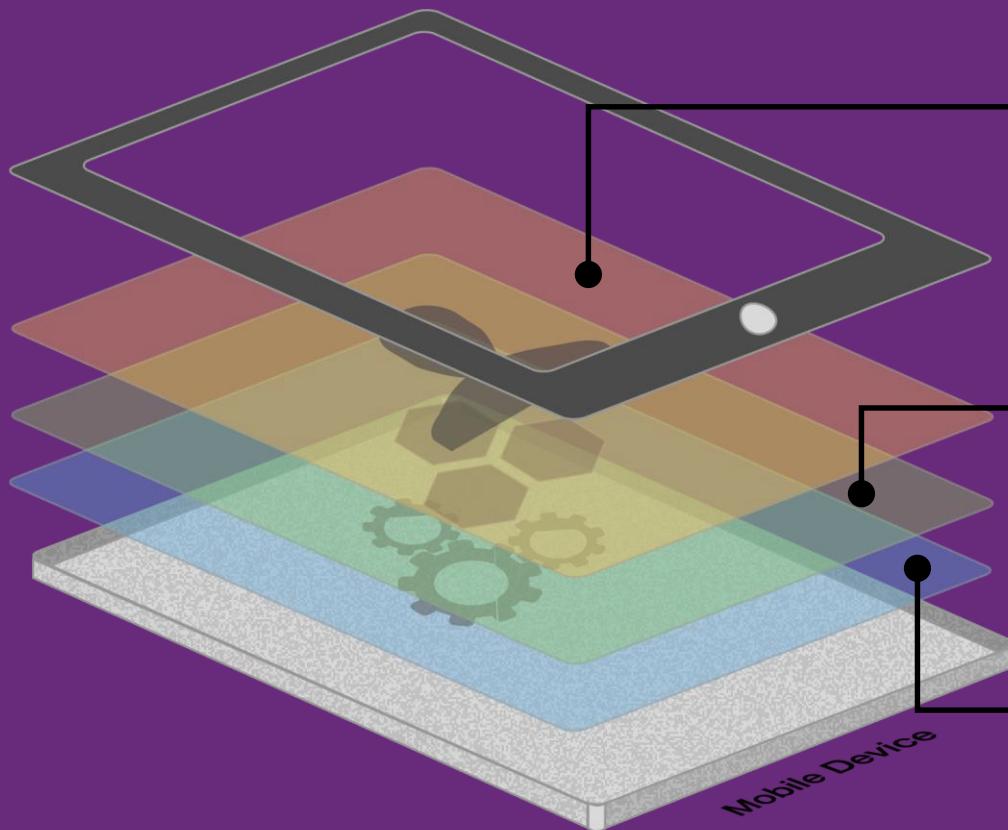
If you know C#, you can build a mobile app for any platform without learning native UI patterns

Over 40 controls and layouts already included



Xamarin is native in all 3 ways

The best of all worlds



Native user interface

Apps are built with standard, native user interface controls for easy and familiar interactions



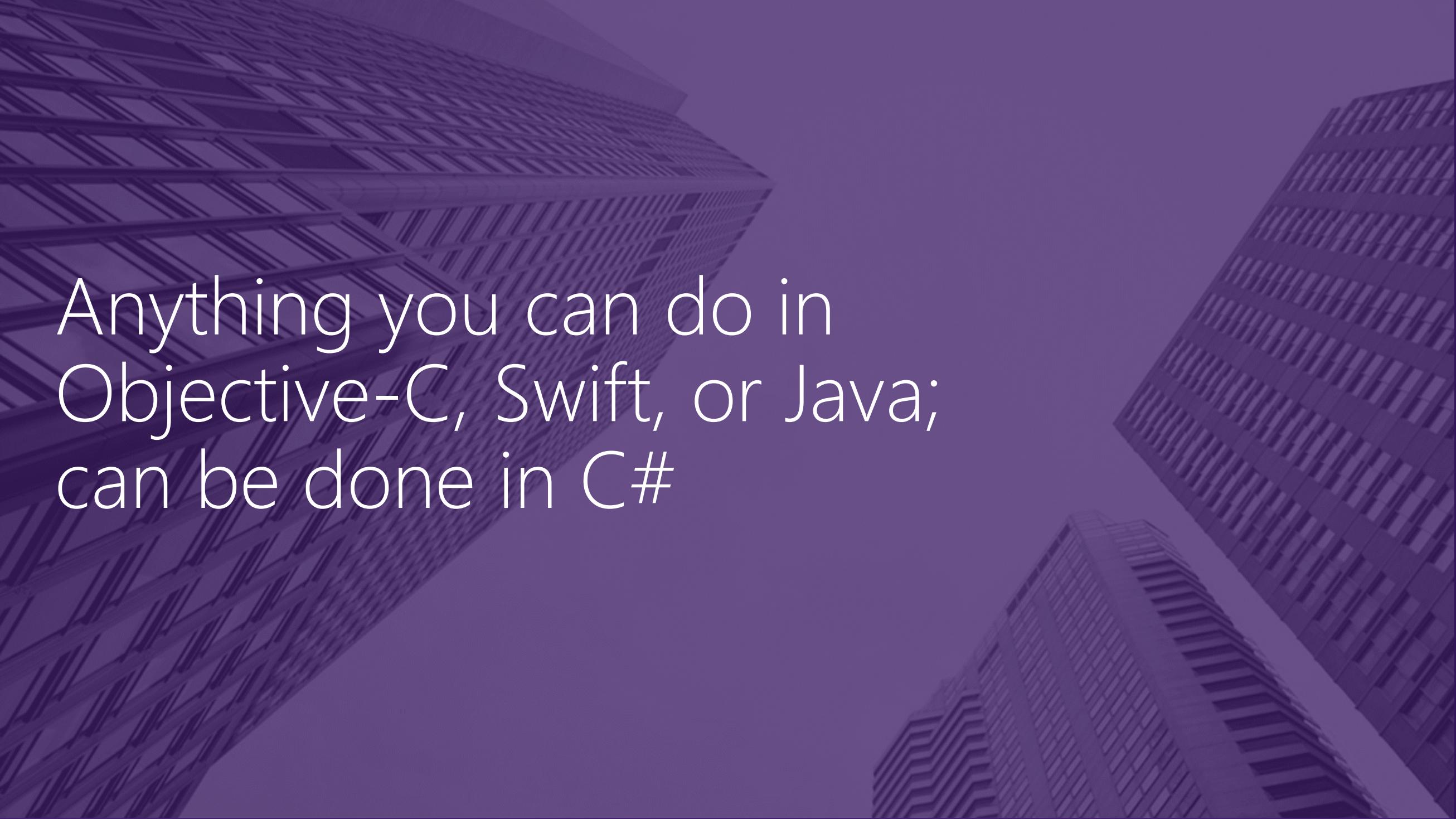
High-fidelity API access

Apps have access to the full spectrum of functionality exposed by the underlying platform and device



Native performance

Apps leverage platform-specific hardware acceleration, and are compiled as native binaries, not interpreted at runtime



Anything you can do in
Objective-C, Swift, or Java;
can be done in C#

24

Azure regions
HyperScale
around the world
footprint

More than AWS and
AZURE REGIONS
Google Cloud combined



Platform Services



Infrastructure Services

Compute



Storage



Networking



Datacenter Infrastructure (24 Regions, 19 Online)



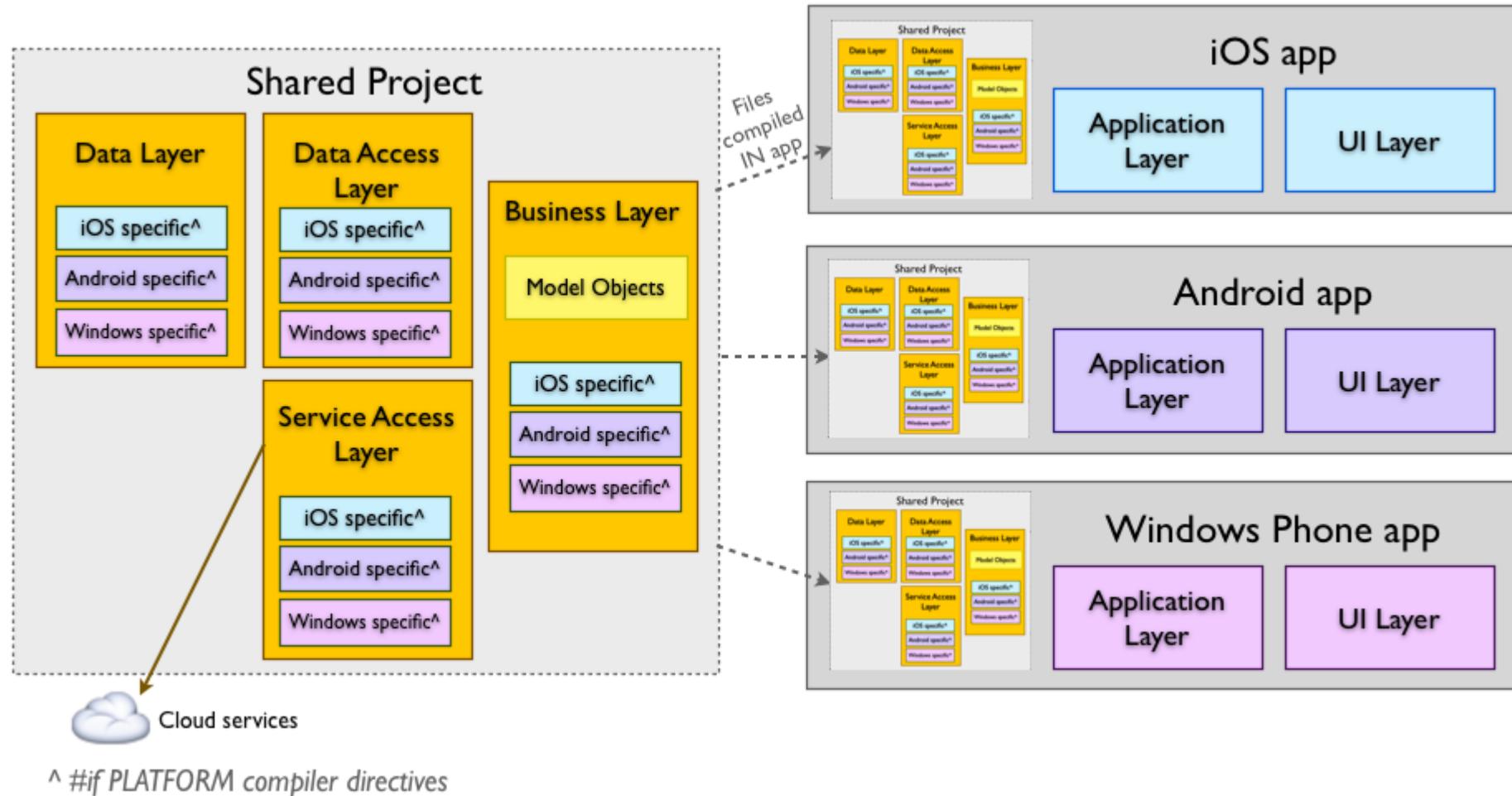
Create a Xamarin Sample Application



Sharing Code

- Shared project
- Portable Class Library (PCL)

Shared Project



Shared Project

- Xamarin projects always define `_MOBILE_` which is true for both iOS and Android application projects

```
#if __IOS__
// iOS-specific code
#endif

#if __TVOS__
// tv-specific stuff
#endif

#if __WATCHOS__
// watch-specific stuff
#endif



---


#if __ANDROID__
// Android-specific code
#endif

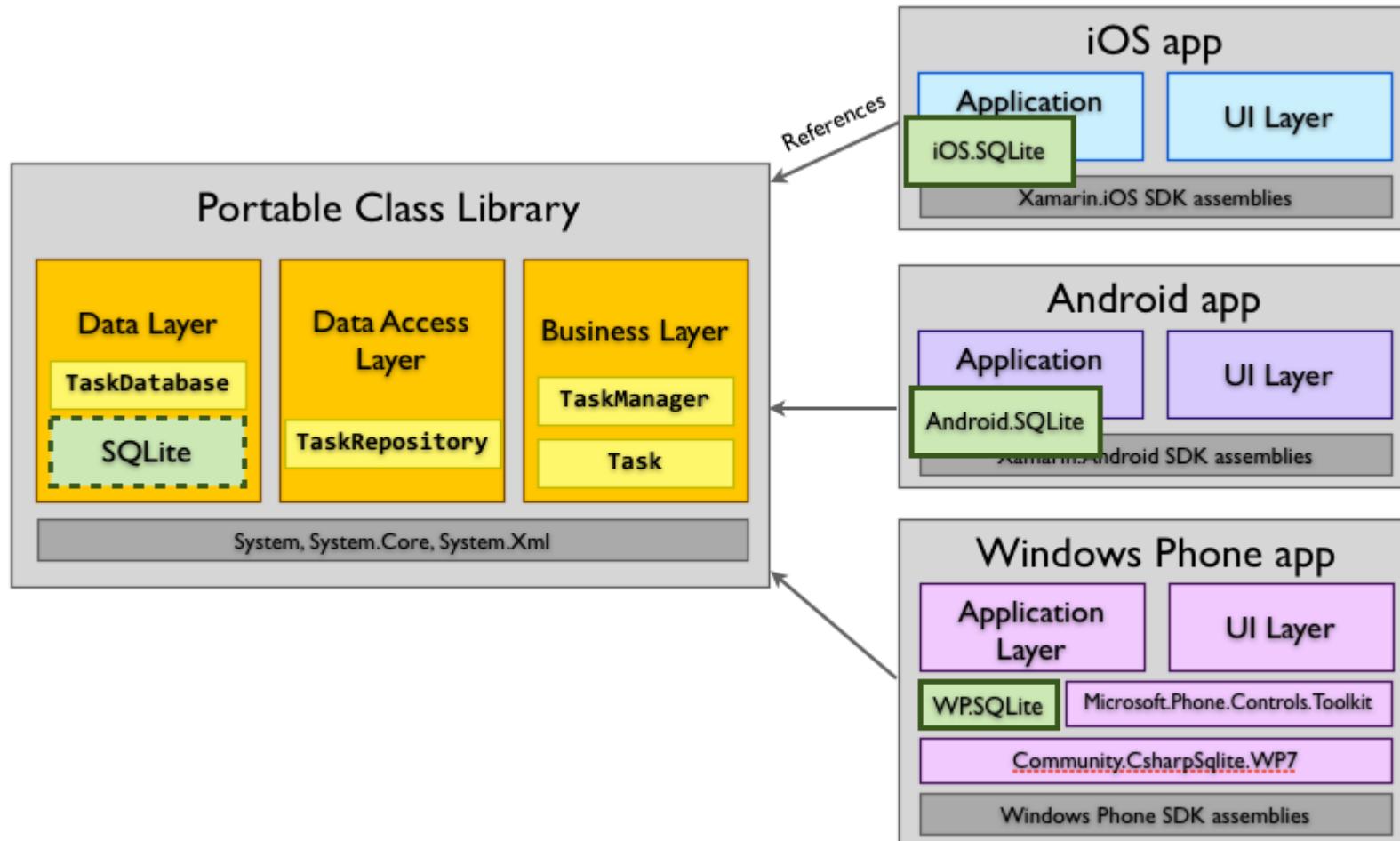
#if __ANDROID_11__
// code that should only run on Android 3.0 Honeycomb or newer
#endif
```

Shared Project

```
public static string DatabaseFilePath {
    get {
        var filename = "TodoDatabase.db3";
#if SILVERLIGHT
        // Windows Phone 8
        var path = filename;
#else

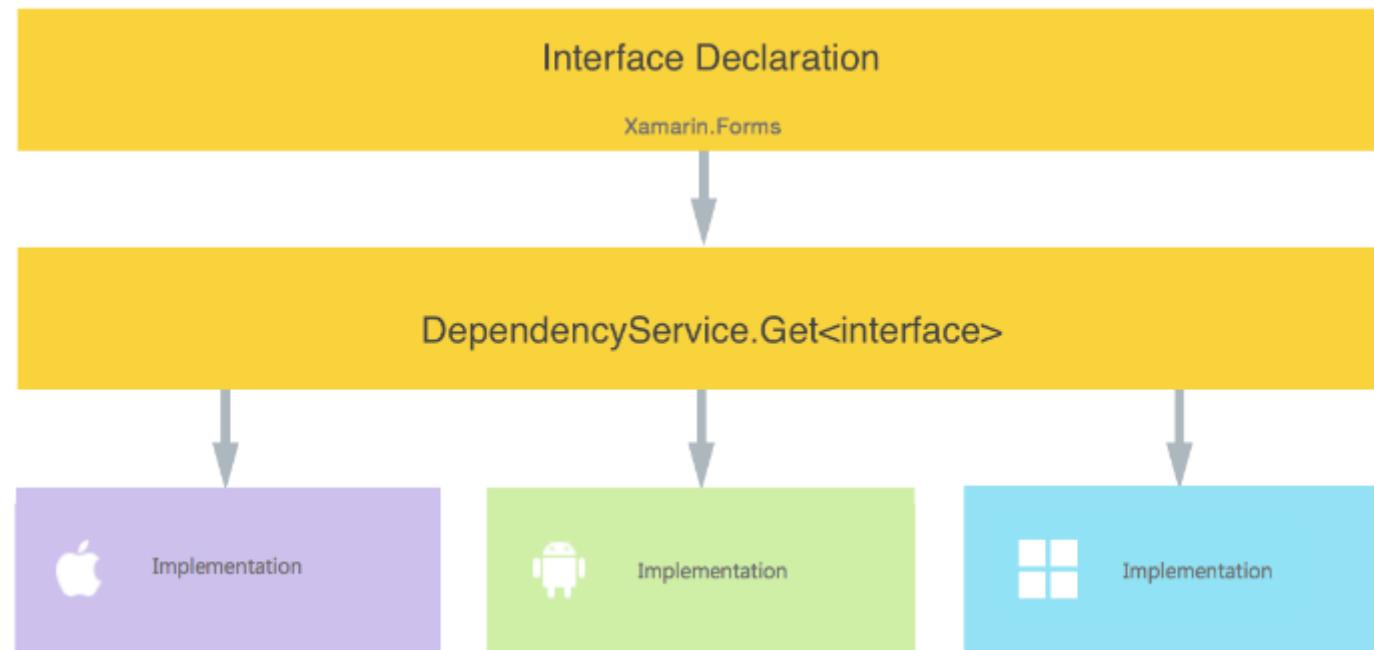
#if __ANDROID__
        string libraryPath = Environment.GetFolderPath(Environment.SpecialFolder.Personal); ;
#else
#if __IOS__
        // we need to put in /Library/ on iOS5.1 to meet Apple's iCloud terms
        // (they don't want non-user-generated data in Documents)
        string documentsPath = Environment.GetFolderPath (Environment.SpecialFolder.Personal);
        string libraryPath = Path.Combine (documentsPath, "..", "Library");
#else
        // UWP
        string libraryPath = Windows.Storage.ApplicationData.Current.LocalFolder.Path;
#endif
#endif
        var path = Path.Combine (libraryPath, filename);
#endif
        return path;
    }
}
```

PCL



PCL – Use DependencyService

- Use DependencyService to get native implementation for shared interface



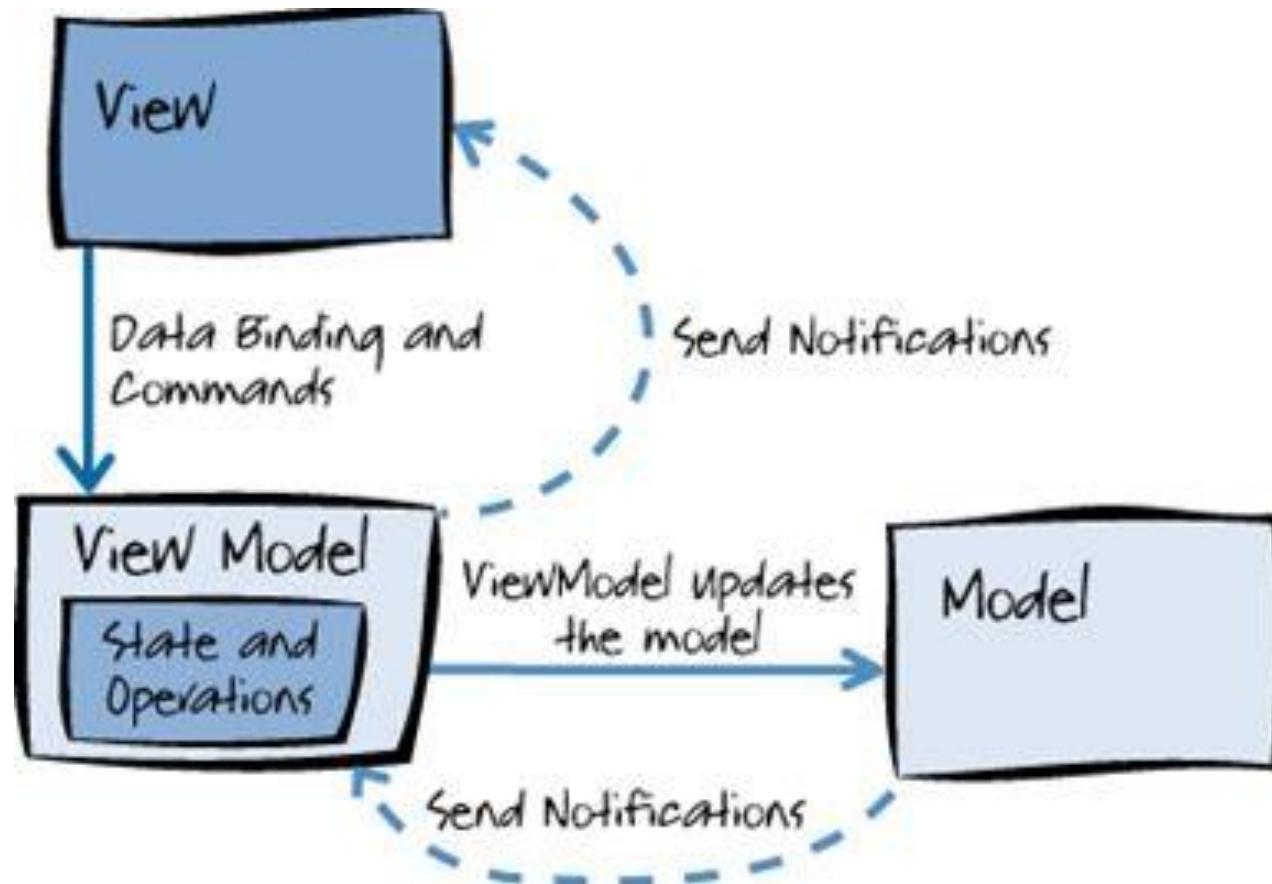
Many PCL libraries covering shared code

- PCL Storage – IO APIs cross platform
- SQLite-net
- Json.net
- Microsoft.Net.Http
- Xamarin.Forms.* by Xamarin / XamarinNuget
- ...

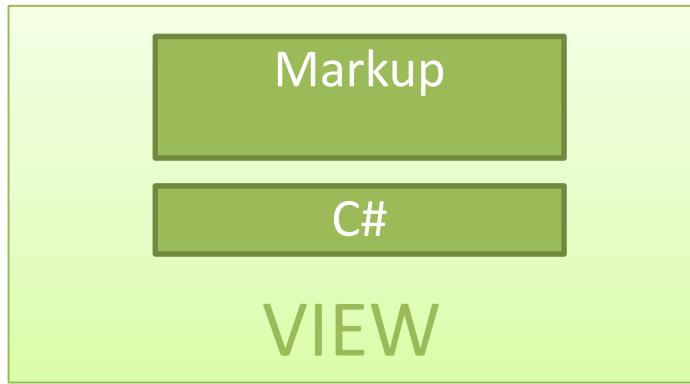
Packages & Components

- Components are for android and iOS, a way to have reusable plugins before PCL packages on NuGet
- Packages = NuGet packages
- Components = Xamarin Components for platform
- Prefer using PCL packages!

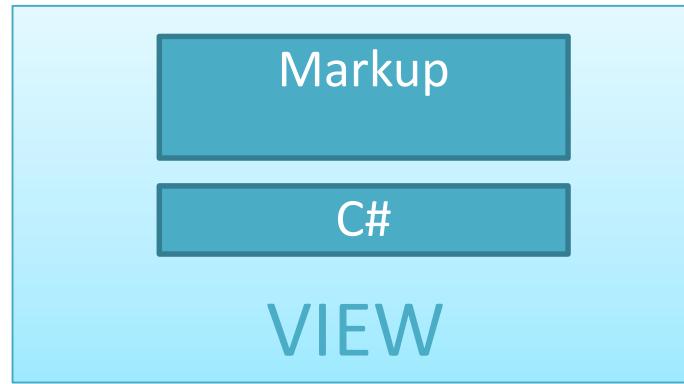
Structuring code - MVVM



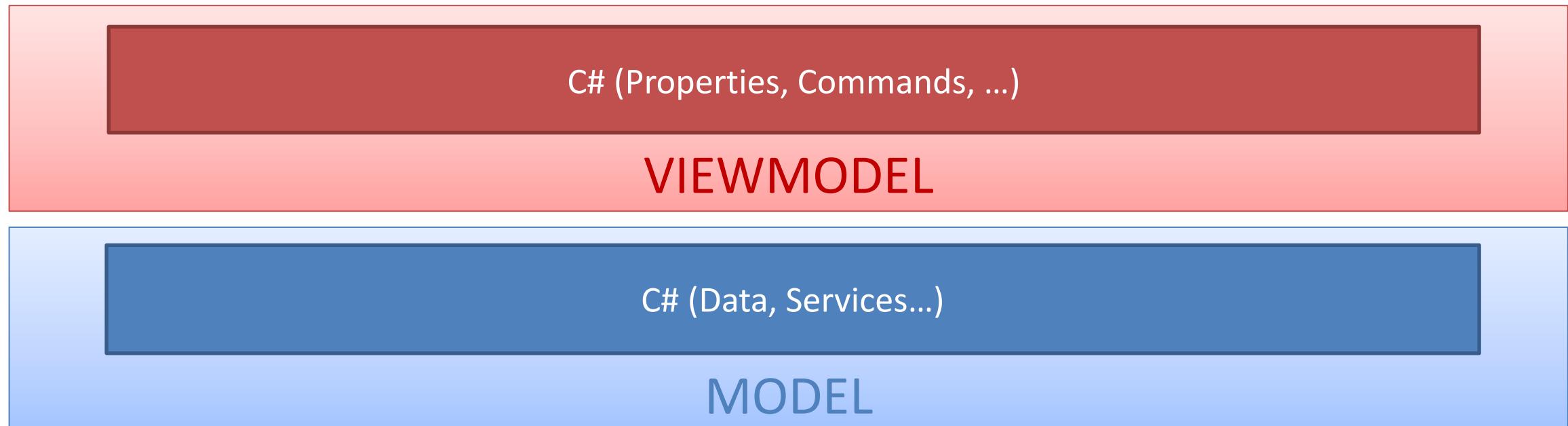
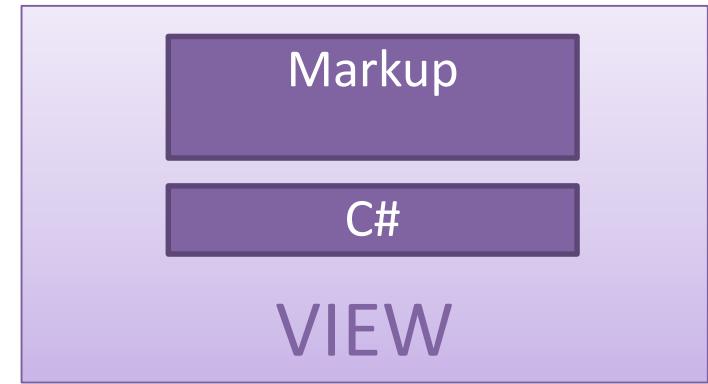
Android



Windows



iOS



Xamarin and MVVM (by Laurent Bugnion)

Frameworks

- MVVM Light
- MVVMCross
- MugenMvvmToolkit
- ..

Demo



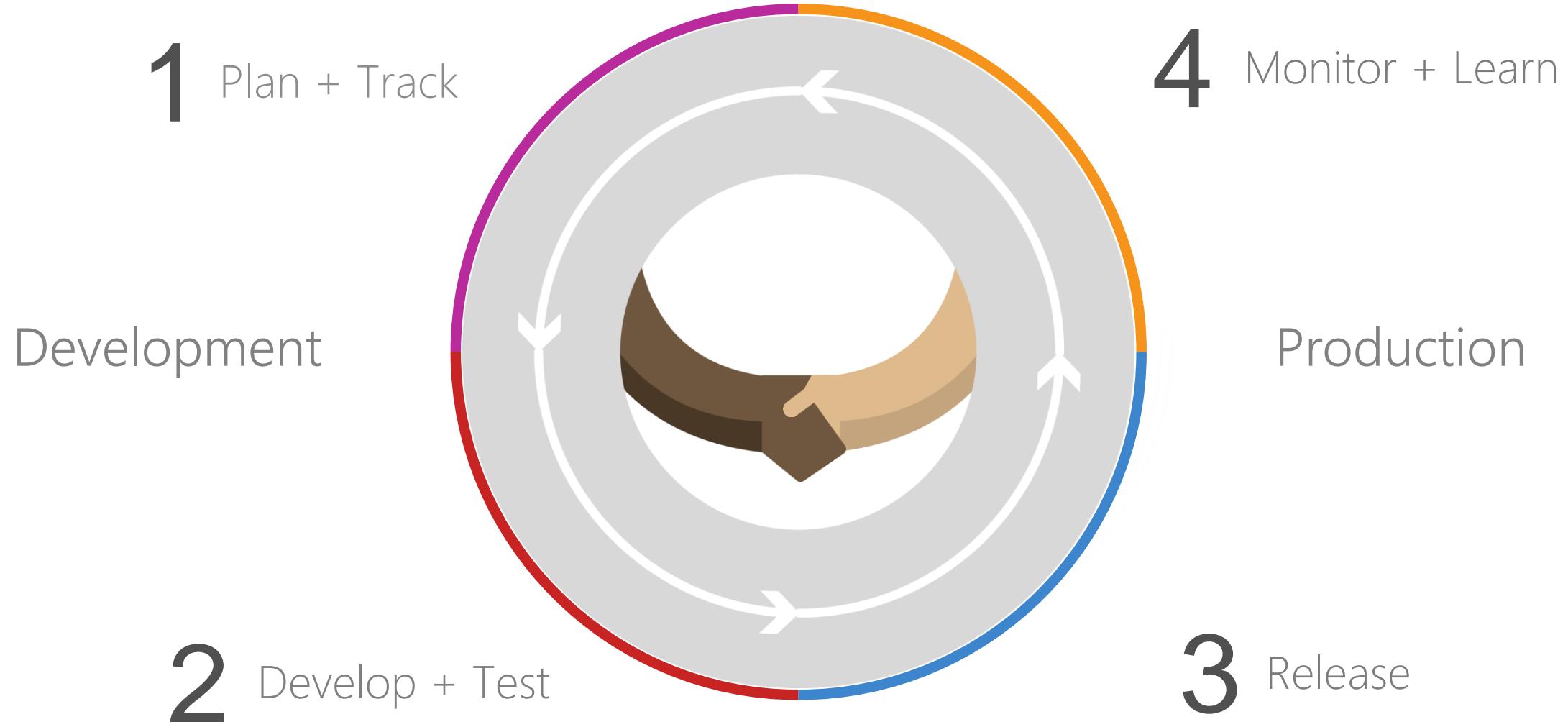
Exercise 1:

<http://bit.ly/2j4DTL6>

Intro VSTS: Setting up Build, Release Management



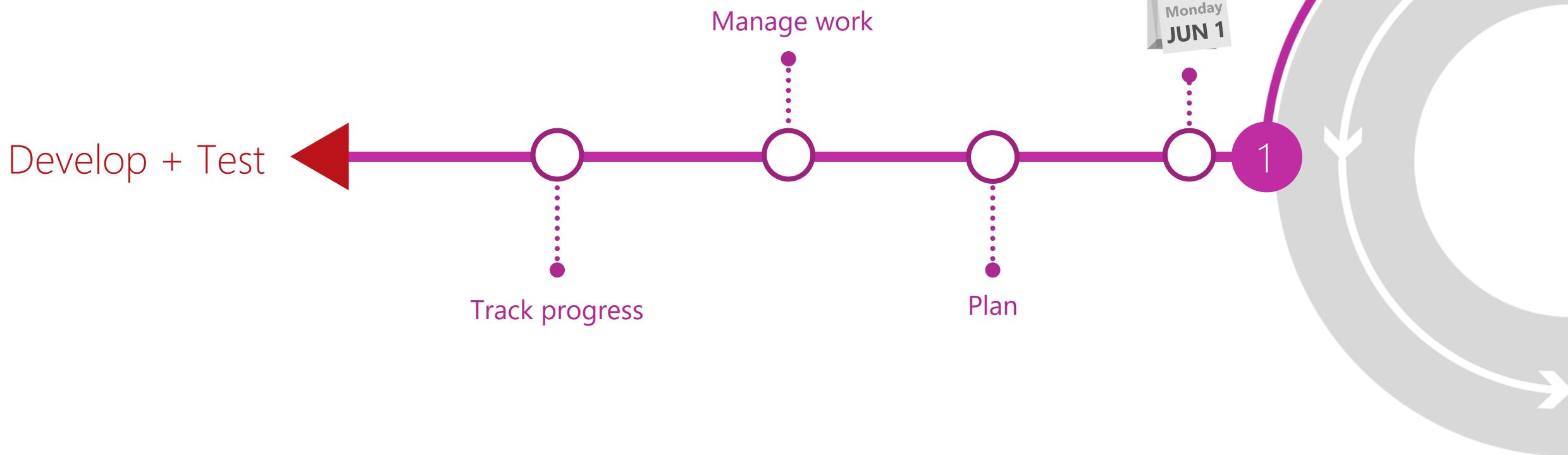
End-to-end DevOps





Plan + Track

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



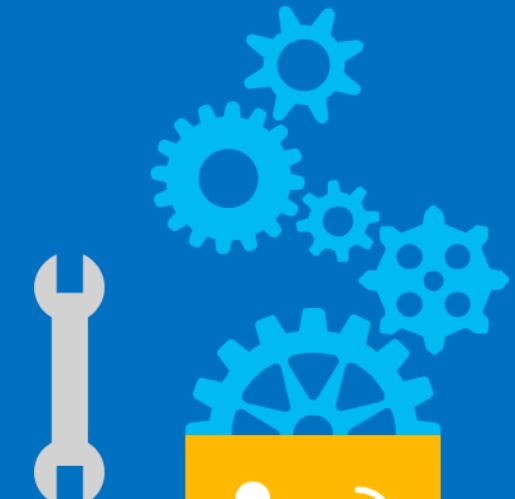
Plan & Track Work

Visual Studio Team Services and Team Foundation Server give you the tools you need to effectively create, manage and deliver against your backlog.

Enterprise collaboration



Agile planning tools



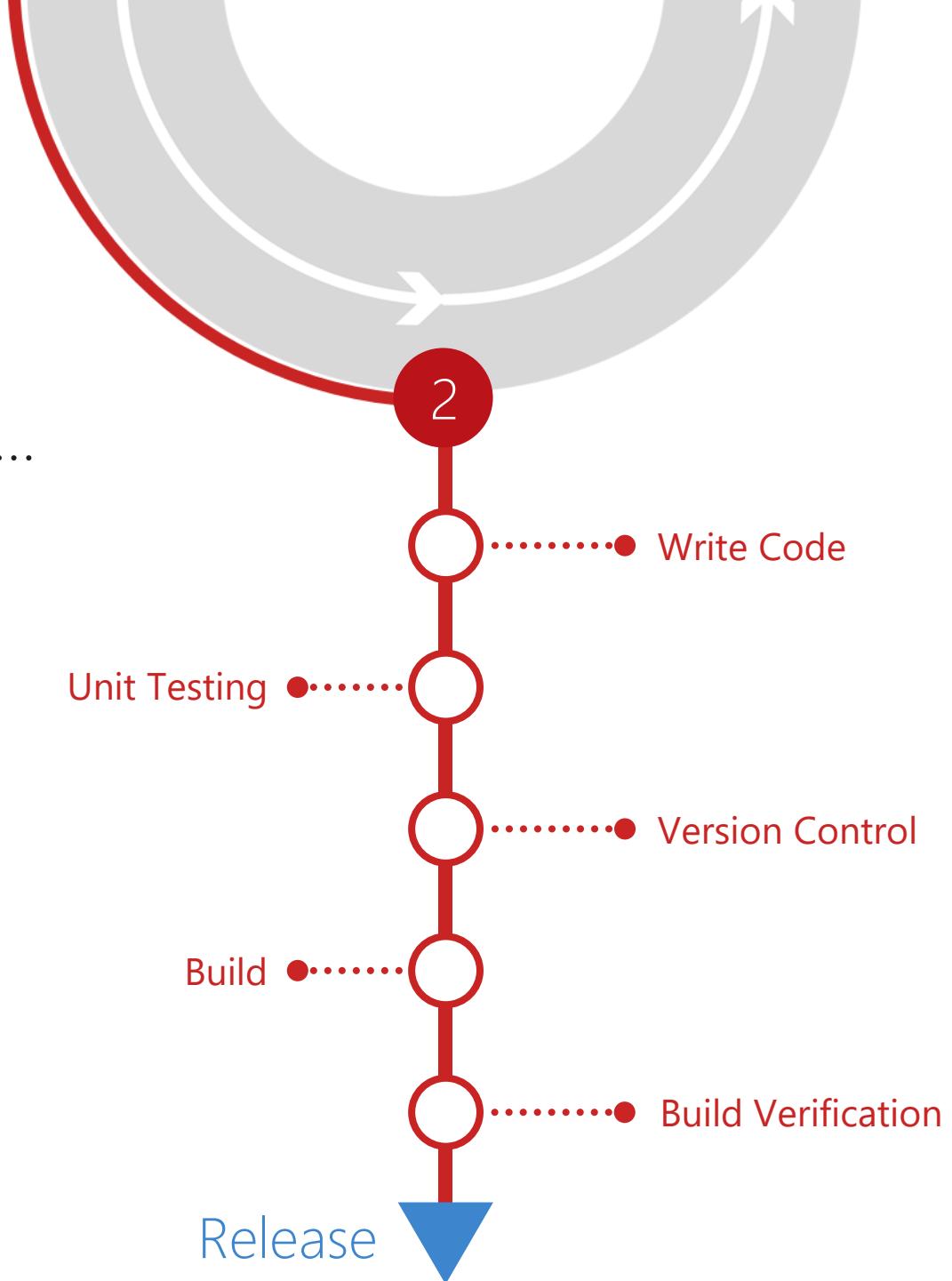
Dashboards & charts





Develop + Test

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



Source Code Management

VSTS and TFS provide unparalleled flexibility for your evolving codebase.

All your code is linked directly to the story, bug, or task driving the work.

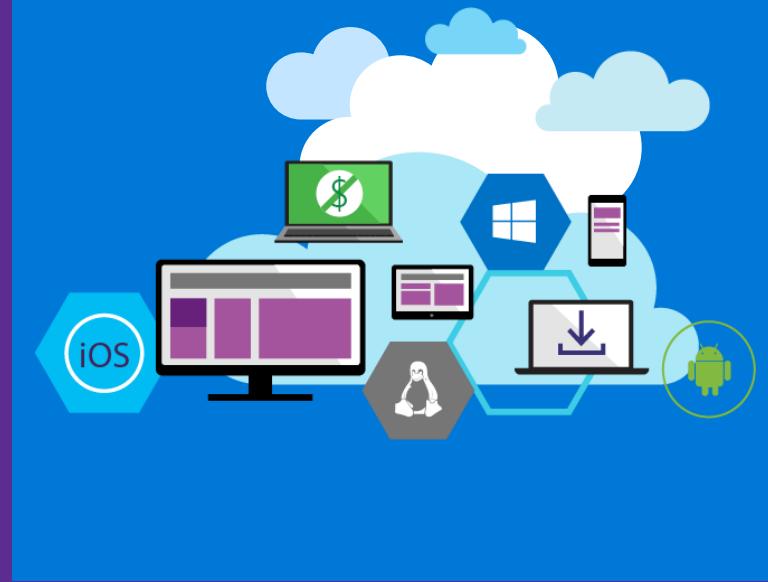
Flexible version control



Collaborate on code



Any IDE, any code



Modern code workflows

Branch and pull requests

Branches isolate risk in a development project. Pull requests provide tools that facilitate collaboration and code reviews for changes being merged back into main.

Effortless collaboration

Rich web experiences for leaving comments in code and approving changes make code reviews simpler and more productive than ever.

Package Management

Share and re-use code across the organization with packages. Private, hosted and authenticated feeds let you share, discover & maintain packages and dependencies.



Any IDE, any code

Eclipse, Xcode, and more

Use your favorite language and development tool. VSTS and TFS version control support any language, as well as any Git client (including Xcode). Java teams can access code and work items through free plugins for Eclipse and IntelliJ – and run continuous integration builds based on configuration files from Ant or Maven.

Code search

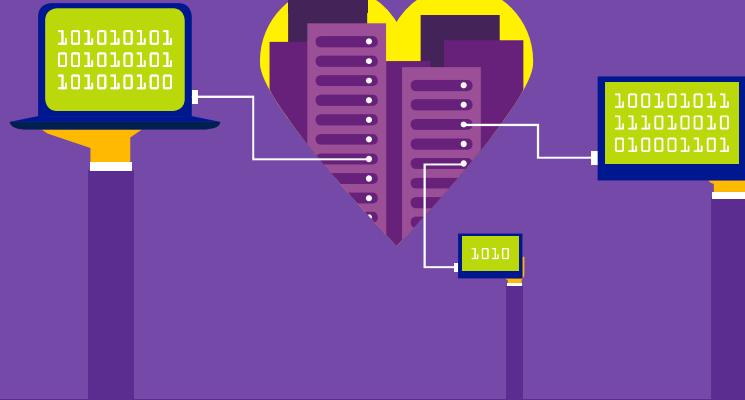
Semantic code search with syntax highlighting enables you to quickly find code across your repositories, as well as the history and when it was changed and by whom.



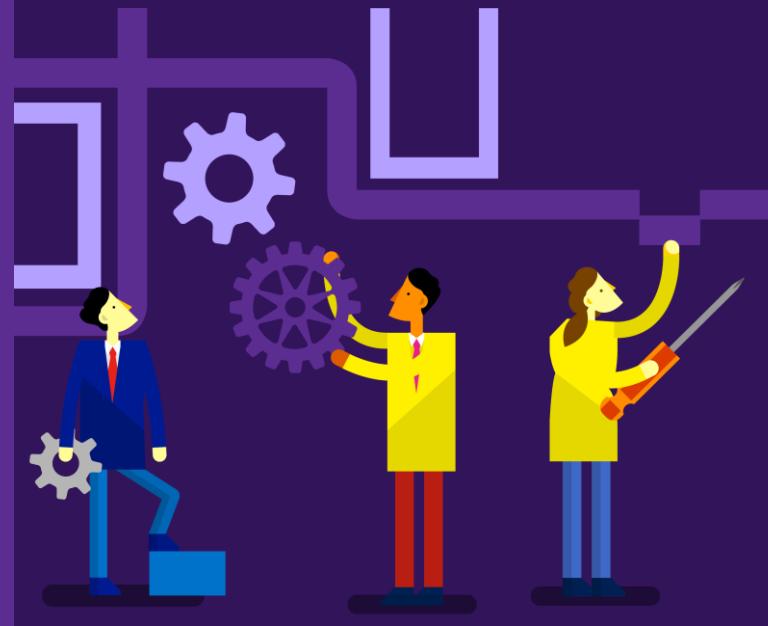
Cross-platform Build

No matter what tools you use or what language you prefer - Team Foundation Build builds your app your way, for your platforms. Just open your browser!

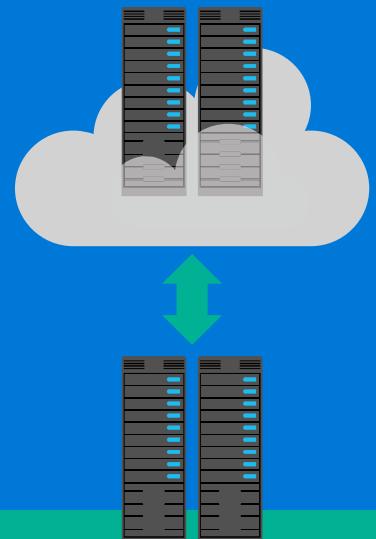
Cross-platform build for
iOS, Java and Android



Flexible, extensible and
customizable builds



Cloud, hybrid or on-
premises infrastructure



No New Build DSL

There are more than enough to go around

Don't make me learn yours just to run mine

ADD BUILD STEPS

 Android Build Run an Android build using Gradle and optionally start the emulator for unit tests		
 Ant Build with Apache Ant		
 CMake Cross platform build system		
 Cmd Script Run a windows cmd or bat script and optionally allow it to change the environment		
 Gradle Run a build using Gra	 VSTest Run tests with VS test runner	
 Jake JavaScript build tool, Node.js.	 Xcode Build Build an Xcode project with xcodebuild tool	
 Maven Build with Apache M	 PowerShell Run a PowerShell script	
 MSBuild Build with MSBuild	 Process Runner Run an executable file	
 Visual Studio Bu Build with Visual Stuc	 Shell Script Run a shell script using bash	
 VSTest Run tests with VS tes	 Azure Cloud Service Deployment via PowerShell Deploy an Azure Cloud Service with PowerShell	
	 Azure PowerShell Run a PowerShell script within an Azure environment	
	 Azure Web Site Deployment via PowerShell Deploy an Azure Web Site with PowerShell	

Cross-platform Build

Any platform, any language

Includes build definition templates to compile .NET, Java, iOS and Android code. A comprehensive library of build steps for a variety of technologies including Ant, CMake, Maven, Xcode and Android enables quick and painless customization of the build process.



No New Build DSL

.NET Core based build agent

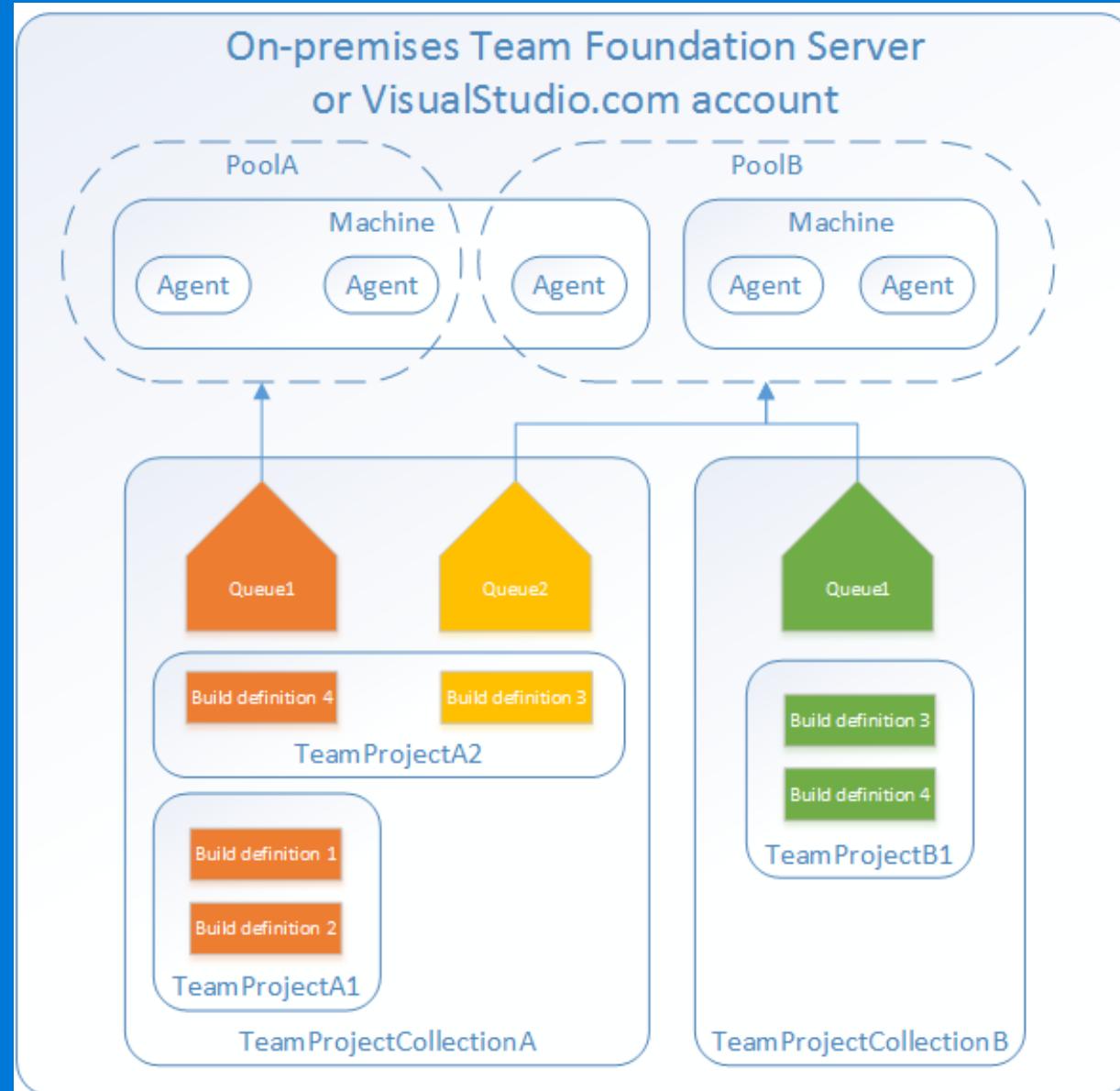
OOB support for common xplat technologies

Open Source

ADD BUILD STEPS

 Android Build Run an Android build using Gradle and optionally start the emulator for unit tests		
 Ant Build with Apache Ant		
 CMake Cross platform build system		
 Cmd Script Run a windows cmd or bat script and optionally allow it to change the environment		
 Gradle Run a build using Gradle	 VSTest Run tests with VS test runner	
 Jake JavaScript build tool, Node.js	 Xcode Build Build an Xcode project with xcdebuild tool	
 Maven Build with Apache Maven	 PowerShell Run a PowerShell script	
 MSBuild Build with MSBuild	 Process Runner Run an executable file	
 Visual Studio Build Build with Visual Studio	 Shell Script Run a shell script using bash	
 VSTest Run tests with VS test runner	 Azure Cloud Service Deployment via PowerShell Deploy an Azure Cloud Service with PowerShell	
	 Azure PowerShell Run a PowerShell script within an Azure environment	
	 Azure Web Site Deployment via PowerShell Deploy an Azure Web Site with PowerShell	

Administer your build and deployment system



Flexible and extensible

Easy customization

Additional build steps and enhancements are available in the Visual Studio Marketplace or can easily be created and customized based on web standards. Edit them in the browser and leverage existing knowledge of popular script languages.

Our web- and script-based build system works with both your windows build agents and the cross-platform build agent (runs on Mac and Linux systems). The cross-platform agent is an open-source Node.js app.

If you need to add a task we don't currently offer, you can create your own activity using our open source activities as examples of best practices.



Cloud, hybrid or on-premises

Cloud build

VSTS provides hosted build agents to start building your projects immediately, as well as a flexible build system that allows you to install your own agents. Our hosted build agents are pre-configured and flexible and can run custom build activities as well as your own build process templates or custom scripts.

Hybrid infrastructure

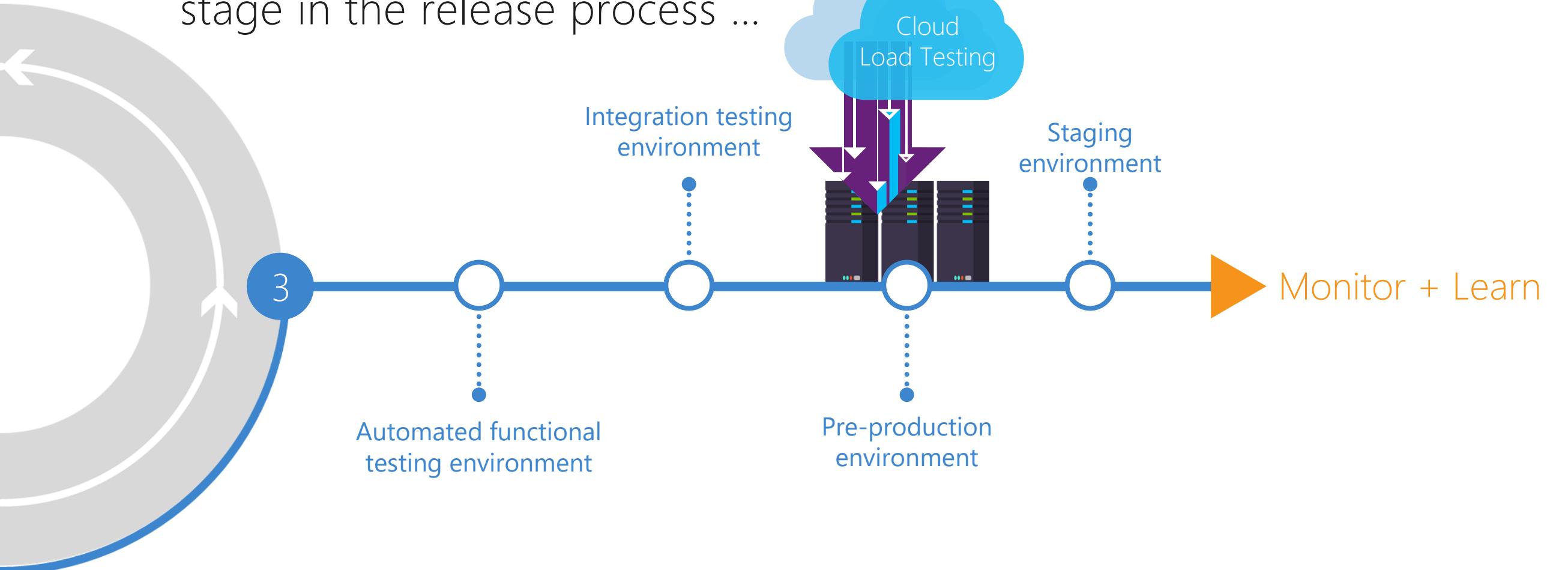
If your build process requires resources outside of the hosted build agents' capabilities, then you can connect on-premises build agents to your hosted team project collection. Or, quickly scale out your on-premises build infrastructure easily by adding hosted build agents.





Release

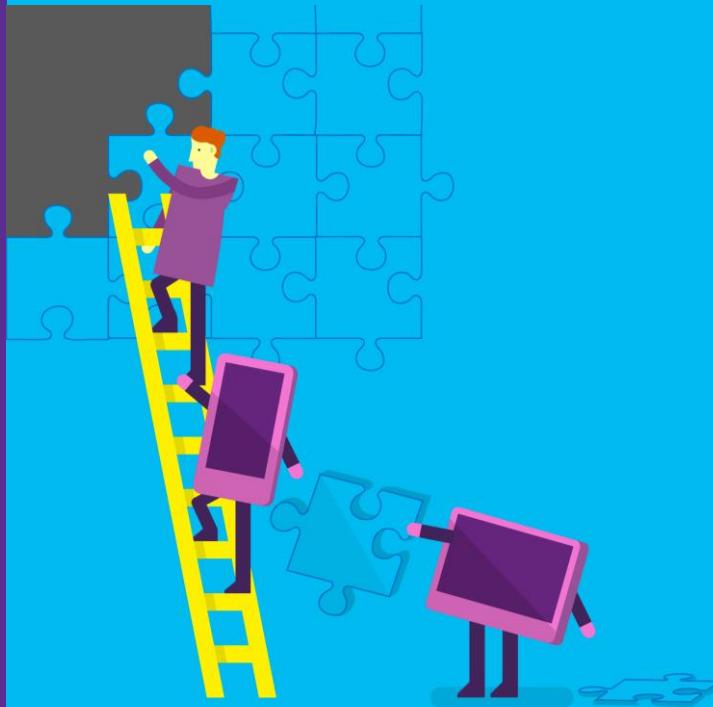
When all tests pass, the build is deployed to testing environments for each stage in the release process ...



Continuous Delivery

Streamline and automate the workflow between development and IT Ops and deliver higher quality software more frequently with less risk.

Continuous Integration



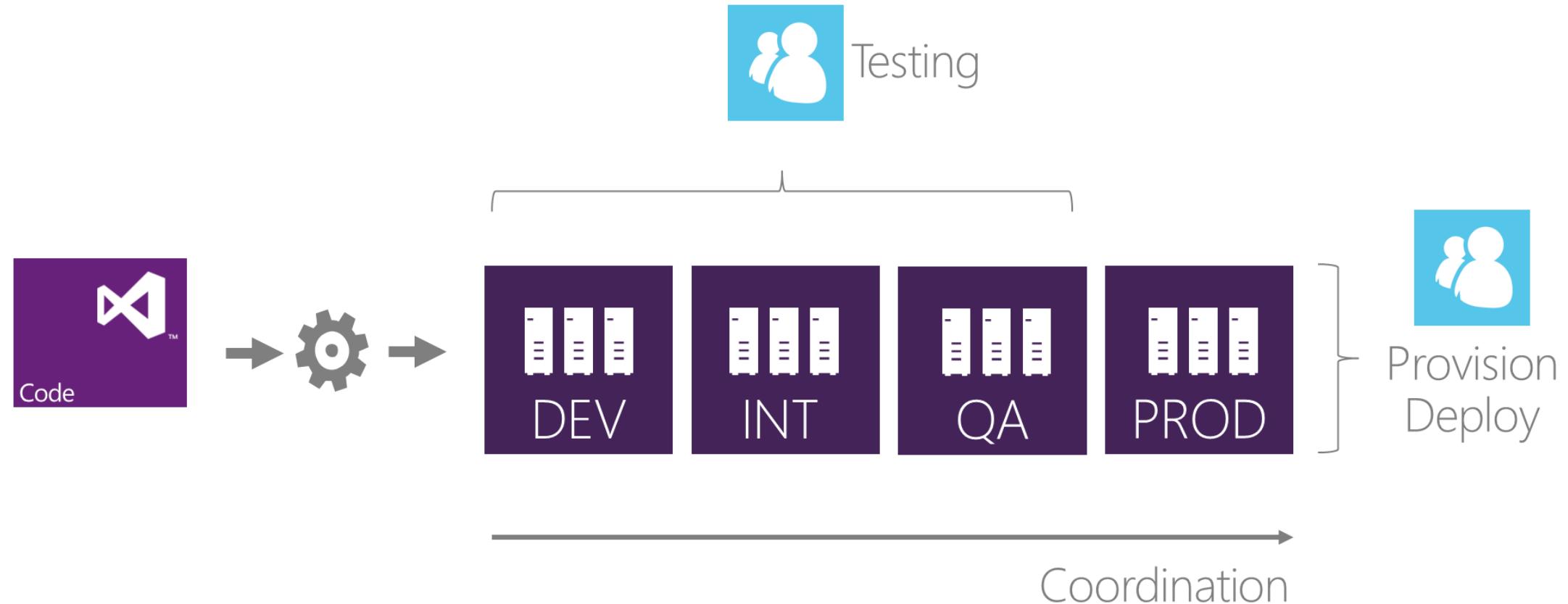
Release Management



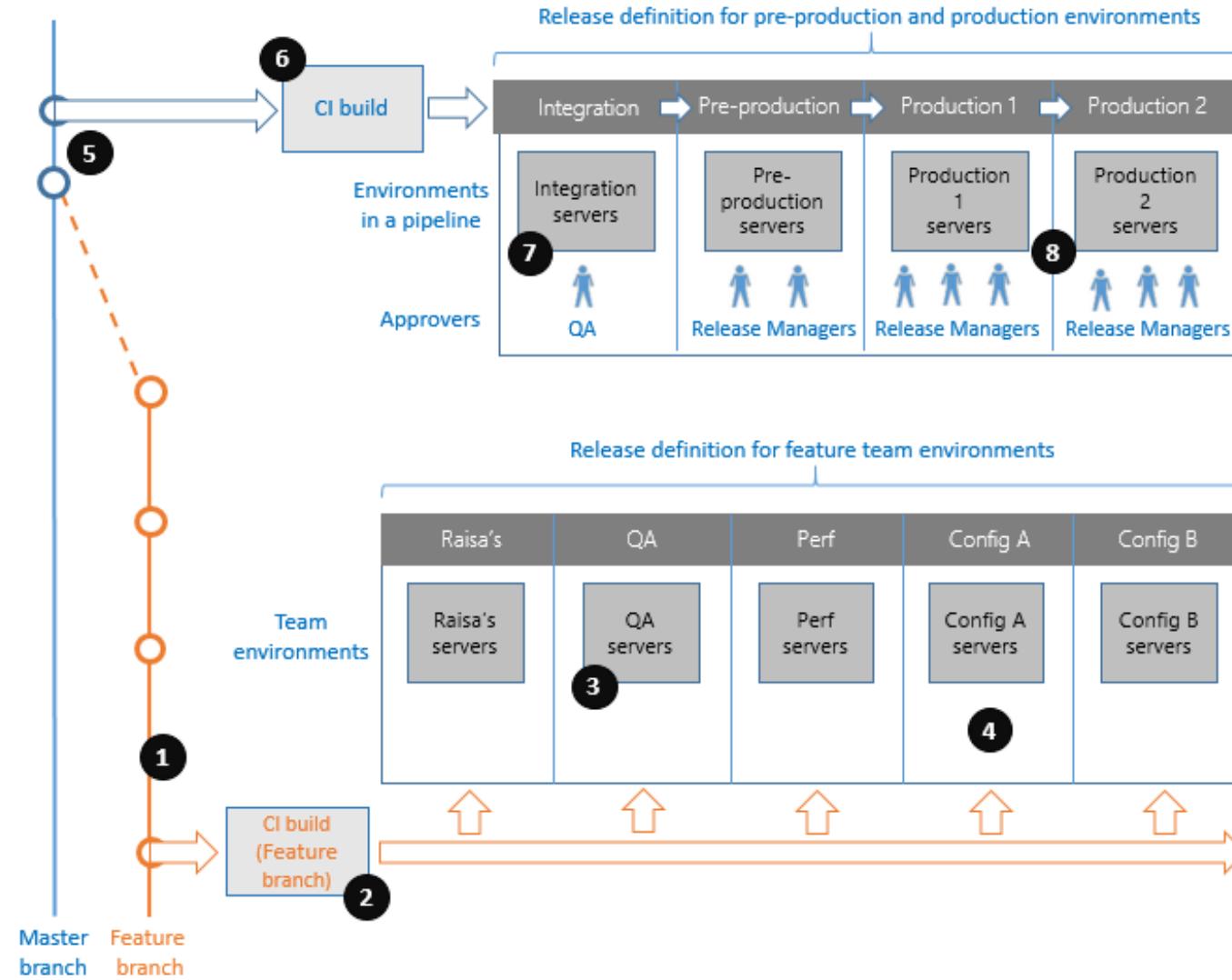
Continuous Quality



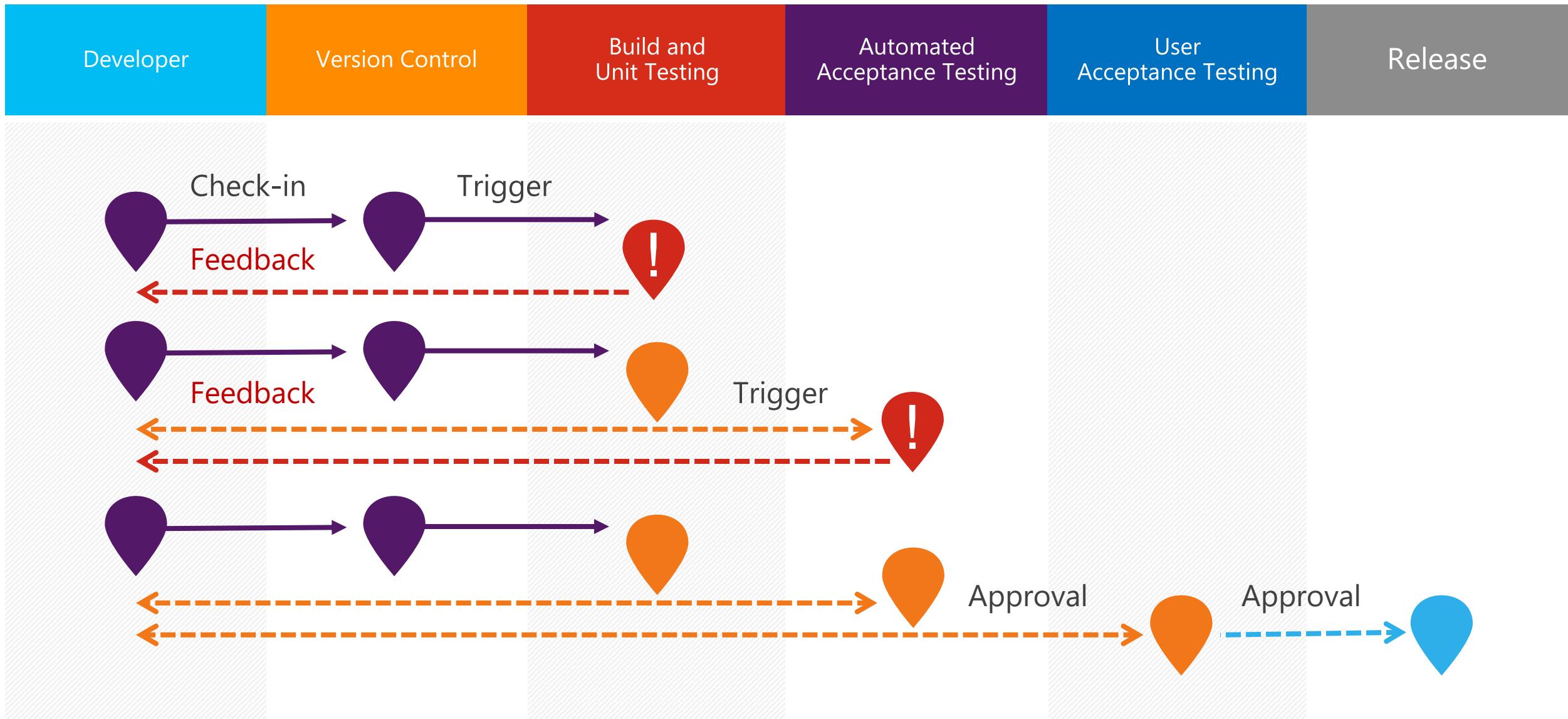
Release Management Overview



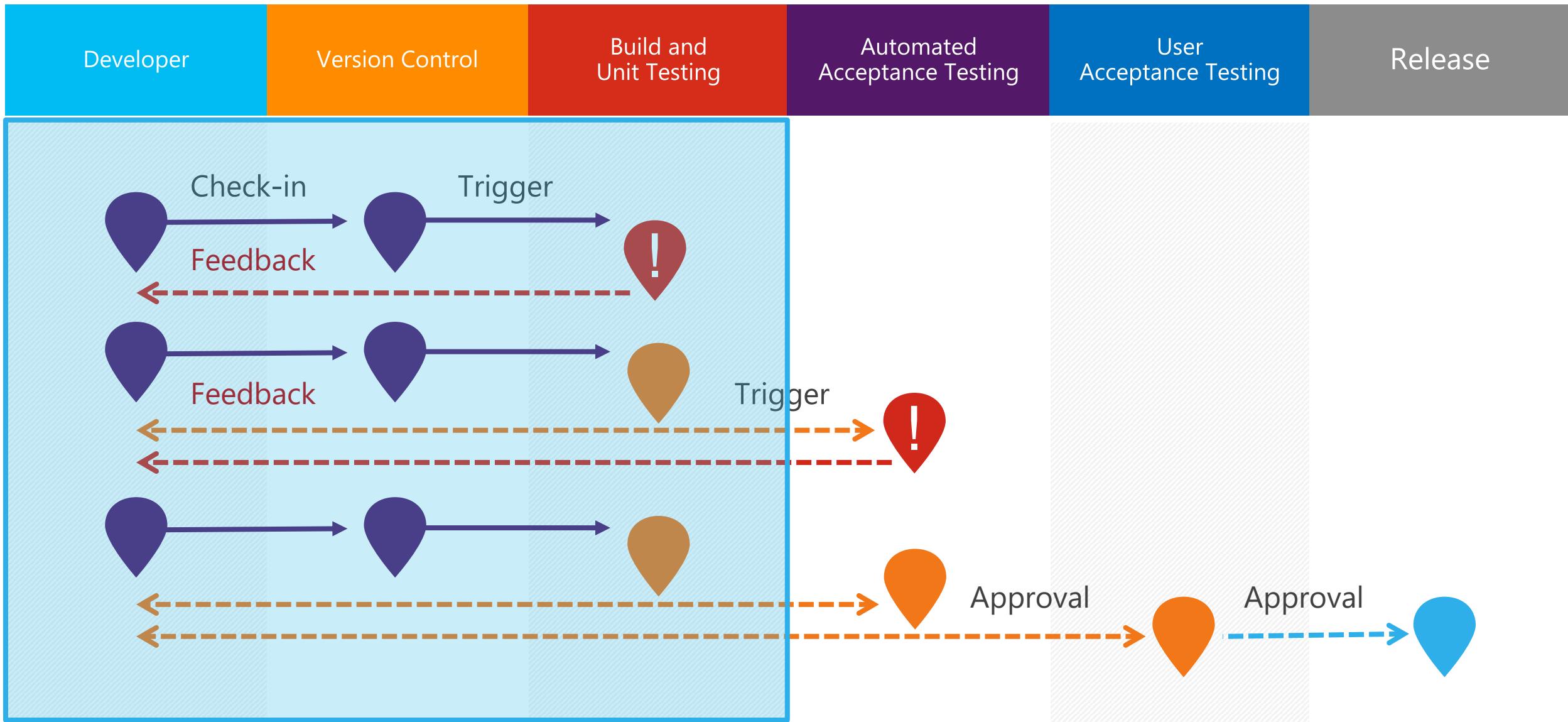
Use case for Release Management



Enabling continuous value delivery



Continuous integration (CI)



Continuous Integration

Build automation

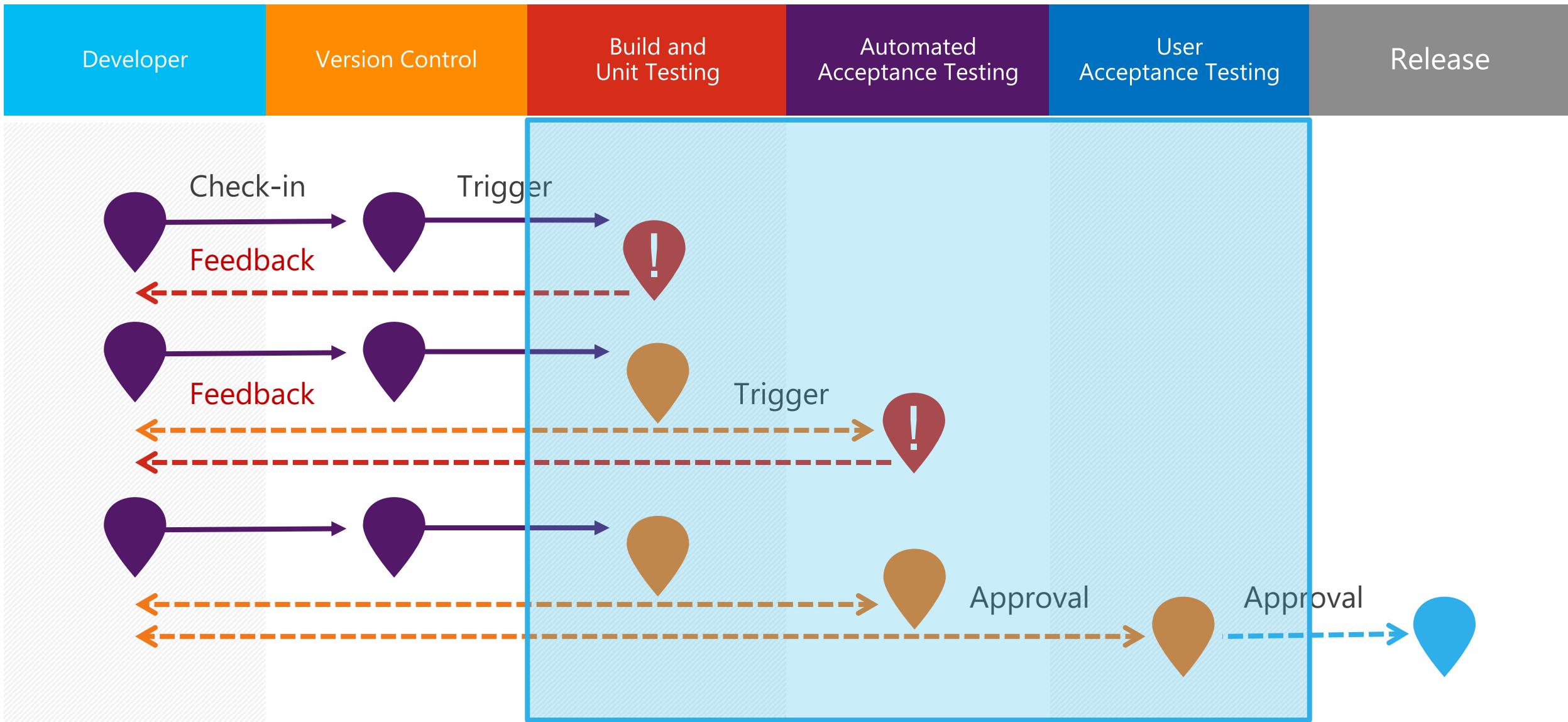
Create and manage build processes that automatically compile and test your application in the cloud or on premises, either on demand or as part of an automated continuous integration strategy.

Integrated

Link builds to test results, work items, code and releases and stop wondering if a bug was fixed in a particular build. By associating work items to code, the work items are listed in the build summary along with code changes and test results.



Continuous quality



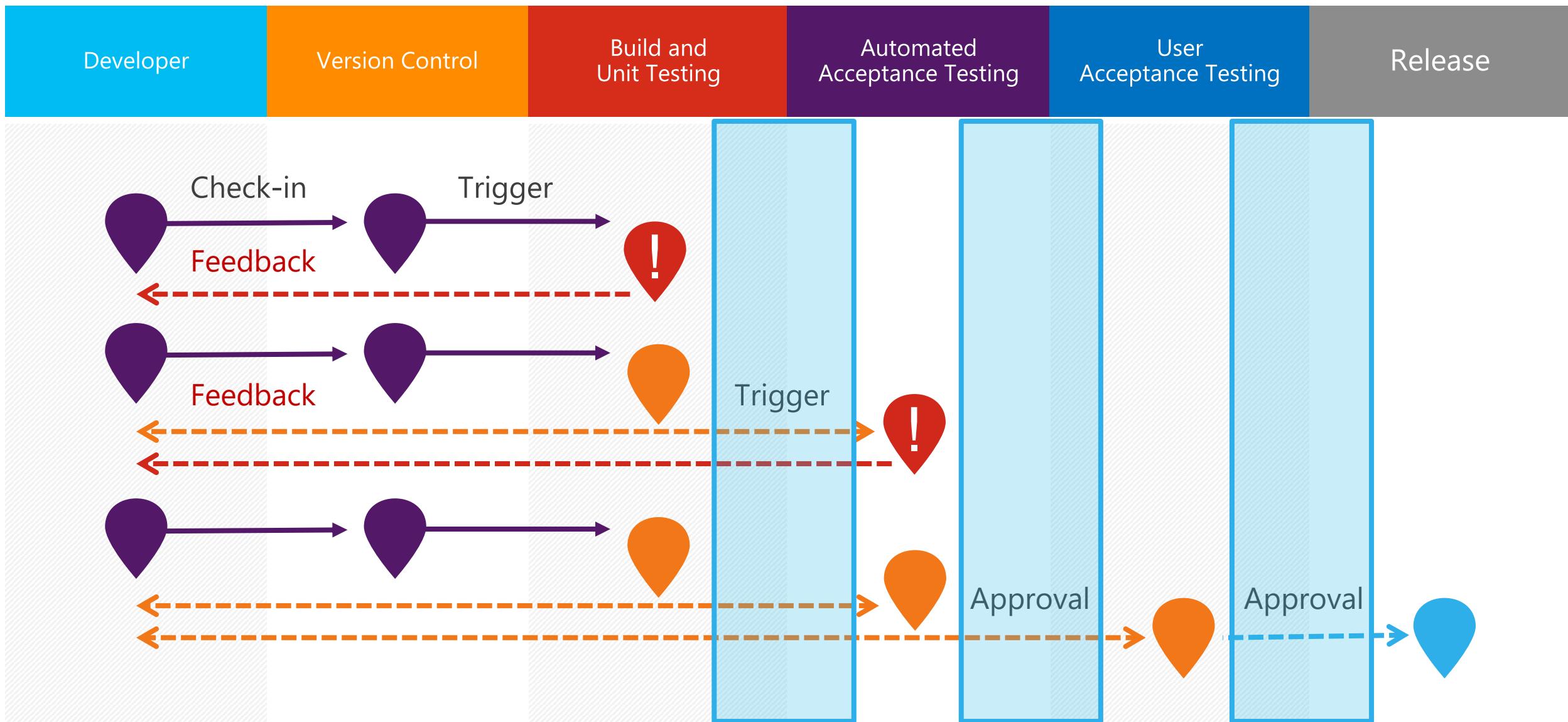
Continuous Quality

Test automation

Choose the technologies and frameworks and test changes continuously in a fast, scalable, and efficient manner. Whether your app is on-premises or in the cloud, you can automate build-deploy-test workflows. Build your TFVC, Git, and GitHub projects from a rich gallery of reusable open source and extensible build tasks that support Ant, Maven, Gradle, etc., and run JUnit, NUnit, xUnit, MSTest, Jasmine, and other types of test frameworks. Use continuous delivery to automatically deploy applications or websites that pass automated test runs and model your release pipelines to match your existing release processes.



Continuous deployment



Release Management

Ship better apps, faster

The Release Management capabilities in VSTS and TFS make it easier to automate deployment and testing of your applications in multiple environments. Teams can author release definitions and automate deployment in repeatable, reliable ways while tracking simultaneous in-flight releases all the way to production.

Full traceability with audit trails

Track the status of recent deployments in each of the environments and retain detailed audit history of all activities performed on a release. Connect to code to see commits and work items that are associated with each release.



Release visibility

Continuous delivery

Eliminate uncertainty, lower risk and get visibility on deployments with Release Management. Easily create automated and repeatable workflows and integrate with any build system to release applications faster with less risk. Audit progress of your release at every stage of the deployment pipeline, all the way to production.

Definition: Main | [Releases](#)

Environments Artifacts Configuration Triggers General History

Save | [Release](#) ▾

+ Add environments

DEV ***

6 / 6 tasks enabled

2 |

QA ***

2 / 2 tasks enabled

2 |

PROD ***

2 / 2 tasks enabled

0 |

+ Add tasks

Azure PowerShell

Azure PowerShell script: \$(System.DefaultW...

Azure Web App Deployment

Deploy Website to Azure

Azure SQL Database Deployment

Deploy Azure SQL DACPAC: Dacpac

Docker

Docker Deployment: namespace/repo_name

Visual Studio Test

Run Tests

PowerShell

Powershell: Post Release Steps

Deploy Website to Azure

Azure Subscription

VSU

Web App Name

DevToolsMa...

Web App Location

Central US

Slot

\$(System.Def...

Web Deploy Package

\$(....Def...

Set DoNotDelete flag

-connectionS...

Additional Arguments

Control Options

Enabled

Continue on error

Always run

[More Information](#)

CONFIGURE - 'DEV' ENVIRONMENT

Approvals Queue View

Approvers

Select the users who can approve

Pre-deployment approver

ADD TASKS

All

Azure Cloud Service Deployment

Deploy an Azure Cloud Service

Add

Build

Azure File Copy

Copy files to Azure blob or VM(s)

Add

Utility

Azure PowerShell

Run a PowerShell script within an Azure environment

Add

Test

Azure Resource Group Deployment

Deploy, start, stop, delete Azure Resource Groups

Add

Package

Azure SQL Database Deployment

Deploy Azure SQL DB using DACPAC

Add

Deploy

Azure Web App Deployment

Publish a Visual Studio Web project to a Microsoft Azure Web App using Web Deploy

Add

Chef

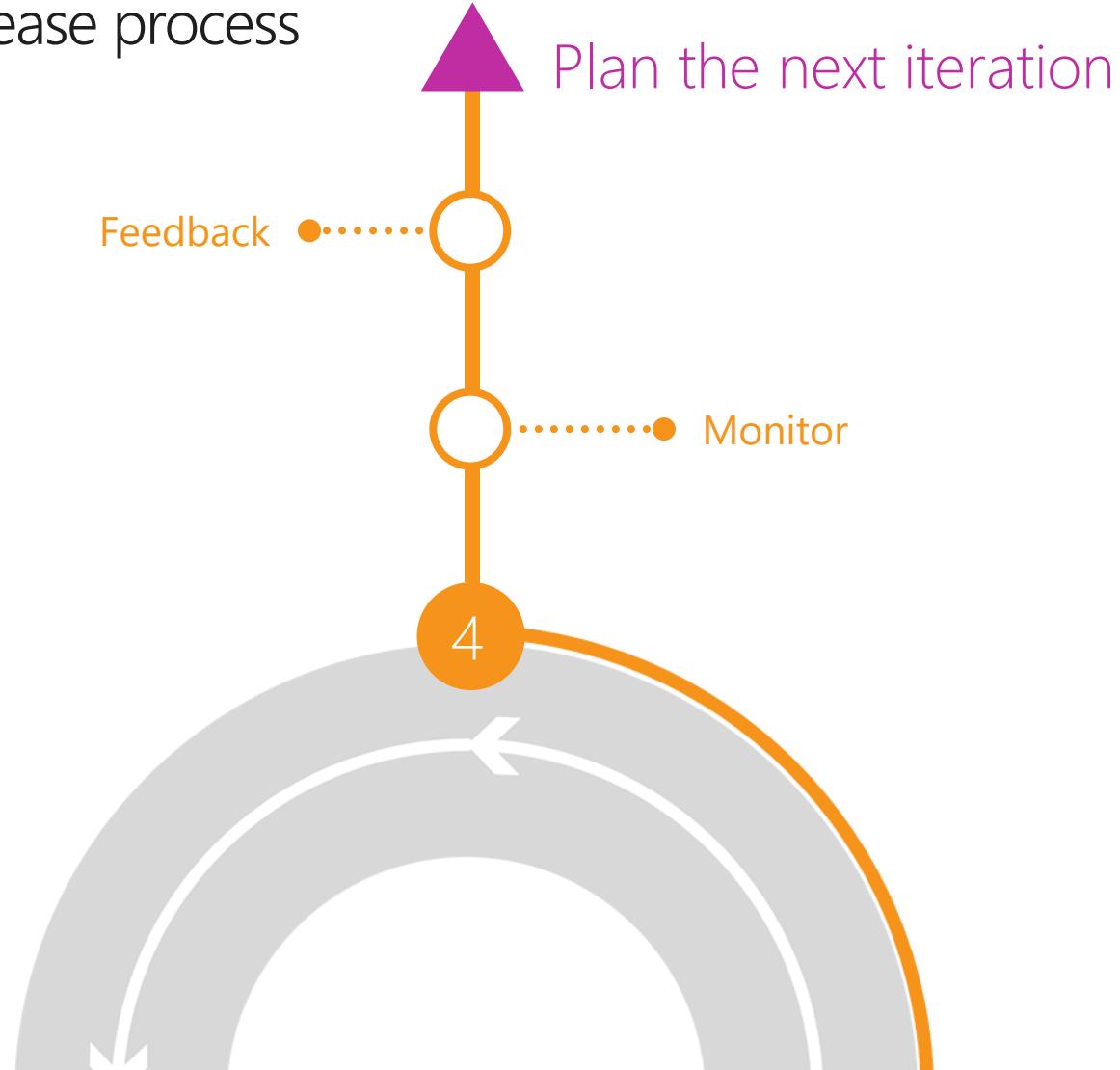
Deploy to Chef environments by editing

Add



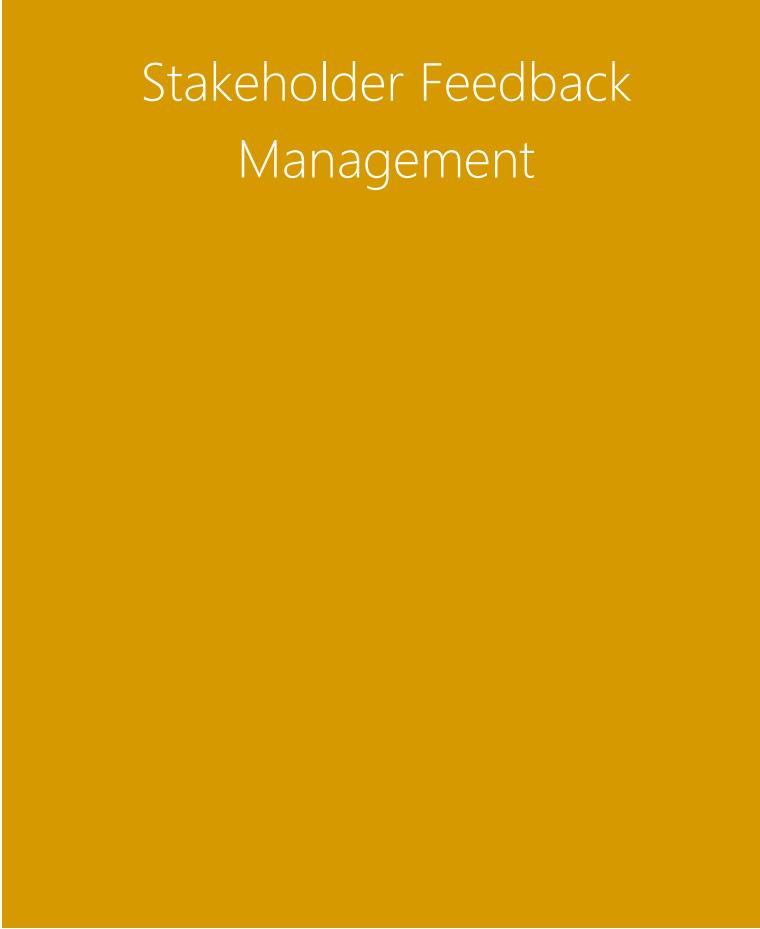
Monitor + Learn

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

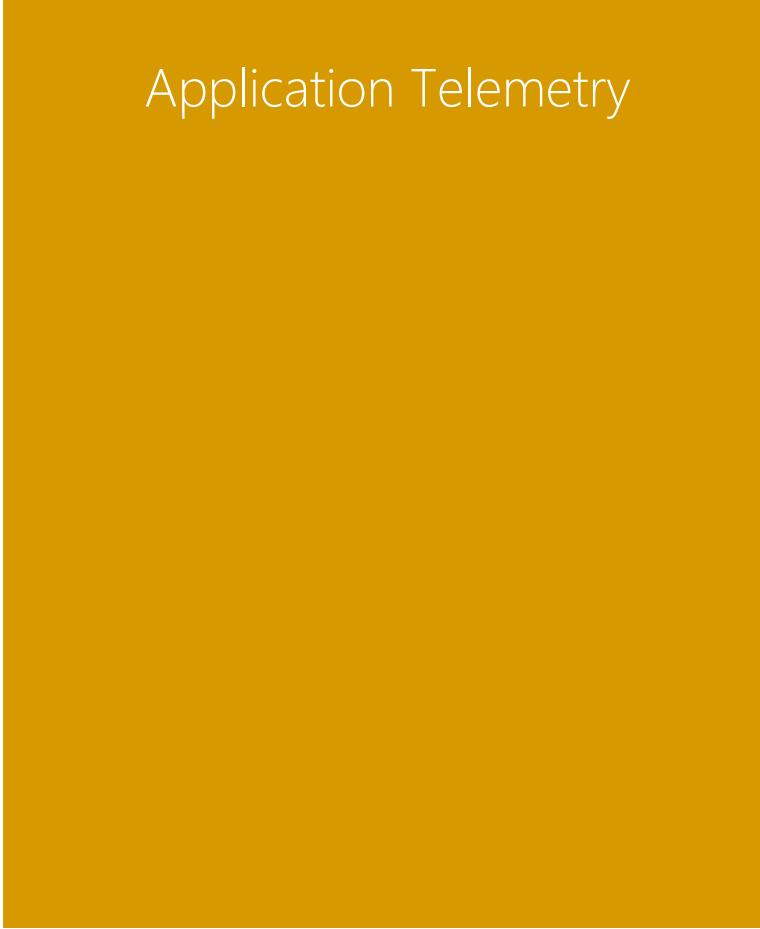


Monitor + Learn

Streamline and automate the workflow between development and IT ops and deliver higher quality software more frequently with less risk.



Stakeholder Feedback
Management



Application Telemetry



Performance and
Load Testing

Feedback management

Actionable feedback

Once you have working software, you're ready to get feedback from your stakeholders. You can ask reviewers to provide videos, screenshots, written comments and ratings. Their feedback is captured into work items that you can review and use to create a bug or suggest a new backlog item.

The screenshot shows the Microsoft Feedback Client interface within Visual Studio Online. It displays two main sections:

- Select Stakeholders:** A text input field contains the email address "bkeller@microsoft.com". Below it is a dropdown menu labeled "Display Name or Windows Live ID" with a "Browse" and "Check name" button.
- Tell Stakeholders How to Access the Application:** This section includes a radio button for "Web Application" (selected), "Remote Machine", and "Client Application". It also shows the URL "http://fabrikamfiber.com" and instructions to "Please launch your browser and go to the URL".

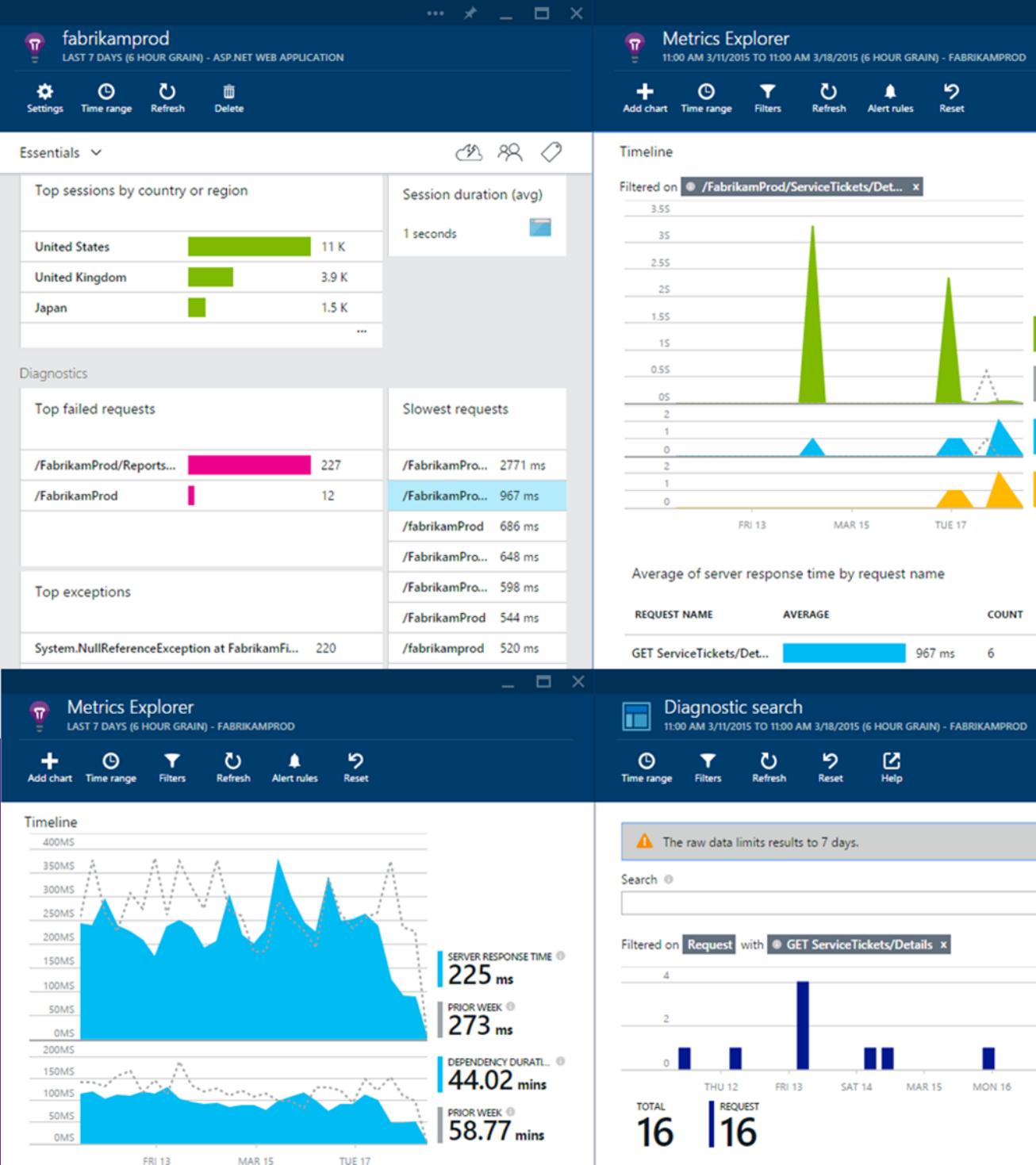
At the bottom, there are buttons for "Start", "Provide" (which is highlighted with an orange oval), and "Submit". The "ITEM: 1" is listed under "Customers page changes". Below this, there are options for "Screen with Voice", "Voice only", "Screenshot", and "Attach file". The status bar at the bottom indicates "Not recording" and "Stop".

This screenshot shows a feedback submission interface on the FF Intranet Portal. It features a large orange logo and the text "FABR Support". The interface includes a "Provide" button, a "Submit" button, and a text area for feedback. Below the text area are icons for "Record", "Voice only", "Screenshot", and "Attach file". The status bar at the bottom indicates "Not recording" and "Stop".

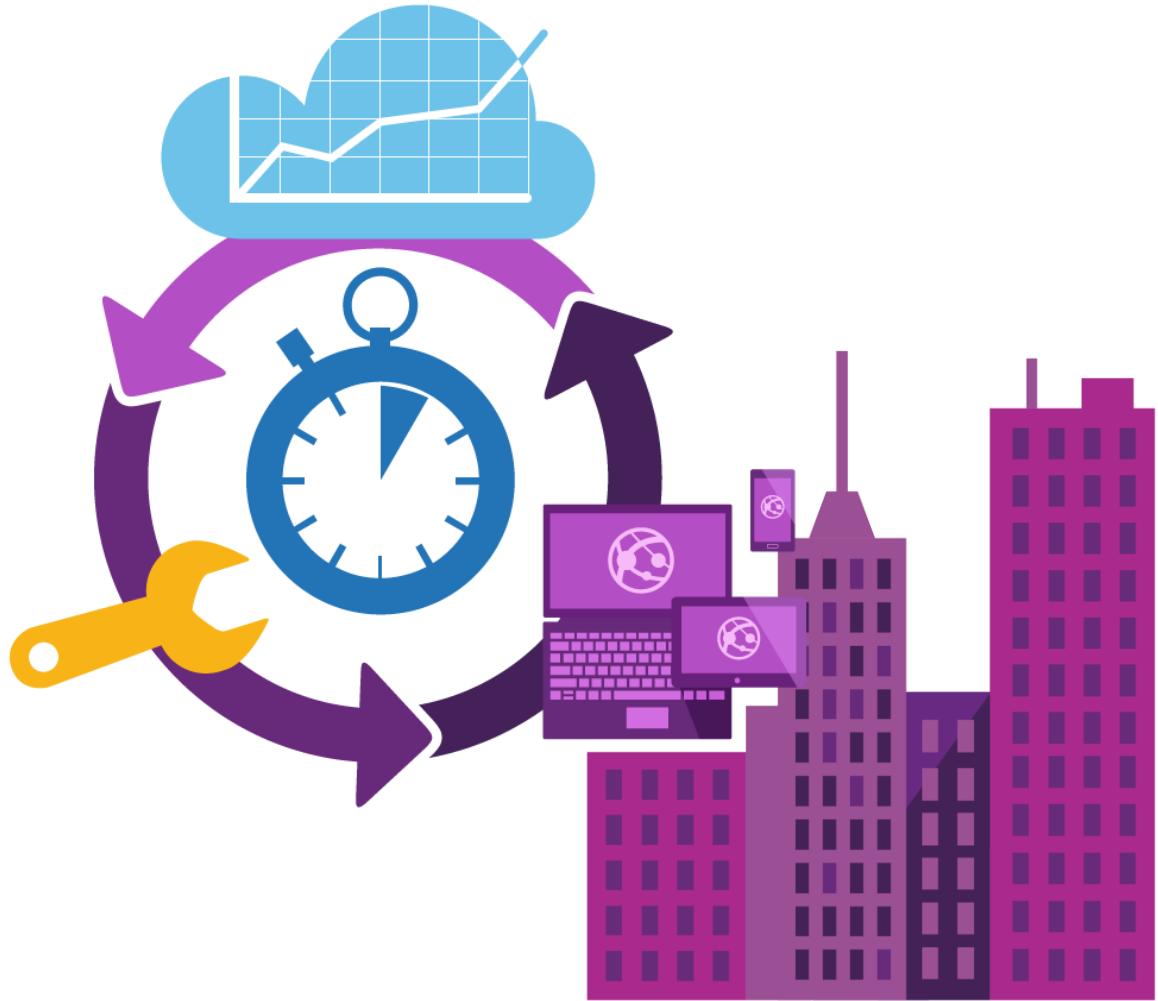
Application Insights

Track usage and performance

Get a 360° view of your app that includes availability, performance and user behavior. Use a powerful and easy-to-use experience to track and improve the success of your application. Analyze data across dimensions and make informed decisions about your product's future.



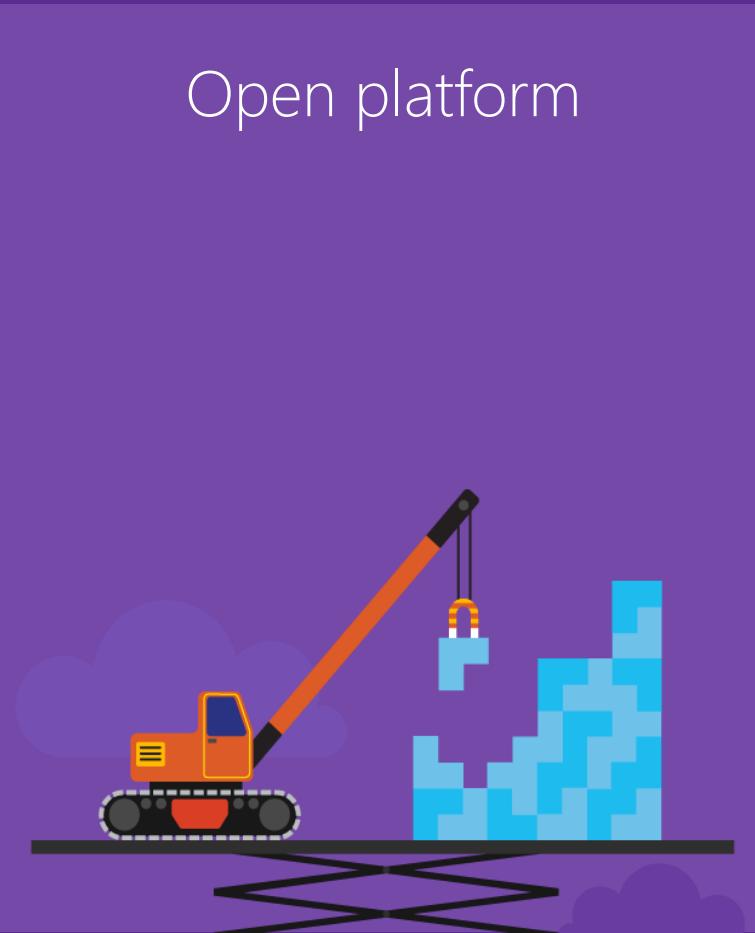
The Microsoft DevOps solution
Open and extensible



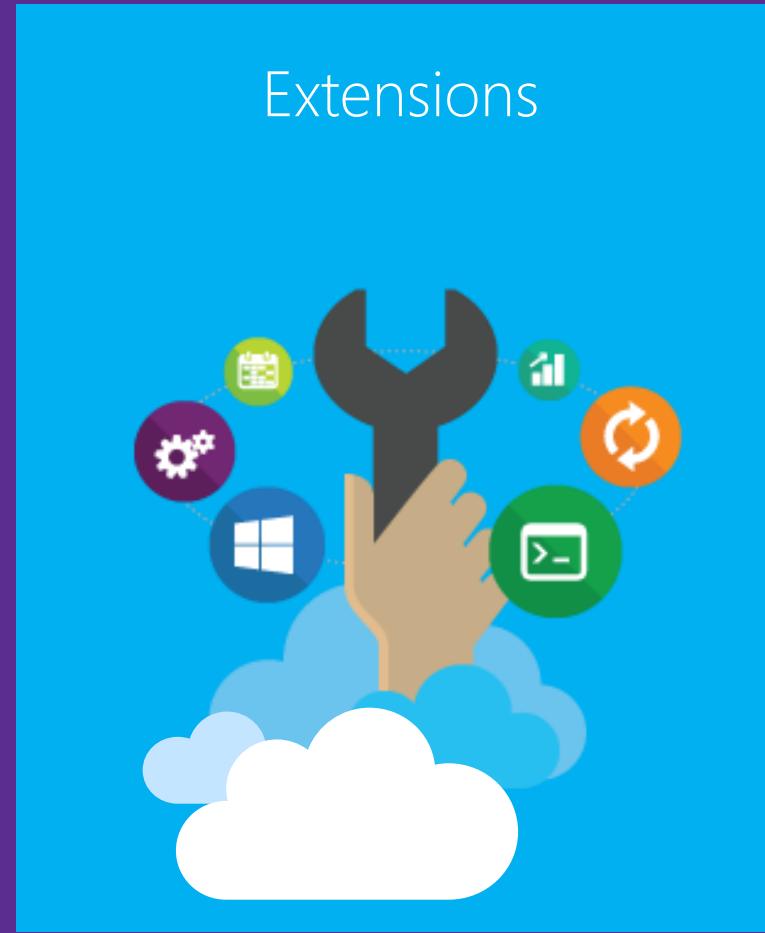
Extend, Customize & Integrate

Customize and extend the Visual Studio platform and create the perfect development environment that you'll love as much as your code.

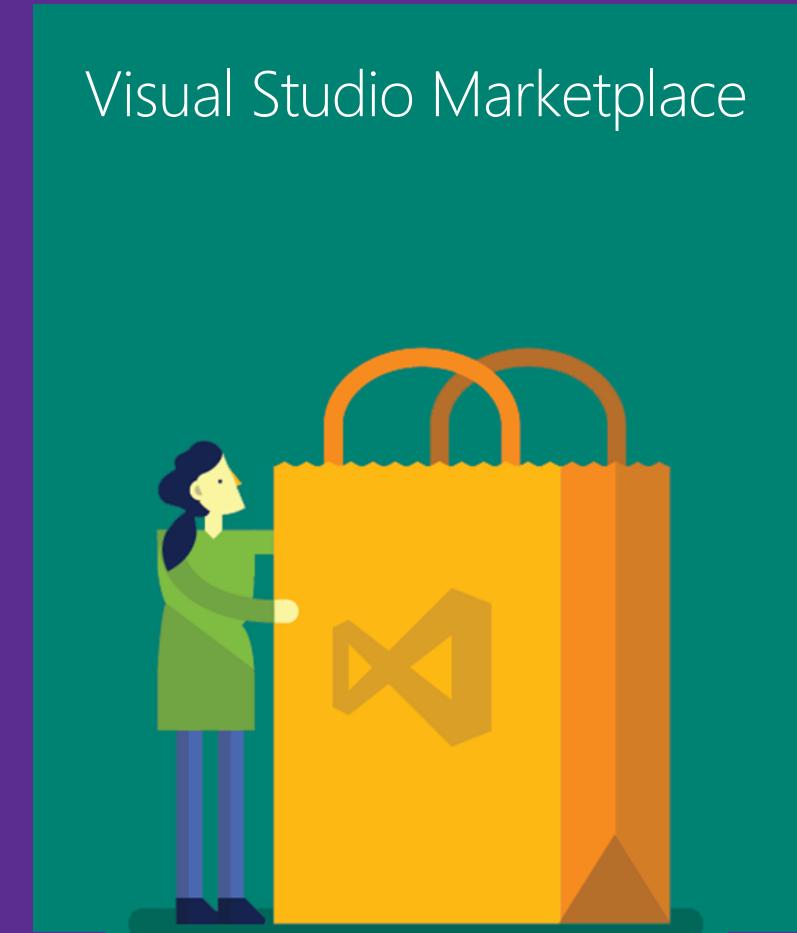
Open platform



Extensions

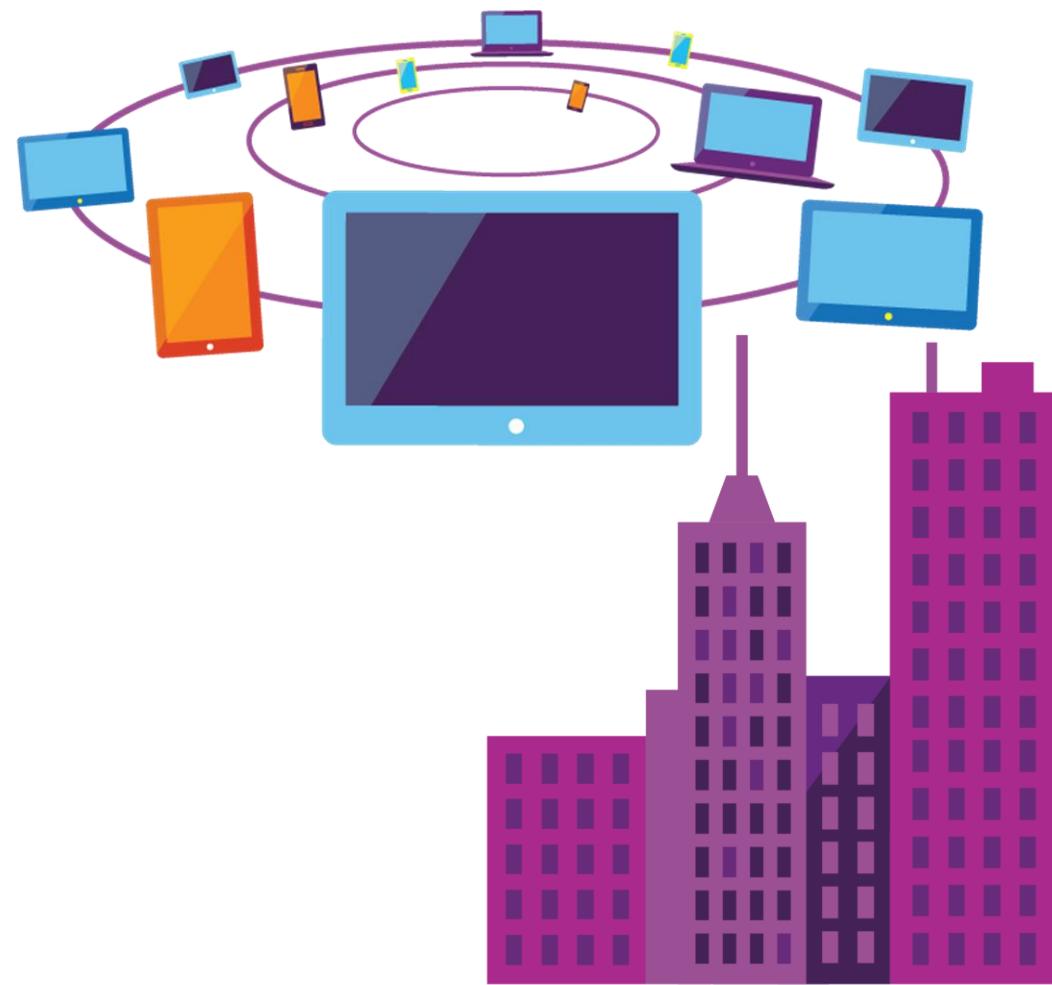


Visual Studio Marketplace

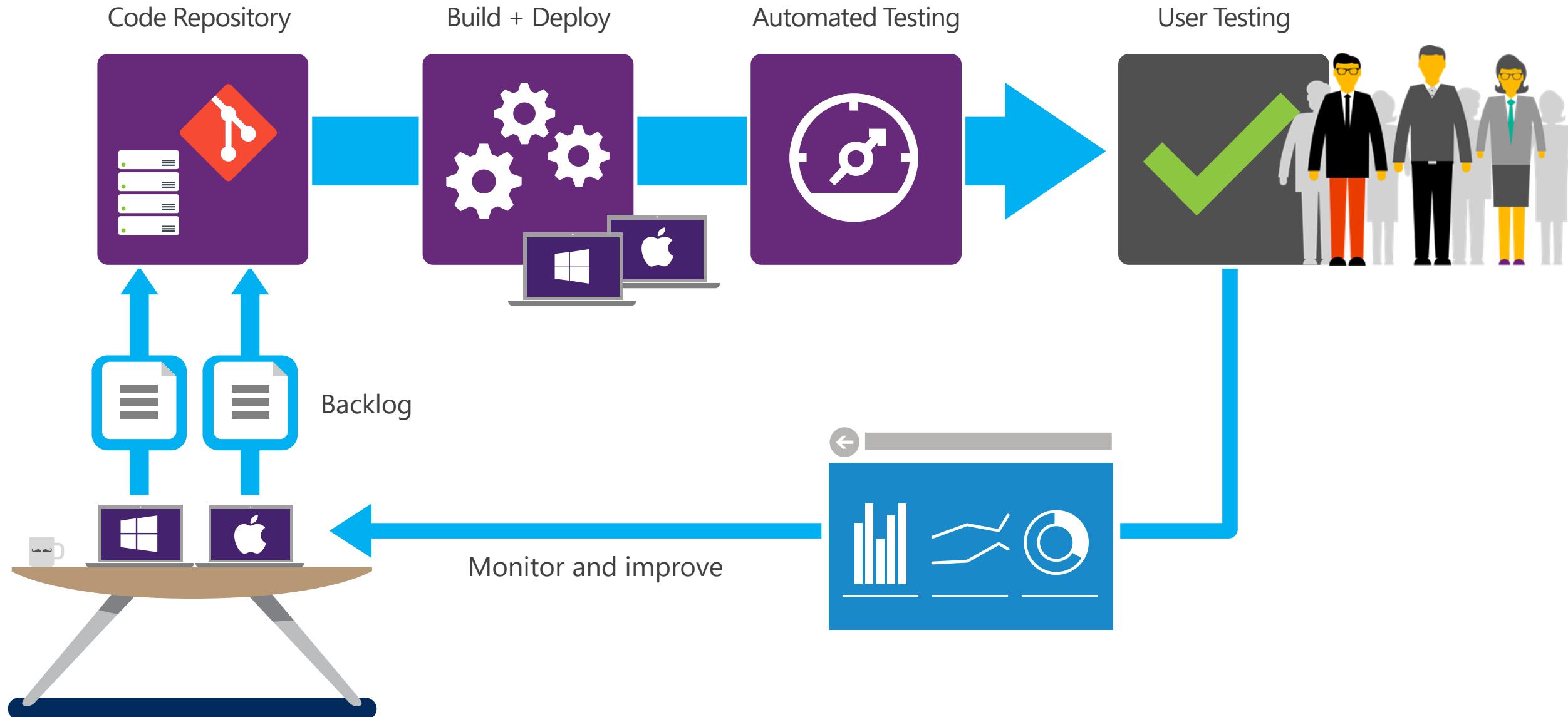


The Microsoft DevOps Solution

DevOps for the digital era



Mobile app CI and CD



Additional Resources

- [Visual Studio Team Services](#)
- [Getting started](#) (link to important chapters in VSTS docs)

Exercise 2:

<http://bit.ly/2kiHl1E>

Importance of testing your projects



Quality Management

A toolset optimized for QA professionals, giving them flexibility in how they work while at the same time keeping them in sync with the rest of the team.

Test planning
and tracking



Manual and
exploratory testing



Performance and
load testing



Test planning and tracking

Single pane of glass

Coordinate all test management activities including test planning, authoring, execution, and tracking from a central location. The test hub in VSTS and TFS gives product owners and business analysts critical insight into progress against the defined acceptance criteria and quality metrics.



Manual and exploratory testing

Browser-based testing

Improved browser-based manual testing tools helps testers author, manage and execute test cases in any modern web browser.

Exploratory testing

Perform ad-hoc and exploratory testing on multiple devices using our integrated 3rd party services without predefined test cases or test steps. The exploratory testing tool includes action recording, screenshots with annotations, video capture and bug-reporting capabilities. Easily create test cases from exploratory test recordings and add them to test plans or suites for subsequent test runs.



Performance and load testing

Unlimited load testing

VSTS and TFS make integrating load testing into your development process easy and will help you avoid nasty surprises in production.

Use cloud-based load testing capabilities to validate and assess application performance under real-world conditions before you deploy. Generate hundreds of thousands of connections in minutes and run tests from one of many global Azure datacenter locations to minimize latency and simulate users' real-world conditions.



Test Execution Type

Guiding Questions

- What is manual software testing?
- What is a benefit of automated testing?
- How is black box testing different from white box testing?

Manual vs. Automated Testing

- Manual and automated test cases are complementary; both types of tests are important for ensuring high quality software.
- Automation is quick and can test many subtle variations in data; it can also easily repeat tests as software evolves. And because it is executed by a computer, the fatigue and mistakes that sometimes accompany repetitive tasks is negated.
- Although manual testing typically takes longer to execute (since it's conducted by a person), it often requires far less set-up time. It is a good choice for tests that only need to be run occasionally, or cases when cost/time of automation setup would outweigh the benefits.

Black Box Testing

- Black box testing is testing conducted without knowledge of the internal workings of the system that is being tested.
- This type of testing simulates end-user experience.
- In general, the tester does not know how the code works - she or he is providing input and examining the output. This person does not necessarily need to know how to program.
- Example scenarios for black box testing include:
 - ✓ Testing that the user interface meets all requirements and is functional.
 - ✓ Testing for a variety of input types (including input outside the expected range, such as entering a negative number for a weight).
 - ✓ Load or stress testing a system.
 - ✓ Testing the security of a project or system.

White Box Testing

- Also know as glass box, clear box, and open box testing.
- White box testing is conducted by examining code for potential failure scenarios.
- White box test cases are created by someone who analyzes the code of the application block and prepares test cases to ensure that the class is behaving in accordance with the specifications.
- White box scenarios include:
 - ✓ Testing internal subroutines that are used “behind the scenes.”
 - ✓ Testing loops and conditional statements for accuracy.
 - ✓ Performance testing of a code path or algorithm

Testing Level

Testing Level

- Unit Test
- Component and Integration Test

Testing Type

Regression Testing

- Whenever any changes are made to a project, it is possible that existing code may no longer work properly, or that previously undiscovered bugs will present themselves. This kind of bug is called a regression.
- To catch these defects, the entire project must be regression tested: a complete retesting of a modified program, rather than a test of only the modified units, to ensure that no errors have been introduced with the modifications.

Stress Testing

- Testing on a small scale, such as a single user running a web application or a database with only a handful of records, may not reveal problems that may occur when the application is used in “real world” conditions.
- Stress testing pushes a system’s functional limits. It is performed by subjecting the system to extreme conditions, such as peak volumes of data or a large number of simultaneous users.
 - ✓ These tests are also referred to as load tests, since they test a system under heavy loads.
- Test automation allows rigorous stress testing without a minimal amount of manual labor.

Performance Testing

- Performance testing determines responsiveness, throughput, reliability, and/or scalability of a system under a given workload.
- In web applications, performance testing is often closely related to stress testing, measuring lag and responsiveness under a heavy load.
- In other applications (desktop and mobile apps, for example), performance testing measures speed and resource utilization, such as disk space and memory.

Security Testing

- Security testing validates an application's security services and identifies potential security flaws.
- Many projects use a black box approach to security testing, allowing security experts with no knowledge of the software to probe the application for holes and weaknesses.

Usability Testing

- Usability testing evaluates a project by studying how real users actually use the software.
- Examples include:
 - ✓ Measuring how long it takes a user to complete a task.
 - ✓ Tracking how many “clicks” or user actions it takes to complete a task or access a feature.

Accessibility Testing

- Accessibility testing validates an application's support for users with disabilities.
- Accessibility testing may include:
 - ✓ **Compliance:** Does it comply with legal requirements regarding accessibility?
 - ✓ **Effectiveness:** Can users with disabilities use the application?
 - ✓ **Usefulness:** Does the application expose adequate functionality for users with disabilities?
 - ✓ **Satisfaction:** How is the application perceived by users with disabilities?
- Accessibility testing may include usability tests with disabled users and assistive technology devices.

Additional Resources

MSDN Software Testing Resources

Testing Process for Application Blocks	http://msdn.microsoft.com/en-us/library/ff649517.aspx
Black Box & White Box Testing for Application Blocks	http://msdn.microsoft.com/en-us/library/ff649503.aspx
Patterns and Practices: Manual System Tests	http://msdn.microsoft.com/en-us/library/jj159334.aspx
Test Early and Often	http://msdn.microsoft.com/en-us/library/vstudio/ee330950.aspx
Unit Testing	http://msdn.microsoft.com/en-us/library/aa292197.aspx
Unit Testing: Testing the Inside	http://msdn.microsoft.com/en-us/library/jj159340.aspx
Integration Testing	http://msdn.microsoft.com/en-us/library/aa292128.aspx
Unit testing, component level testing and UI testing, what to use and when	http://blogs.msdn.com/b/raulperez/archive/2010/04/29/unit-testing-component-level-testing-and-ui-testing-what-to-use-and-when.aspx
Regression Testing	http://blogs.msdn.com/b/narendra_parihar_s_blog/archive/2012/04/20/regression-testing.aspx
Performance and Stress Testing	http://msdn.microsoft.com/library/dd293540.aspx
Security Testing	http://msdn.microsoft.com/en-us/library/aa292190.aspx
Localization Testing	http://msdn.microsoft.com/en-us/library/aa292138.aspx
Testing a User Interface (UI)	http://msdn.microsoft.com/en-us/library/windows/desktop/ff728827.aspx

Exercise 3:

<http://bit.ly/2kw46mo>

Additional Resources

- [Testing Tools and Services](#)
- [Isolating Code Under Test with Microsoft Fakes](#)
- [Azure DevTest Labs](#)

Automated Testing using Xamarin Test Cloud



Testing Xamarin apps

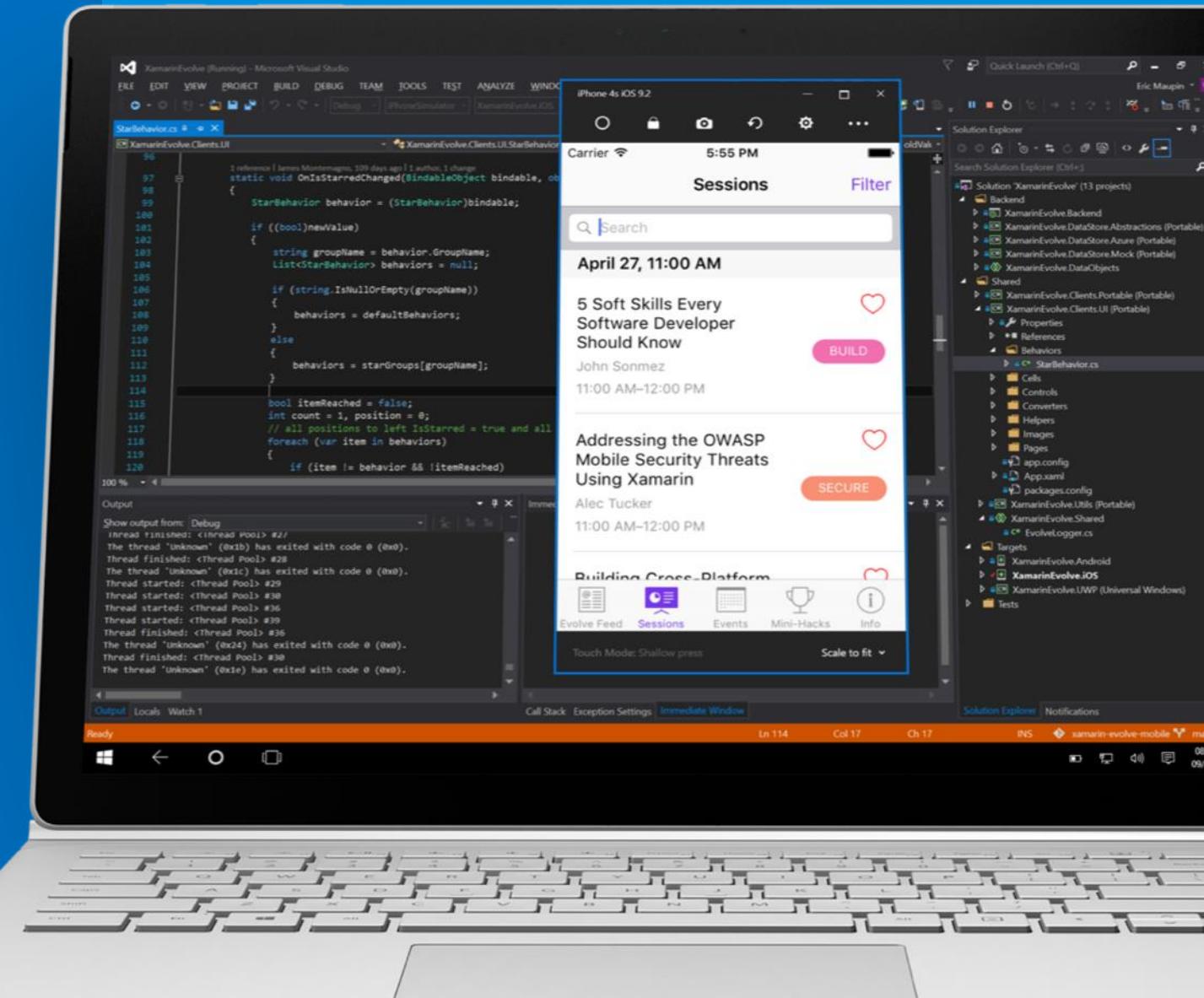
Unit testing

Developers can use their preferred unit testing frameworks and tools for Xamarin projects

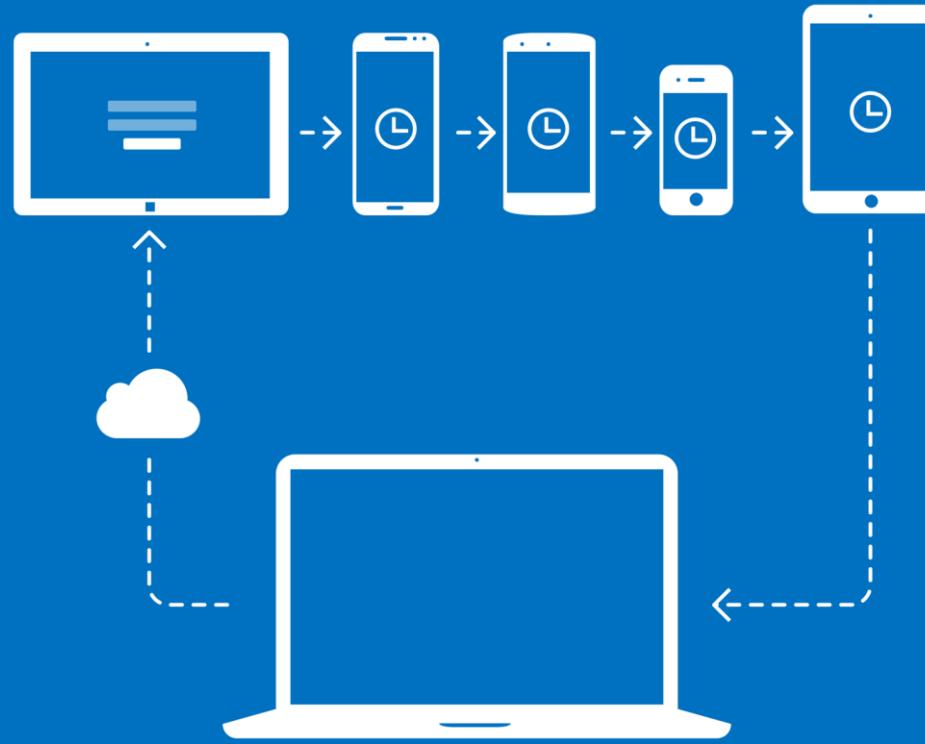
iOS simulator remoting

Test your app by remoting into the iOS simulator running on a physical Mac from Visual Studio

Features like multi-touch, pressure sensitivity, screen rotating, screenshots, location changes are all supported – and it is super-fast! Never leave Visual Studio again ...

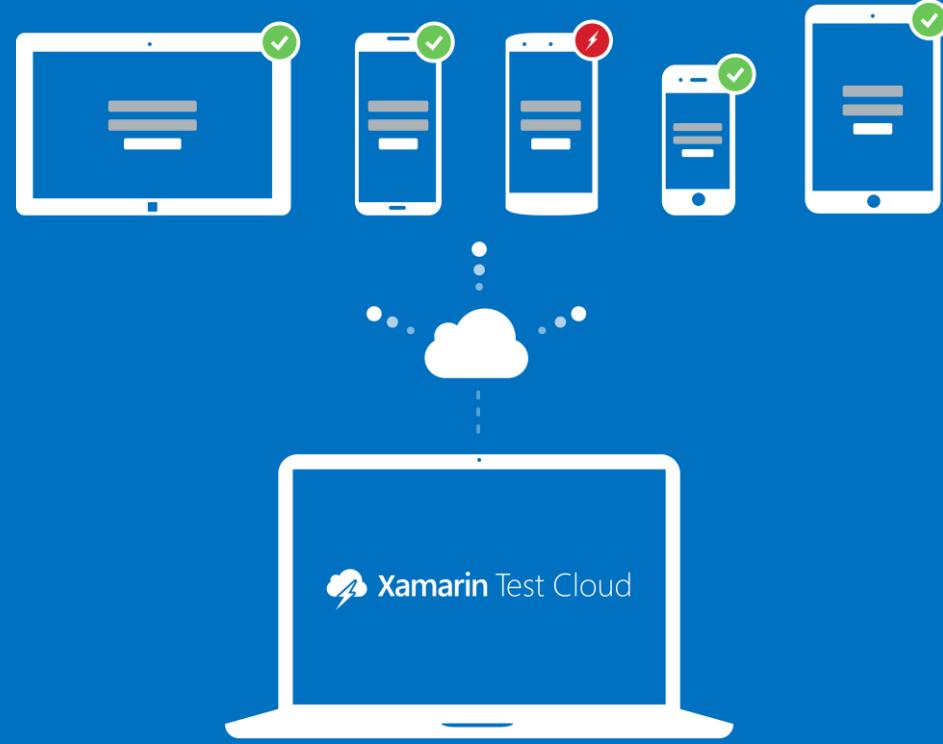


Mobile testing approaches



Device Remoting

Tests are performed one at the time,
which consumes more time and delays
bugs detection



Automated Testing

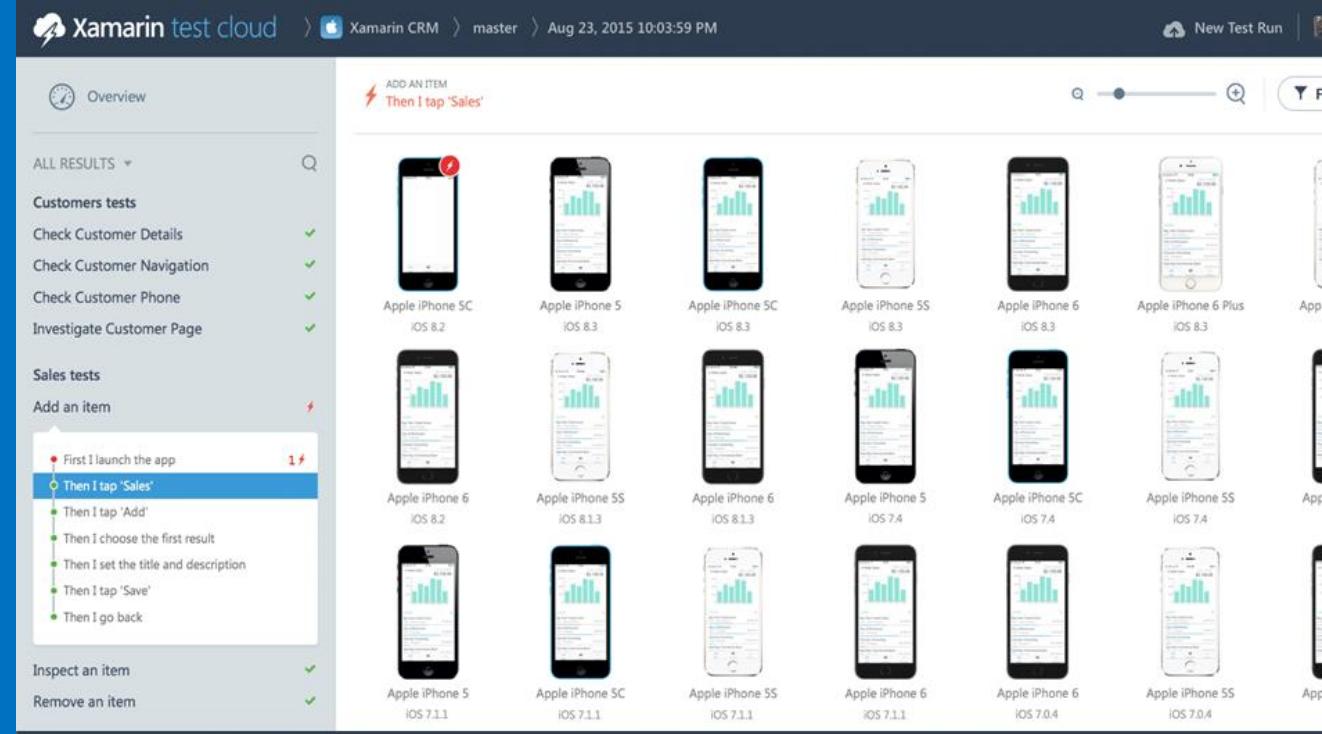
Test on thousands of devices simultaneously,
saving lots of time and detecting
bugs more quickly

Xamarin Test Cloud

for   

Automated mobile app testing

- Diverse library of real devices
- Interact with apps the way users will, including complex gestures
- Test continuously
- Automate UI testing
- Analyze app performance
- Use the tools you know
- Test native device functionality



The screenshot shows the 'Overview' page of the Xamarin Test Cloud. On the left, a sidebar lists various test categories with green checkmarks: Customers tests (Check Customer Details, Check Customer Navigation, Check Customer Phone, Investigate Customer Page), Sales tests (Add an item, Then I tap 'Sales'), and Order tests (Add New Order, Add New Order And Deliver, Deliver Order, Edit Customer Order). Below these are Inspect an item and Remove an item options. A modal window is open over the sidebar, titled 'Then I tap "Sales"', showing a step-by-step sequence of interactions: 'First I launch the app' (green dot), 'Then I tap "Sales"' (blue dot, currently selected), 'Then I tap "Add"' (green dot), 'Then I choose the first result' (green dot), 'Then I set the title and description' (green dot), 'Then I tap "Save"' (green dot), and 'Then I go back' (green dot). At the top right of the main area, there's a red error icon with the text 'ADD AN ITEM Then I tap "Sales"'. To the right of the sidebar, a grid of 20 smartphone icons represents different devices used for testing. Each icon includes the device name (e.g., Apple iPhone 5C, Apple iPhone 5S, etc.) and its iOS version (e.g., iOS 8.2, iOS 8.3, etc.).

The bottom half of the screenshot shows a summary dashboard with the following details:

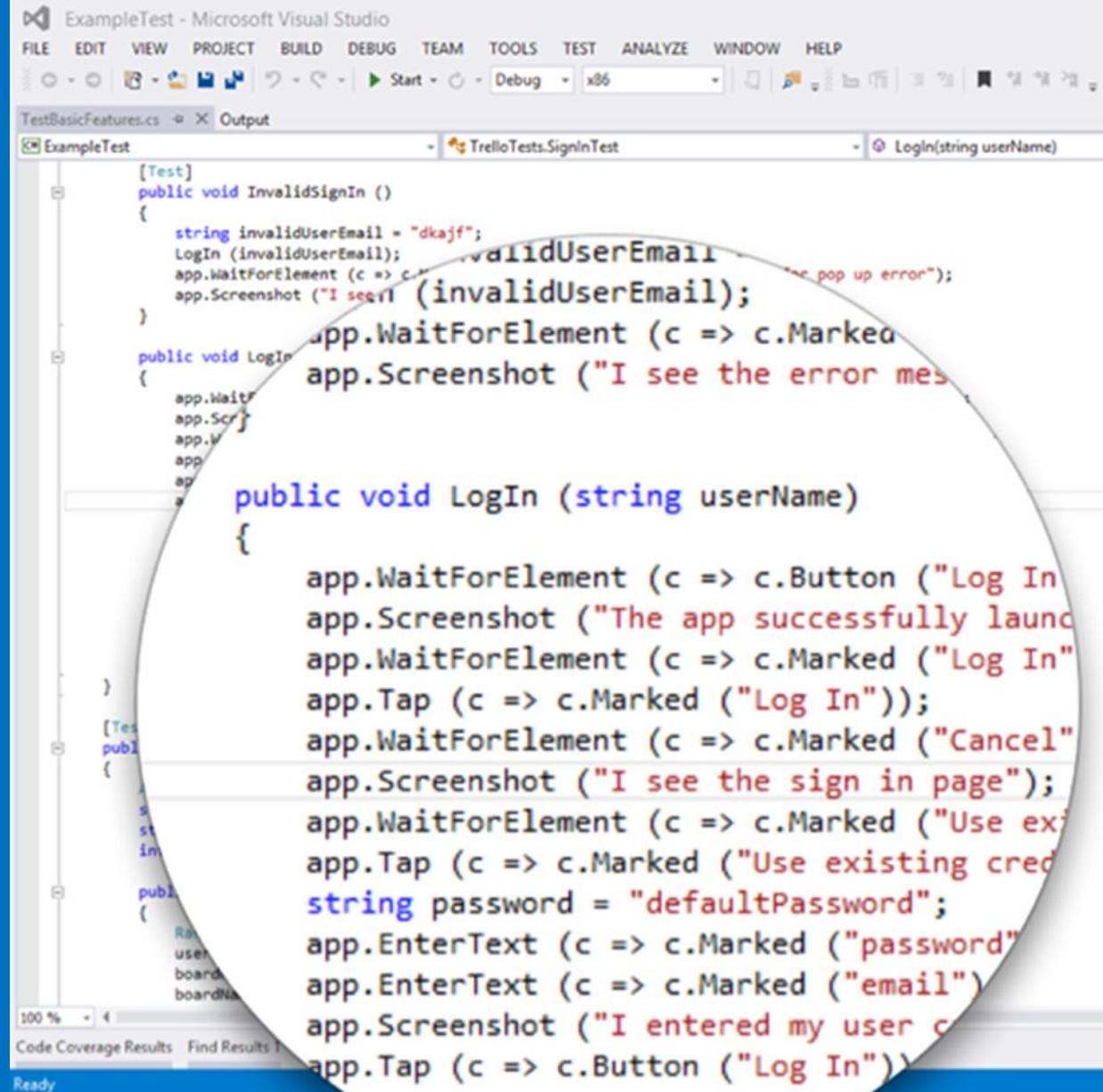
- Date: Aug 23, 2015 10:03:59 PM
- App Version: Xamarin CRM Version 2.0 (1.0)
- Test Status: 4/15 tests failed (represented by a green circle with a red segment)
- Device Status: 4/10 devices failed (represented by a red circle with a green segment)
- Performance Metrics:
 - Total Device Time: 266 min
 - Devices: 10
 - OS Versions: 3
 - Peak Memory: 159.70 MB
 - App Size: 17.78
- Failure Categories: Failures by category (By OS version, By form factor, By manufacturer)

Xamarin.UITest

UI testing framework

Xamarin.UITest is a framework that ties in directly to the Nunit testing framework to write UI tests. You can run UI tests directly against a simulator for regression testing of your applications.

- Create automated user interface tests in C#
- Upload to Xamarin Test Cloud or run against a physical device or emulator
- Run your UI tests directly from Visual Studio or Xamarin Studio
- Works for **any** app: C#, Java, Objective-C/Swift, or hybrid



The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Title Bar:** ExampleTest - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
- Toolbars:** Standard, Debug, Start, Build, Analyze, Window, Help
- Status Bar:** Ready
- Code Editor:** The file `TestBasicFeatures.cs` is open, showing C# code for UI testing. A red oval highlights the `LogIn` method and its implementation.
- Task List:** Shows `TrelloTests.SignInTest` and `Login(string userName)`.
- Output Window:** Shows the output of the test execution.

```
[Test]
public void InvalidSignIn ()
{
    string invalidUserEmail = "dkajf";
    LogIn (invalidUserEmail);
    app.WaitForElement (c => c.Marked ("Log In"));
    app.Screenshot ("I see the invalidUserEmail");
}

public void LogIn (string userName)
{
    app.WaitForElement (c => c.Marked ("Log In"));
    app.Screenshot ("The app successfully launched");
    app.WaitForElement (c => c.Marked ("Log In"));
    app.Tap (c => c.Marked ("Log In"));
    app.WaitForElement (c => c.Marked ("Cancel"));
    app.Screenshot ("I see the sign in page");
    app.WaitForElement (c => c.Marked ("Use existing credentials"));
    app.Tap (c => c.Marked ("Use existing credentials"));
    string password = "defaultPassword";
    app.EnterText (c => c.Marked ("password"));
    app.EnterText (c => c.Marked ("email"));
    app.Screenshot ("I entered my user credentials");
    app.Tap (c => c.Button ("Log In"))
}
```

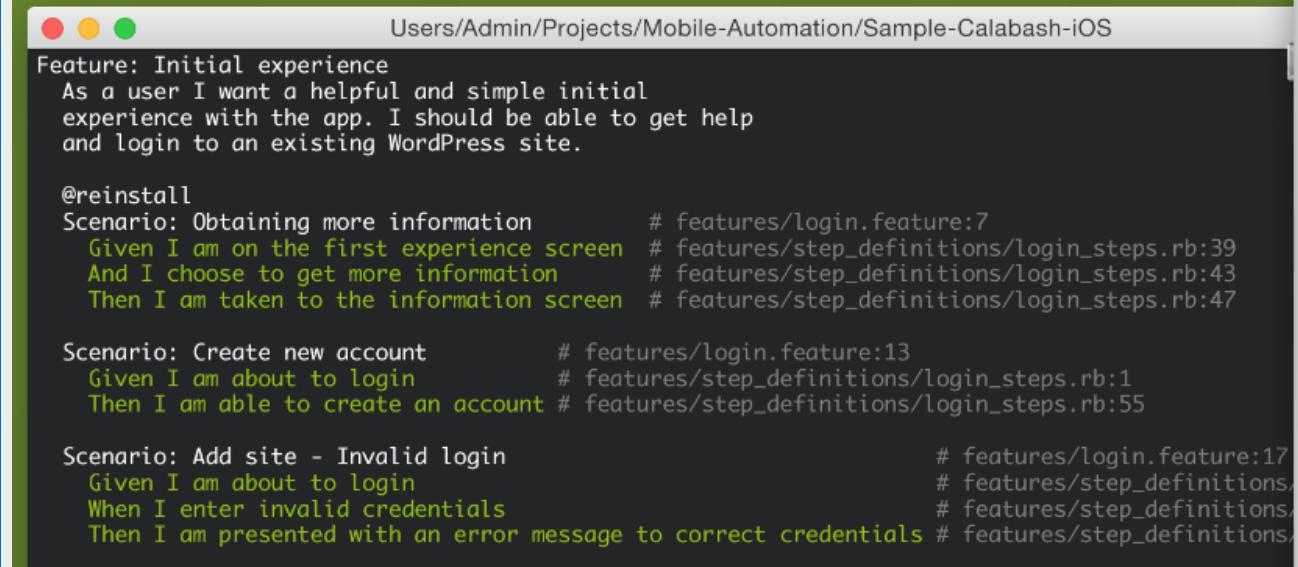
Calabash UI Acceptance Testing framework

Calabash allows you to write and execute tests that validate the functionality of iOS and Android Apps. It uses the concepts of Behavior Driven Development and tests are written with cucumber and ruby.

Calabash

Automated acceptance testing for mobile apps

Calabash enables you to write and execute automated acceptance tests of mobile apps. It's cross-platform, supporting Android and iOS native apps. It's also open source and free, developed and maintained by [Xamarin](#).



The screenshot shows a terminal window with the following content:

```
Users/Admin/Projects/Mobile-Automation/Sample-Calabash-iOS

Feature: Initial experience
As a user I want a helpful and simple initial
experience with the app. I should be able to get help
and login to an existing WordPress site.

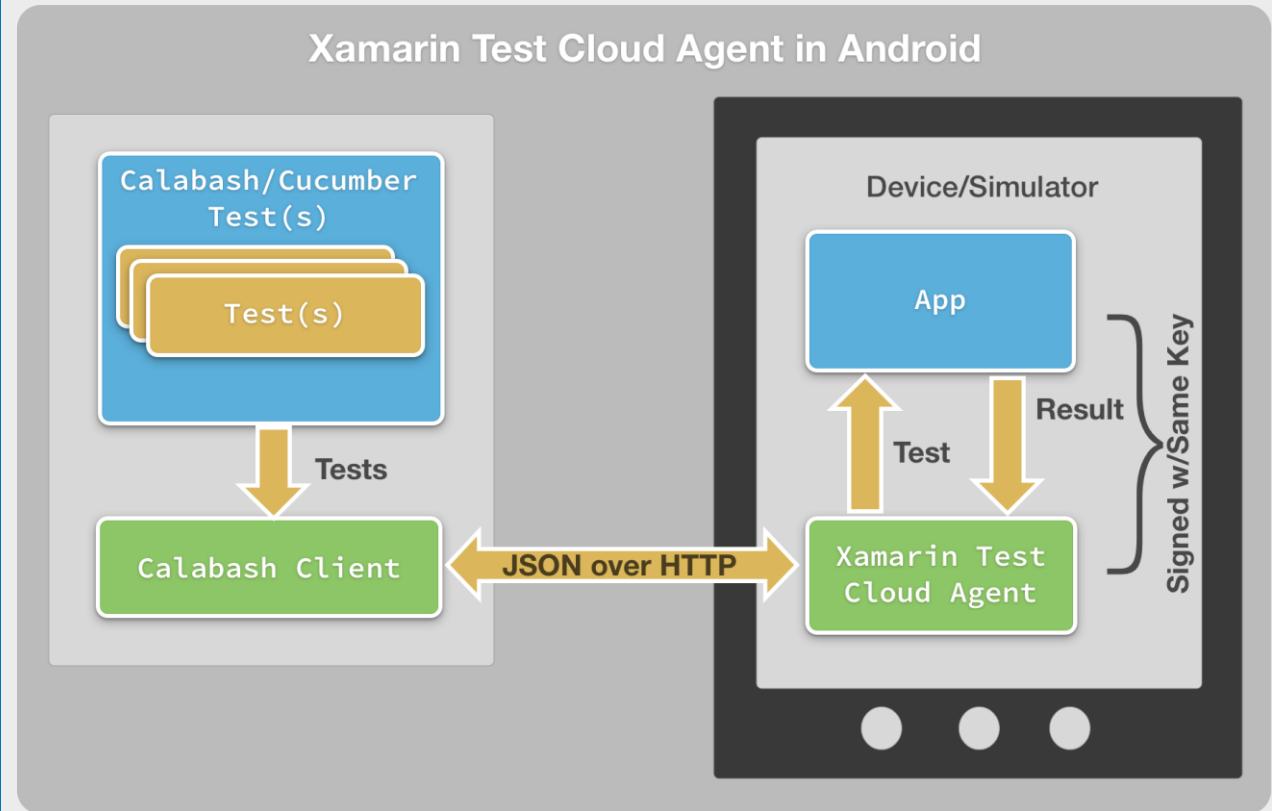
@reinstall
Scenario: Obtaining more information      # features/login.feature:7
  Given I am on the first experience screen # features/step_definitions/login_steps.rb:39
  And I choose to get more information     # features/step_definitions/login_steps.rb:43
  Then I am taken to the information screen # features/step_definitions/login_steps.rb:47

Scenario: Create new account      # features/login.feature:13
  Given I am about to login      # features/step_definitions/login_steps.rb:1
  Then I am able to create an account # features/step_definitions/login_steps.rb:55

Scenario: Add site - Invalid login      # features/login.feature:17
  Given I am about to login      # features/step_definitions/login_steps.rb:1
  When I enter invalid credentials    # features/step_definitions/login_steps.rb:5
  Then I am presented with an error message to correct credentials # features/step_definitions/login_steps.rb:55
```

Xamarin Test Cloud Agent on Android

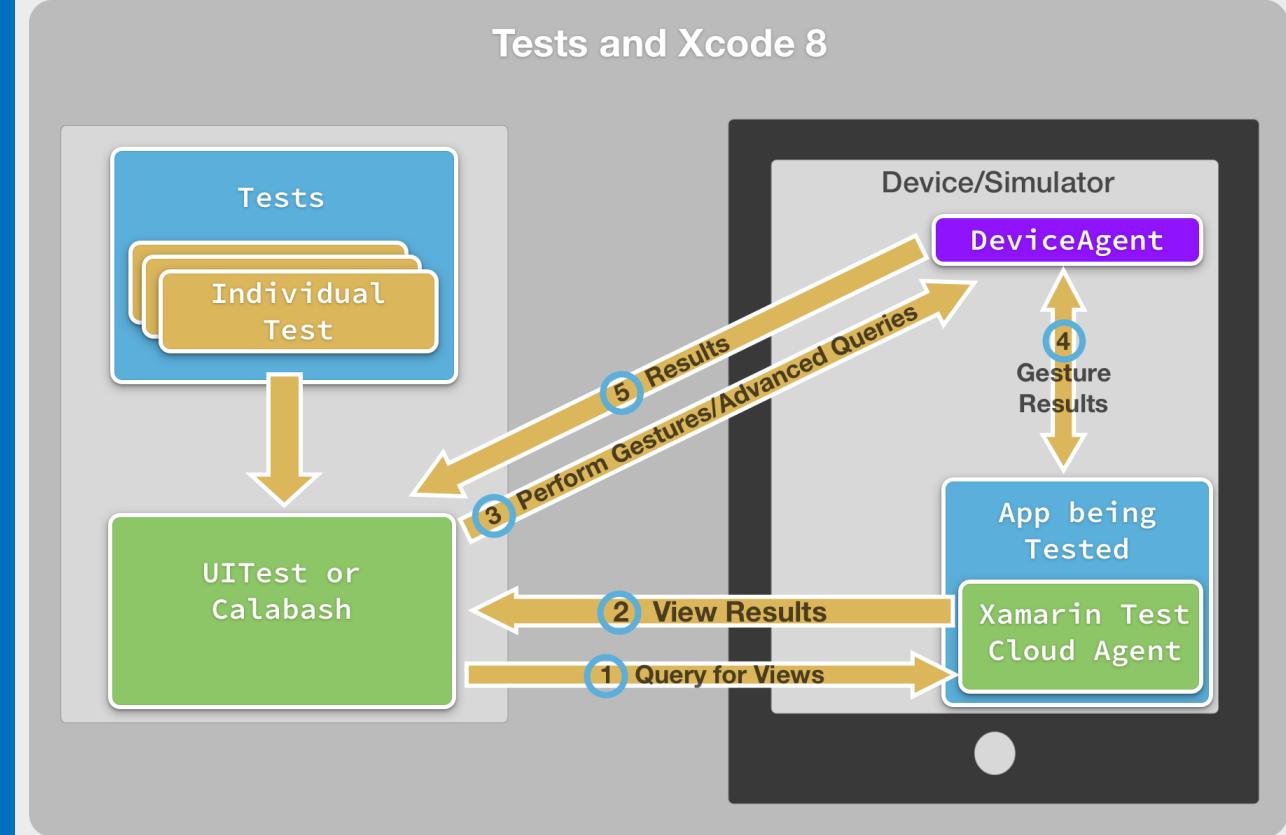
The Xamarin Test Cloud Agent is responsible for using the Android automation APIs to control the user interface and to locate views so that a test may interact with them.



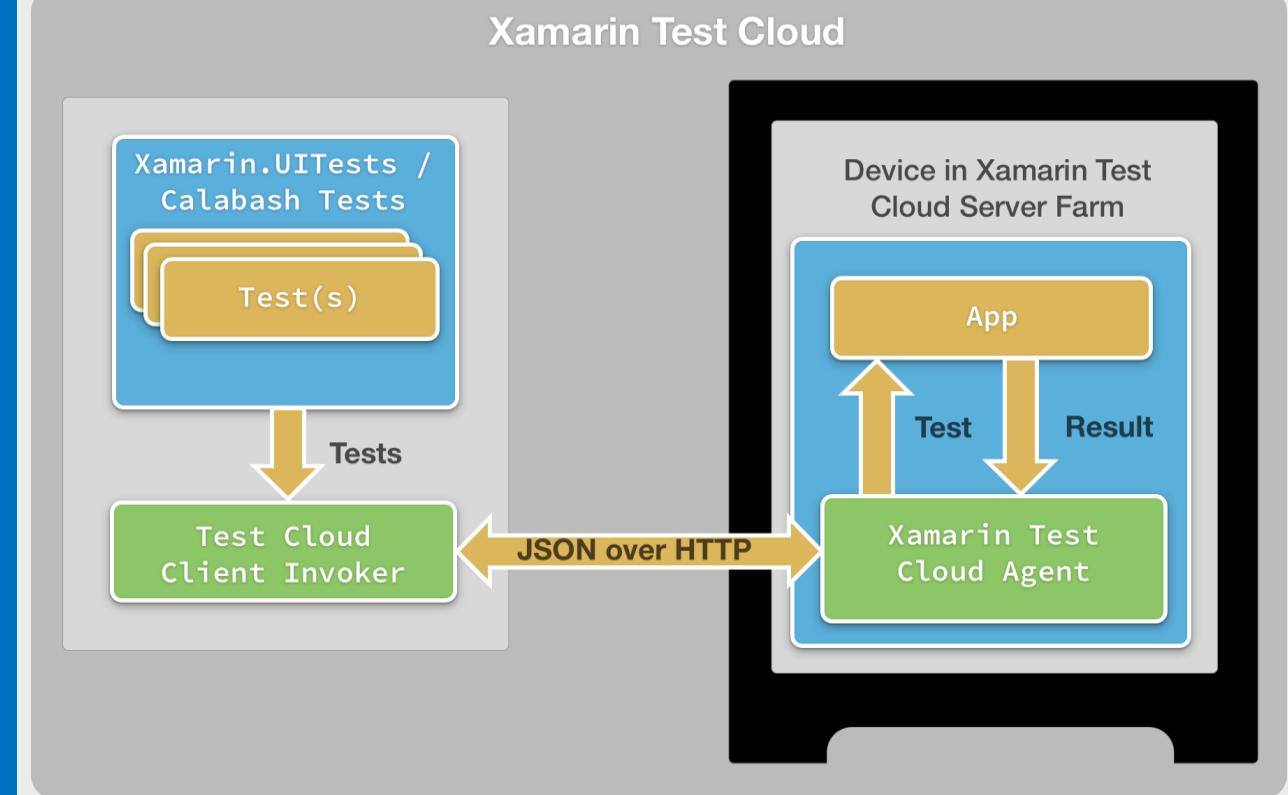
Xamarin Test Cloud Agent on iOS

Test Cloud Agent will interrogate the active window and retrieve information about the views for to the user.

The DeviceAgent will simulate the gestures and actions for the test using the automation API's provided with Xcode 8.

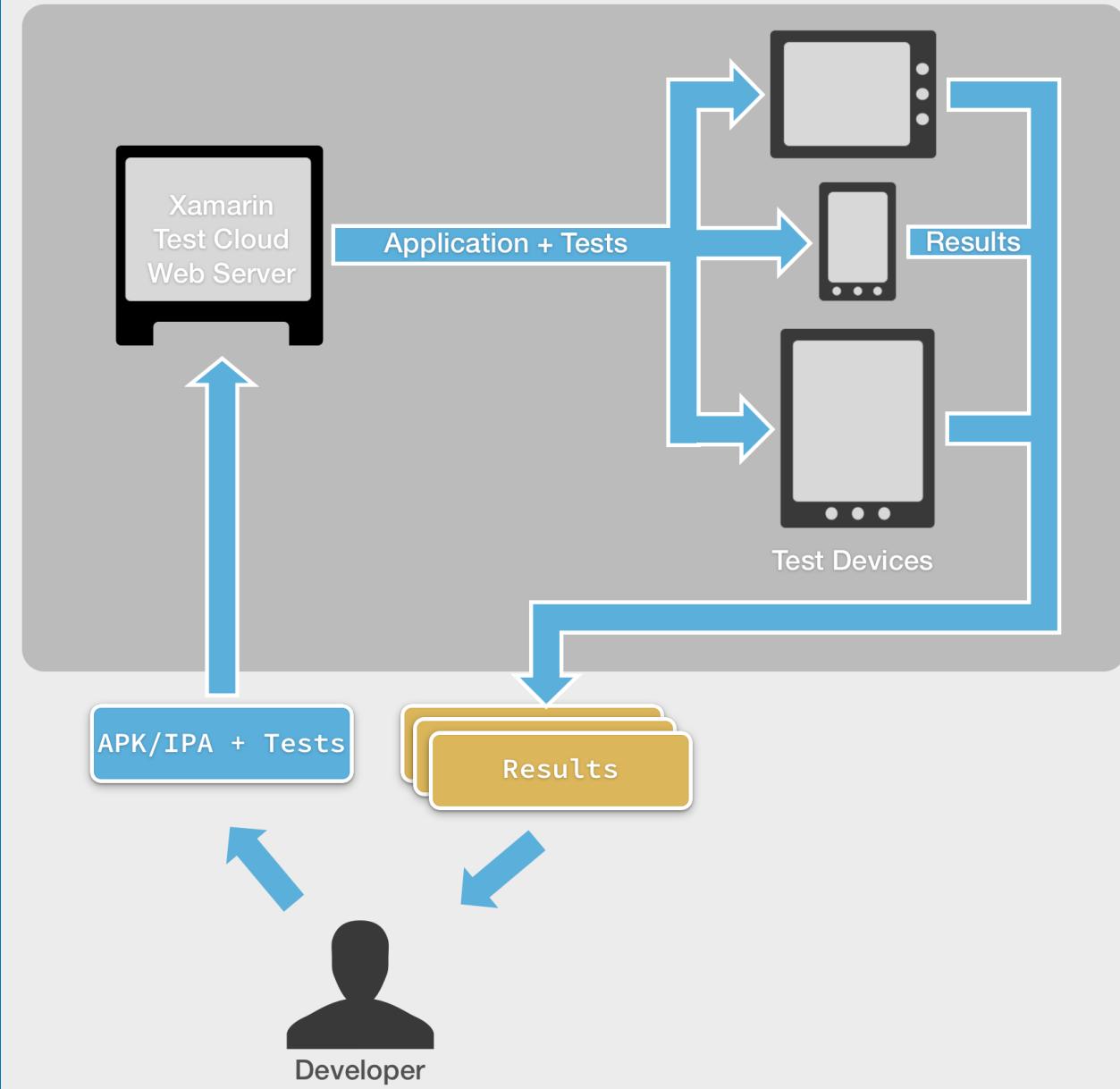


Running tests in Xamarin Test Cloud

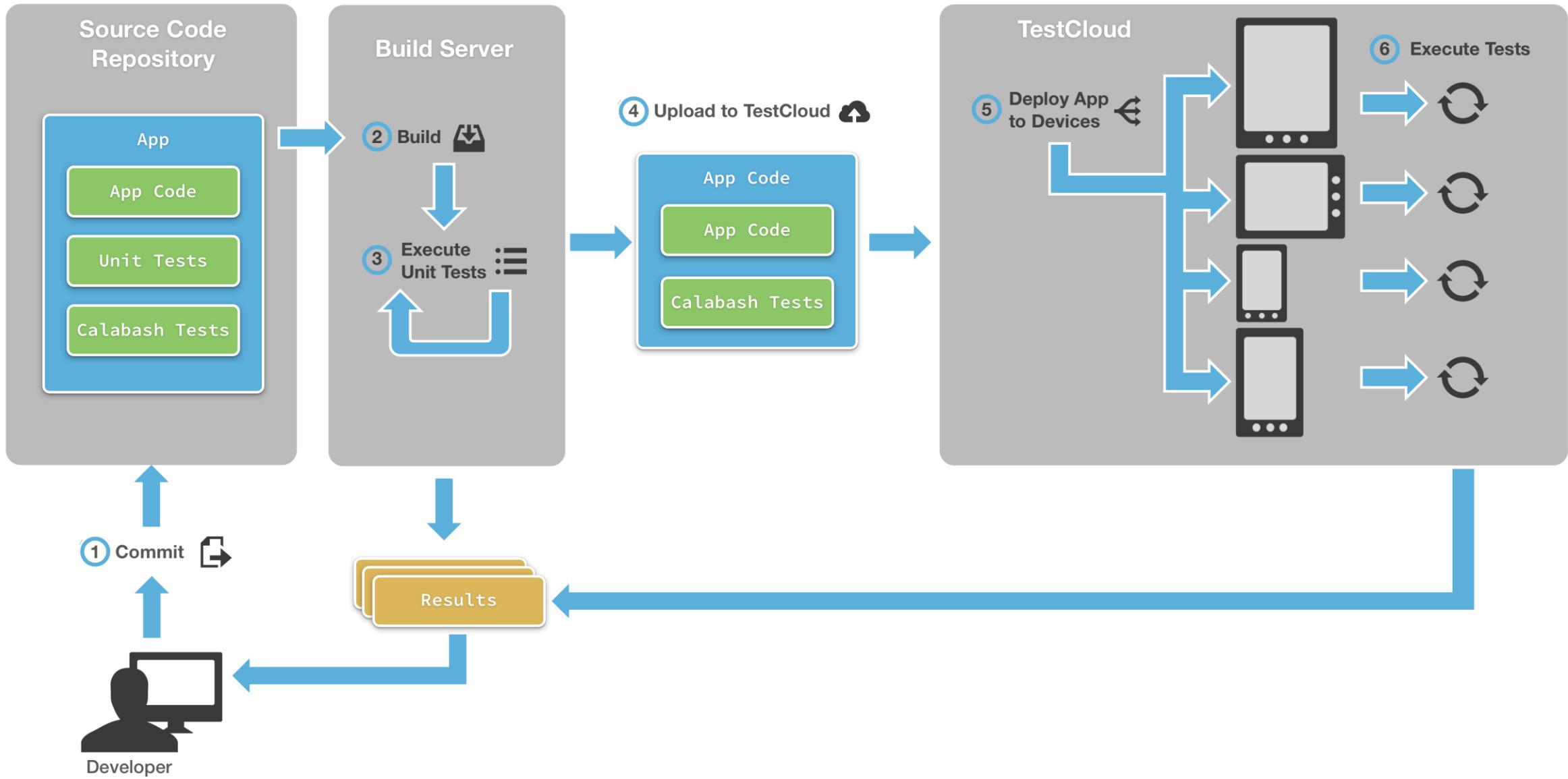


Xamarin Test Cloud flow

- Upload package + tests
- Xamarin Test Cloud distributes, installs and runs on devices
- Back comes results



CI / CD flow with Xamarin Test Cloud



Demo: Xamarin UITest

Xamarin.UITest REPL

```
[FormsTextView] text: "Many many years ago I wrote a blog post about what .NET Developers ought to know. Unfortunately what was just a list of questions was abused by recruiters and others who used it as a harsh litmus test..."  
[ImageCircleRenderer > FormsImageView]  
[View]  
[ConditionalFocusLayout > ... > Platform_DefaultRenderer] (center not on screen)  
[LabelRenderer]  
[FormsTextView] text: "Teaching coding from the Metal Up or from the Glass Back?"  
[LabelRenderer]  
[FormsTextView] text: "Friday, January 6, 2017"  
[Toolbar] id: "toolbar"  
[Textview] text: "Blog"  
[ImageButton] label: "OK"  
[View] id: "navigationBarBackground"  
->>> tap("OK")  
(1,2): error CS0103: The name `tap` does not exist in the current context  
->>> tap  
(1,2): error CS0103: The name `tap` does not exist in the current context  
->>> app.Tap("OK")  
->>> app.Tap(c => c.Marked("Ratchet"))  
Tapping coordinates [ 77, 143 ].  
->>> app.Tap(c => c.Marked("Ratchet"))
```

Hanselman.Forms

>About

Blog

Twitter

Hanselminutes

Rat THE Ratchet

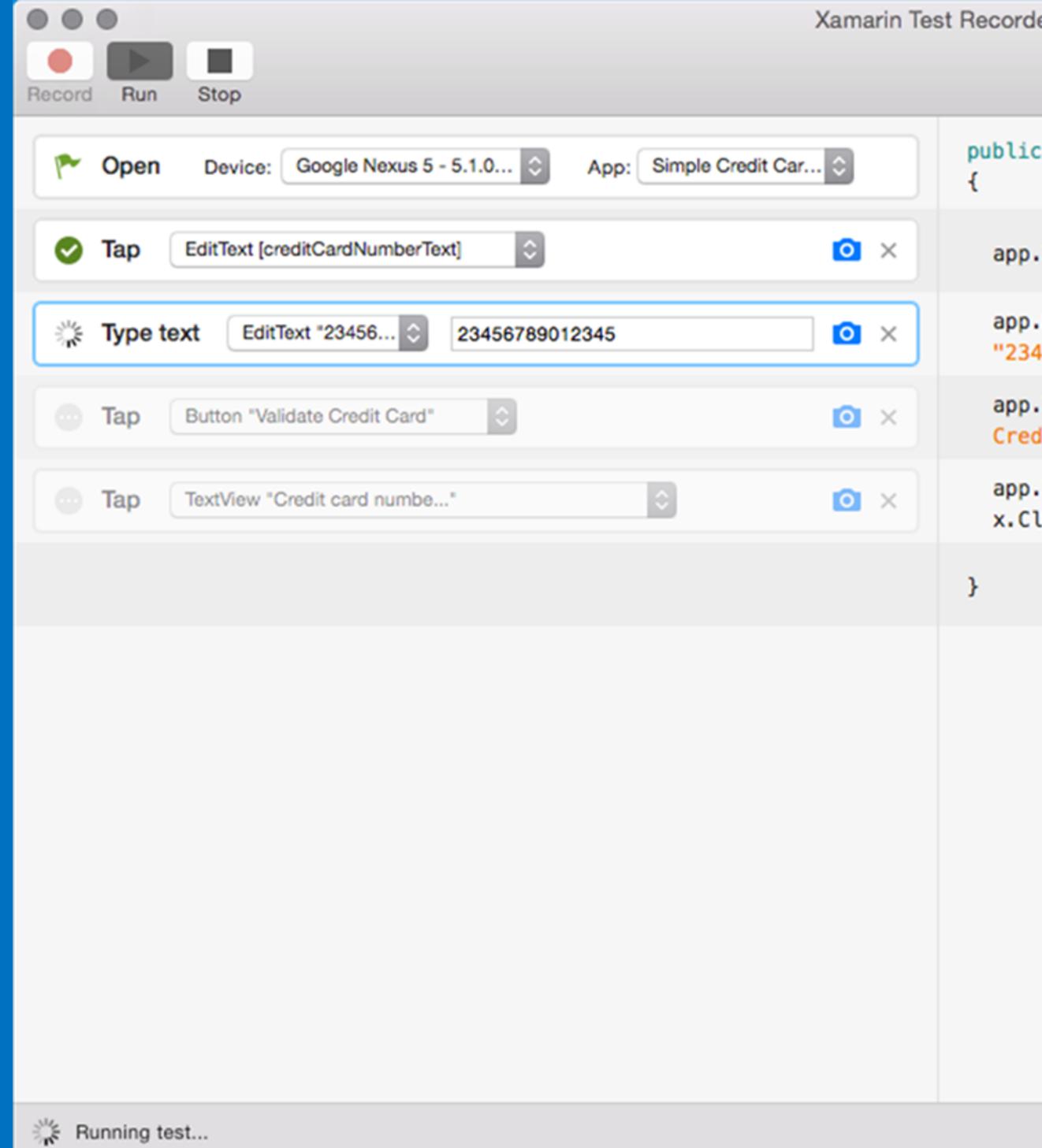
Developers Life

Xamarin Test Recorder (Preview)

Record UI test steps

Standalone application (OS X) or Visual Studio feature that watches and records how the user interacts and creates automated tests in C# based on those interactions.

Mobile tests can easily be exported to Xamarin Test Cloud for further automation of UI testing.

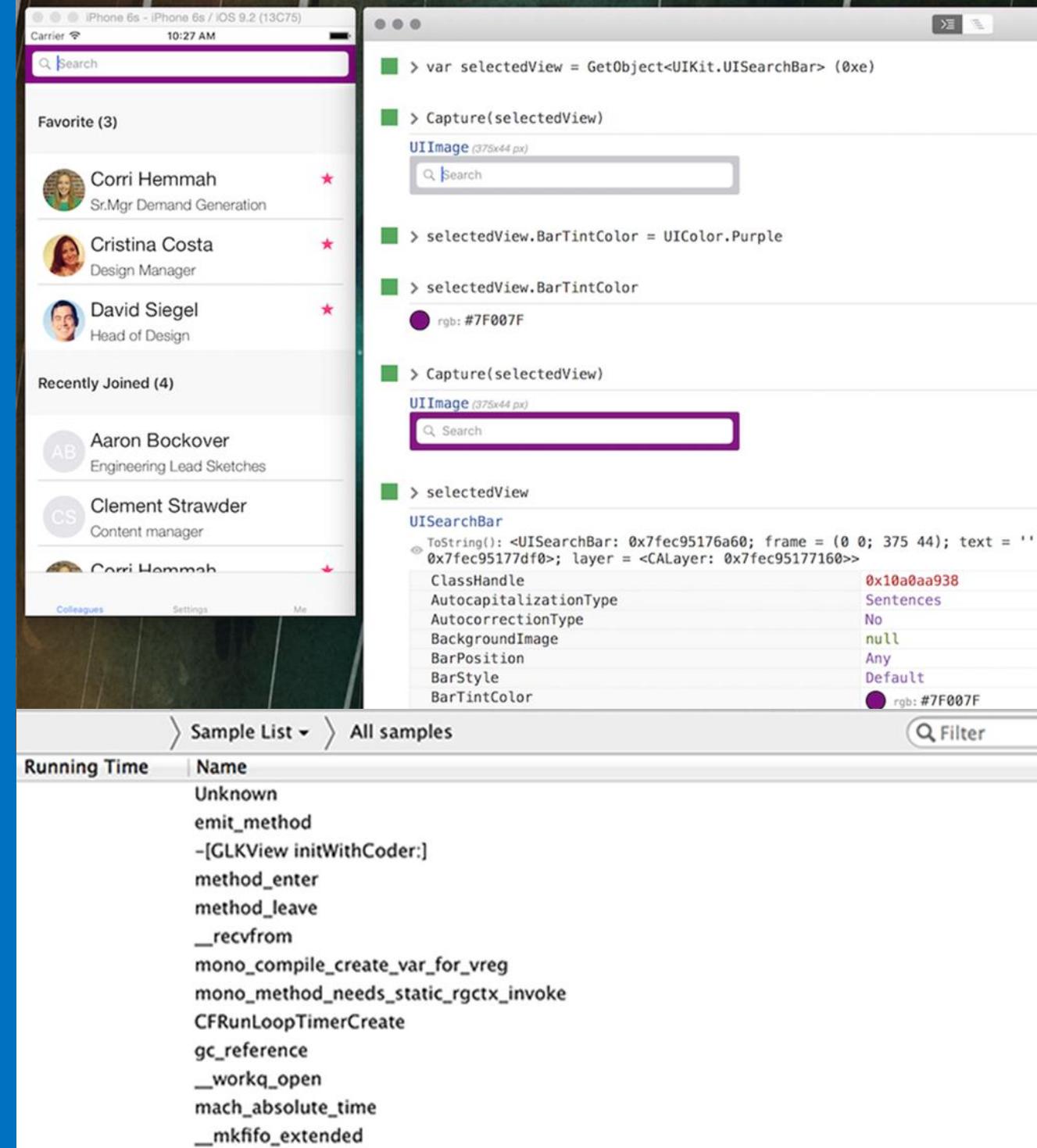


Xamarin Inspector (Preview)

UI debugging made simple

Access a full C# Read-Eval-Print-Loop (REPL) for your mobile applications:

- Dive into the details of UI controls by selecting them to either inspect their details or change them on the fly.
- On Macs, toggle the expanded view to see a detailed view of your application and its properties.

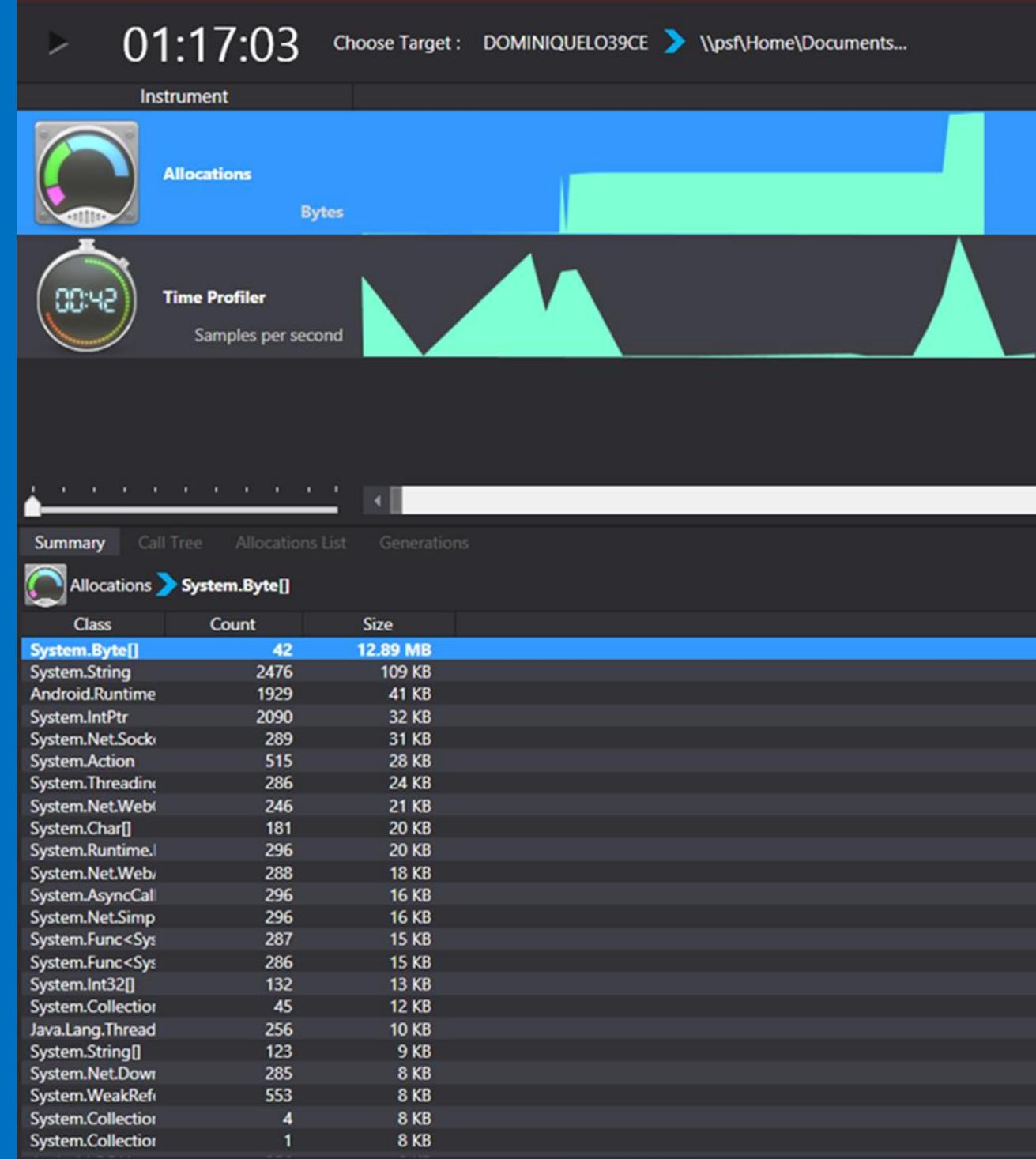


Xamarin Profiler (Preview)

Fine-tune app performance

The Xamarin Profiler collects and displays information about Android and iOS applications from inside of Xamarin Studio or Visual Studio. The profiler is separate from the debugger, so it can run independently to capture profiling metrics.

- Find memory leaks, resolve performance bottlenecks, and polish your apps.
- Run sampling to gain insight into where your app is spending most of its time.
- Integrated with Visual Studio and Xamarin Studio



Excercise 4:

<http://bit.ly/2jAieXE>

In Build – MSBuild task `**/*test*.csproj add
/p:JavaSdkDirectory=" C:\Program Files (x86)\Java\jdk1.8.0_102"`

Additional Resources

- [NUnit](#)
- [Introduction to Xamarin Test Cloud](#)
- [Introduction to Xamarin.UITest](#)

Gathering User Feedback using HockeyApp



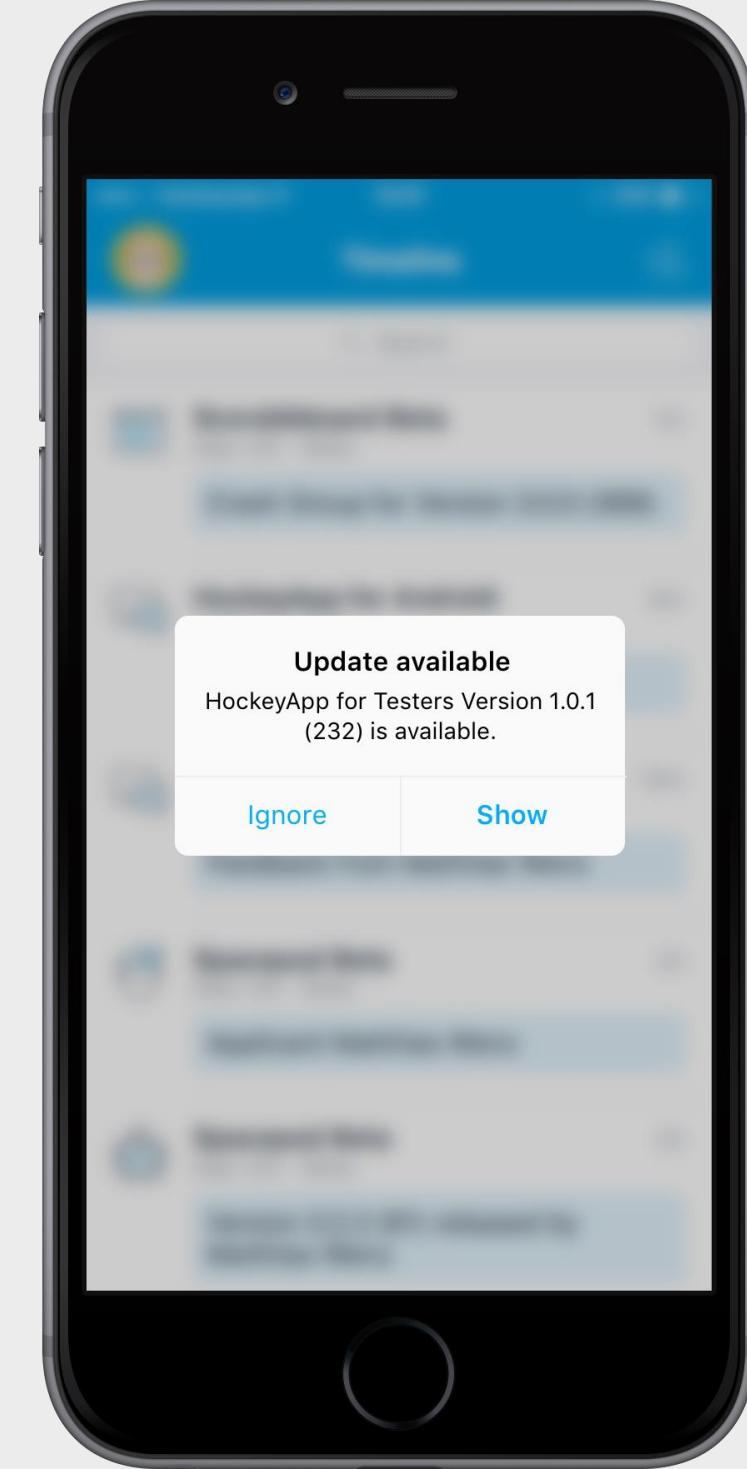
HockeyApp Beta Distribution

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.



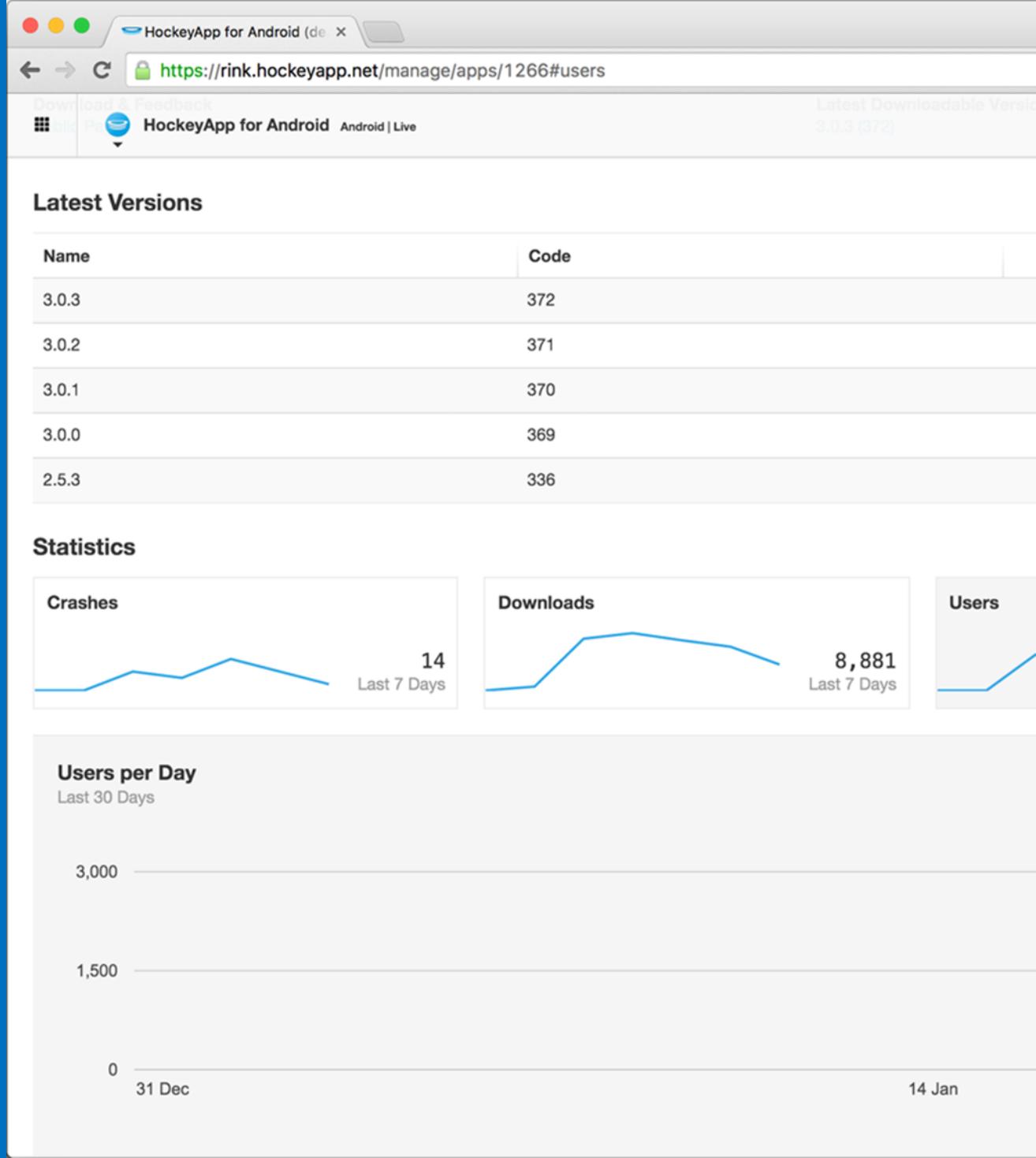
Beta test coverage

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.



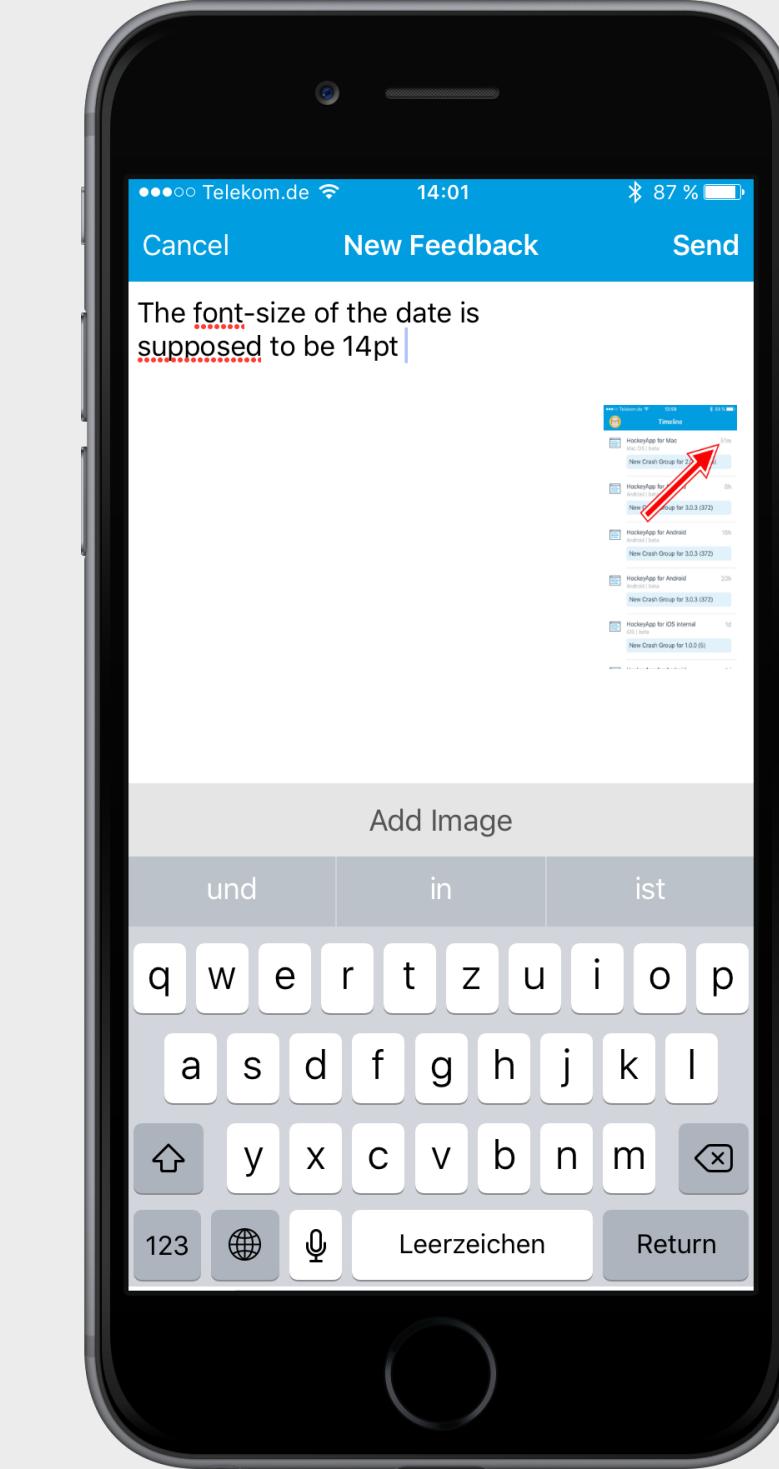
User feedback management

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 at <https://rink.hockeyapp.net/manage/apps/>. The app listed is 'HockeyApp for Android' (Android | Live) and the version is 'Version 2.5.1 (334)'. The 'Crash Group' tab is selected. A specific crash entry is highlighted:

net.hockeyapp.client.HomeActivity\$3.onErrorResponse
in HomeActivity.java, line 267

Reason: java.lang.NullPointerException

Number of Crashes: 132
Percentage for Version: 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

Demo



Azure App Service

Azure App Service

Build and scale great cloud apps



=

 Web Apps

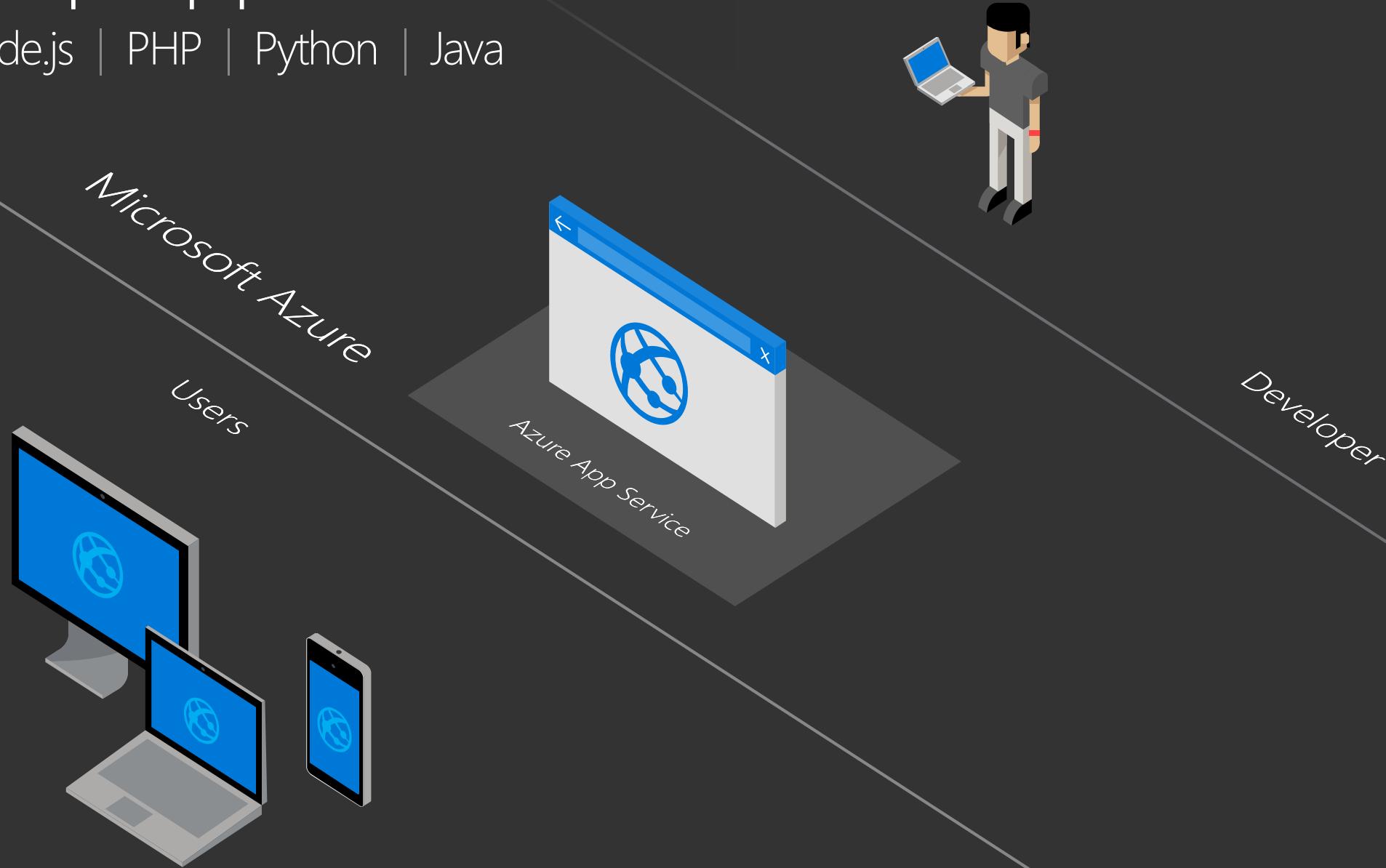
 Mobile Apps

 Logic Apps

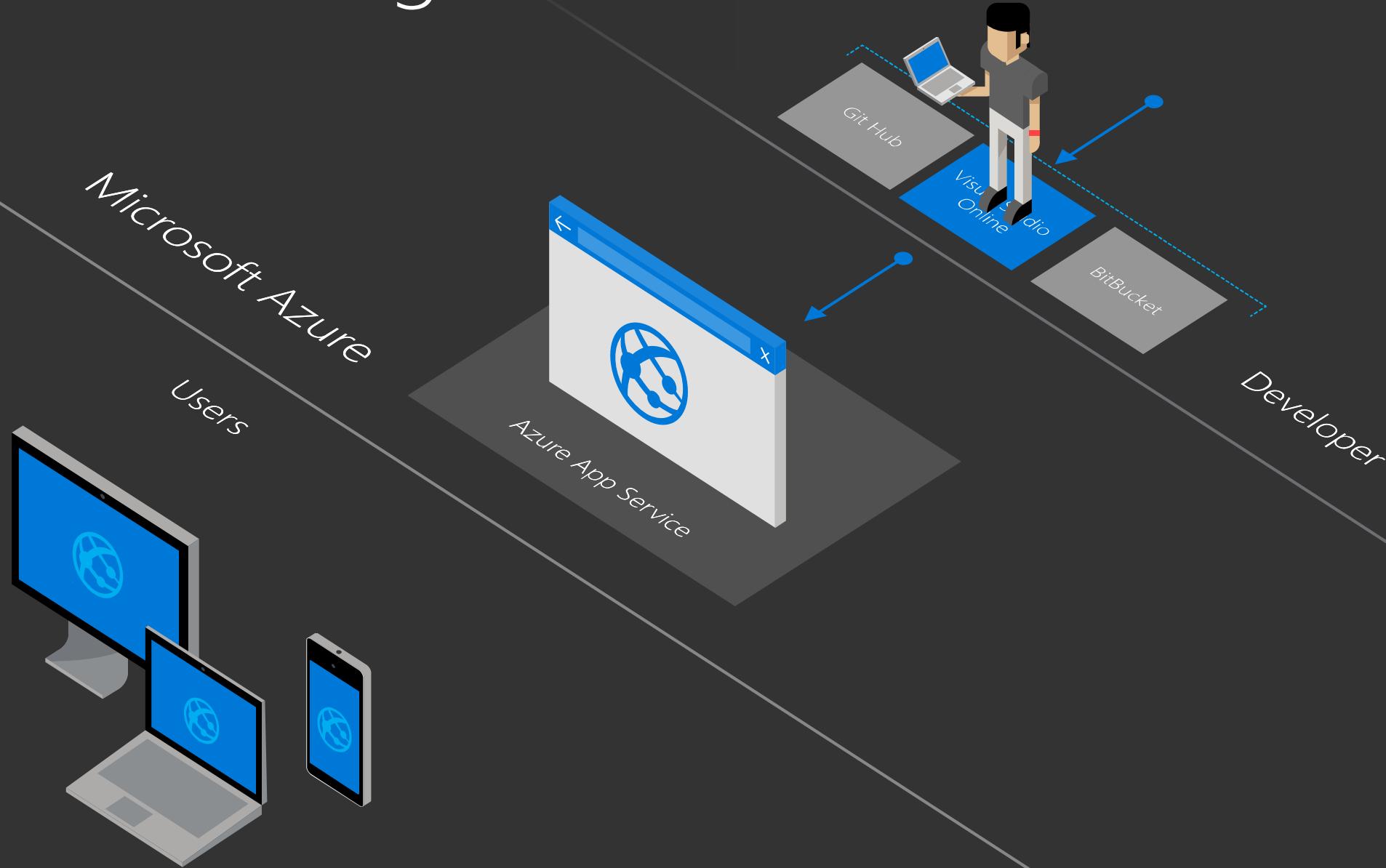
 API Apps

Develop apps with...

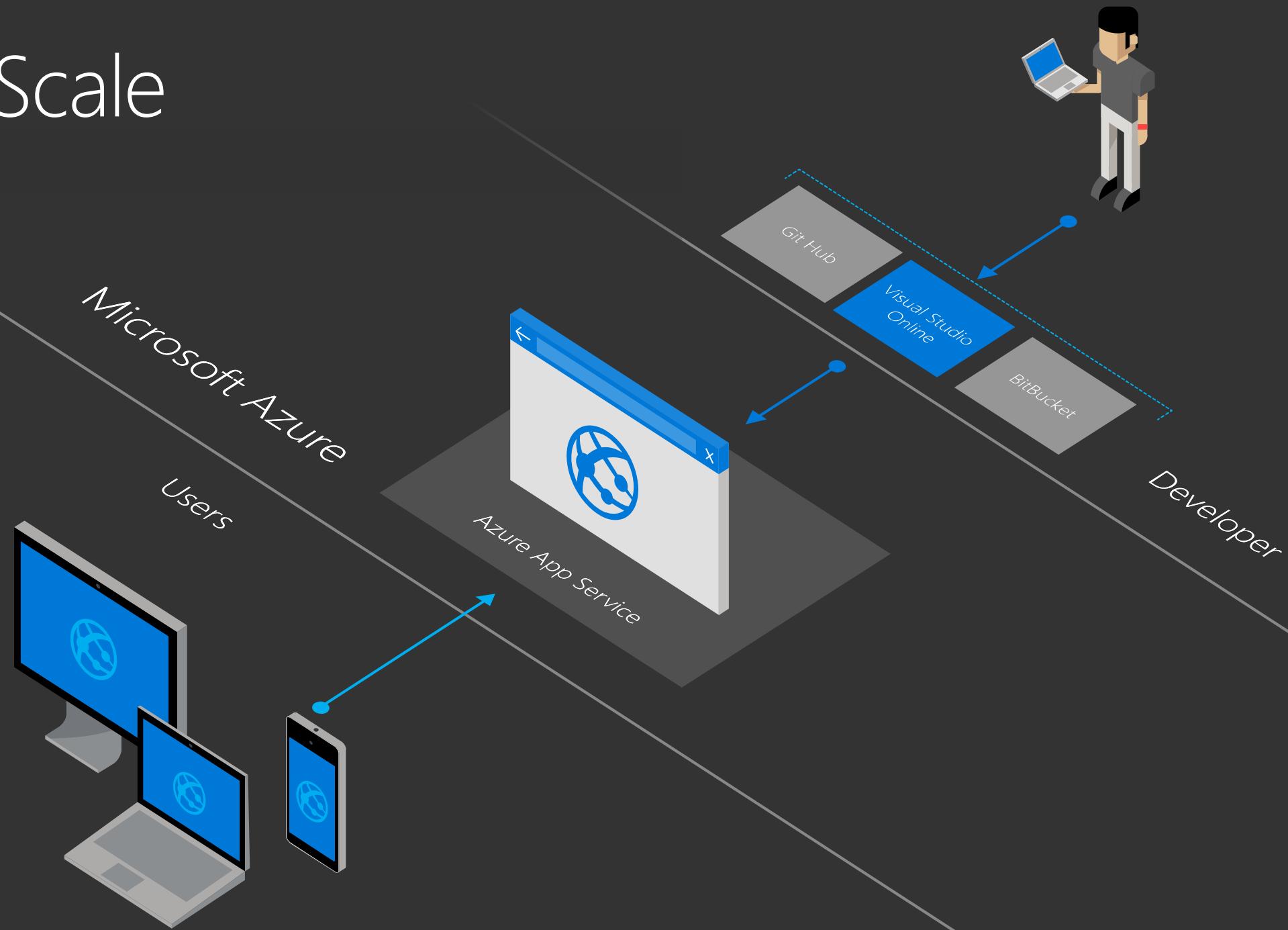
.NET | Node.js | PHP | Python | Java



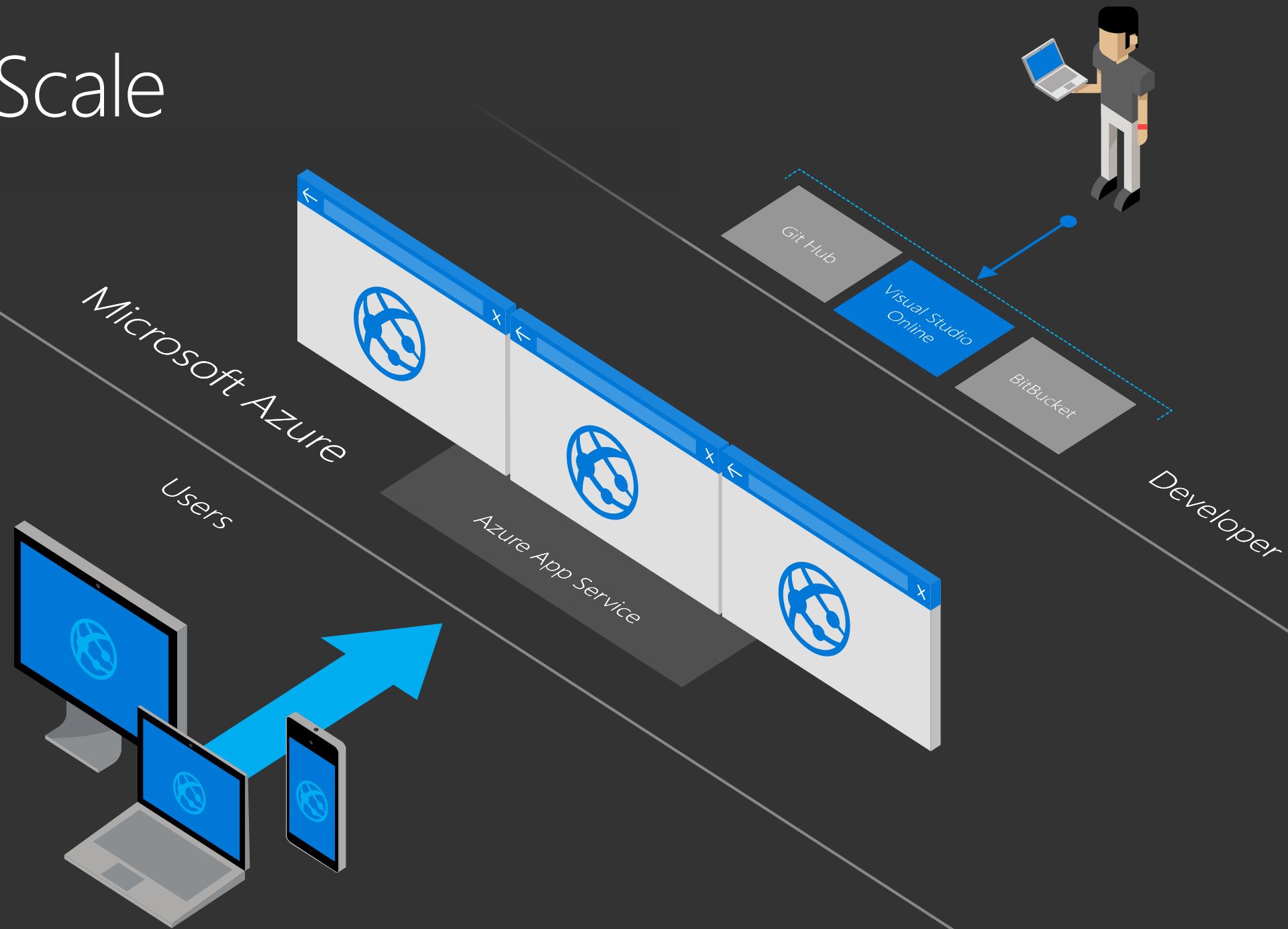
Continuous integration



AutoScale



AutoScale

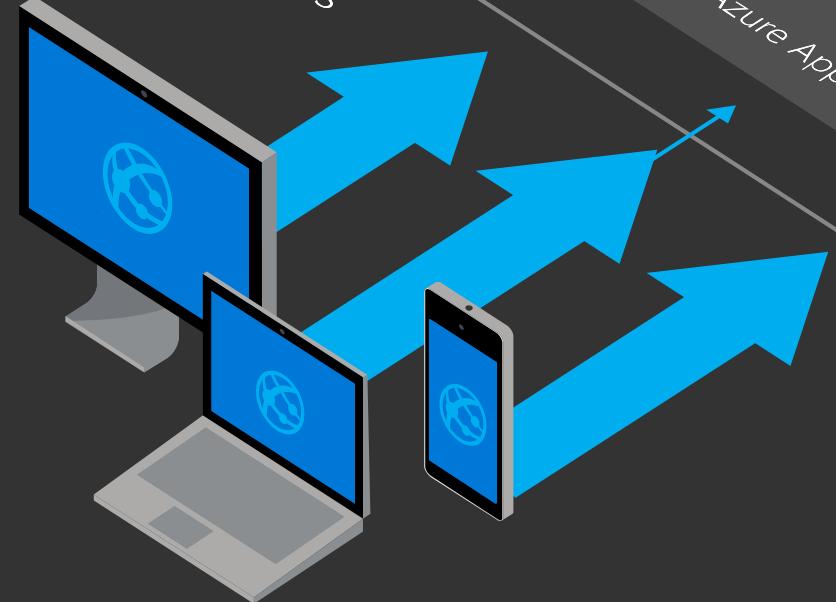


Aus

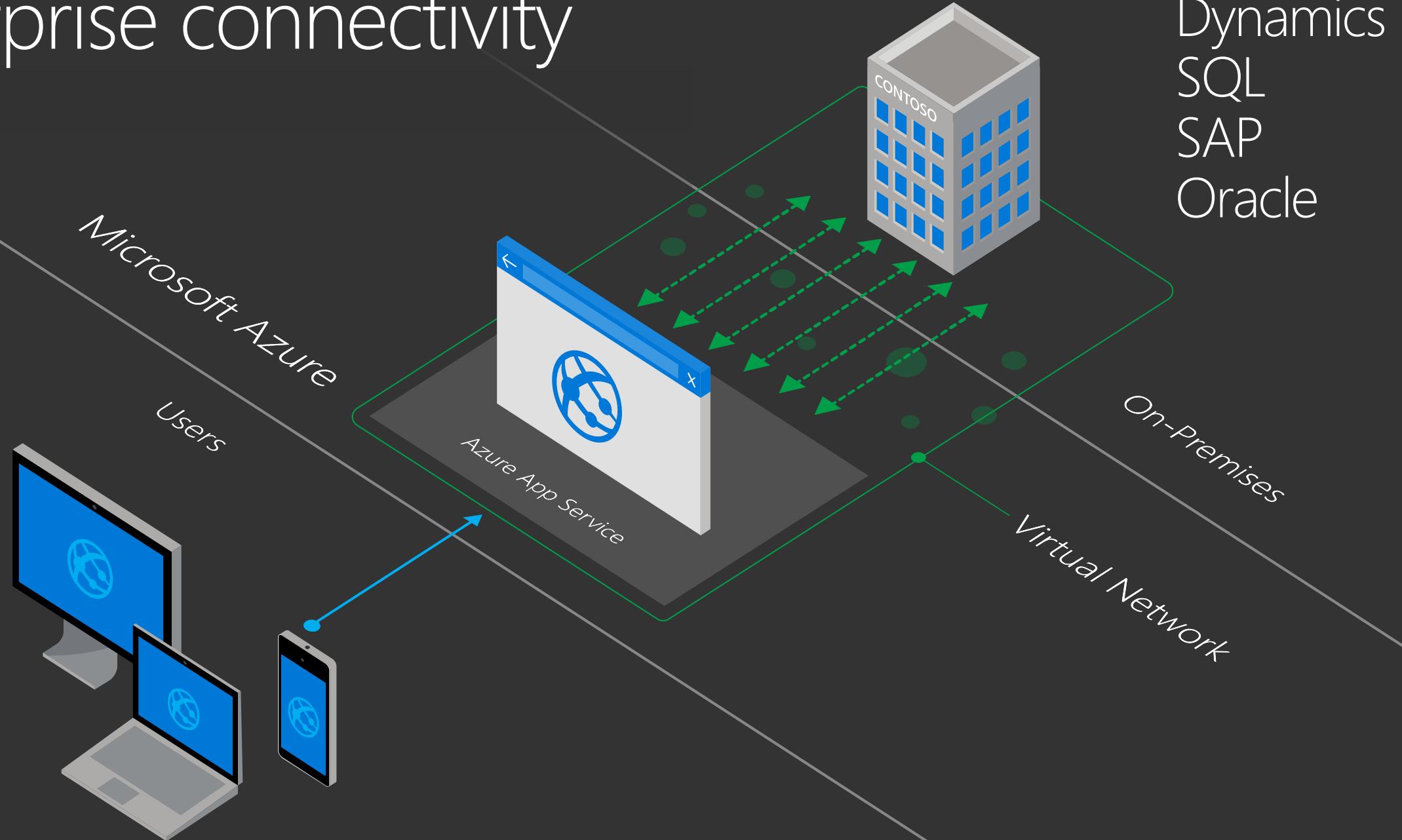
Microsoft Azure

Users

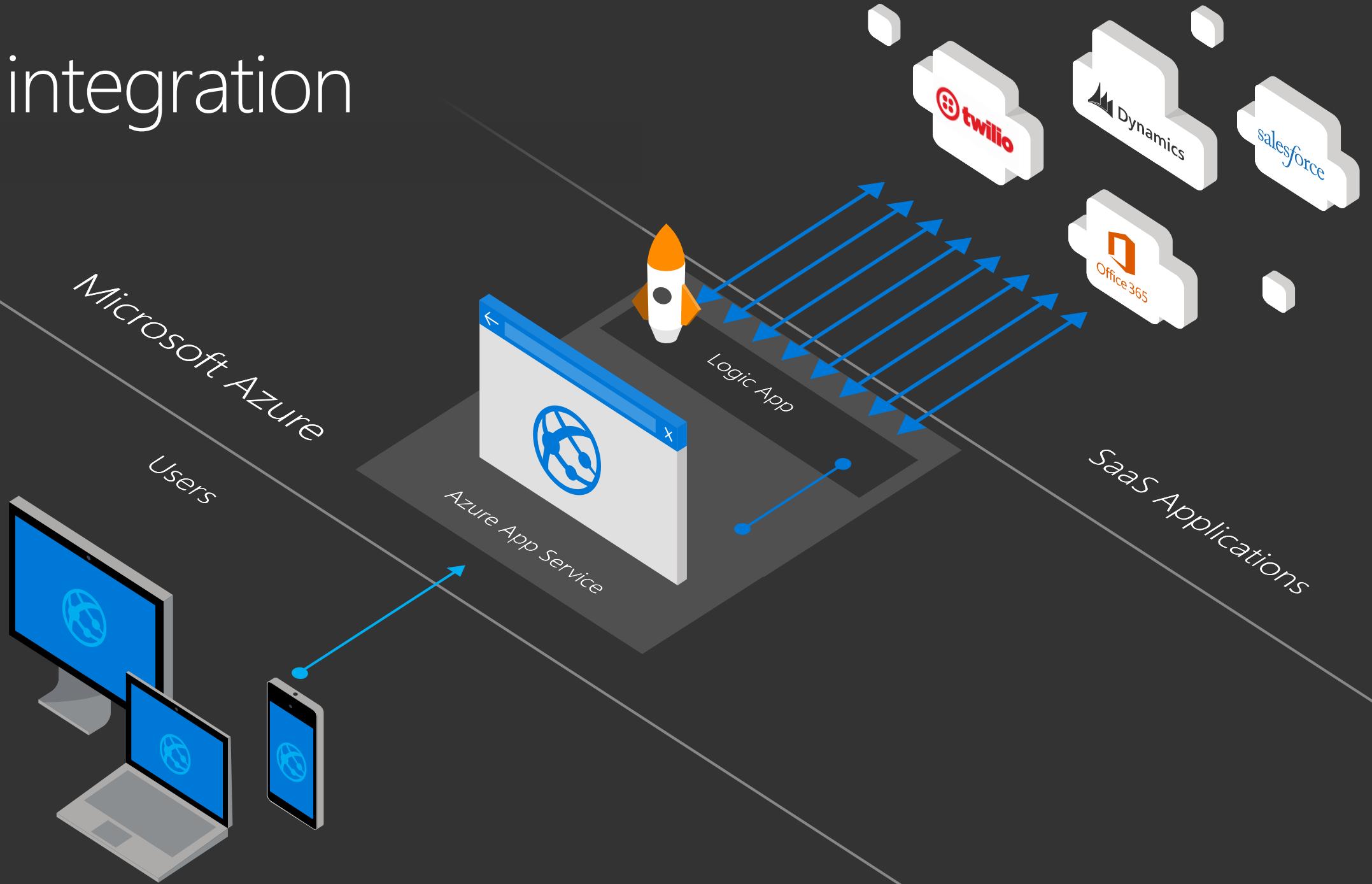
Azure App Service



Enterprise connectivity



SaaS integration



API Apps

- Services: Why?
 - Single Page Applications, Mobile Applications
 - Platform independent
 - Multitier architecture
 - Scalability
- Metadata: Swagger
- CORS-Management
- Deployment Slots, Scaling, ...
- Integration to Logic Apps
- .NET, Node, Python, Java, PHP, ...



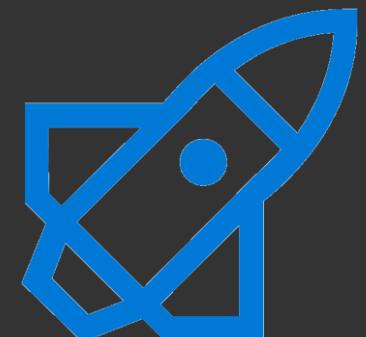
Swagger

- <http://swagger.io/>
- Metadata for APIs
- .NET: Swashbuckle | NuGet
- Swagger UI (Browser for metadata)
- Proxy generation for several platforms



Logic Apps

- Create workflows declaratively
- Design and configuration in the Azure portal
- Define triggers that start the workflow
- Call APIs



FTP Connector

Settings

* Server Address Specify the FTP Server name or IP address

* User Name Specify the user name to use to connect

* Password Specify the password to use to connect

Root Folder Specify a root folder path. File paths relative to this folder will be used.

Use Binary true

Use SSL false

SMTP Connector

Settings

* User Name Username used to authenticate with

* Password Specify the password to use to connect

* Server Address SMTP Server Address

Server Port 587

Enable SSL true

APIs

In DecisionMaker resource group

Microsoft.ApiApp Microsoft.ApiApp

HTTP Microsoft

Recurrence Microsoft

Recommended apps

Dropbox Connector Microsoft

Office365 Connector Microsoft

Salesforce Connector Microsoft

Facebook Connector Microsoft

SharePoint Server Connector Microsoft

Twitter Connector Microsoft

Marketplace apps

API App Microsoft

decision-logic

Logic app - PREVIEW

Settings Select Trigger Edit Delete Move

Essentials

Resource group DecisionMaker

Status Enabled

Location West Europe

Subscription Name Visual Studio Enterprise with MSDN

Subscription ID d4af678a-8b04-41a3-9a50-4dc2ec27db29

Last trigger: Just now Errors: 0

Operations All runs

START TIME	IDENTIFIER	DURATION
▶ 2/19/2016 6:45 AM	08587457461515586947	--
✓ 2/19/2016 6:40 AM	08587457464379379237	2.14 Seconds

Triggers and actions

decision-logic - PREVIEW

Save Discard Designer Code view Zoom in Zoom out Language Reference Help

// HTTP ...

← GET URI http://decisionmaker-api.azurewebsites.n...

Headers

Authentication

headers body

body Status

Twilio Connector ...

← Send Message From Phone Number +43 676 800

To Phone Number +436991434

Text @{{body('http')}}

body

Start logic

Start by adding a trigger of choice from the list on the right

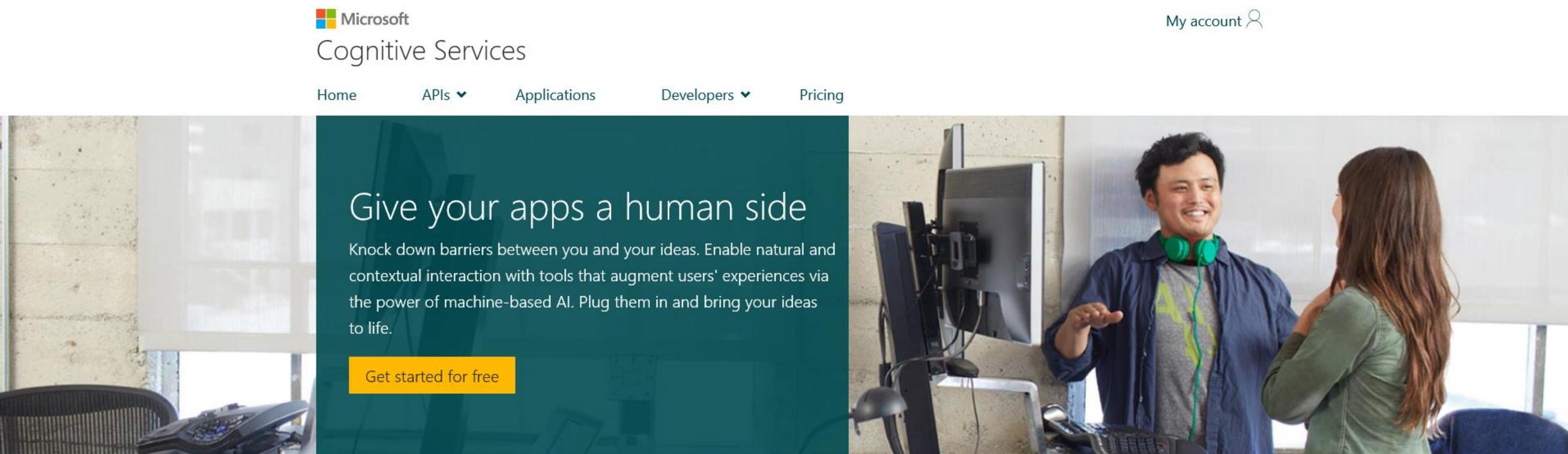
Run this logic manually

Mobile Apps

- Backend for mobile apps
- Solutions for common scenarios
 - Authentication
 - Store data
 - Push Notifications
 - Auto Scaling



Cognitive Services



The screenshot shows the Microsoft Cognitive Services homepage. At the top, there's a navigation bar with the Microsoft logo, a "My account" link, and a search icon. Below the navigation is a main banner with a teal overlay. The banner features the text "Give your apps a human side" and a description about enabling natural interaction through machine-based AI. A yellow "Get started for free" button is at the bottom of the banner. To the right of the banner is a photograph of two people, a man and a woman, smiling and interacting near a computer setup with multiple monitors. On the far left, there's a partial view of what looks like a keyboard or a typewriter.

Microsoft

My account

Cognitive Services

Home APIs Applications Developers Pricing

Give your apps a human side

Knock down barriers between you and your ideas. Enable natural and contextual interaction with tools that augment users' experiences via the power of machine-based AI. Plug them in and bring your ideas to life.

Get started for free

Put intelligence APIs to work

Microsoft Cognitive Services let you build apps with powerful algorithms using just a few lines of code. They work across devices and platforms such as iOS, Android, and Windows, keep improving, and are easy to set up.

Cognitive Services



VISION



SPEECH



LANGUAGE



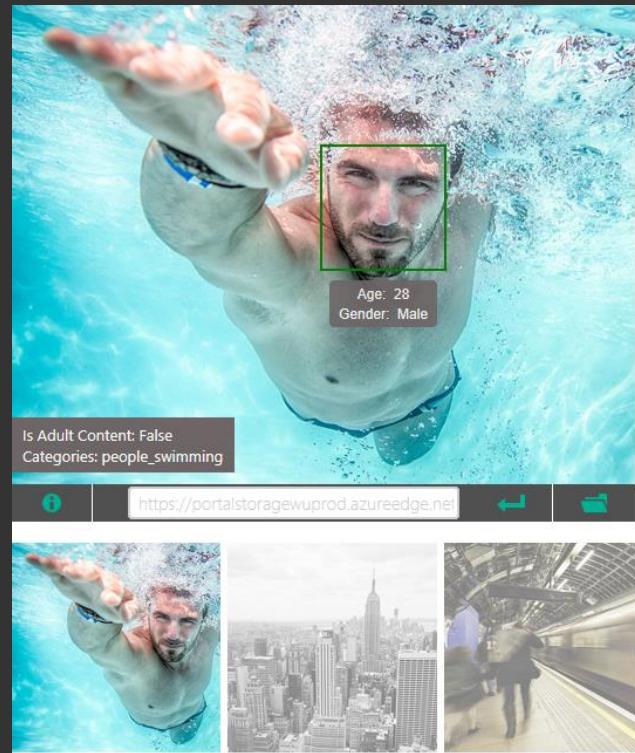
KNOWLEDGE



SEARCH

Vision

- Analyze images and videos
- Face recognition
- Age estimation
- Detect emotions



Is Adult Content: False
Categories: people_swimming

Age: 28
Gender: Male

Features:

Feature Name	Value
Description	{ "type": 0, "captions": [{ "text": "a man swimming in a pool of water", "confidence": 0.7850108693093019 }] }
Tags	[{ "name": "water", "confidence": 0.9996442794799805 }, { "name": "sport", "confidence": 0.9504992365837097 }, { "name": "swimming", "confidence": 0.9062818288803101, "hint": "sport" }, { "name": "pool", "confidence": 0.8787588477134705 }, { "name": "water sport", "confidence": 0.631849467754364, "hint": "sport" }]
Image Format	jpeg
Image Dimensions	1500 x 1155
Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-LineDrawing
Black & White Image	False

Speech

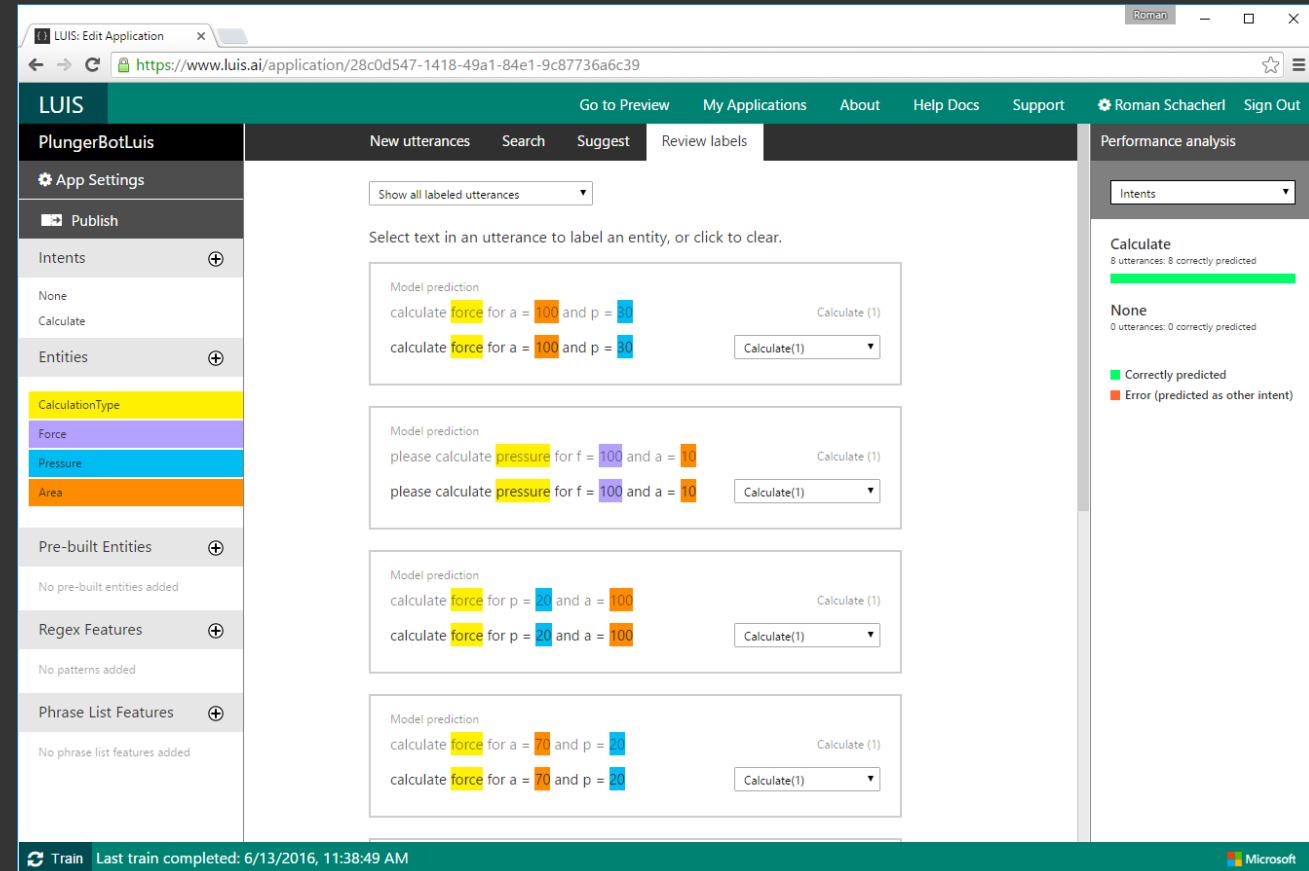
- Speech-to-text
- Text-to-speech
- Custom recognition (CRIS)
- Speaker recognition

The screenshot shows a user interface for speaker recognition. At the top left, there's a section titled "Speakers provided" with a grid of five portrait photos of US presidents: Barack Obama, George W. Bush, Bill Clinton, George H.W. Bush, and Ronald Reagan. Below this, a green button labeled "Upload audio for identification" with an upward arrow icon is visible. To the right, a "Result" tab is selected, showing the text "President **Barack Obama** is the one identified speaking in the selected audio." A "JSON" tab is also present. At the bottom, there are five green buttons labeled "Audio 1" through "Audio 5", each with a play icon.

LUIS

(Language Understanding Intelligent Service)

- Understand speech
 - Intention
 - Keywords
- Machine Learning
- Costs
 - 100.000 transactions per month for free
 - 10 transactions/second: 0,75 \$ per 1000 transactions



Knowledge

- Frequently Bought Together recommendations
- Item to item recommendations



Developing better software today,
every day.

Conveniently located at the
intersection of software
engineering, technology
operations, quality assurance, and
the business.

Practitioners are looking for faster
time-to-market, lower failure rates,
faster fixes, and shorter recovery.

Learn how the DevOps Factory
can help you deliver better apps
faster, while making the most
efficient use of all IT resources.

<https://thedevopsfactory.com>

Proposed follow up actions for YOU:

- **Try out** Visual Studio Enterprise with Xamarin for free: [go here](#)
- For partners: we encourage you to **redeliver this training to your customers**. Here are all the materials you need:
 - Github repo where all the **sample apps** and **hands on labs** reside: <https://github.com/softawaregbh/MobileDevOps/>
 - Main Slides for the training: <https://1drv.ms/p/s!AnByKdQdlw5KgjifodS-DutFnfQjn>
 - Additional slide deck just focusing on Xamarin + Android (as an extension to the above slide deck): <https://1drv.ms/p/s!AqM7MHvEOS1jisggm0eeZNpyPYUiUQ>
- Don't miss to **join** our [ALM and DevOps Yammer network](#) (for customers & partners) and our dedicated [Yammer DevOps/ALM partner group](#).
- **Watch and share** the following webinar series **with colleagues or customers**:
 - Series 1: **Cross platform mobile development using Xamarin** for building apps for Android, iOS & Windows. [Webinar 1](#), [Webinar 2](#), [Webinar 3](#)
 - Series 2: **Web and [Mobile] API back-ends on Azure** that allow developers to focus on code and use the Azure platform to take care of hosting, deployment, scaling & security. [Webinar 1](#), [Webinar 2](#), [Webinar 3](#)
- **Study and share** the following interesting assets **with colleagues or customers** :
 - Read the "[Total Economic Impact of the Microsoft DevOps Solution](#)" Forrester report.
 - Get the [Azure Developer Guide](#).
 - Watch a webinar to Learn from the experts [how Visual Studio and Xamarin help you create rich apps across platforms](#).
 - Study the [guide to building great apps](#).
 - Explore [how Azure and Xamarin connect together](#).
 - Download this [ebook about Xamarin Forms](#).



Microsoft