

Как сделать быстрое мобильное приложение один раз для всех платформ

Ромуальд Здебский
Microsoft
rzdebski@microsoft.com

Трудности разработки
одновременно под
несколько платформ

Знакомые ситуации

- Нужно иметь отдельные команды разработки под каждую платформу
- Одну и ту же логику приложения приходится программировать несколько раз
- Разработка обходится дороже и занимает больше времени – несколько команд.
- Иногда её реализовывают несколько по разному для каждой платформы = баги
- Каждый баг надо исправлять несколько раз
- Иногда баги исправляют разные команды по разному = новые баги



Нативный
интерфейс



Нативные API



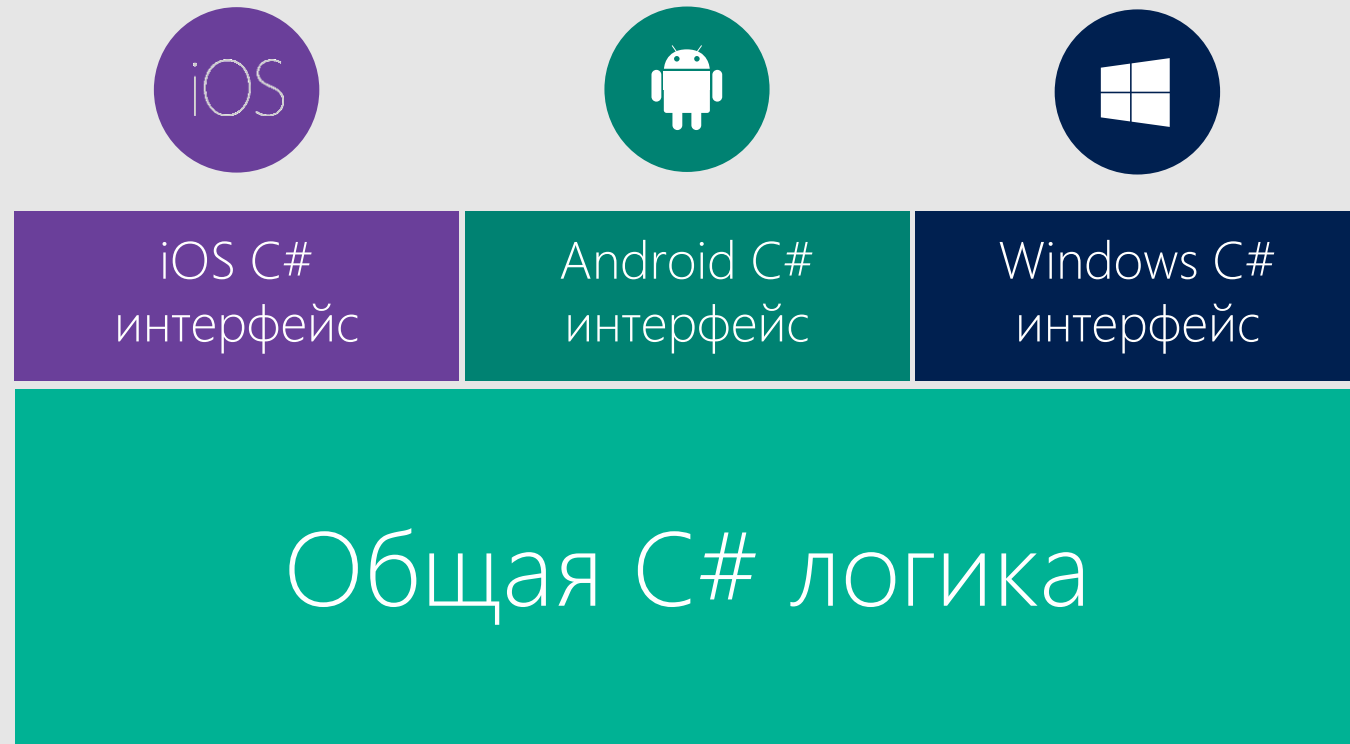
Нативная
скорость

Фреймворки на основе веб технологий



Ограниченный доступ к API • Медленно • Неудобный интерфейс

Подход Xamarin Classic/Native



Общий C# код • 100% доступ к нативным API • Высокая производительность

✓ Всегда доступ к актуальным API

Доступность в день
выпуска:

- iOS 5
- iOS 6
- iOS 7
- iOS 8
- iOS 9
- iOS 10
- iOS 11

Полная поддержка:

- Apple Watch
- Apple TV
- Android Wear
- Amazon Fire TV
- Google Glass
- и многих других

Сравнение производительности нативных приложений и приложений Xamarin

Most likely, users won't be able to tell the difference between native code and Xamarin.iOS or Xamarin.Android.

Too many factors impact user experience. Low connection speed combined with the poor backend is the main source of annoyance among app users. Only a small portion of UX problems are related to installed apps directly. It's worth mentioning that a human can perceive and recognize performance delays [above 500 ms](#). We tested both simple and complicated operations and most of them stay within or below this benchmark.

<https://www.altexsoft.com/blog/engineering/performance-comparison-xamarin-forms-xamarin-ios-xamarin-android-vs-android-and-ios-native-applications/>

ARM, Samsung Galaxy Note 2, power saving off, Android 4.1.1

Xamarin:

Grand total time (5 runs): 6201 ms, with file reading total: 6476 ms

Java:

Grand total time (5 runs): 7141 ms, with file reading total: 7479 ms

<https://stackoverflow.com/questions/17134522/does-anyone-have-benchmarks-code-results-comparing-performance-of-android-ap>

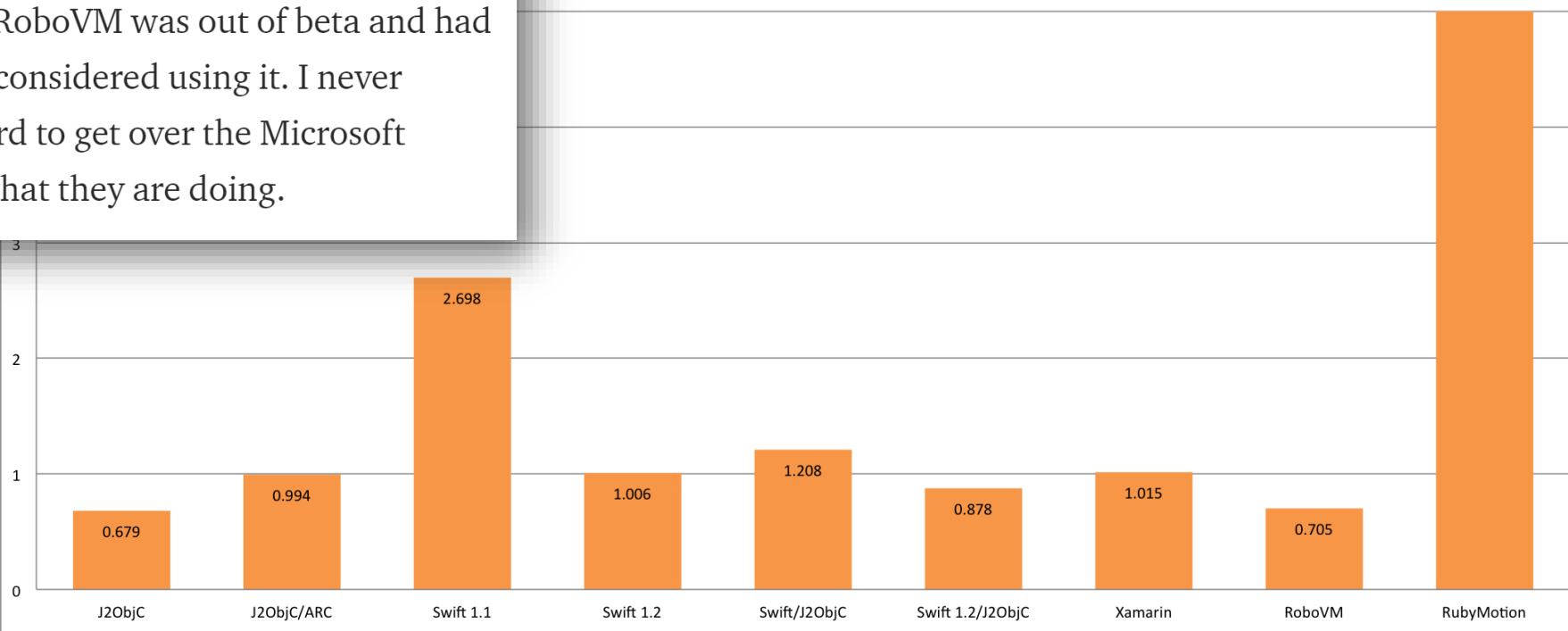
⊕ Performance Close to Native

Unlike traditional hybrid solutions, based on the web technologies, a cross-platform app built with Xamarin, can still be classified as native. The performance metrics are comparable to those of Java for Android ([as explained here](#)) and Objective-C or Swift for [native iOS app development](#). Moreover, [Xamarin performance](#) is constantly being improved to fully match the standards of native development. Xamarin

<https://www.altexsoft.com/blog/mobile/the-good-and-the-bad-of-xamarin-mobile-development/>

Having said that, the performance-minded engineer in me plans on developing the rest of my app in Xamarin. If RoboVM was out of beta and had better support for storyboards, I would have considered using it. I never thought I would be a C# developer as it is hard to get over the Microsoft stigma, but alas, these Xamarin guys know what they are doing.

Average iOS Computational Performance



Average iOS Computational Performance in seconds (lower is better)

	J2ObjC	J2ObjC/ARC	Swift 1.1	Swift 1.2	Swift 1.1/J2ObjC	Swift 1.2/J2ObjC	Xamarin	RoboVM	RubyMotion
Launch	17.83 MB	17.84 MB	18.51 MB	17.45 MB	17.91 MB	17.90 MB	9.96 MB	3.63 MB	19.90 MB
Peak	18.17 MB	18.29 MB	19.09 MB	17.90 MB	18.80 MB	18.73 MB	10.06 MB	3.67 MB	21.95 MB
Settle	18.10 MB	18.27 MB	18.77 MB	17.53 MB	18.48 MB	18.23 MB	10.12 MB	3.64 MB	21.75 MB
Total	973.59 MB	895.85 GB	901.15 MB	104.99 MB	980.52 MB	964.53 MB	147.49 MB	29.97 MB	5.49 GB
Run time	14.219 s	23.052 s	67.262 s	1.6 s	25.560 s	15.401 s	4.28 s	4.108 s	606.589 s

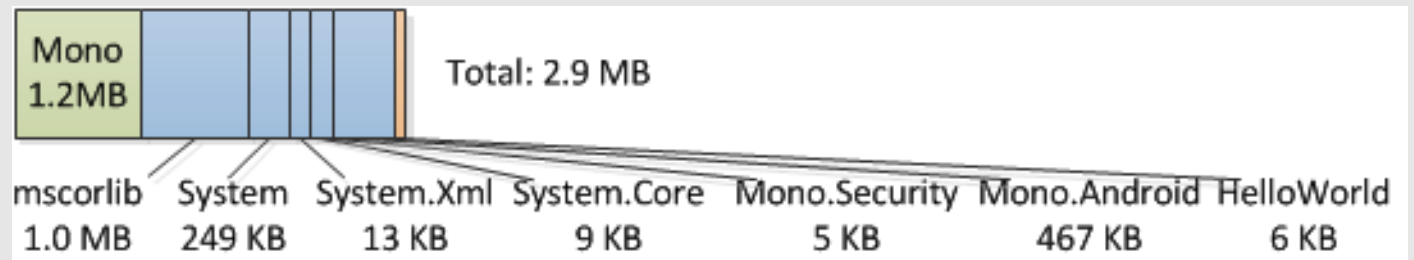
iOS Memory Performance table

Размер пакетов приложений

- Базовый



- Оптимизированный с помощью базового Linking



Linker Options

Linker behavior:

Link SDK assemblies only

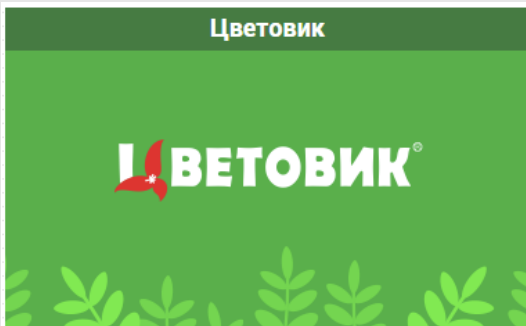
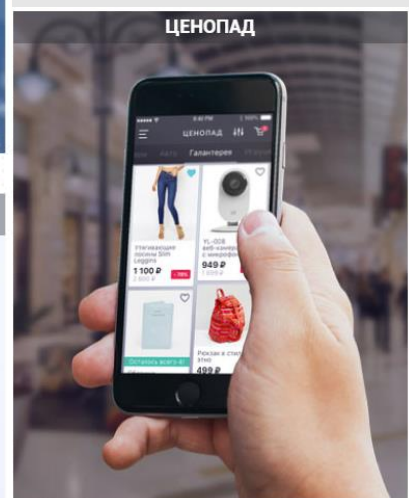
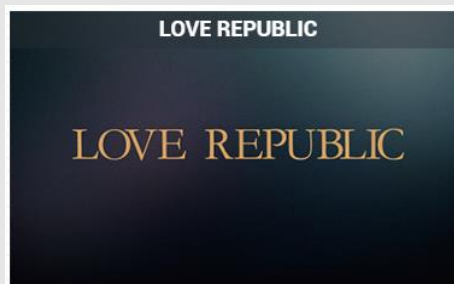
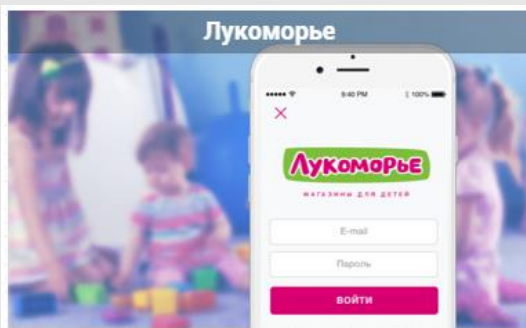
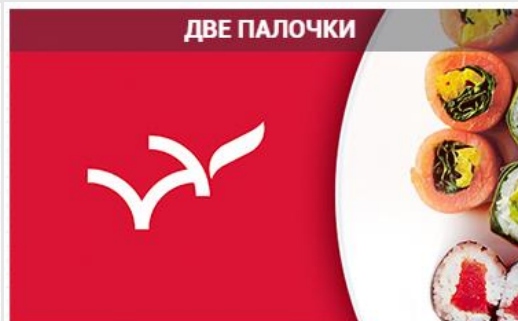
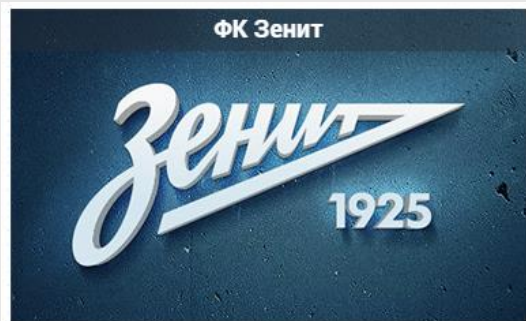
Don't link

Link SDK assemblies only

Link all assemblies

Packaging Options

Примеры в РФ Xamarin Classic/Native



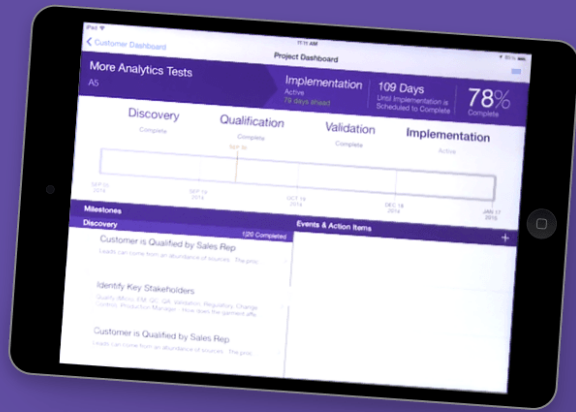
Глобальные примеры Xamarin



Mobilizing the selling process with Xamarin drives millions in extra revenue for Kimberly-Clark.

Kimberly-Clark's Global Scientific Division uses the Apex iPad app to streamline the selling process for clean room environments. The app, built by Xamarin Premier consulting partner BlueTube, is drastically reducing the sales cycle after their first attempt with an HTML5-based app wasn't adopted by sales representatives.

- ✓ Reused the majority of their existing C# code and existing web services
- ✓ App integrates with Salesforce and Sitecore
- ✓ Kimberly-Clark has seen a significant increase in revenue



National Instruments uses LabVIEW to turn tablets into powerful engineering tools.

LabVIEW is the industry standard in design software for engineers and scientists creating and deploying measurement and control systems. The app helps engineers remotely control and monitor running LabVIEW applications. Users can create dashboards to display the values of network-published shared variables and deployed LabVIEW Web services on indicators.

- ✓ The company has been creating, deploying, and testing the Internet of Things for decades
- ✓ Mobilizing machine monitoring means less downtime as failures are predicted and fixed faster
- ✓ Xamarin turned their teams into mobile developers in just weeks



Coca-Cola Bottling Co. Consolidated drives sales with dynamic sales proposals.

Coca-Cola Bottling Co. Consolidated is the largest independent Coca-Cola bottling distributor in the US, with over \$1.5 billion in annual revenue. The field sales teams use iPads equipped with the Marketplace Xamarin app to help them sell more Coke products and create awareness around various programs.

- ✓ Streamlines DevOps across distributed teams with Visual Studio, Xamarin, Azure, and Visual Studio Team Services
- ✓ Allows productivity from remote locations or limited connectivity customer sites with Azure sync
- ✓ Leveraged in-house .NET skills to ship apps in four months

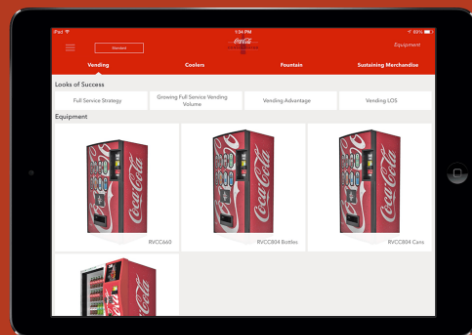
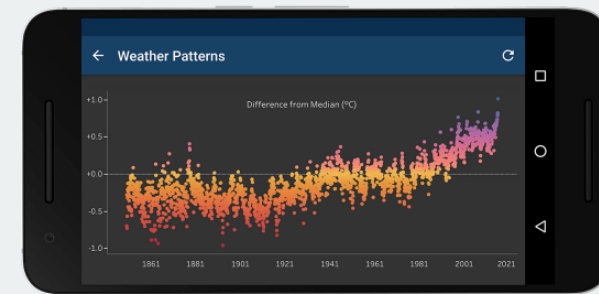


Tableau brings data to life with beautiful apps.

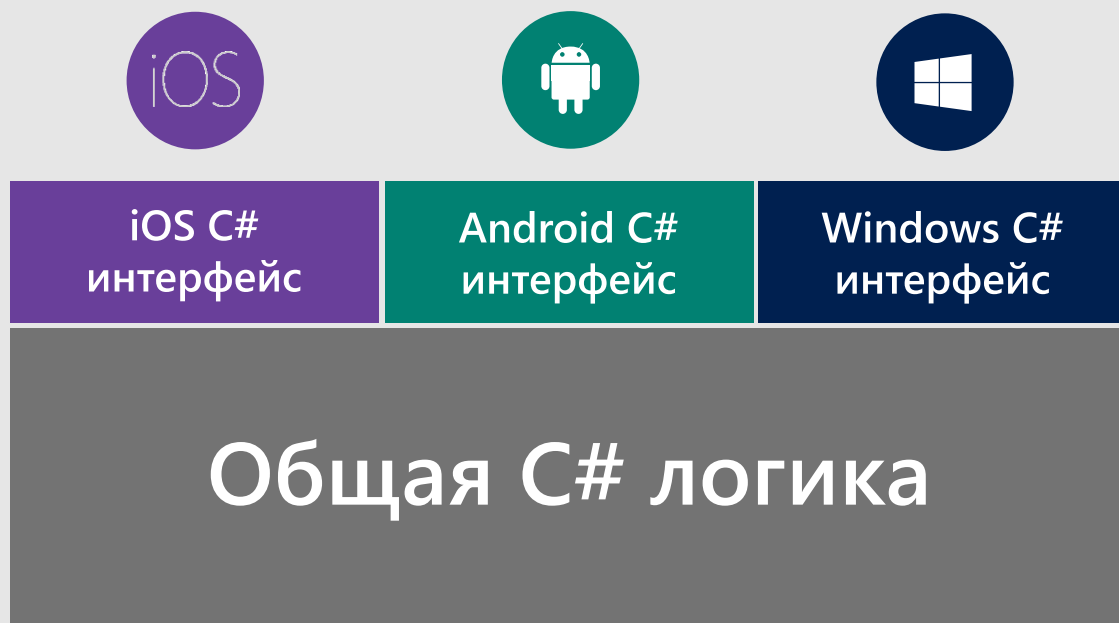
Tableau is revolutionizing visual analytics, helping its customers see and understand their data. On-the-go users need immediate access to data from airports, restaurants, or customer sites to make informed decisions. Tableau recently rewrote its legacy Tableau Mobile app with Xamarin, enabling its development team to deliver high fidelity, high quality, and delightful apps that reach more global users and allow customers to maximize their Tableau investments.

- ✓ Migrated existing Android and iOS platform-specific apps to Xamarin, resulting in a higher quality app with near-zero crash rate
- ✓ Secure and frictionless connection with enterprise SSO
- ✓ Fast and fluid user experience makes data the star
- ✓ Customers go from questions to answers in just a few taps
- ✓ Automatic sync displays beautiful dashboard snapshots while offline

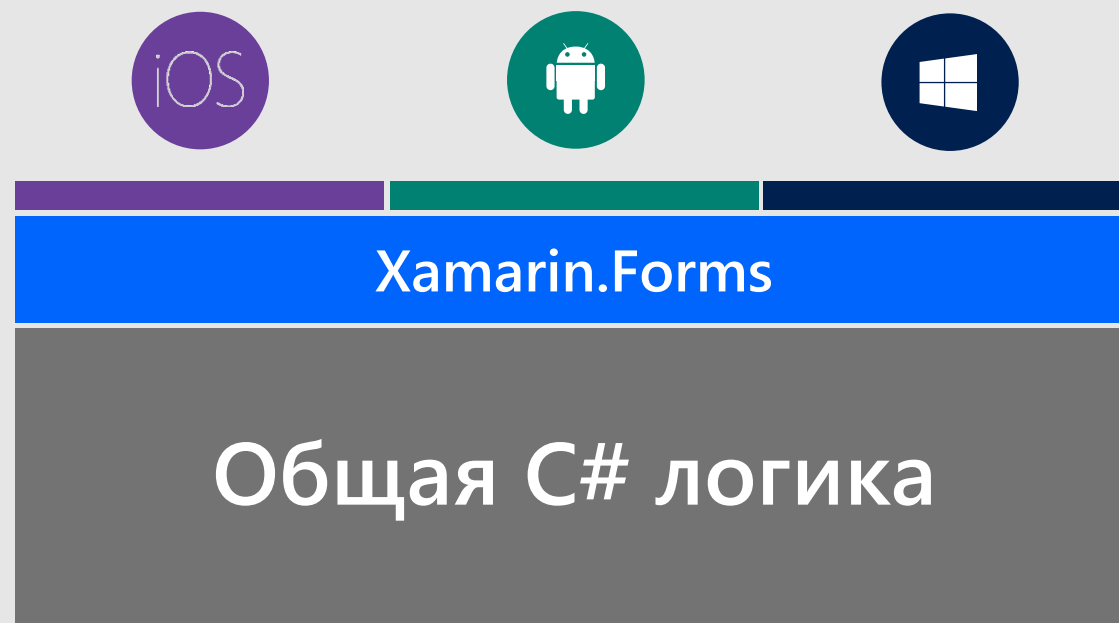


<https://www.xamarin.com/customers>

Xamarin Forms – еще больше общего кода

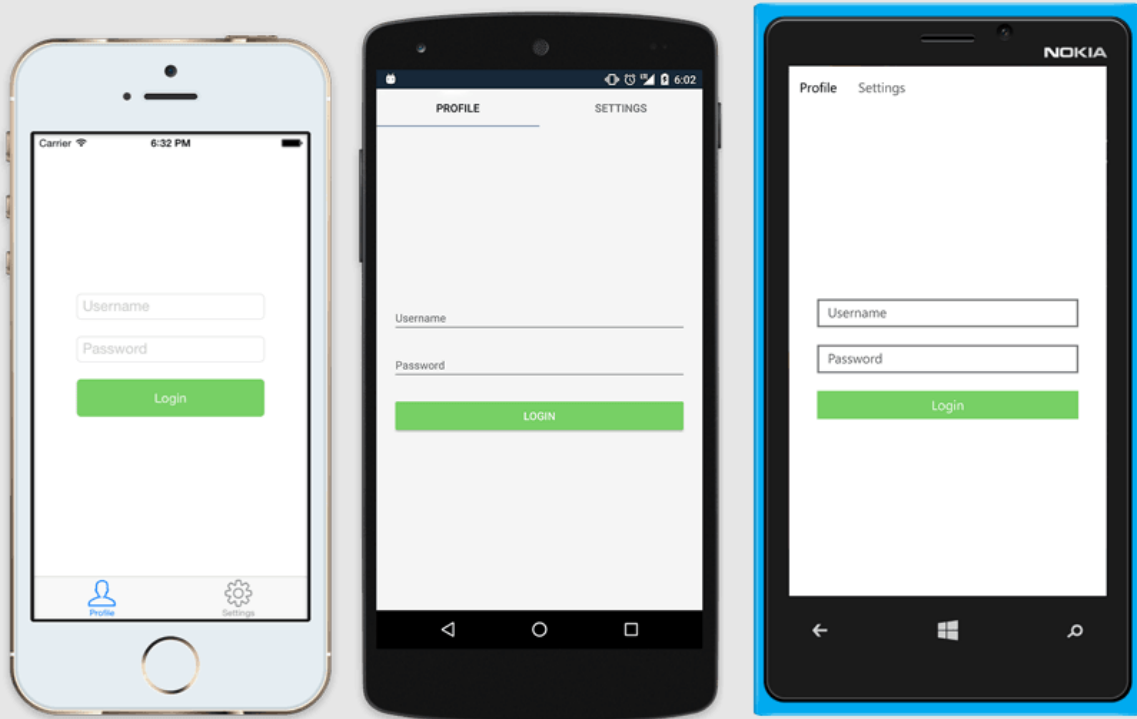


Xamarin Classic/Native



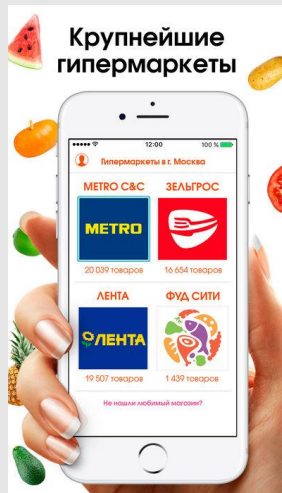
С Xamarin.Forms:
все по прежнему нативное,
еще больше общего кода

Нативный интерфейс из общего кода

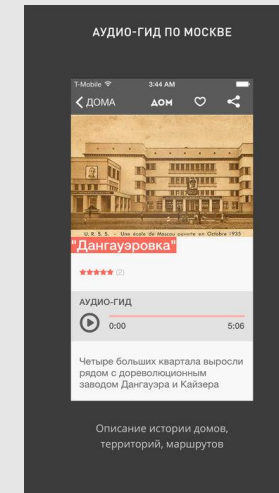
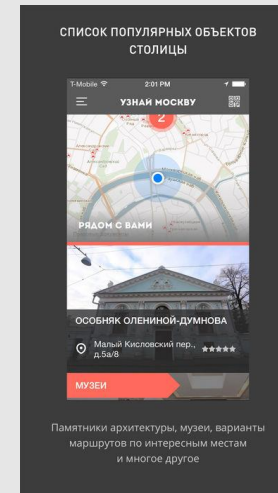
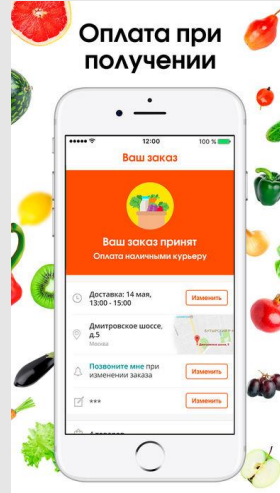
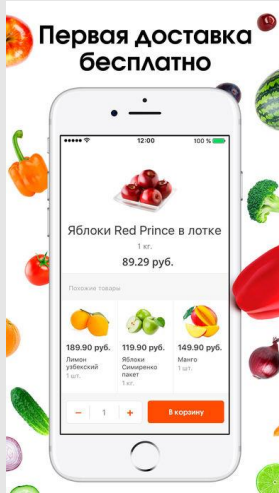


```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="MyApp.MainPage">
  <TabbedPage.Children>
  <ContentPage Title="Profile" Icon="Profile.png">
    <StackLayout Spacing="20" Padding="20"
      VerticalOptions="Center">
      <Entry Placeholder="Username"
        Text="{Binding Username}"/>
      <Entry Placeholder="Password"
        Text="{Binding Password}"
        IsPassword="true"/>
      <Button Text="Login" TextColor="White"
        BackgroundColor="#77D065"
        Command="{Binding LoginCommand}"/>
    </StackLayout>
  </ContentPage>
  <ContentPage Title="Settings" Icon="Settings.png">
    <!-- Settings -->
  </ContentPage>
  </TabbedPage.Children>
</TabbedPage>
```

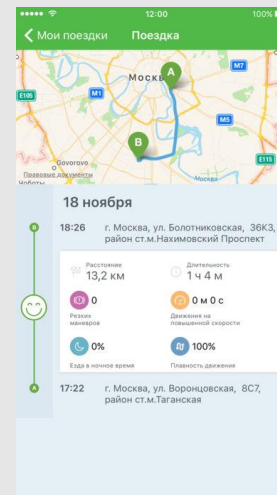
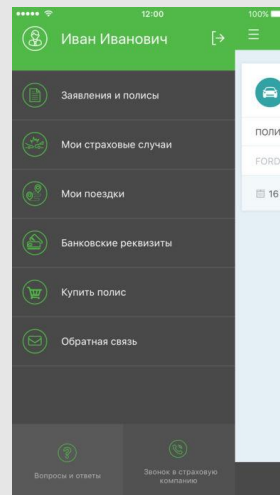
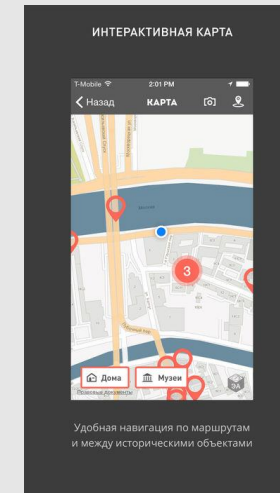
Примеры приложений Xamarin Forms



Инстамарт



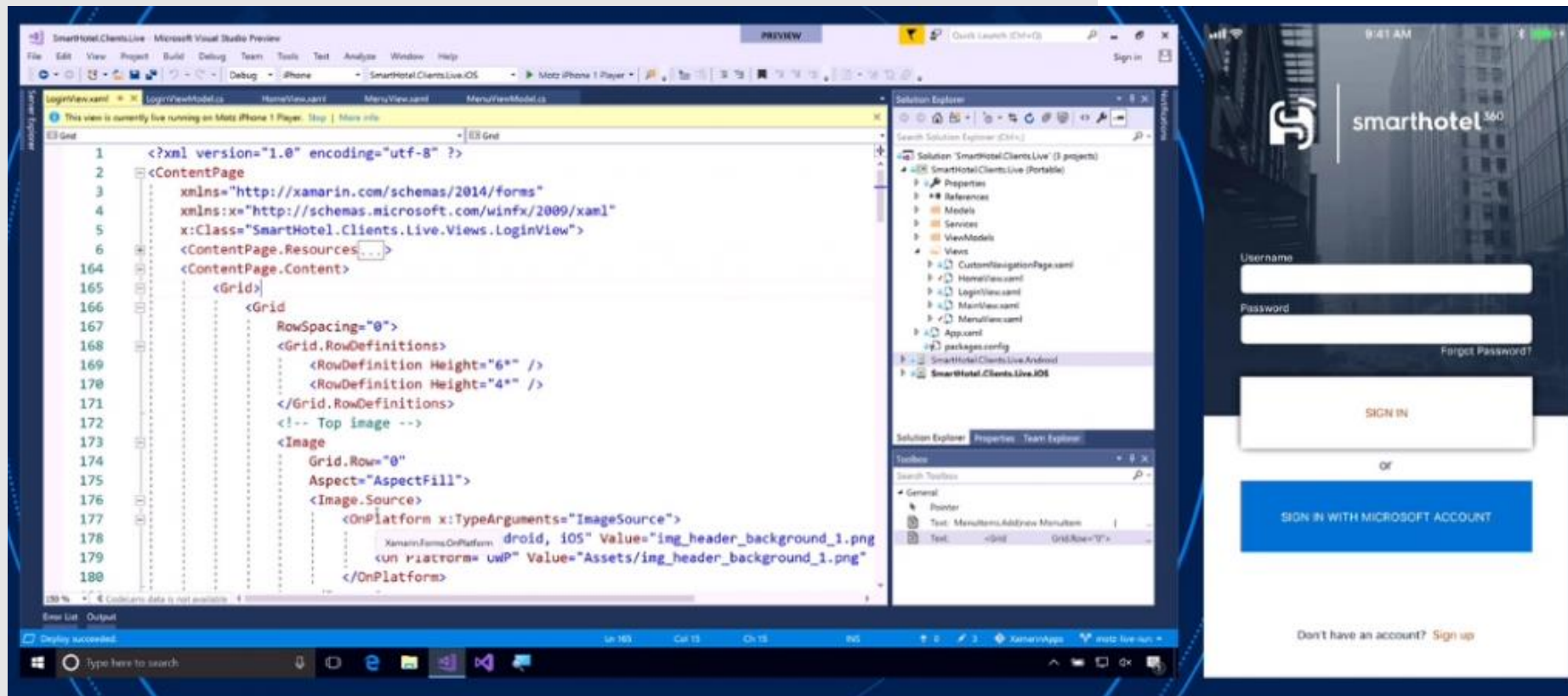
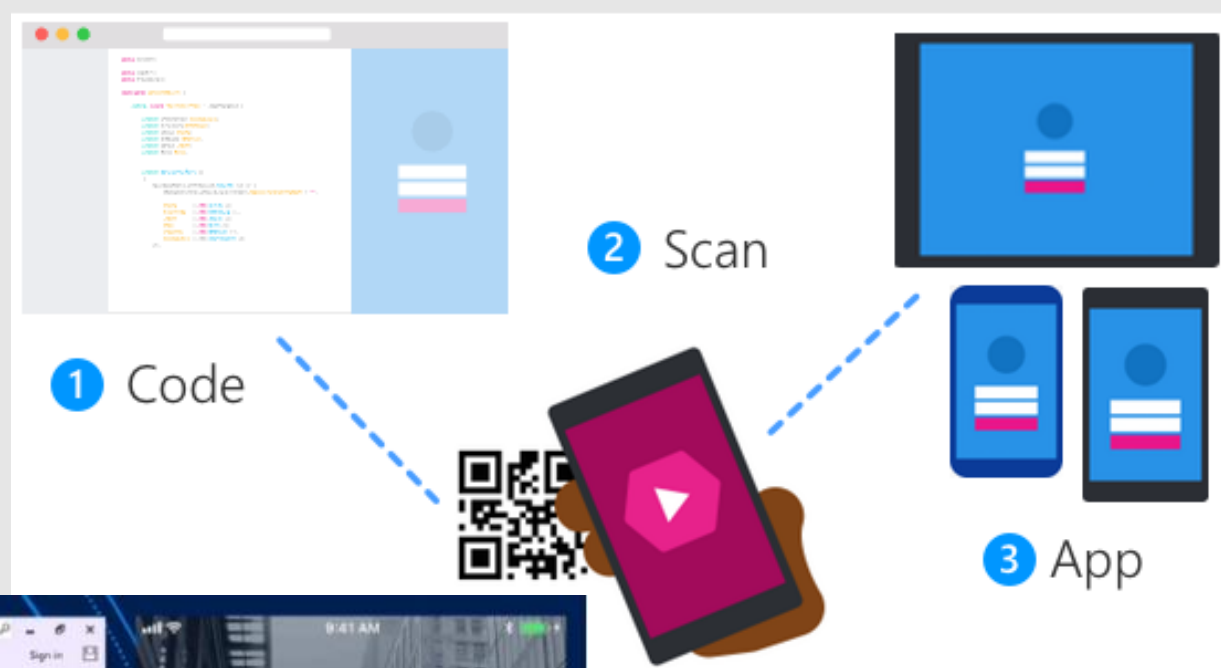
Узнай Москву
От Правительство Москвы



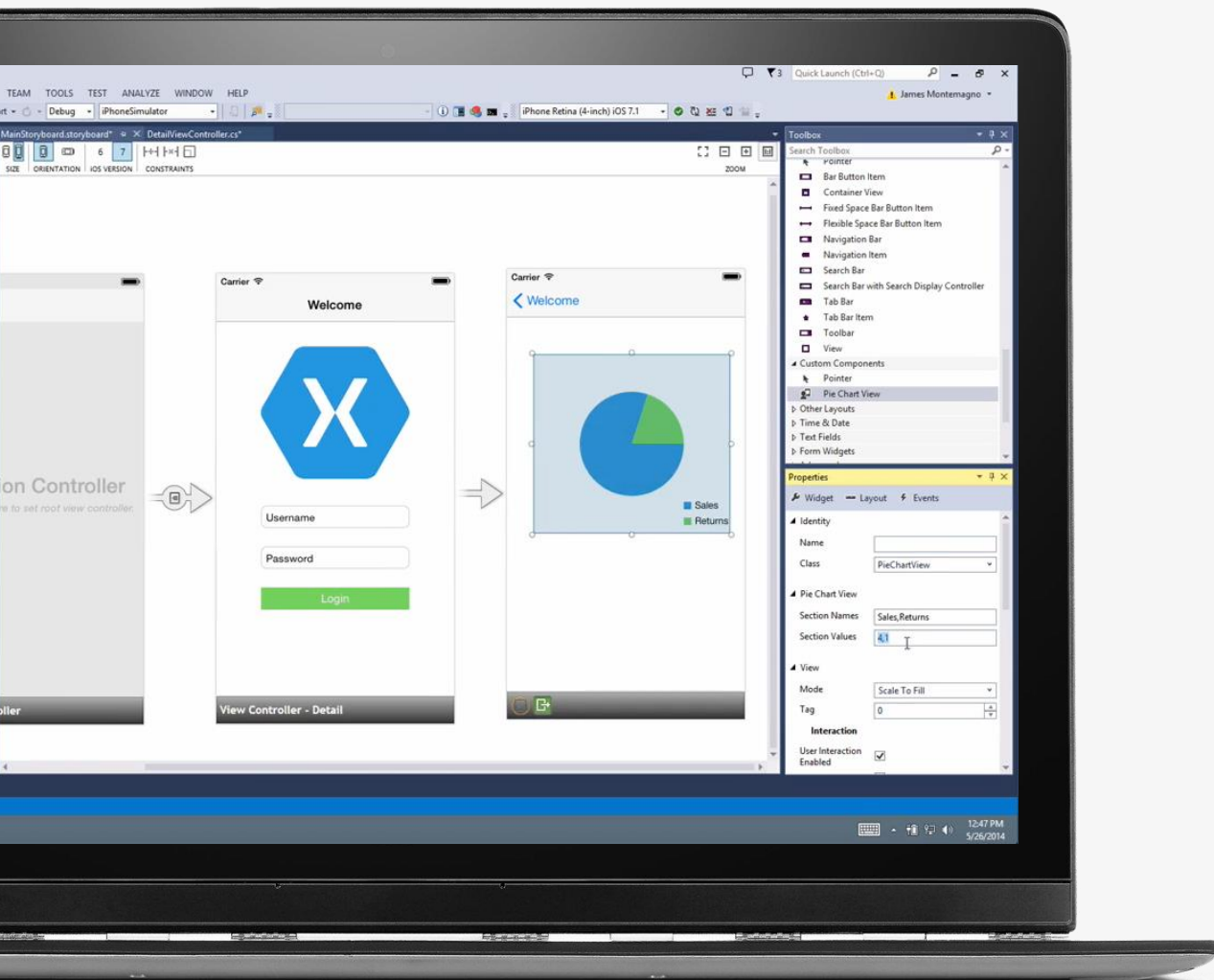
ИНТАЧ Страхование

Xamarin Live Player

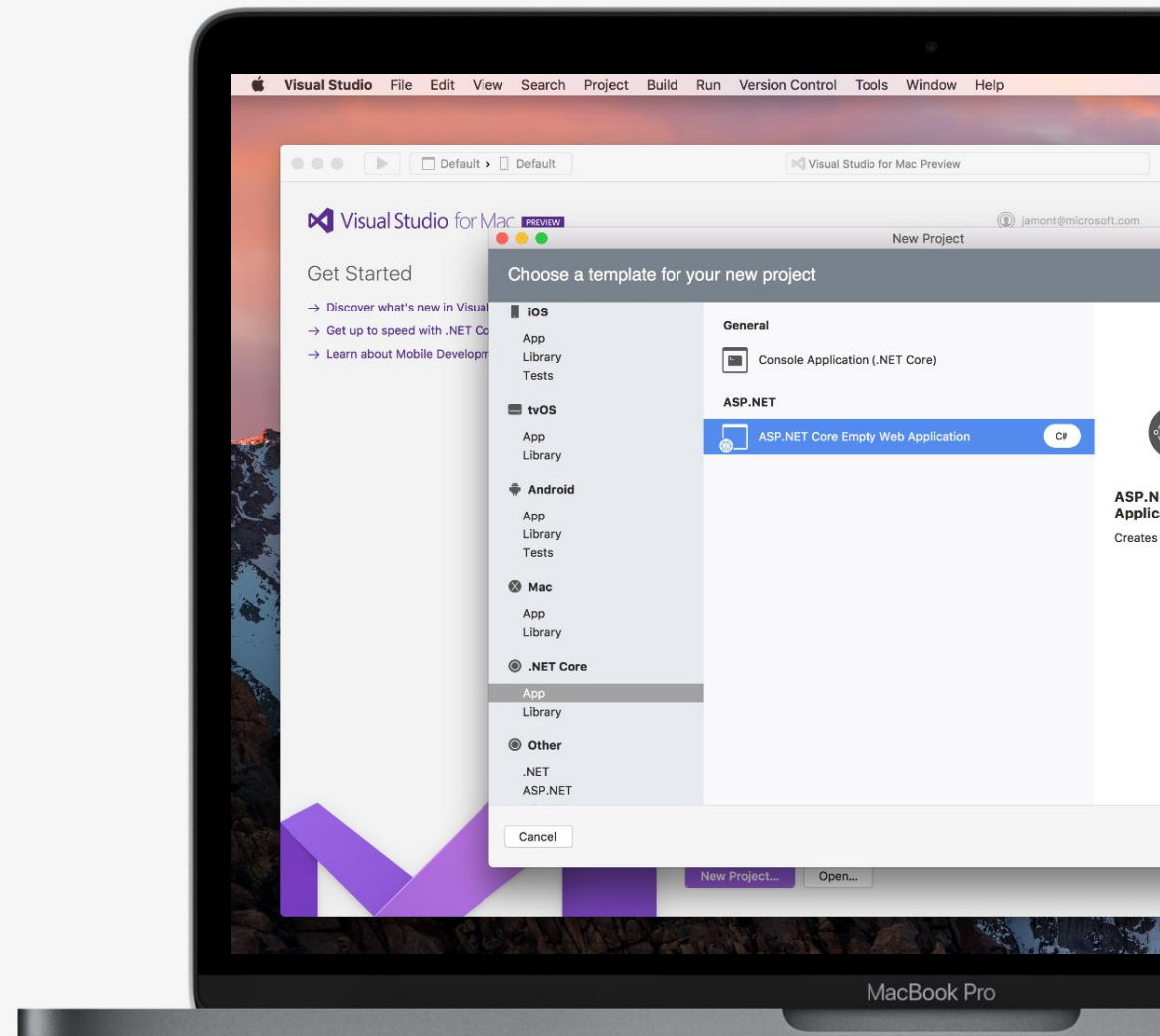
Разработка приложения «ВЖИВУЮ»



Visual Studio



Visual Studio for Mac



Трудности жизненного цикла разработки

Знакомые ситуации

- Сроки завершения разработки срываются
- Требования меняются во время разработки
- Сложно и долго находить и устранять баги
- При изменении функционала проявляются старые баги
- Невозможно протестировать приложения на большом количестве разных конфигураций



HockeyApp

Дистрибуция • Поломки • Аналитика



Xamarin Test Cloud

Тестирование на устройствах



Xamarin Insights

Поломки • Аналитика



Visual Studio App Center



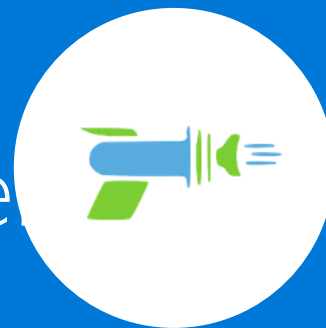
Azure Mobile Engagement

Аналитика • Нотификации



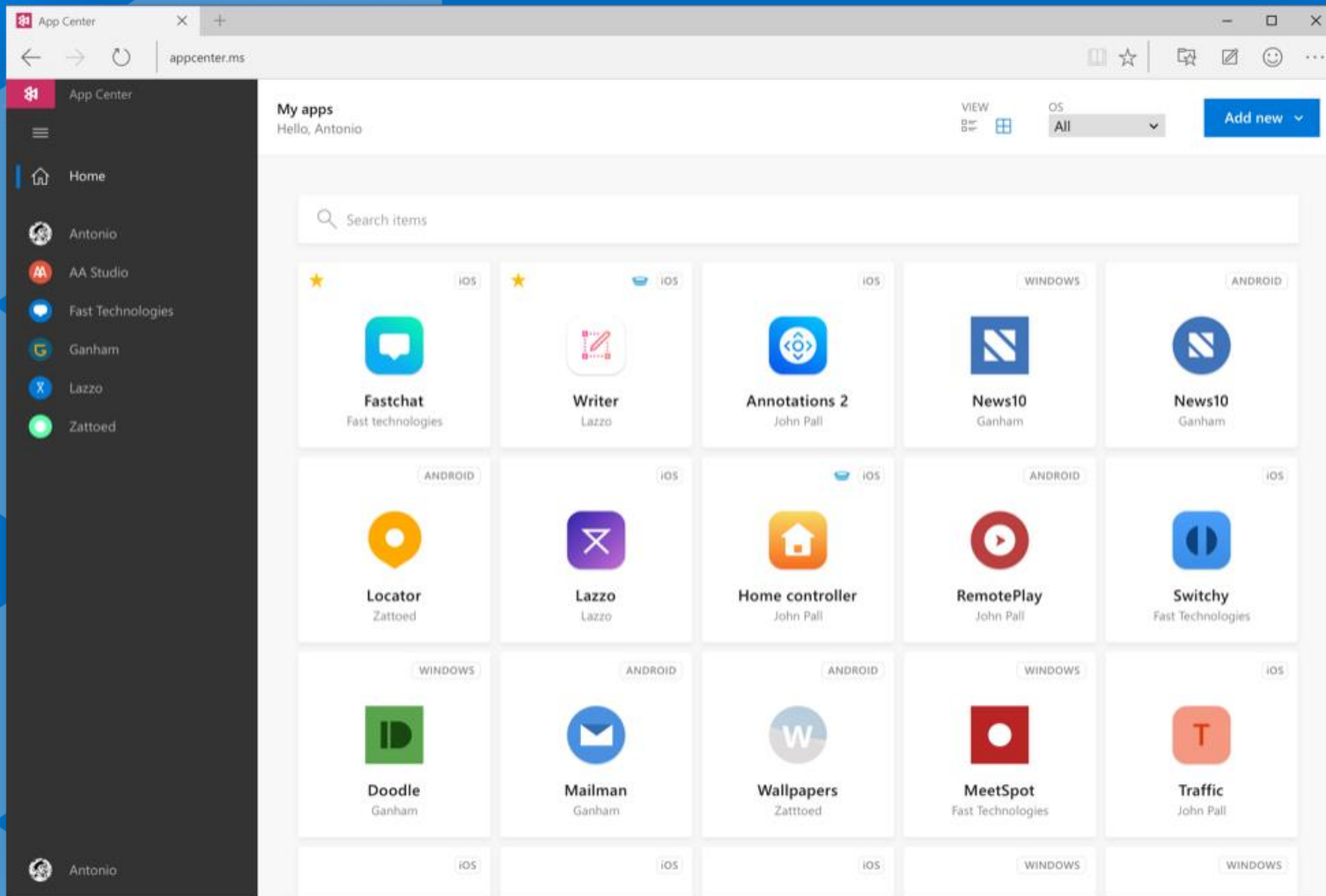
Azure App Service

Таблицы • Аутентификация
• Нотификации



CodePush

Дистрибуция



Visual Studio App Center

appcenter.ms

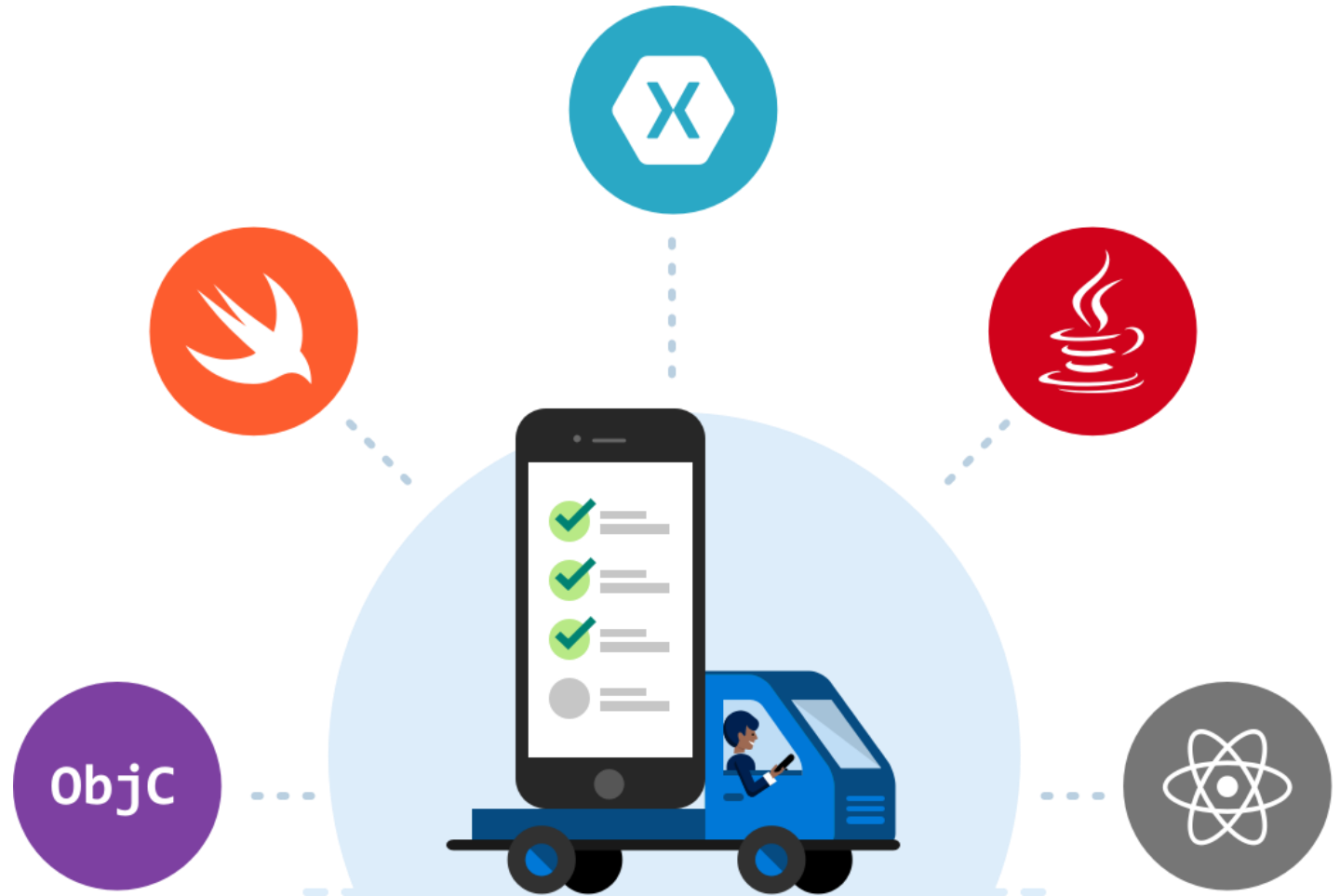
Поддержка основных современных платформ

Платформы

- UWP - Windows
- Android
- iOS
- macOS

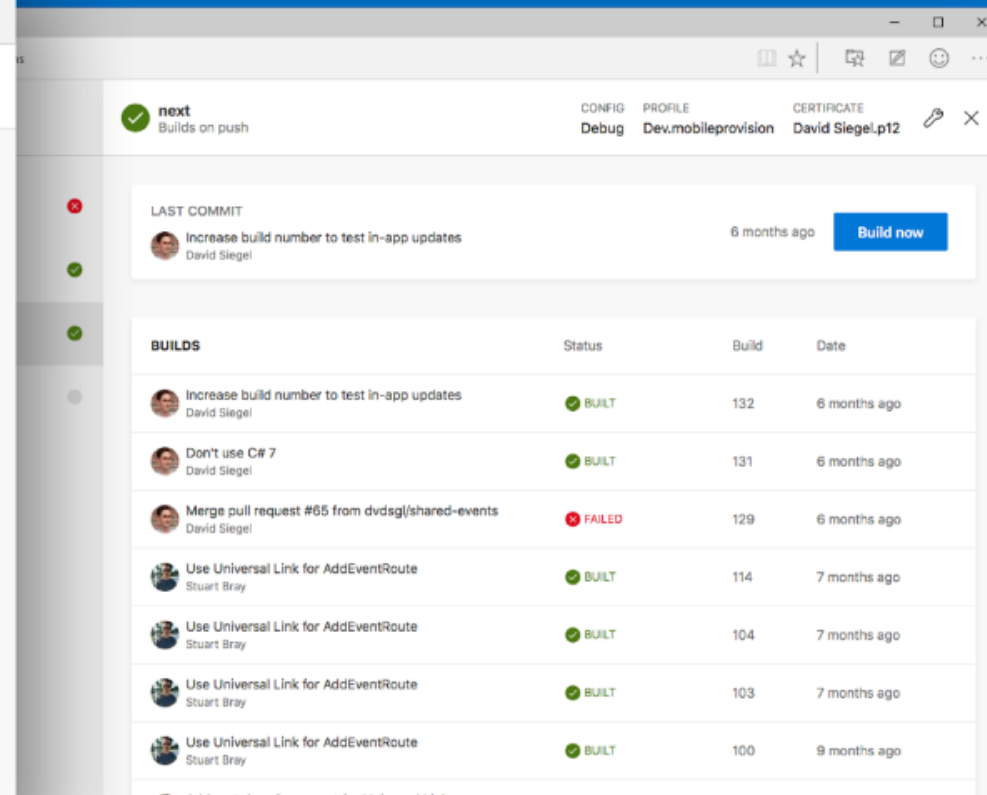
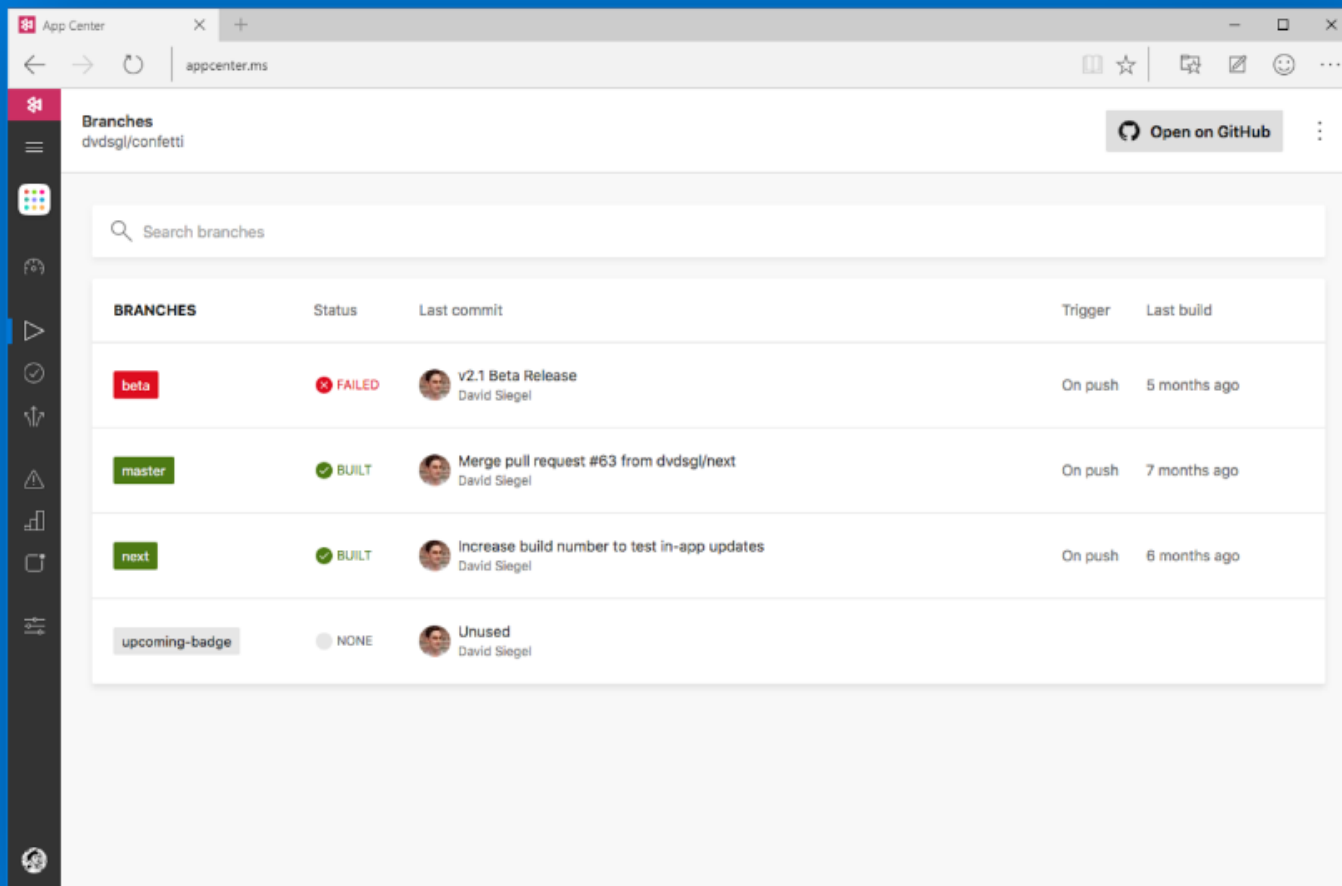
Среды и фреймворки

- Xamarin - .NET
- Java
- Obj-C/Swift
- React Native



Visual Studio App Center – быстрая сборка приложений

Source Code: VSTS, GitHub, Bitbucket



Результат - готовность новых версий в разы быстрее



Build



Test



Distribute



Crashes

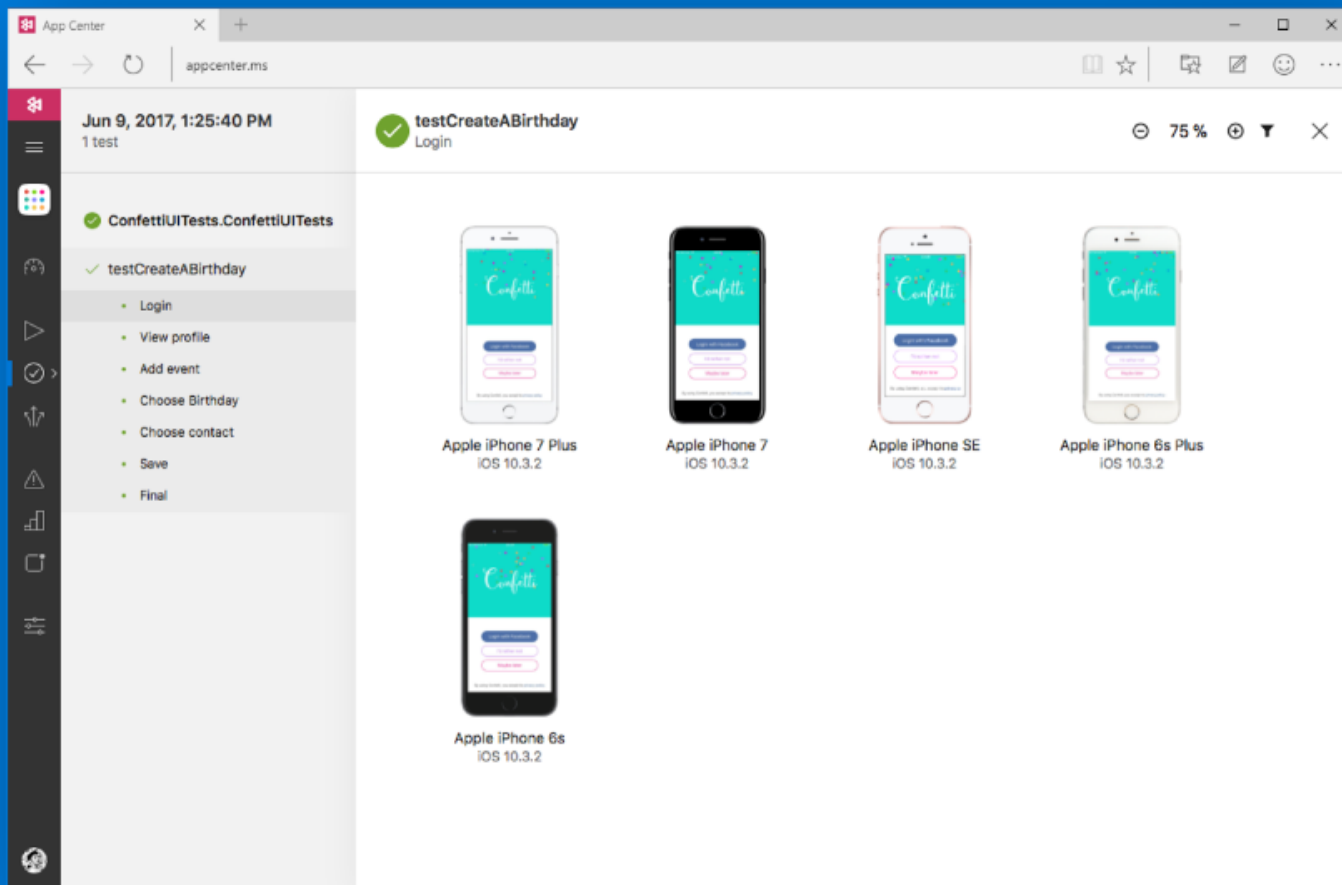


Analytics

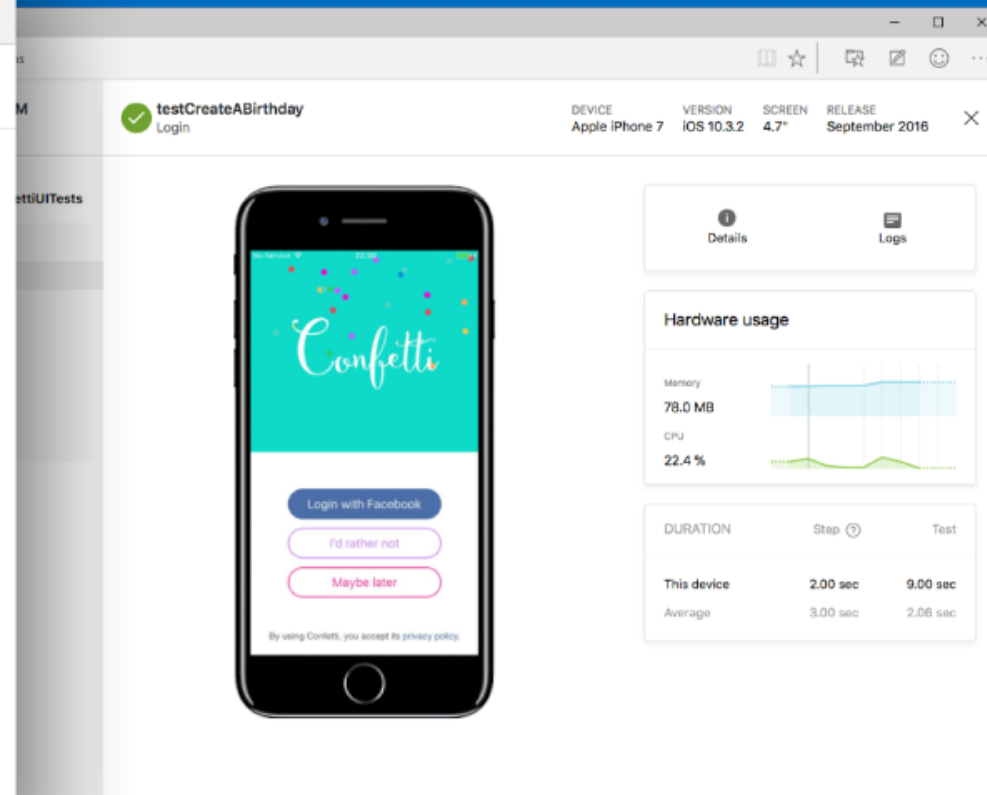


Push

Visual Studio App Center – тестирование на устройствах сотен конфигураций



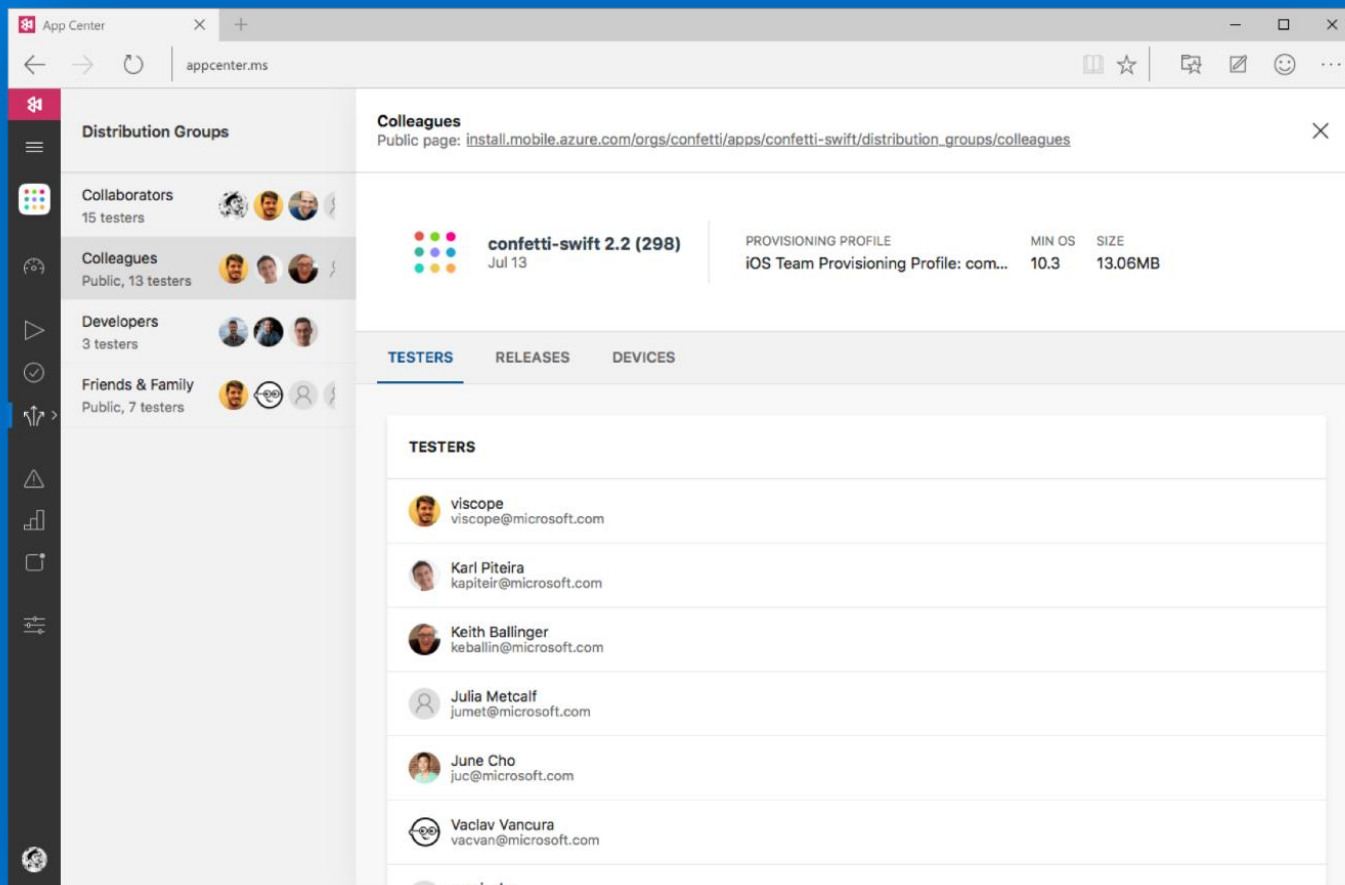
Test Frameworks: XCTest, Appium или нативные Espresso и XCUITest



Результат – меньше ошибок и жалоб пользователей

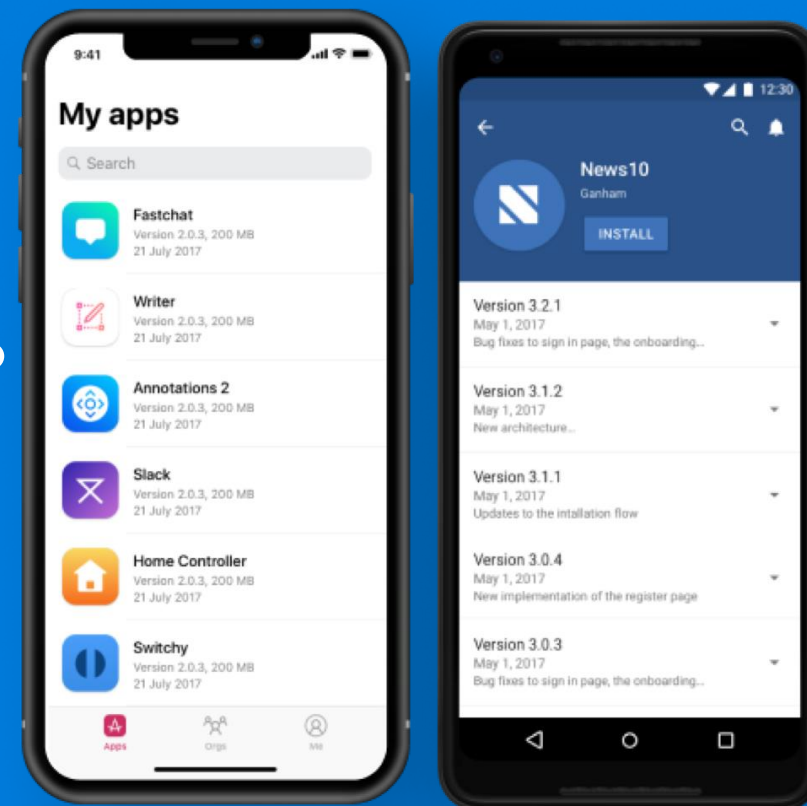


Visual Studio App Center – дистрибуция бета-тестерам и в магазины



HockeyApp

iOS, Android, UWP



Результат – меньше ошибок и жалоб пользователей



Build



Test



Distribute



Crashes

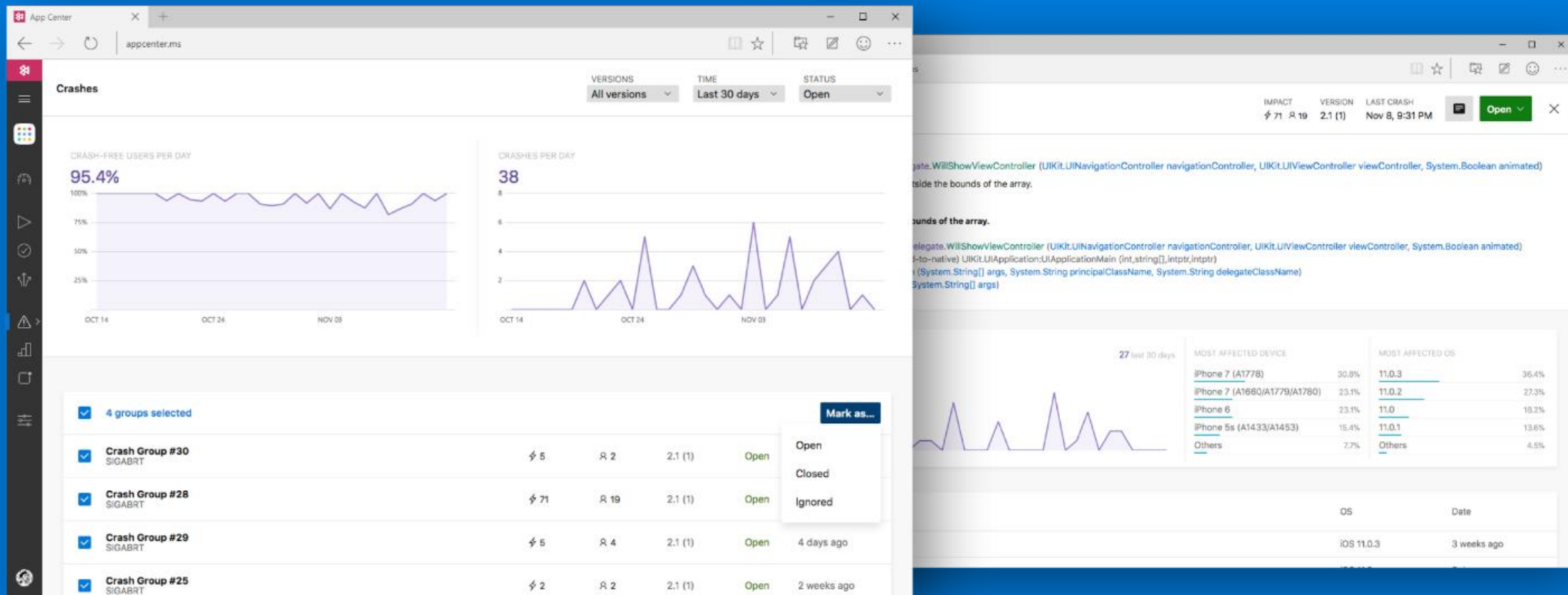


Analytics



Push

Visual Studio App Center – анализ аварийных завершений приложений



Результат – быстрее обнаружение и исправление ошибок



Build



Test



Distribute



Crashes

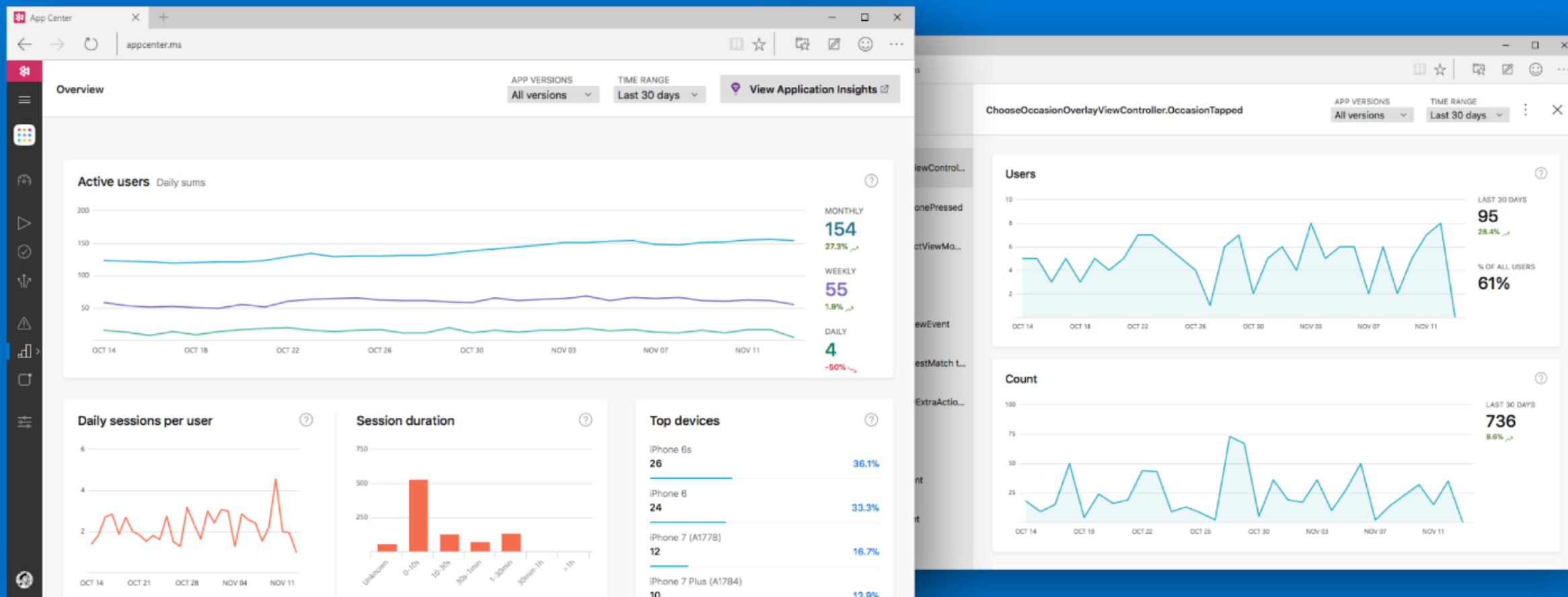


Analytics



Push

Visual Studio App Center – аналитика использования



Результат – понимание поведения пользователей и востребованности функционала



Build



Test



Distribute



Crashes

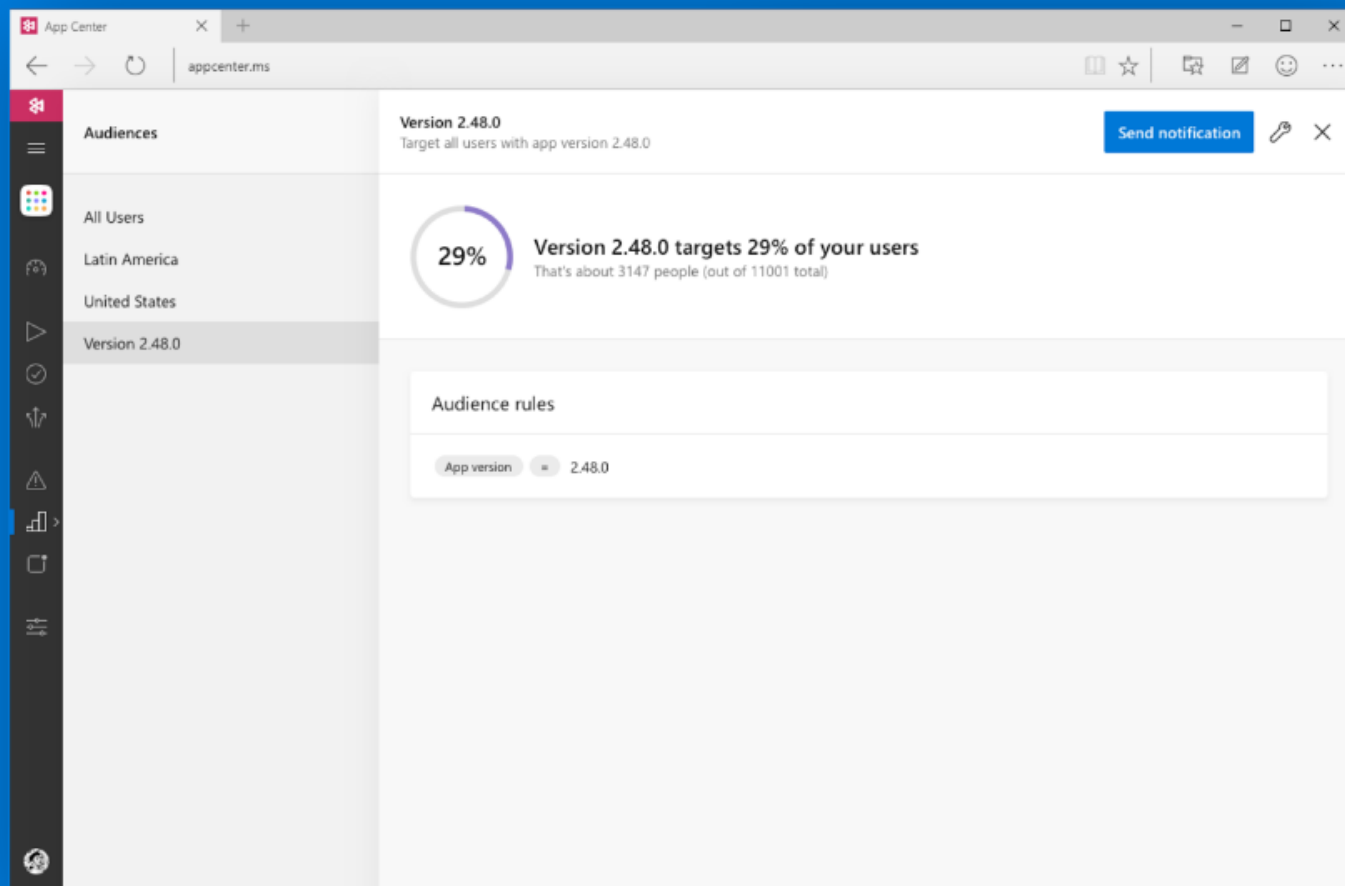


Analytics



Push

Visual Studio App Center – адресные коммуникации пользователям



Target	Results	Date
All devices	1651/2346 sent	Jan 7, 4:42 PM
All devices	1651/2346 sent	Jan 7, 4:30 PM
All devices	1947/2346 sent	Jan 6, 2:05 AM
All devices	1193/2346 sent	Jan 2, 7:10 PM



Результат – эффективность маркетинга выше, стоимость ниже



Build



Test



Distribute



Crashes



Analytics



Push



Highrise



Трудности создания масштабируемого backend для мобильных приложений

Знакомые ситуации

- Долго и дорого создавать инфраструктуру для приложений – большие задержки с запуском
- Большие затраты времени и денег на её обслуживание
- При удачном запуске приложения - сервера не справляются с нагрузкой
- При неудачном запуске приложения – сервера простаивают
- При скачущей нагрузке – обе проблемы вместе
- Инциденты с безопасностью данных ...

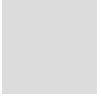
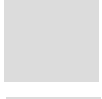
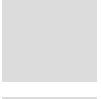
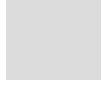
Ответственность и масштабирование

On-premise – хостинг у себя/в собственном ЦОД

IaaS – использование виртуальных машин

PaaS - использование платформенных сервисов для запуска логики или хранения данных

SaaS – использование готового сервиса, API или пользовательского интерфейса.

Ответственность	On-Prem	IaaS	PaaS	SaaS
Приложения				
Данные				
Runtime				
Middleware				
ОС				
Виртуализация				
Серверы				
Хранилище				
Сеть				

 Разработчик

 Microsoft

Azure – открытое облако

DevOps



Управление



Приложения



App frameworks и инструменты



Базы данных и middleware



Инфраструктура



Глобальное покрытие Azure

больше всех крупнейших облачных провайдеров

54 regions worldwide **140** available in 140 countries



* Two Azure Government Secret region locations undisclosed

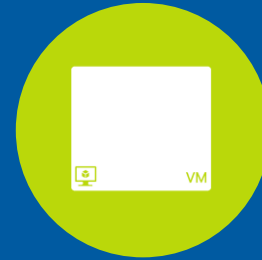
<https://azure.microsoft.com/en-us/regions/>

Infrastructure (IaaS) масштабирование ручное/авто



Виртуальные машины

Образы Windows Server и Linux ОС



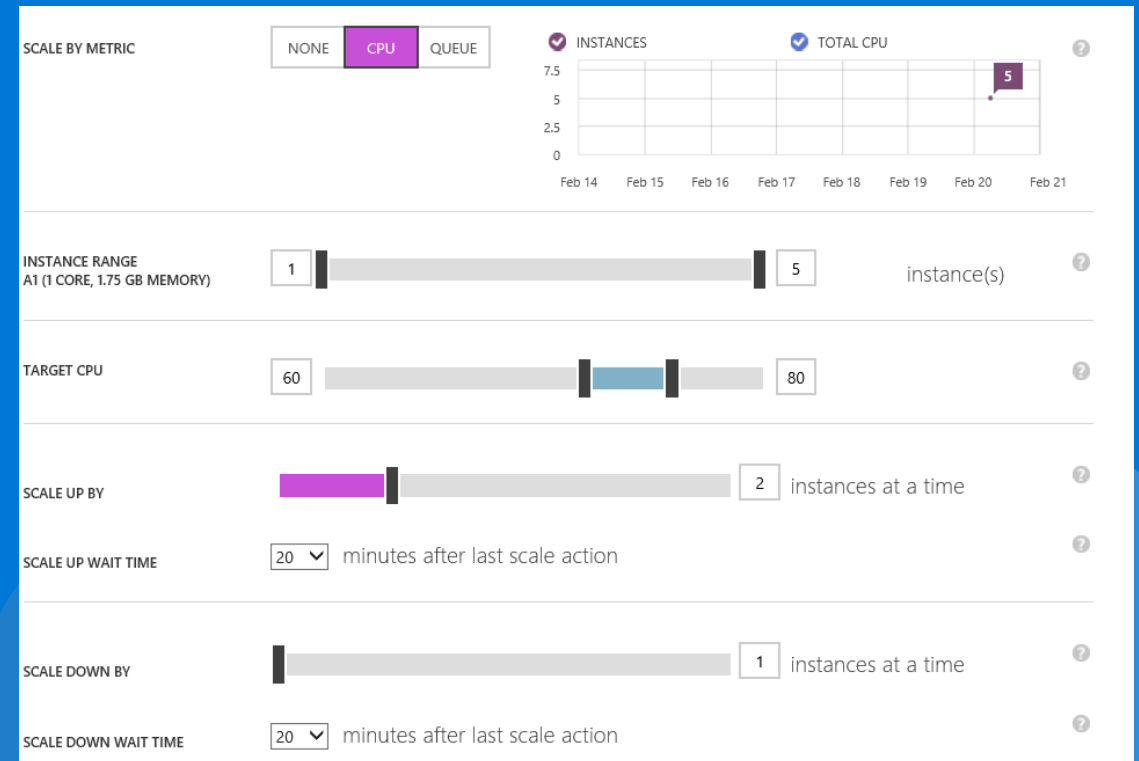
Virtual Machine ScaleSets

Механизм масштабирования stateless VMs с помощью ползунка/настройки



Container Service

Оркестрирование Docker-образов в рабочей среде



Контейнеры в Azure

Поддержка
Docker для Azure

VMs



РaaS на базе
контейнеров



Service
Fabric



CLOUD
FOUNDRY™



OPENSIFT

Azure
Container Service



DC/OS



docker

Управление
контейнерами



docker



DC/OS



kubernetes



CoreOS



RANCHER

Microsoft Azure

Azure Application Platform – бесконечное масштабирование

Создание логики на платформенных сервисах (PaaS)



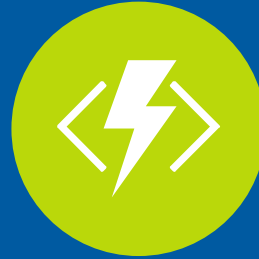
App Service

Web, mobile, API,
и приложения логики



Service Fabric

Облачные
микросервисные
приложения



Functions

Бессерверные функции на
базе событий



Cloud Services

Собственные монолитные
3-уровневые stateless
приложения



Другие PaaS

Cloud Foundry, OpenShift,
Apprenda, Jetlastic, etc.

Web и mobile

Microservices
Микросервисы

Serverless
Compute
Бессерверные
вычисления

Существующие
Frameworks

Third-party
Frameworks

Azure PaaS сервисы хранения данных

Blobs

Масштабируемое
облачное объектное
хранилище (REST)

Tables

Авто-
масштабируемое
NoSQL хранилище

Queues

Надежная
масштабируемая
доставка сообщений

Disks

Постоянные диски
для Azure IaaS
VMs"

Files

"SMB доступ к
Azure Storage"

Доступно в Windows & Linux VMs

Data Lake

Big Data
аналитика с
поддержкой
HDFS

Cosmos DB

NoSQL БД
планетарного
масштаба с
MongoDB API

Service Bus

Надежные
сообщения в
облаке как сервис

Redis cache

"High throughput,
consistent low-
latency data access"

Azure Search

Облачное
индексирование и
мгновенный поиск

Azure SQL DB + Instance

Управляемая
реляционная SQL
БД

SQL Data Warehouse

Эластичное
хранилище данных
как сервис

Azure DB MySQL

Управляемая
MySQL БД (preview)

Azure DB PostgreSQL

Управляемая
PostgreSQL БД
(preview)

Azure DB MariaDB

Управляемая
MariaDB БД
(preview)



Azure Cosmos DB

Key-Value



Column-family



Documents



Graph



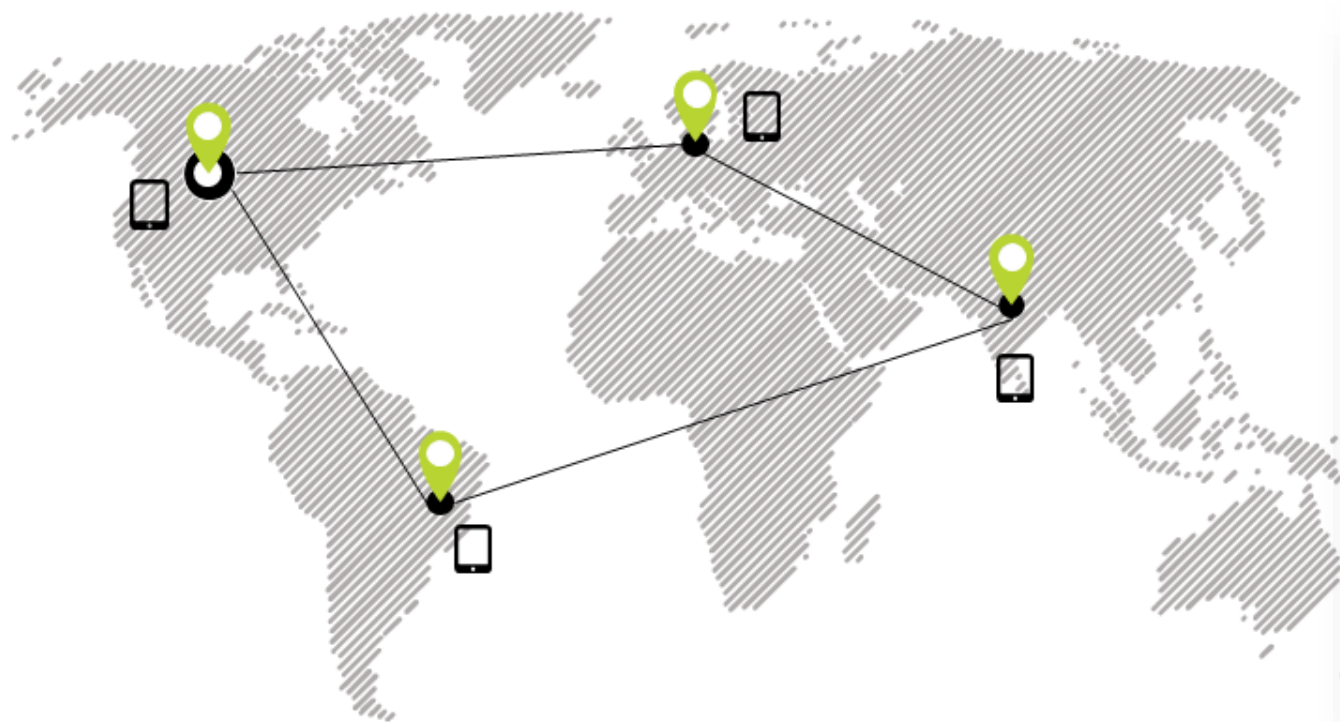
Global distribution

Elastic scale out

Guaranteed low latency

Five consistency models

Comprehensive SLAs

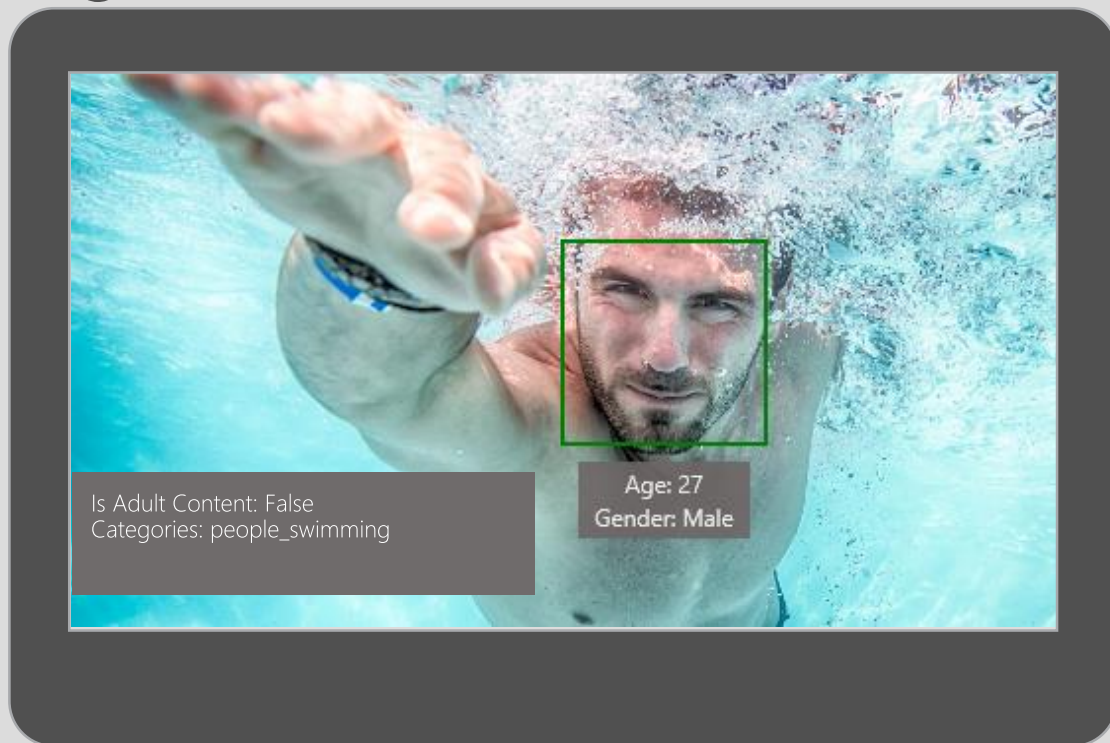


APIs: [MongoDB](#), [DocumentDB SQL](#), [Cassandra API](#) (preview), [Gremlin](#) (preview), and [Azure Table API](#)

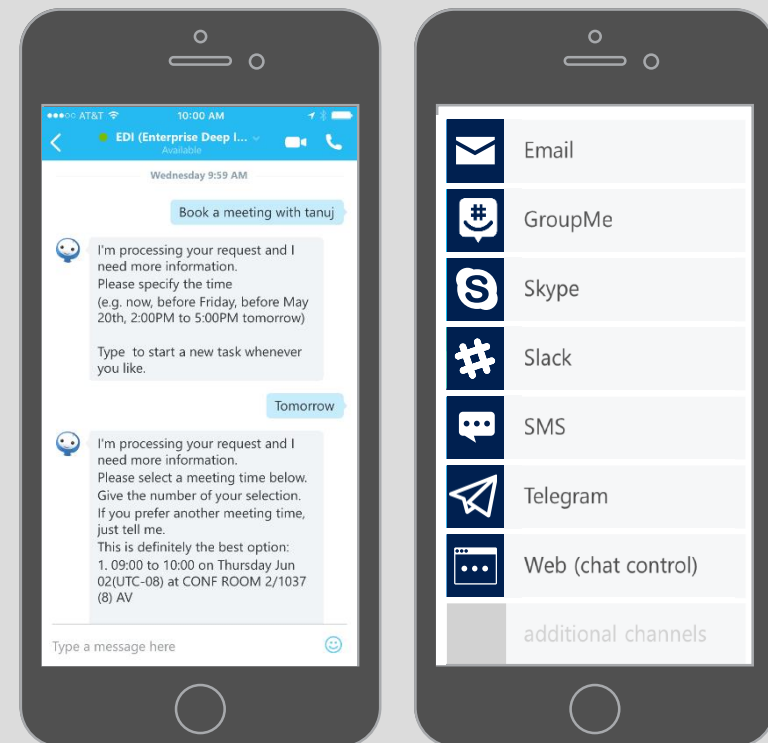
Использование искусственного интеллекта



Когнитивные сервисы Cognitive services



Bot framework

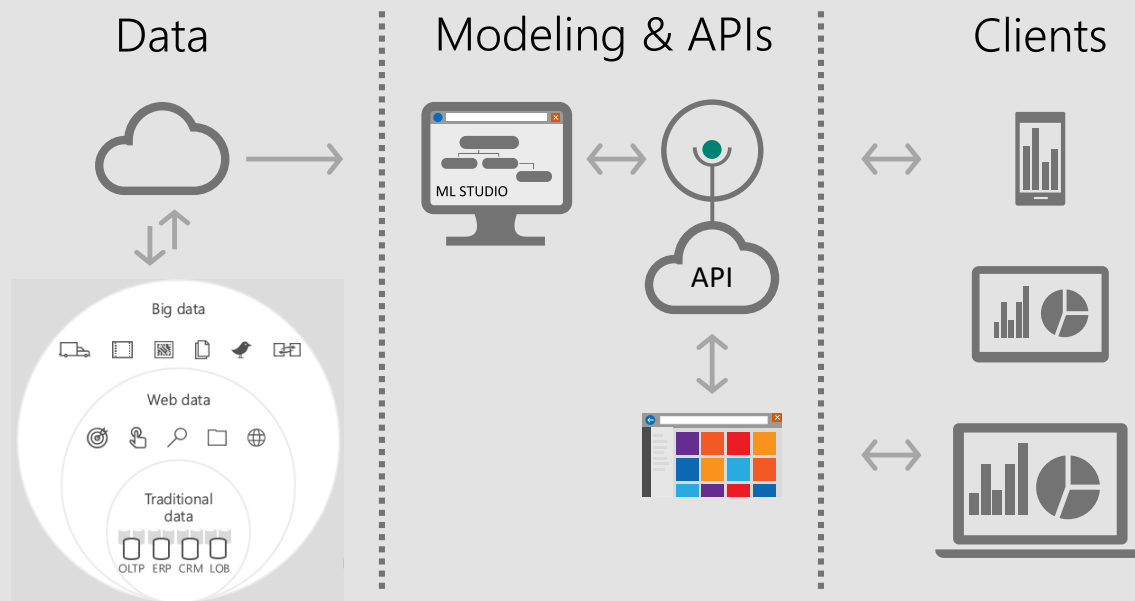


Принятие решений на базе данных



- Предсказание поведения
- Построение искусственного интеллекта (artificial intelligence)

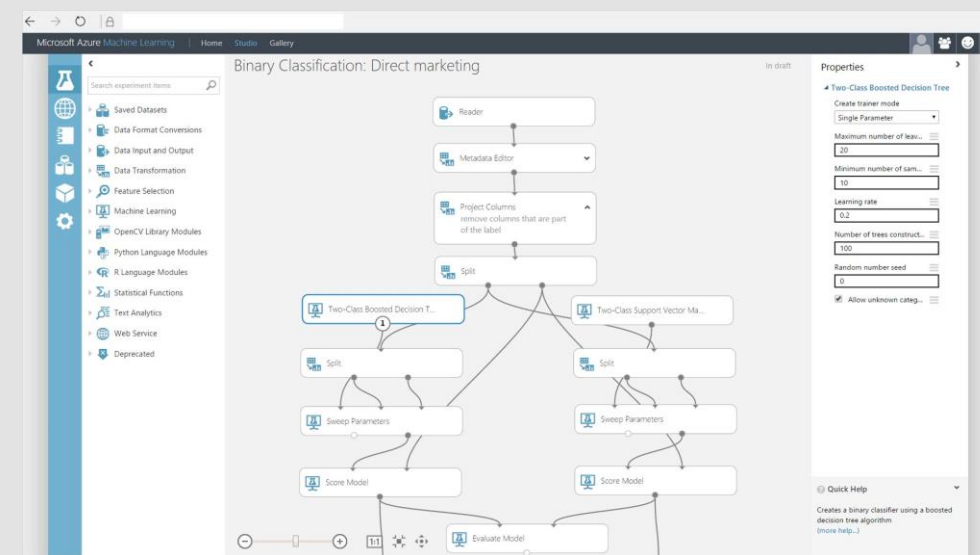
Azure Machine Learning service



Клиентская визуализация



Azure ML Studio



Как сделать быстрое мобильное приложение один раз для всех платформ

Ромуальд Здебский
Microsoft
rzdebski@microsoft.com