Xamarin Proces

CASE27 2015-06-02

Miljenko Cvjetko

[mcvjetko@holisticware.net](mailto:mcvjetko@holisticware.net)

[miljenko.cvjetko@xamarin.com](mailto:miljenko.cvjetko@xamarin.com)

<http://holisticware.net>

<http://xamarin.com>

[Xamarin danas 2015-05](#h.fqowrbklqt3u)

[Kronologija Xamarin Strategije](#h.wlgi9g3gqceo)

[Xamarin Proces](#h.po038rwi9iw0)

[Razvoj (Development) i integracija (Integration)](#h.1c4g6th68dz8)

[Xamarin.Profiler](#h.p2w6xgflv6gp)

[Faza Testiranja](#h.sm06sl5jxwux)

[Xamarin.UITest](#h.d97ocdo7yf9y)

[Calabash](#h.x6vfvx4gq3k4)

[Koncepti testiranja](#h.jxs4aryogfpp)

[Priprema Testova](#h.72fi7s82octw)

[Xamarin.Test.Cloud](#h.7r124x41mngr)

[Screenshots](#h.uq1yyfkic4sk)

[Error reporting](#h.9e556ngmastz)

[Device DrillDown](#h.uxxq7wi0kv8d)

[Monitoring - Xamarin.Insights](#h.15hj95we4jb2)

[Zaključak](#h.caim2eiy8a7e)

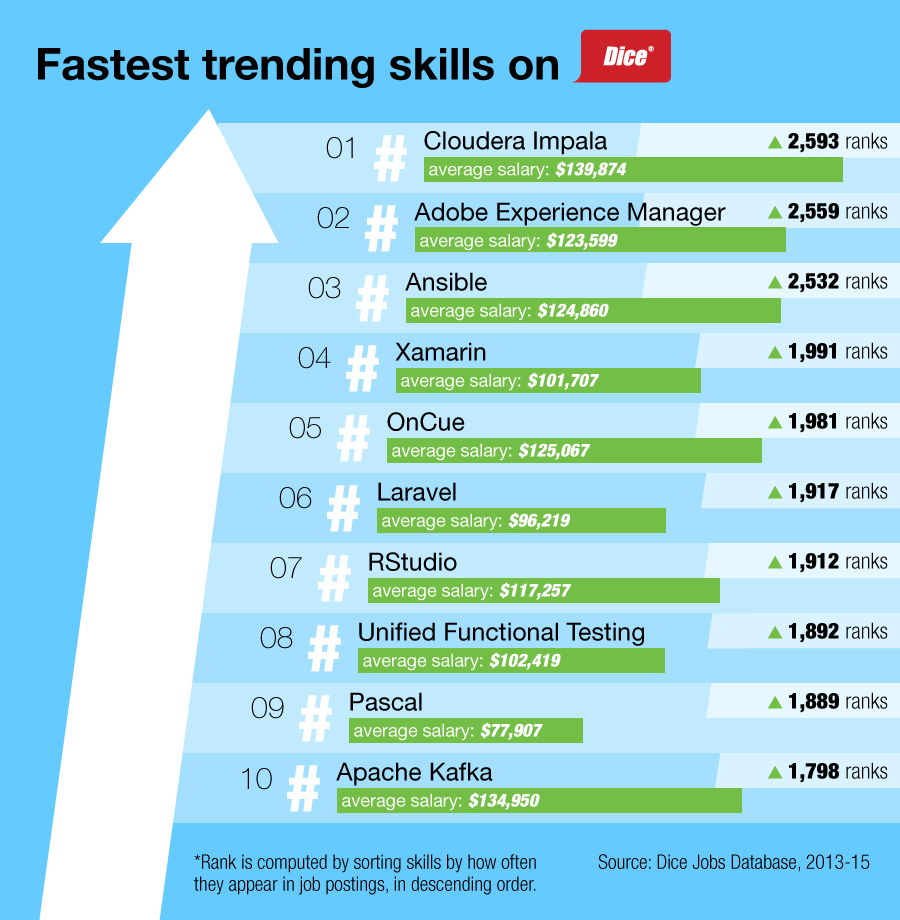
# 

# 

# Xamarin danas 2015-05

Xamarin tehnologije dobijaju sve veći i veći zamah među programerima, naročito onima koji dolaze iz .net svijeta. U poslijednjih nekoliko mjeseci došle su i potvrde iz nezavisnih izvora o performansama Xamarin aplikacija.

Potražnja za Xamarin vještinama raste iz mjeseca u mjesec i trenutaćno se nalazi na 4. mjestu po Dice portalu.

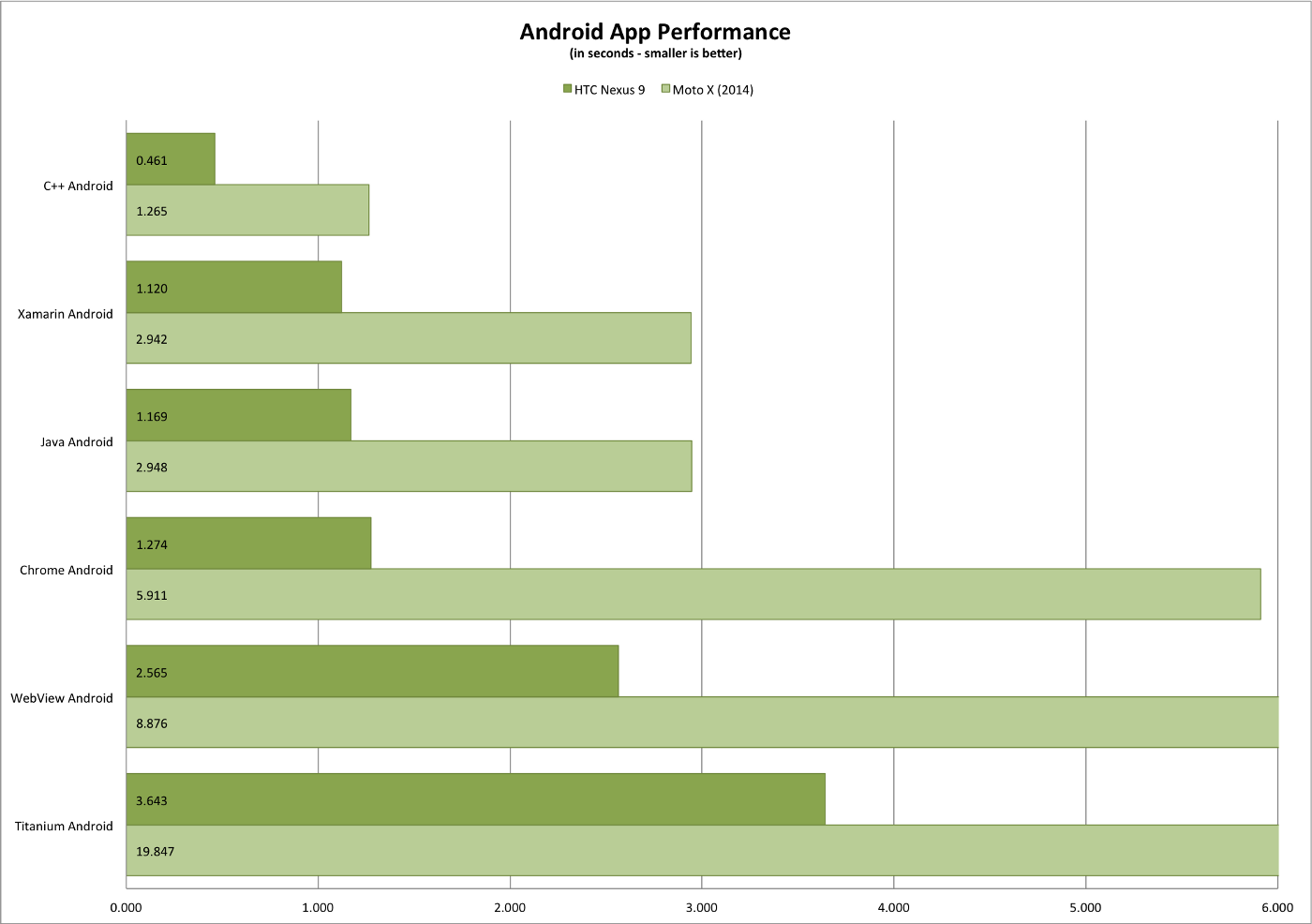


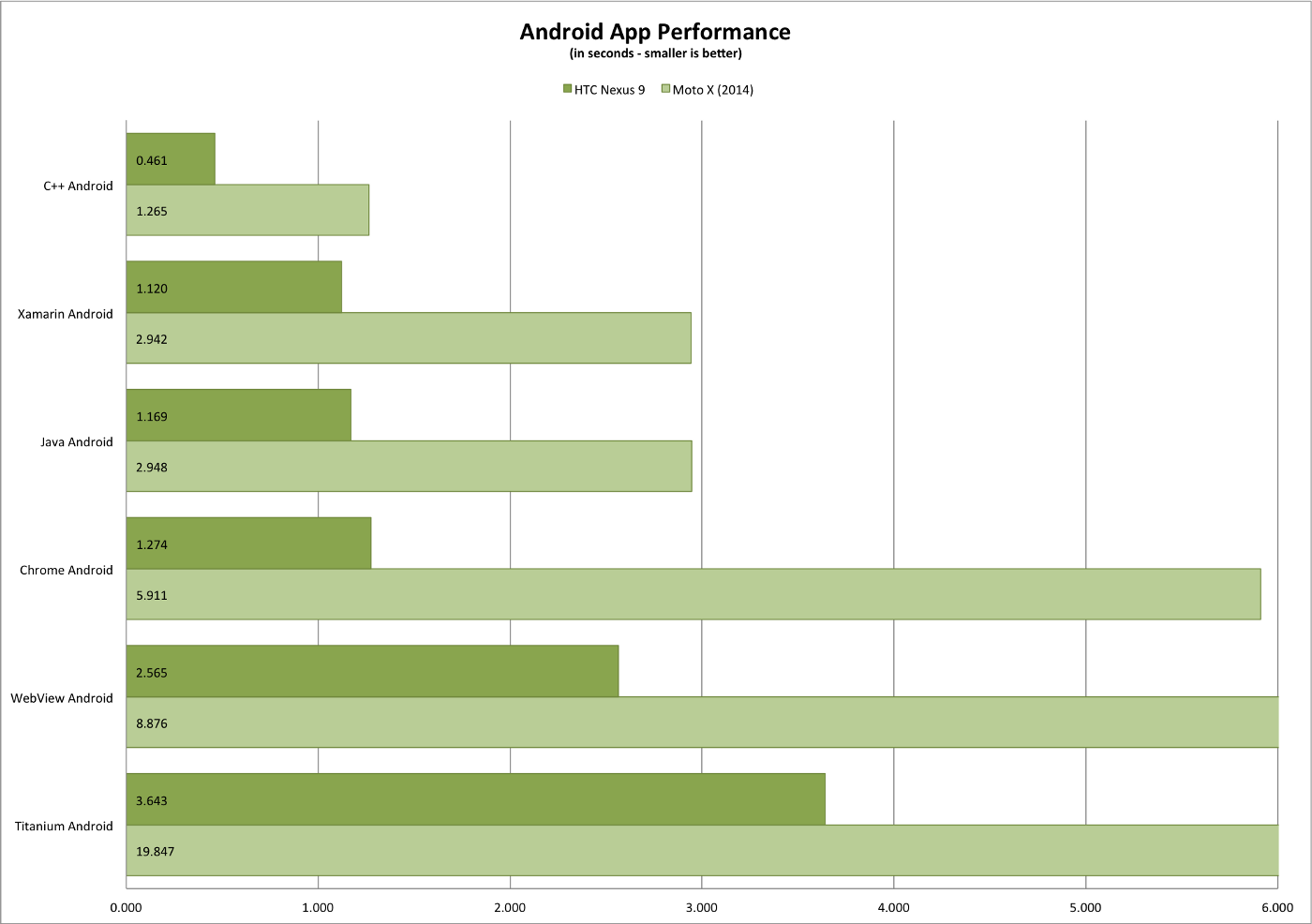
Harry Chenug - bivši google developer (jedan od prvih) napravio je jedan od prvih i dubljih benchmark testova za alate za razvoj mobilnih aplikacija.

<https://medium.com/@harrycheung/cross-platform-mobile-performance-testing-d0454f5cd4e9>

<https://medium.com/@harrycheung/mobile-app-performance-redux-e512be94f976>

Po tim analizama Xamarin tehnologije ravnopravno se bore sa “nativnim” tehnologijama (java i objective-c), a u nekima su čak i bolje.

Android



iOS

# 

# 

# 

# Kronologija Xamarin Strategije

U razdoblju od 2010.-2013. strategija Xamarin-a je bila razvoj alata za višeplatformski razvoj aplikacija, sa naglaskom na mobilne aplikacije. U tom razdoblju radilo se prvenstveno na alatima koji povezuju (toolchain) mono-.net svijet sa nativnim Android i iOS ekosistemima.  
U periodu od 2013.-2014. izdane su nove tehnologije koje su bile usmjerene na povečanju produktivnosti i smanjenju vremena izdavanja aplikacija (time to market), kao što su Xamarin.Forms. Ideja iza Xamarin.Forms-a je bila i ta da se tehnologije prisutne u .net ekosustavu dovedu i u Android i iOS svijet i time olakša .net i c# programerima ulazak u nove tehnologije.

U 2014. godini Xamarin prebacuje strategiju na kvalitetu aplikacija, a i svojih proizvoda kroz neke nove proizvode, te kroz integraciju starih proizvoda u niz alata za višeplatformski razvoj.

U tom razdoblju naglasak je na konceptima testiranja:

* TDD (Test Driven Development) sa konceptima Unit Testing-a
* BDD (Behaviour Driven Development) sa konceptima User Interface odn. Acceptance Testing-a
* Monitoring

Analizom povratnih informacija od kupaca i korisnika došlo se je do zaključka da je kvaliteta aplikacija vrlo bitan faktor u uspjehu i monetizaciji, te su dodatne snage i resursi prebačeni na testiranje aplikacija, alata, te integraciju testnih alata u Xamarin ekosustav.

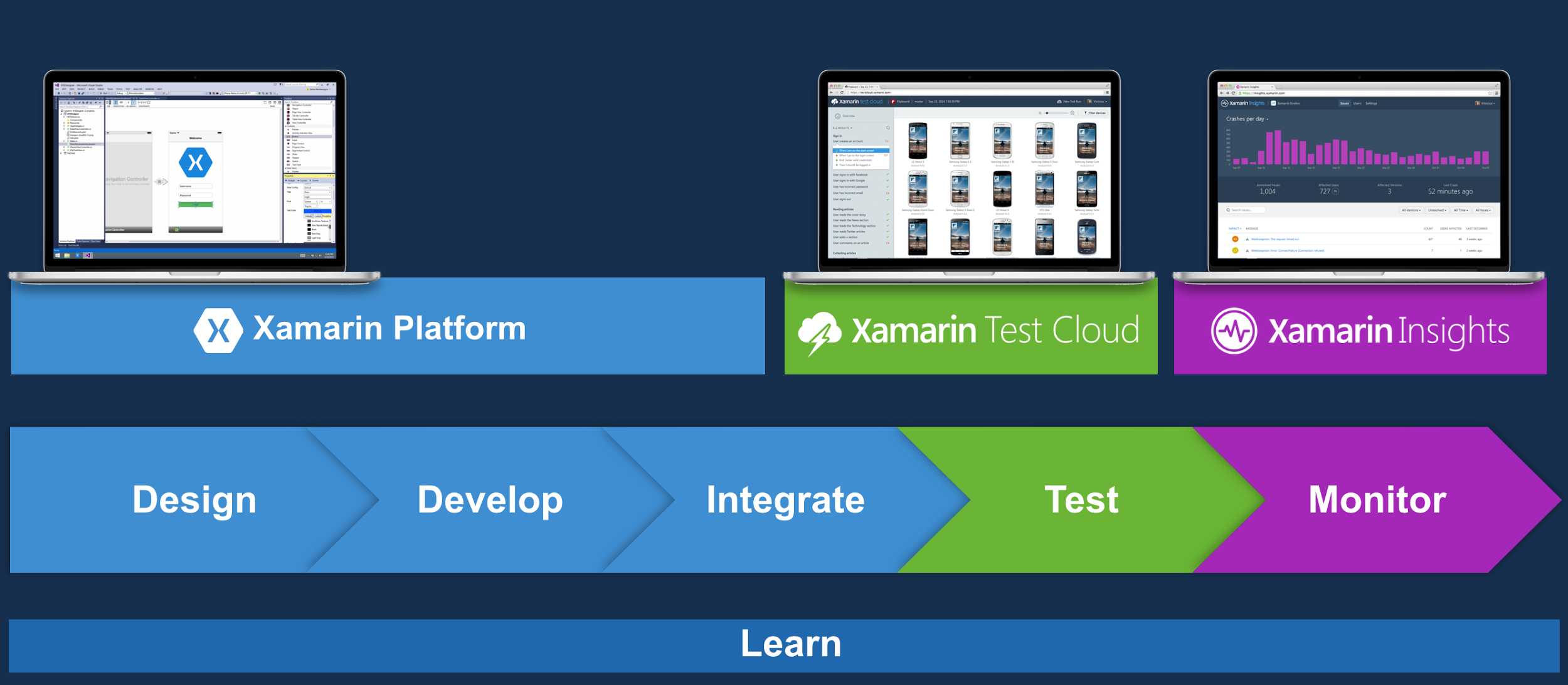
U jesen 2014. Xamarin team definira proces razvoja mobilnih višeplatformskih aplikacija.

# 

# Xamarin Proces

U Xamarin procesu definirano je 5 faza:

1. Design
2. Development
3. Integracija
4. Test
5. Monitoring



Faza dizajna omogućena je kroz Android i iOS Designer alate koji su integrirani u Xamarin Studio i Visual Studio. Ti alati omogućuju kreiranje nativnih korisničkih sučelja u alatima poznatim .net programerima, bez potrebe za dubljim poznavanjem Apple-ovog XCode-a za iOS, odn. alata Android Studio i Eclipse za Android razvoj.

Faza razvoja (development) pokrivena je već 5-6 godina kroz razvojne alate Xamarin.Android (nekada Mono for Android, odn. Monodroid), te Xamarin.iOS (MonoTouch). Ti alati donose mogućnost nativnog razvoja u c# i f# programskim jezicima koristeći .net i pristup 100% nativnim API-u platforme koju se targetira.

Faza integracije (Integrate) donosi alate za publiciranje aplikacija, te stavljanje aplikacija u proces beta testing-a i monitoring-a.

Testiranje kao faza postala je jedan od bitnijih koraka u procesu i ovdje su se pojavili mnogobrojni alati i usluge za:

* TDD alati bazirani na NUnit framework-u koji su prilagođeni pojedinim platformama i omogućuju testiranje Business Logic sloja aplikacije (Domain Logic). U te alate pripadaju Xamarin.Android NUniti Xamarin.iOS NUnit projekti.
* BDD alati koji se koriste za testiranje korinisčkog sučelja UI odn. Acceptance testing. Ovi alati su namjenjeni pokrivanju što većeg broja use-case slučajeva i eliminaciju manualnog testiranja koje su mnogi korisnici Xamarin ekosustava primjenjivali, a na njih se nadograđuju servisi u oblaku za test na urešajima.  
  U ove alate spadaju:
  + Xamarin.UITest kao alat (framework) za UI Acceptance testing na bazi Calabsh tehnologije.
  + Xamarin.Test.Cloud kao usluga u oblaku

Posljednja faza monitoring donosi alat Xamarin.Insights za praćenje:

* problema u aplikaciji (crash and issue reporting)
* korisničkih navika i interakcija (session and user monitoring)

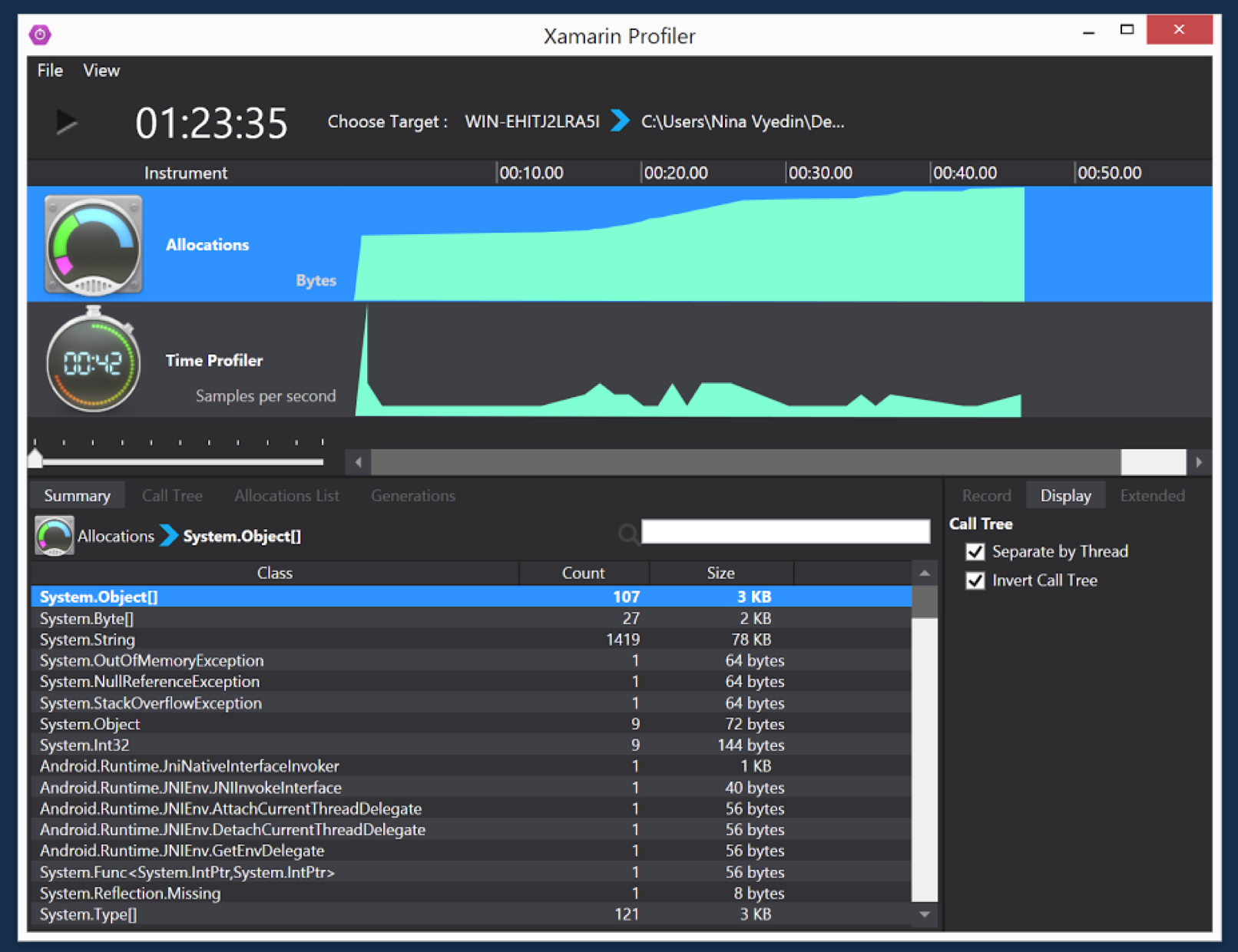
koji je napisan u potpunosti u .net-u, sa naglaskom na privatnost podataka i sigurnost, te omogućuje integraciju sa mnogim servisima (github, Visual Studio Online, HipChat, Campfire, Slack, jira ...).

# Razvoj (Development) i integracija (Integration)

U samom procesu razvoja nije dolazilo do velikih promjena, a naglasak je bio prvenstveno na podizanju kvalitete postojećih alata. Jedini novitet je Xamarin.Profiler.

## Xamarin.Profiler

Xamarin.Profiler je programerski alat koji pruža podršku u procesu razvoja i implementecaije aplikacija, a namjena mu je pružanja detaljnih informacja o resursima uređaja: memoriji i procesoru, što je vrlo bitno zbog karakteristika Xamarin tehnologija.



# Faza Testiranja

Fazi testiranja pružena je najveća pažnja i ovdje su se desili najveći noviteti i tehnološki napredak.

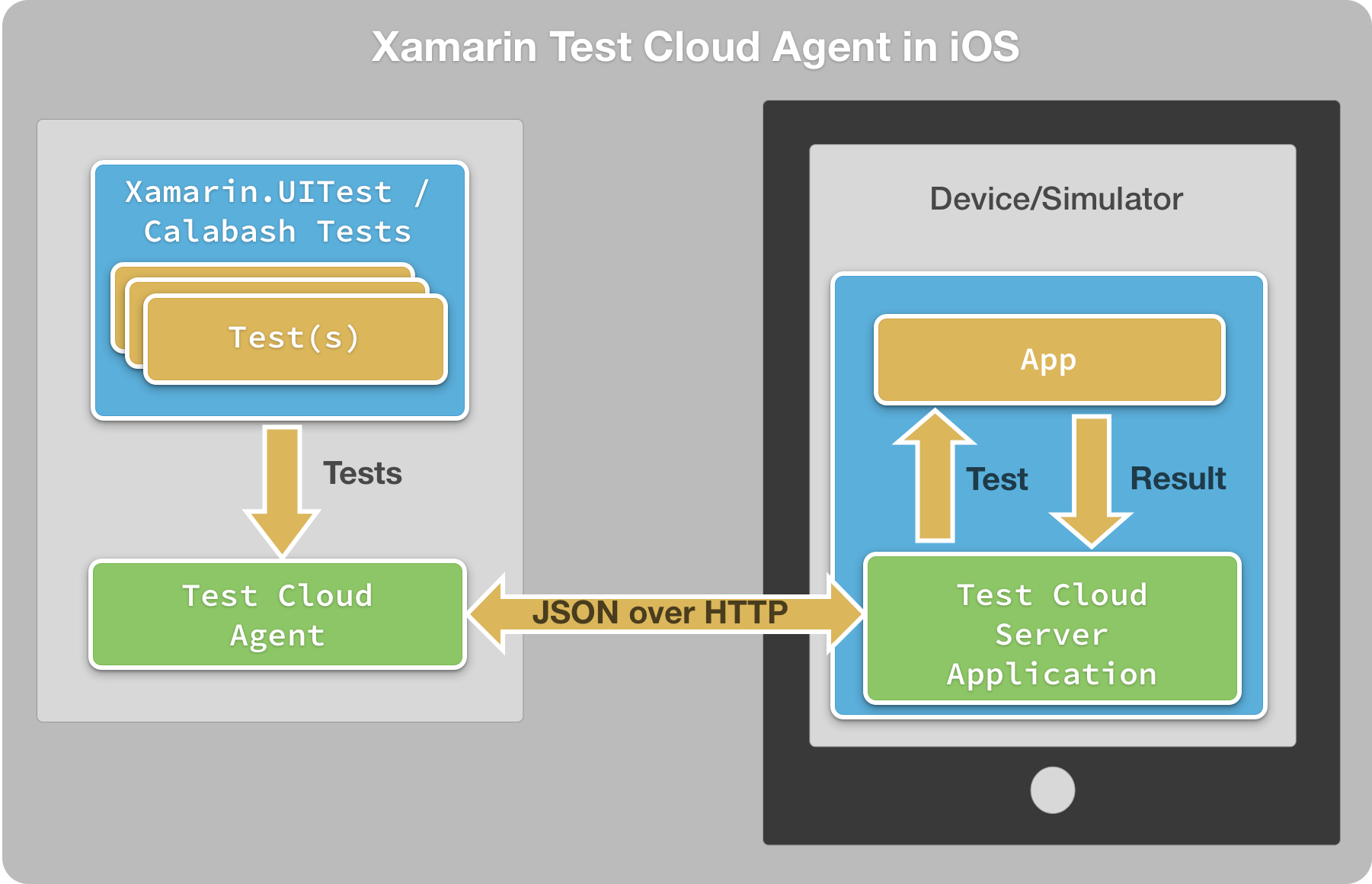
## Xamarin.UITest

Xamarin.UITest je automatizirani “UI Acceptance Testing framework” baziran na Calabasj tehnologiji i omogućava korisnicima izvršavanje NUnit testova pisanih u c# jeziku u cilju validiranja funkcionalnosti Android i iOS aplikacija.

Framework je usko povezan sa Xamarin.Android i Xamarin.iOS tehnologijama, no može biti korišten i za aplikacije pisane u objective-c za iOS i java za Android programskim jezicima.

Xamarin.UITest je biblioteka za automatizaciju interakcije sa korisničkim sučeljem za unos podataka, akcije na sučelju (events - button tap, itd.), geste (gestures - swipes itd.).

Xamarin.UITest oslanja se na Xamarin Test Cloud Agent, biblioteku koja sadrži HTTP server koji sa Calabash odn. Xamarin.UITest testovima komunicira preko JSON-a.i Sam agent ubacuje (injektira) se u testiranu aplikaciju za iOS.



Kod Android sustava agent se ne injektira u aplikaciju nego se instalira kao posebna serverska aplikacija koja na uređaju ili emulatoru komunicira sa testiranom aplikacijom.

## 

Instalacija na Androidu nije potrebna, dok za iOS instalra se preko NuGet paketa “Xamarin Test Cloud Agent”.

Testovi se mogu izvršavati lokalno na uređaju ili emulatoru, ili pak u Xamarin.Test.Cloud-u na 1400+ uređaja (2015-05).

### Calabash

Calabash framework je automatizirani UI Acceptance Testing sustav koji korisnicima omogućuje izvršavanje automatskih testova za iOS i Android. Ovaj framework uvodi koncepte Behavior Driven Development-a i vrlo je usko povezan sa Xamarin.Android i Xamarin.iOS proizvodima, no omogućava testiranje i aplikacija pisanih u objective-c i java za iOS i Android sustave.

Automatiziranost ovog framework-a omogućuje mu integraciju u CI (Continuous Integration) sustave.

Test Cloud Agent sadrži:

* Testove pisane u Gherkin-u (Feature i Step Definitions)  
  skupu gramatičkih pravila za specifikaciju testova  
  Feature: Credit card validation.  
  Credit card numbers must be exactly 16 characters.  
    
  Scenario: Credit card number is too short  
   Given I use the native keyboard to enter "123456" into text field number 1  
   And I touch the "Validate" button  
   Then I see the text "Credit card number is too short."  
    
  Scenario: Credit card number is too long  
   Given I try to validate a credit card number that is 17 characters long  
   Then I should see the error message "Credit card number is too long."
* Cucumber framework koji radi parsing i izvršavanje testova preko poziva API-a Calabash framework-a koji je pisan u Ruby-u
* Calabash framework koji izvršava testove

Calabash je open source projekt, no Xamarin ga je prilagodio .net svijetu i pretvorio u proizvod.

### Koncepti testiranja

Testovi se izovde u pattern-u Arrange-Act-Assert klasičnim za većinu testing frameworka, počevši od unit-testinga.

1. Arrange - priprema testa i uvjeta (podaci u korisničkom sučelju)
2. Act - simulacija korisničke interakcije (button press, swipe…)
3. Assert - provjera akcija koje su odrađene i stanja podataka kao i korisničkog sučelja.

Na višem nivou proces se opisuje kroz slijedeće korake:

1. razvoj aplikacije (Android ili iOS)
2. pisanje testova i izvršenje testova lokalno na emulatoru-simulatoru ili manjem broju uređaja
3. upload i izvršenje testova u Xamarin.Test.Cloud-u na velikom broju uređaja
4. ispravljanje grešaka
5. ponavljanje lokalni i Test.Cloud testova

### 

### Priprema Testova

Aplikacije su apstrahirane kroz klase

Xamarin.UITest.iOS.iOSApp

Xamarin.UITest.Android.AndroidApp

Ovi objekti su kreirani kroz ConfigureApp klasu koja instancira iOSApp i AndroidApp objekte i inicijalizira ih kroz NUnit metode dekorirane slijedećim atributima:

* SetUp  
  priprema logičke grupe testova (recimo na jednom ekranu)  
  Preporuka - odvojeni IApp objekti
* TestFixtureSetup  
  priprema sitnijih detalja za pojedine testove
* Test  
  izvršenje testa  
  app.Tap(c=>c.Button("ValidateButton"));

inicijalizacija:

string path\_apk1 = "/Users/nb/sample.iOS/bin/Debug/Sample.Droid.apk";

ConfigureApp

.Android

.ApkFile(path\_apk1)

.Debug()

.WaitTimes(new HolisticWare.XamarinUITest.WaitTimes())

//.ApiKey(api\_key)

.StartApp();

string path\_app1 = "/Users/nb/sample.iOS/bin/iPhoneSimulator/Debug/SampleXamarinFormsiOS.app";

ConfigureApp

.iOS

.AppBundle(path\_app1)

//.ApiKey("")

.StartApp();

Sam Xamarin.UITest framework ne kompajlira i pakira aplikacije, stoga to se mora prethodno napraviti.

IApp objekt korisiti AppQuery za lociranje UI elemenata (Views).

app.Tap(c=>c.Text("Save"));

app.Tap(c=>c.Marked("SaveUserDataButton"));

app.Tap(c=>c.Marked("Pending").Parent().Class("AppointmentListCell").Index(0));

AppResult[] results = app.Query(c=>c.All())

app.Query(c=>c.Class("UILabel"));

app.Query(c=>c.Id("txtUserName"));

app.Query(c=>c.Class("UILabel").Text("Hello, World"));

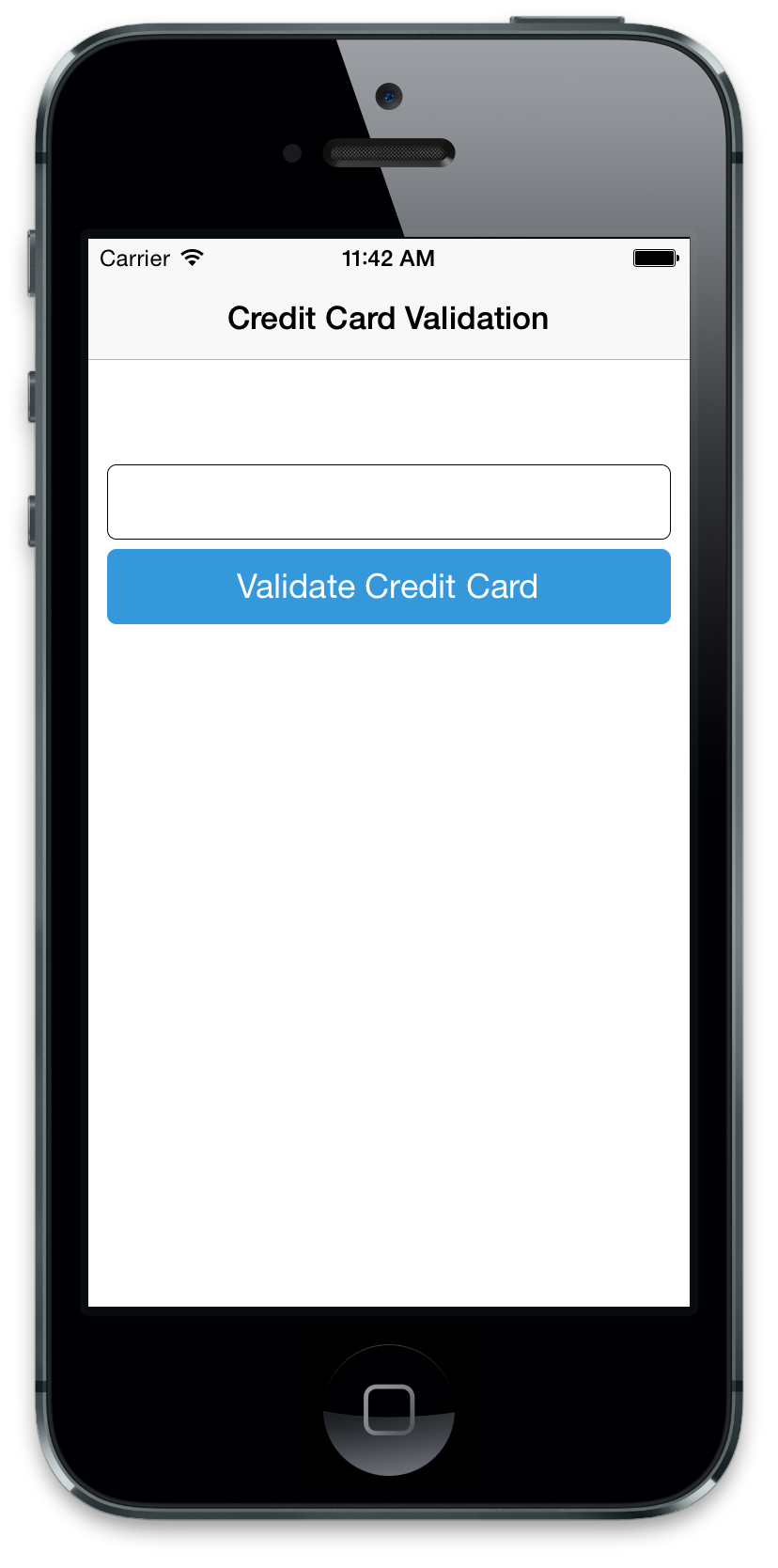
results = app.Query(c=>c.Marked("ValidateButton"));

IApp metoda Repl otvoriti će REPL podsustav (Read, Evaluate, Print, Loop) za interakciju sa aplikacijom - nešto slično kao primitivan shell za operativne sustave.

app.Repl();



Primjer izdavanja komandi (tree):



App has been initialized to the 'app' variable.  
Exit REPL with ctrl-c or see help for more commands.  
  
>>> tree  
[UIWindow > UILayoutContainerView]  
 [UINavigationTransitionView > ... > UIView]  
 [UITextView] id: "CreditCardTextField"  
 [\_UITextContainerView]  
 [UIButton] id: "ValidateButton"  
 [UIButtonLabel] text: "Validate Credit Card"  
 [UILabel] id: "ErrorrMessagesTestField"  
 [UINavigationBar] id: "Credit Card Validation"  
 [\_UINavigationBarBackground]  
 [\_UIBackdropView > \_UIBackdropEffectView]  
 [UIImageView]  
 [UINavigationItemView]  
 [UILabel] text: "Credit Card Validation"  
>>>

## Xamarin.Test.Cloud

Xamarin.Test.Cloud je komercijalna usluga za testiranje mobilnih aplikacija na preko 1000 uređaja (2015-04 broj je oko 1400 većinom Android uređaja).

<http://xamarin.com/test-cloud>

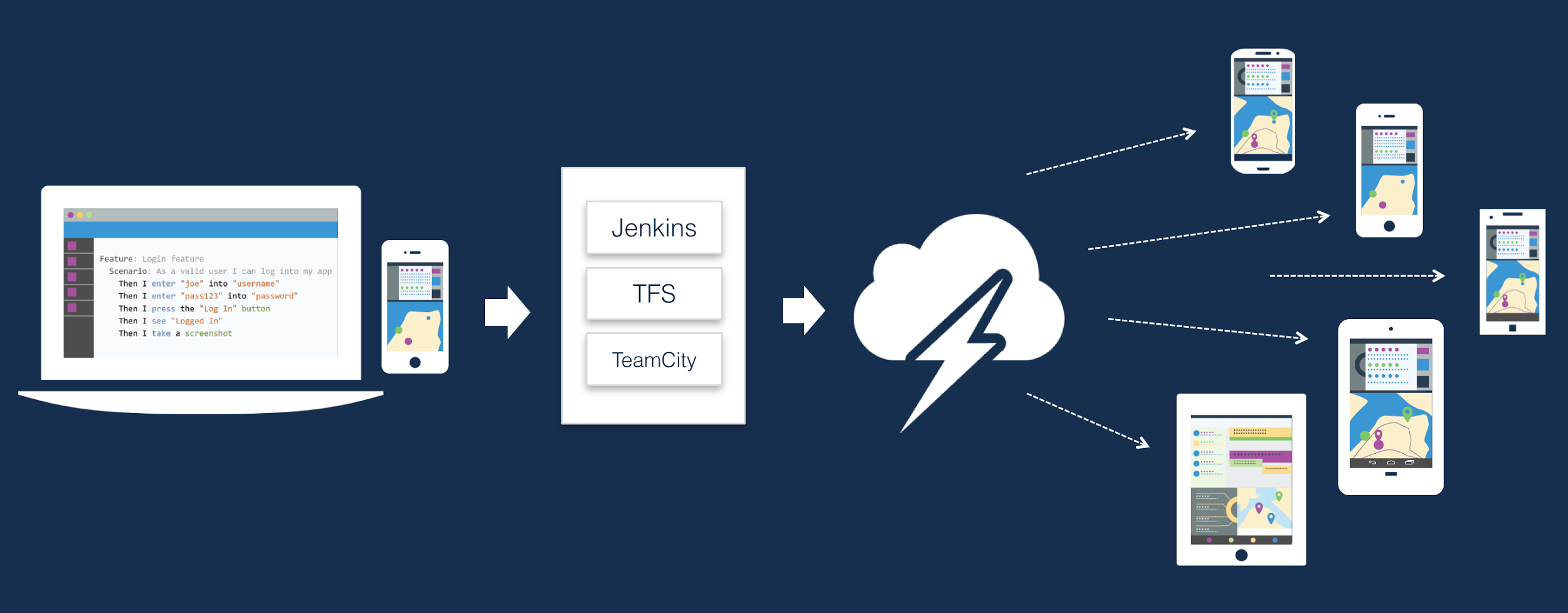
<https://www.youtube.com/watch?v=PQMBCoVIABI&feature=youtu.be&t=57m30s>

Xamarin.Test.Cloud Continuous Integration

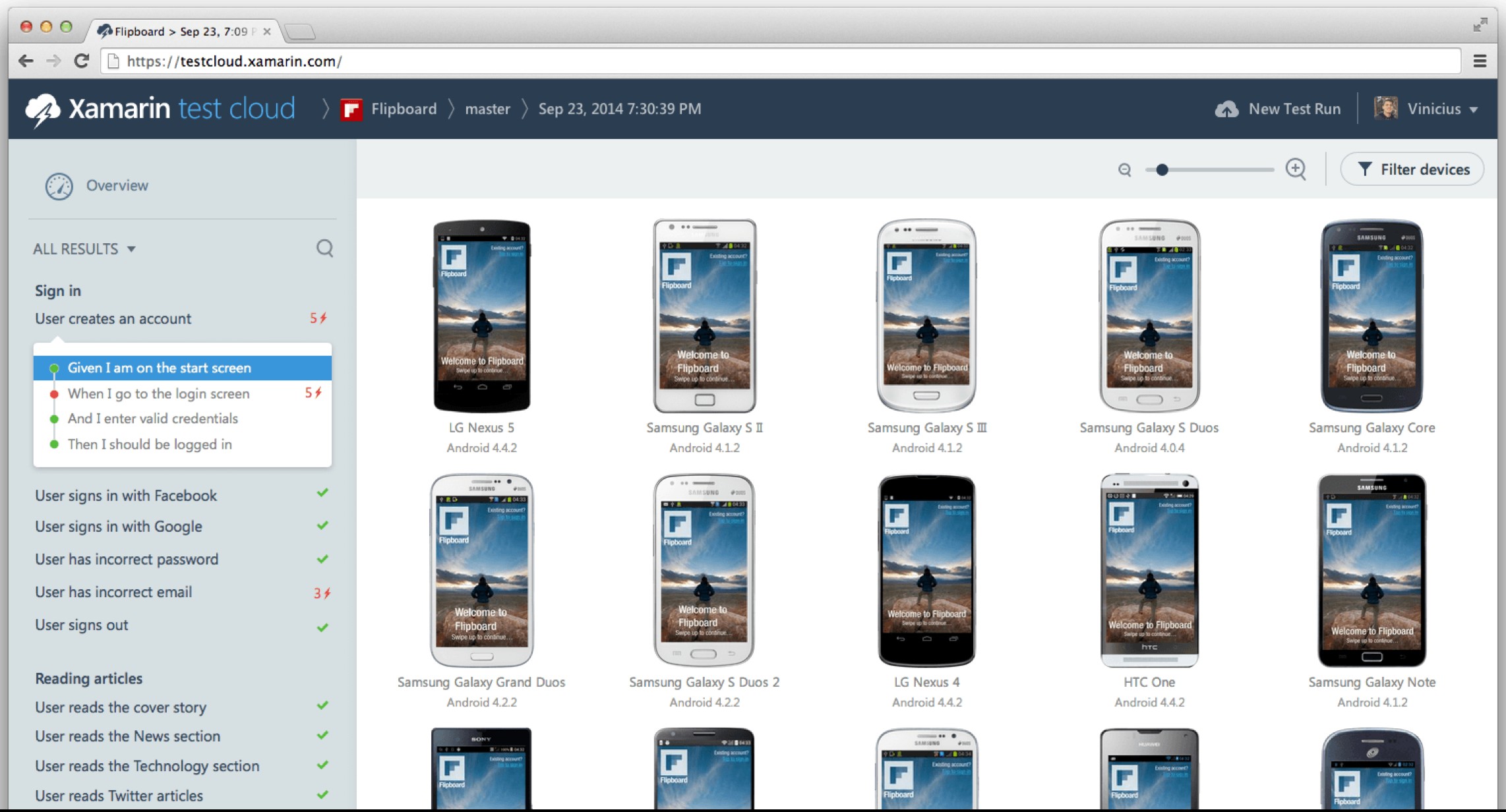
Zbog svoje automatske prirode Xamarin.Test.Cloud se vrlo lako integrira u procese Continous Integration.

# 

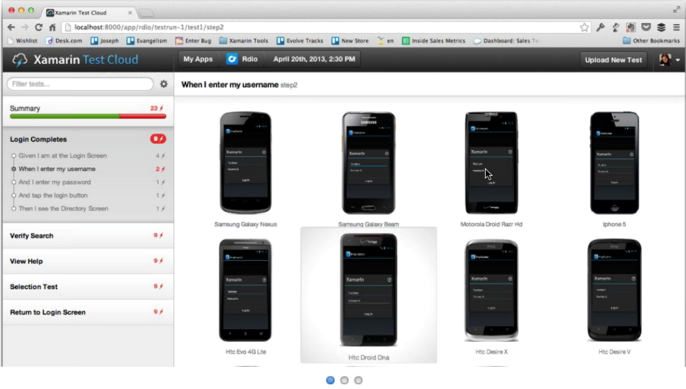
Xamarn.Test.Cloud se po svojim karakteristikama razlikuje od ostalih rješenja, prvenstveno paralelnog testiranja aplikacija, na velikom broju uređaja, što je prednost, jer se rezultati testiranja dobijaju daleko brže.



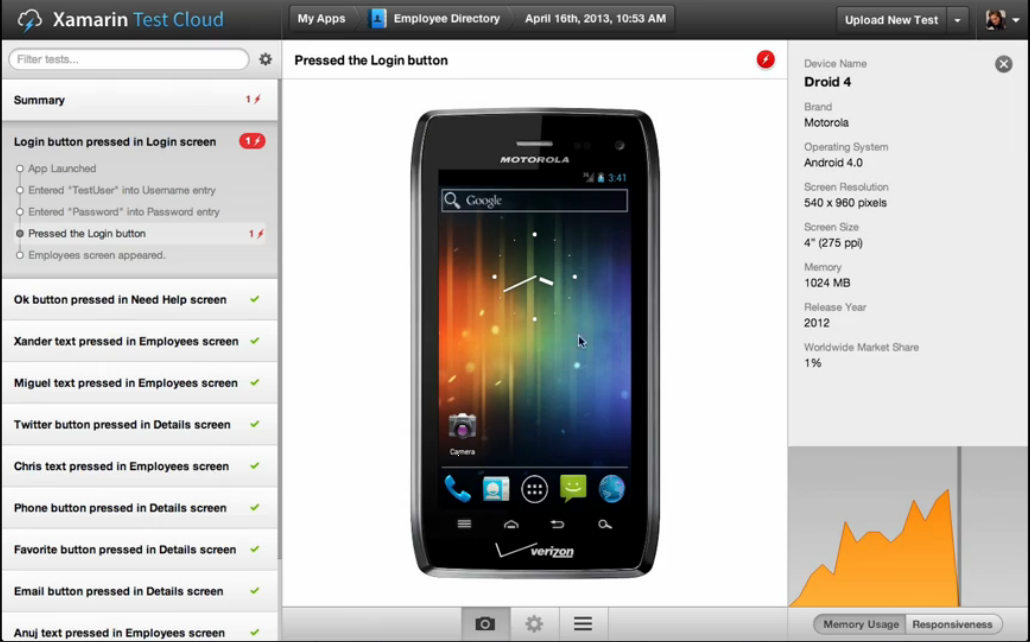
### Screenshots



### Error reporting



### Device DrillDown

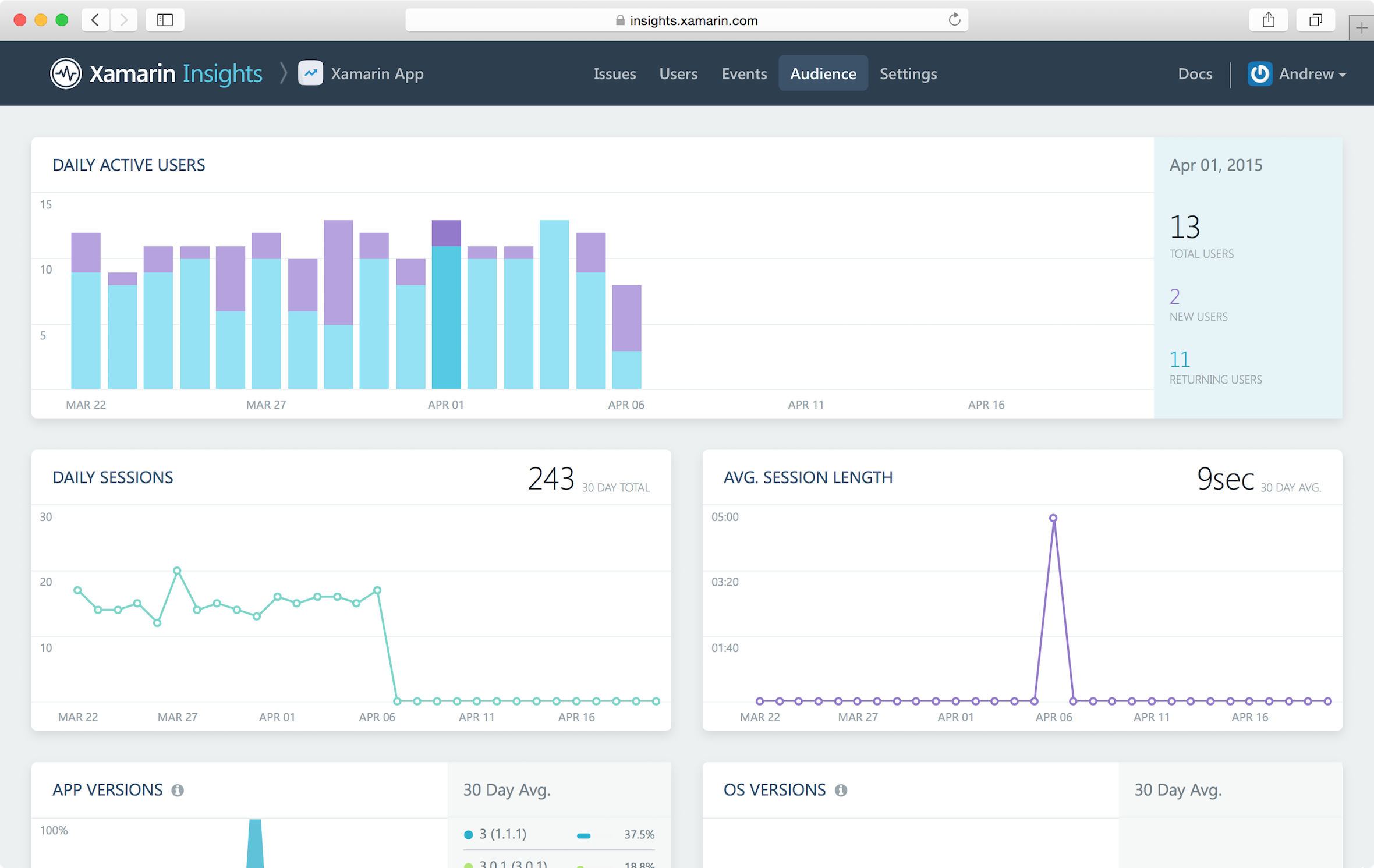


# Monitoring - Xamarin.Insights

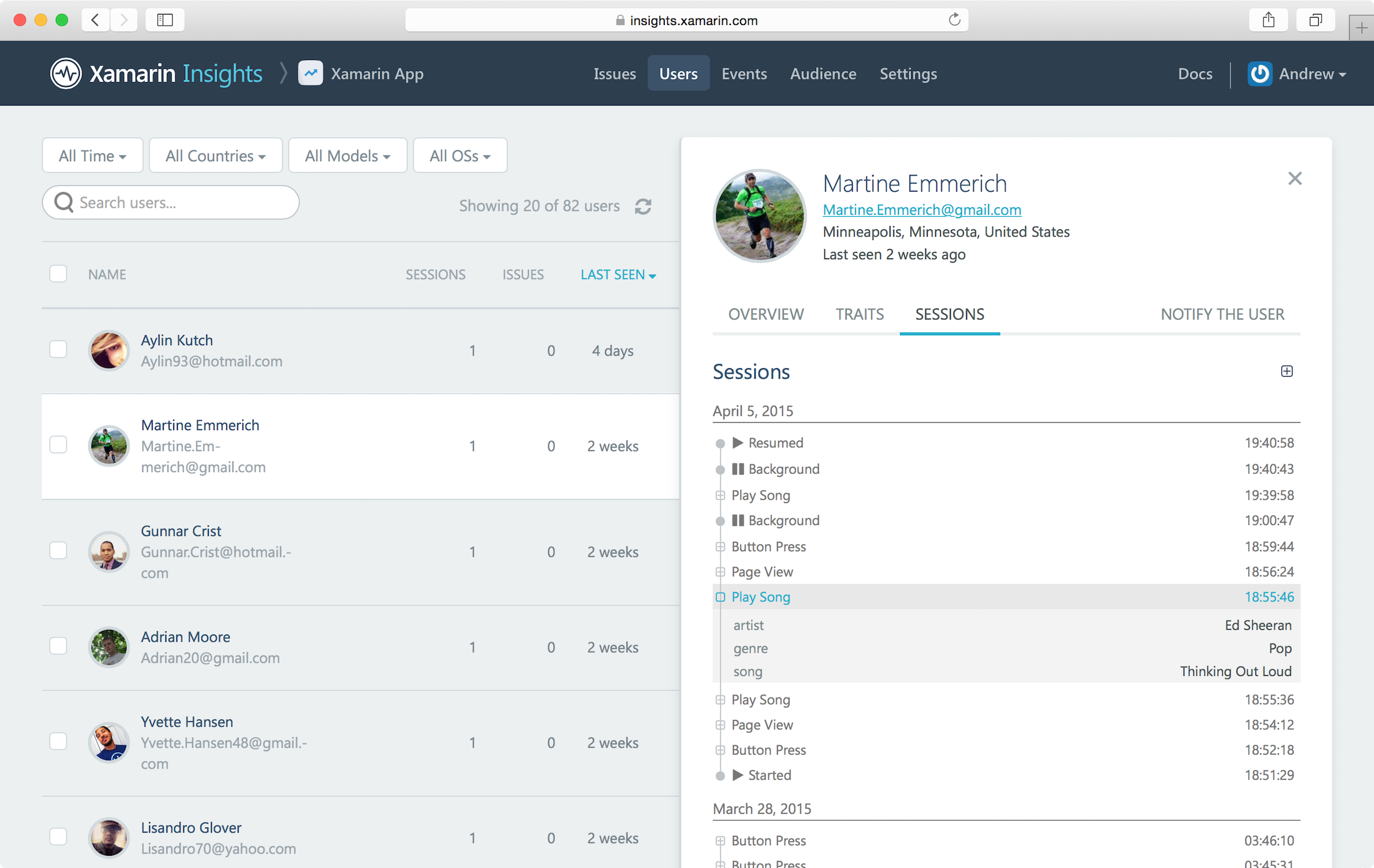
Xamarin.Insights je alat koji omogućava monitoring aplikacije, te mnogobrojna izvješća o stanju aplikacije (greškama, performansama - memorija i procesor).

# 

Proizvod prati i korisnike, njihove demografske podatke (geolokaciju i sl), njihovu interakciju sa aplikacijom do vrlo sitnih detalja ili kumulativno.



Uz korisnike dostupna su i izvješća po geografskim lokacijama.



# Zaključak

Xamarn je kroy novi proces donio nove alate za korisnike Xamarin tehnologije, no i za ostale koji razvijaju nativne i hibridne mobilne aplikacije.

Sam proces je samo prijedlog za “best practice” prema dosadašnjim iskustvima u razvoju aplikacija, koji se temelji na “feedback-u” samih korisnika alata i aplikacija.