FUTURE DECODED

6-7 OTT ‘16 / MILANO

IN PARTNERSHIP WITH: CommunityDays.it

www.futuredecoded.it

#FutureDecoded

Automated UI Testing for iOS and Android Mobile Apps

Massimo Bonanni

Microsoft Sr. Consultant - EMEA Modern App Domain

massimo.bonanni@microsoft.com

@massimobonanni

www.futuredecoded.it

#FutureDecoded

Agenda

Perchè test!!!
Unit vs Coded UI Test
Xamarin Test Cloud
Xamarin.UITest, REPL, Test Recorder
Continuous Integration
Quanto mi costi.....

Perché test!!!!

+ Individuazione dei bug o defect prima del rilascio
+ Incremento della qualità
+ Garanzia di non regressione a seguito di refactoring o nuove funzionalità
+ Se unito alle build automatiche, utilissimo in cicli rapidi di sviluppo

- Più codice da scrivere
- Maggiori skill da parte del team
- Maggiori costi del prodotto

Nell'ambito mobile, cross-plattform, c'è un'altra cosa che ci spinge ad adottare politiche di test.....

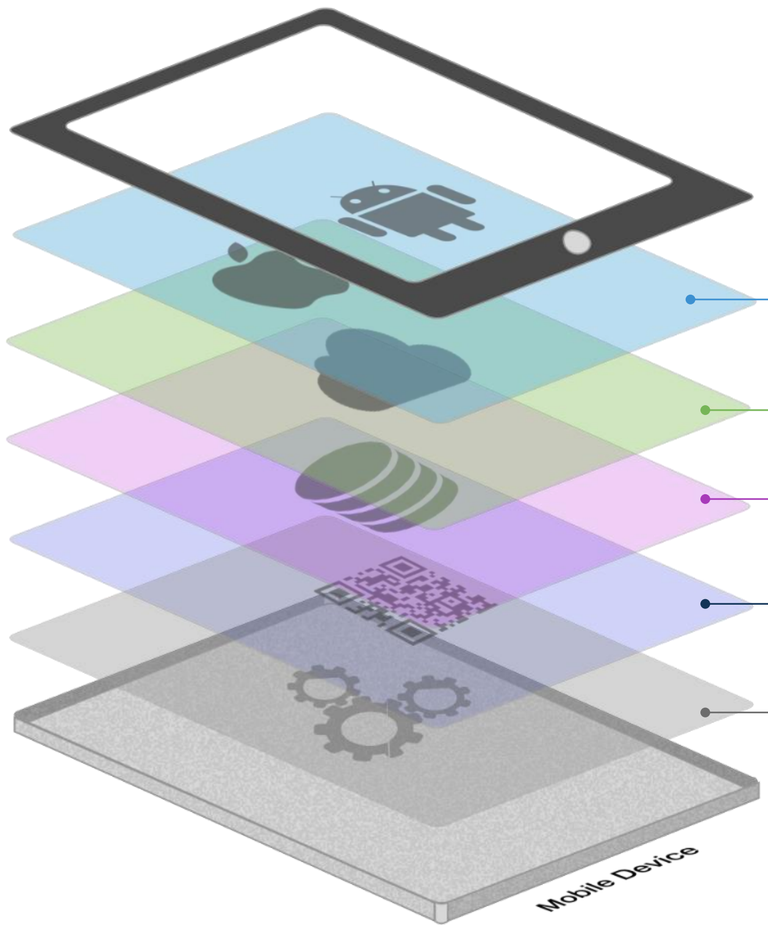

OpenSignal
Galaxy S III
Galaxy S5
SM-G530H
Galaxy S4
Galaxy S4
Moto G
Moto G
Galaxy Note 3
GT-S7580
Moto G
D6503
SM-N9005
SM-G355H
Galaxy S II
SM-G7102
G3
GT-I8262
Galaxy S5
One
GT-N7100
GT-S7582
Galaxy Y
C6903
G2
One (M8)
GT-I8552B
S6 Edge
K012
X
ZenFone 5
One
A369i
A536
Moto G
C2105
Z970
7040N
XL
SM-T211
D5803
D6603
Moto E
D6633
Nexus 6
Joy
D5503

Tutta una questione di numeri…

- 5 **Versioni**
- 20 **Devices**
- 20 **Lingue**
- 35 **Localizzazioni**
- 6 **Formati schermo**

- 9 **Versioni**
- 24K+ **Differenti Device**
- 39 **Lingue**
- 57 **Localizzazioni**
- 27 **Formati schermo**
- 1,294 **Produttori**
- 6 **Configurazioni Schermo**

Complessità delle app



Thousands of iOS and Android APIs

Public web APIs

Backend integrations

Third party libraries

Memory, CPU and network constraints

Unit Test….non basta!!

Disaccoppiare la UI dalla logica restrostante (es. MVVM) e testare i ViewModel con degli Unit Test non basta a garantire che la User Experience sia delle migliori:

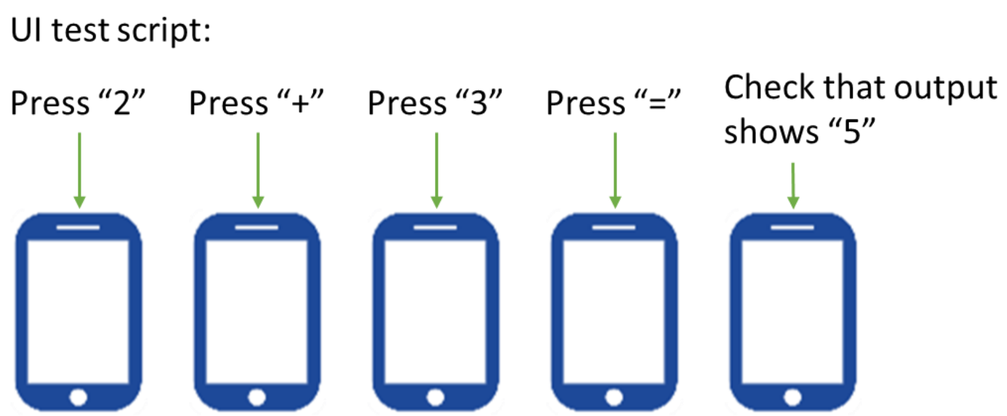
- I tempi di reazione dell'interfaccia variano in base alla potenza del dispositivo
- La corretta visualizzazione dei controlli della UI dipendono dalla dimensione e dal form factor dello schermo
- I tempi di interazione con i sistemi esterni possono creare problemi non previsti in sede di unit test
- ….

E…..allora?

Il modo migliore per validare il comportamento di una applicazione è utilizzarla.

Se l'applicazione ha il comportamento atteso senza errori o eccezioni, allora può essere rilasciata.

Il processo di test del comportamento della UI prende il nome di UI Acceptance Test o Coded UI Test (quando sono «codificati»).

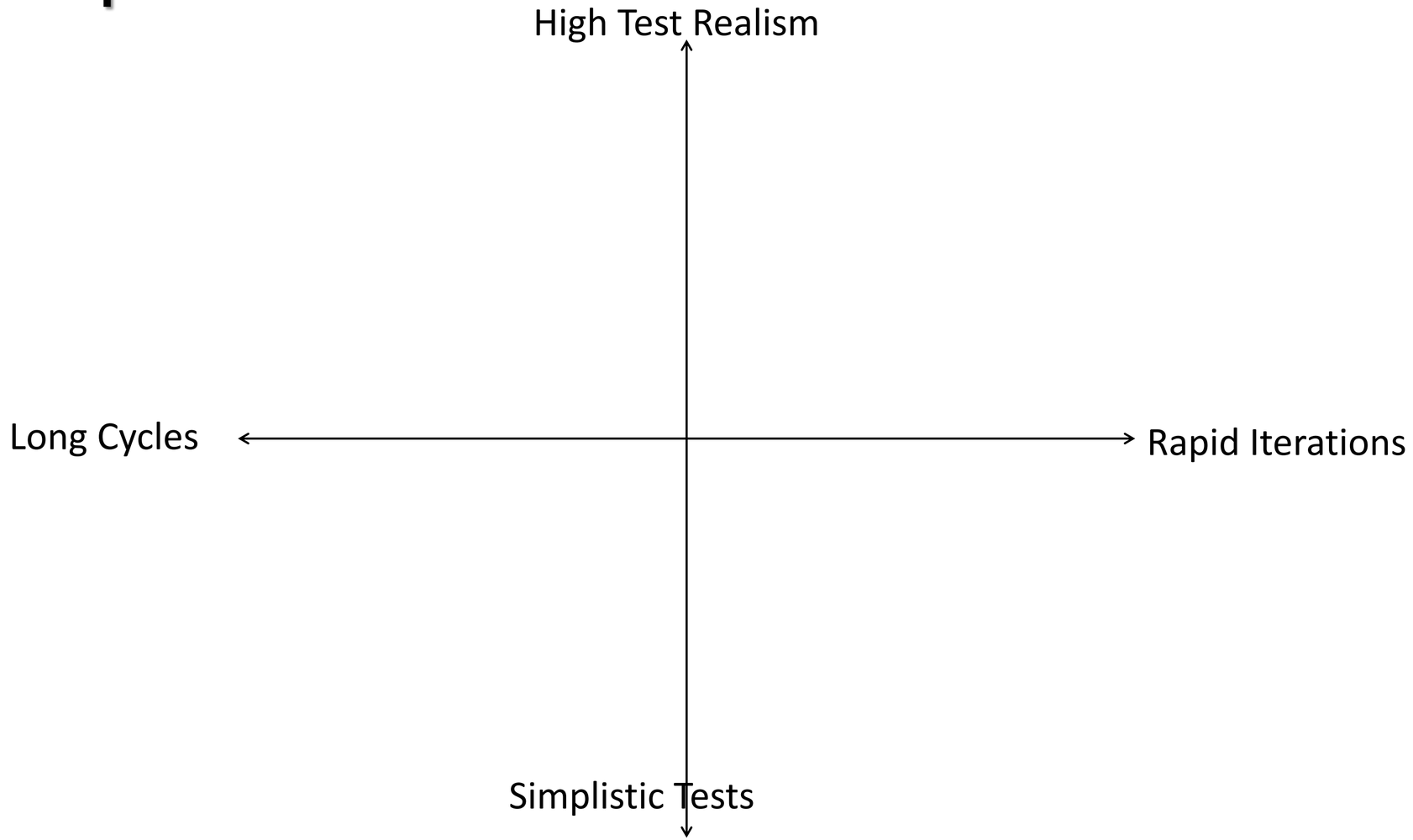


UI Acceptance Test

Questo processo ha i seguenti limiti:

- E’ costoso perché necessita dei dispositivi hardware;
- E’ costoso perché debbono essere impiegate persone per il test manuale;
- E’ dispendioso a livello di tempo perché il tester è un essere umano (processo poco adatto a rilasci rapidi)
- E’ prono di errori perché l’uomo tende a distrarsi e sbagliare quando esegue operazioni ripetitive e meccaniche
- E’ di difficile rendicontazione (il tester può semplicemente dire se il test è andato bene o andato male ma senza l’utilizzo di tecnologie accessorie non è in grado di ottenere altre informazioni come memoria o cpu occupata)

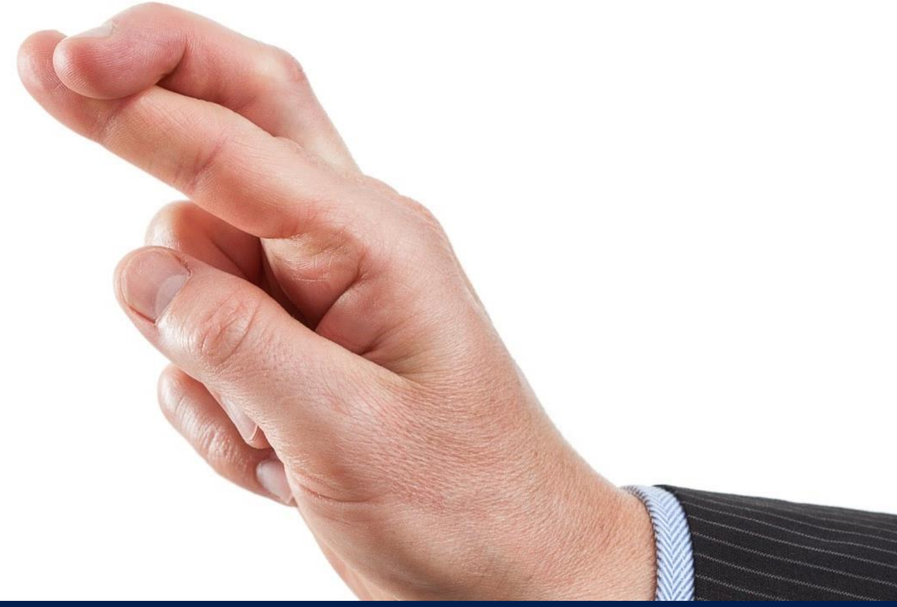
Il magic quadrant dei test



Cosa facciamo?

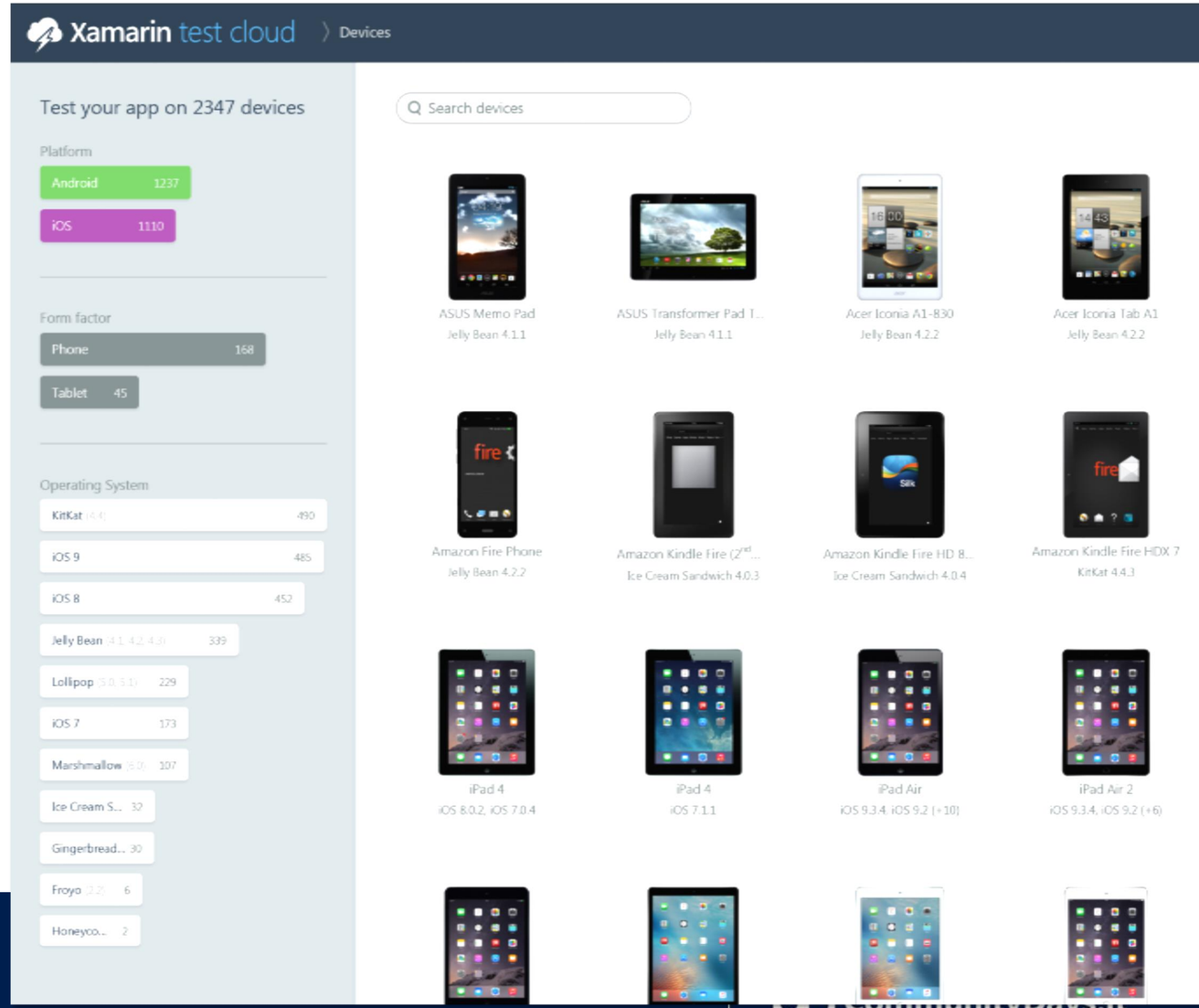
Soluzioni:

- Non si fa UI Test !!! ☹ ☹
- Si spera che tutto funzioni al meglio!!! ☹ ☹ ☹
- Si sfrutta il concetto di beta
- …
- Si utilizza una piattaforma di testing automatica, scalabile e veloce: **Xamarin Test Cloud**



Cosa è Xamarin Test Cloud?

- Xamarin Test Cloud è un servizio basato su cloud che fornisce una piattaforma di test per la user interface automatizzata
- Fornisce centinaia di dispositivi diversi di differenti brand e sistemi operativi
- Consente a chiunque di verificare che il comportamento della UI sia conforme ai requisiti attraverso una varietà di dispositivi con il minimo sforzo.
- La manutenzione e l'approvvigionamento di nuovi dispositivi viene semplificata di molto dalla natura Cloud del sistema



Cosa è Xamarin Test Cloud?

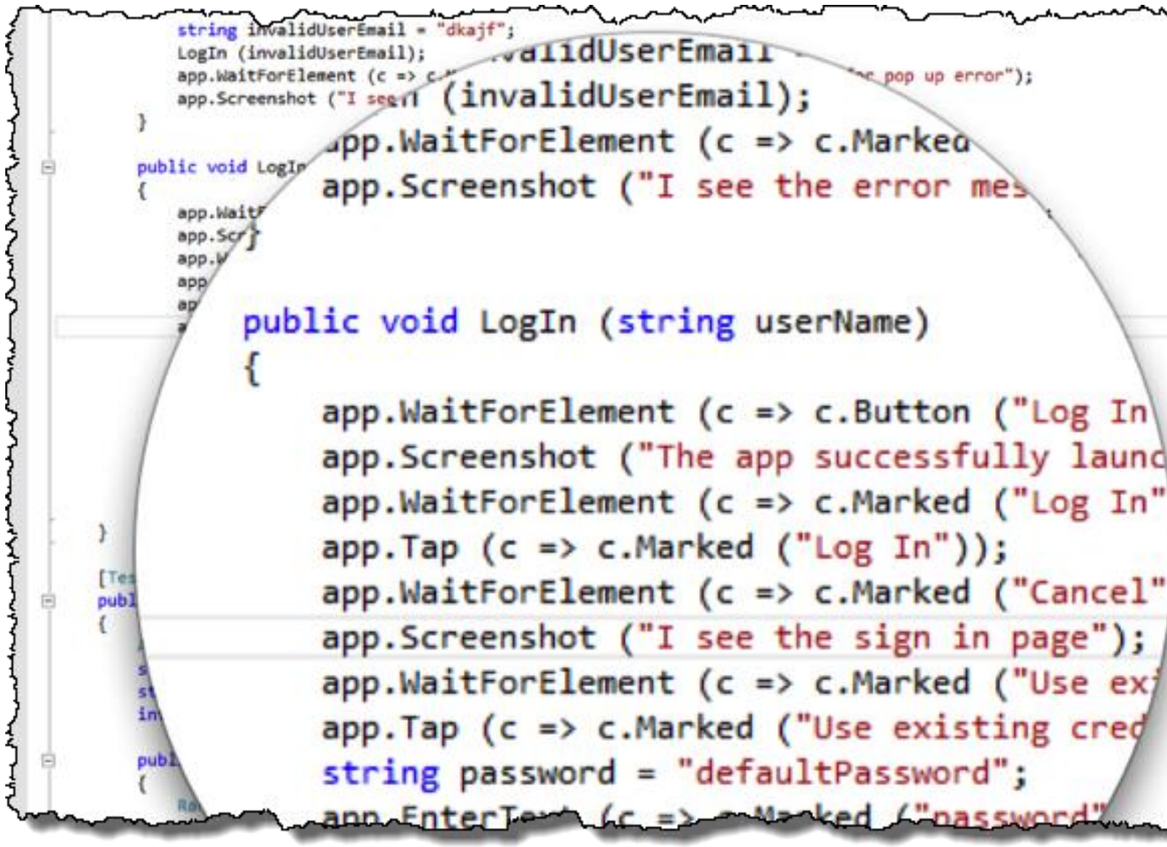
Xamarin Test Cloud supporta i seguenti framework di test:

Xamarin.UITest: framework che permette di scrivere test in C#. Basato su Nunit. Utilizzabile nelle applicazioni cross-plattform scritte con Xamarin (sia native che Forms)

Calabash: framework che permette di scrivere test in Ruby utilizzando Cucumber. Permette di scrivere test utilizzando «la lingua del business» sfruttando le regole grammaticali di Cucumber. Utile per le applicazioni native scritte senza l'uso di Xamarin

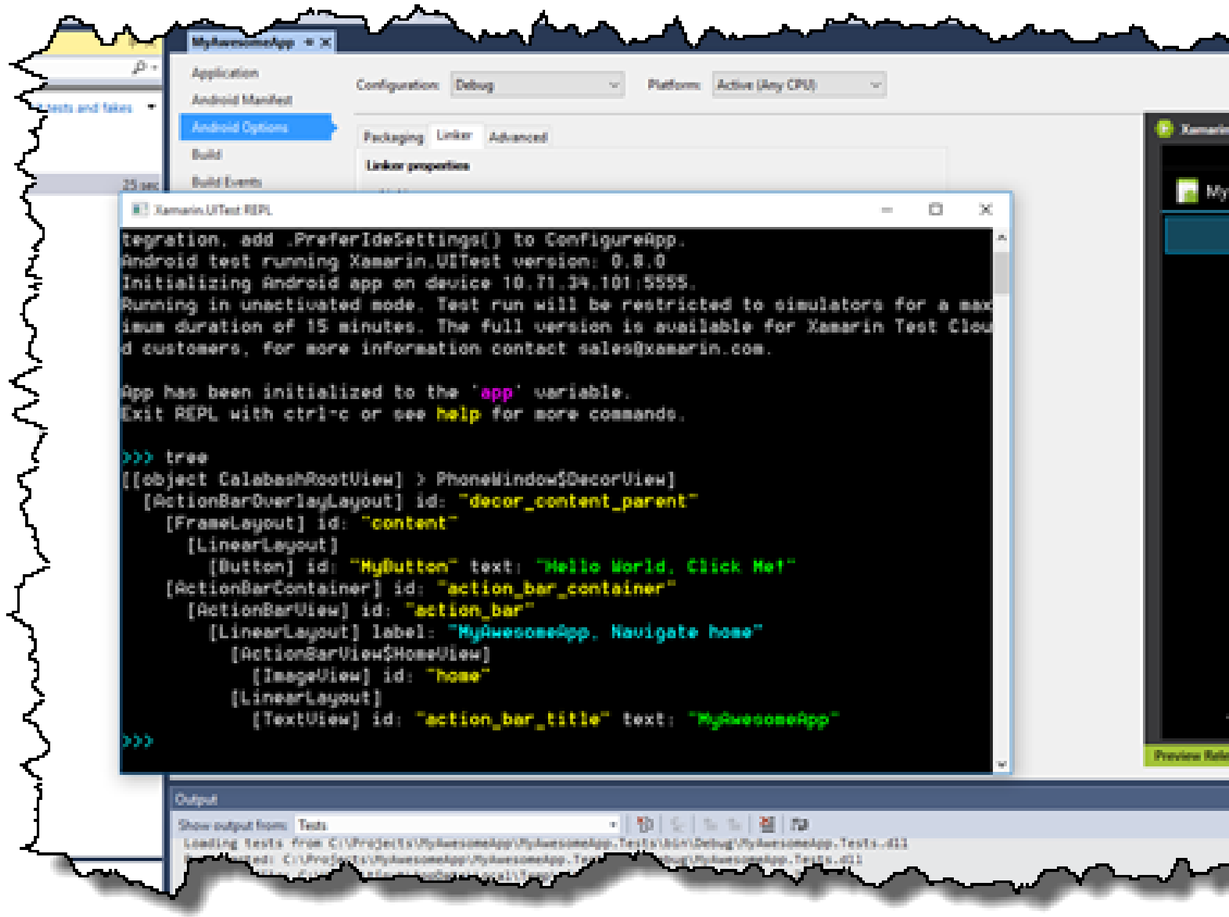
Xamarin.UITest

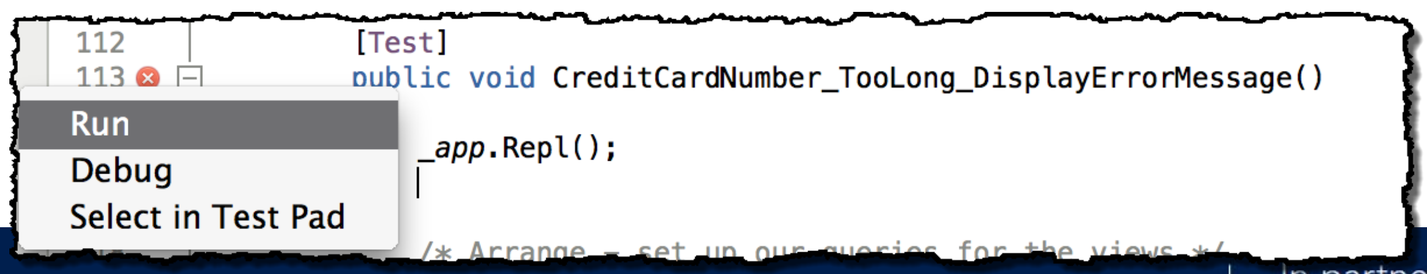
- Basato su Nunit
- Visual Studio e Xamarin Studio forniscono template di progetto per UI Test
- Fornisce supporto alle gesture
- Struttura simile agli unit test
- Modalità per eseguire Xamarin.UITests:
  - Eseguire l'upload dei test (e dell'app) su Xamarin Test Cloud.
  - Eseguire i test localmente utilizzando un device, un emulatore (Android), o un simulatore (iOS) e sfruttando il Test Runner in Xamarin Studio (iOS o Android) o Visual Studio (Android).



REPL

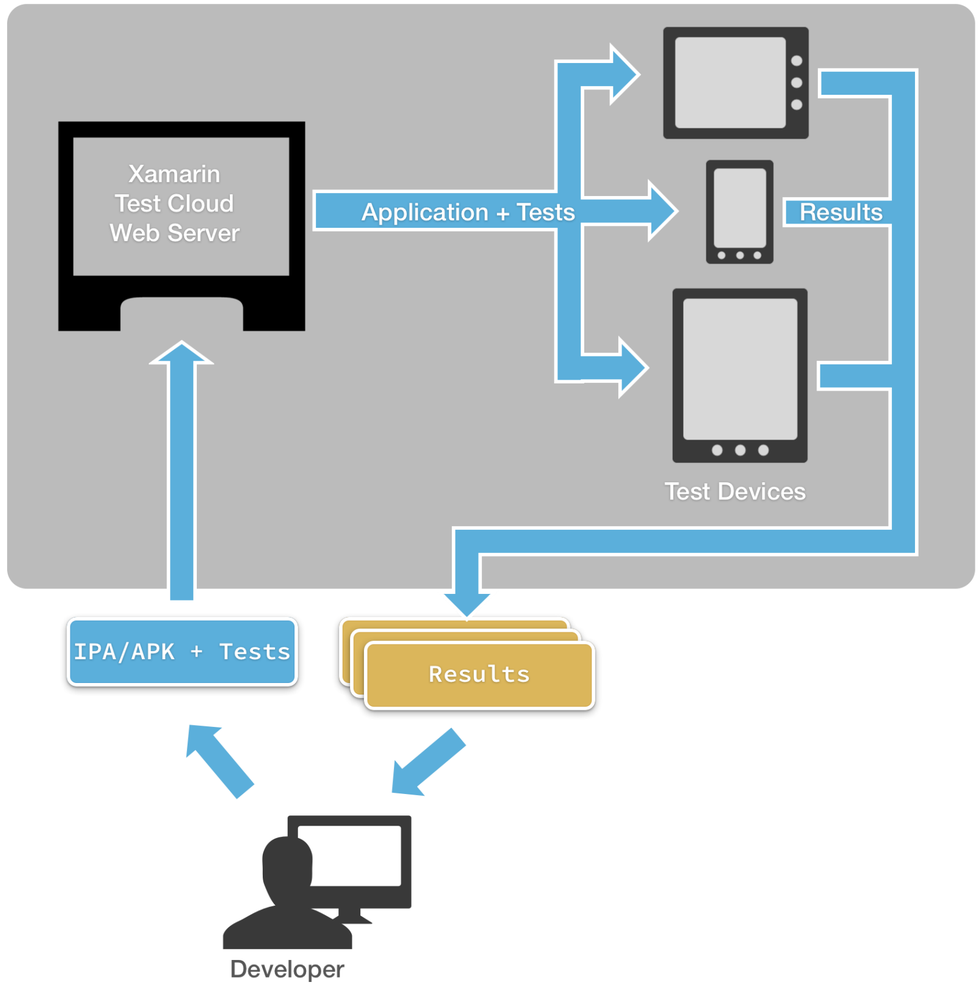
- REPL (**read-eval-print-loop**) è uno dei più importanti tools per la creazione di UI Test.
- Tool a riga di commando con cui gli sviluppatori possono eseguire espressioni e commandi.
- Permette di “esplorare” l’interfaccia utente ed interagire in tempo reale con essa.
- I comandi utilizzati possono essere esportati in Visual Studio (o Xamarin Studio)





Come funziona Xamarin Test Cloud

1. I test vengono creati localmente (in un progetto Visual Studio di test)
2. L'applicazione sotto test e i relativi test vengono inviati a Xamarin Test Cloud
3. Applicazione e test vengono installati ed eseguiti sui device selezionati dall'utente
4. Al termine di tutti i test, Test Cloud colleziona i risultati e li invia all'utente



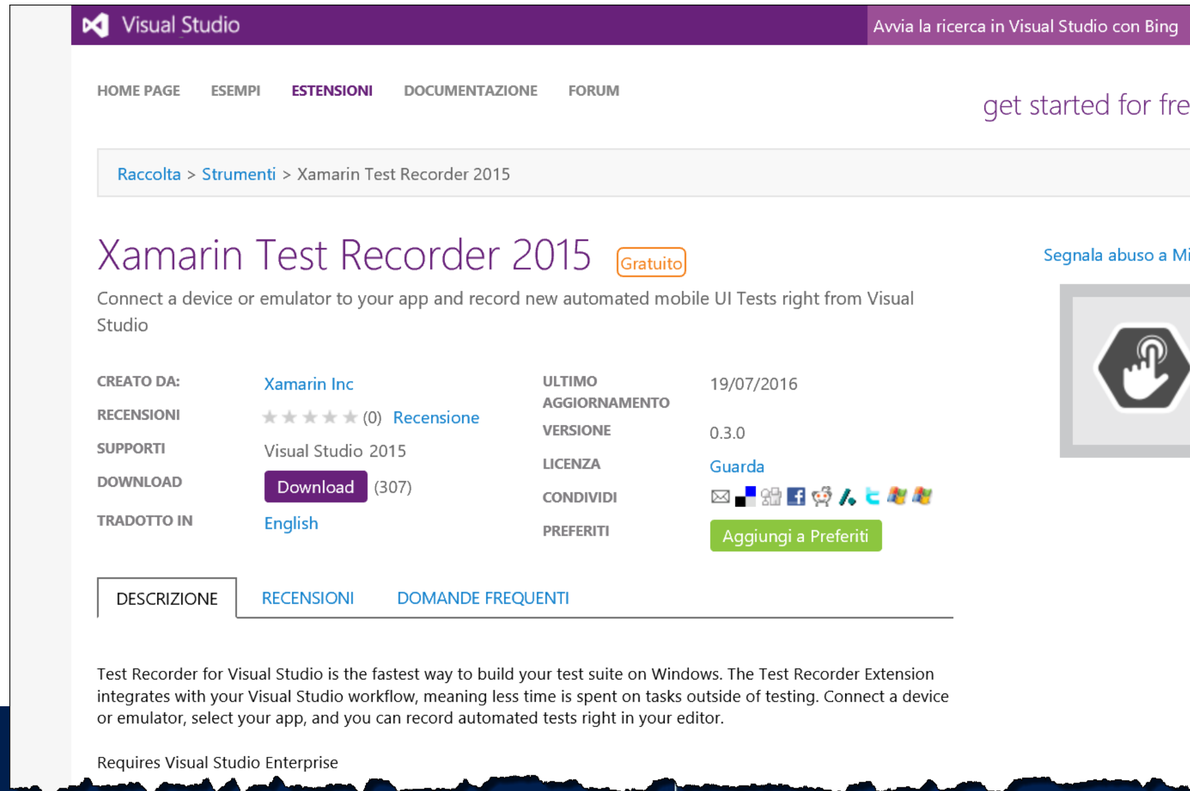
DEMO – My first UI.Test

www.futuredecoded.it

#FutureDecoded

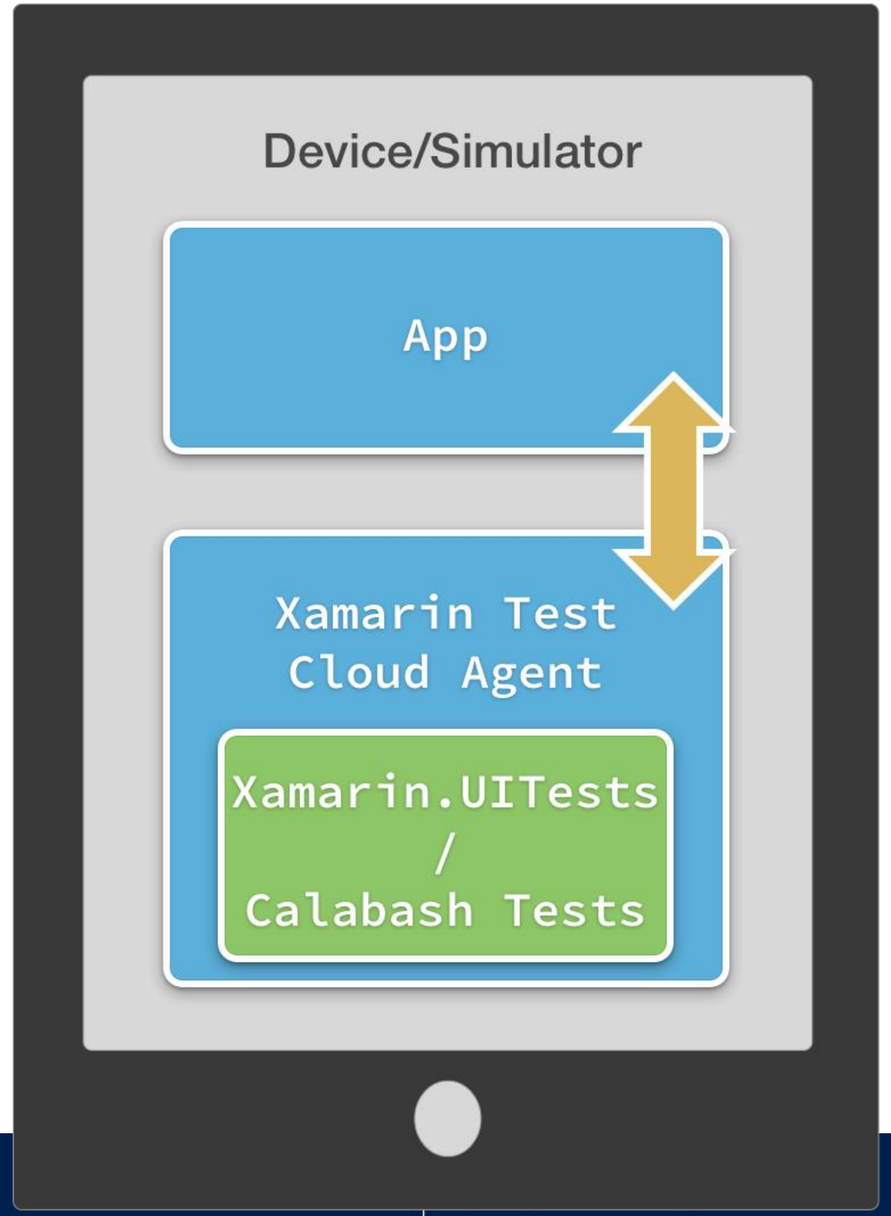
Xamarin Test Recorder for Visual Studio

- Consente di registrare coded UI test utilizzando un device fisico (via USB) o un emulatore;
- Solo con Visual Studio Enterprise;
- Al momento supporta solo Android (per iOS Xamarin Test Recorder for Mac OSX)
- La registrazione necessita di Android 4.3 (API Level 23) o superiori ma il test generato può essere eseguito anche su versioni precedenti



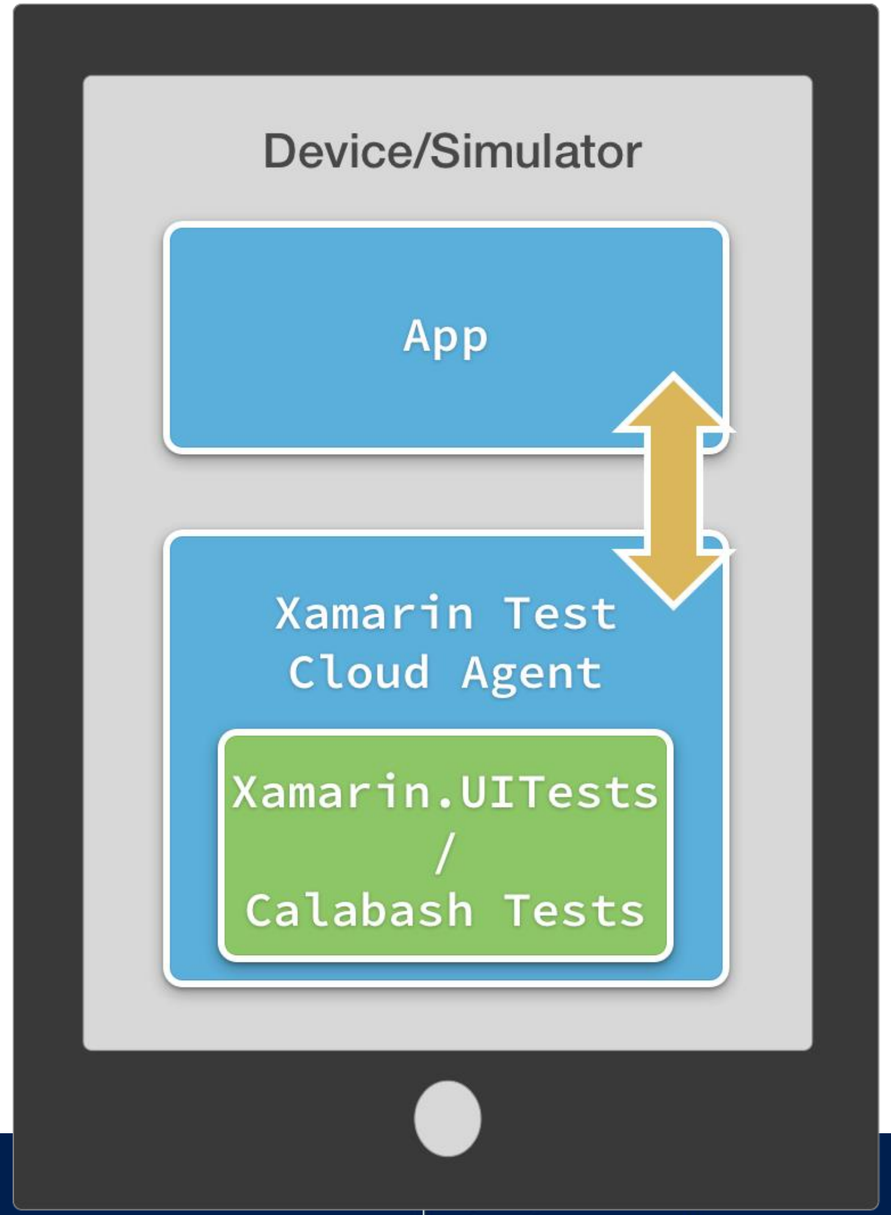
Anatomia di un test

- Calabash e Xamarin.UITest da soli non sono in grado di interagire con l'interfaccia dell'applicazione (semplicemente perché non ne fanno parte)
- Hanno la necessità di una «libreria di automazione» che effettivamente si occupa di eseguire il test sul device e si occupa dell'interazione con la UI
- Xamarin Test Cloud Agent si occupa di «hostare» i test e di eseguirli nel device



Xamarin Test Cloud Agent

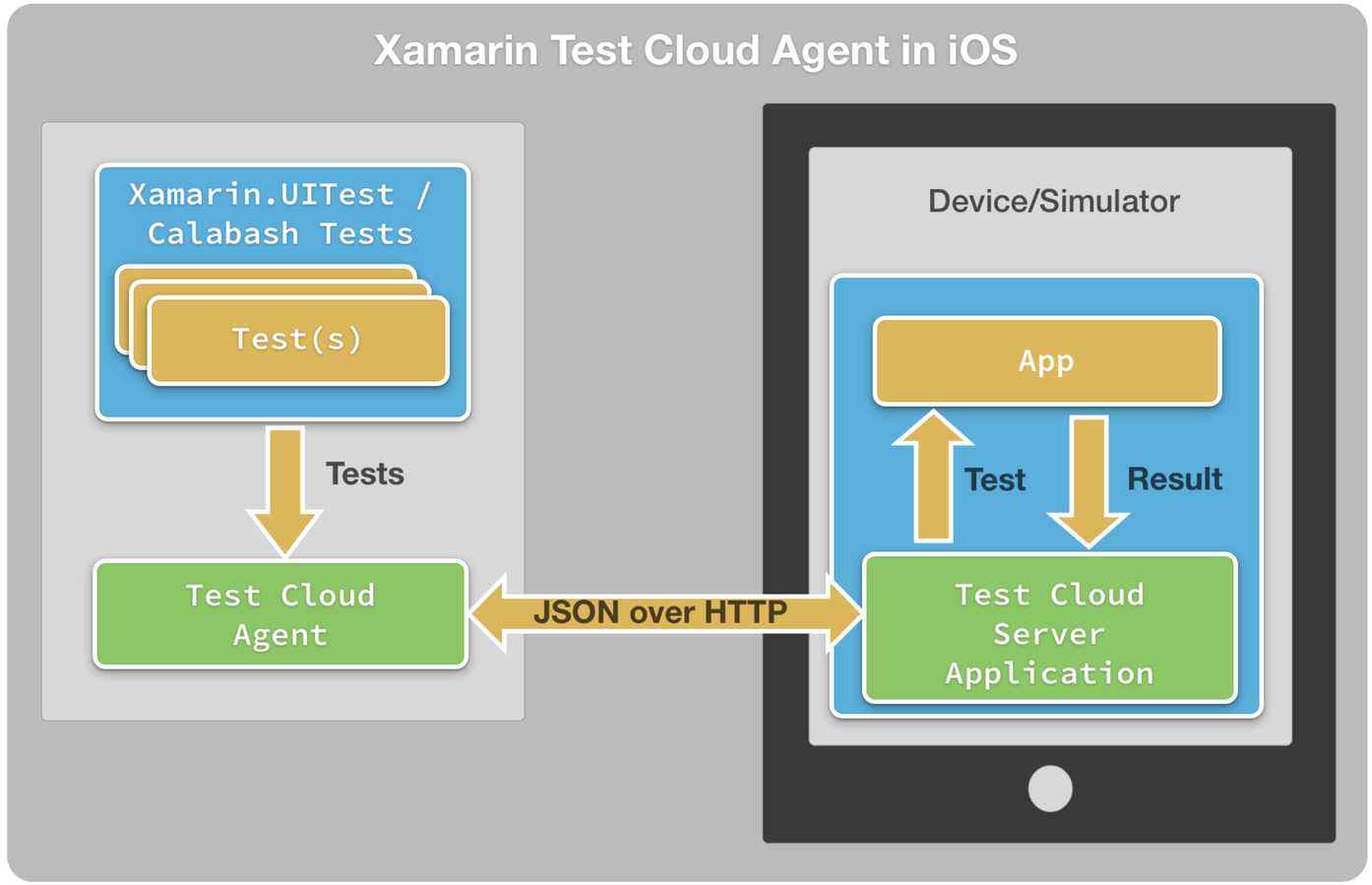
- Xamarin Test Cloud Agent è installato da Test Cloud assieme all'applicazione
- Implementa un'architettura client/server per far in modo che la piattaforma Test Cloud possa effettivamente comandare l'esecuzione dei test e il recupero dei risultati
- La parte server è, di fatto un web server (molto semplice), che viene installato sul device ed è in ascolto per rispondere ai comandi (JSON over HTTP) inviati dal client
- Il client si trova all'interno della piattaforma Test Cloud



Xamarin Test Cloud Agent su iOS

A causa delle limitazioni imposte dalla piattaforma Apple, il Test Cloud Agent è compilato assieme all'applicazione.

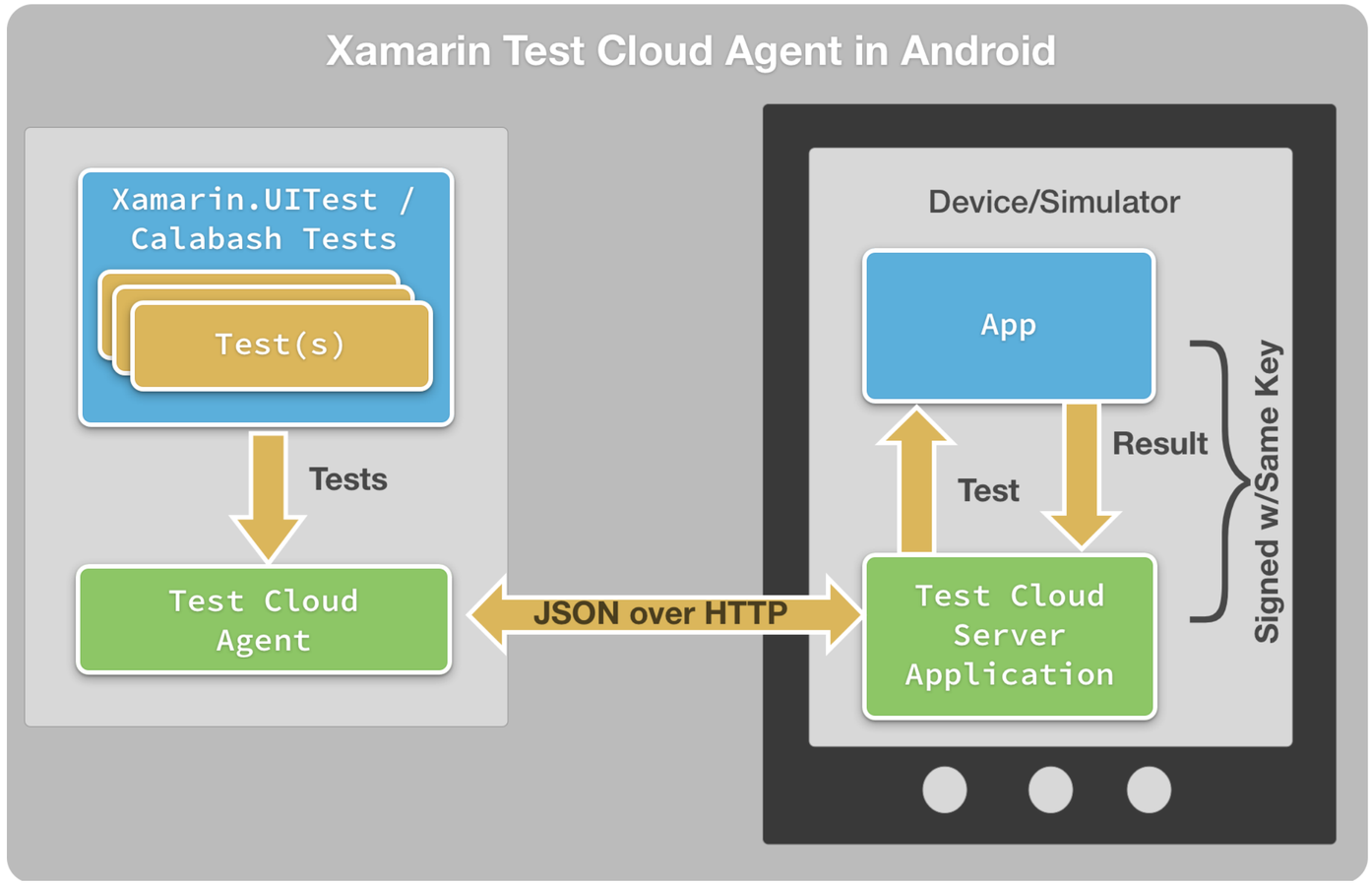
Viene utilizzato un package NuGet per referenziare l'agent nell'applicazione



Xamarin Test Cloud Agent su Android

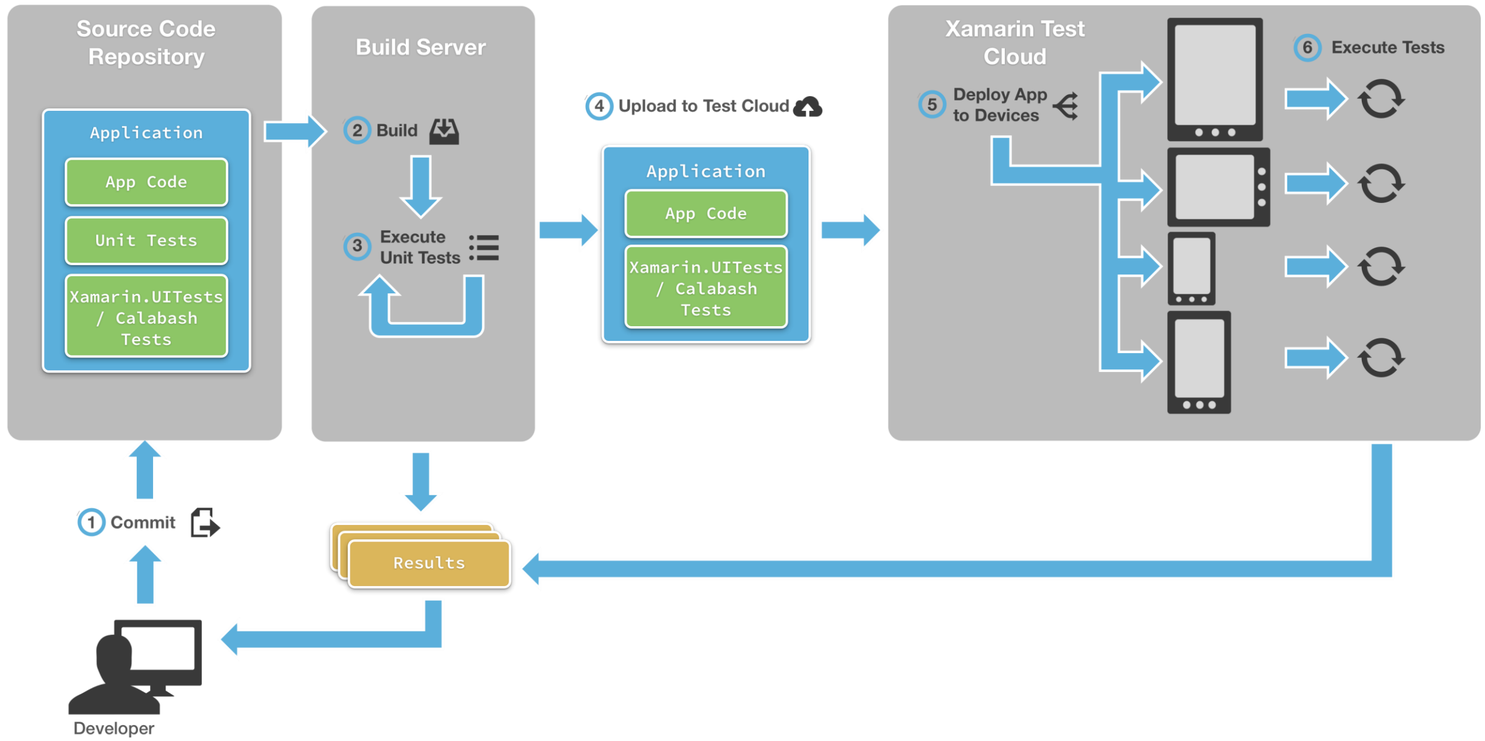
Test Cloud Agent è un'applicazione a se stante che viene distribuita con i permessi per instrumentare l'applicazione che stiamo testando.

L'applicazione da testare e il Test Agent sono firmati con la stessa chiave in modo da poter tra loro comunicare.



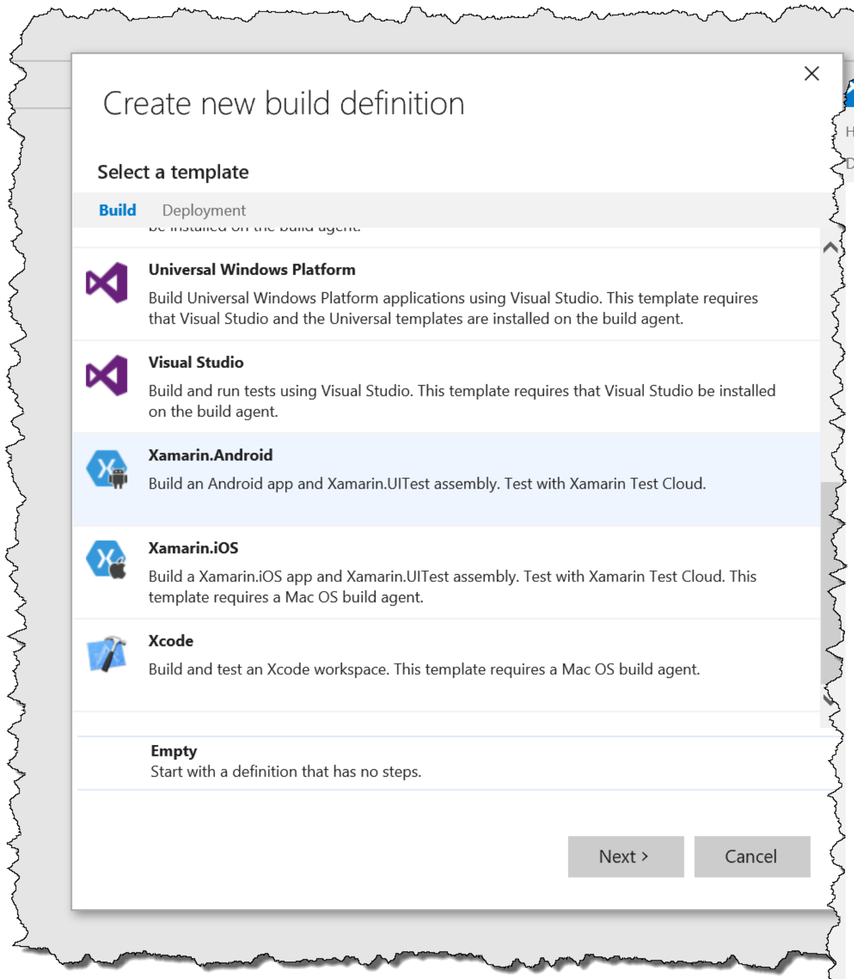
Xamarin Test Cloud Continuous Integration

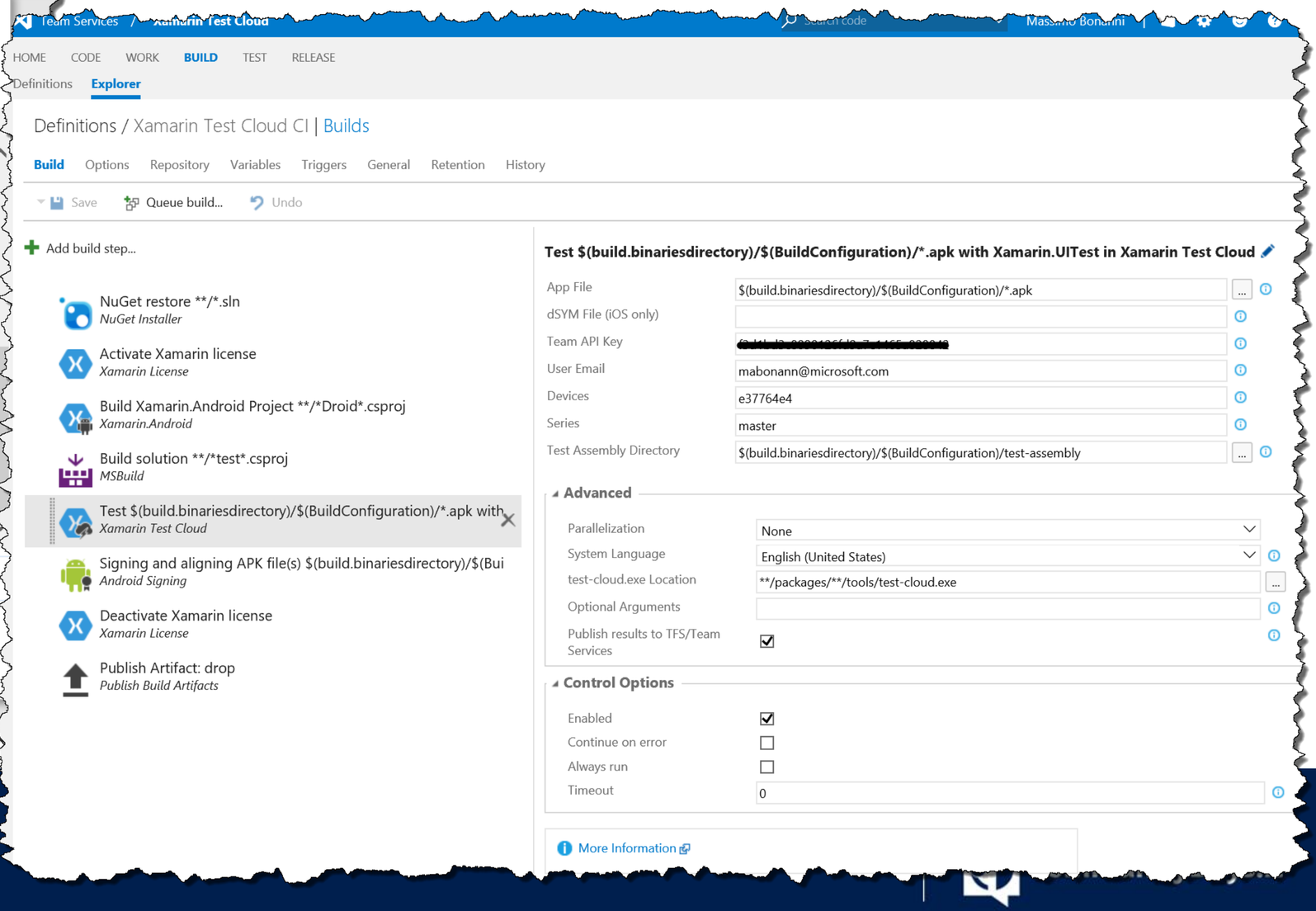
Il meccanismo di pubblicazione dell'app e dei test può avvenire anche in un contesto di Continuous Integration (CI)



Microsoft

Xamarin Test Cloud Continuous Integration





Ma quanto mi costi....

START

$99/month

or $1,009.80 billed annually

1 concurrent device

1 device hour per day

✔ Unlimited apps

✔ Forum support

Start a 30-day free trial

SCALE

$379/month

or $3,865.80 billed annually

3 concurrent devices

5 device hours per day

✔ Unlimited apps

✔ Email support

msdn subscriptions

-25%

ACCELERATE

$799/month

or $8,149.80 billed annually

5 concurrent devices

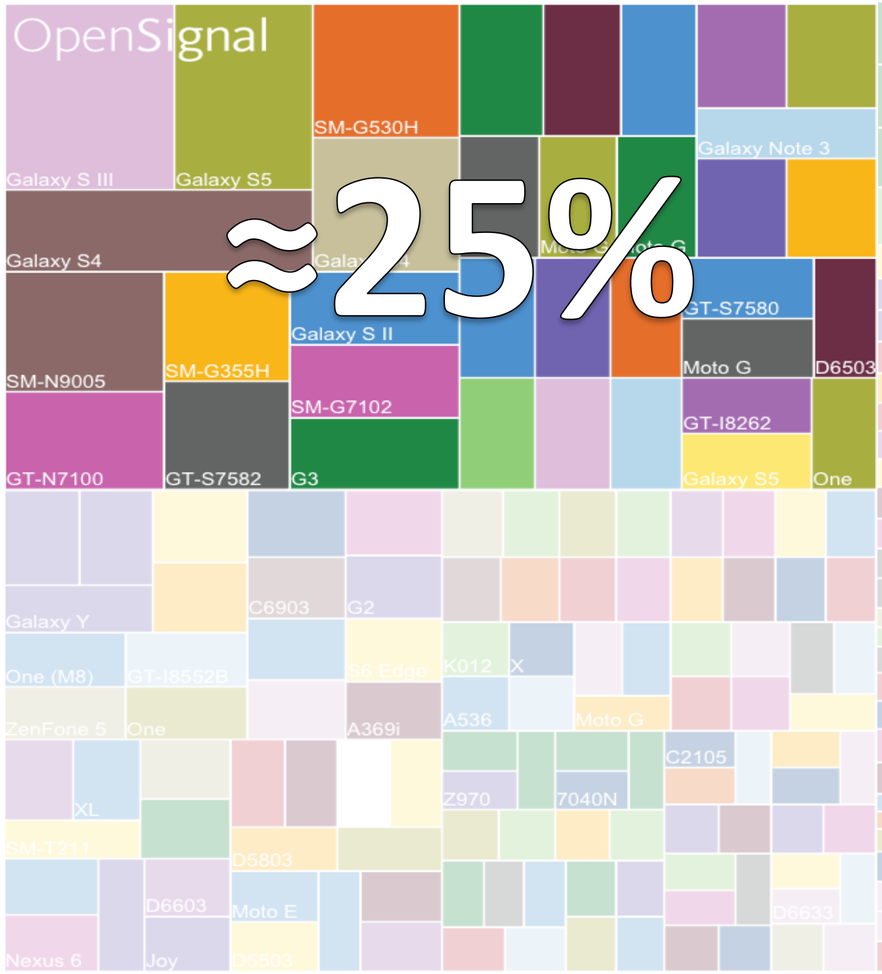
10 device hours per day

✔ Unlimited apps

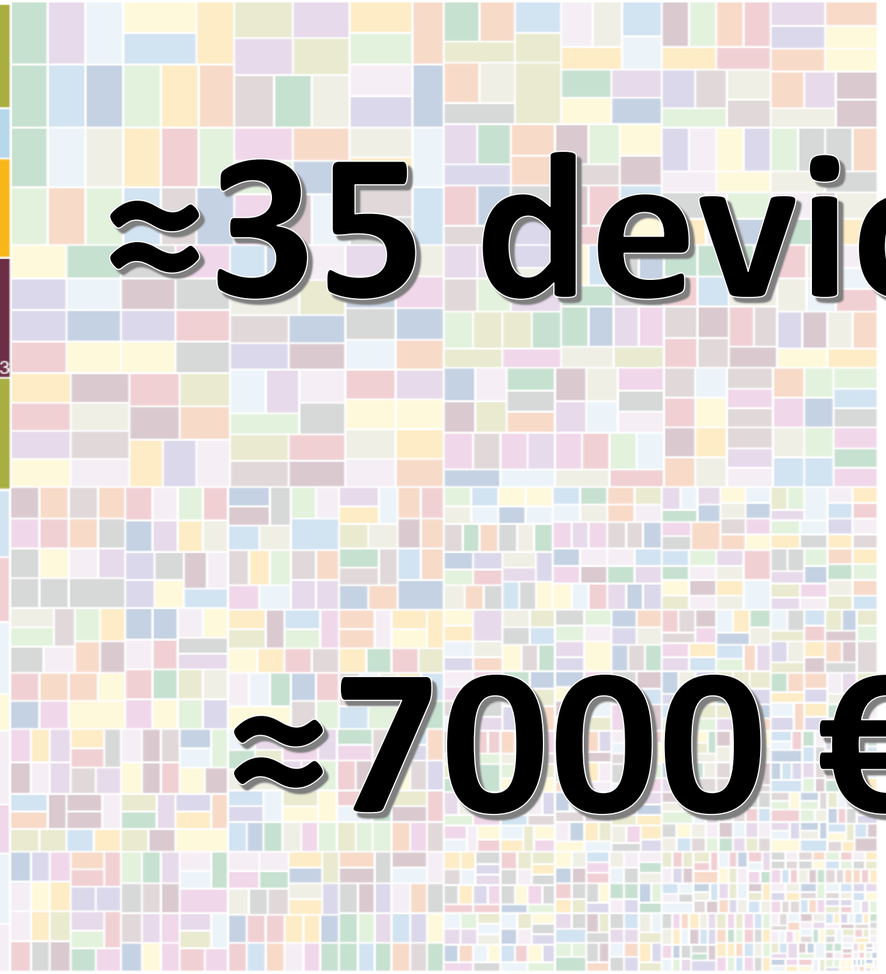
✔ Email support

Start a 30-day free tr

Ma quanto mi costi....



≈25%



≈35 device

≈7000 €

Take Away

- L'utilizzo di Xamarin Test Cloud va valutato in base ai risultati desiderati

- Xamarin Test Cloud va bene anche se non usate Xamarin (app native iOS o Android)

- Migliaia di device a disposizione + report su utilizzo di risorse

Reference

Xamarin Test Cloud Web Site

https://testcloud.xamarin.com

Documentazione ufficiale

https://developer.xamarin.com/guides/testcloud/

Test Recorder for Visual Studio

https://developer.xamarin.com/guides/testcloud/testrecorder/visual-studio/

Devices

https://testcloud.xamarin.com/devices

Demo

https://github.com/massimobonanni/DDN-XamarinTestCloudDemo

Q&A