

Przyszłość ma na imię Mobile

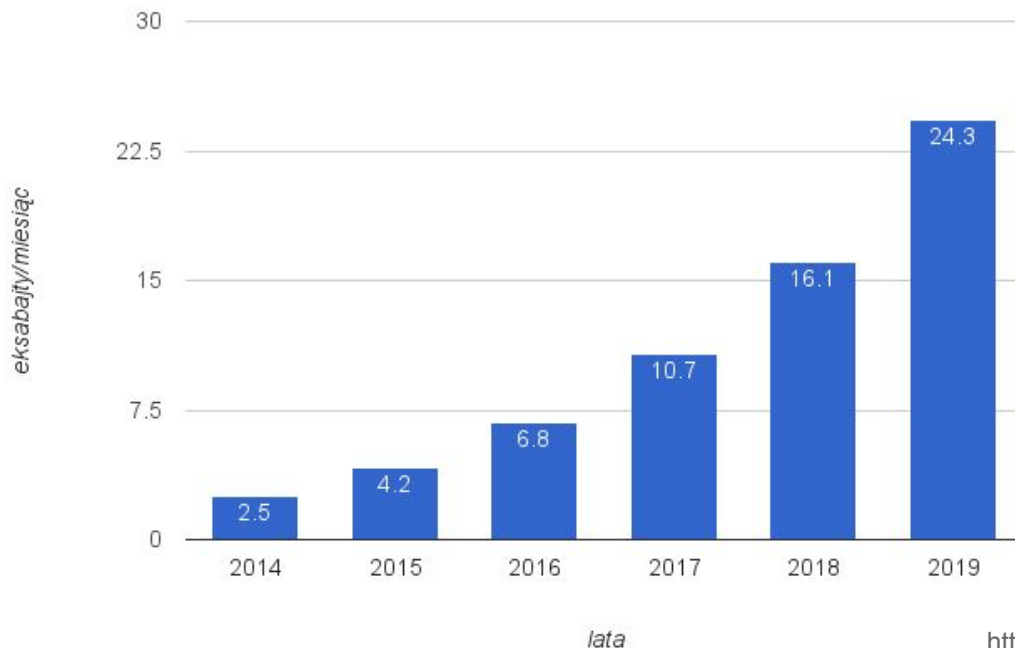
Testy aplikacji mobilnych
na platformę iOS



Ewa Bielska & Tomasz Mnich
TestWarez 2015

Dlaczego mobile?

Ruch generowany przez urządzenia mobilne w latach 2014-2019



<http://www.statista.com/statistics/271405/global-mobile-data-traffic-forecast/>

Cel prezentacji

Przedstawienie **sprawdzonych rozwiązań** z zakresu automatyzacji testów aplikacji mobilnych na system iOS

Agenda

1. Wymagania wobec narzędzi do testów
2. **Live testing** przy użyciu narzędzi:
 - **KIF**
 - **Xcode UI Testing**
 - **Calabash**
 - **FBSnapshotTestCase**
3. Podsumowanie

Jakie narzędzie wybrać? 1/2

- testy na jedną / kilka **platform**



- **język** programowania



- **niezawodność** i szybkość biblioteki



- możliwość **zrównoleglenia** testów



Jakie narzędzie wybrać? 2/2

- wsparcie i **rozwój** biblioteki - *open source*
- testy na **urządzeniach** / symulatorach
- wsparcie dla ***Continuous Integration***



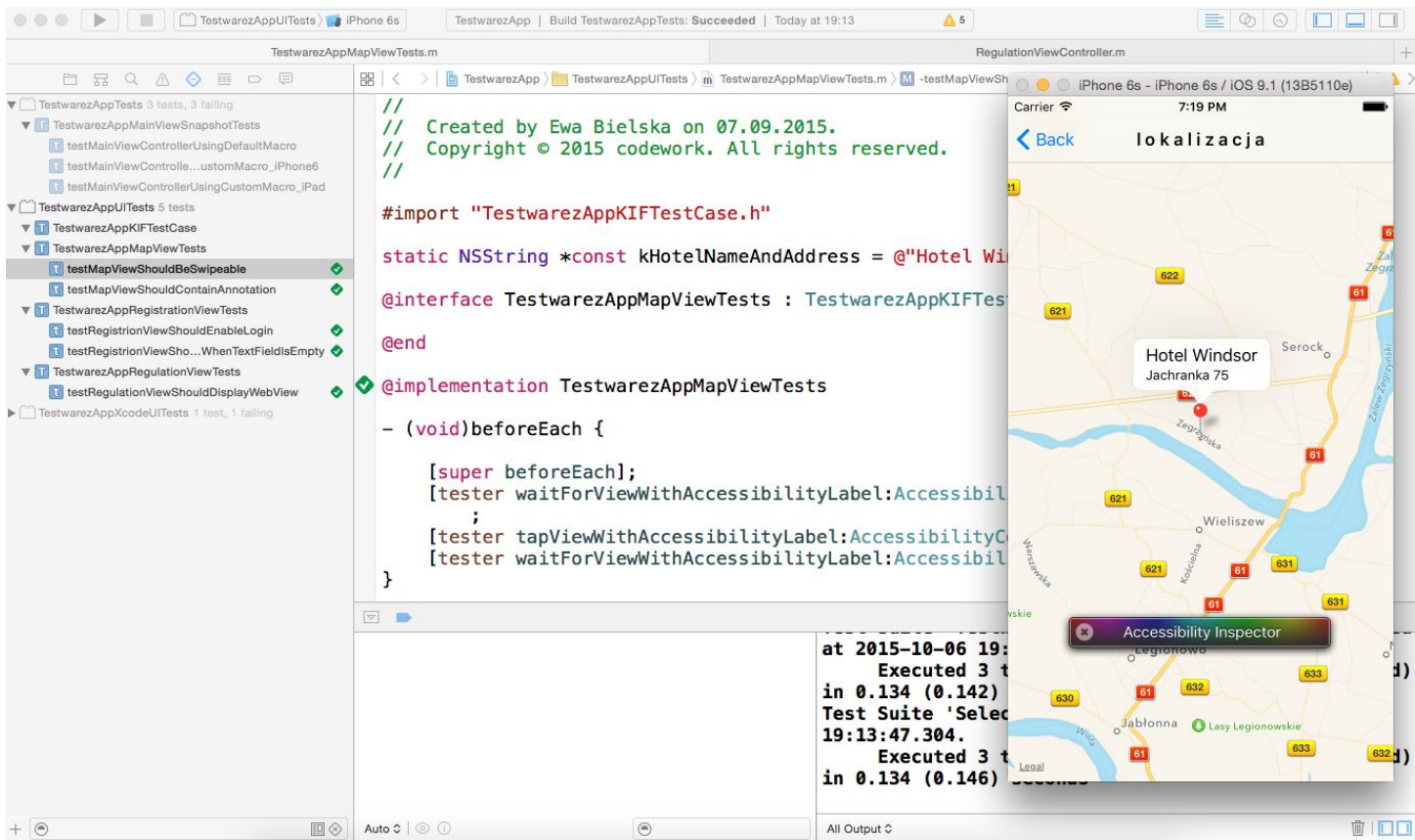
KIF - *Keep It Functional* 1/2

- testuj jak **użytkownik**
- użycie atrybutów ***accessibility***
- kod testów w **natywnym** języku aplikacji

KIF - *Keep It Functional* 2/2

- rozszerzenie biblioteki **XCTest**
- środowisko: **Xcode** lub linia komend
- instalacja: **cocoapods** lub github

KIF - *demo*



KIF - podsumowanie

- osobny Xcode **target** na testy UI
- użycie klas **KIFUITestCase** i **KIFUITestActor**
- uruchamianie testów przez **Test Navigator**
- użycie **accessibilityLabel/Identifier**
- Xcode **Coverage + UI Testing**

XCode UI Testing - podsumowanie

- osobny Xcode **target** na testy UI
- użycie klas **XCTestCase** i **XCUIApplication**
- uruchamianie testów przez **Test Navigator**
- użycie **accessibilityLabel/Identifier** lub hierarchii
- możliwość **nagrywania**

Calabash iOS 1/2

- multi-platformowe narzędzie
- prosta integracja w projekcie - *install gem*
- open source
- *specyfikacje przez przykłady*

Calabash iOS 2/2

- wsparcie aplikacji natywnych i hybrydowych
- uruchamianie testów na emulatorach & urządzeniach
- kod skryptów testowych w Ruby
- spójne API w iOS & Android w wersji Calabash 2.0

Calabash iOS & Cucumber

cucumber

Calaba.sh

- **Gherkin** - specyfikacja w naturalnym języku
Given, When, Then
- **Cucumber** - narzędzie uruchamiające testy
- **Calabash** - interfejs łączący Cucumber'a z testowaną aplikacją na platformie iOS

Calabash iOS - demo



Calabash iOS - podsumowanie

- Multi-platformowe zastosowanie
- Bardzo prosta integracja w projekcie
- Duże wsparcie społeczności open source
- Stabilne i dojrzałe narzędzie

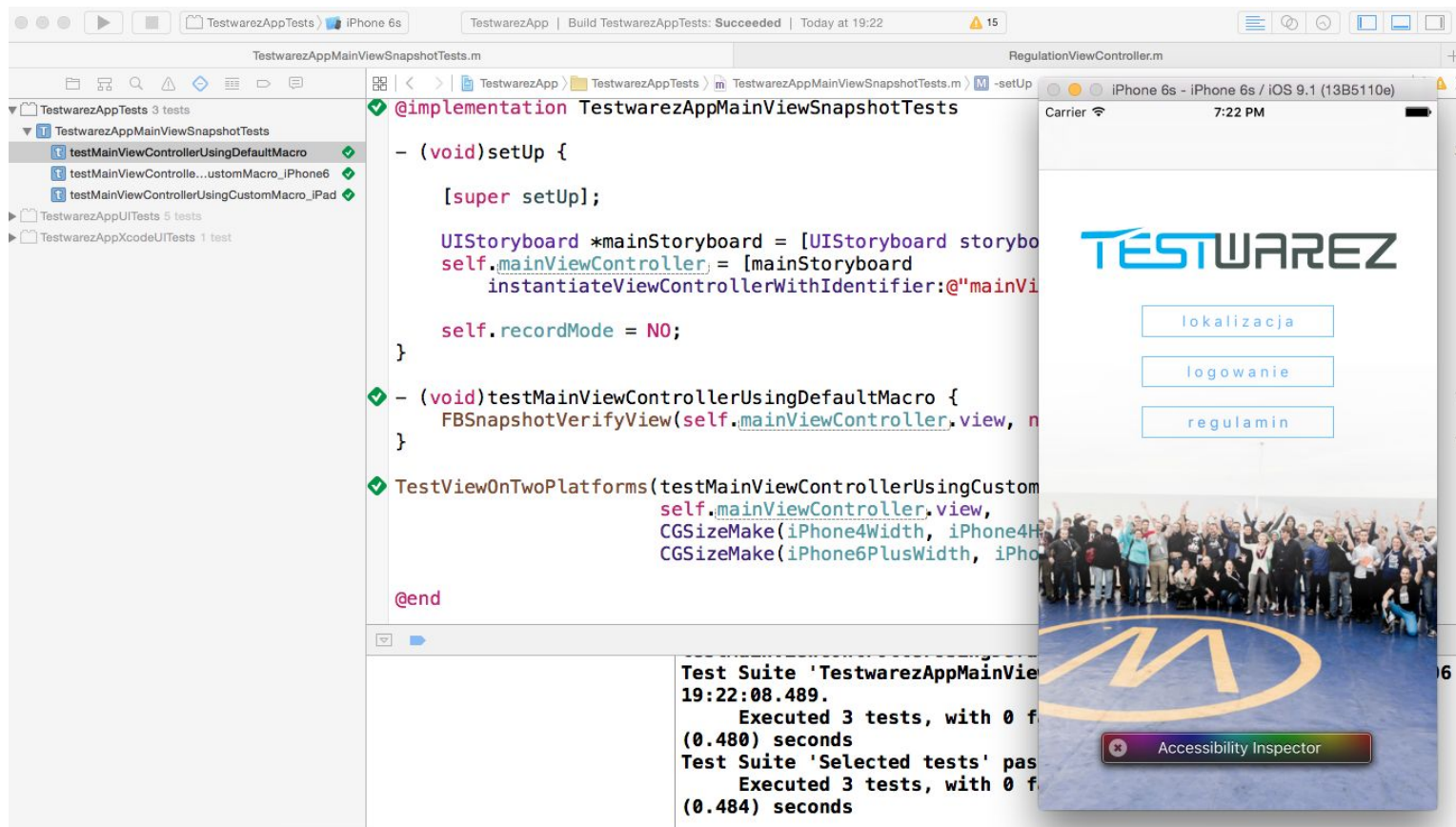
FBSnapshotTestCase 1/2

- **porównywanie** obrazów
- obraz **referencyjny** zapisany w projekcie
- kod testów w **natywnym** języku aplikacji

FBSnapshotTestCase 2/2

- rozszerzenie biblioteki **XCTest**
- środowisko: **Xcode** lub linia komend
- instalacja: **cocoapods** lub github

FBSnapshotTestCase - demo



FBSnapshotTestCase - podsumowanie

- Xcode **target** na testy jednostkowe
- użycie klasy **FBSnapshotTestCase**
- `self.recordMode` i **FBSnapshotVerifyView**
- użycie **macro**
- Xcode **Coverage** > suma testów UI i jednostkowych

Zestawienie narzędzi

	Calabash	KIF	FbSnapshot TestCase	Xcode7 UI Testing
język programowania	Ruby	Obj-C / Swift	Obj-C / Swift	Obj-C / Swift
<i>cross-platform</i>	✓	✓	✓	✓
wsparcie webView	✓	✗	✗	✗
zrównoleglenie testów	✓	✓	✓	✓
szybkość	✓	✓	✓	✓
open source	✓	✓	✓	✗
testy na urządzeniu	✓	✓	✓	✓
CI	✓	✓	✓	✓

Dziękujemy!

Q&A

Bibliografia

- <https://github.com/calabash/calabash-ios>
- <http://groups.google.com/group/calabash-ios>
- <http://calaba.sh/>
- <https://cucumber.io/>

- <https://github.com/kif-framework/KIF>
- <https://github.com/facebook/ios-snapshot-test-case/tree/master/FBSnapshotTestCase>
- <https://cocoapods.org>

- <https://github.com/bielski/TestwarezApp>