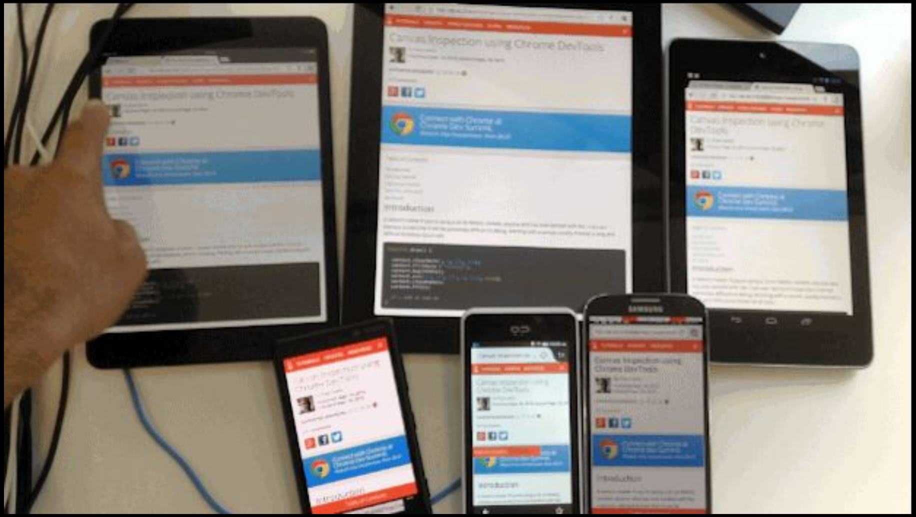
Mobile testing



www.pragmatic.bg

Lector: Viktor Slavchev

E-mail: mr.slavchev@gmail.com
Twitter:@TheTestingTroll
Blog: mrslavchev.com

Types of devices.

- Mobile phones.
- Tablets.
- Smart watches.
- Wearable fitness wristbands.
- Google glass.
- VR headsets.

The two biggest vendors

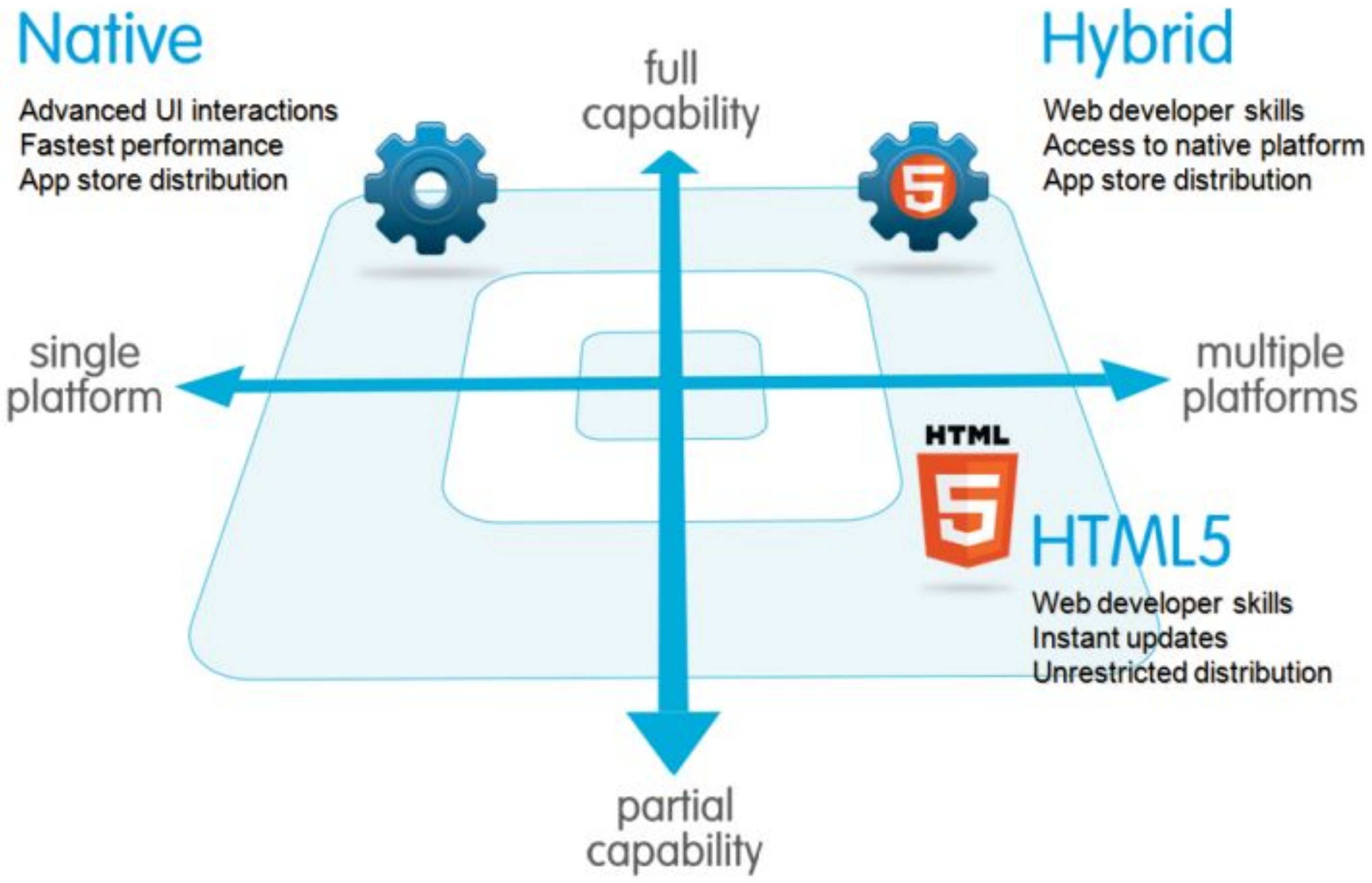
- Apple - close source, paid, elite, different, closed ecosystem, secure.

- Android - open source, open for any platforms and apps. Open for adjustments

What’s a mobile app?

1. Types of mobile apps:
   a. **Native** - iOS, Android, Windows phone etc.
   b. **Mobile web** applications.
   c. **Hybrid** -use html5 components wrapped in native container, to gain advantage of both.
2. Technologies used in mobile app development:
   a. Android - Java and Android SDK
   b. iOS - Objective C, Swift
   c. Other languages- Xamarin(C#), Phone gap, React Native, Native Script (HTML5)
3. Native, HTML5, or Hybrid comparison article.

Native/web/hybrid



Native

Advanced UI interactions
Fastest performance
App store distribution

Hybrid

Web developer skills
Access to native platform
App store distribution

HTML5

Web developer skills
Instant updates
Unrestricted distribution

Differences with web

1. Architecture of hardware. *System on chip.*
2. Connection to electricity. *Limited*
3. Connection to internet. *Wireless / mobile*
4. Data transfer. *Limited*
5. Screen size. *Small and varies*
6. Interaction - specific, per device type and size.
7. Mobility. *Everywhere*
8. Multi purpose.

Sensors on mobile

- *Ambient light sensor* - able to determine how much light there is in the current location and adjust the light balance on your screen for example.
- *Proximity sensor* - detect close objects, like your face, so you don’t dial a number while talking.
- *Acceleration sensor* - detects changes in device orientation, often between landscape and portrait.
- *Gyroscope sensor* - track the actual position of the device, being able to locate it on six axes.
- *Magnetic sensor* - able to detect magnetic fields, mainly used in compass apps.

Sensors on mobile

- *Pressure, Temp, humidity sensors* - able to provide information about altitude, atmosphere temperature and humidity.
- *Location sensor* - also known as GPS.
- *Touchless sensor* - swipe between photos with a gesture, for ex.
- And many more - heart rate sensor, fingerprint sensor, etc.

Mobile terminology.

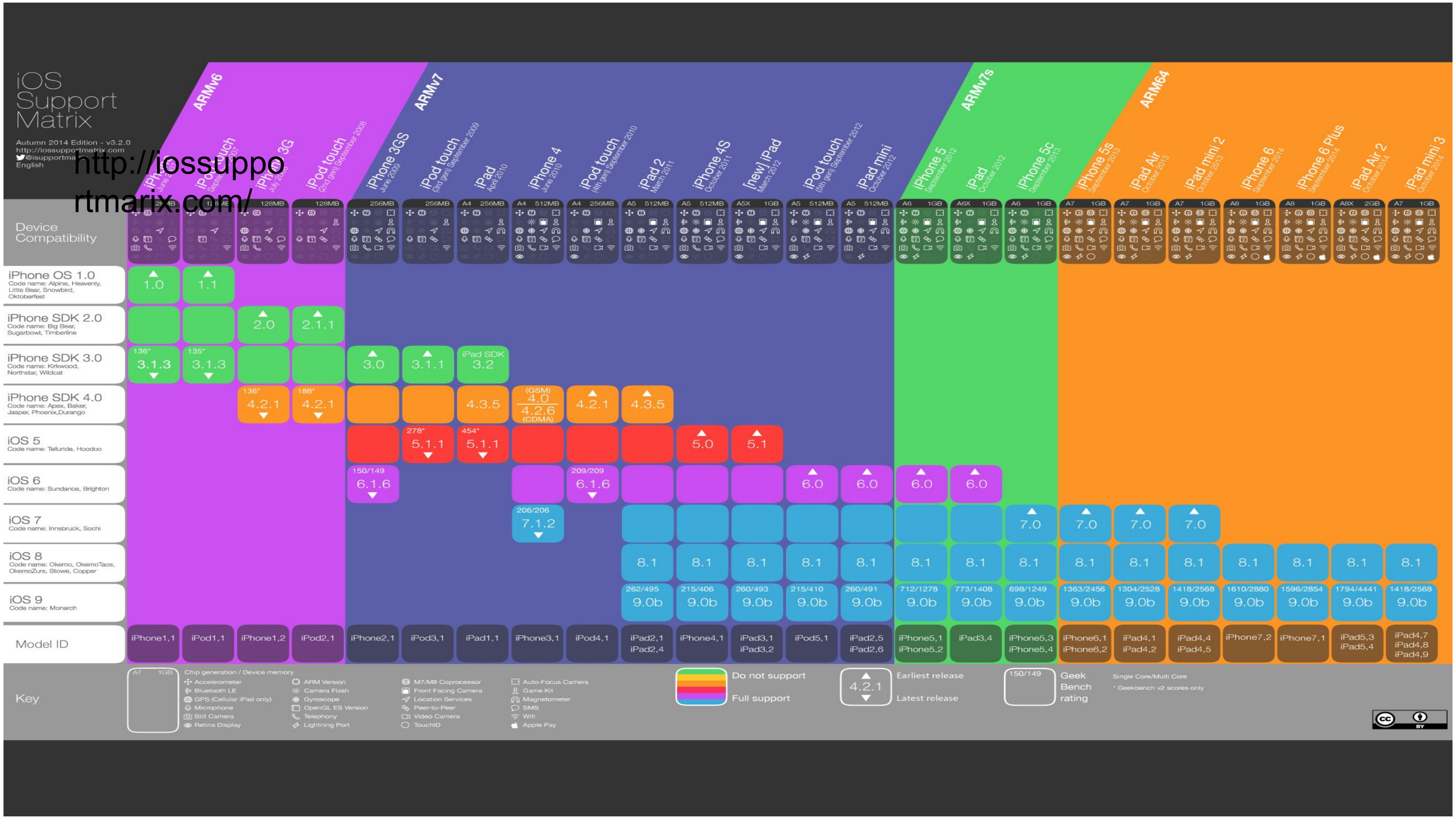
- Not scroll, but swipe.
- Not click, but tap.
- Zoom in, zoom out - pinch in, pinch out.
- Portrait and landscape orientation.
- Views.
- Push notification
- Unique Device Identifier (UDID)
- etc...

Mobile interactions

- **Tap/Double tap** - the mobile alternative of a “click”
- **Press** - tap and hold for a couple of seconds. Press and drag
- **Swipe** - moving left/right or up/down with a sliding action on the screen.
- **Long swipe** - same as the above, but with a longer path.
- **Flick** - really short swipe used for highly interactive elements.
- **Multi-touch** - interaction with the display in more than one point of contact.
- **Tilt** - change device angle on the z axis

iOS device compatibility

Source: IOS support matrix



http://iossuppo
rtmarix.com/

iOS Support Matrix

Autumn 2014 Edition - v3.2.0
http://iossupportmatrix.com
@isupportma
English

	ARMv6				ARMv7										ARMv7s			ARM64						
	iPhone June 2007	iPod touch September 2007	iPhone 3G July 2008	iPod touch (2nd gen) September 2008	iPhone 3GS June 2009	iPod touch (3rd gen) September 2009	iPad April 2010	iPhone 4 June 2010	iPod touch (4th gen) September 2010	iPad 2 March 2011	iPhone 4S October 2011	[new] iPad March 2012	iPod touch (5th gen) September 2012	iPad mini October 2012	iPhone 5 September 2012	iPad October 2012	iPhone 5c September 2013	iPhone 5s September 2013	iPad Air October 2013	iPad mini 2 October 2013	iPhone 6 September 2014	iPhone 6 Plus September 2014	iPad Air 2 October 2014	iPad mini 3 October 2014
Device Compatibility	128MB	128MB	128MB	128MB	256MB	256MB	A4 256MB	A4 512MB	A4 256MB	A5 512MB	A5 512MB	A5X 1GB	A5 512MB	A5 512MB	A6 1GB	A6X 1GB	A6 1GB	A7 1GB	A7 1GB	A7 1GB	A8 1GB	A8 1GB	A8X 2GB	A7 1GB
iPhone OS 1.0 Code name: Alpine, Heavenly, Little Bear, Snowbird, Oktoberfest	▲ 1.0	▲ 1.1																						
iPhone SDK 2.0 Code name: Big Bear, Sugarbowl, Timberline	🟩	🟩	▲ 2.0	▲ 2.1.1																				
iPhone SDK 3.0 Code name: Kirkwood, Northstar, Wildcat	136* 3.1.3 ▼	135* 3.1.3 ▼	🟩	🟩	▲ 3.0	▲ 3.1.1	iPad SDK 3.2																	
iPhone SDK 4.0 Code name: Apex, Baker, Jasper, Phoenix,Durango			136* 4.2.1 ▼	188* 4.2.1 ▼	🟧	🟧	4.3.5	(GSM) 4.0 4.2.6 (CDMA)	▲ 4.2.1	▲ 4.3.5														
iOS 5 Code name: Telluride, Hoodoo					🟥	278* 5.1.1 ▼	454* 5.1.1 ▼	🟥	🟥	🟥	▲ 5.0	▲ 5.1												
iOS 6 Code name: Sundance, Brighton					150/149 6.1.6 ▼			🟪	209/209 6.1.6 ▼	🟪	🟪	🟪	▲ 6.0	▲ 6.0	▲ 6.0	▲ 6.0								
iOS 7 Code name: Innsbruck, Sochi								206/206 7.1.2 ▼		🟦	🟦	🟦	🟦	🟦	🟦	🟦	▲ 7.0	▲ 7.0	▲ 7.0	▲ 7.0				
iOS 8 Code name: Okemo, OkemoTaos, OkemoZurs, Stowe, Copper										8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1	8.1
iOS 9 Code name: Monarch										262/495 9.0b	215/406 9.0b	260/493 9.0b	215/410 9.0b	260/491 9.0b	712/1278 9.0b	773/1408 9.0b	698/1249 9.0b	1363/2456 9.0b	1304/2528 9.0b	1418/2568 9.0b	1610/2880 9.0b	1596/2854 9.0b	1794/4441 9.0b	1418/2568 9.0b
Model ID	iPhone1,1	iPod1,1	iPhone1,2	iPod2,1	iPhone2,1	iPod3,1	iPad1,1	iPhone3,1	iPod4,1	iPad2,1 iPad2,4	iPhone4,1	iPad3,1 iPad3,2	iPod5,1	iPad2,5 iPad2,6	iPhone5,1 iPhone5,2	iPad3,4	iPhone5,3 iPhone5,4	iPhone6,1 iPhone6,2	iPad4,1 iPad4,2	iPad4,4 iPad4,5	iPhone7,2	iPhone7,1	iPad5,3 iPad5,4	iPad4,7 iPad4,8 iPad4,9

Key

A7 1GB Chip generation / Device memory

Accelerometer	ARM Version	M7/M8 Coprocessor	Auto-Focus Camera
Bluetooth LE	Camera Flash	Front Facing Camera	Game Kit
GPS (Cellular iPad only)	Gyroscope	Location Services	Magnetometer
Microphone	OpenGL ES Version	Peer-to-Peer	SMS
Still Camera	Telephony	Video Camera	Wifi
Retina Display	Lightning Port	TouchID	Apple Pay

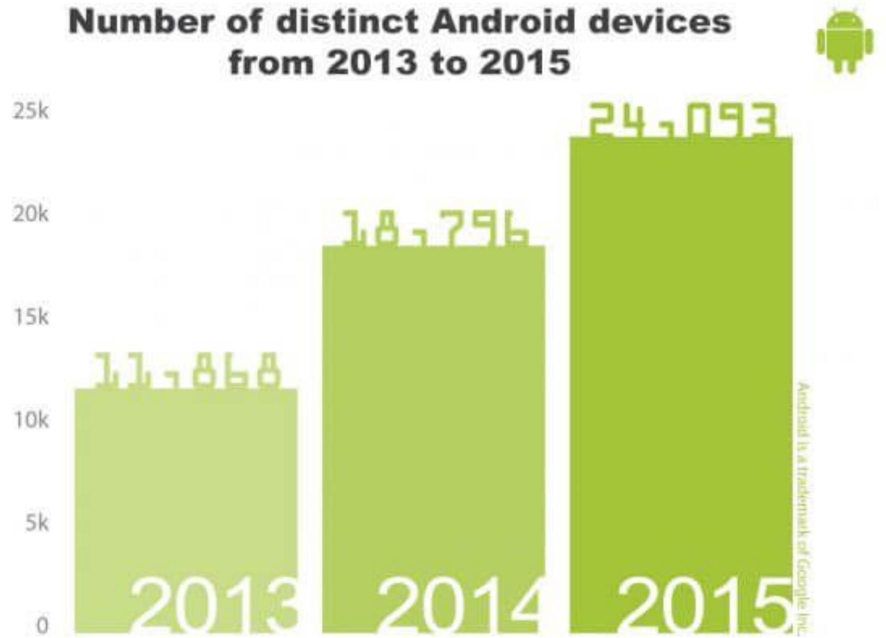
Do not support
Full support

▲ 4.2.1 ▼ Earliest release / Latest release

150/149 Geek Bench rating

Single Core/Multi Core
* Geekbench v2 scores only

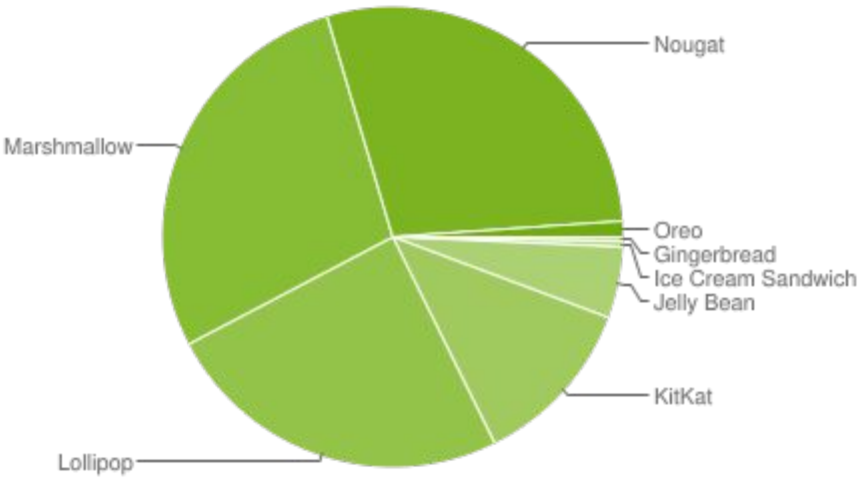
Android device fragmentation

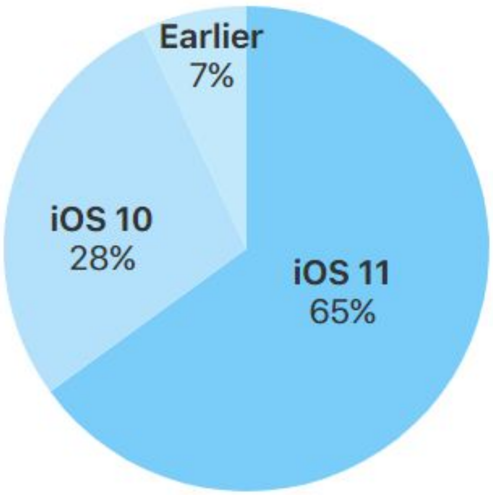


Suggested by Google: 3 important criteria:

- Size of screen.
- Amount of memory.
- Version of SDK.

Fragmentation visualized





As measured by the App Store on January 18, 2018.

Android device fragmentation

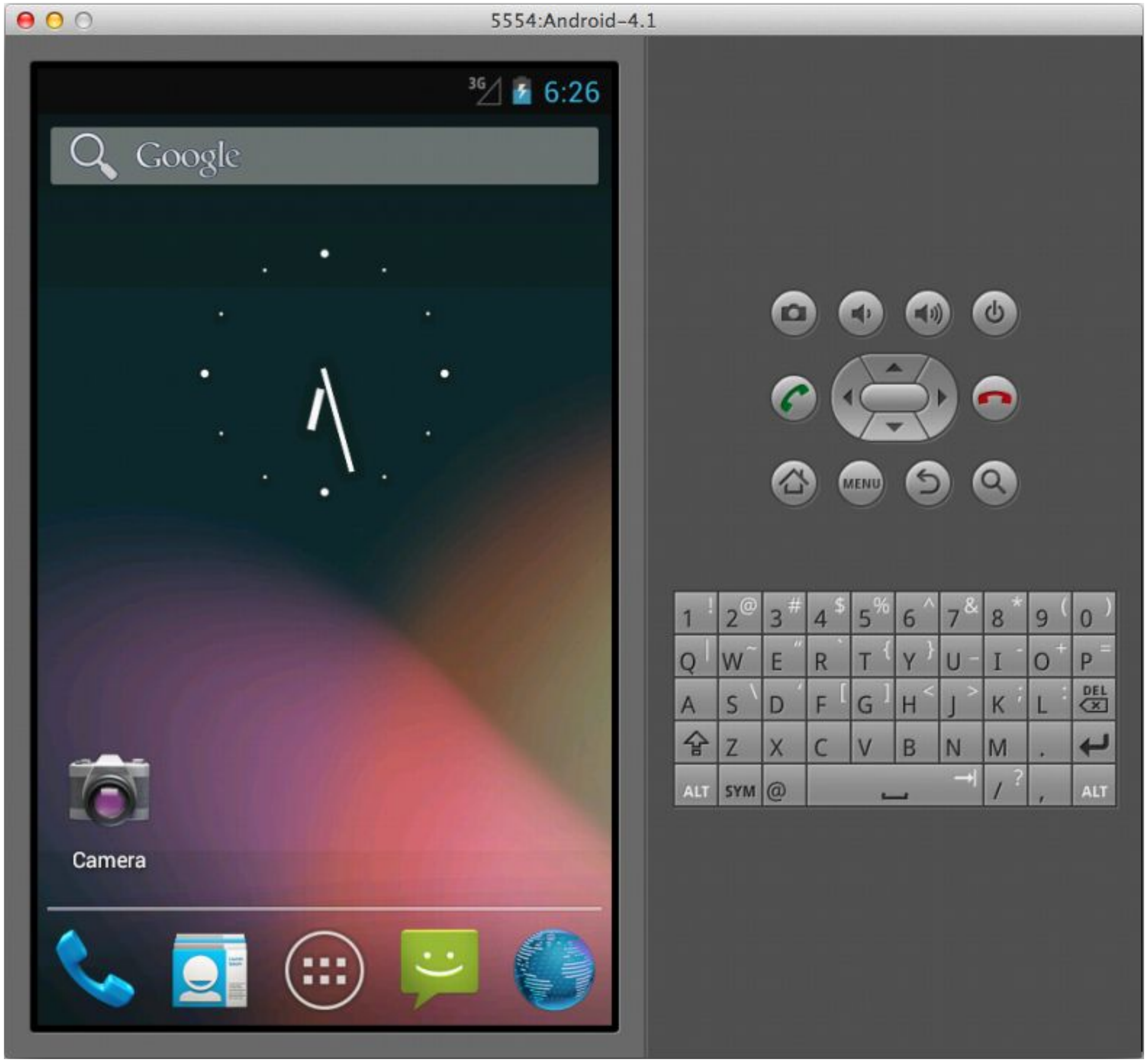
Fighting device fragmentation in testing:

- Using mobile device cloud:
  - Google
  - Oracle
  - Telerik
  - Xamarin
- Crowd testing:
  - Using volunteering testers with their personal devices to test your app:
  - https://99tests.com/
  - uTest

Device emulators.

- For Android - every IDE has some implementation of device emulator. Ex: Device emulator Android Studio
- For iOS - XCode also has device simulator.
- Might use virtual machine image: Virtualbox Image Android 4.4
- Emulators are useful only for quick tests, they have limited abilities and are sometimes slooooow.

Emulator on Android



Emulator vs. simulator

- **Emulator** - aims to emulate not only the software, but also the hardware and the native OS. Programmed using low level programming approaches.
- **Simulator** - mainly focuses on simulating the software, not the hardware. Not so reliable when it comes to debugging.
- In general, both have huge drawbacks. Comparison: Mobile Testing - Emulator Vs Simulator

Installation of apps.

- For Android:
  - From Google play store.
  - By building directly from the IDE to the device.
  - By custom .apk file. (developer settings enabled)
- For iOS:
  - Via Apple app store.
  - By building directly from the IDE to the device.
  - Via iTunes. Supports custom .ipa files, also.
  - Via Test Flight. UDID needed.

How to log a mobile defect.

- Summary
- Severity/Priority
- Description.
- Step by step, actual, expected.
- Device type, OS version, app version
- Logs if any.
- Screenshots, screencasts if any.
- Additional info.
- Found in build.

Debugging tools.

- Important stuff to have when debugging Android:
  - Enable “Developer settings” (Tap on “about device” 7 times)
  - Enable USB debugging.
  - For Windows install OEM device drivers
  - For Linux create entry in /etc/udev/rules.d
  - For MacOS it just works!
  - More detailed info for the above here:http://developer.android.com/tools/device.html

Debugging tools.

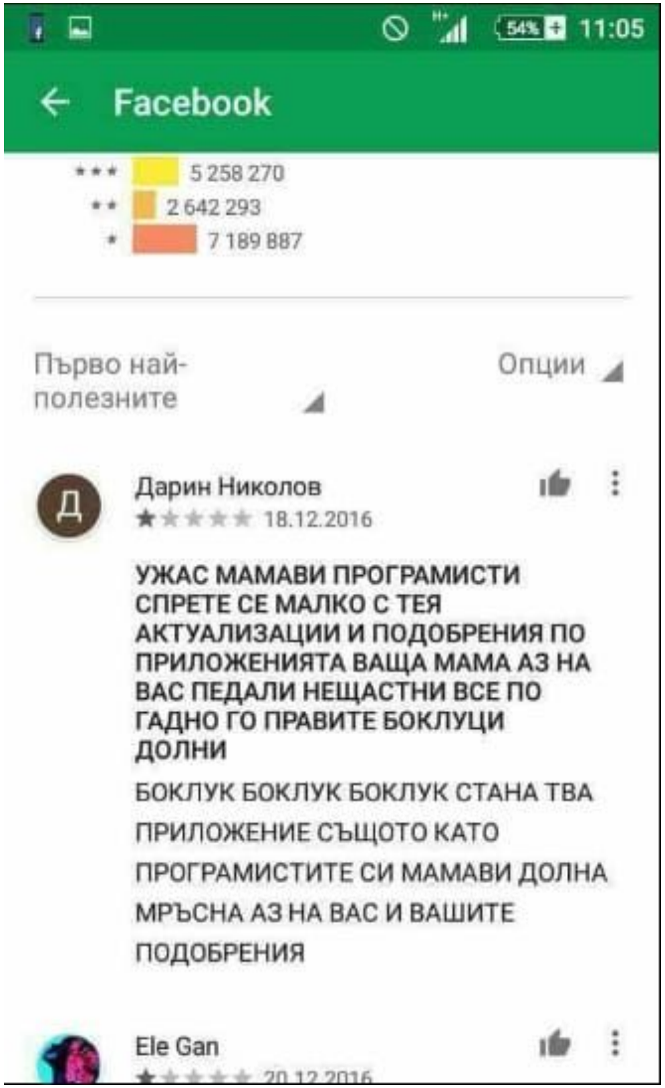
- For Android: Android studio, Eclipse + ADB [Android debugging Android studio]
- For iOS - Xcode, iTunes
- For hybrid and web apps - web proxies like Fiddler and Charles could be used.
- **Live demo!**

Risk areas

- **Deletable offensive** - any reason in the app that will make your users delete it. This might be, but not limited to:
  - App crashes at startup
  - App crashes somewhere along the flow
  - Poor performance - app hangs or is unresponsive for long time.
  - Poor usability - interface is non-intuitive, user feels lost using it or it simply takes too long to perform a trivial action.
  - App requires access to personal data.
  - It’s a copycat - has no value for the user.

Risk areas

- **Rantable offensive** - A reason for the user not to delete the app, but instead give negative feedback or low rating.
- Might be:
  - Negative review in app store.
  - Low rating.
  - Write a public statement or blog.



Дарин Николов
★★★★★ 18.12.2016

УЖАС МАМАВИ ПРОГРАМИСТИ СПРЕТЕ СЕ МАЛКО С ТЕЯ АКТУАЛИЗАЦИИ И ПОДОБРЕНИЯ ПО ПРИЛОЖЕНИЯТА ВАЩА МАМА АЗ НА ВАС ПЕДАЛИ НЕЩАСТНИ ВСЕ ПО ГАДНО ГО ПРАВИТЕ БОКЛУЦИ ДОЛНИ

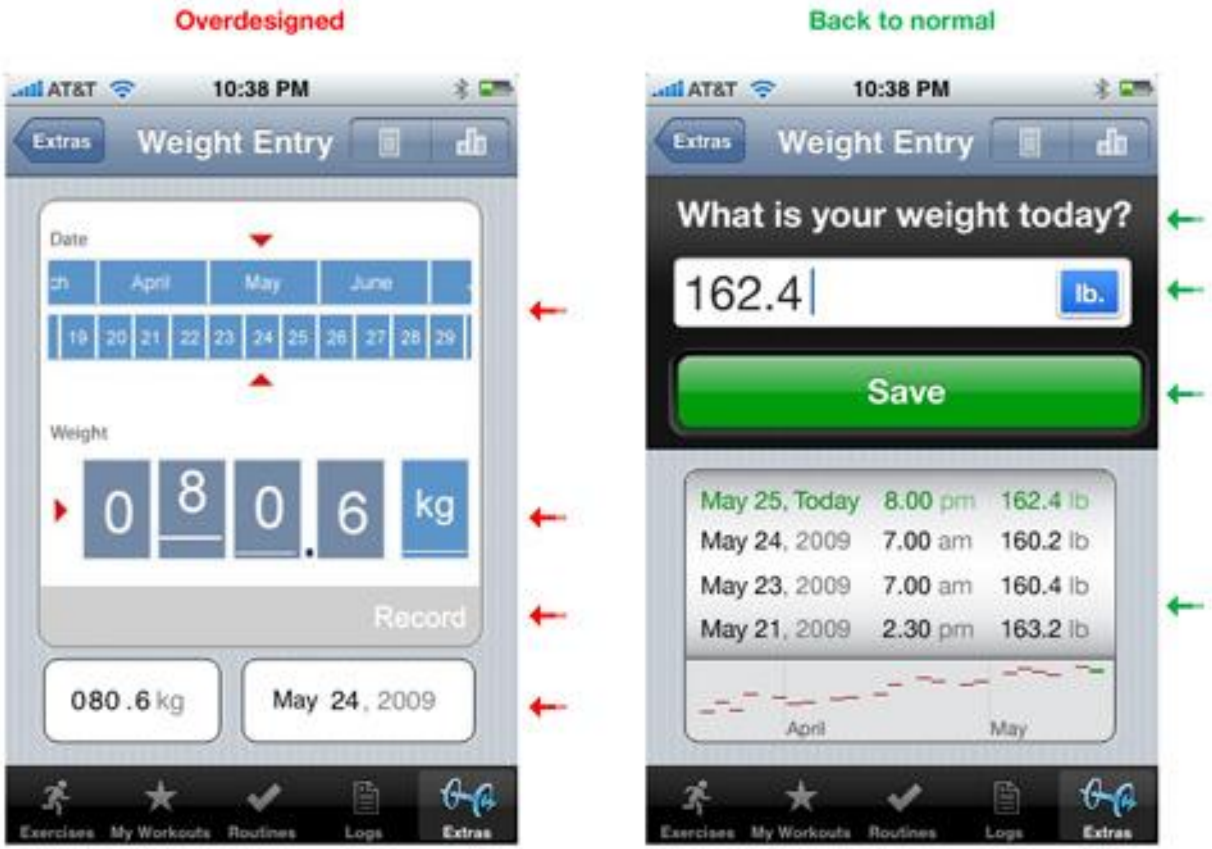
БОКЛУК БОКЛУК БОКЛУК СТАНА ТВА ПРИЛОЖЕНИЕ СЪЩОТО КАТО ПРОГРАМИСТИТЕ СИ МАМАВИ ДОЛНА МРЪСНА АЗ НА ВАС И ВАШИТЕ ПОДОБРЕНИЯ

Ele Gan
★★★★★ 20.12.2016

Usability testing.

- Critical for mobile testing.
- Different approach for design on tablet, mobile and wearables.
- Consider landscape and portrait uses in tablet and mobile.
- Consider working with one hand on a mobile.
- Reuse user data when possible. Ex. login with facebook, google etc. (permission).

Some epic fails in mobile

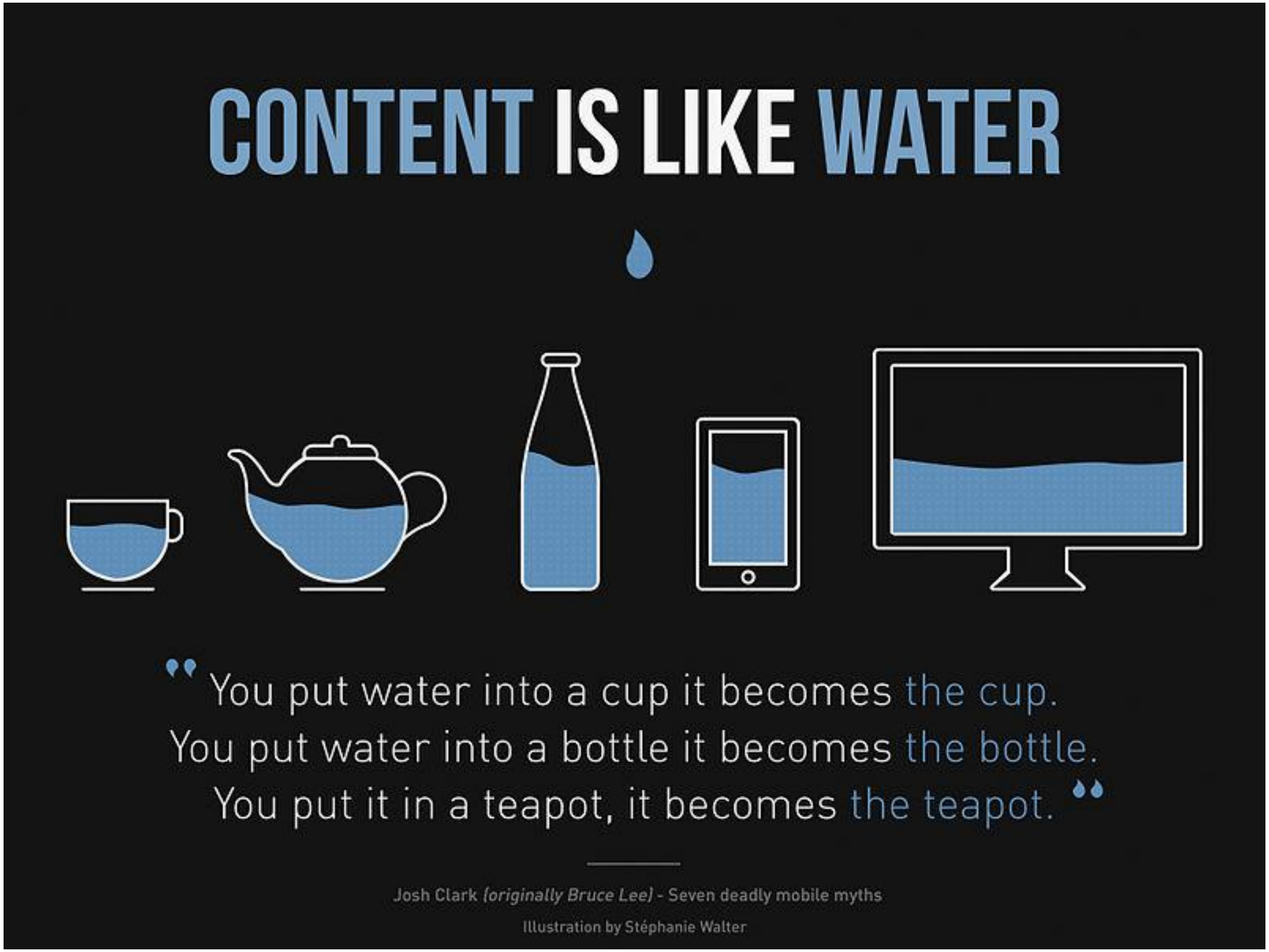


May 25, Today	8.00 pm	162.4 lb
May 24, 2009	7.00 am	160.2 lb
May 23, 2009	7.00 am	160.4 lb
May 21, 2009	2.30 pm	163.2 lb

Usability testing.

- Consider interaction is totally different, mouse is more precise than a finger.
- Make sure the client **doesn’t have to learn**, how to use the app.
- Emotional response is important for usability.
- For the good or bad, usability testing is part that slowly moves off the QA field to the design field - UX design.

Responsive design



CONTENT IS LIKE WATER

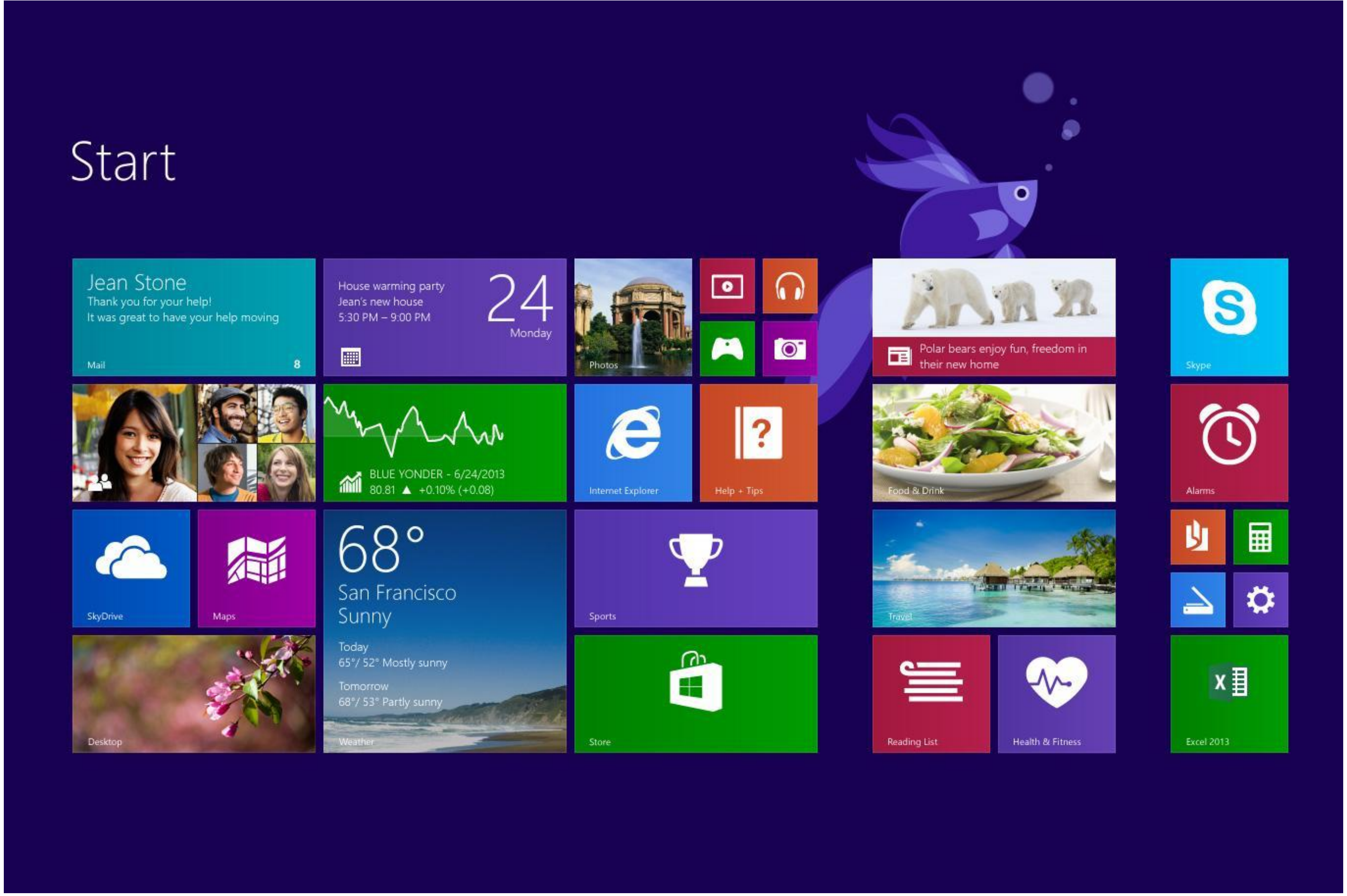
"You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot."

Josh Clark *(originally Bruce Lee)* - Seven deadly mobile myths

Illustration by Stéphanie Walter

Epic fails in usability.

- Windows 8 and http://wtfmobileweb.com/



Jean Stone
Thank you for your help!
It was great to have your help moving

House warming party
Jean's new house
5:30 PM – 9:00 PM
24
Monday

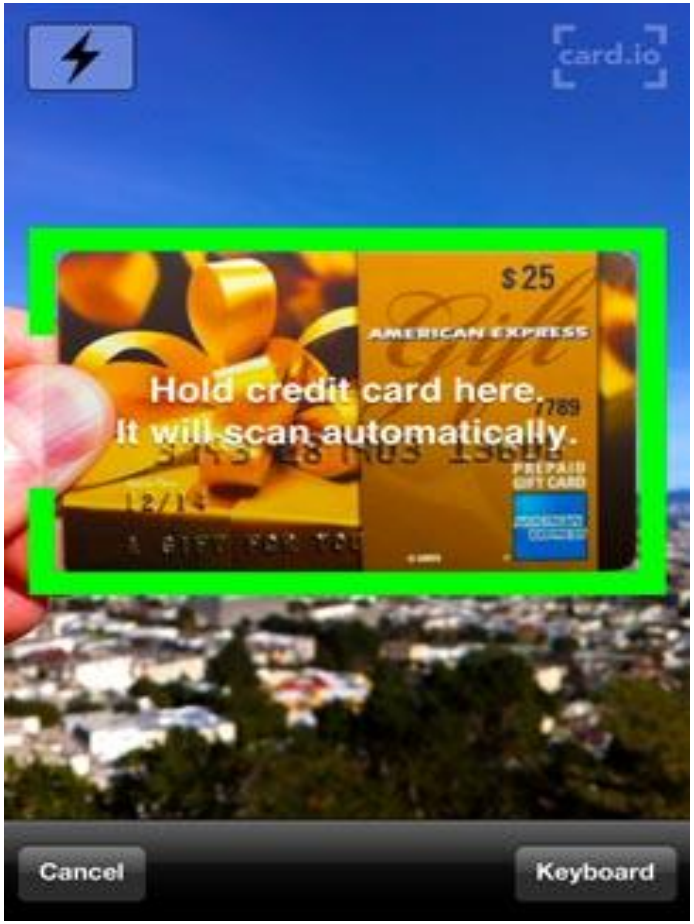
Polar bears enjoy fun, freedom in their new home

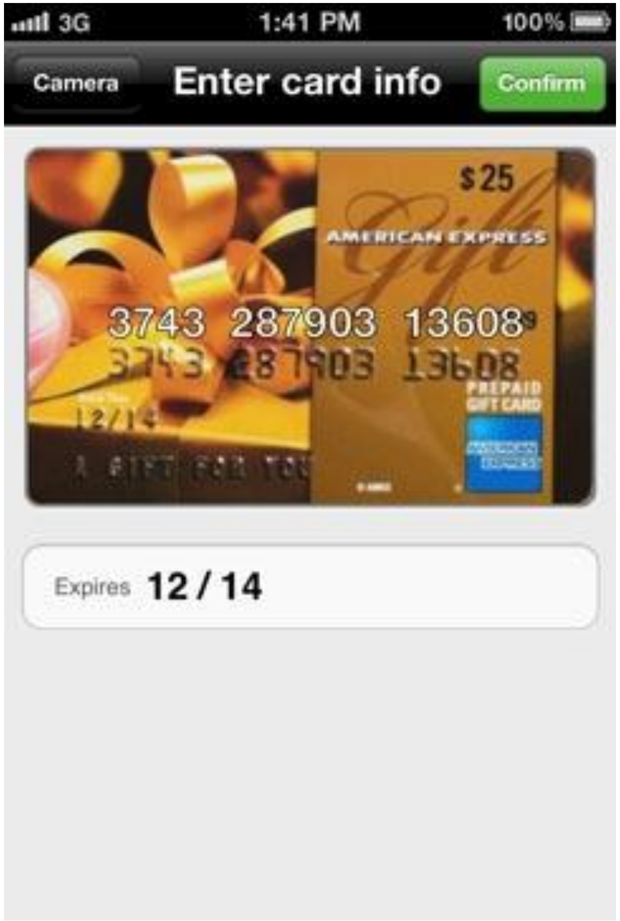
BLUE YONDER - 6/24/2013
80.81 ▲ +0.10% (+0.08)

68°
San Francisco
Sunny
Today
65°/ 52° Mostly sunny
Tomorrow
68°/ 53° Partly sunny

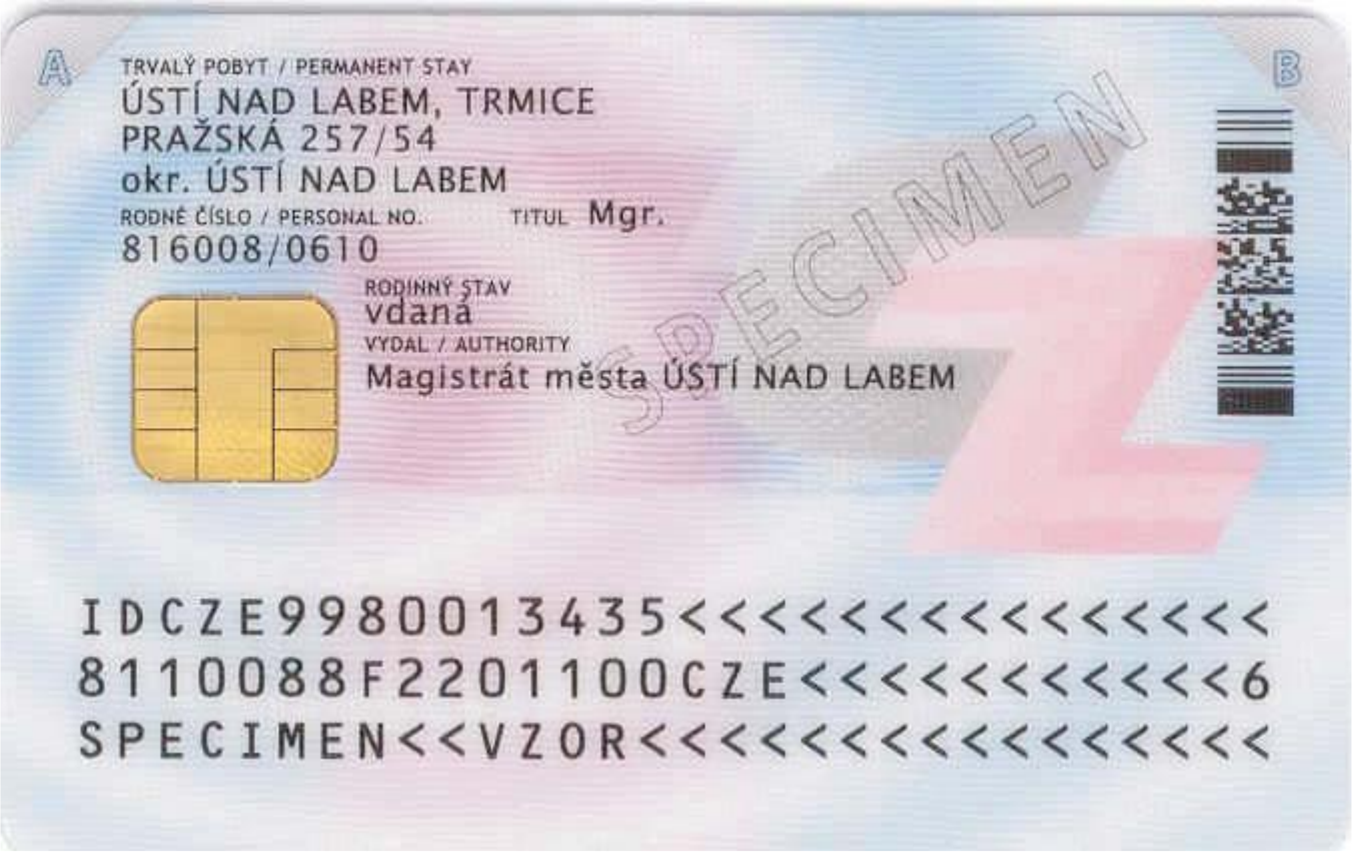
Some good usability decissions

https://www.card.io/ Uber and Paypal





Some good decisions



A

B

TRVALÝ POBYT / PERMANENT STAY
ÚSTÍ NAD LABEM, TRMICE
PRAŽSKÁ 257/54
okr. ÚSTÍ NAD LABEM

RODNÉ ČÍSLO / PERSONAL NO.
816008/0610

TITUL Mgr.

RODINNÝ STAV
vdaná

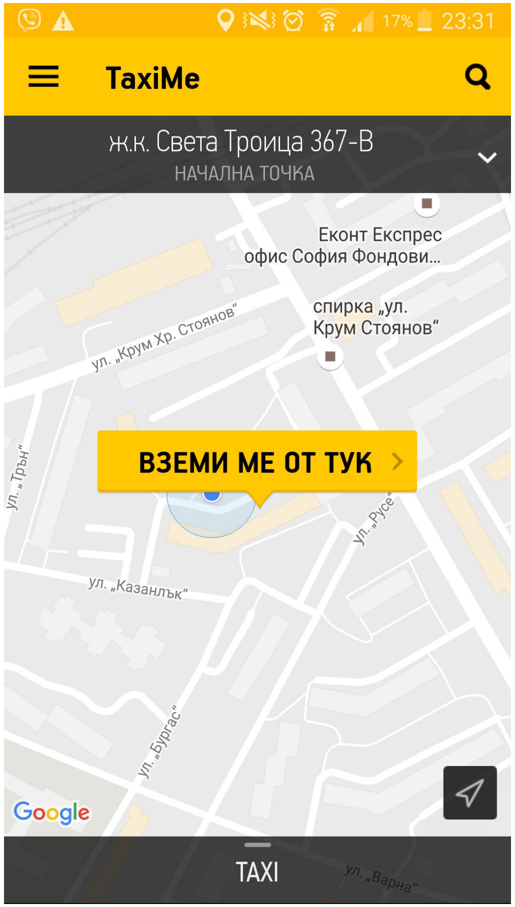
VYDAL / AUTHORITY
Magistrát města ÚSTÍ NAD LABEM

SPECIMEN

```
IDCZE99800134 35<<<<<<<<<<<<<<<
8110088F2201100CZE<<<<<<<<<<<6
SPECIMEN<<VZOR<<<<<<<<<<<<<<<<
```

Some good usability decissions

Using GPS data to locate address for you - TaxiMe and Dominos.



17% 23:31

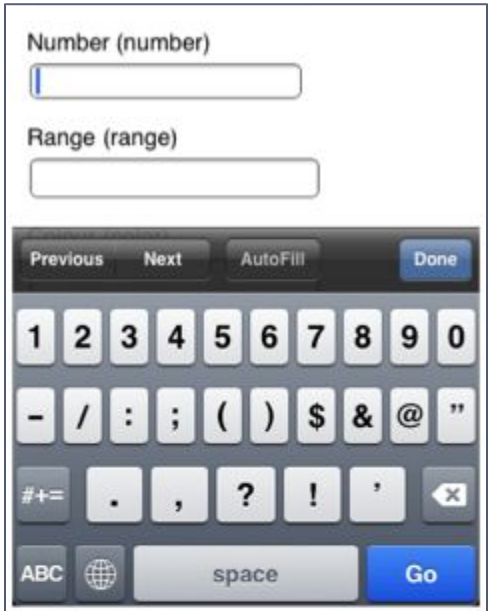
TaxiMe

ж.к. Света Троица 367-В
НАЧАЛНА ТОЧКА

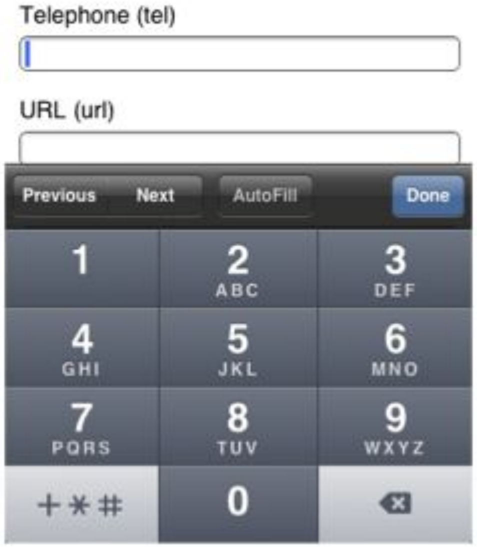
ВЗЕМИ МЕ ОТ ТУК

TAXI

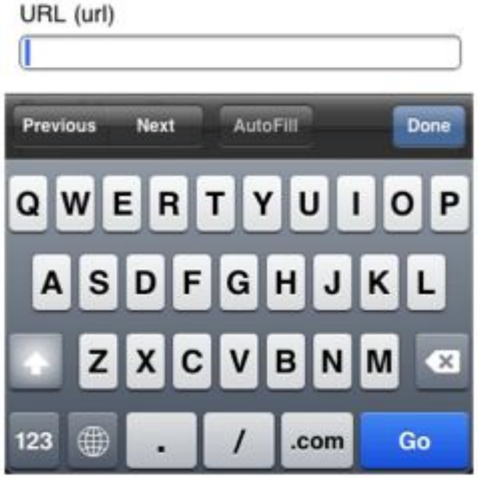
Provide the right keyboards



Keyboard when
<input type="number">



Keypad when
<input type="tel">



Keyboard when
<input type="url">

Performance

Performance

- Performance is normally an attribute of the back end
- In a normal client-server architecture we should have a “smart server” and a “dummy client” (smartphone)
- All expensive, memory and CPU consuming actions should be pulled at the back end.
- The app should deal only with user interactions and displaying information.
- We can assess performance by looking for slow loading components, operations with huge amounts of data, a lot of images, graphics, large lists.
- Fast interactions, a lot of applications running at the same time.

Use of resources

- Aims test if the application doesn’t get too greedy on resources.
- Be careful for memory leaks - hard to reproduce and debug. With strong impact
- Battery consumption gets impact by:
  - Many calls to web service.
  - Use of the screen.
  - Use of the camera.
  - Use of GPS location tracker
  - Use of other data streams.

Testing interruptions.

- Mobile phone is not a single threaded device.
- We need to make sure our app acts nice, when interrupted by processes like:
    - Call, incoming message.
    - Alert from calendar.
    - Alarm.
    - Email.
    - Push notification from another app.
    - Text message from instant messenger.

Testing connectivity.

- The app needs to behave correctly under circumstances of low/no connectivity.
- Switching between different wifi networks.
- Switching between wifi and mobile network.
- Poor or no network - the elevator test.
- Apple devices have built-in functionality for that purpose: How To Simulate A Bad Network Connection On Your iOS Device and Simulator
- In Android we can use emulator.

Mobile testing cheat sheet

- Look out for bugs.
- Think about the possible mobile interactions
- Think about the possible usages of the application.
- Think about the possible places and social situations where the application could be used in.
- Think about network usage
- Think about interruptions

Quest - Birlibam

Apple

Android

- What oracles can we use?
- What are mobile specific usages should we test?
- What situations/contexts will this be used at?
- What integrations do we have?
- If you find any bugs, take notes in additional file.

Questions