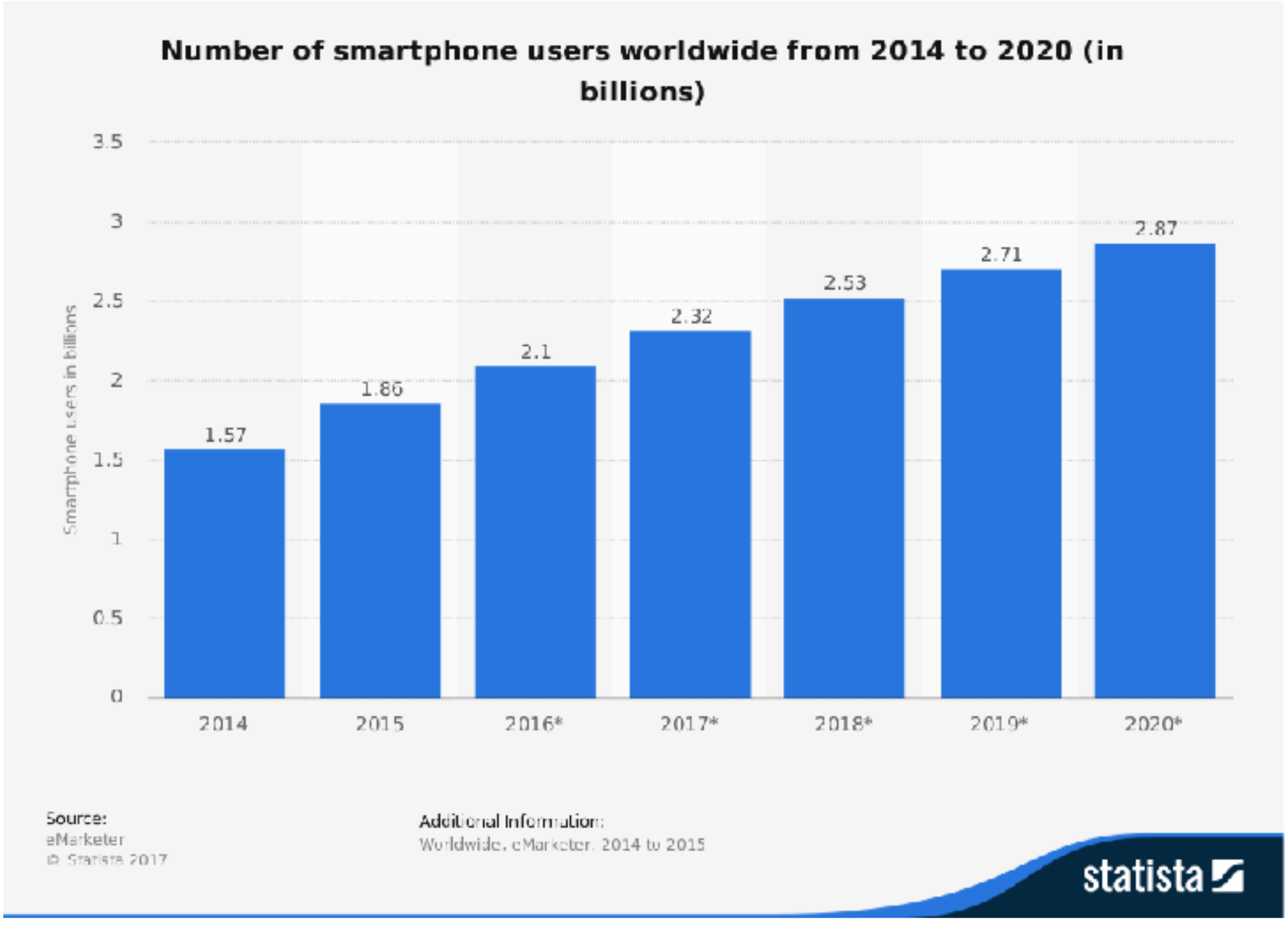
María Gómez

Software Engineering Course - Summer Semester 2017

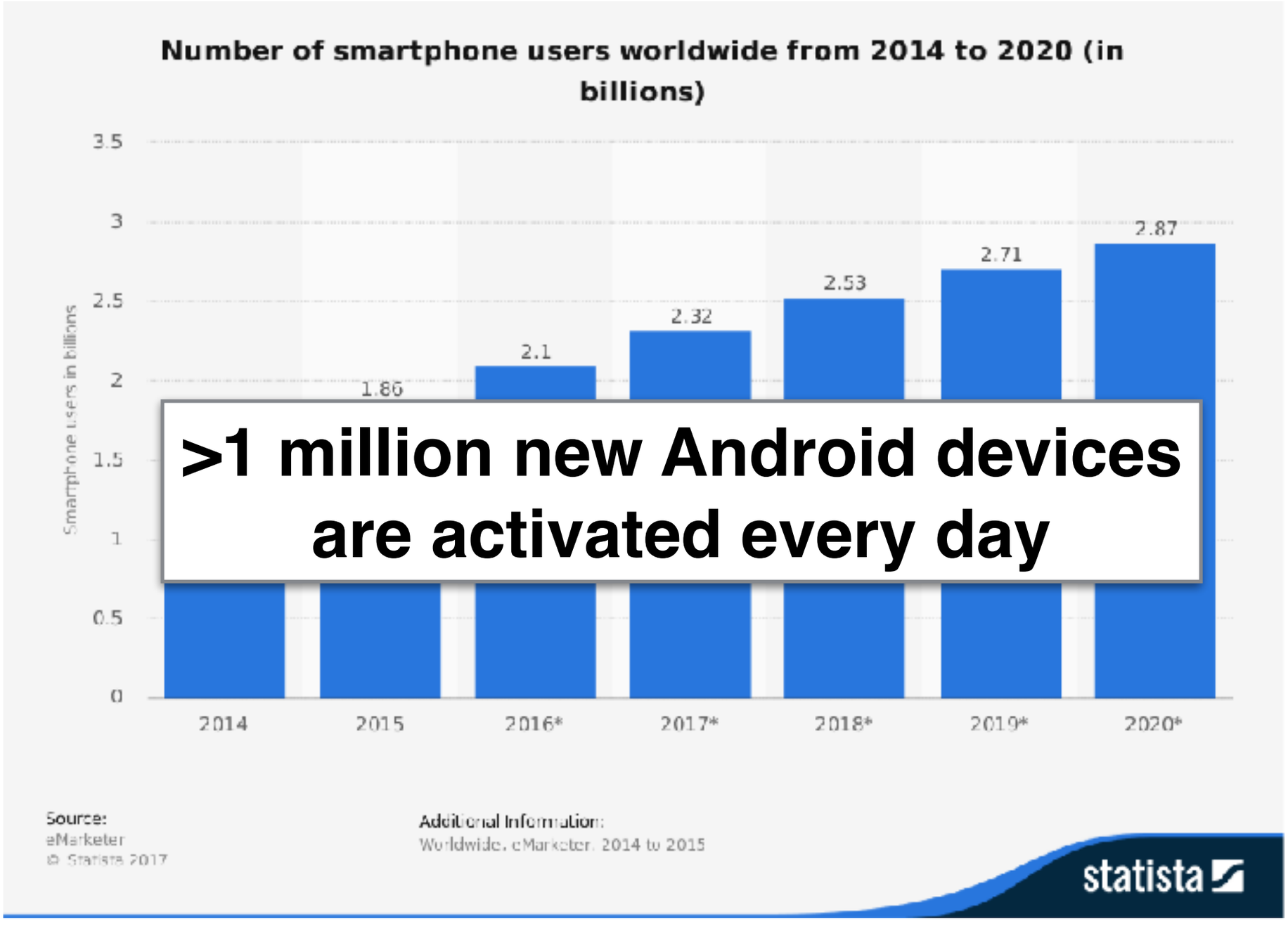
Mobile Market Today

Growth of mobile devices



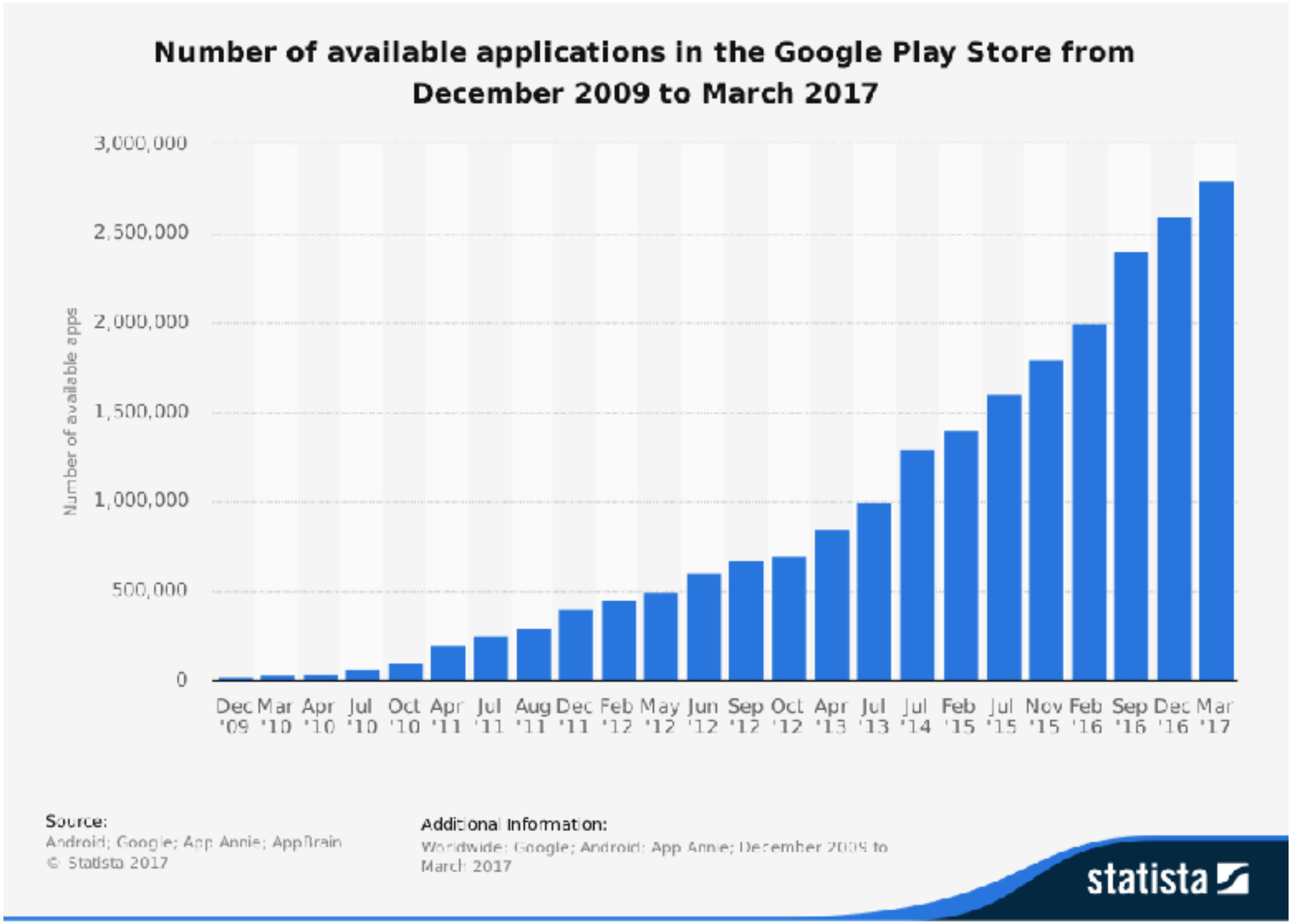
Mobile Market Today

Growth of mobile devices



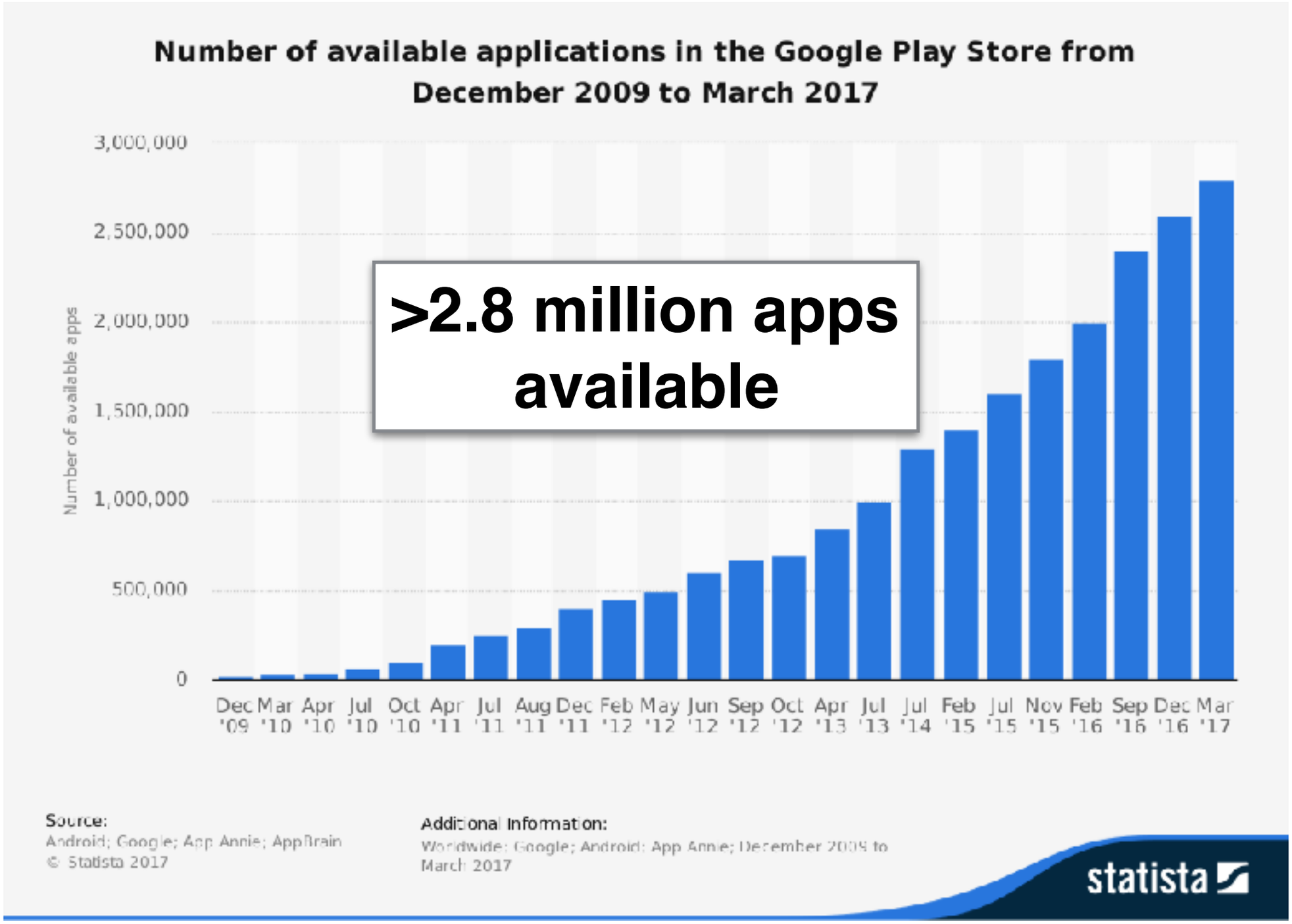
Mobile Market Today

Growth of mobile apps



Mobile Market Today

Growth of mobile apps



What makes mobile apps different from web/desktop apps?

VS

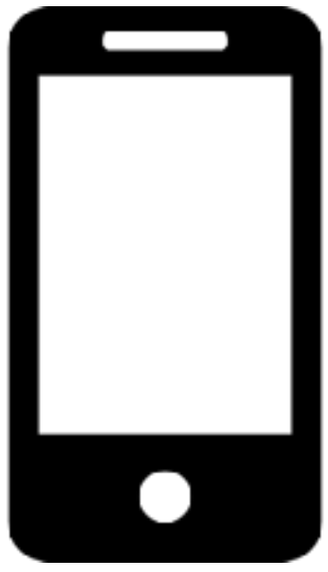
Web/Desktop Apps

- Always plugged in
- Big screen
- Physical keyboard and mouse
- Users seated with attention
- Reliable & fast network

Web/Desktop Apps

Mobile

- ~~Always plugged in~~
- ~~Big screen~~
- ~~Physical keyboard and mouse~~
- ~~Users seated with attention~~
- ~~Reliable & fast network~~



Mobile App Challenges

- Device limitations
  Limited power, computations, memory, screen

- Sensors
  GPS, accelerometer, gyroscope, compass, light, finger-print, proximity…

- Mobility

- Context

- Privacy and security of user information

Opportunities

Opportunities as result of constraints

- Context-detection
- Context-aware behaviour
- Information available anytime-anyplace
- Location-awareness
- Real-time location-based experiences
- Augmented reality
- Virtual reality

Mobile Development Considerations

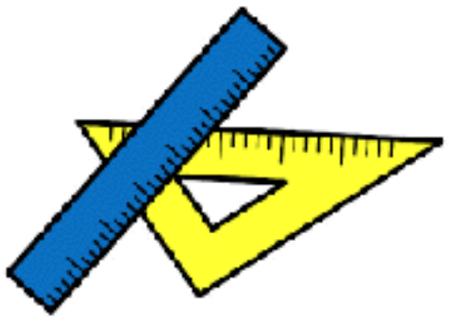
- Distribution channels (app stores)
- Fast time-to-market
- Huge global competition
- Short release cycles
- Development teams (1 person)

Mobile Sw Development Lifecycle

Inception

Monitor & Updates

Release



Design

Development

Test

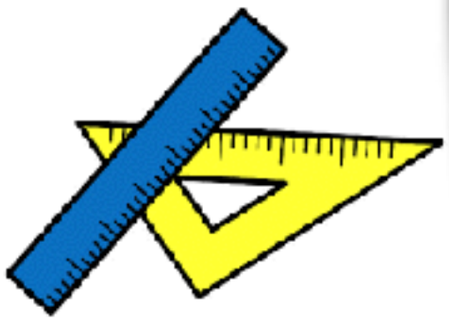
Mobile Sw Development Lifecycle

Inception

Release

These phases can be used with different methodologies (e.g., Agile, Spiral…)

Tendency to agile and ignore formal methodologies



Design

Development

Test

Mobile Sw Development Lifecycle

Inception

Monitor & Updates

Release

Design

Development

Test

Inception

- All apps start with an idea
- Questions to consider*:
  - **Competitive Advantage**. *Are there similar apps? How does this app differentiate from others?*
  - **Value**. *What value does this app bring to users? How will users use it?*
  - **Form/Mobility**. *How will this app work in a mobile form factor? How can I add value using mobile technologies such as location awareness, camera, etc.?*

*https://developer.xamarin.com/guides/cross-platform/getting_started/introduction_to_mobile_sdlc/

Mobile Sw Development Lifecycle

Inception

Monitor & Updates

Release

Design

Development

Test

Design

User Interface and **Responsiveness**

are **critical**!

User Experience (UX) Design

- User-centered design

    1. Identify **Personas**

        Personas = Proxy for group of users of the app

    2. Identify **Use Cases**

        Use Cases = Scenarios *when*, *where* and *how* a persona will use the app

    3. Define **Feature Lists**

Wireframing

High-level flow of the app screens

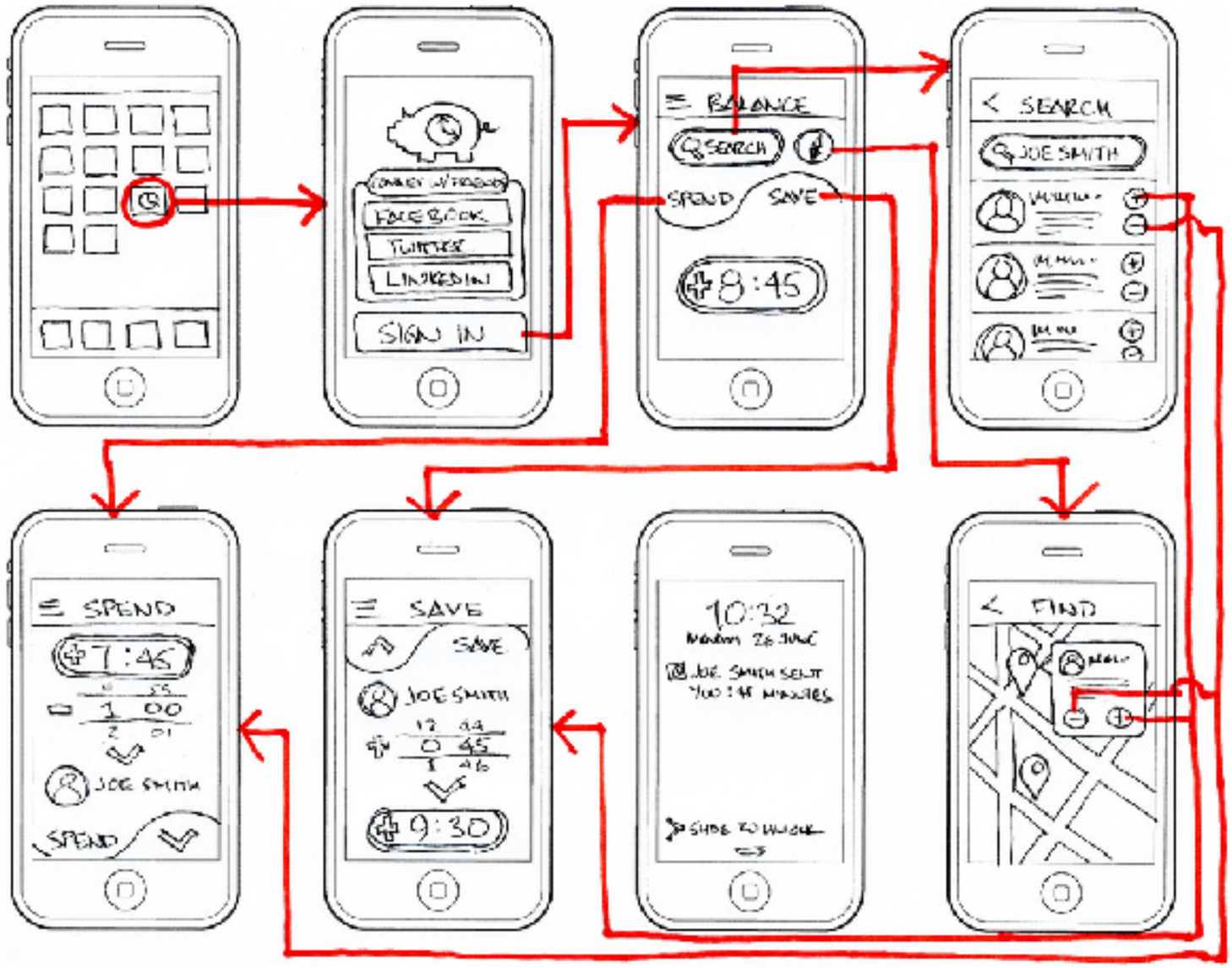


*Image extracted from: https://www.appfutura.com/blog/mobile-app-development-report-wireframes-the-key-to-usability/

Wireframing

How to create one?

Pencil & Paper

Tools

MockFlow

balsamiq

Mockingbird

Visio

...

Mobile User Constraints

Constraints should be respected when designing the app

*Udacity: UX Design for mobile developers (by Google)

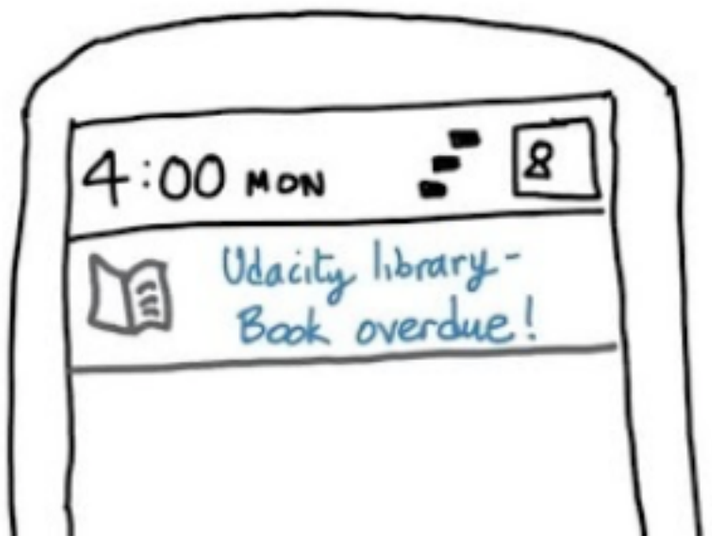
Finite Data & Battery

- Data & Power Consumption are critical considerations
    - Impact on the entire app design process
- Get the adequate Data and Memory model
    - Consuming data, discarding data, managing scarce memory

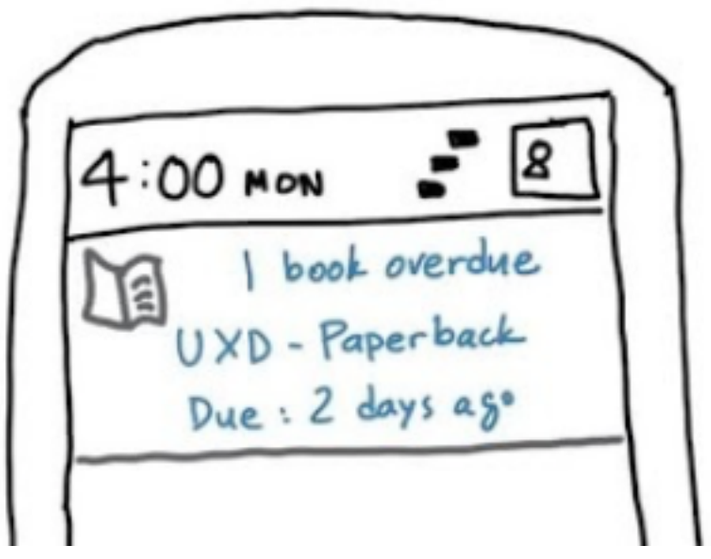
Divided Attention

Respecting Divided Attention

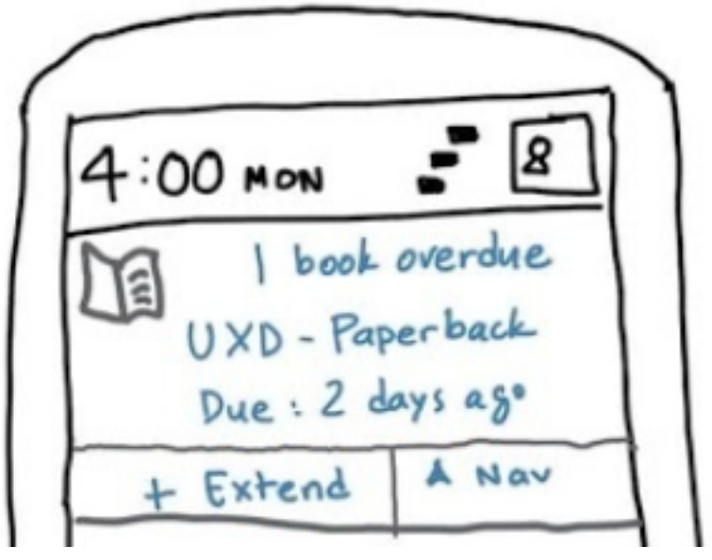
Which is the best way to notify?



1



2



3

*Image extracted from Udacity: UX Design for mobile developers (by Google)

Handedness

Handedness

Which is the best screen when user uses the mobile with one hand?

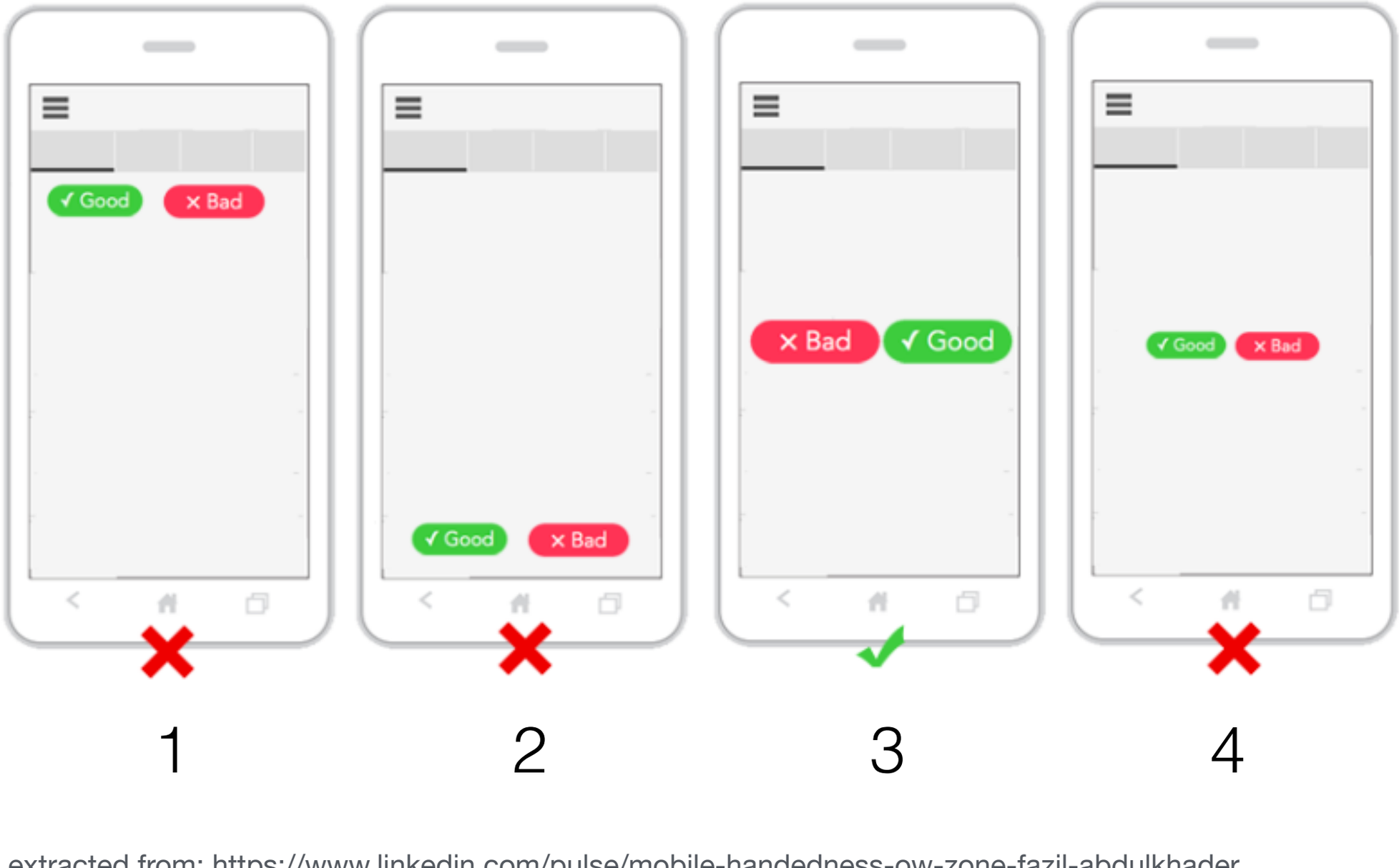


*Image extracted from: https://www.linkedin.com/pulse/mobile-handedness-ow-zone-fazil-abdulkhader

Handedness: Respecting OW Zone

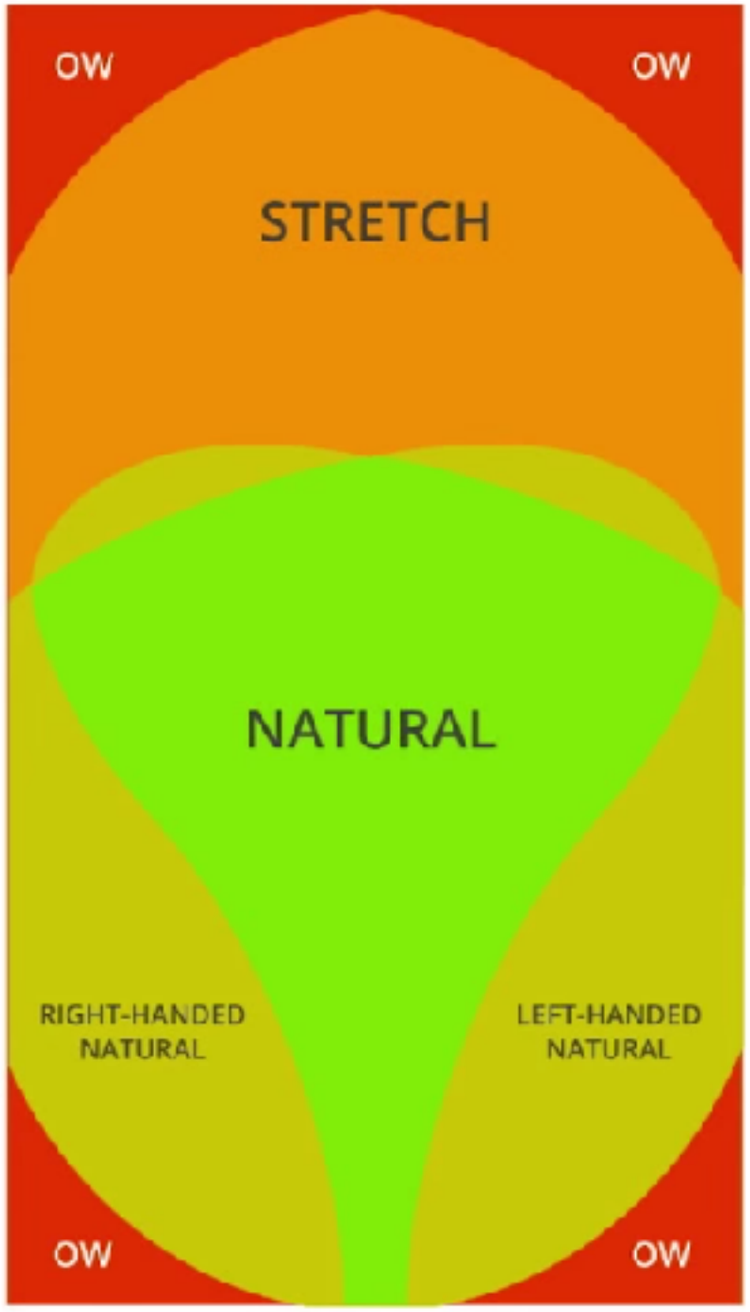


*Image extracted from Udacity: UX Design for mobile developers (by Google)

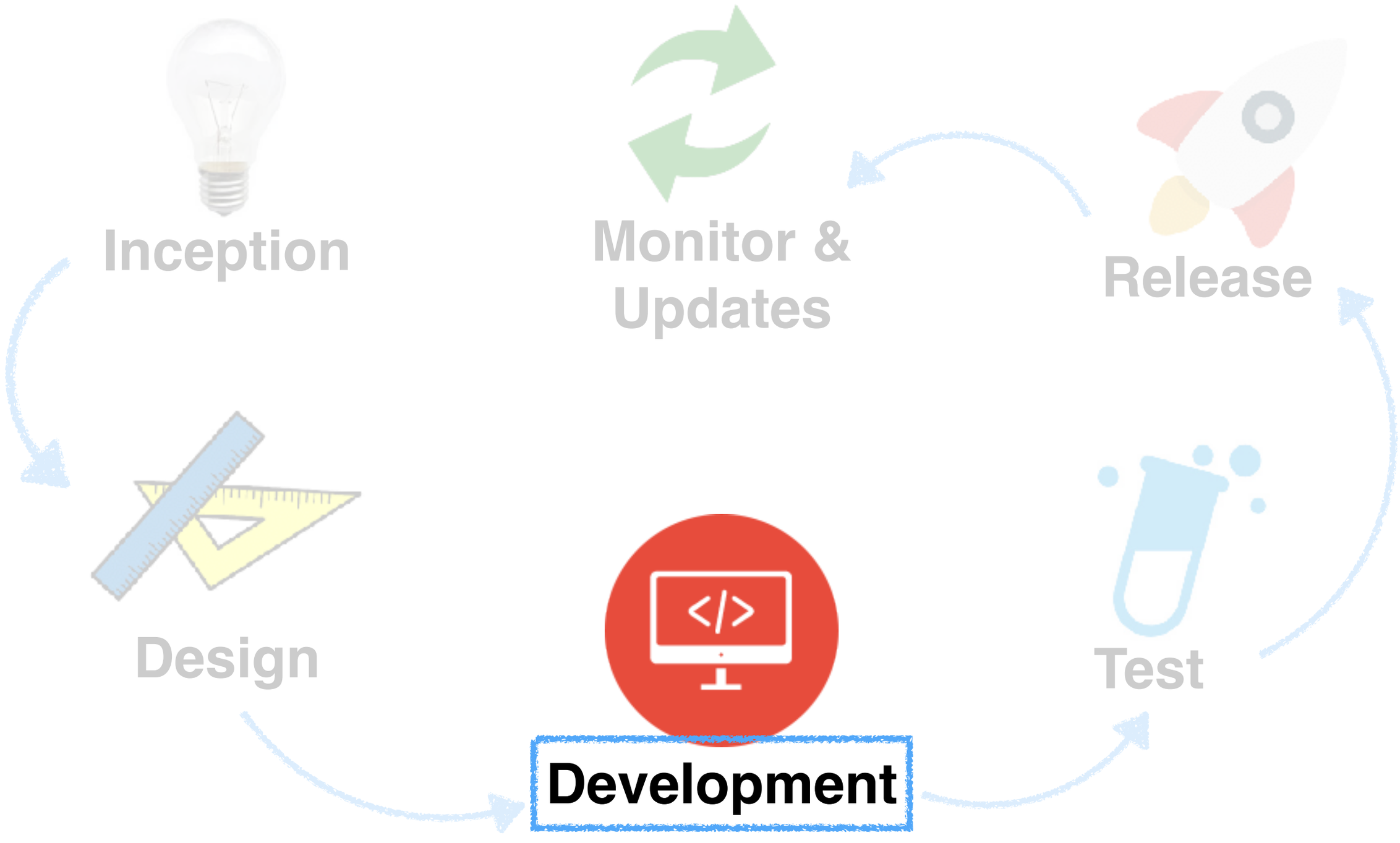
Small Screens

- Don’t overwhelm the user with much information
- Split screens
- Images over text

Unreliable Networks

- Assume that communications will often fail
  - Recover automatically
  - Defensive design for a good user experience

Mobile Sw Development Lifecycle



Development

Which platform?

Development

Which platform?

Questions to consider:

- *Which platform has more users?*
- *Which platform has more competitors?*
- *Which platform is more expensive to develop for?*
- *Which platform makes more money for developers?*

Development Tools

Android

Android Studio
https://developer.android.com/studio/index.html

iPhone

Apple Developer SDK and Tools
http://developer.apple.com

Windows Phone

Visual Studio IDE and Phone SDK
http://developer.windowsphone.com

Blackberry

http://developer.blackberry.com/blackberry_world/

Cross-platform Development

Target multiple platforms with one code base

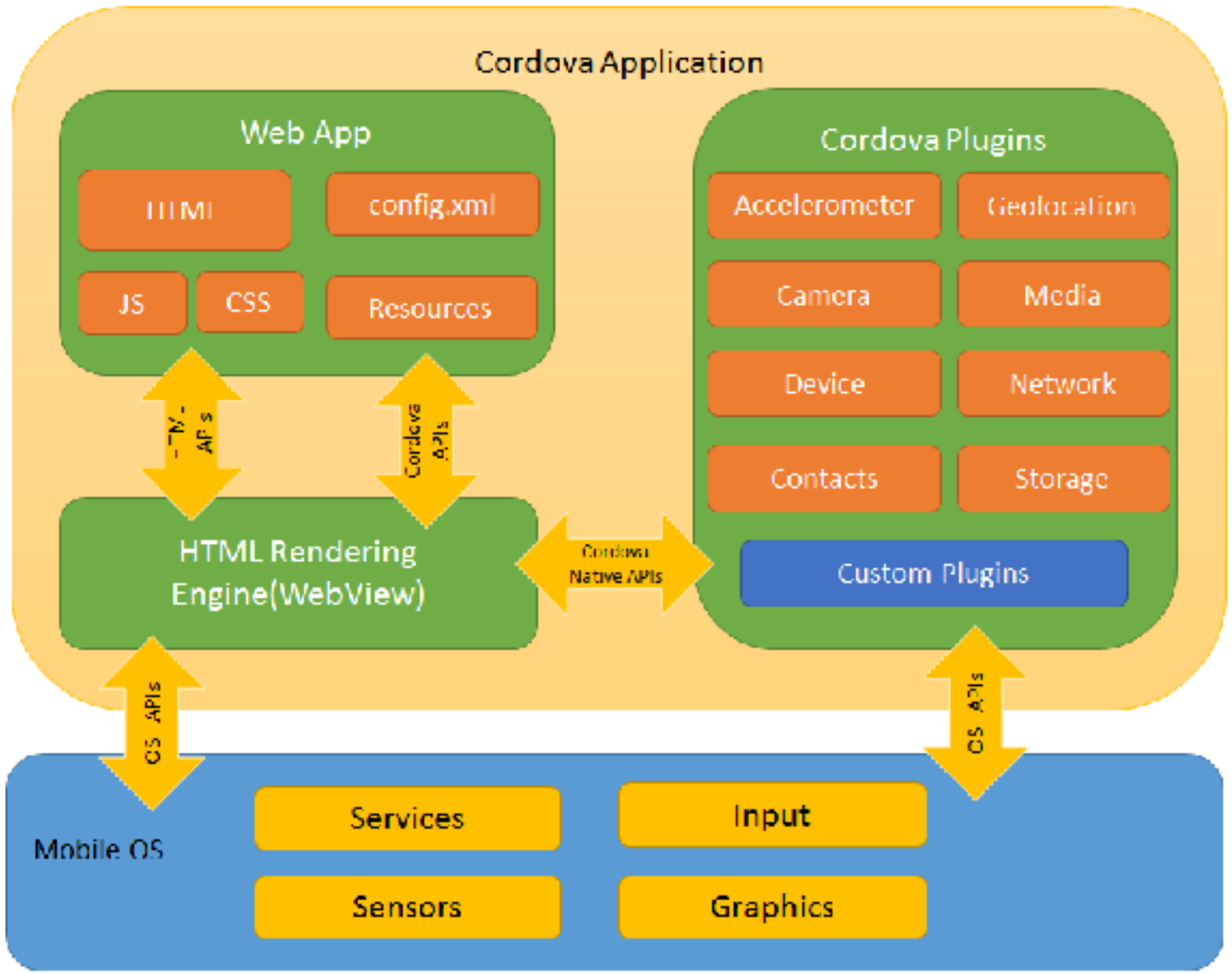
Two ways:

- Hybrid HTML5 web app that executes within wrapper in devices
    - E.g: *Apache Cordova*

- SDK that exposes the native APIs for multiple platforms, using a single programming language
    - E.g: *Xamarin with C#*

HTML5 - Cordova

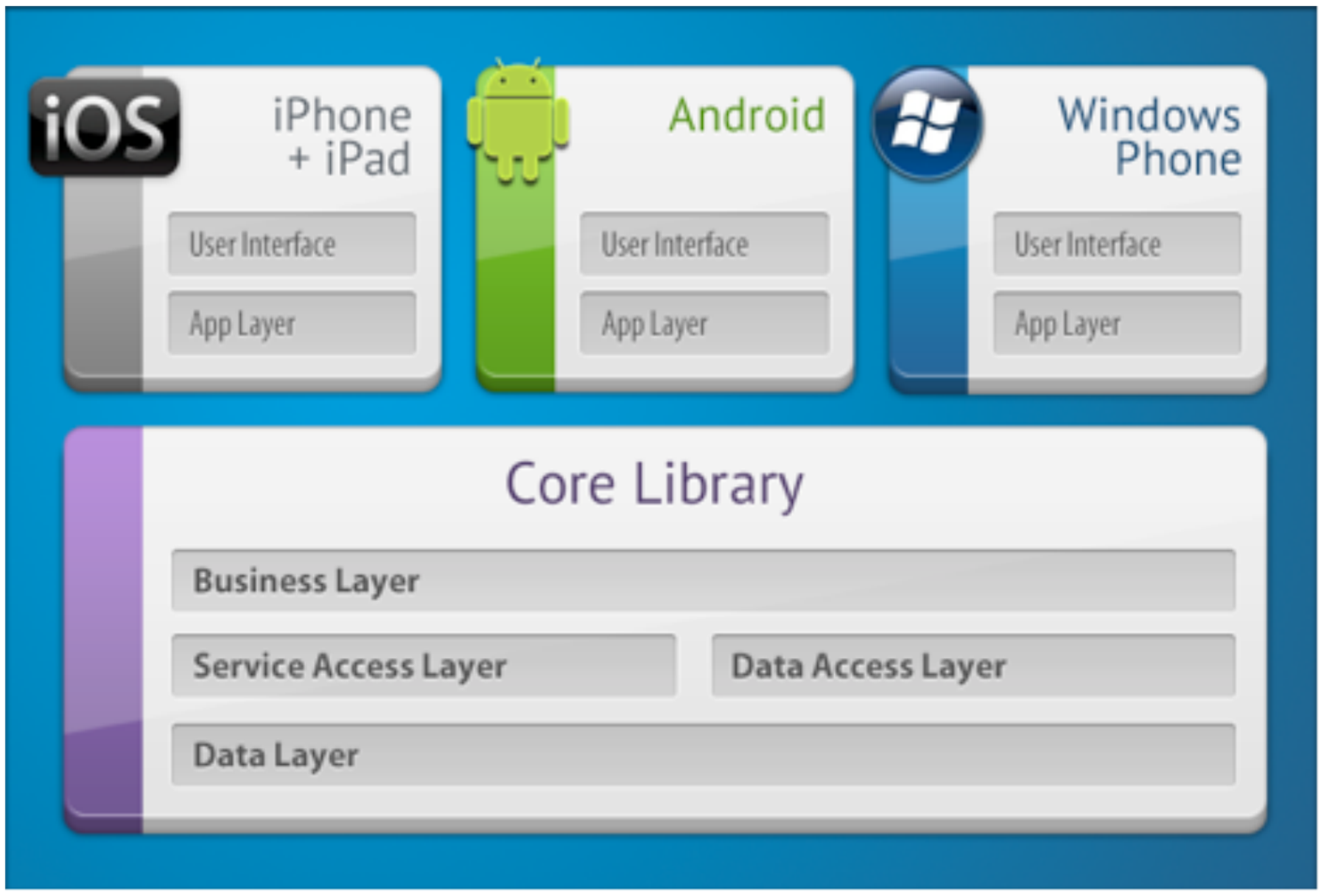
- Use standard web technologies - HTML5, CSS3, and JavaScript



https://cordova.apache.org/docs/en/latest/guide/overview/index.html

Xamarin SDK

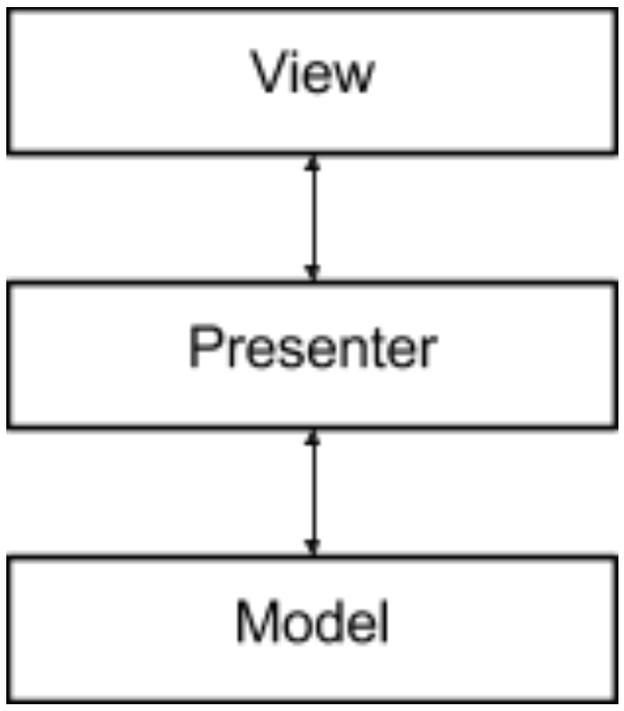
- Use C#



https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_0_-_overview/

Model-View-Presenter (MVP) Architecture

- Most common architecture for mobile apps
- MVP makes easier to test and maintain apps



Reacts to user actions
Displays data

Syncs the UI with data

Provides and stores the internal data

https://codelabs.developers.google.com/codelabs/android-testing/#3

Android Architecture Blueprints

- Architectural tools and patterns for Android apps:

  https://github.com/googlesamples/android-architecture

Mobile Sw Development Lifecycle

Inception

Monitor & Updates

Release

Design

Development

Testing

“Apps that receive negative user feedback in the first release, never become popular afterwards”

J. Mojica, M. Nagappan, B. Adams, T. Berger, S. Dienst, and A. E. Hassan. An examination of the current rating system used in mobile app stores. IEEE Software, 2015

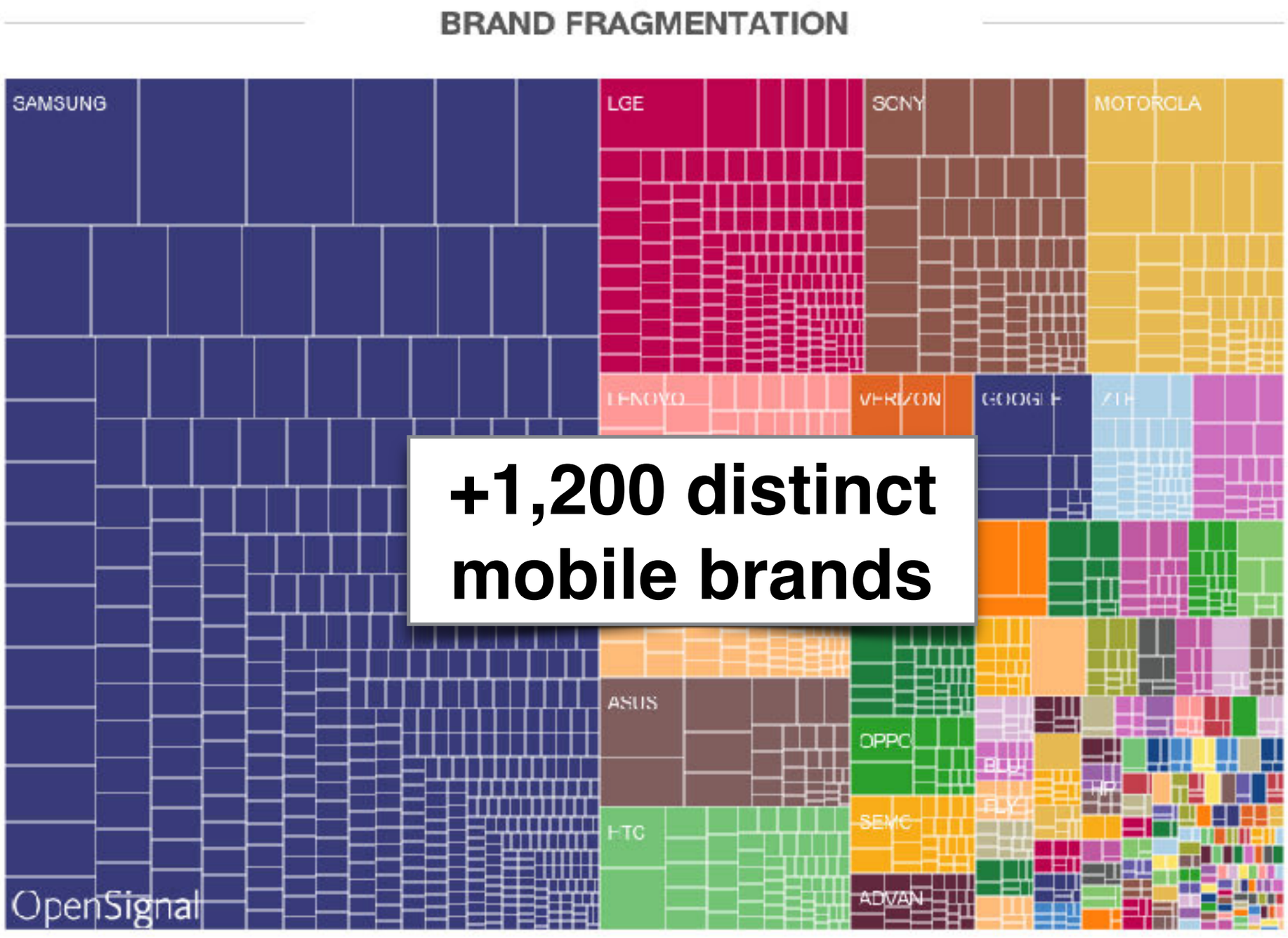
Testing

- Testing to verify correctness, functional behaviour and usability before releasing app publicly.

Mobile App Testing Challenges

- Device Fragmentation

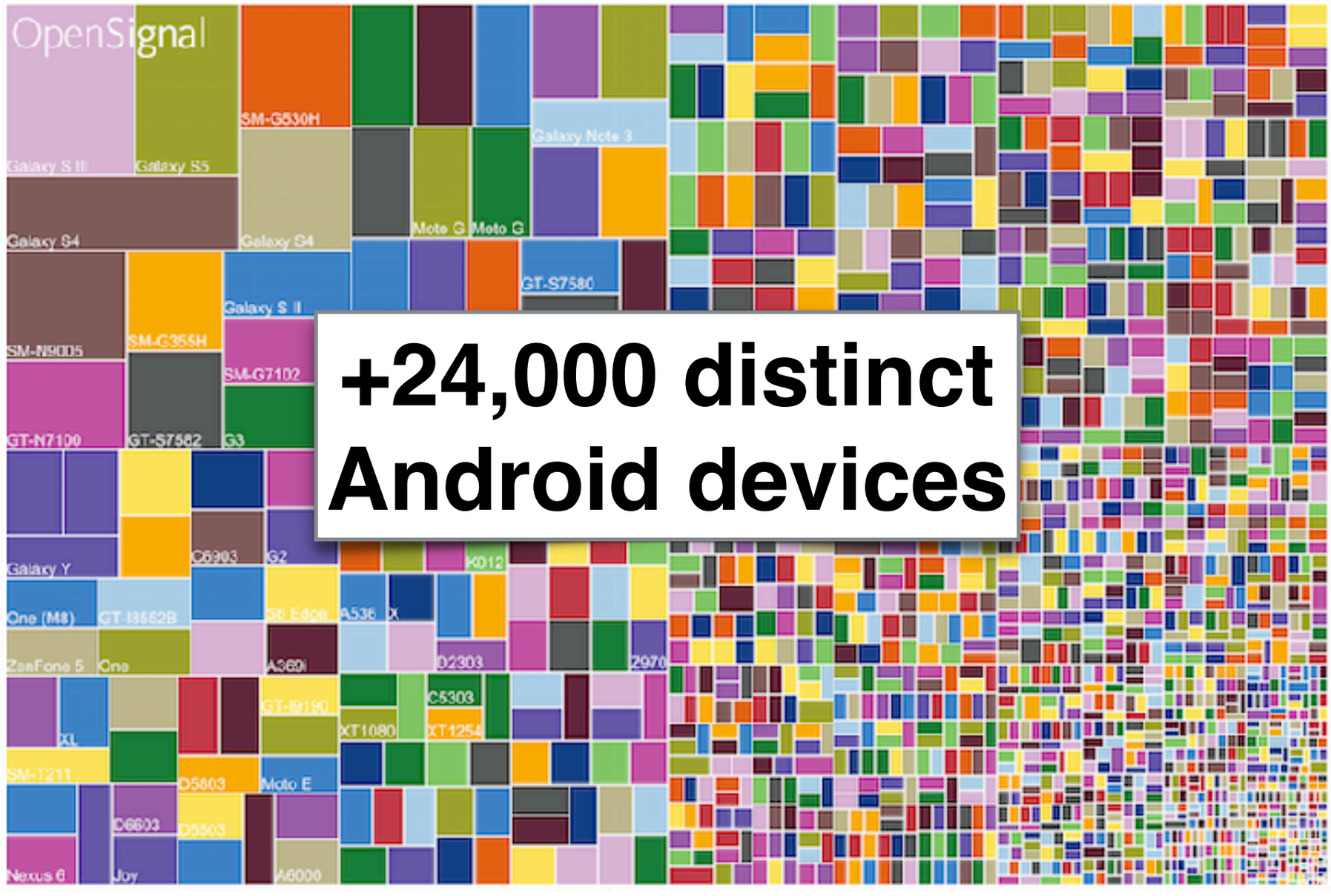
Mobile App Testing Challenges



https://opensignal.com/reports/2015/08/android-fragmentation/

Mobile App Testing Challenges

Device Fragmentation



https://opensignal.com/reports/2015/08/android-fragmentation/

Mobile App Testing Challenges

Android OS Fragmentation

Cupcake
Android 1.5

Donut
Android 1.6

Eclair
Android 2.0/2.1

Froyo
Android 2.2.x

Gingerbread
Android 2.3.x

Honeycomb
Android 3.x

Ice Cream Sandwich
Android 4.0.x

Jelly Bean
Android 4.1.x

KitKat
Android 4.4.x

Lollipop
Android 5.0

Marshmallow
android 6.0

Nougat
android 7.0

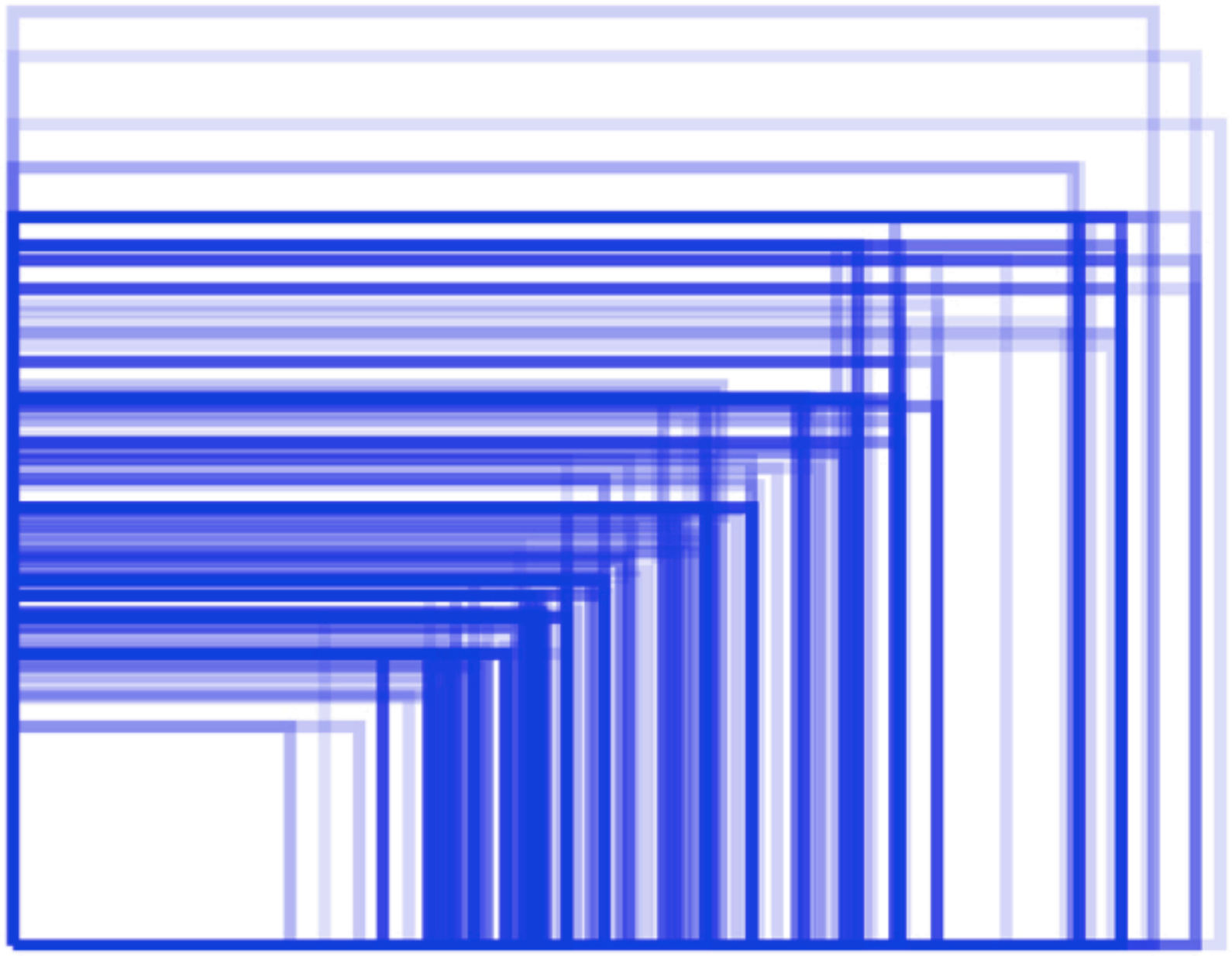
https://www.jsys.co/android-flavors-and-its-features/

Mobile App Testing Challenges

- Device Fragmentation
- **Screen size Fragmentation**

Mobile App Testing Challenges

Screen Size Fragmentation



https://opensignal.com/reports/2015/08/android-fragmentation/

Mobile App Testing Challenges

- Device Fragmentation
- Screen Size Fragmentation
- **Heterogeneous Contexts**
  - **Networks**
  - **Locations**
  - ***How to simulate real conditions in lab?***

Types of Testing

- **Black-box Testing**. Check the result. Don't look what happens inside a function.

- **White-box Testing**. Check which code is executed.

Types of Testing

- **Unit Testing**. Test individual functions (code).
- **Functional UI Testing**. Checks if the app behaves as expected when UI interactions happens.
- **Performance Testing**. Checks the performance of the app (memory, responsiveness, UI rendering, etc…)
- **Security Testing**. Checks security vulnerabilities and user privacy violations.
- **Regression Testing**. Compare with previous app versions.

What to test?

- **Key functionality**
- **Key use cases**
- **UI interactions**
- **Sensor data**
- **Phone interactions**
  - What happens if there is an input call? A message?

What to test?

- **Change in orientation**
  - Is the screen re-drawn correctly? Does the app maintain its state?
- **Change in configuration**
  - Eg., Changes in system language, keyboard availability, etc.
- **Battery life**
  - Write app to minimize battery usage
  - Test methods that manage battery usage
- **Dependence on external resources**
  - What happens when the network/Bluetooth/GPS are unavailable?

https://stuff.mit.edu/afs/sipb/project/android/docs/tools/testing/what_to_test.html

Monkey: UI/App Exerciser

- Program that generates pseudo-random user events (clicks, touches, gestures…) and system events
- Automatically explore apps
- Stress test applications

https://developer.android.com/studio/test/monkey.html

Test Automation Frameworks

- **Espresso** (Android)

  https://developer.android.com/topic/libraries/testing-support-library/index.html#Espresso

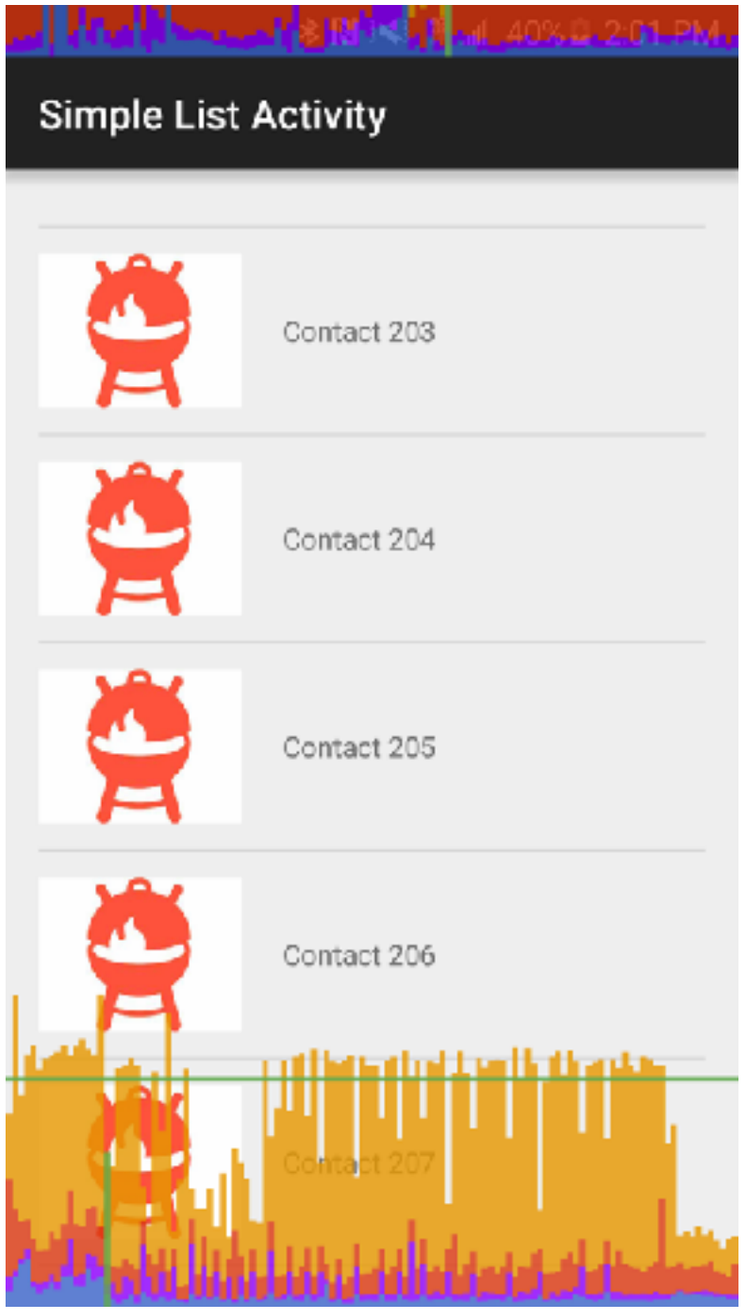
- **UIAutomator** (Android)
- **Robotium** (Android): https://github.com/RobotiumTech/robotium
- **Selendroid** (Android): http://selendroid.io
- **Calabash** (cross-platform): http://calaba.sh

UI Performance Testing

Test UI performance of apps

- Mobile apps should run a **60 fps** (frames per second) = **16 mspf**
- Frames taking more time are skipped! -> janky app perceived by users

UI Performance Testing



Frame rendering

Over the limit the app is seen janky!!

16mspf limit

https://codelabs.developers.google.com/codelabs/android-perf-testing/index.html?index=..%2F..%2Findex#0

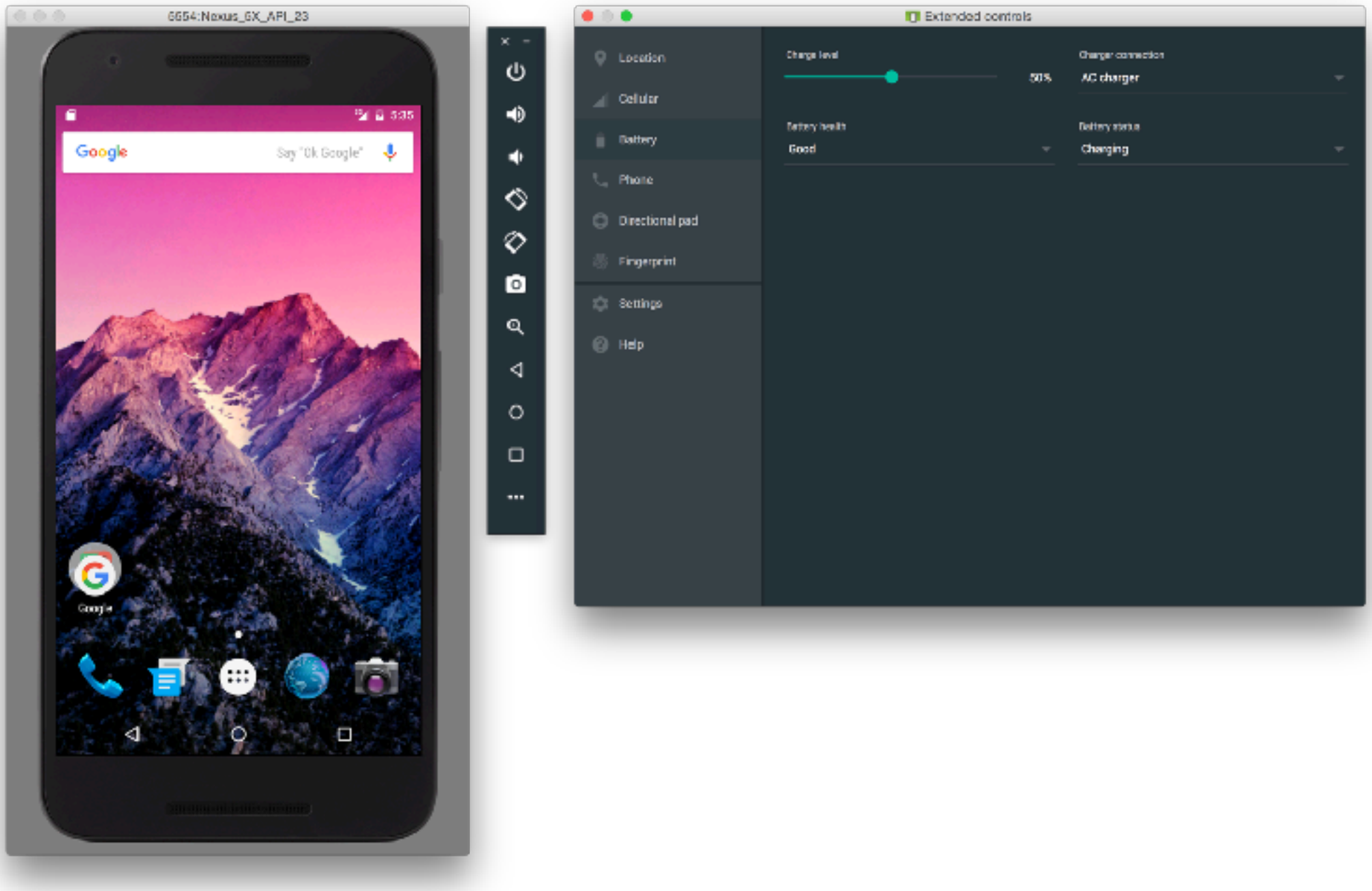
Real Devices & Emulators

- Testing can be done using:
  - **Real devices**
  - **Emulators**

    Emulators are useful but cannot substitute real devices!

Emulators

- **Android ADV**

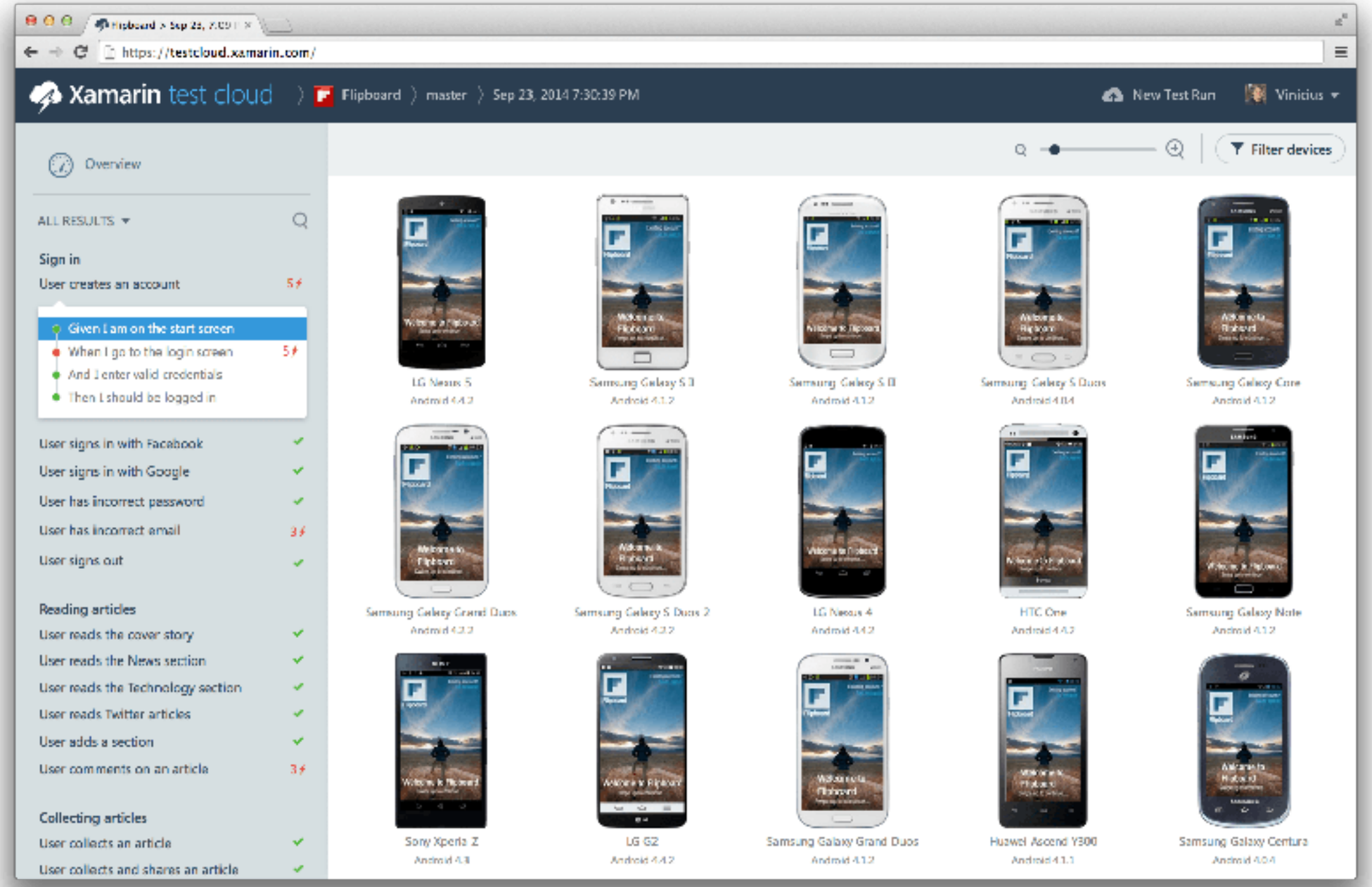


Testing Multiple Devices

- Due to fragmentation, testing on multiple devices is necessary
- Only common devices is not enough
- Cloud-based solutions

Testing Multiple Devices

- Xamarin Test Cloud: https://developer.xamarin.com/testcloud/

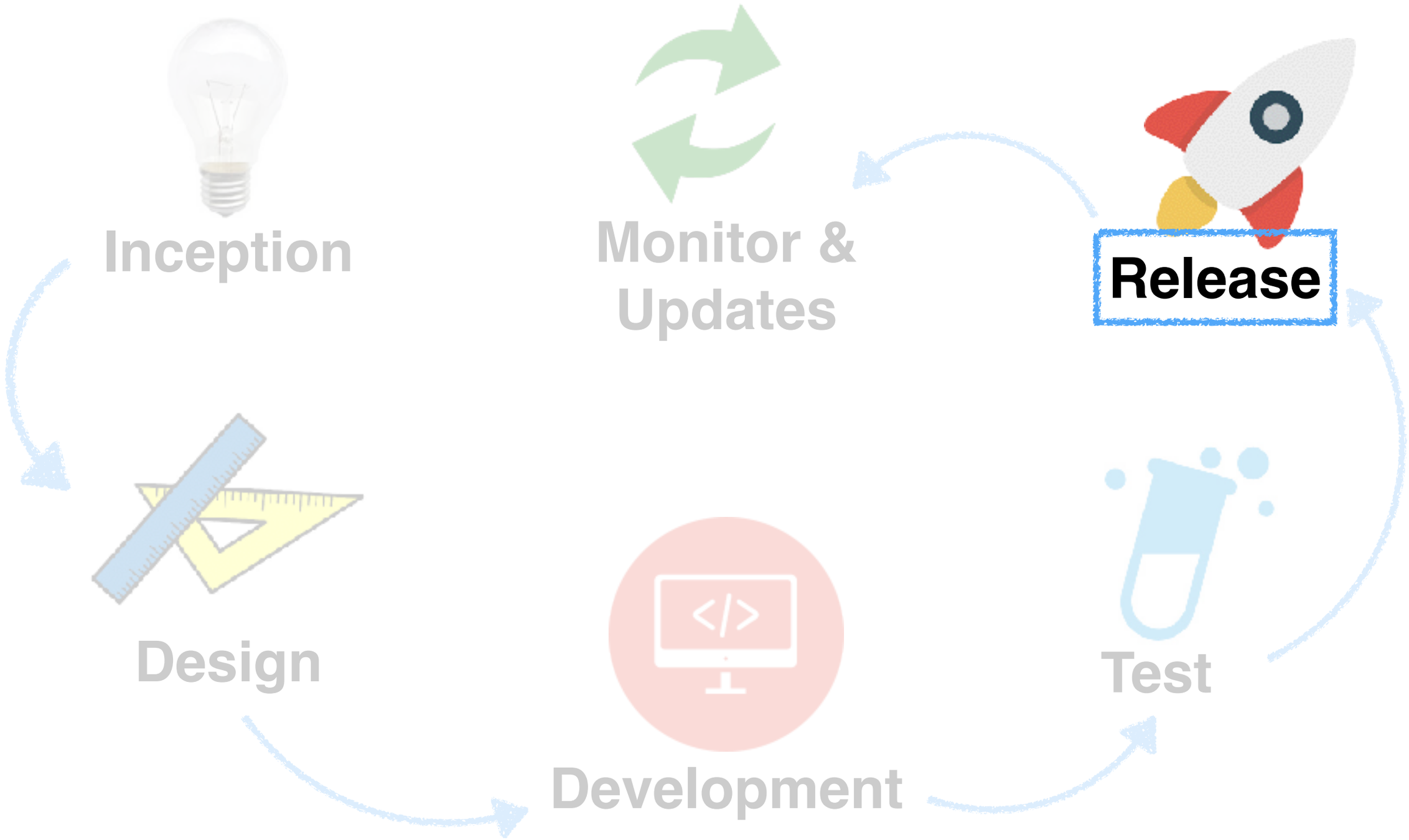


Testing Multiple Devices

- Firebase Test Lab for Android:
  https://firebase.google.com/docs/test-lab/

- Amazon Device Farm:
  https://aws.amazon.com/device-farm/

Mobile Sw Development Lifecycle



Distribution

Publish the app!

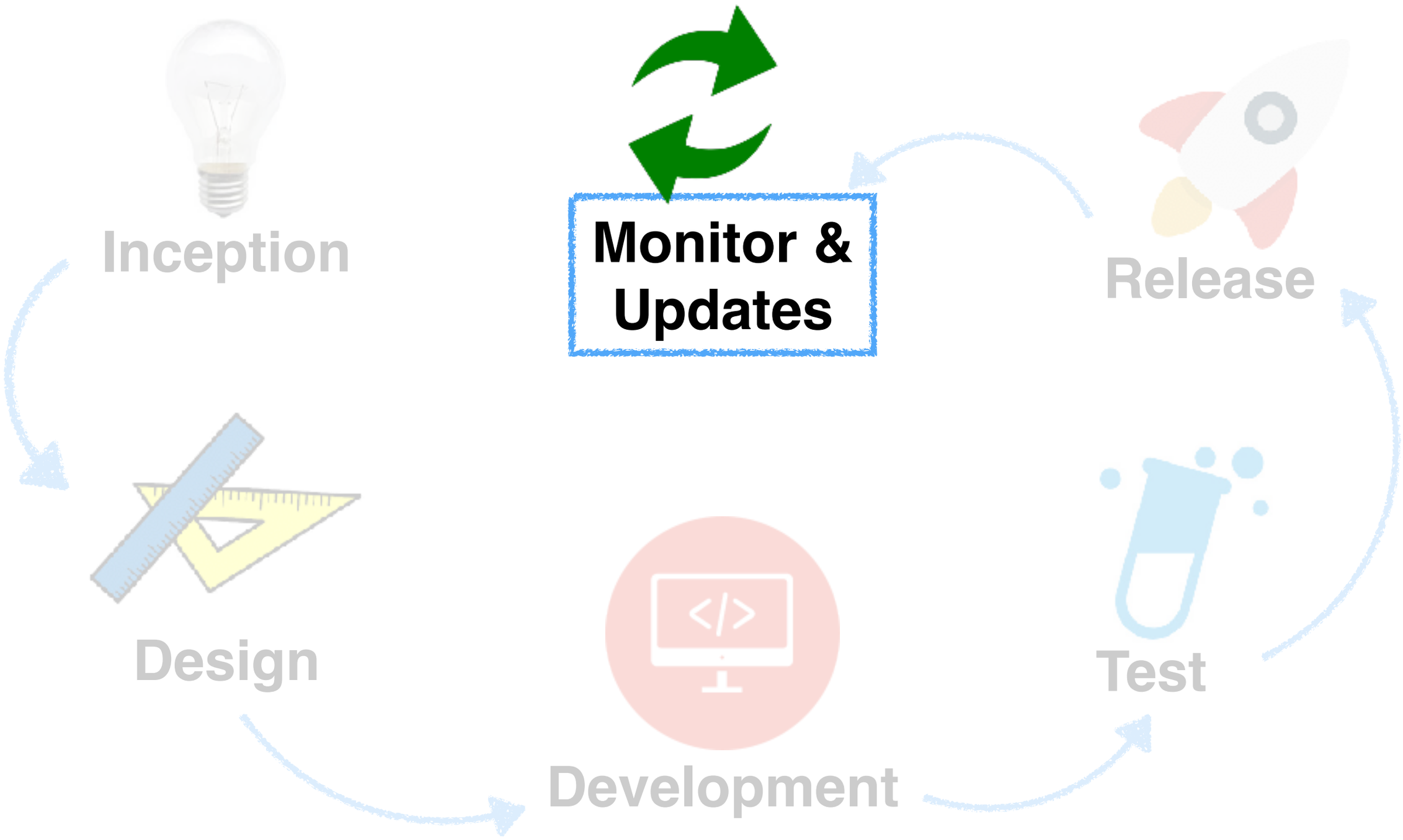
1. **Prepare the app for release**
   - What needs to get deployed with the app? Executable, images, database, libraries ?
   - Constraints (app runs on specific devices?)
   - Versioning

2. **Release the app to users**
   - Typically through App Marketplaces, *e.g., Google Play Store, Apple App Store, etc.*
   - Own distribution channels, *e.g., website*

Release Progressively

- Release progressively to ensure a positive reception
    - *Alpha- and Beta- Testing*
- Release early version of the app with a subset of users
- Fix technical or user experience issues before releasing the app widely
- Also release updates progressively

Mobile Sw Development Lifecycle



Monitor App Stats

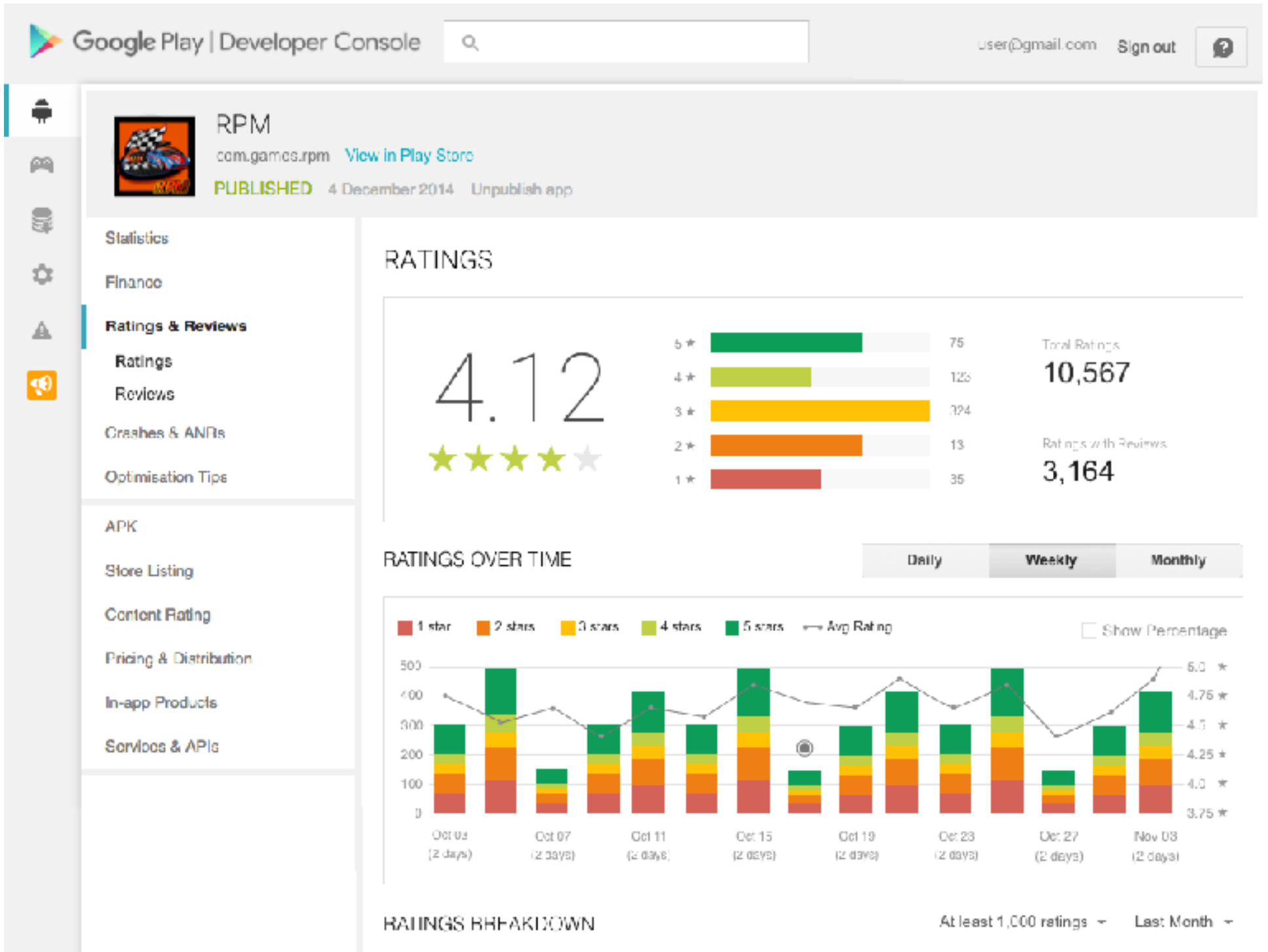
- **Gather and process data** about the app to identify how the app is performing

- Review information about the app: installs, ratings, crashes…

- Changes in the app's performance can indicate good and bad things

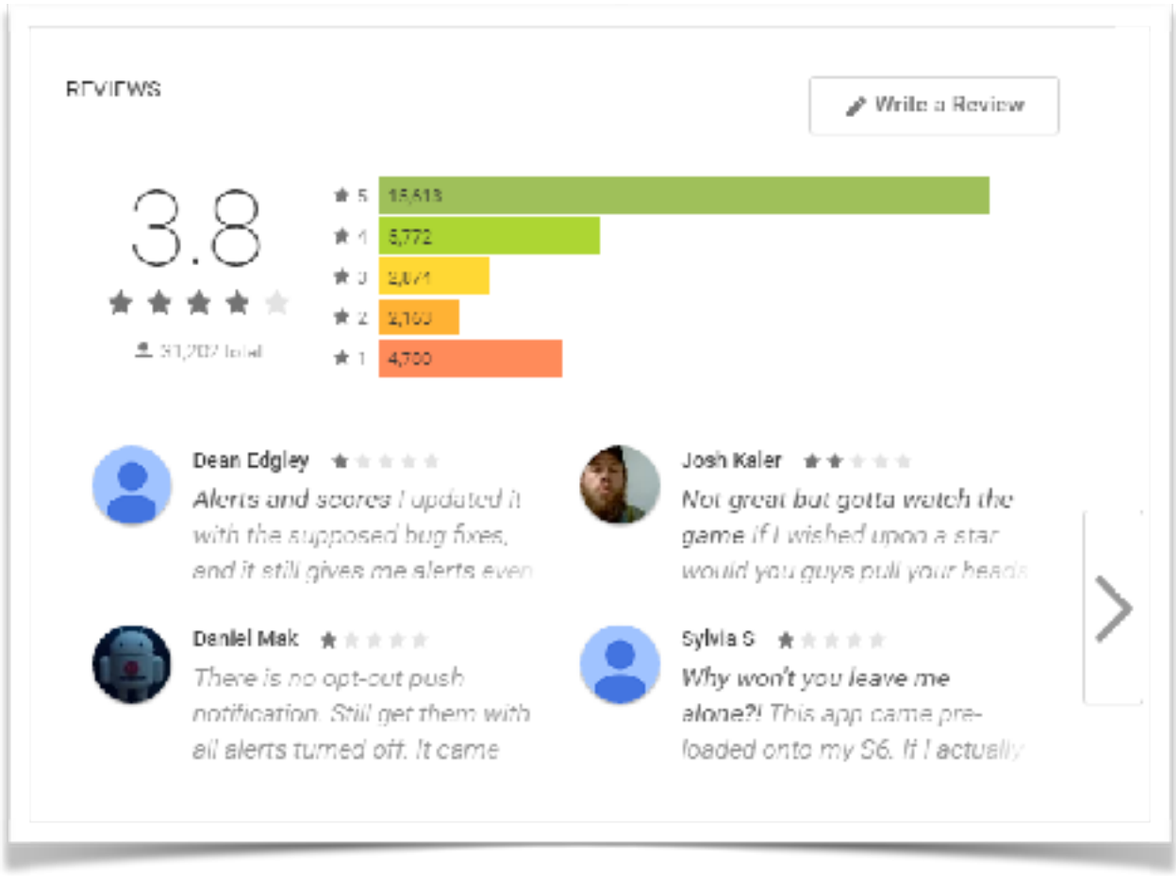
- **Quickly identify and correct** issues before they massively affect users' experience and harm app reputation

Google Play Developer Console



Monitor User Reviews

- **Keep an eye on users' reviews!**
- User reviews contain valuable feedback and suggestions for improving the app
- Read and reply to user reviews
  - Improve users' loyalty!

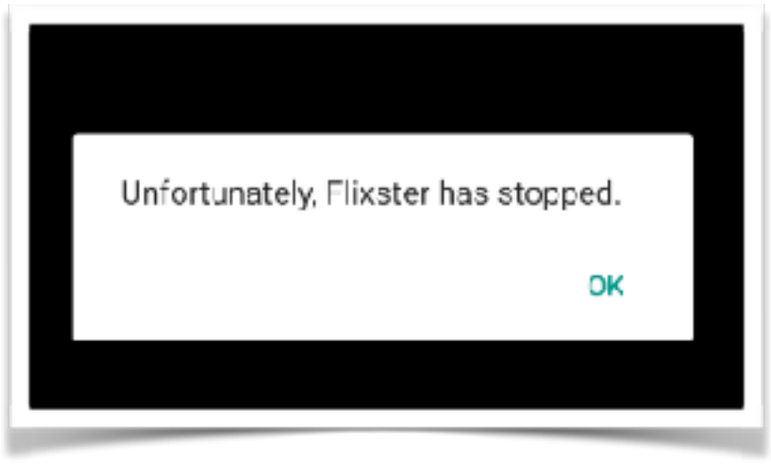


Crash Reports

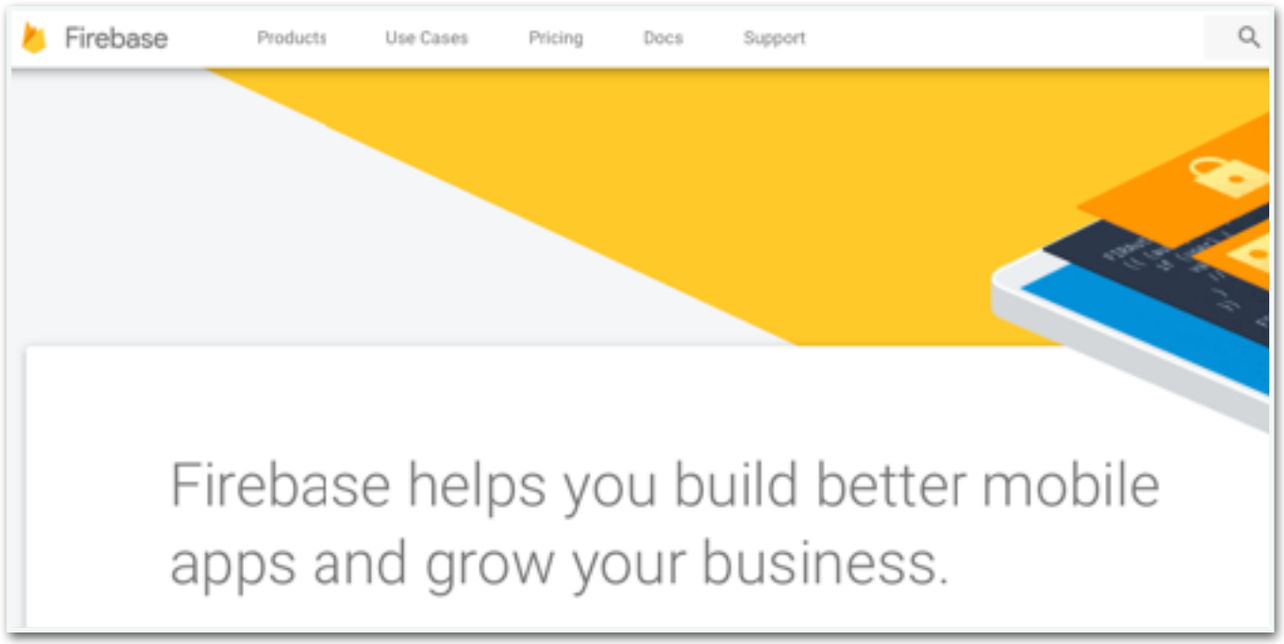
- App crashes and ANRs (*Application Not Responding*) heavily disrupts users experience
  - Lead to negative reviews and ratings

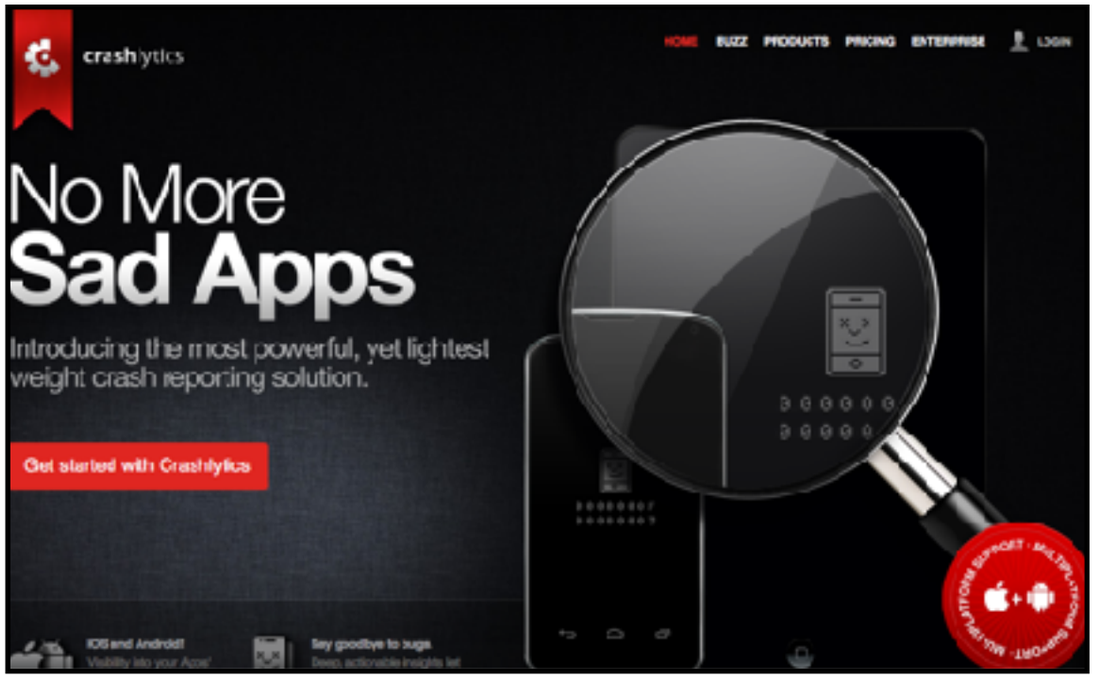
- Use crash reports to debug and improve your app

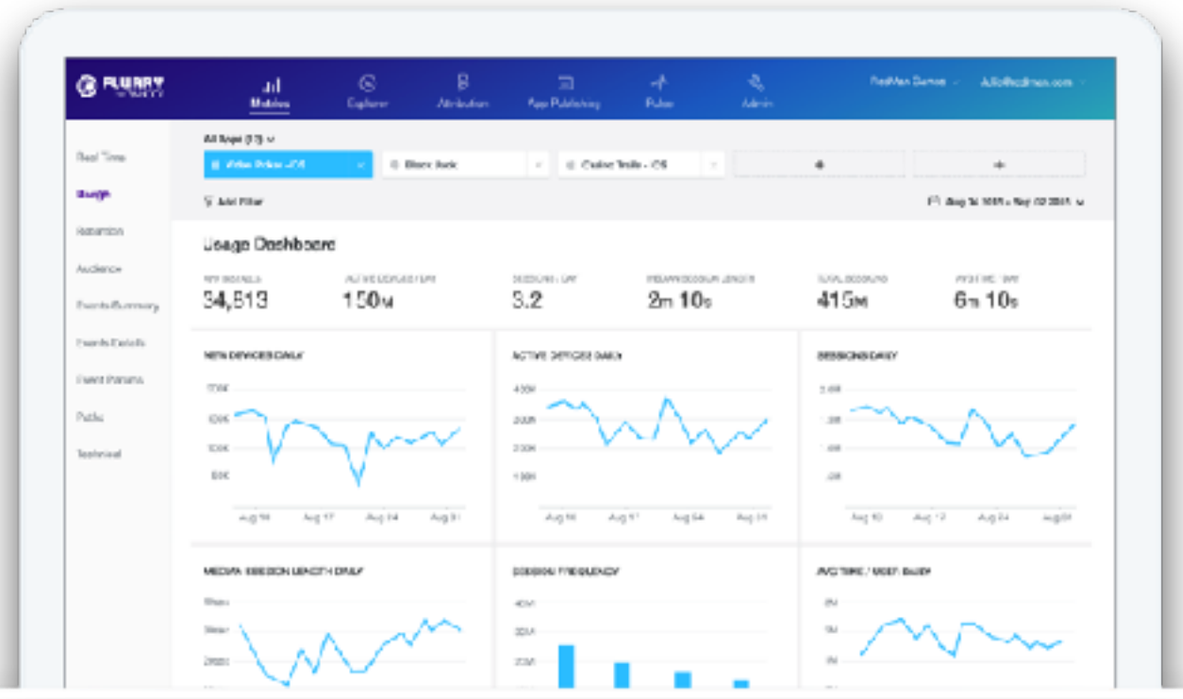
- **Correct any issues quickly!**
  - Avoid bad reputation
  - Reverse negative reviews



Analytics & Reporting Libraries





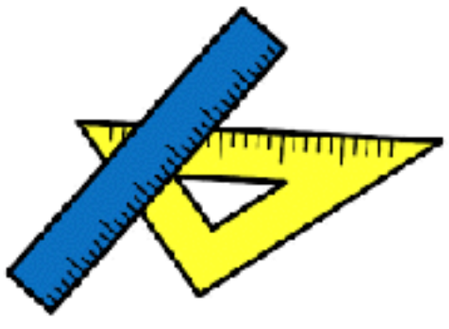


Mobile Sw Development Lifecycle

Inception

Monitor & Updates

Release



Design

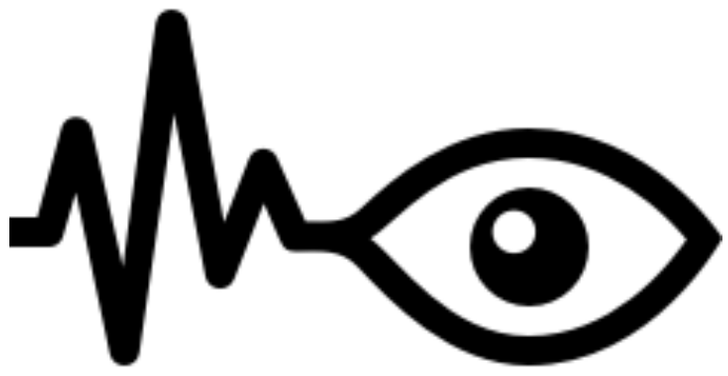
Development

Test

Security Considerations

Devices have access to many sensitive information!

- Personal data
    - User name, address, id…
    - Passwords
    - Banking data
    - Confidential documents
- Sensor data
    - GPS location. Track people!
    - Camera & Micro. Surveillance!



Security Considerations

- Privacy policies
- Security policies
- User agreements

References

Android developers:
https://developer.android.com/develop/index.html

Udacity. “UX Design for developers” by Google:
https://www.udacity.com/course/ux-design-for-mobile-developers--ud849

Xamarin. Introduction to the Mobile Software Development Lifecycle
https://developer.xamarin.com/guides/cross-platform/getting_started/introduction_to_mobile_sdlc/

Google developers Codelabs:
https://codelabs.developers.google.com