Introduction to Mobile Programming 2016 Fall

신민호

Information

- Teacher
  - Minho Shin
  - 5742 (mhshin@mju.ac.kr)
- TA
  - jinsung Kim
  - 5745 (nadas9029@gmail.com)
  - 010-4805-5377
- Class
  - Time: Mon 11-12, Wed 2-4
  - Place: 5420
- Grades
  - Att (10%), HW (50%), Mid (20%), Fin (20%)

Goal

- Understand Mobile Development Process
- Understand Mobile Design (Material Design)
- Understand Ionic-2 Hybrid App Framework
- Understand Android Native App Development

How the class works

- This class does not teach everything
  - Only directions & resources are provided
  - You have to make your way
- This class is not step-by-step walk-thru class
  - Difficult to set-up lab environment
  - Time consuming
- I only explain theory/methods/resources
  - You do the job in homework
- All class material in English

Drop the class IF

- I cannot read English
  - Think about changing your major
- I am not interested in Mobile Development
  - Need to be self-motivated
- I cannot do things by myself
  - Need to be self-learner
- I am lazy but need a good grade
  - Take other course

Human-centric Mobile Computing Lab

- Topics
  - Mobile Security & Privacy
  - Mobile Sensing
  - Internet of Things
  - Smart car technology (EV, V2V)
- Looking for Grad students, who are
  - interested in mobile technology
  - interested in security and privacy
  - interested in power IT (smart grid, EV)

Challenges in Mobile App Dev.

- Form-factors
- Input methods
- Interaction model
- Implementation methods
- Deployment
- Testing

Form Factors

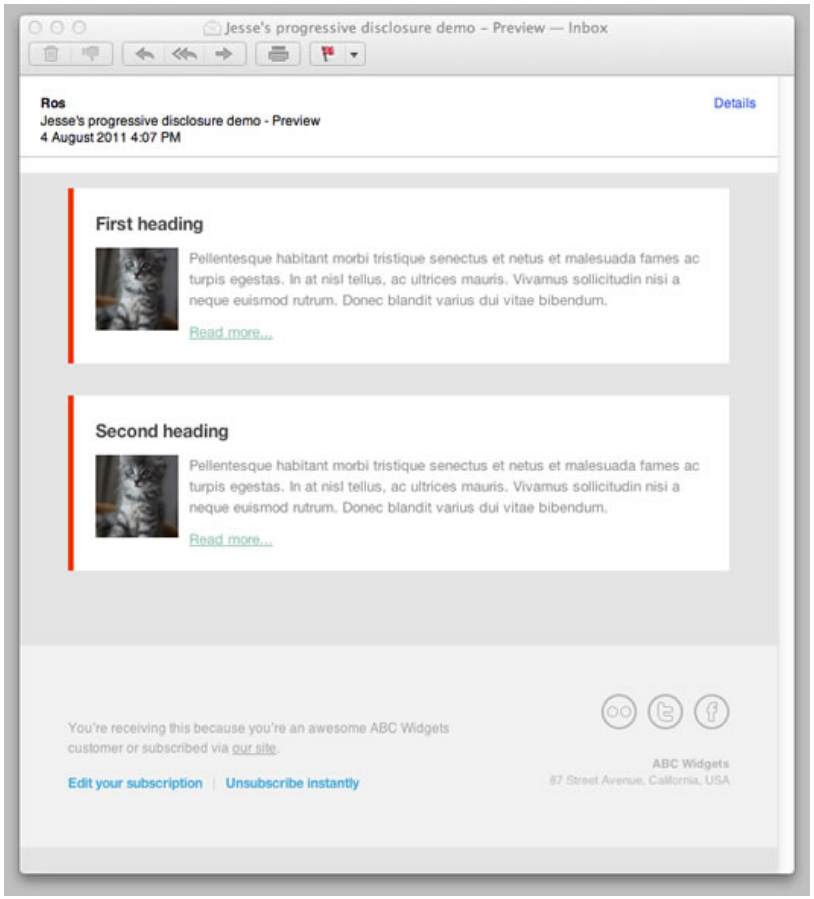
= Physical size & shape of a computer HW

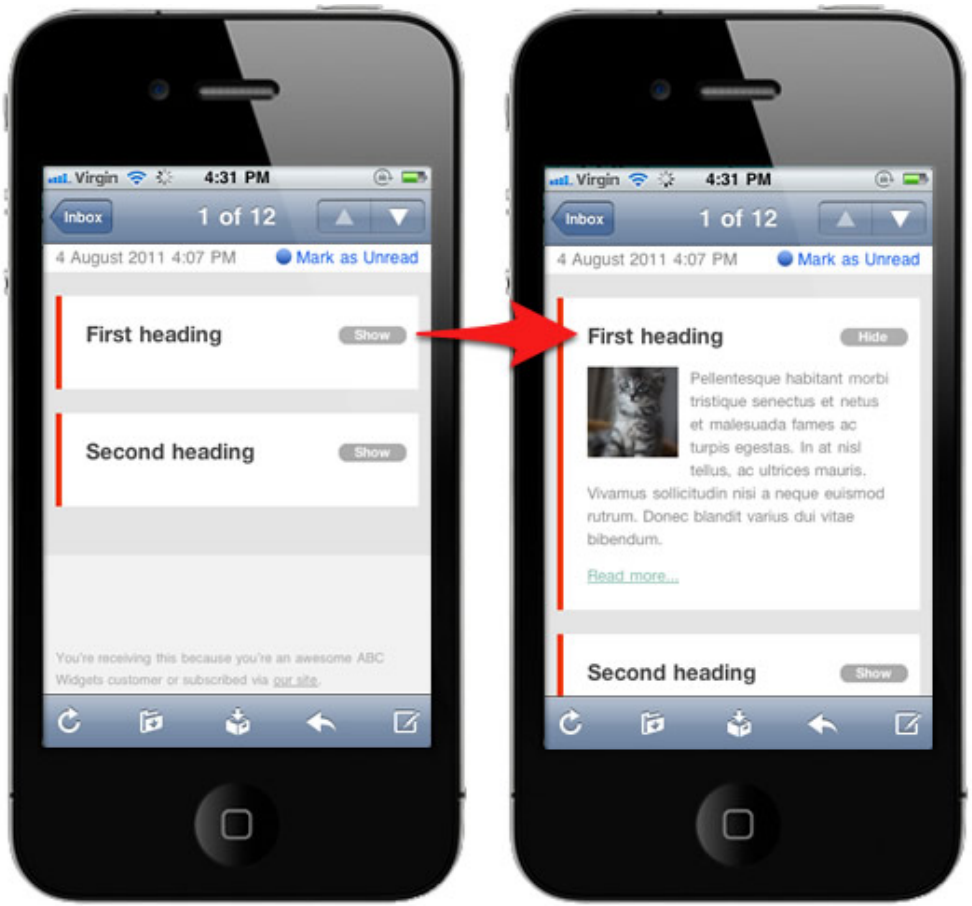
Mobile Form factors

- Diversity
  - Before: Standardized
  - Mobile: Difference screen sizes, ratios, orientations
    - *Responsive Design*
- Screen real estate
  - Before: Spacious
    - Show everything, let user find information
  - Mobile: Cramped
    - *Progressive disclosure*

Responsive Design

Progressive Disclosure





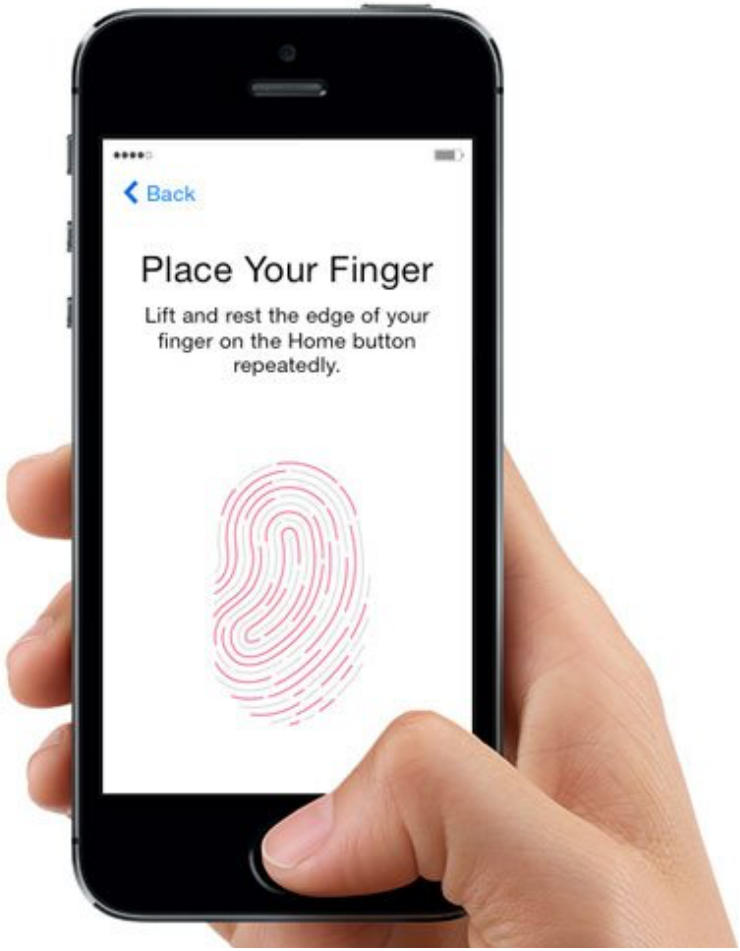
Operating Systems

- Before: biased
  - End-user: Windows
  - System: Unix, Linux
  - Special: Mac OS
- Mobile: mixed
  - Android, iOS, Window Mobile, BlackBerry, …
  - Different UI, SW stack, API, capabilities, ...
  - Portabiltiy is a PLUS

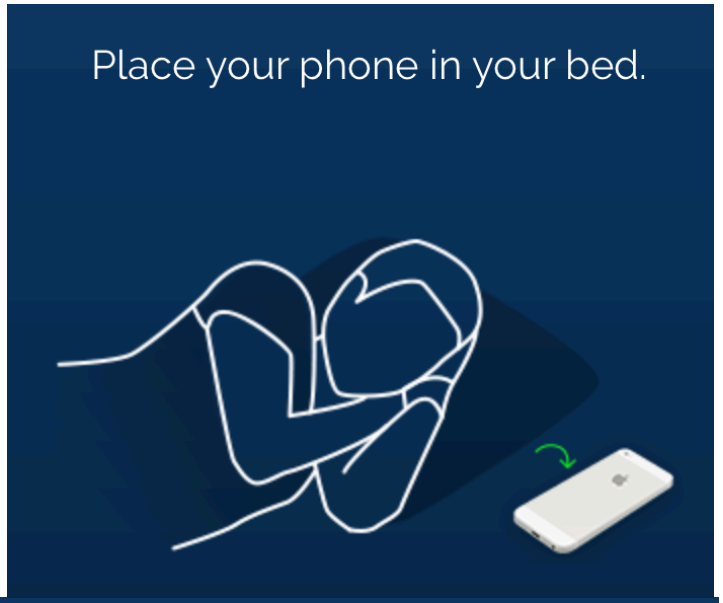
Input Methods

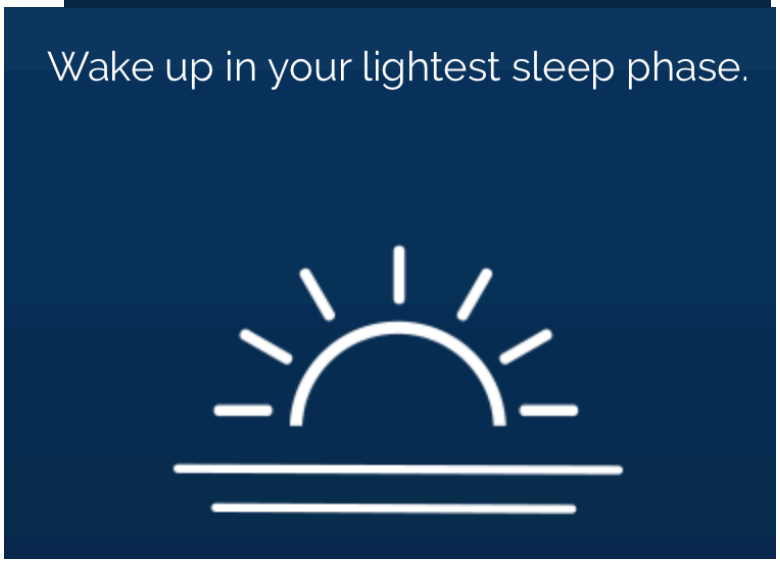
- Before
  - Keyboard
- Mobile
  - Confined keyboard
  - Camera, Mic, Sensors (Accel., Gyro., Prox., GPS...)
  - *Limited but new opportunities*

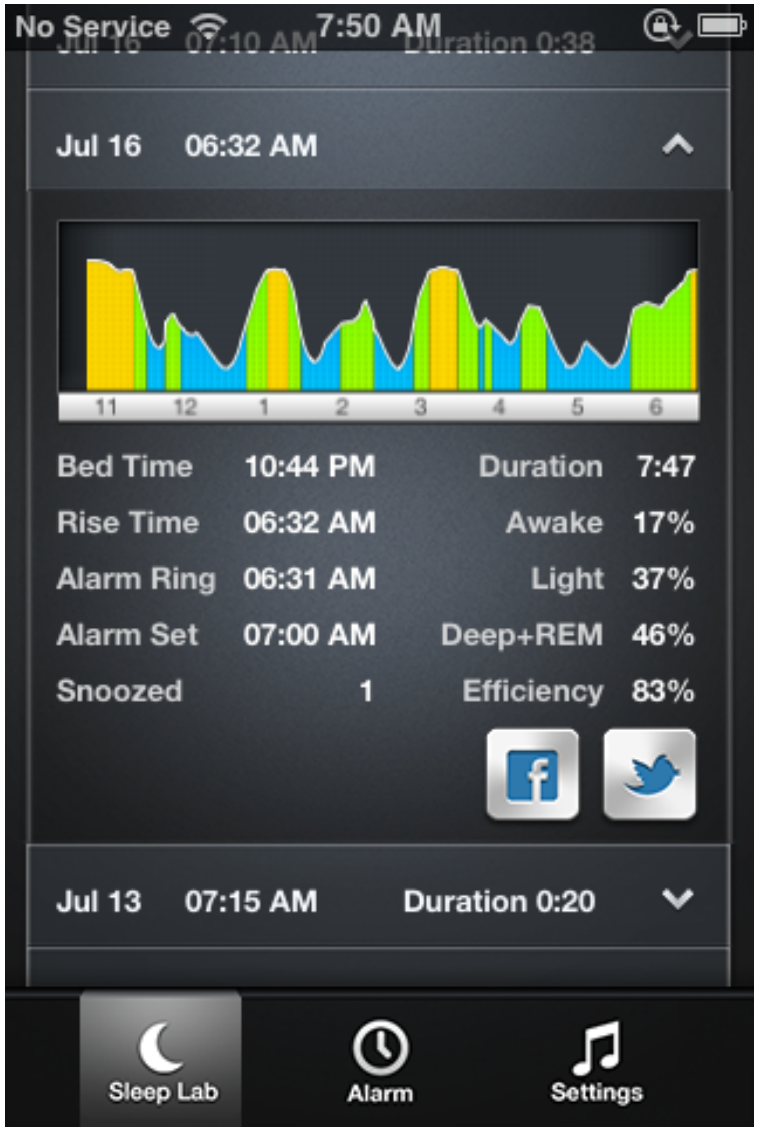
Sensors in Smartphone



SleepTime App







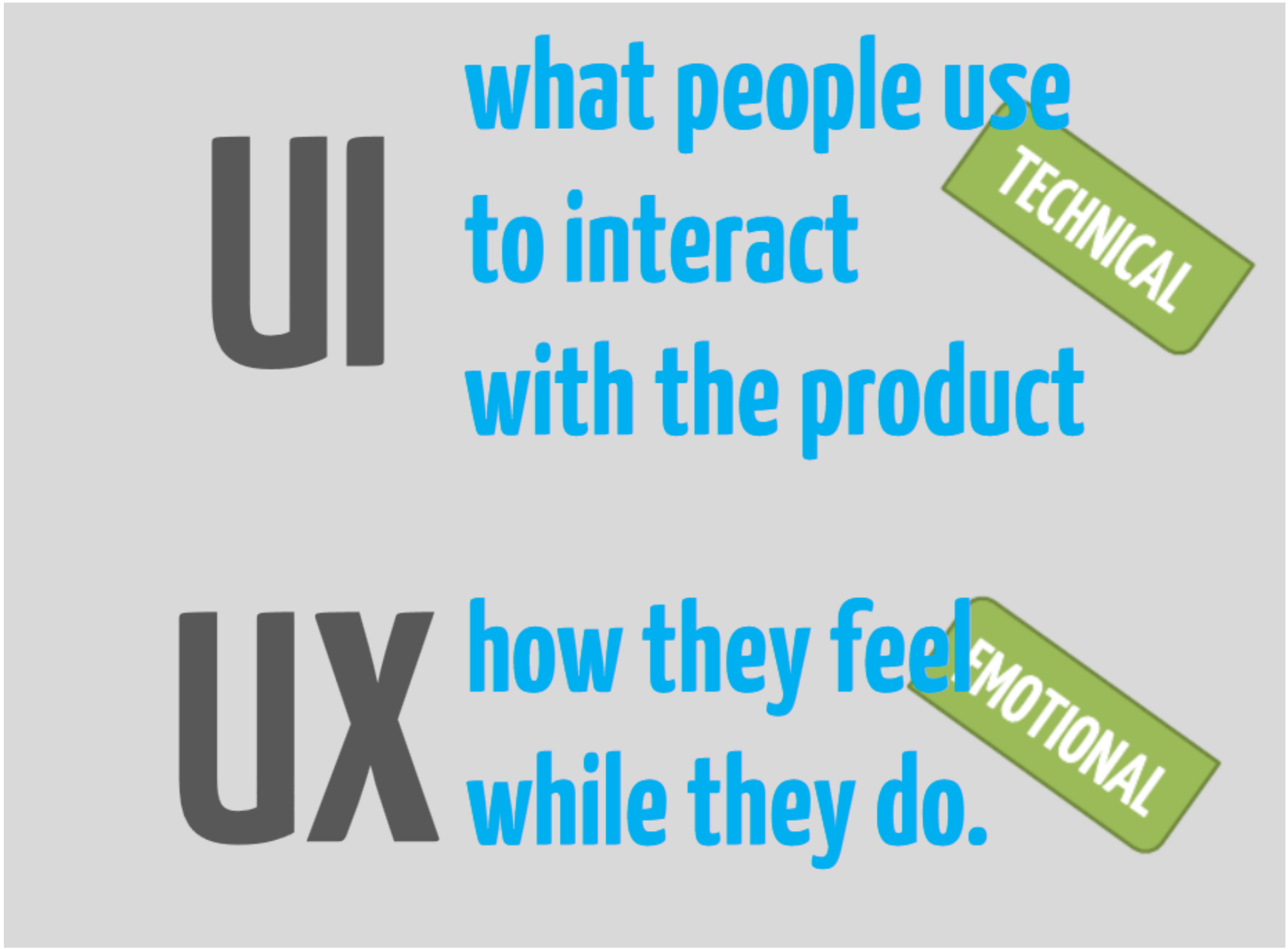

Chaini
Chaini
Chaini

Interaction Model

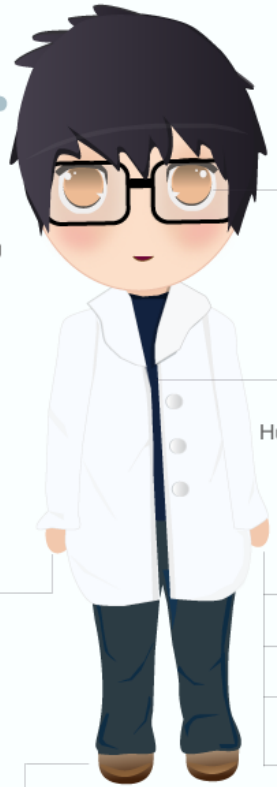
- Before
  - Long
  - Continuous
  - Steady
- Mobile
  - Short
  - Interrupted
  - Bursty

UX & UI Design is critical

- Before: UX & UI Design in later stages
- Mobile: UX & UI Design in Early stages


UI
what people use
to interact
with the product
TECHNICAL
UX
how they feel
while they do.
EMOTIONAL

UX Designer VS UI Designer



Thought process

Critical Thinking
CreativeThinking

Visual Strenghts

Task Flows
Scenarios
Colors
Typography

Thought process

Creative Thinking
Convergent Thinking

Design Principles

Human Centered Design
Visual Design

Design Based on

Expertise

Wireframes
Prototypes
Research
Mockups
Graphics
Layouts
Patterns

Design Based on

Sometimes Fills

Information Architect
Program Manager
Content Strategist
Functional Analyst

Sometimes Fills

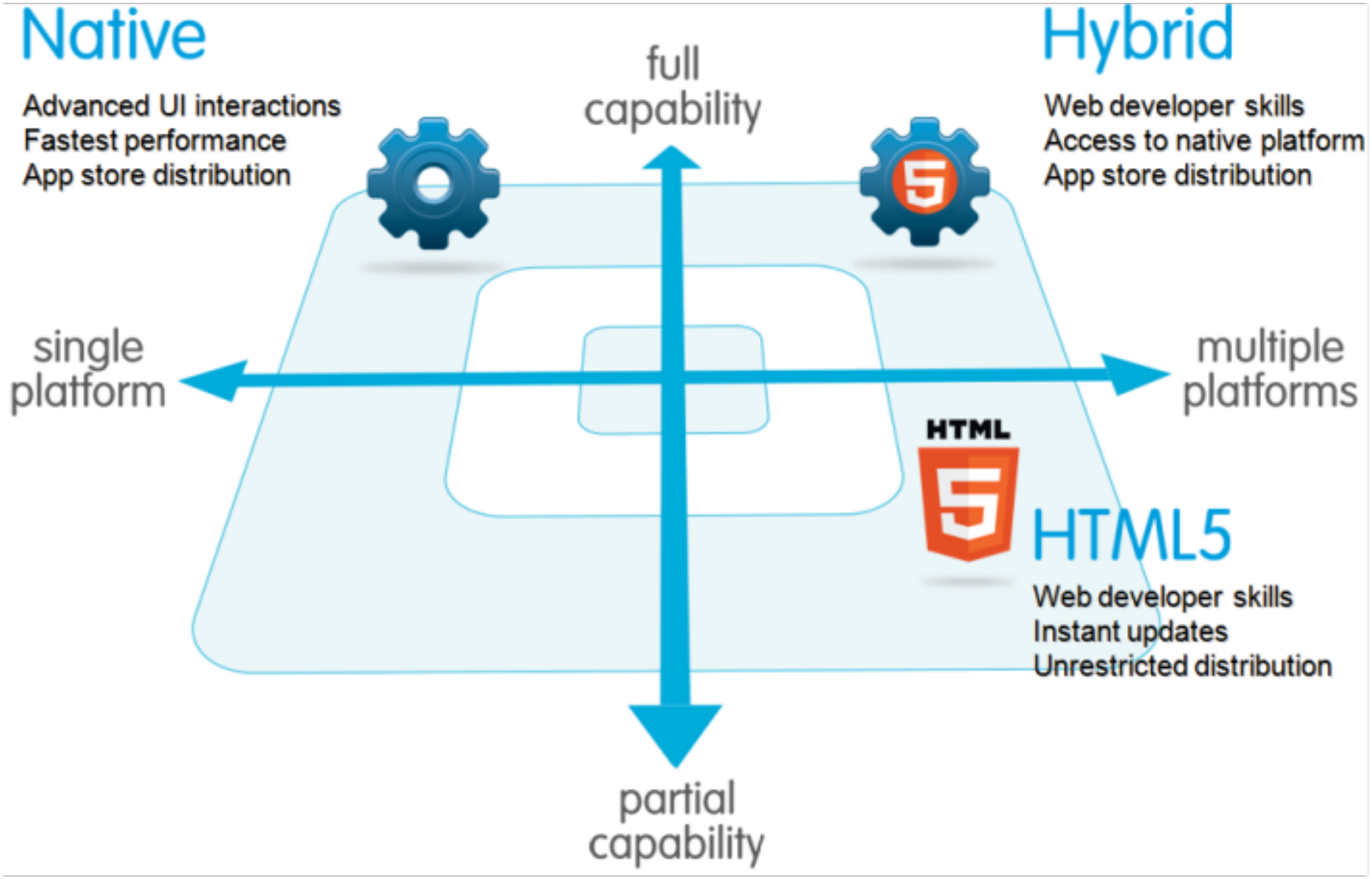
Graphic Designer
Brand Designer
Web Designer
Frontend Developer

Median Salary

Infographic created by Ana Harris
UXdesigner21.com

Implementation Methods

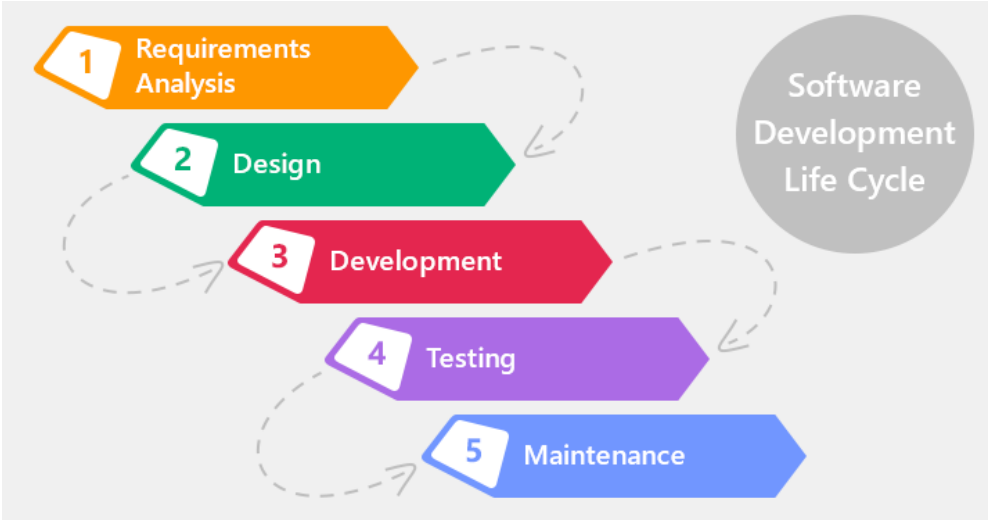
- A variety of implementation choices
- Not one choice for all situations
- Choose carefully in an early stage
- Choices
  - Native App
    - Good: Fidelty, Full feature of device (HW, Services)
    - Bad: non-portable
  - Web App
    - Good: Easy, portable, easy deploy
    - Bad: no access to native features (Camera, Contacts, …)
  - Hybrid
    - Web App inside, but Native App outside, with native plugins
    - Good: Easy, portable, some device features
    - Bad: ?


Native
Advanced UI interactions
Fastest performance
App store distribution
full capability
Hybrid
Web developer skills
Access to native platform
App store distribution
single platform
multiple platforms
HTML
HTML5
Web developer skills
Instant updates
Unrestricted distribution
partial capability

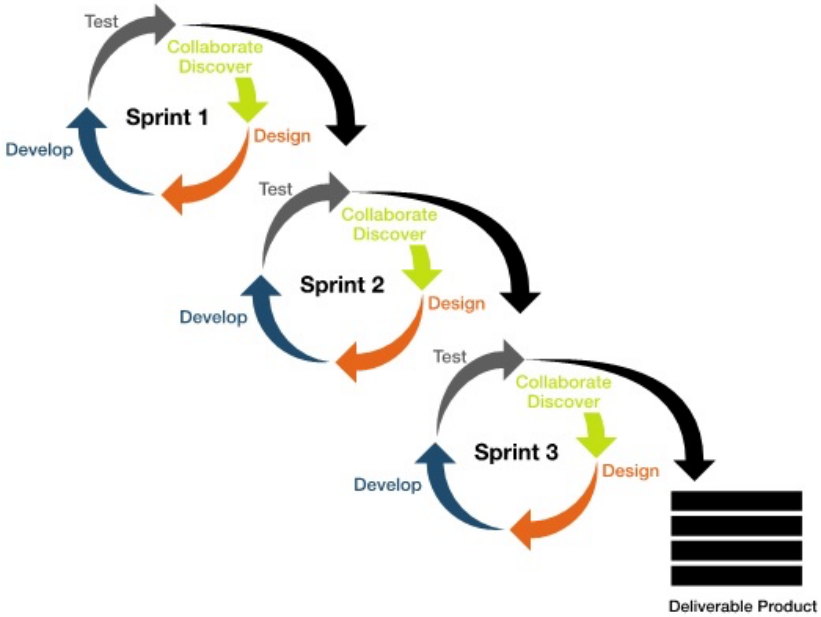
Deployment

- Before
  - Build for 2-3 years and do marketing, sell on web
- Mobile: aggressive
  - Wonderful eco-system
  - Inception-to-delivery: 2-3 months
  - Upgrade-to-delivery: Daily, Hourly
  - Agile method is popular

Waterfall Dev. Life Cycle



Agile Dev. Life Cycle



Waterfall method

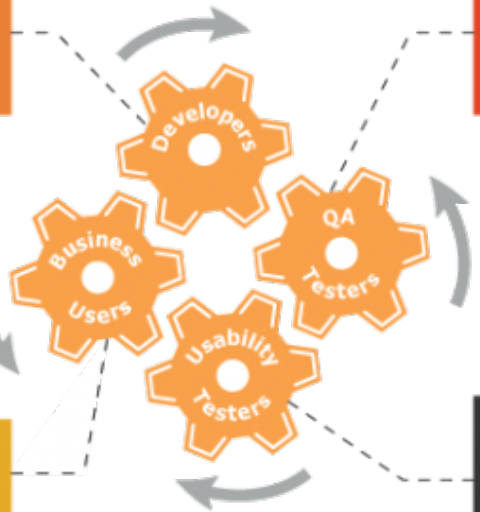
- App team works sequentially rather than collaboratively.
- Long, slow development cycles that delay time to market.
- Inflexible and slow to make changes and updates.

VS

Agile method

Developers can build apps quickly without coding and easily make changes and updates based on feedback from team

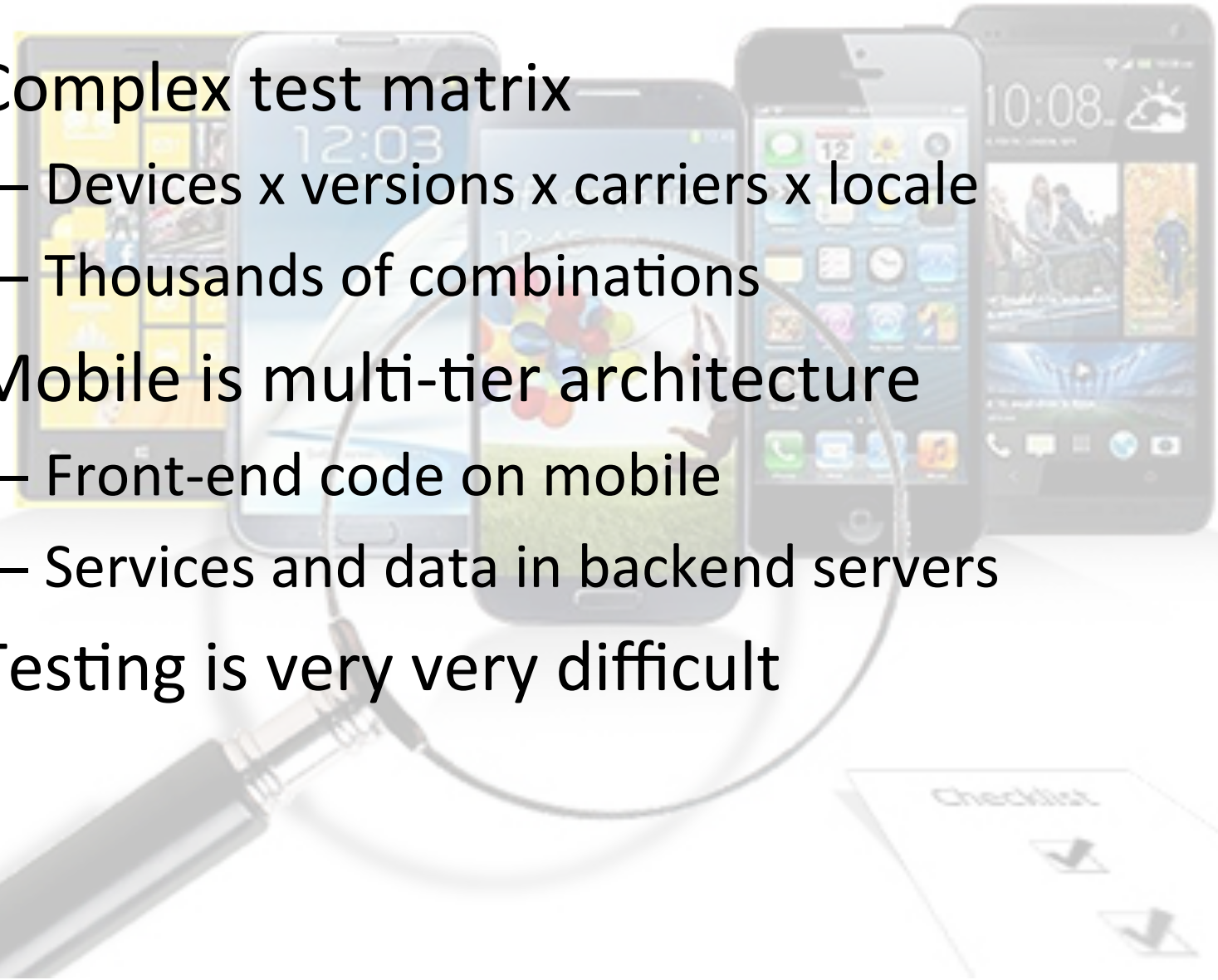
QA testing starts immediately using Preview App and in parallel with app changes made by developers with feedback reported instantly



Business users can easily review updates and instantly provide feedback

Usability testing starts as soon as the first screen is complete. Long usability cycles dramatically improve app usability and user satisfaction

Testing



- Complex test matrix
  - Devices x versions x carriers x locale
  - Thousands of combinations
- Mobile is multi-tier architecture
  - Front-end code on mobile
  - Services and data in backend servers
- Testing is very very difficult

Homework #1

- Research on Hybrid App Frameworks
- Answer
  - What is hybrid app?
  - What is the benefit of hybrid app?
  - What kind of methods exist?
  - Compare them
  - What are pros/cons of each method?
- by 9/12, email