

# AFFINITY ANALYSIS ANDROID/IPHONE BASICS

21W.789 CLASS 2

# Data Analysis

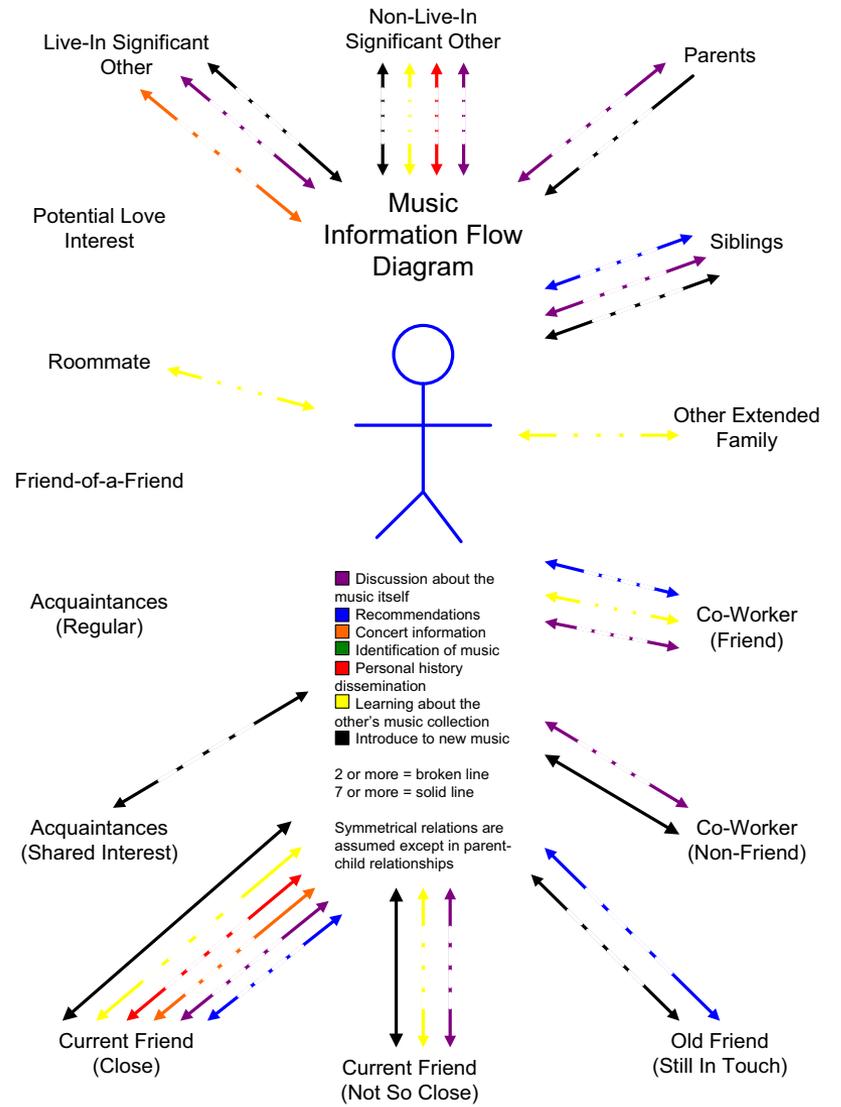
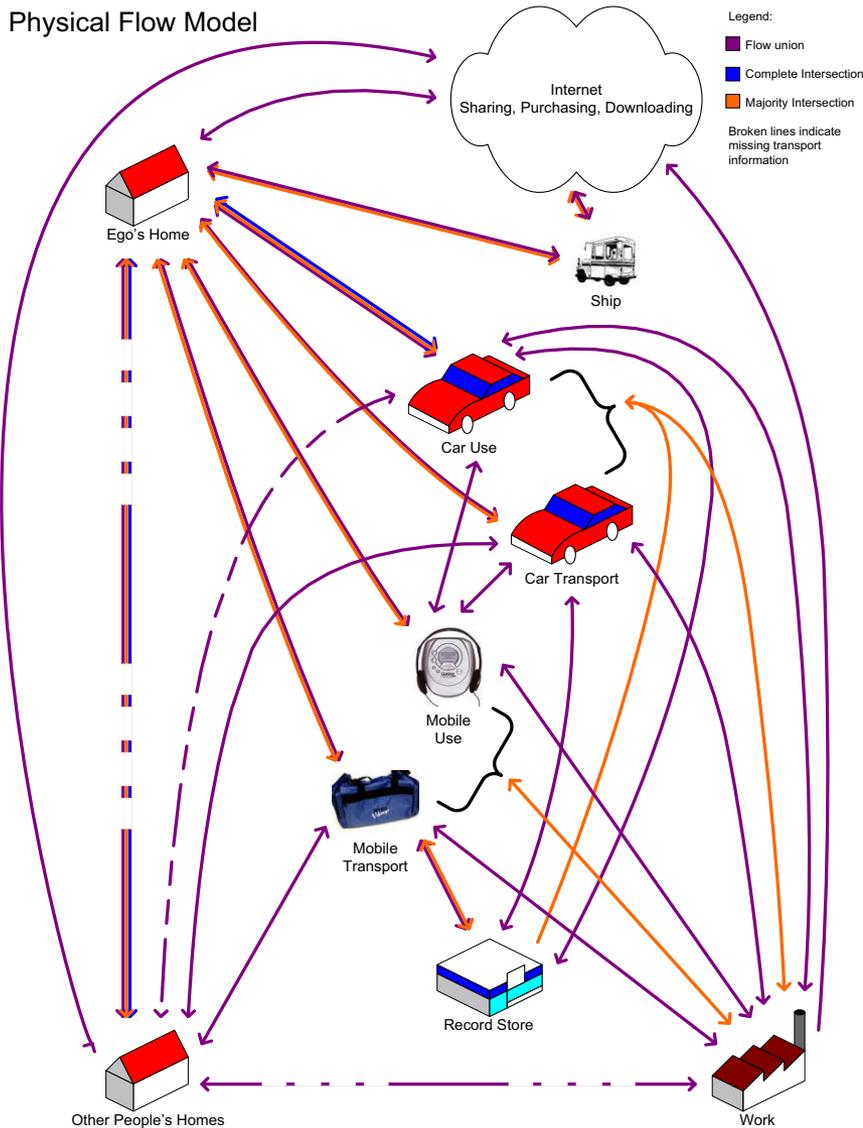
- Qualitative methods generate a LOT of data
- Quantitative analysis can sometimes be performed on logging or ESM data
- Most data very descriptive in nature
- Analysis used to build models of use and inspire new ideas

# Flow Models

- Developed in Contextual Design by Beyer and Holtzblatt
- Builds model of how information, physical objects flow through the environment and between people
- By looking through data, or collecting directly in-situ helps understand bottlenecks, smooth-points in interaction

# Examples:

Physical Flow Model



# Grounded Theory/Affinity Diagrams

- A tool used to organize large amounts of qualitative data into logical and linked categories based on recognizable relationships.
- Helps to generate holistic explanations of interrelated phenomena.
- Provides the foundation of inductive explanations.
- Accommodates brainstorming for solutions to problems.
- What it does *\*not\** do:
  - ▣ Test hypotheses
  - ▣ Prove/disprove theories

# What an affinity looks like



# Where the Affinity Method Comes From

- Japanese Anthropologist Jiro Kawakita (KJ Method)
  - Crisis of method: standard field techniques weren't working
  - Rejects the imposition of preconceived ideas and hypotheses
  - Inductive method for the “holistic integration of qualitative data” (examining interrelationships between phenomena)
  - Used the method for technological innovation! (ropelines and pipelines for the Nepalese Sikha Valley villagers)
- Hugh Beyer and Karen Holtzblatt
  - Psychologist and anthropologist
  - Adapted the affinity from the KJ method being taught currently
  - Different, as well, from the original KJ method
  - Popularized the affinity method in the design and HCI community in the U.S.

# Steps in the Basic Method

- Qualitative Fieldwork and Data Collection
- Creating Post-It Notes (or Data Cards)
  - 1-2 Sentences
  - Try to get a single idea on the note
- Putting up the notes
  - “Memory game” problems
  - Bucketing problems
- Grouping the notes
  - Check the interpretation of the note
  - Do the “sniff test” (group the notes based on their affinity to each other)
  - Label the groups
  - Create groups of groups, in a hierarchical tree-like diagram, eventually bringing all the data together to tell a single story

# How To: Rules to Work By

- **Creating the Team**
  - Who?
  - How many?
- **Grouping the Items**
  - Think about design implications
  - Think about the research questions
  - Think about what the research is meant to inform
  - Think about how your perspective is biasing the interpretation
- **Working as a Team**
  - Read each note aloud as you put it up
  - Talk about what “goes with” what until the groups make sense (negotiated truth)
  - Be open to other people’s interpretations

# Pitfalls of the Team-Based Affinity

- Team biases
- “While PD [product development] team members group customer needs based on how the firm *builds* the product, customers instead group needs based on the way they *use* the product.”

Source: Weitz, B., and R. Wensley, eds. *Handbook of Marketing*. Sage Publications Ltd, 2002. ISBN: 9781412921206.

© Sage Publications Ltd. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

- Bucketing
  - ▣ By key word
  - ▣ By possible solution
  - ▣ By previous results from past affinities
- Losing touch with the data
- Being afraid to go beyond the data

# Variations of the Method

- The Beyer and Holtzblatt Method (B-H)
  - ▣ Uses researcher insights from the data
  - ▣ Rapidly generates descriptions, furthest from the data
  - ▣ Better if you want to quickly devise possible solutions for the problem/issue at hand
- The Original KJ Method (KJ)
  - ▣ Uses researcher summaries of the data
  - ▣ Rapidly generates explanations, closer to the data
  - ▣ Better if you want to understand the complexity of the situation being studied
- The Grounded Theory Method (GT)
  - ▣ Uses the data itself
  - ▣ Semi-rapidly generates hypotheses, closest to the data
  - ▣ Better if you want predictive explanations of behavior that can be used for other projects

# Basics of the GT Affinity

## (Part 1)

### What is it: Inductive Hypothesis Generation

- ▣ Item level (create the post-its)
  - *“As analytic categories emerge, pull all the data (that is, exemplars) from those categories together...”*
  
- ▣ Pattern level (create the groupings)
  - *“...and compare them, considering not only what [items belong] in each emerging category but also how the categories are linked together.”*
  
- ▣ Constitutive level (create the story)
  - *“Use the relationships among categories to build theoretical models, constantly checking the models against the data...”*

*(Quotes from H. Russell Bernard, 1998, Handbook of Methods in Cultural Anthropology, p.608)*

# Basics of the GT Affinity

(Part 2)

- Identifying Themes (Patterns)
  - Ask “What is this expression an example of?”
- Look for:
  - Repetitions (“topics that occur and reoccur”)
  - “Indigenous categories” (locally specific terms, expressions)
  - Similarities and differences (constant comparison method)
  - Analogies
  - Linguistic connectors (causal such as “because,” sequential such as “before,” conditional such as “if,” etc.)

*(From Ryan, Gery W., and H. Russell Bernard, 2003, “Techniques to Identify Themes,” Field Methods 15(1) 85-109)*

# Affinity Example

Yeah, I only have two friends on Facebook. That's all I really care about

I only use Facebook to keep up with my mom.

I wish I could completely hide my profile so no one else can find me.

I'm constantly changing the visibility of my posts. I have groups with just 1-2 people in them.

# Affinity Example (2)

I use Facebook to keep in touch with just a few people who are really close to me.

Once I've established my connections on Facebook, I don't want to be bothered by other people who aren't as close to me.

Yeah, I only have two friends on Facebook. That's all I really care about

I wish I could completely hide my profile so no one else can find me.

I only use Facebook to keep up with my mom.

I'm not in the market for new friends.

# Affinity Example (3)

Some people only want to share with close friends, not to the world

I use Facebook to keep in touch with just a few people who are really close to me.

Yeah, I only have two friends on Facebook. That's all I really care about

I only use Facebook to keep up with my mom.

Once I've established my connections on Facebook, I don't want to be bothered by other people who aren't as close to me.

I wish I could completely hide my profile so no one else can find me.

I'm not in the market for new friends.

# Affinity Example (3)

Some people only want to share with close friends, not to the world

I use Facebook to keep in touch with just a few people who are really close to me.

Once I've established my connections on Facebook, I don't want to be bothered by other people who aren't as close to me.

Yeah, I only have two friends on Facebook. That's all I really care about

I wish I could completely hide my profile so no one else can find me.

I only use Facebook to keep up with my mom.

I'm not in the market for new friends.

DI: Provide a means to set which groups can see a given post

DI: Provide way to see who could not see a given post

DI: Provide means to hide your profile from search results

# Evaluating the Results of All Methods

- Establishing trustworthiness
  - Comparison and contrast within the team
  - Triangulation with other analysis methods
  - Dialectical interpretation (ideally)
  - Inspection of results by the professional community
- Establishing usefulness
  - It's only beneficial if you can do something with it
    - Design ideas
    - Design guidelines
  - What does your team think?
  - What do other teams think?
- Establishing timeliness (ROI)

# Ideation

- Design ideas should be:
  - ▣ Inspired by data
  - ▣ Grounded in real-world observations
- In brainstorming, no idea is a bad idea
- Think beyond what people are doing today



# In-Class activity

- Join with other groups that observed same area (e.g. point of sale, navigation, etc.)
- Perform affinity analysis from observations recorded from last week
- Identify first and second-level groupings
- Brainstorm design ideas for applications based on data

# Android Fundamentals

Anatomy of an app

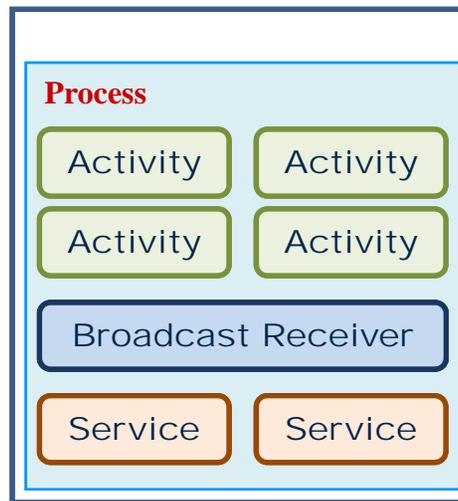
Basic APIs

# Anatomy of an Android Application:

- **Applications:** Process (set of screens)
- **Activities:** Application components (screens)
- **Intents:** Messages among components (what tasks an activity can perform)
- **Services:** Background tasks that can be performed without an application-specific UI visible

# Android application model

- One application (apk file) = one process



- Processes are isolated
  - IPC is done through Intents or Services

# Major app components

---

- Activity
- Service
  - runs in the background, little/no user interaction
- Broadcast Receiver
  - receives and handles Intents
- Content Provider
  - makes an app's data available to other apps

# Components – Activity

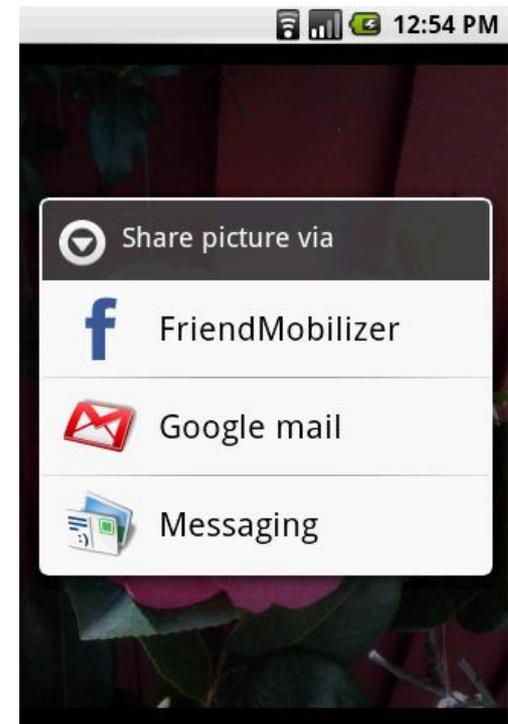
- Single, focused thing that a user can do
  - Consists of a hierarchical collection of Views
  
- One activity = one screen in app
  - Current activity starts next one (next screen)
  - One activity marked to be shown at app launch
  - Window does not have to be full screen
    - floating, embedded within another activity

# Components – Service

- Used for background tasks
  - e.g. site polling, data synch, network download
  - CPU intensive (e.g. MP3 playback) or blocking (e.g. networking) services should spawn their own thread
  - Can run when application UI is not visible (unlike iPhone)
- Also used for IPC
  - Android Interface Definition Language (AIDL)

# Intent

- Forms the glue between Activities
- An abstract description for
  - an operation to be performed
  - something that has happened



© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

- **Syntax:**

```
startActivity(new Intent(ACTION_DIAL, Uri.parse("tel:#####")));
```

# Example Intents

<code>ACTION_VIEW</code>	<code>content://contacts/people/1</code>
<code>ACTION_DIAL</code>	<code>tel:16175551212</code>
<code>ACTION_SEND</code>	Extras for subject, text, recipients, data, etc.

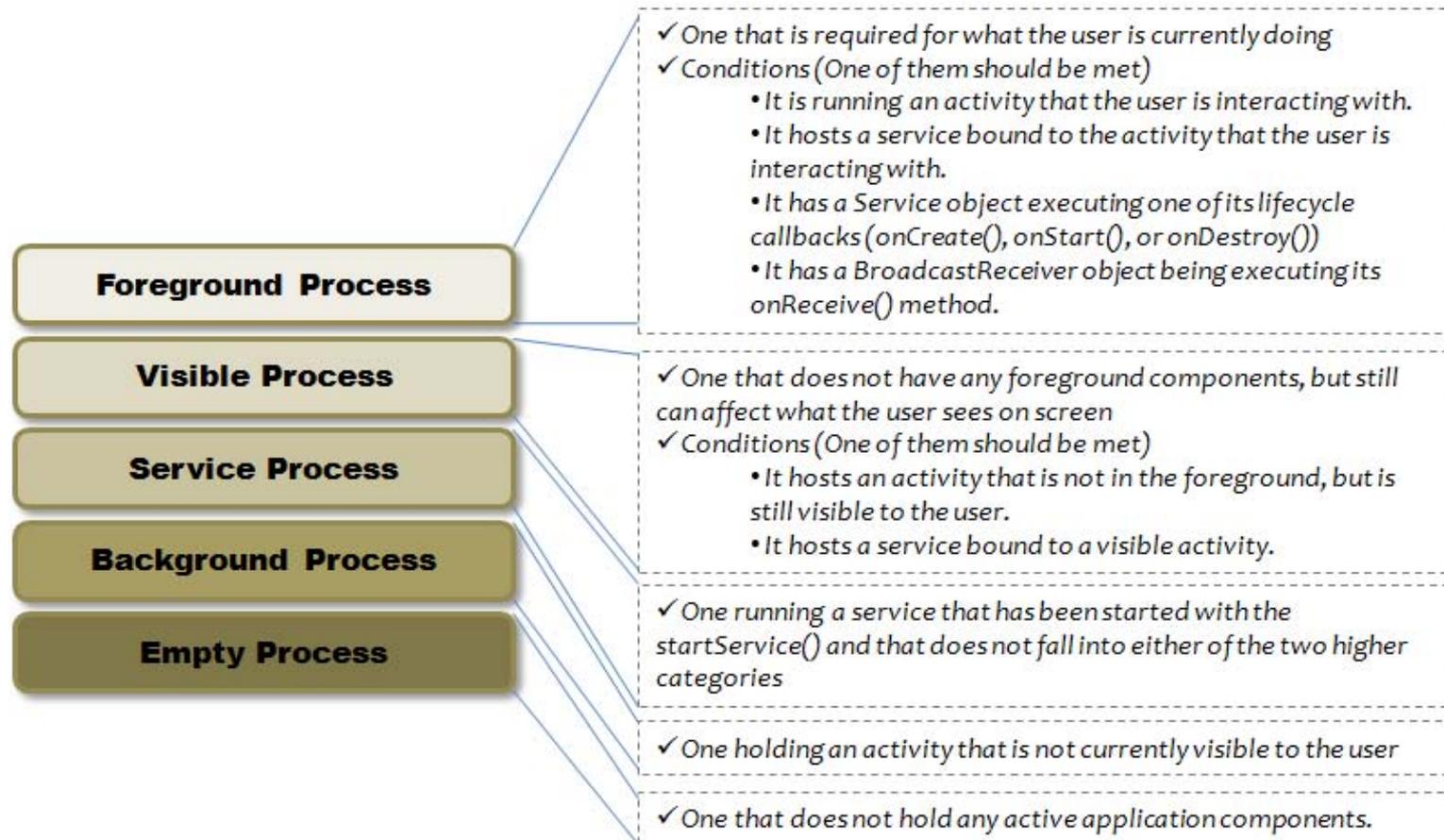
Action

Data

Intent = "Show the data pointed to by this URI"



# Activity priority list



# Attaching Views to code

```
<EditText
    android:id="@+id/usernameField"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
/>
```

**main.xml**

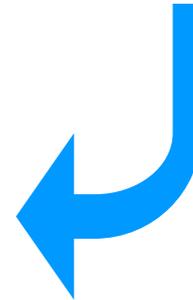
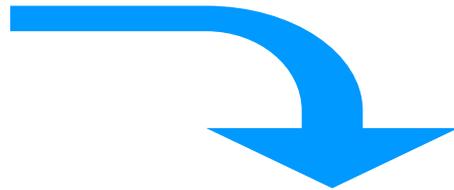
**myactivity.java**

```
setContentView(R.layout.enter_login);

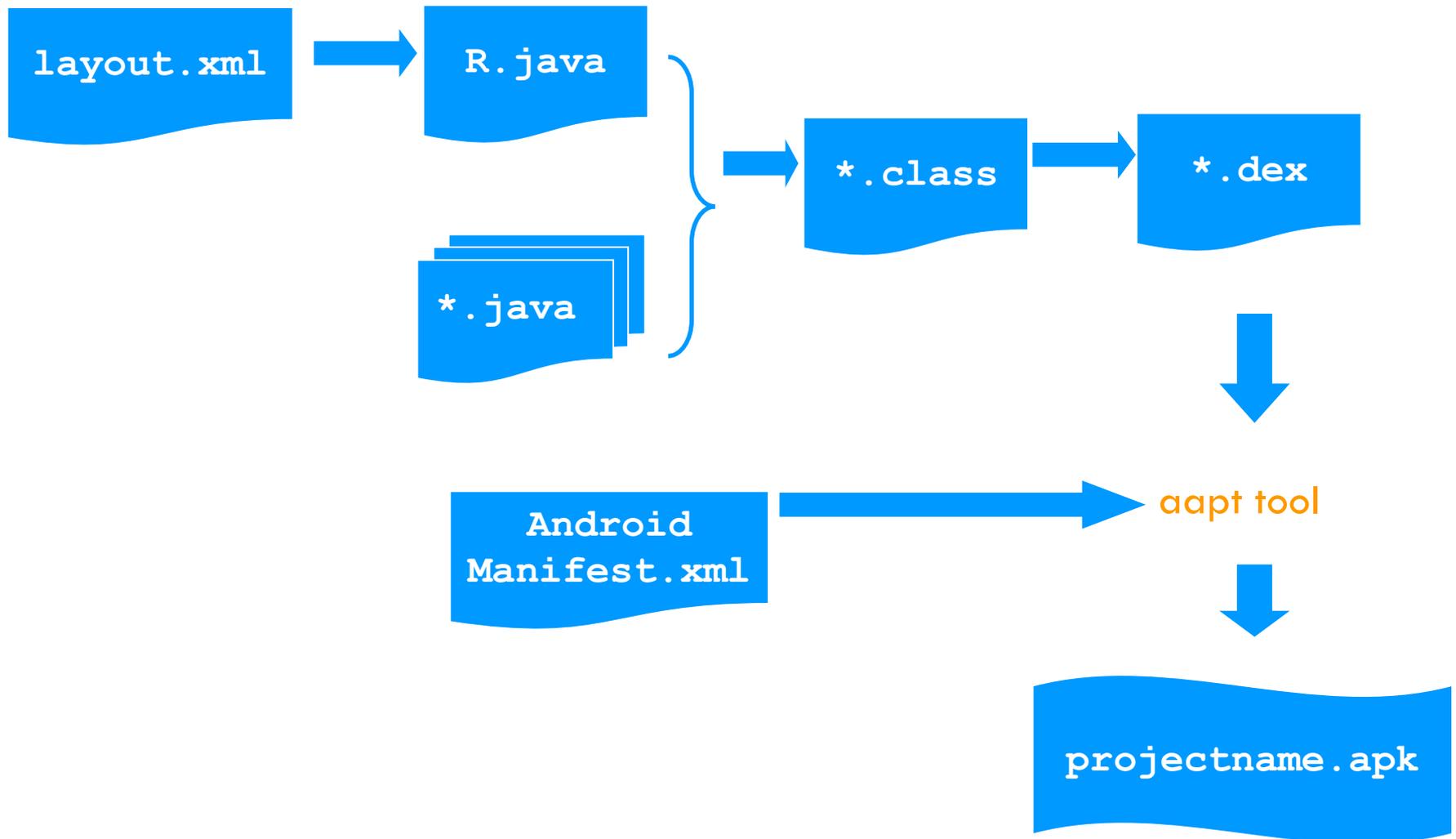
usernameField =
    (EditText) findViewById(R.id.usernameField);
```

```
public static final class id {
    public static final int
        usernameField = 0x7f050001;
}
```

**R.java**



# Putting everything together



# Loading an APK on your device

- Generate APK file in eclipse
- Use “adb” tool in the Android SDK
- Enable debugging tools in settings on device
- Plug in phone with USB cable
  - ▣ Charging only mode
- Run “adb install nameofapk.apk”
  
- To take a screenshot of app run “ddms”
  - ▣ Useful for posters, final reports

# Other useful APIs

## □ HTTP

- ▣ Uses Apache Commons library

```
String url = "http://web.mit.edu/";
HttpClient client = new DefaultHttpClient();
HttpGet request = new HttpGet(url);
Try
{
    HttpResponse response = client.execute(request);
}
```

## □ Accelerometer

- ▣ Good example code here:

<http://mobilehealth.posterous.com/example-for-accessing-the-accelerometer-with>

# iPhone Development

- Need to have a paid Apple Developer account to launch app on a real device
- Generate certificate with UDIDs of devices
- Distribution
  - ▣ Debug load directly on phone
  - ▣ AdHoc distribution (.mobileprovision file + app bundle)
  - ▣ iTunes store (1-9 month approval time)

# Objective C

- Superset of C
  - Can Mix C/C++ and Objective C
  - Single Inheritance
  - Loosely typed (treat compiler warnings seriously!)
- Syntax:
  - [instance method];
  - [instance method:arg1 arg2name:arg2];

# Strings, Logs, and Arrays

## □ Strings

- ▣ NSString \*myString = @"my string";
- ▣ [NSString stringWithFormat:@"with number: %d",5];

## □ Logging

- ▣ NSLog(@"debug info here");

## □ Arrays

- ▣ NSArray \*array = [NSArray arrayWithObjects:@"One", @"Two", @"Three", nil];
- ▣ If any of your objects is nil, array will not be full!!

# View Controllers

- Application contains a UINavigationController
- Each screen is a UIViewController
- New screens appear with a push of a View Controller onto Navigation Controller:

```
[[self navigationController] pushViewController:targetViewController animated:YES];
```

# Application Lifecycle

- Only one (non-Apple) application can be running at a time
- Applications suspended when phone sleeps or when interrupted (e.g. incoming call)
  - ▣ On wake-up, `-(void)applicationDidBecomeActive` called on `AppDelegate`
  - ▣ All state maintained, but no execution occurs while application is inactive

# iPhone resources

- <http://www.stanford.edu/class/cs193p/cgi-bin/index.php> Stanford iPhone Class
- <http://ericasadun.com/> Erica Sadun's iPhone Cookbook
- <http://www.cocoabuilder.com/archive/bydate> CocoaBuilder
- <http://cocoadevcentral.com/articles/000082.php> CocoaDevCentral: Cocoa Style for Objective-C: Part I
- <http://www.iphonesdkarticles.com/> iPhone SDK Articles
- <http://cocoadevcentral.com/> Cocoa Dev Central
- <http://icodeblog.com/> iCodeBlog
- <http://theocacao.com/document.page/510> Theocacao
- <http://idevkit.com/forums/tutorials-code-samples-sdk/30-custom-uitableviewcell.html> Custom UITableViewCell - iDevKit
- <http://pegolon.wordpress.com/2008/11/15/using-uitableviewcell-with-interfacebuilder/> Building UITableViewCell with IB
- <http://cocoawithlove.com/2008/12/heterogeneous-cells-in.html> Heterogeneous cells in a UITableViewController
- [https://www.nearinfinity.com/blogs/scott\\_leberknight/iphone\\_bootcamp\\_blogs.html](https://www.nearinfinity.com/blogs/scott_leberknight/iphone_bootcamp_blogs.html) iPhone bootcamp blogs
- [http://www.sleberknight.com/blog/sleberkn/entry/iphone\\_bootcamp\\_day\\_4](http://www.sleberknight.com/blog/sleberkn/entry/iphone_bootcamp_day_4) iPhone bootcamp blogs 2
- <http://www.iphonedevsdk.com/forum/iphone-sdk-development/4879-uitableview-cell-deletion-methods.html> UITableViewCell deletion methods
- <http://savoysoftware.com/blog/enhancing-performance-iphone>
- <http://stackoverflow.com/questions/328391/last-indexed-cell-in-uitableview-is-taking-on-wrong-font> Cell Identifiers
- <http://stackoverflow.com/questions/tagged/iphone> StackOverflow
- <http://www.cocoa-dev.com/index.pl?NSUserDefaults> NSUserDefaults
- <http://knol.google.com/k/usman-ismail/iphone-sdk-application-preferences/34oprzanmpe7q/8#> Application Preferences tutorial
- <http://icodeblog.com/2009/02/02/great-resource-for-all-iphone-developers-ibetatestcom/> iBetaText.com
- <http://blog.coriolis.ch/2008/11/09/add-an-uiprogressview-or-uiactivityindicatorview-to-your-uialertview/> progressView
- <http://idevkit.com/forums/general-sdk/299-nsurlconnection-nshttpcookie.html> NSURLConnection, NSHTTPCookie - iDevKit
- <http://stackoverflow.com/questions/576265/convert-nsdate-to-nsstring> Convert NSDate to NSString - Stack Overflow
- <http://www.cocoa-dev.com/index.pl?DescriptionWithCalendarFormat> CocoaDev: DescriptionWithCalendarFormat
- <http://www.planetcocoa.org/> Planet Cocoa

# Written proposal for next class...

Length: 10 page max (including figures, page count does not include “Front Matter”). Although a single student may be serving as editor and content gatherer, all students in the groups are required to author sections of the proposal related to their chief area of responsibility.

## **Front Matter**

Title page: name of project, names of team members, group email address, type of report (proposal), and current date.

Abstract: one paragraph, ca. 150 words; state the problem, methods, expected results; no figures or references in abstract; do not use first person pronouns.

Table of contents

List of figures, if you have four or more. Figures should be numbered and labeled.

## **Body**

Introduction: background motivation for the project. This section establishes the need for the project; state primary and secondary audience.

Statement of objectives: the clear objectives set for your project, purpose of the service/site; its scope.

Description of project: makes us “see” the project by describing proposed look and feel (use figures and flowchart), design strategies you will employ, technical requirements, tools needed and how you will acquire them, any platform/browser dependencies.

Tasks and milestones: show a Gantt chart which divides the life of the project into definable tasks (vertical axis) over time in weeks (horizontal axis). Punctuate the horizontal axis with important milestones you are expected to meet.

Roles each team member will perform.

Please Note: All figures are given a caption and a figure number (placed below the figure) and are referenced in the text (“See Figure 1”). Figures should be placed within the text as close as possible to the reference.

## **End Matter**

References (if applicable)

# Oral Presentation for next class...

## □ Oral Presentation Format

- **Time limit: 8 minutes (max.), followed by 7 minutes of Q & A.**
- **Introduction: background motivation for the project. This section establishes the need for the project and states the primary and secondary audience.**
- **Statement of objectives (be specific as possible)**
- **Description of project: makes us “see” the project by describing proposed look and feel, design strategies you will employ, technical requirements, tools needed and how you will acquire them; note any platform/browser dependencies.**
- **A preliminary mockup (can include visuals, wireframe, flowchart)**
- **Project timeline**
- **Gantt chart.**
- **List of deliverables: what you can realistically finish by the end of the semester.**
- **Project team roles.**
- **Q &A**

# Hello World

- Create an application that displays “Hello 21 w.789” and has a button. When this button is pressed, the text should change to “Goodbye 21 w.789”
- Install this application on your group development phone and show it in class at the end of your presentation

MIT OpenCourseWare  
<http://ocw.mit.edu>

21W.789 Communicating with Mobile Technology  
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.