THE UNIVERSITY OF ARIZONA

MOBILE MATTERS
SYMPOSIUM 2012

App Development Best Practices

Jonathan Sprinkle, ECE

Wayne Peterson, UITS

An old chestnut...

Catch a fish for a man,
Feed him for a day.

Teach a man to fish, and he will ask,
"Will salmon eggs be on the final?"

David P. Murdock
Professor of Physics, TTU

THE UNIVERSITY OF ARIZONA

It’s the most amazing software course yet.

ECE473-573, MW 4:00-5:15, (Sprinkle)

http://www.ece.arizona.edu/~ece473/

Focus of ECE473-573

- Without *reflection*, deep learning is not really possible
- Focus for engineering education is on *lifelong learning habits*
- Consider then the following two strategies:

Focus of ECE473-573

1. Anchor 4th year undergraduates to phone programming, in order to prepare them for potential careers in phone programming. Use a course steeped in tutorials and demonstrations in order to allow them to understand the complex APIs and platform details, and to **deliver a full-fledged application within the confines of a single term course**

Focus of ECE473-573

2. Treat 4th year software engineering as a traditional software engineering course, focused on the development and delivery of requirements, validated through design, test, and validation of applications developed on the mobile phone platform of the student's choice. **Students are responsible for learning the details of their device and its APIs on their own**.

Requirements

- Teams must produce requirements for their application, which are approved, and which become the contract for their grade
- Two tiers of requirements:
  - B-requirements: satisfaction allows a maximum grade of "90" for project implementation
  - A-requirements: satisfaction allows a maximum grade of "100" for project requirements

- If there is no test for a requirement, then the requirement is considered unmet!!!
  - This is why games are terrible ideas for software engineering courses!
    - How do you test, “fun to play, with great graphics” ??
- And…a twist
  - Instructor can modify requirements at any time during the semester
  - Encourages regression testing, and modular design

How do we do it?

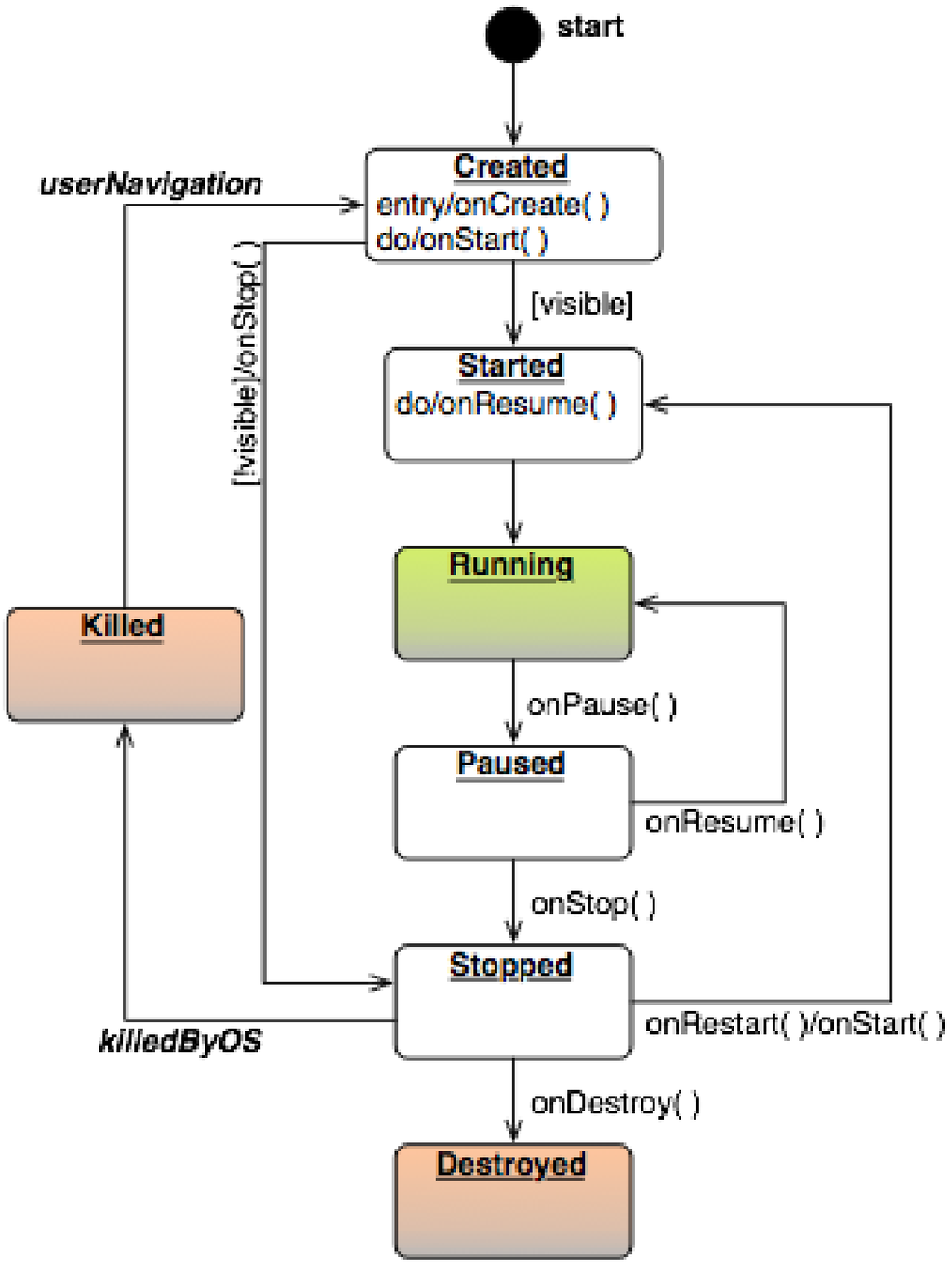
- Focus on fundamentals of design
- Brief descriptions of application best practices
  - Rewrite documentation in the form of formal models and formal design
- Require students to learn the rest, and follow best practices
- Hand out bad grades if the requirements are not met by tests

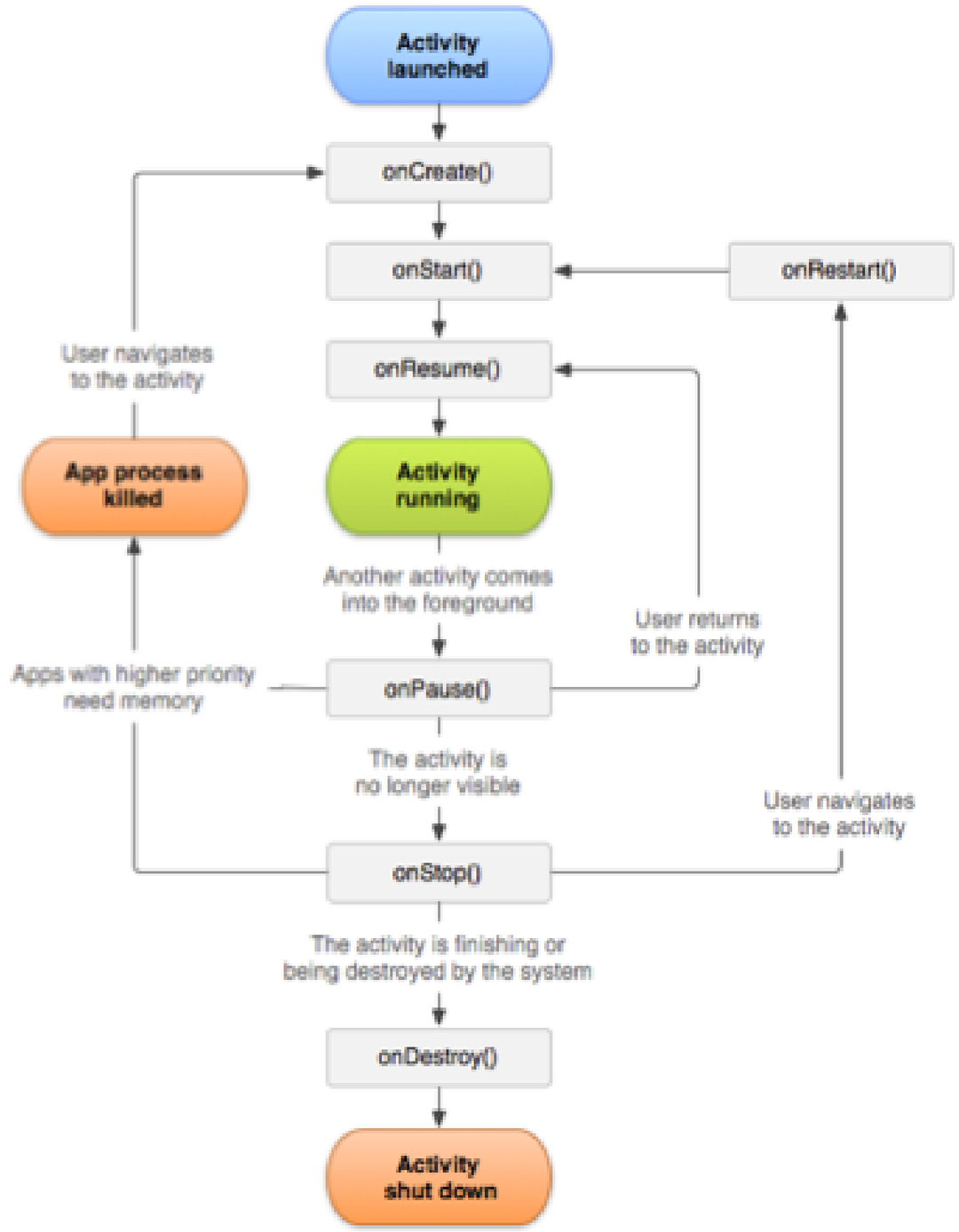
Example: Application Lifecycle

- All applications go through some lifecycle
- You need to learn
  - when to write data to memory, vs. when you can keep that data in your variables,
  - what data to write between different executions, depending on your requirements,
  - whether your app can be killed immediately after (or during) this method
- We adapt the (terrible) semi-activity diagrams in the APIs to state models to understand the process

And, we do it for each platform

THE UNIVERSITY OF ARIZONA





STATE Diagram

"Activity" Diagram

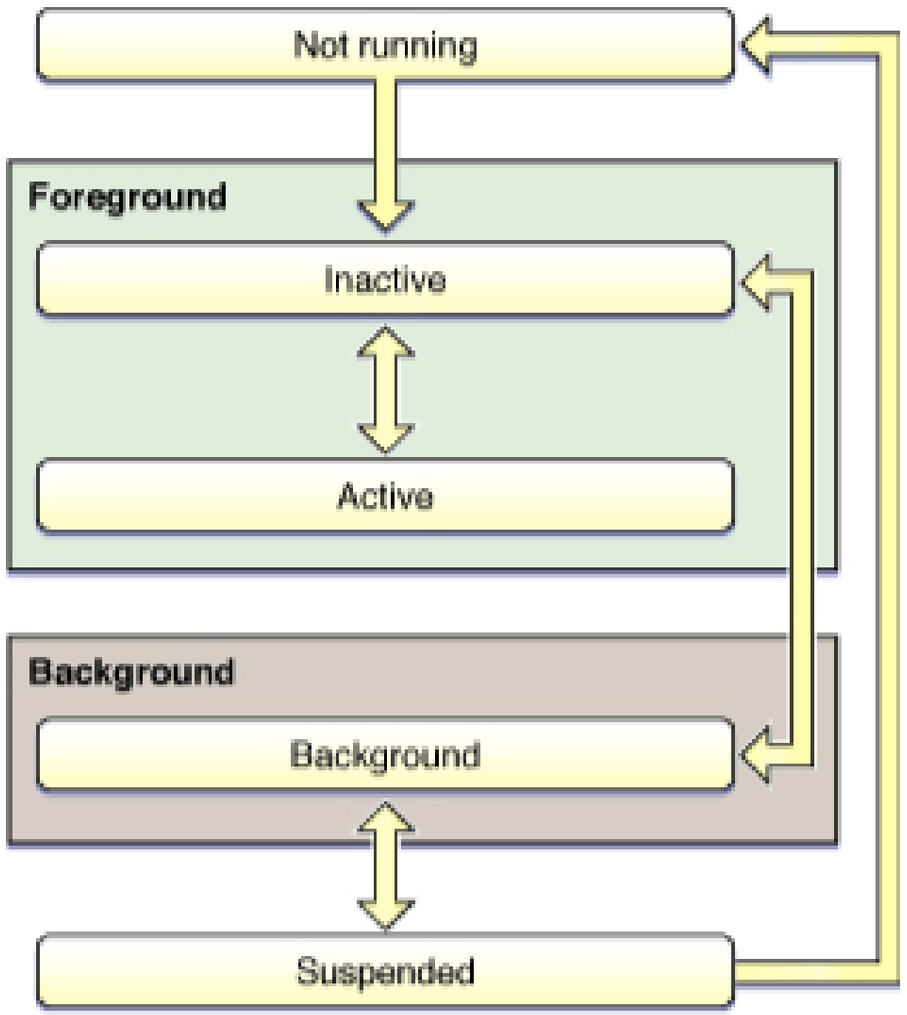
Table 3-1 App states

State	Description
Not running	The app has not been launched or was running but was terminated by the system.
Inactive	The app is running in the foreground but is currently not receiving events. (It may be executing other code though.) An app usually stays in this state only briefly as it transitions to a different state.
Active	The app is running in the foreground and is receiving events. This is the normal mode for foreground apps.
Background	The app is in the background and executing code. Most apps enter this state briefly on their way to being suspended. However, an app that requests extra execution time may remain in this state for a period of time. In addition, an app being launched directly into the background enters this state instead of the inactive state. For information about how to execute code while in the background, see "Background Execution and Multitasking."
Suspended	The app is in the background but is not executing code. The system moves apps to this state automatically and does not notify them before doing so. While suspended, an app remains in memory but does not execute any code. When a low-memory condition occurs, the system may purge suspended apps without notice to make more space for the foreground app.

Really, Apple?

REALLY?

Figure 3-1 State changes in an iOS app



So this is how the world ends. Not with a bang, but with an “activity” diagram.

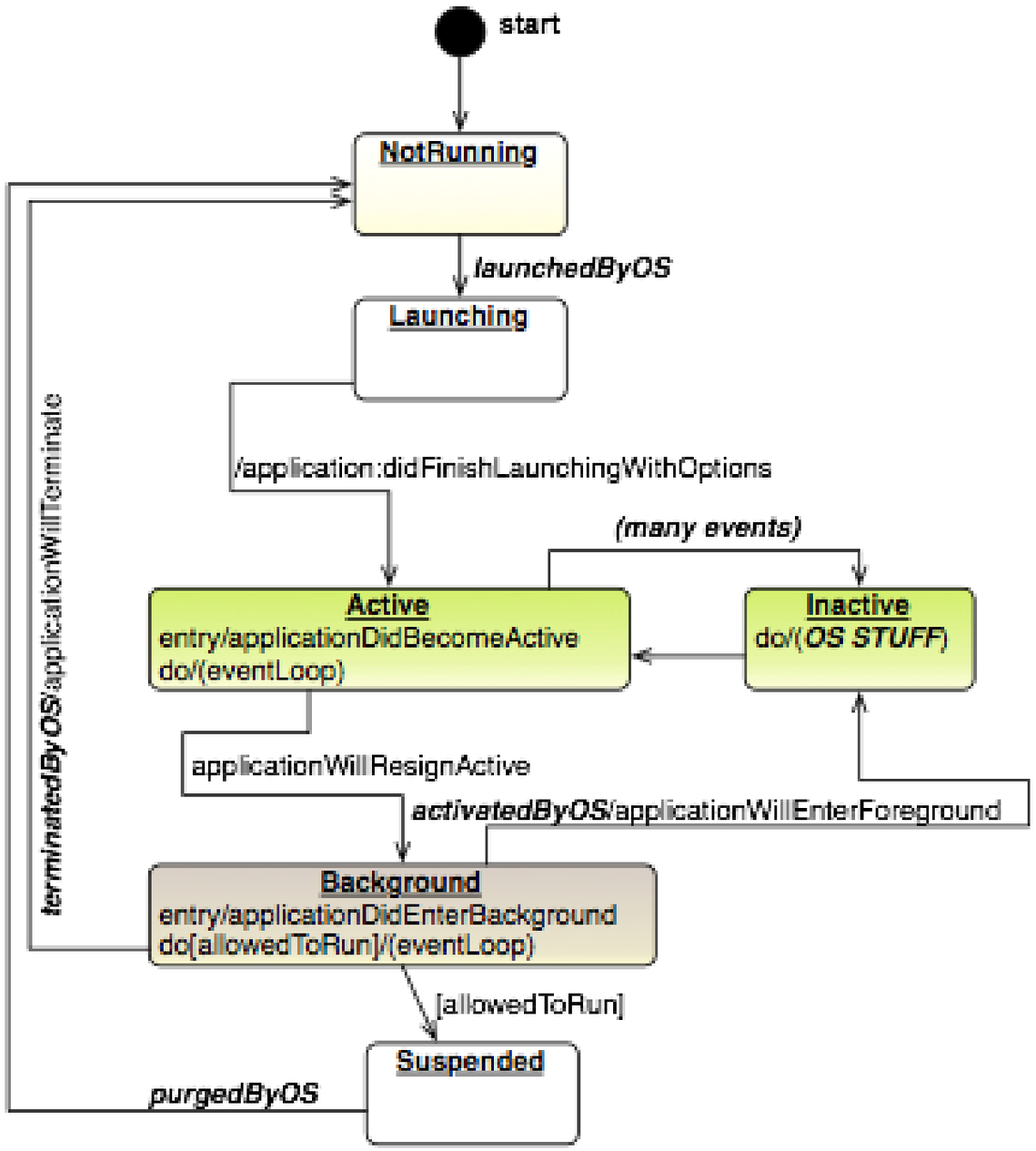
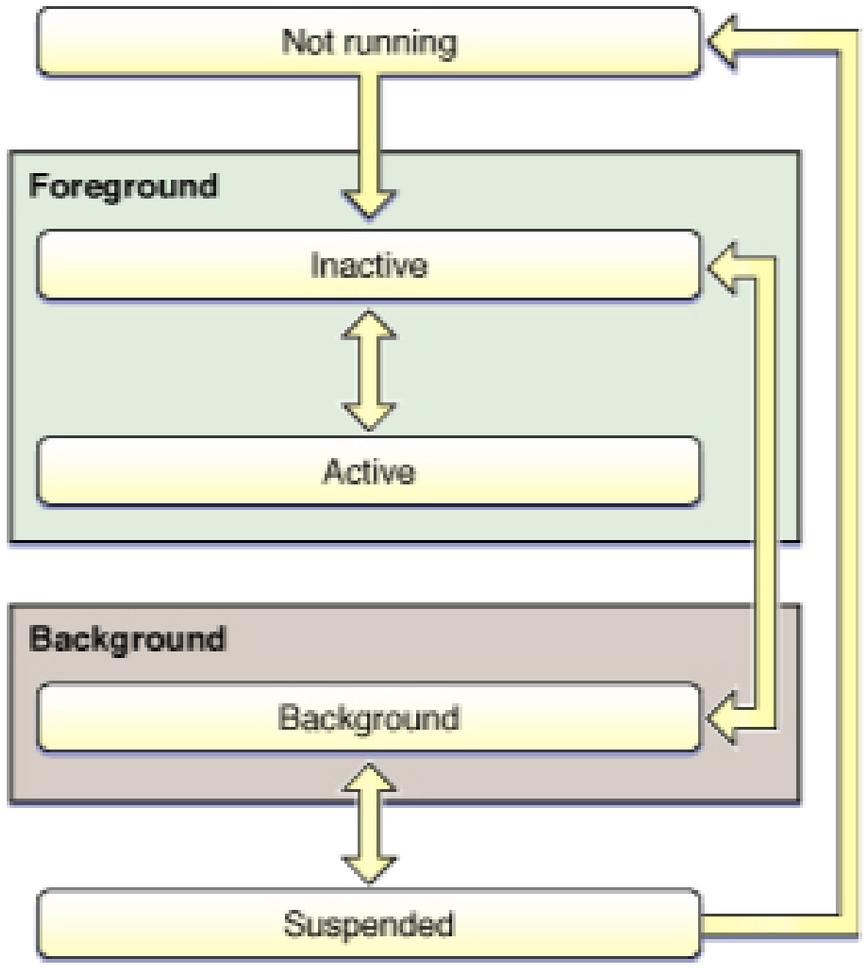


Figure 3-1 State changes in an iOS app



STATE Diagram "Activity" Diagram

HOMEWORK 1
ECE473-573

d2l: `hw01`

Due: 26 January 2011, 11:59 PM

This assignment ensures that you can load the SDK and emulate (or simulate) a simple application that you create. *Use concepts of good style and object-oriented design (as much as you know them) when you complete this assignment.* As always, document your code thoroughly.

All the below solutions should be contained in a single application that has a selector that determines (and clearly indicates) which solution to the three below problems is being displayed at any given time.

---

1 Your Name Here (10 points)

Create an application that shows your name, and your course number (either ECE473 or ECE573), and this homework assignment id (together called the "assignment text") on the phone screen, in two lines. Touching or rotating the screen should not change the displayed information. E.g.,

```
Firstname Lastname
ECE473 hw01
```

2 Your Name There (45 points)

When touching the display surface where the assignment text could be displayed, the assignment text should move to a new location, but should still be visible.

3 Your Your Name Name There There (45 points)

Double-tapping the display surface should move the assignment text, but single touches should have no effect.

ECE573 Students Only

Your application should satisfy these additional requirements:

- The entirety of the assignment text should be visible at all times.
- When the assignment text moves, its travel should be animated*.
- For solutions to questions 2 and 3, if the device is rotated between portrait and landscape mode, then the assignment text must be appropriately rotated.

Course Results:

- Most undergraduates choose Android over iOS
  - Undergrads: 27 (Android) vs. 15 (iOS)
  - Grads: 10 (Android) vs. 10 (iOS)
  - This is additionally because, in a team environment, everyone must be capable of running the same phone IDE -- rules out iOS for almost all teams, unless students are passionate about that OS
    - Grad student teams are teams of 1
  - At least 6 students have gone on to work at either Apple or Google writing software

Course Results (2):

- Students inform that this course was instrumental in their getting an interview or a job
- Several students tell that they used their mobile phone app while on their interview to show an example of the things they did in their coursework
- Approximately 12-15 apps developed by course students have been deployed to the app store for either iOS or Android

Transition to demo

- Students who succeed in this class are prepared to be successful in mobile app development
- UA (and other) stakeholders can now depend on a group of students to deliver a final version that obeys the necessary requirements--and is delivered with tests to verify those requirements!
- There are many pitfalls in app development that are not immediately apparent to good developers who have never developed for apps before