ATLS 4320: Advanced Mobile Application Development
Week 1: Course Overview, Agile, and Navigation

Class Overview

Syllabus
Canvas
Github repos
Project journals
Agile semester-long project

Project Management Process

The waterfall development method has been around since 1970 and treats the software development process as a single path through different phases where each phase is completed before the next starts.
Analysis – design – development – testing – deployment – maintenance
The advantage of the waterfall process is that each phase is well defined and forces the client and development team to spend time thinking thought the requirements and design up front.
The disadvantage is it creates a long development process where all requirements and design must be done up front and there's no opportunity to modify, improve, or fine-tune along the way.
This comes more from the hardware industry and is being used less in the software industry.

The agile development method takes an iterative approach where you break up projects into smaller chunks of work called iterations, or sprints, so you can quickly define, create, and iterate. Each sprint receives feedback and then priorities and tasks are set for the next sprint. This allows for client feedback, early user testing, time to react to business changes, and the ability to modify and fine-tune along the way if needed. It provides the agility to both create and respond to change. Some form of the agile development process is most commonly used today for software development.

Agile Methodology

Agile is a development methodology where you break up projects into smaller chunks of work called iterations, or sprints, so you can quickly define, create, and iterate. This lifecycle makes it easier to adapt to user feedback and react to business changes. It provides the agility to both create and respond to change.

Agile Manifesto Principles https://agilemanifesto.org/principles.html

What is Agile? https://www.linkedin.com/learning/devops-foundations-lean-and-agile/what-is-agile-2?u=42275329

Model-View-Controller (MVC) Review

It's important to understand the MVC architecture as it's used in both iOS and Android

- Model: holds the data and classes
  - Should be UI independent
- View: all items for the user interface (objects in IB)
- Controller: links the model and the view together. The backbone or brain of the app.
  - usually subclasses from UI frameworks that will allow the view and the model to interact
  - Controllers are usually paired with a single view, that's where ViewController comes from
- The goal of MVC is to have any object be in only one of these categories.
  - These categories should never overlap.

- Ensures reusability

Navigation
https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/navigation/
App navigation should feel natural and familiar. It should complement the content and purpose of the app. In iOS there are 3 main navigation types:

Flat navigation
Tab bars switch between content views
https://developer.apple.com/design/human-interface-guidelines/ios/bars/tab-bars/

- provide a flattened hierarchy for your app
- provide access to several peer information categories or modes at once
- content is not passed between tabs
- Strictly for navigation
    - Don't confuse with toolbars to perform actions
- Clock, Music, App store

Hierarchical navigation
Navigation controllers w/table make one choice per screen until the desired destination is reached. To go to another destination, you must retrace your steps or start over from the beginning and make different choices.

- Use table views to present hierarchical content https://developer.apple.com/design/human-interface-guidelines/ios/views/tables/
- Navigation bars enable navigation through hierarchical content https://developer.apple.com/design/human-interface-guidelines/ios/bars/navigation-bars/
- Settings, mail, contacts

Content Driven/experience driven navigation
Move freely through content, or the content itself defines the navigation.

- The content itself defines the navigation
- Games, books, other immersive apps

Next week we'll start with flat navigation by looking at the tab bar controller.
We'll also be getting into advanced Swift and some new Swift 5.3 features next week. Any topics you want to review? Optionals?