➔ Mostly, everyone's project is too ambitious

- This is as much an exercise in "triage" as anything else
    - 3 weeks is ¼ of a drop in the bucket for a professional App developer
    - Lots of things will end up being huge amounts of work that you didn't expect
    - Your love/hate with the Swift compiler will only grow
    - 3 hours can disappear into fussing with a layout for a single label
    - And when something does work, horizons will open up and you'll want to expand for hours in that direction
- *WHAT really matters to the user versus what's just 'nice'?*
- Unit testing, unit testing, unit testing
    - Save a single string to disk
    - Navigate from a one-cell master screen to a one-label detail screen
    - Capture a single image to Camera Roll
    - Get the current GPS coordinates & precision and output them to a plain label
- Focus on iOS APIs – that's what you're here for. Use fake stub data for your data sources until the App itself works
    - Table editing is a lot more important than tricky JSON parsing
    - Layout – without good thought here, your App won't be usable
    - Try it on your device – the simulator is very limited
    - Good labeling of items, either well-known icons or text
    - **Touch, not type**
    - Responsive – tappable; zoomable; details available; saveable; deleteable
- When dealing with external sources:
    - Start with the simplest possible disk persistence: write your name to disk
    - Web data grabbing – single array of strings first, then structured data if you must
    - Don't scrape web pages
    - Use a reliable fast web service
    - DON'T write your own server / DB Schema – use one that's already available, or leave it for the next release

➔Several projects are just a re-do of the traditional MapKit demo: show a scrolling map based on user's GPS location, and allow them to tag it with pictures

- Need something based on your life/work experience to make it real / motivated
- Start with (of course, cited) Map demo code and expand it in a unique way
    - Think specialization: bike trips; winemaking tour; rock climbing journal

➔When you integrate a feature, it should (must) be *central* or *very important* to your App.

- If you're going to bother with Social Media, don't just make it an add-on that you leave for the last two days.

- Rather than Social Media as an add-on, we'd rather see core iOS like layouts, navigation, device rotation handling, GPS, and tagging your data with Audio and Image snapshots.

➔Give time limits, we will give generous partial credit, even for code that doesn't even compile yet

- BUT no excuses for spaghetti code
    - To get credit it must show good convention following (CamelCase; indentation; separate classes in separate files; no dead code snippets; good global constant declarations; thoughtful use of stored, computed & observed properties; MVC; full-sentence comments)
    - We can only spend so much time 'mining' very broken code for credit-worthy pieces, so document everything clearly that is broken and/or partially done.
- Obviously more credit for all of the above, but also compiles
- We will look at your StoryBoard closely, constraints, segues, attributes, so even if code is problematic, the StoryBoard should actually "tell the story" – lots of credit there
- Lots of credit for code that only crashes occasionally or on the edge cases (such as out of bounds mouse drag, no network available) – this is the 'A' level project
- And finally, for code that handles everything cleanly and you may be invited as a TF next year