Advanced Programming Techniques

## Course Project Details

Christopher Moretti

Approval Meeting - By 3/8

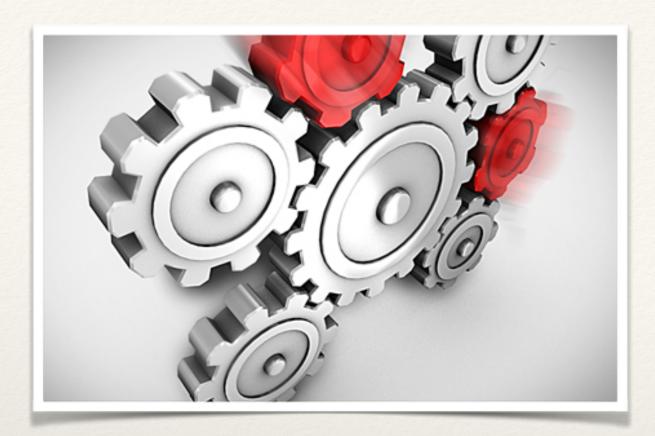


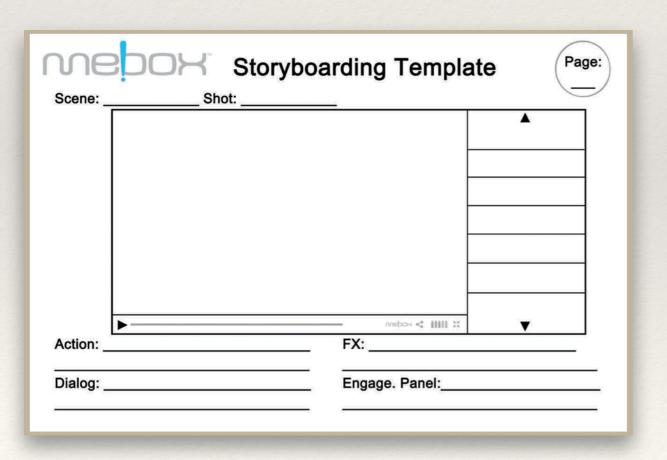
#### Design Document - 3/15











Design Document - 3/15





## "Project Manager"

- You will run your project!
  - \* It's your project to design and scope
  - \* It's your code to write and debug
  - \* It's your application to put in the hands of users to test
- \* We will help: one grad TA will serve as "Project Manager"
  - \* Some combination of product owner, client liaison, scrum-master, and actual product manager
  - \* We will advise and monitor so you don't get stuck
  - \* We will liaise to find other groups that have already overcome your current hurdles to remove blockers, as necessary

## CIRCLES Method

Comprehend situation why? How? Identify customer — 22 Personas

Report customer needs - as\_, I want\_so that

Cut, through prioritization → Rol estimate? List solutions

Evaluate trade-offs - thoughtful, analytical, obj

Summarize recommendation - What, recap, why is others

#### Weekly Meetings/Demos - starting 3/21



Project Website - 3/21



#### Project Website - 3/21







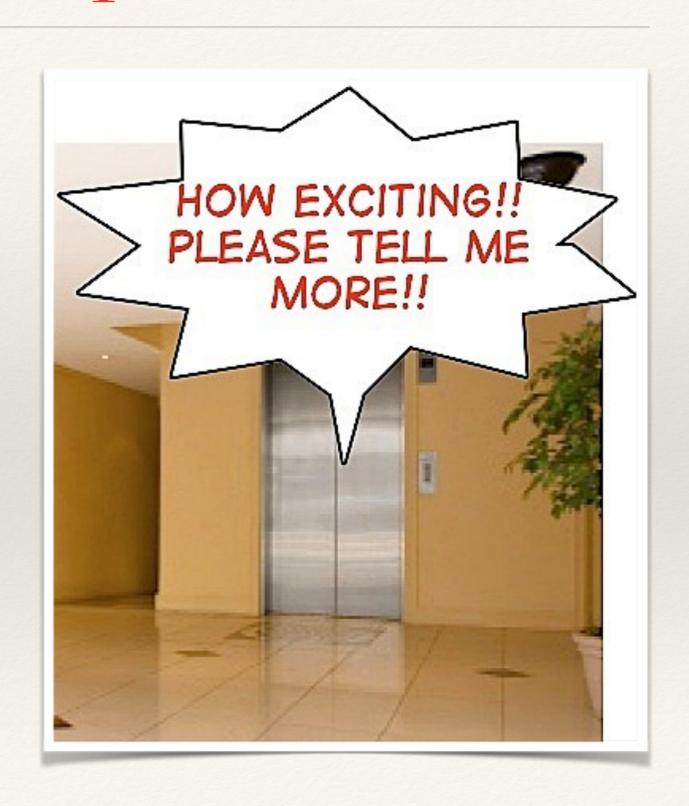
#### Elevator Speech

- \* Attention grabbing pitch to describe (sell) your idea in a soundbite, a sentence, a paragraph, a coffee break, etc.
- \* The idea isn't directly to get a funding commitment (but that's great if you do), but more to get a 2nd conversation.

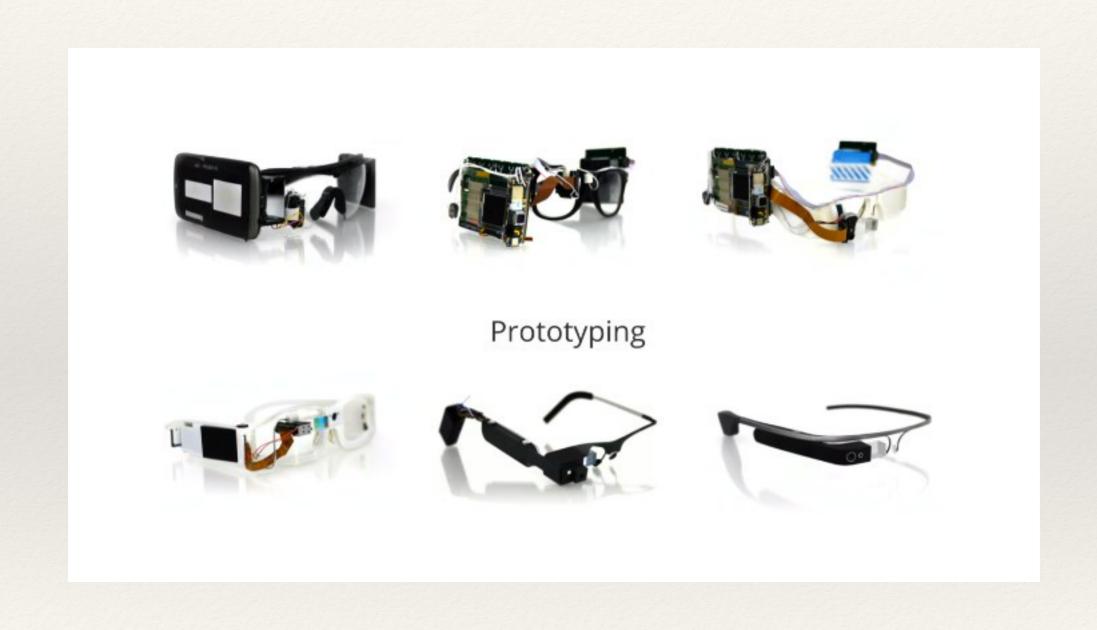


#### Elevator Speech

- \* One formula:
  - \* What's the problem
  - \* Why does the state of the art suck?
  - \* What are you going to do about it and how?
  - \* Where do you go from there?



### Prototype (March 28)





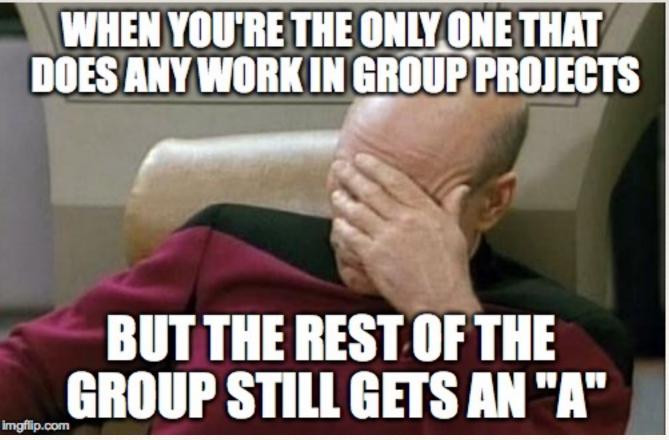






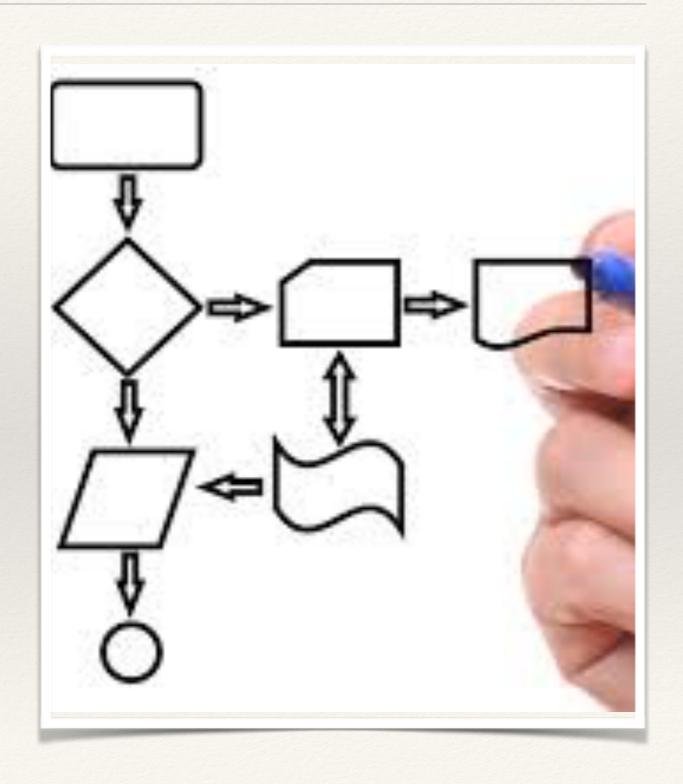
#### Peer Evaluation





#### Have a process to make progress

- \* This is **not** a (good) process:
  - \* talk about project at lunch
  - hack some code together
  - \* test it a bit
  - \* fix the obvious bugs
  - \* repeat from the top until the semester ends



# Do this from the start, and keep it up till the end:



- Keep scope and schedule in mind
- Keep a log of what you've done and what's up next on your plate
- Avoid a "big bang" project in all stages of planning
- \* Simplify: don't take on too big a job, don't try to do everything at once; do take "reasonable bites"
- \* Use source code control for **everything**.
- \* Test **everything**. Build automated tests from the beginning to perpetually keep your code in shape
- \* Remember that you have deliverables along the way!
- \* Remember that no battle plan survives the first encounter with the enemy.