Intro to Coding with Python– Designing

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

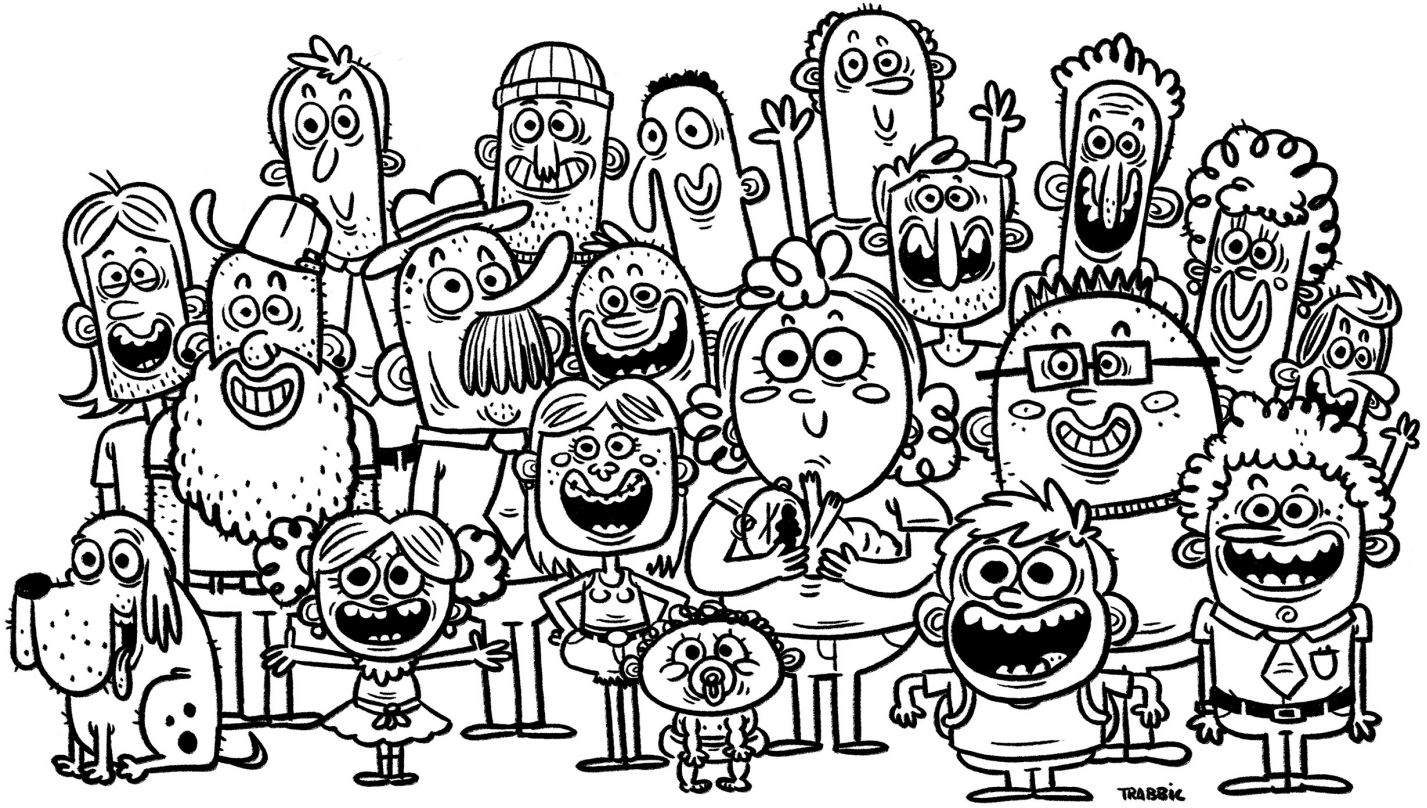
- User-centered design
  - What it is
  - Why do it
  - Ways to do it

Hypothetical example

“Advising Assistant”

Hypothetical example

- **Overview:** over 100 majors and minors in CS, CIS, and DS



Hypothetical example

- **Overview:** three advising faculty

Hypothetical example

- **Overview**: Degreeworks not hugely helpful…

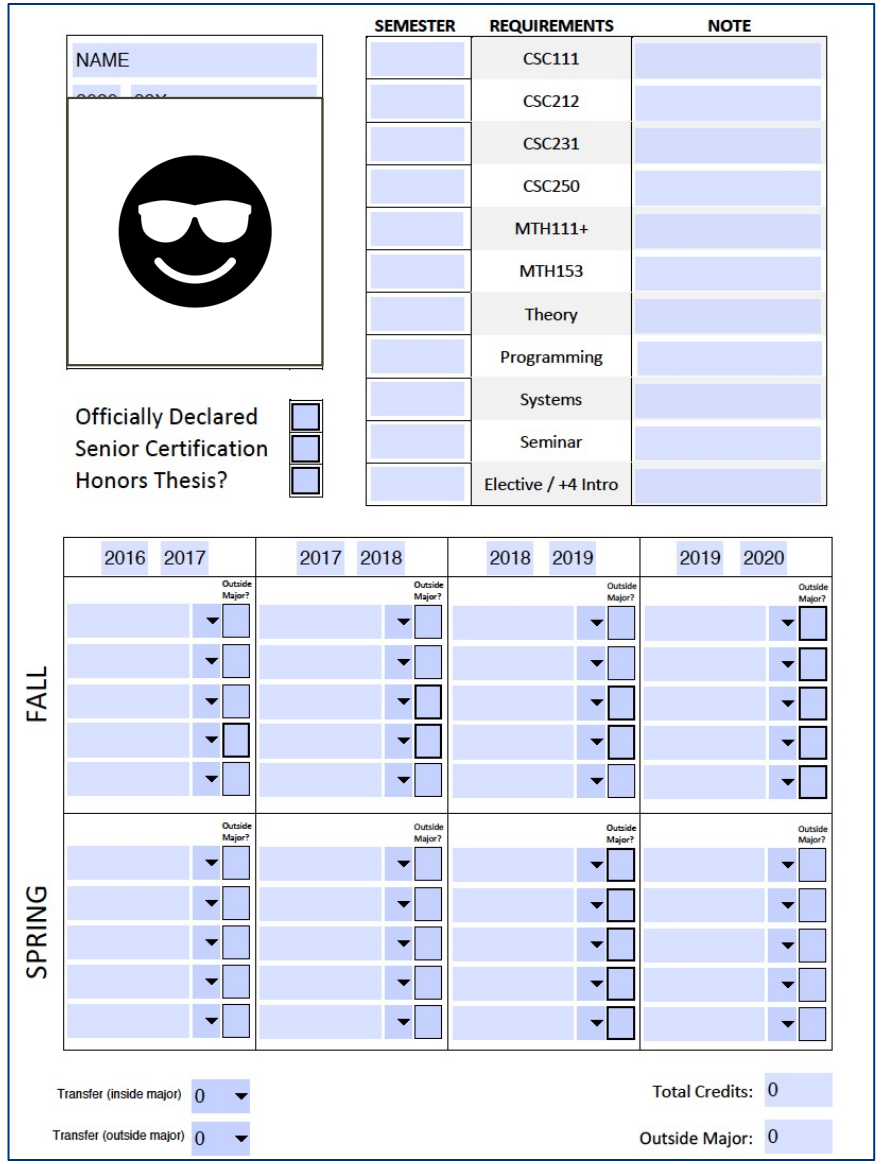
Hypothetical example

- **Overview**:

current process is **manual**

Goal 1:
detect which courses a student needs for the major and when they will next be offered

Goal 2:
give adviser an overview of all advisees



Hypothetical example

- **Ideal:** we would like to build a plugin that sits on top of Degreeworks and does everything for us, but that won't work because of FERPA concerns
- **Actual strategy:**
  - Define **mapping** from course numbers to major/minor designation
  - Use historical data to determine the rotation on which classes are offered
  - **Export** CSV of all advisees from Degreeworks
  - Build a **parser** that extracts data from unofficial transcript (i.e. courses taken) and joins with mapping, majors/minors
  - Build **authenticated frontend** for adviser to track student progress

Hypothetical example

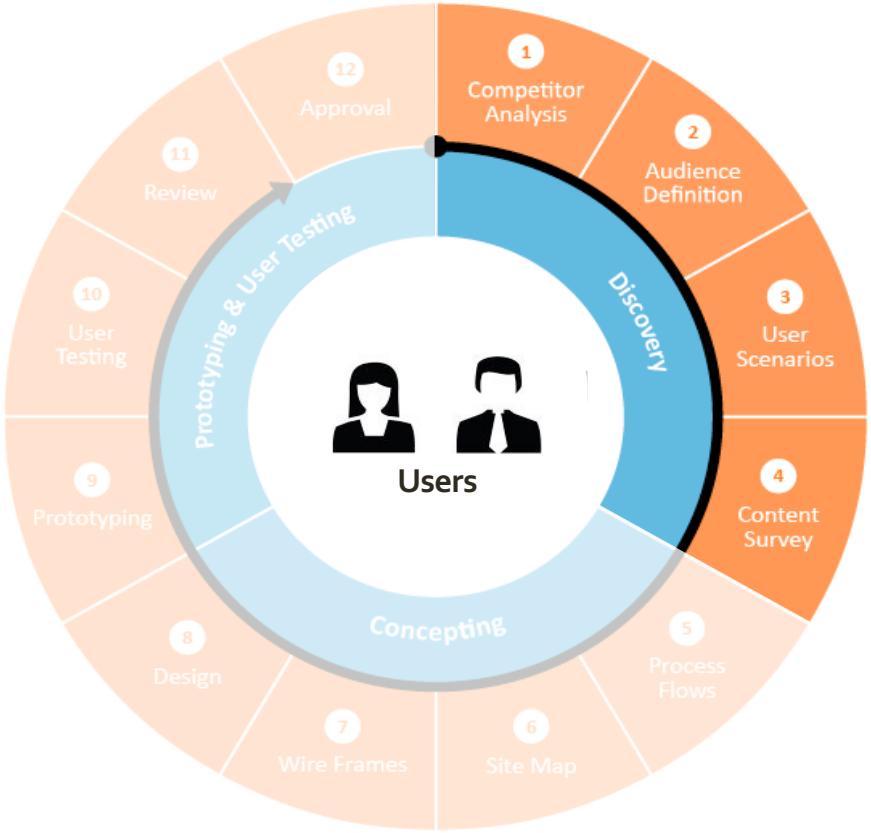
- **Ideal:** we would like to build a plugin that sits on top of Degreeworks and does everything for us, but that won't work because of FERPA concerns
- **Actual strategy:**
  - Define **mapping** from course numbers to major/minor designation
  - Use historical data to determine the rotation on which classes are offered
  - **Export** CSV of all advisees from Degreeworks
  - Build a **parser** that extracts data from unofficial transcript (i.e. courses taken) and joins with mapping, majors/minors
  - Build **authenticated frontend** for adviser to track student progress

Discussion

Let’s say you wanted to actually implement this;

where do you start?

User-centered design framework



1) Discovery

- Learning about your users
- Modeling your users
- Analyzing your users' tasks
- Eliciting and defining clear product requirements

2) Concepting Phase

- Developing conceptual models
- Solving design problems through ideation
- Detailed design activities

3) Prototyping + User Testing

- Delivery of a high-quality product that meets users' needs and is easy to learn and use

http://www.uxmatters.com/mt/archives/2010/07/design-is-a-process-not-a-methodology.php

Defining your audience

- Learning about their problem
  - Semi-structured interview
- Analyzing their tasks
  - Hierarchical task analysis
- Modeling users
  - Personas

Semi-structured interviews

- **Why?**
  - gather qualitative data about users to understand the problem
  - can help identify key differences between designer and target user
- **How?**
  - ask open-ended questions
  - bring along a "cheat sheet" to ensure that you gather all the information you need

- **Some tips**:
  - establish trust at the beginning
  - participant engagement will vary
  - be flexible, but make sure you get what you came for
  - consider recording or note-taking to help with recall

Wood, L. E. (1997). Semi-structured interviewing for user-centered design. *interactions*, 4(2), 48-61.

Defining your audience

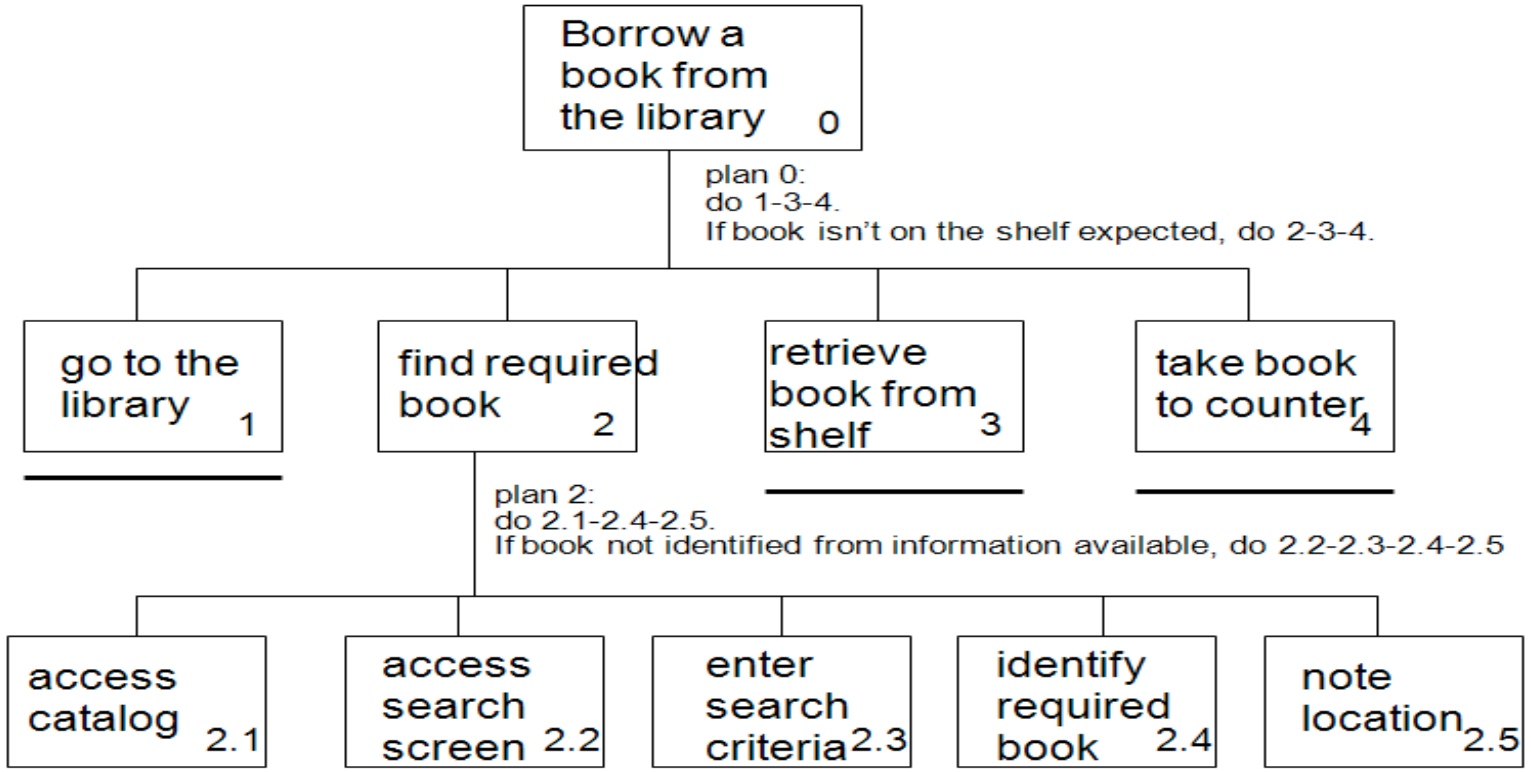
- Learning about their problem
  - Semi-structured interview
- Analyzing their tasks
  - Hierarchical task analysis
- Modeling users
  - Personas

Hierarchical task analysis

- **Why?**
  - Understand user workflow
  - Identify pain points and areas for optimization
- **How?**
  - Decompose tasks into 4-8 sequential steps
  - Identify patterns, sequences and skips in the tasks
  - An example:

Hackos, J. & Redish, J. (1998). User and Task Analysis for *Interface* Design. Chichester: Wiley.

Task analysis example



http://i1206241.blogspot.com/2013/01/task-descriptions-hierarchical-task.html

Defining your audience

- Learning about your users
  - Semi-structured interview
  - Contextual inquiry
- Analyzing users' tasks
  - Hierarchical task analysis
- Modeling users
  - Personas

Personas

- **Why**?
  - mechanism for reasoning about user needs
  - model behavioral characteristics of target users
  - doesn't require access to ACTUAL users
- **How?**
  - fictionalization
  - narrative, goals, needs, "pain points"
  - attributes specific to the problem space
  - data-driven method* using info from interviews
  - mapping persona to software features

Personas

Example: Persona for a user of the advising app

- Dr. Betty is a visiting lecturer for WSU. She has never worked for this university before but has worked at others. She was assigned 35 advisees, and half of them are juniors and seniors.
- Dr. Betty is familiar with technology and has no problem using software to help with advising (in fact, she likes it). However, because she has never been at Westfield before, she does not know which classes are offered yearly, every other year, every third year, etc.. Because she is advising juniors and seniors, it is crucial that she has access to this information.
- In addition, Dr. Betty has limited time. She needs to be able to quickly see which of her advisees are on track and which require some extra attention when it comes to selecting classes.

Activity: personas

Goal: come up with **3 personas** that characterize users of a a public transit app be specific

Now that we've got some end users in mind, what would a **prototype** look like?