**CS 361 – Module 2 Notes – Prototyping and Usability**

* there are multiple levels (**fidelities**)of UI design prototypes, **low, medium, and high-fidelity**
  + **low-fidelity**: A rough sketch that is often drawn by hand, drawn using an app and stylus, or made with low-fidelity creating software. You can gather feedback on high-level features and have the flexibility to make large, low-cost changes
  + **medium-fidelity**: Detailed illustration often created using a professional drawing or presentation tool (Visio, PowerPoint, etc.) or a detailed hand-drawing. Although, digital is more common. You can gather feedback on changes defined and accepted features that you plan to keep to keep costs low at this level.
  + **high-fidelity**: Polished and detailed illustration that looks like a finished UI. Can be created in a full-featured graphics editor (Photoshop or the like) or a GUI builder. To keep costs low at this level, you can gather feedback about detailed tweaks to specific features to make very focused and incremental improvements
* a quick and low way to begin prototyping and to begin getting feedback on your UI is to create a **paper prototype**
  + **paper prototype** – A hand drawn sketch used to communicate a potential UI design to be implemented, especially a GUI design. These are generally simple and not very detailed but show an overall introductory idea to the UI design.
* Some paper prototypes can even show **interaction design**
  + **interaction design** – indicates what users can interact with, how they can interact and what happens when they do. (i.e. user clicks and drags a slider up and down, the page moves with it. User clicks a down arrow and a drop down menu appears)
* **Cognitive Style Heuristics (CSH)** are nine principles of **interaction design** used to improve software usability. They are based around different cognitive styles different people use when they problem-solve in software, how different people use software in different ways. CSH were created with new users in mind: people who have never seen, interacted with, or received previous direction on the software. But they can also improve usability for existing/experienced users.
* Below are the five **cognitive style facets**:
  + **Motivations –** Why someone is using the software (task completion vs. interest)
  + **Information processing style –** How a person looks through or absorbs information in software (comprehensively vs. selectively). A person processing comprehensively may want to understand details, implications, or to get a sense of overall structure before taking action in software. A person processing selectively may taking action as soon as they detect what seems like the beginning of a promising path.
  + **Computer self-efficacy** – A person’s confidence in their ability to use computers or software (low vs. high). A person with low efficacy may think it’s their fault when an error occurs in the software. A person with high efficacy may think the software is poorly made/not their fault when an error occurs.
  + **Attitude toward risk** – How willing a person is to take chances in software (risk-tolerant vs. risk-averse). Someone risk-averse may avoid taking actions that have unknown consequences or seem dangerous or irreversible. A person who is feeling risk-tolerant may take actions even if they know those actions could lead to bad consequences.
  + **Learning Style** – How a person prefers to move through software (tinkering vs. process).
    - Each facet has two polar **cognitive style facet values** (the text within parentheses)
* **Cognitive Style Personas** are a group of **personas** representing the users within a target audience, they’re generally named Abi, Pat, and Tim
  + There are only three
  + Each have different sets of cognitive styles, yet they are fixed
  + Each persona represents multi-personas (Ex. on pg. 4 of Sprint 2 – CSH Reading)
  + Abi and Tim are at opposite ends of the spectrum as far as usability/cognitive styles go, Pat is in the middle.
* Heuristics 1 – 9 start on pg. 5
* **Heuristic #1 –** Explain what *new* features do and why they are useful
  + Examples: Each featured extension has a brief description that says what the extension does and why somebody would use it.
    - A new feature is listed and how to use it
* **Heuristic #2 –** Explain what *existing* features do and why they are useful
  + Examples: A tooltip explains what the search is for and why someone might use it
    - each tooltip tile explains a feature and the benefit of using that feature
* **Heuristic #3 -** Let people gather as much information as they want, and no more than they want
  + Allow someone to find the info they want, but don’t force them to spend excessive time or effort to gather that info
  + Examples: Users can choose to view code documentation while still viewing their code
    - Users can quickly see the contents of the webpage and jump to the section they’re interested in
* **Heuristic #4 –** Keep familiar features available
  + Allow users to use the software in ways they’d expect. Is especially beneficial for those with low computer self-efficacy
  + Examples: Although the “following” page is gone, the new update looks similar to the previous version so that users are still familiar with the app
    - The smartphone and tablet versions of an app offer the same features which makes switching between the two very easy
* **Heuristic #5 –** Make undo/redo and backtracking available
  + This allows both novice and experienced users a familiar and easy way to recover from mistakes
  + Examples: Browser back/forward buttons allow users to backtrack through their browsing history
    - An undo button allows users to recover from mistakes. Version control systems also help with this and revert from any code submitted that may have been a mistake or had an error.
* **Heuristic #6 –** Provide ways to try out different approaches
  + Allows users to approach something in a different way if they feel like they’re stuck
  + Examples: If users don’t find what they need on a “Choose a Question” drop-down menu, they can try the chat.
    - If users encounter a problem using the SecureChat UI, they can attempt the same operations using the command line interface
* **Heuristic #7 –** Communicate the amount of effort that will be required to use a feature
  + Allows a user to determine if a particular feature will require too much effort to use
  + Examples: Placing “Advanced Options” at the bottom of the menu indicates to the user that “advanced” features may take more effort/knowledge
    - The dialog indicates that “cor launcher” will be needed with “associate files with Coral” and that the user will need write permissions for the installation folder
* **Heuristic #8 –** Provide a path through the task
  + Allow users to go through tasks with a clear process while also allowing them to choose to not use step-by-step/tutorials
  + Examples: Users can choose their entry point, and each path is explained
    - Users get to choose either the path of learning more about the new feature or going back to what they were doing
* **Heuristic #9 –** Encourage mindful tinkering
  + Encourage users to tinker in ways that lead to them discovering task-relevant functionality
  + Examples: This design encourages users to tinker mindfully by showing they will notify them before impactful actions are executed, like deleting an important file or emailing 237 people at once.
    - Keybase encourages uses to try out new “slash” commands by showing all the commands when a user types “/” and explaining what each does and how to use it
* CSH’s are meant to be used in a **heuristic evaluation**
  + **heuristic evaluation** - A usability inspection method where evaluators independently check that a design reflects a set of heuristics, then compare results to determine if the specific design reflects the heuristic
  + **GenderMag Method** is a process for finding and fixing gender-inclusivity bugs in software. This is used to make sure software isn’t necessarily geared towards one gender or another, if the goal is to be as neutral as possible.