Introduction to UX Principles and Processes

What is UX? What are UX Research and Design?

- How to make UX easy
  - Follow an iterative prototyping process
  - Apply user-centered research and design methods
  - Understand a bit about human behaviour
  - Apply common sense
- **UX** = the experience people have when they interact with your product
  - Using the product
  - Choosing the product
  - Acquiring the product
  - Learning to use the product
  - Fixing the product
  - Upgrading the product
  - …
- Why does UX matter?
  - Experience success = intersection between *What users want to do* and *What you want users to do*
    - Meet product goals
    - Repeat usage
    - Recommend to others
- Why is UX hard?
  - You are NOT the user
  - Computers are weird
  - Software is (usually) complex
- Understand how people work
  - What can people perceive - How do people extract information from visual stimuli?
  - How do people do things – How do people decide how to act in the world, and how do they process information about the results of their actions?
  - How does emotion play a role – How, when, and why does emotion affect decision-making, and what role does emotion play in user experience?
- Interactive Design = Assess → Design → Build → *Repeat*
- Key methods of **UX Research**
  - Interviews
  - Observations
  - Surveys
  - User Testing
  - Inspection Method
- Key methods of **UX Design**
  - Personas, Scenarios, User Stories
  - Sketching and Ideation
  - Storyboarding
  - Mapping and Navigation Design

    - Comparative Research
    - Lo-, Mid-, and Hi-Fidelity Prototyping
- Fail Fast
    - You won't get it right
    - Get it wrong as quickly and as often as possible
    - Learn from mistakes
    - Get it less wrong each time
- **Components of UX**

	Understanding Users	Evaluating Designs
Value	What do users need?	Does this design fulfill the need?
Usability	How do they do it now?	Can they get it done with this?
Desirability	What do they desire?	Is the design appealing?
Adoptability	Where do users look for things?	Can users find and access this?

UX Design Overview

- What is special about UX Design?
    - Experiences are interactive
        - Time-based
        - Action-response rules
        - Action: command option
        - Response: information presentation
        - Complex system behaviour (focus on usability)
    - Context is critical
        - Other interactions
        - Other activities
        - Other people
- Design process
    - Understand the problem – Study users: tasks and context
    - Generate possible solutions – Sketch, storyboard, wireframe…
    - Analyze and select – Apply UX criteria
    - Embody solutions – Build prototype
    - Assess (find new problems) – Apply UX research methods
    - *Repeat*
- Design = a plan for arranging elements for a purpose
    - Problem solving (as much finding the problem as finding the solution)
- Iteration prevents premature commitment
    - Problems with premature commitment
        - Waste money and effort

        - Exhaust project resources/timeline, stuck with a bad design
        - Cognitive and emotional commitments hard to undo
- Fidelity of prototypes **increases** with time
- "The best prototype is the one that, in the simplest and most efficient way, makes the possibilities of a design idea visible and measurable."
- **Lo-fi prototypes**
    - Address
        - Functionality
        - Basic organization
        - Task flow and coverage
    - Ignore
        - Graphics
        - Programming
        - Real data
- **Mid-fi prototypes**
    - Address (*lo-fi concerns plus…)*
        - Layout
        - Interactivity
        - Navigation
    - Ignore
        - Graphics
        - Programming
        - Real data
- **Hi-fi prototypes**
    - Address (*mid-fi concerns plus…)*
        - Graphic design
        - Interaction details
        - Realistic data
    - Ignore
        - Backend programming
        - Complete functional coverage
- How to sketch? (no need to be good at drawing; quantity > quality)
    - Use pencil and paper (whiteboard OK)
    - Go fast
    - Don't perfect
    - Make lots of them
- Purpose of sketching
    - Reflect
    - Explore
    - Communicate
- What to sketch
    - **Problem** – How would someone experience this problem?
    - **Solution**
        - What would it look like for the problem to be solved?
        - How would a system help solve the problem?

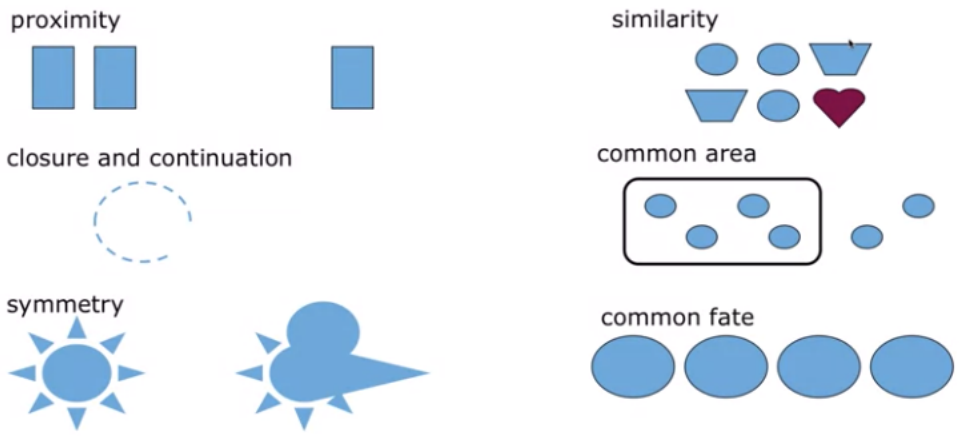
- Sketches vs prototypes

Sketch	Prototype
Evocative	Didactic
Suggest	Describe
Explore	Refine
Question	Answer
Propose	Test
Provoke	Resolve
Tentative	Specific
Noncommittal	Depiction

- Qualities of a sketch
    - Quick
    - Timely
    - Inexpensive
    - Disposable
    - Plentiful
    - Minimal detail
    - Allow ambiguity

How do People Perceive Information?

- People don't read everything and don't read everything equally. Their reading patterns tend to follow an F pattern.
- Basic principles
    - Make important info & actions visible
        - Essential info can be *below the fold* and invisible to users
    - Leverage "the read"
    - When evaluating, ask yourself "did they see it?"
- Feature detection (
    - Very fast
    - Allows "pop-out
    - Supports subsequent stages
        - Patterns
        - Objects
    - Lists of features from fastest to slowest
        - Color
        - Values (shade)
        - Angles
        - Slope
        - Length
        - Texture
        - Motion
- Pattern Identification: Gestalt Principles



    - When two shapes are not symmetric, we see them as different shapes
- Gestalt principles
    - Use "pop out" (primitive feature) to attract attention
    - Use Gestalt principles to associate like items
    - Use Gestalt principles to organize for skippability
- Short-term memory
    - Limited capacity
        - The *magic number* = 7 +/- 2 items
        - Maybe more like 4 +/- 1
    - Information that is not retained is lost
        - *Retained* means *committed to long term memory*
        - Also known as learning
    - Principles
        - Keep lists of options short
        - Give users tools for reducing options
        - Don't expect users to remember stuff
- Memory process = Sensory register → Perception → Short-term memory → Long-term memory
    - From *Sensory register* to *Perception*
        - Requires attention
        - Small amount of what is available is actually perceived
- Long-term memory principles
    - Learning will work better if learner can fit into a schema
        - Use metaphors (e.g., shopping cart)
        - Leverage standards and consistency (e.g., google drive files work the same as themselves and the competition)
        - Avoid asking users to memorize stuff
    - Prefer recognition over recall
- Likelihood of remembering
    - Strength of association
    - Recency
    - Frequency

How do People Act in the World?

- Seven stages of action
  - Forming the goal
  - Forming the intention
  - Selecting the action
  - Executing the action
  - Perceiving the state of the World
  - Interpreting the state of the World
  - Evaluating the outcome
  - *Repeat*
- Discoverability = how we bridge the gulfs of execution and evaluation
  - Affordances = feature of an object or environment that indicates the possibility of action (e.g., button raised means pressable)
  - Signifier = indication of *what* action will occur and *where* the action can occur (e.g., STOP label on button)
  - Constraints = unavailable actions should be disabled and we should limit the total number of options to make selection easier (e.g., start button not pressable when machine is working)
  - Feedback = users need to know that the system received their input and what it did with their input
- Conceptual Models
  - Support *simulation* of future actions
  - Appropriate use of affordances, signifiers, feedback and constraints leads to **formation of accurate conceptual models**
- Bridging the gulfs (**execution** and **evaluation**; something preventing the user from achieving their goal or evaluating their success)
  - Understand
    - Users' goals
    - How they think about accomplishing them
  - Make sure likely actions are
    - Visible when needed
    - Make sense
  - Make sure the results of actions
    - Are visible
    - Make sense
- Example of bridging a gulf of execution
  - Gulf = user does not know whether to push or pull a door and where to perform this action
  - Bridge = installing a universal push bar
- Example of bridging a gulf of evaluation
  - Gulf = user does not know whether the pedestrian button worked
  - Bridge = light turns on when button is pressed

Design Heuristics

- Jakob Nielsen's 10 heuristics
  - Heuristic = rule of thumb (slightly more general than a guideline)
  - Derived from a systematic review of usability problems
  - Intended to be a small, complete, and usable set
  - Able to be taught in a few hours (you get better with practice)
  - Well-supported by theories of perception and cognition
- From knowledge to guidelines
  - Important to understand
    - How people perceive
    - How people remember
    - How people act to pursue goals
  - Choosing guidelines
    - Are they well-supported and focused on user experience?
    - Do they cover all the important best practices?
    - Do they apply to your platform/situation?
    - Are they easy to use?
- Heuristic #1: **Visibility of system status**
  - The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time
  - Manifests through
    - Feedback
    - Available actions
    - General availability (loading wheel)
      - Strive for <100 msec response time
      - Up to 1 second, no indicator needed
      - 1 to 10 seconds, use wait cursor
      - Over 10 seconds, complete in background, use progress indicators and estimates
- Heuristic #2: **Match between system and real world**
  - The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order
  - Manifests through
    - Language (don't use error codes)
    - Order of operations
    - Metaphor
- Heuristic #3: **User control and freedom**
  - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
  - Manifests through
    - Emergency exits
    - Undo

        - Redo
- Heuristic #4: **Consistency and standards**
    - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
    - Manifests through
        - Consistency of language
        - Layout
        - Behaviour
        - Consistency across products (google docs & sheets) and competitors (word and excel)
- Heuristic #5: **Error prevention**
    - Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone condition or check for them and present users with a confirmation option before they commit to the action.
    - Manifests through
        - Provide constraints (display date format in input label or break up in 3 fields)
        - Confirmation of risky actions
        - Prevent actions that are likely to fail
- Heuristic #6: **Recognition rather than recall**
    - Minimize the users' memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
    - Manifests through
        - Direct manipulation (display all objects/actions instead of requiring the user to remember a command)
- Heuristic #7: **Flexibility and efficiency of use**
    - Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
    - Manifests through
        - Keyboard shortcuts
- Heuristic #8: **Aesthetic and minimalist design**
    - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility
        - Use Gestalt Principles for non-linear reading
- Heuristic #9: **Help users recognize, diagnose, and recover from errors**
    - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution
- Heuristic #10: **Help and documentation**
    - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be

easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
    - Best if help is not needed
    - If required, make sure help is
        - Searchable
        - Task-focused
        - Concrete

Heuristic Evaluation

- Using multiple methods
    - Perform **User Testing** first to make sure you're on the right track
    - **Heuristic Evaluation** to identify possible issues
    - Perform **User Testing** again to test the final product
- Heuristic evaluation
    - Discount UX Research method
        - Cheap
        - Fast
        - No users
    - An inspection method
        - Systematic close read of a user interface
        - Apply heuristics to find and explain problems
    - Multiple evaluators better (sweet spot = 3-5)
        - Only one finds 35% of "true problems"
- Process
    - Choose a set of screens/interactions for focus
    - Step through applying heuristics to find potential problems
        - Be sure to test error cases
        - Be sure to look at help system (if any)
    - Write down all violations, big and small
        - And the heuristic they violated
    - Assess the severity of each problem
        - 1 = cosmetic problem; no real usability impact
        - 2 = minor usability problem; fix if there is time
        - 3 = major usability problem; important to fix
        - 4 = usability catastrophe; imperative to fix
    - Create a prioritized list of problems to fix
        - Highlight top 5-10 problems
        - Ranked in decreasing order of severity
        - Use heuristics to explain why they matter
- Heuristic evaluation vs User Testing
    - Heuristic evaluation
        - Cheap
        - Fast
        - Don't use up potential users

- User Testing
  - More realistic
  - Find more problems
  - Assess other UX qualities beyond usability