* Design is human power to conceive, plan and make products and services that will help us reach individual and collective purposes
* What tasks are performed? How are tasks performed? How often? Additional resources?
* Cybersecurity and legal constraints: password length to protect privacy
* Storyboard: imitate sequence of events
* Ethics: IRB approval, avoid distressing without debriefing
* Usability engineering
  + Slips: right goal, wrong action
    - Capture (frequent action takes over), associative (phone rings, say come in)
  + Mistakes: wrong action
  + Mode error: same input, different output due to mode
  + Heuristic evaluation: comparing prototypes against set list of heuristics (visibility of system status, match between system and real world, recognition over recall, consistency and standards)
    - Pros: cheap and fast
    - Cons: needs trained people, not one correct heuristic
    - Mode error:
      * caps
      * 33 instead of 3.3
* Wireframes: basic skeleton of layout: validate information architecture and general user flow early on
* Design specs: documentation of design
  + (work) Designs user experience, details user interface, offers design rationale
  + (build) Not technical specification: implementation methods, languages and tools
  + (money) Not business plan: business goals and organization information
* Diary studies/experience sampling: study over time
  + Interval (regular time), signal (promopted), event (specific), retrospective (after)
  + Pros: no Hawthorne effect, comprehensive, in depth
  + Con: low compliance, reliance on recall
* A/B Testing: comparing versions of a web page to see which one does better
  + Pros: easy to compare options
  + Cons: don’t know why, need multiple users
* Parallel vs. serial
  + Parallel: better comparison, more exploration, more feature sharing, increase in group rapport
* Design patterns: good solutions to common problems
  + Use because conventional, tested, raise level of abstraction
  + Don’t use because may not be the best, stifles
  + Navigation bar, search bar, shopping cart, account, grid layout, hamburger menu, logo to go home
* Document Object Model: how programs access and modify HTML > tree
* ID vs. class
  + Id: unique, weighted heavier
  + Class: can be applied to multiple elements
* Libraries: set of code previously written > convenient, works
* Object based organization: every element on screen is an object; input/output/state; own behaviors and state: can draw itself, can handle input, can change status
* MVC: separates content (view) from data (model) and logic Controller
  + Separation of concern, modularity, loose coupling
* Event queue: all generated events go to one queue
  + To ensure temporal ordering of events
  + Can group events together (combine all mouse motions)
  + Loose coupling, so don’t have to know what inputs
  + LAMP: layered architecture, with different responsibility at each layer
    - Linux (operating system) , Apache (server), MySQL (database) , PHP (templating)
* Fovea: indent in the retina, center of field of vision highest concentration of cones
* Fitt’s law: things that are closer together or bigger are easier to hit, and vice cersa
  + Subtle source of usability bugs
  + Improve usability by increasing target size
  + Way of improving performance
  + Rods: gray, mreo, edges
* Mapping: relationship between controls and the effects on world
* Feedback: response by system to user action; important for cause and effect, important for forming mental models
* Visual grouping: proximity, similarity, connected
* Design, system, user
* Bringing things closer (pie menu), avoiding pointing tasks
* Shapewriter: gestures + dictionary