

## Interface Prototyping

- Why prototype?
  - Stakeholders can see, hold, interact with a prototype more easily than a drawing
  - Team members communicate effectively, can test ideas
  - Prototypes answer questions, support decisions etc
  - Fast, cheap feedback, enables experimentation
  - Easier to change/throw away when we discover problems
- **Fidelity**
  - Breadth: % of features covered - only enough to test certain tasks
  - Depth: degree of functionality for each - limited choices, no error handling
  - **Low-fidelity prototypes**
    - Medium used is not like final medium (e.g. paper/cardboard) - quick and cheap
    - E.g. sketches of screens, task sequences, 'Post-it' notes, storyboards, 'Wizard of Oz'
    - Paper prototypes: interactive mockup, interaction is natural (point with finger)
      - Person simulates the computer operation
      - Why? fast, easier to change, focuses on big picture, non-programmers can help
      - **Team fills roles: Computer** (provides system feedback), **Facilitator** (presents interface), **Observer** (takes notes)
      - **Paper prototypes show:** conceptual model, functionality, navigation and task flow, screen contents
  - **Storyboards**
    - Used with scenarios, series of sketches showing progression through a task - used in early design
- **High-fidelity prototyping**
  - Use materials that would be expected in finished product, resembles the desired output
  - Evolutionary prototyping vs Throw-away prototyping
  - Risk involved - users may think it is the final system
- **What is a prototype?**
  - Can be: sketches, storyboard, slideshow, video simulation, physical rendering (e.g. wood), cardboard mock up, software with limited functionality
  - **What to prototype?**
    - Technical issues, screen layouts, information display, **critical areas**
- **Conceptual design: from requirements to design (first step)**
  - Transform requirements/user needs into a conceptual model
  - Iterate, Iterate, Iterate - consider alternatives, experiment
  - Interface metaphor and interaction style - Xerox star gave GUI that most OS use e.g. files and folders, dragging equivalent to physical relocation
    - Logical consistency - can expand metaphor logic in a natural way e.g. dragging onto printer to print - not relating to physical but makes sense in comparison to file relocation

- **Expanding the conceptual model**
  - What functions will it perform - what is the systems job and what is the person's job
  - How are functions related to each other
  - What information needs to be available
- **Scenarios in conceptual design**
  - Express proposed situations, use throughout design in various ways
- **Wizard-of-Oz prototyping**
  - The user thinks they are interacting with the system, but actually developer is responding
  - Used to understand user expectation
  - Used to simulate future technology: speech recognition, learning, context aware behaviour
- **Participatory Design**
  - User is a member of the design team
  - Collaborative and iterative design
  - Using brain-storming, storyboards, workshops