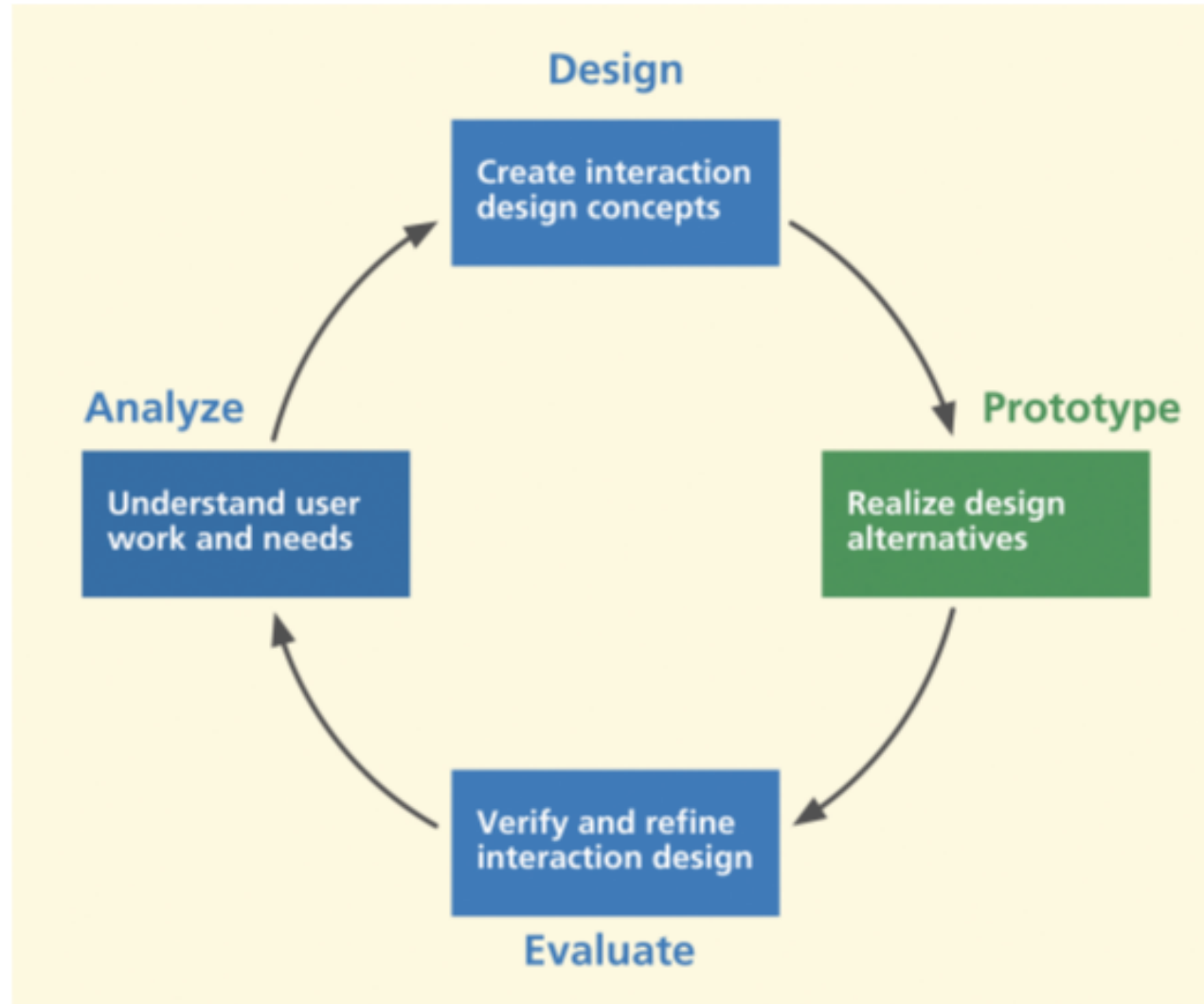


# Prototyping

SWEN-444

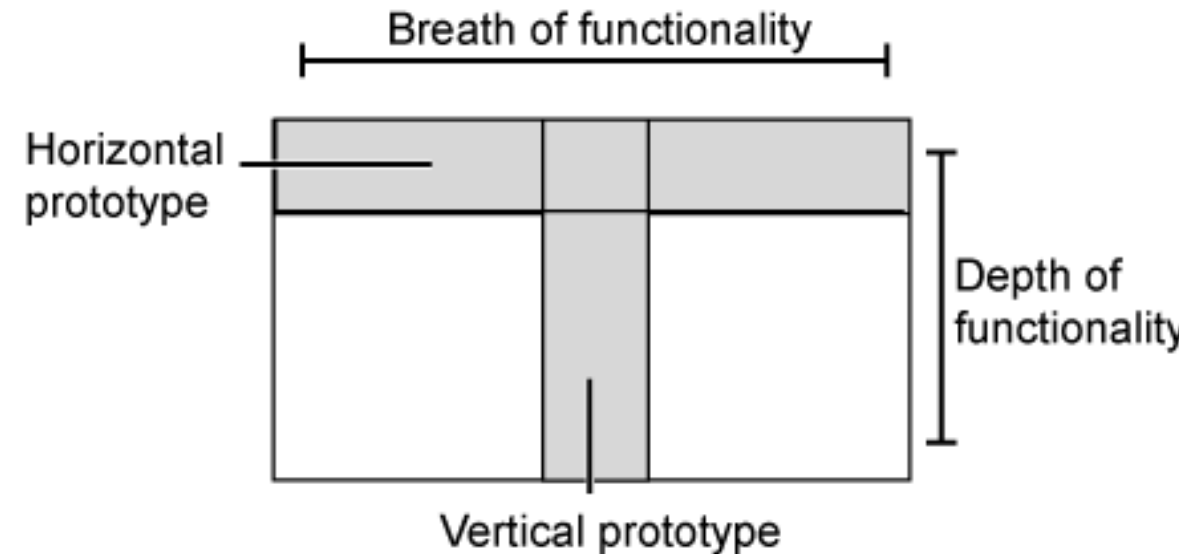
Selected material from *The UX Book*, Hartson & Pyla

Evaluate design before it's too late and too expensive.



# Prototypes must be less than the full system.

- **Horizontal** – overview of feature coverage, but more abstract so less effective evaluation
- **Vertical** – more depth for a few features with more effective evaluation
- **“T”** – the “middle way”, mostly horizontal with a few vertical features
- **Local** – where horizontal and vertical slices intersect; analyze isolated concern, e.g., icon design



# Dimensions of Fidelity of Prototypes

- **Breadth** - % of features covered
- **Depth** – degree of functionality
- **Look:** appearance, graphical design
  - Sketchy, hand-drawn
- **Feel:** input method, degree of interaction
  - Pointing and writing feels very different from mouse and keyboard

# Fidelity of Prototypes

- **Low fidelity**
  - **Paper** sketches, story boards or simple wireframes
  - Low fidelity in look and feel, more abstract
  - Can be effective in user evaluation
- **Why paper?**
  - Easy, fast, and low cost to create and change
  - Creative focus on design not on the drawing or programming tool
  - Designer in control during user evaluations



# Fidelity of Prototypes

- High fidelity
  - Include details of appearance and interaction behavior
  - Users see more complete design
  - Advanced wireframes with navigation (medium fidelity)
  - Programmed without being the final product
    - Backend simulation
    - Storyboard animations

# What Type of Prototyping?

- **Progress** prototype **fidelity** during the design life cycle to...
- Understand the **ecological** (low), **interactive**, and **emotional perspectives** (high) and to ....
- **Focus on behavior first** then appearance

Ideation	Low fidelity paper sketches
Conceptual design	Low fidelity paper sketches, storyboards
Intermediate design	Low to medium fidelity wireframes
Detailed design	High fidelity wireframes, programmed prototypes

# Cautions

- Rationalize cost-value tradeoffs to gain budget support
- Do not **oversell** - capabilities that can't be delivered, development completeness
- Do not **overbuild** – “good enough” as a prototype
- Decide early on exploratory or evolutionary prototypes