

# Prototyping

Some content from: Camburn, Bradley, et al. "Design Prototyping Methods: State of the Art in Strategies, Techniques, and Guidelines." *Design Science*, vol. 3, 2017, <https://doi.org/10.1017/dsj.2017.10>.

# Each Prototype Focuses on a Question

- "Acquire sufficient information to move forward in product development with minimal expenditure of time and cost. Therefore, every prototype test should answer a specific question (Otto & Wood 2001)."
- Web app or native app?
- Accounts, or anonymous access?
- Free-form text, or fields?
- How "live" does interaction need to be?

# General Findings on Prototyping

Variable	Design Heuristic
Testing	Construct a clear testing objective
Timing	Early prototyping is the most critical
Ideation	Prototypes lead to functional ideas
Fixation	Fast prototyping reduces fixation
Feedback	Feedback may induce corrections but also increase fixation
Usability	End-user testing may enhance performance assessment accuracy
Fidelity	Higher fidelity representations lead to accurate interpretation of the design

# Timing

- Earlier is better (first 30% of project by time)
- Late prototyping is correlated with unsuccessful efforts
- "late adoption of prototyping was an especially clear hallmark of unsuccessful design teams." (Jang and Shunn)

# Ideation

- Frequent early prototyping leads to new design ideas
- As in the Klemmer video: don't just prototype once
- In chess, it's hard to think many moves ahead
  - Instead, actually play the game, starting several different ways

# Fixation

- We tend to get "stuck" (fixate) on designs that seem compelling
- But are those designs really the best?
- "a slow fabrication process may induce fixation, but a rapid one will not, as compared to sketching only (Viswanathan & Linsey 2013)."
- Implementing your idea is usually slow

# Feedback

- Reluctant to show managers? Show peers.
- Managers often expect high-fidelity prototypes
  - Show peers your low-fidelity prototypes
- Receiving feedback on a design may increase fixation
  - To mitigate, present *multiple* designs

# Usability

- Interactivity level is a key fidelity parameter
- High interactivity levels can enable usability testing

# Fidelity

- Higher-fidelity representations lead to more accurate interpretations by reviewers
- But more rapid prototyping reduces fixation
- Select fidelity level wisely

# Prototyping Techniques

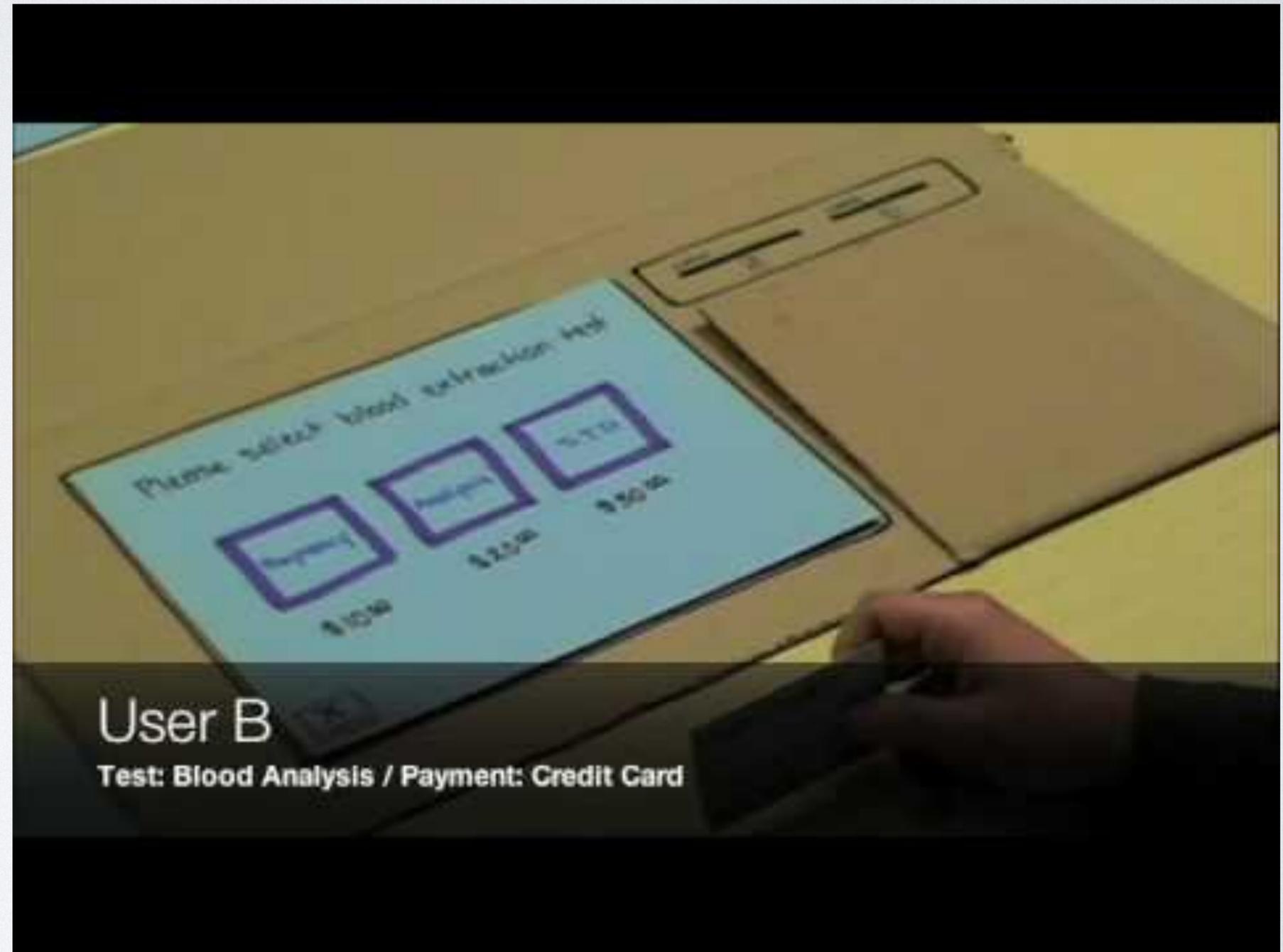
- Sketches
- Paper or LEGO prototypes
- Wireframes
- Wizard of Oz
- Full-Resolution Mockups, Storyboards

# Sketches

- Very informal, whiteboard-style diagrams

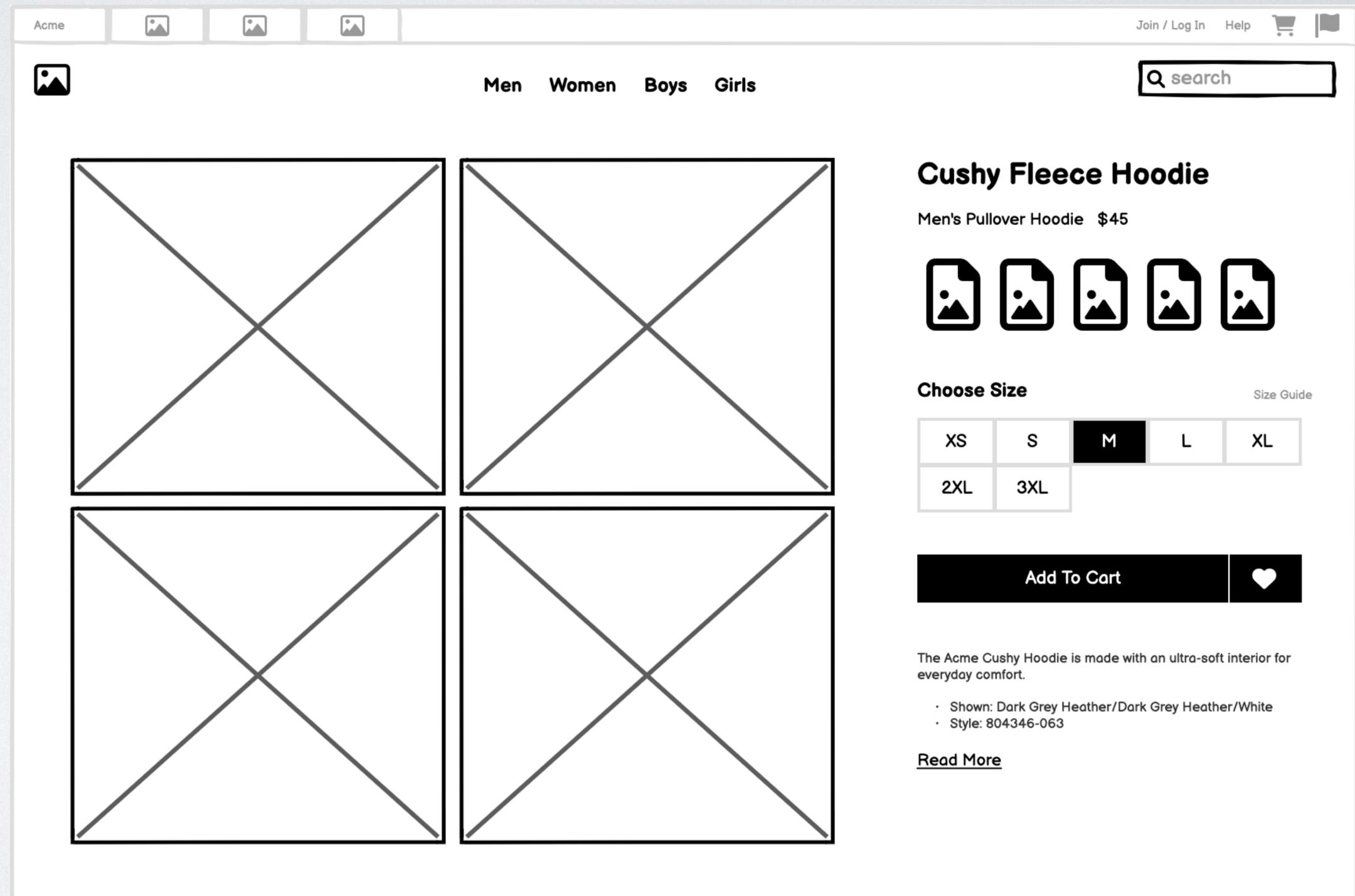
# Paper Prototypes

- Neater drawings of various interfaces
- Enable showing the interface to potential users
- Ask users to tell you what they would do



# Wireframes

- Balsamiq (<https://balsamiq.com/>)



# Wizard of Oz

- Goal: evaluate usability of a very sophisticated system
- Problem: implementing it is expensive
- Solution: Someone remote-controls a dumb prototype

# Full-Resolution Mockups

- Sometimes expensive to create
- But these avoid people getting distracted by nonessential aspects of prototypes

# Activity

- Sketch the home page of a canoe rental website. It should allow users to select a date and time to rent a canoe, and it should look enticing so that users will feel excited to go canoeing. You can do this on paper or in a drawing app.
- Upload a photo or screenshot of your design to Gradescope.

# Specifying Requirements

# Requirements, User Stories

- Question: how to express requirements?
- Answer: "As a <stakeholder>, I want <something> so that <need>."
- Example: "As a student, I want to filter recipes by cost so I can keep dinner under \$5 per person."

# User Story Criteria: "INVEST"

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

# Independent

- Ideally: want to implement requirements in any order
- In practice, there may be dependencies

# Negotiable

- Details to be negotiated during development
- Good Story captures the essence, not the details

# Valuable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues

# Estimable

- Helps keep the size small
- Need to complete each user story in 1-2 weeks (or less)

# Small

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate
- Too big == may not finish in time for delivery

# Testable

- Ensures clarity
- If not testable, when do we say the task is done?

# Summary

- Write open-ended, high-quality questions to elicit requirements
- Use INVEST criteria to write good user stories