

Designing the Ribbon

- <http://channel9.msdn.com/Events/MIX/MIX08/UX09>
- Lots of info on the design process that resulted in the ribbon interface for word
- Very relevant to lecture on prototypes and testing.

Prototyping

Rationale

- Prototyping is designed to get you to the best UI for your application
- Assume you know what your application should do, it's purpose
 - How should interaction unfold before the user?
 - How can we ensure user can accomplish necessary tasks?
 - How do we map UIs to these processes?

Prototyping Process

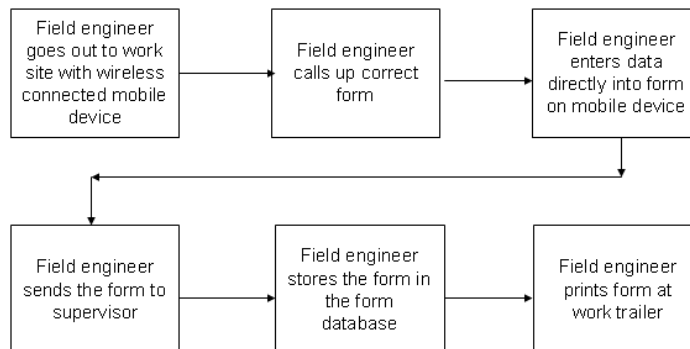
- Figure out exactly what the user needs to do
 - Scenarios
 - Storyboards
- Then prototype, i.e. design screens to satisfy users functionality
 - Brainstorm many alternative designs
 - Cherry pick ideas
- THEN evaluate designs with user

Take Aways

- Scenarios and storyboarding
- Low-Fidelity and wizard-of-oz prototyping
- Application design guidelines and their role in prototyping
- Evaluating prototypes

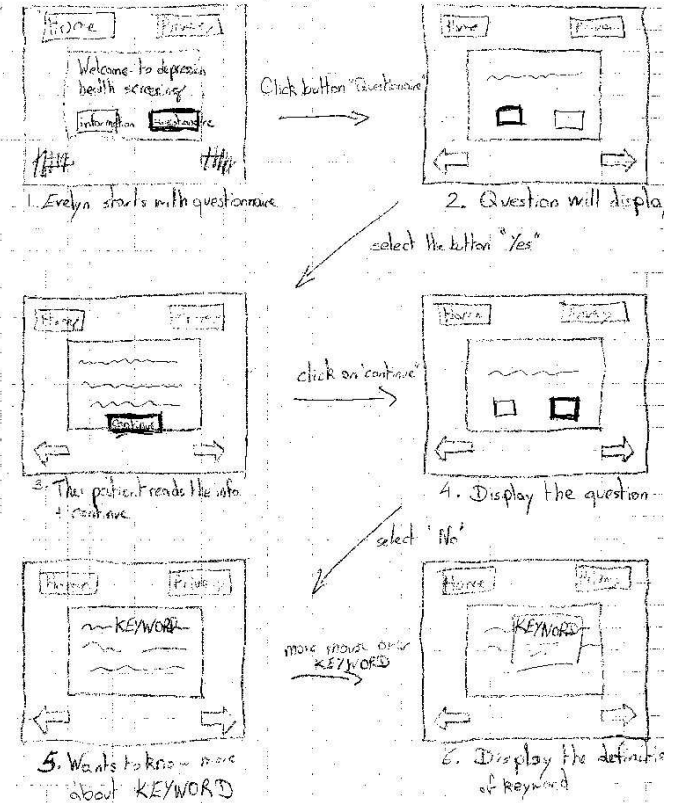
Storyboards

- Makes scenario real
- Shows people, words, screenshots, whatever is appropriate



40

3. Alternate question/info



Prototyping Dimensions

- Horizontal versus Vertical
 - Broad perspective on system functionality
 - Deeply exploring one aspect of system behaviour
- High fidelity versus Low fidelity
 - Lowest fidelity is paper/cardboard/crafts supplies
 - Higher fidelity includes Wizard of Oz prototypes
 - Wozzing

Wozzing

- Can fake interaction using pre-canned input and output
 - You have all done this
- Can also build elaborate applications
 - Sketchwizard is for pen-based UIs

Sketchwizard

SketchWizard: Wizard of Oz Prototyping of Pen-based User Interfaces

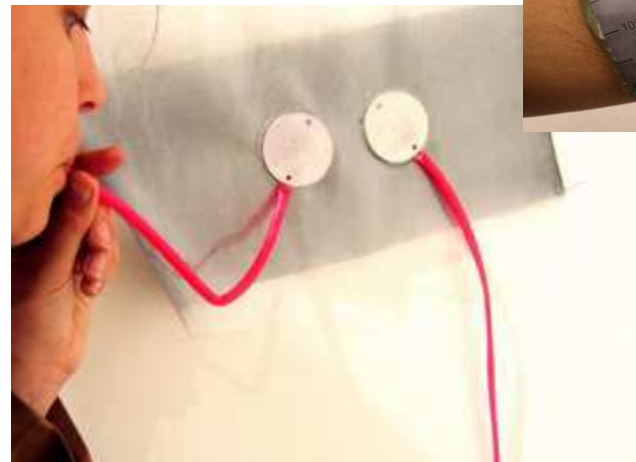
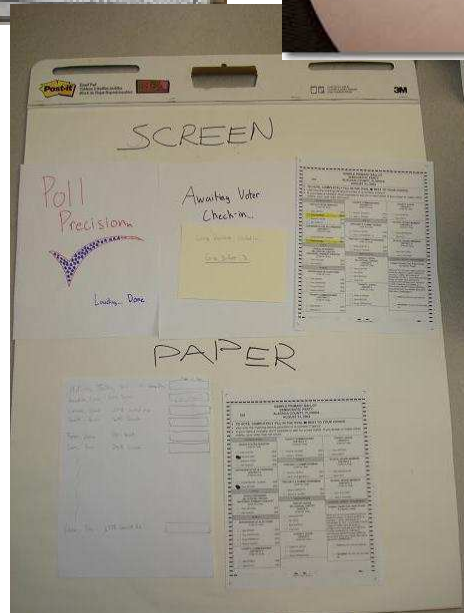
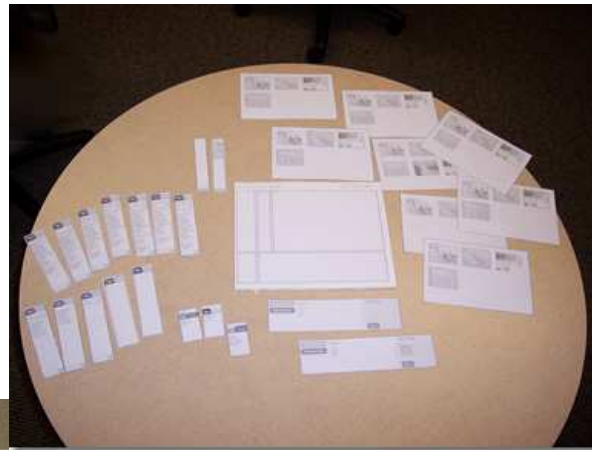
Richard C. Davis
U.C. Berkeley

T. Scott Saponas
U. of Washington

Michael Shilman
ChatterPop, Inc.

James A Landay
U. of Washington
Intel Research Seattle

Low-fidelity prototypes



Do Low Fidelity Prototypes Make Sense?

- See Erick Schonfeld's article "Will you try my iphone app?"
 - Approached at CVS in Palo Alto
 - Asked by a Stanford student in Human-Computer Interaction course to try iPhone app
 - The app was a paper prototype

What Erick Schonfeld said:

- “... you might want to wait until you have an actual working app on an iPhone before testing it out in the wild and asking for feedback ...”
- “... I blame his professor for sending him on this hapless mission.”
- “There is really no way to test an iPhone app on paper, <sic> the buttons don’t do anything.”
- “The best part: the course is called ‘Introduction to Human-Computer Interaction Design’.”

Just a Note

- A blogger is a blogger because he writes well
 - Not because he knows anything
- Further
 - Low-fidelity prototyping is not unique to HCI
 - Wireframes in software engineering, web design
 - Wireframes Magazine
 - Sketches in architecture

Why use low-fidelity prototyping?

1. Easier implementation
2. Faster iteration
3. More variety
4. Quality of feedback

Creating low-fidelity prototypes

- Materials
 - Paper (various sizes)
 - Scissors, glue, tape, X-acto knives
 - Markers (various colors)
 - Overhead transparencies
- Build fast
 - Draw ideas quickly (don't worry about neatness)
 - Start with window
 - Use smaller papers for things that change
 - Have menus ready
 - Think about interaction – anything customer might want to see
- Remember to think about device differences!

Faster iteration

- High fidelity prototypes require careful implementation
 - Get caught up in details
 - Lose sight of big picture
 - Design – prototype – evaluate – iterate
- Require only kindergarten level skills
 - User can participate in design process as well
 - ... Or can alter design
 - ... Or can design on their own as well
 - Implies buy-in
- User feedback starts earlier

Variety

- Build lots of prototypes
 - Use photocopier and go to town
- Think creatively
 - Try different things
 - Get a feel for what client likes
- Overcome *Representational Determinism*
- Tools don't bias form

Representational Determinism

- The medium constrains your approach to solving a problem.
- Initial work done by Jiajie Zhang
 - Tic-Tac-Toe
 - Did four other equivalent representations
 - Showed that people did worse with other representations
- Why is representational determinism a problem? Who suffers from it?

Quality of user feedback

- High-fidelity prototype implies finished product
 - Users reluctant to make large-scale modifications
 - They are paying ...
 - Architecture story
 - Users view hi-fi differently???
- High-fidelity prototype implies less variety
 - Limited time to build
 - Even prototyping takes time ...

Does Feedback Differ?

- CHI study in 2003
- Found no difference between feedback from a computer prototype vs feedback from a paper prototype for two projects
 - Ticket machine
 - Calendar system
- Notes still lots of reasons to use paper prototypes, including
 - Prototyping tools don't support components and ideas
 - Want to incorporate non-technical in design process
 - When evaluation may lead to lots of drawings

Does Feedback Differ?

- Maybe
 - You can't really prove that it doesn't in every case with only two projects
 - Issue with independent variable
- So what?
 - Cost is a huge factor
 - Designers are cheaper than developers
 - See comments in Erick Schonfeld's post.

What should prototypes look like?

- Designs can be
 - New and innovative
 - Boring and common-place
- Frequent suggestion
 - Look at current designs for strengths and weaknesses, then emulate and improve
 - There's a huge advantage to not being first-to-market
- Follow platform design guidelines unless you have a very good reason not to

Testing Low-Fidelity Prototypes

- Low-fidelity prototypes are tested in unique ways
 - No system, only rough screen shots
- Goal is to understand “what user is thinking”
 - Need techniques that prompt for this
- Common approaches
 - Person down the hall testing
 - Walkthroughs
 - Thinkalouds

Person down the hall testing

- Common in the real world
- When designing something, grab someone at work and get them to take a look
- Usually some form of walkthrough where you show them each screen, describe functionality
- Feedback is usually based on their background
 - Does it do what it is supposed to
 - Can they see functionality
- But ... highly useful as a sanity check

Walkthroughs

- A series of sketches
- Walk user representatives through different screen shots
- Ask users what they would do on each screen
- Advantages
 - Fast overview of system
 - Very useful for early stage sketches
- Disadvantages
 - Feedback limited by no “doing”
 - Risk of over-control of execution by experimenter
- Can augment walkthroughs with “think-aloud” protocol

Thinkalouds

- Two methods
 - Retrospective
 - Capture video of users using system
 - Watch video with users
 - Users comment on their actions and present their thinking
 - Very common with Difficult-to-evaluate systems like ATC
 - Can introduce post-hoc rationalizations
 - Concurrent
 - Most common during UI design

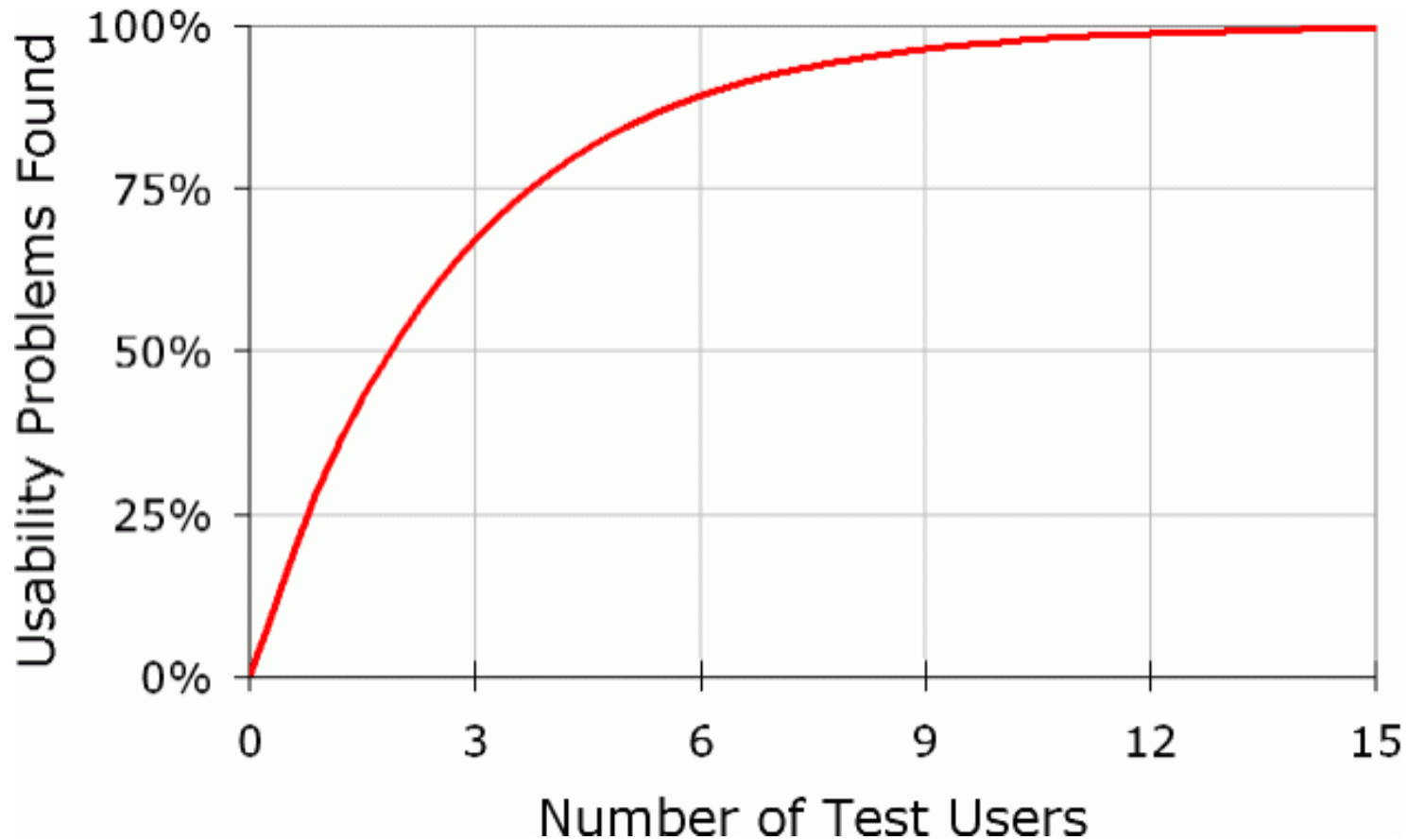
Concurrent Thinkalouds

- Observe user using your prototype
- Encourage them to “think-aloud”
 - Express what they are thinking and wondering at each moment
- When user is not having problems they work fast
 - Faster than they think
- When user is having problems, they slow down
 - Think aloud can reveal aspects of bad mental models, poor affordances, insufficient constraint, poor feedback, etc.
- Sometimes, when under heavy load, user will pause
 - Essential to continue to encourage them to think-aloud, but in a friendly way
- Tasks can be specified (“Could you schedule a reservation?”) or open-ended (user chooses what he/she would like to do with system)
- Informal technique – creating an informal atmosphere will result in more successful session

ThinkAlouds – Advantages and Disadvantages

- Advantages
 - Not limited to paper prototypes
 - Mathbrush
 - Rapid, high-quality qualitative feedback
 - Data is rich
 - Observations, hearing
 - Can ask questions to get complete information
 - Can help user if it becomes necessary
 - Flexibility in initiative
 - Doing, so less opportunity to give rote positive assessment
- Disadvantages
 - Limited sample?

Why you only need to test with five users



But recall the assumption that any usability problem typically affects 31% of users