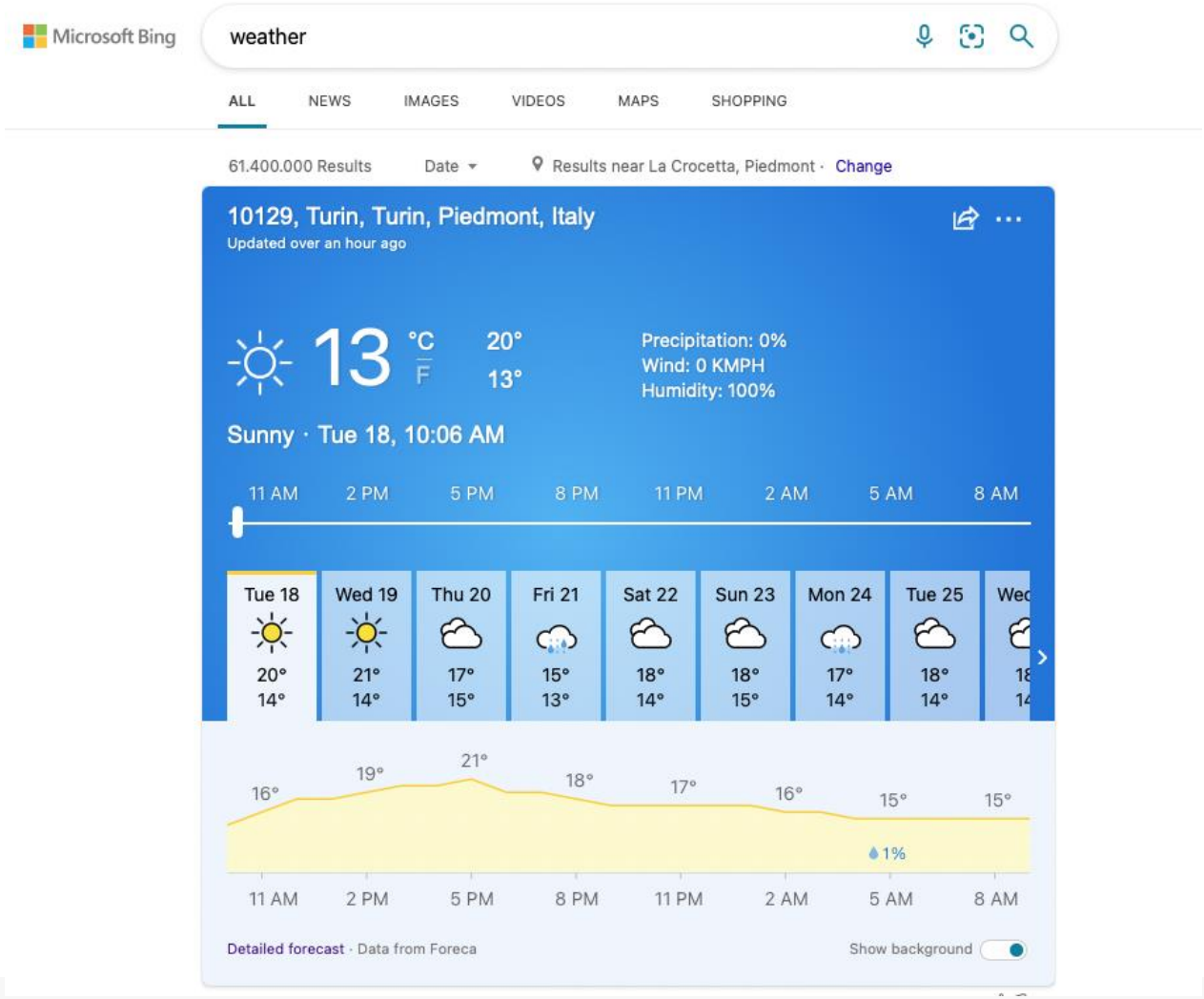
Scenarios, Storiboards, and Low-Fi Prototyping

User Experience Design

Alberto Monge Roffarello

Academic Year 2025/2026

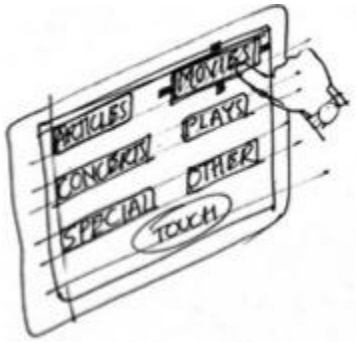
Hall of Fame or Shame?



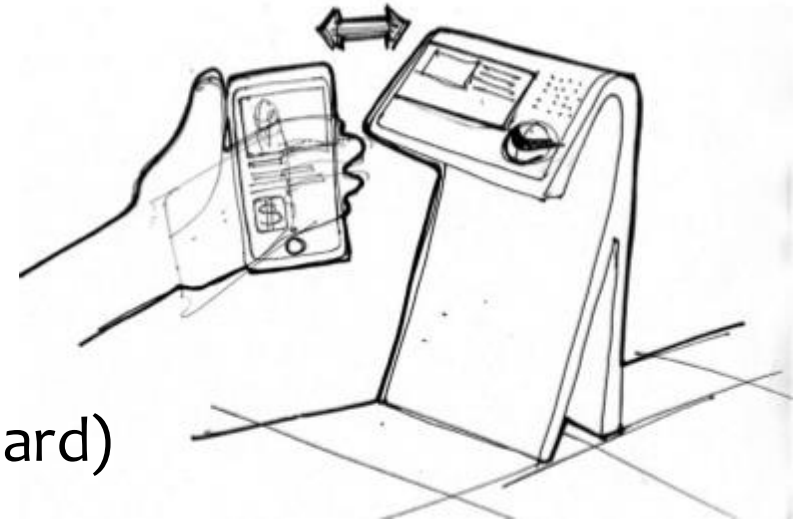
Sketches

Quick drawings to convey a part of the interface, or a feeling about a design

Sketch



- An individual drawing showing
  - A single user interface screen
  - A drawing of an artifact part of the system
  - The shape of an interaction object



- Gives a <u>static</u> view of a possible interaction
- Helps setting the interaction context
- Often, part of a longer representation (e.g., a storyboard)

Scenarios

Possible sequences of actions for reaching user goals

Scenario

- Scenarios are stories for design: rich stories of interaction
- Description of how the user engages the interactive system to solve a specific task/goal
- Formats:
  - Written summary, Use Case
  - Graphical sketches (→ Storyboard)
  - Flowcharts, Transition Diagram…

Level of Details In Scenarios

- **Stories**
  - From needfinding
  - Used for understanding what people do and what they want
- **Conceptual** (abstract) **Scenarios**
  - Used for generating ideas and specifying requirements
  - Abstracts tasks from stories
  - No reference to technology
  - May lead to different concrete scenarios
- **Concrete Scenarios**
  - Used for envisioning ideas and evaluation
  - One possible solution to a Conceptual Scenario (may try many alternatives)
  - Shows how technologies are used in the user context
  - Key design features are included
- **Use Cases**
  - Used for specification and implementation (→ software engineering)

Storyboards

Comic book – like representation of user scenarios, with emphasis of how the system supports users in the development of the task

Storyboard

- "A graphical depiction of the outward appearance of the intended system, without any accompanying system functionality"
- A hand-drawn comic that features the execution of a task (like a concrete scenario)
- With a few panels (sequence of sketches) it conveys what a person may accomplish
  - Always include people
- They communicate **flow**, showing what happens **at key points** in time
- No artistic skills are required
  - Not about "nice pictures"
  - About communicating ideas

What To Find In a Storyboard

- Illustrate a goal (for the task)
- How a task unfolds (people interacting among themselves and with devices)
  - Repeated for all significant steps
- At the end, how they accomplish their goals (satisfactory outcome)

- Storyboards are **all about tasks**

Example

This storyboard illustrates how the app had already downloaded the daily recipe to the user's smartphone, so he could look it up and check the shopping list while on the underground, before shopping for ingredients and making a healthy meal.

http://alexmevissen.com/2014/07/16/storyboarding/

Example

This storyboard illustrates how the app can show the user that a home cooked meal can be quicker than ordering food delivery, using left over ingredients in the fridge.

http://alexmevissen.com/2014/07/16/storyboarding/

Storyboards Should Convey…

- Setting
  - People involved
  - Environment
  - Task being accomplished
- Sequence
  - What steps are involved?
    - Not the detailed UI
    - What role the UI plays in helping users achieve their goal?
  - What leads someone to use the system?
    - The "trigger" for the task
  - What task is being illustrated?
- Satisfaction
  - What's the motivation for the user?
    - The end point to reach after all the steps
  - What's the end result?
  - What need are you "satisfying"?

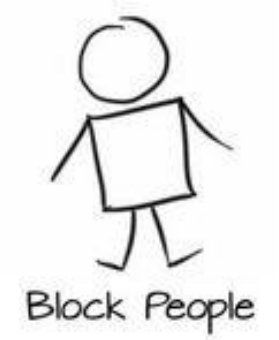
Handling Dynamicity In Storyboards

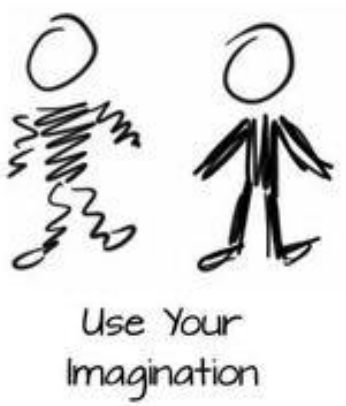
- Traditional storyboarding
  - “Comic book” conventions: actors, speech bubbles, background
  - Notes attached to each scene explaining what is happening
- Scored storyboards
  - When the user interface is highly dynamic, or contains specific media elements
  - Add specific annotations focusing on movement, colors, sounds, …
- Text-only storyboards
  - When the interaction behavior is too complex to compact into an illustration, use a longer text description

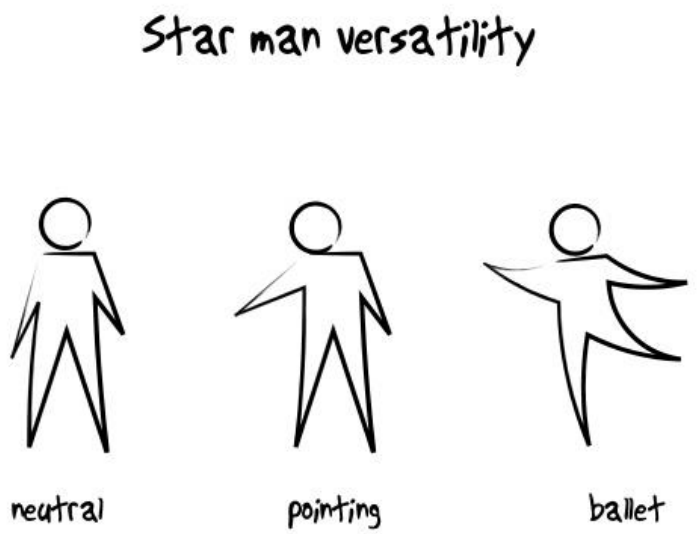
Why Hand-drawn?

- Quick
  - No need to spend time in graphics tools (they would "push" you to focus on details, too)
  - Able to experiment different scenarios
- Imprecise
  - Users feel free to express more comments and suggestions w.r.t. a more "polished" version
  - Focus on the content (the graphics is obviously ignored)
  - No distraction by fonts, colors, icons, …

~~Drawing~~ Sketching People







Benefits of Storyboards

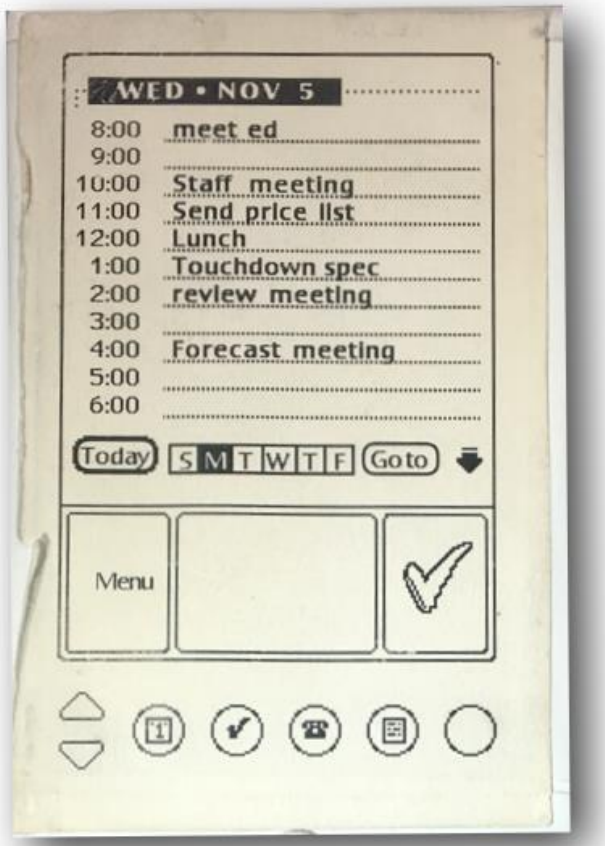
- Emphasize how an interface accomplishes a task
- Focus the conversation and feedback on user tasks
- Get everyone on same page about the app's goals
- Avoid nitpicking about user interface details (buttons, etc.)

Prototypes

Tangible approximations, at various levels, of system behavior and appearance, to cheaply and quickly evaluate and explore design decisions

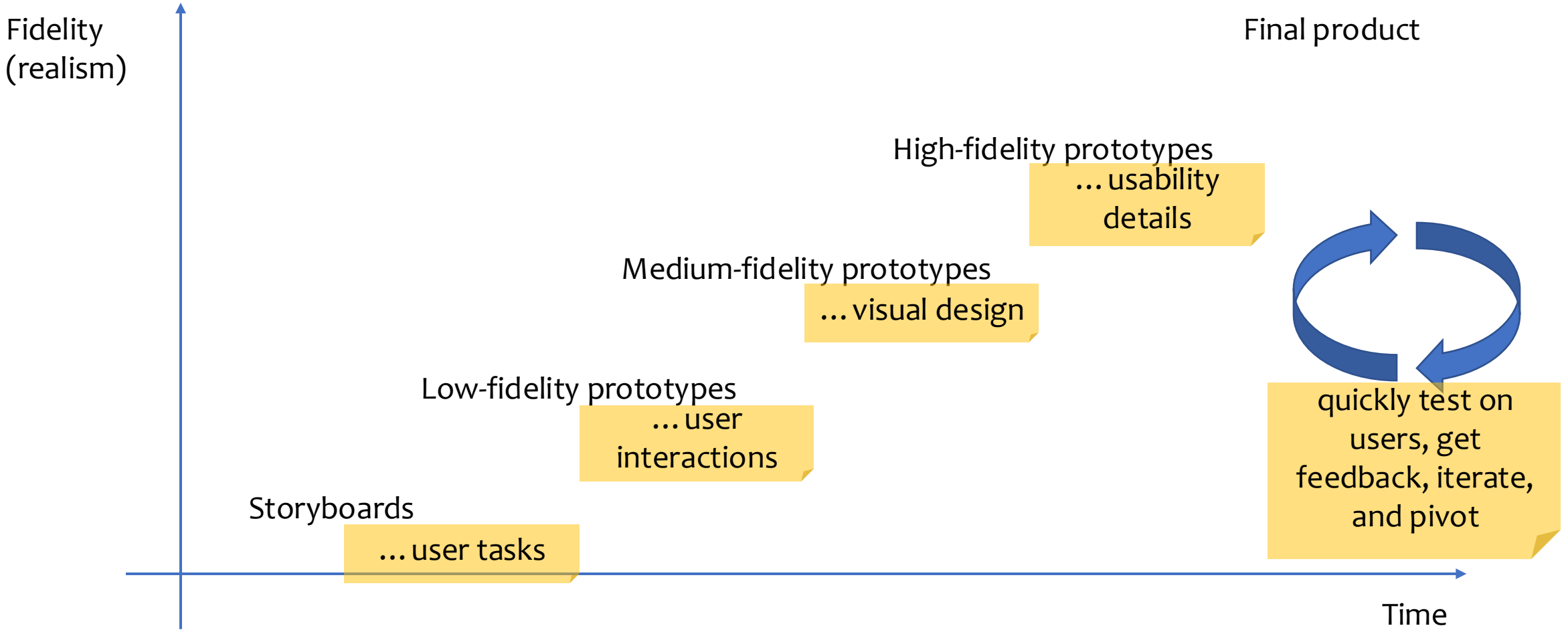
Prototypes

- «A prototype is a concrete but partial representation or implementation of a system design»
- «An easily modified and extensible model (representation, simulation or demonstration) of a planned software system, likely including its interface and input/output functionality»
- One of the most powerful tools for design exploration, visualization, and testing
- They let us ‘see’ and ‘feel’ interactivity (simulated or real)



source: https://albertosavoia.medium.com/the-palm-pilot-story-1a3424d2ffe4

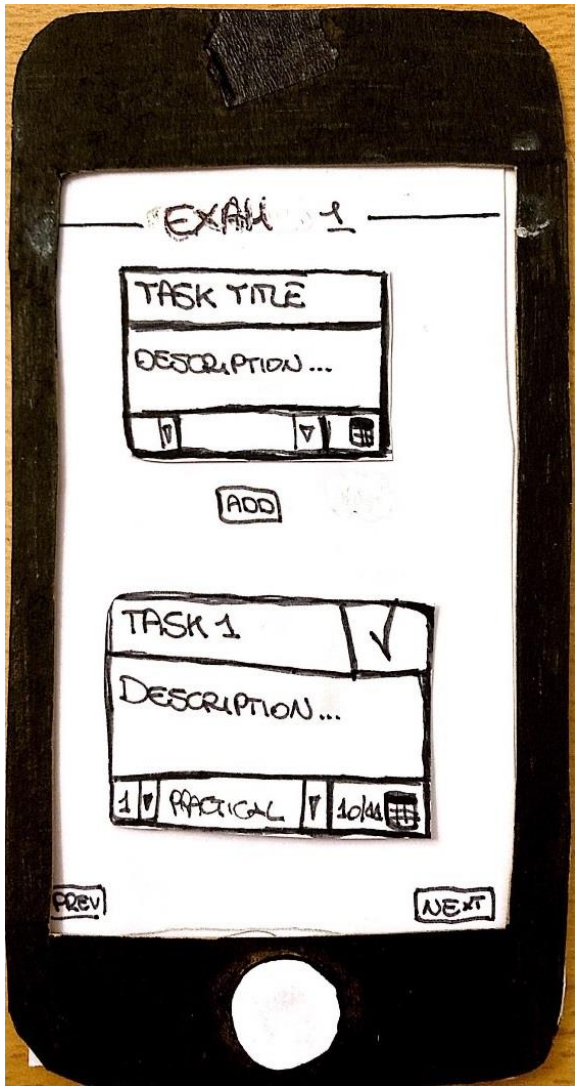
Prototypes Facilitate Conversations About…



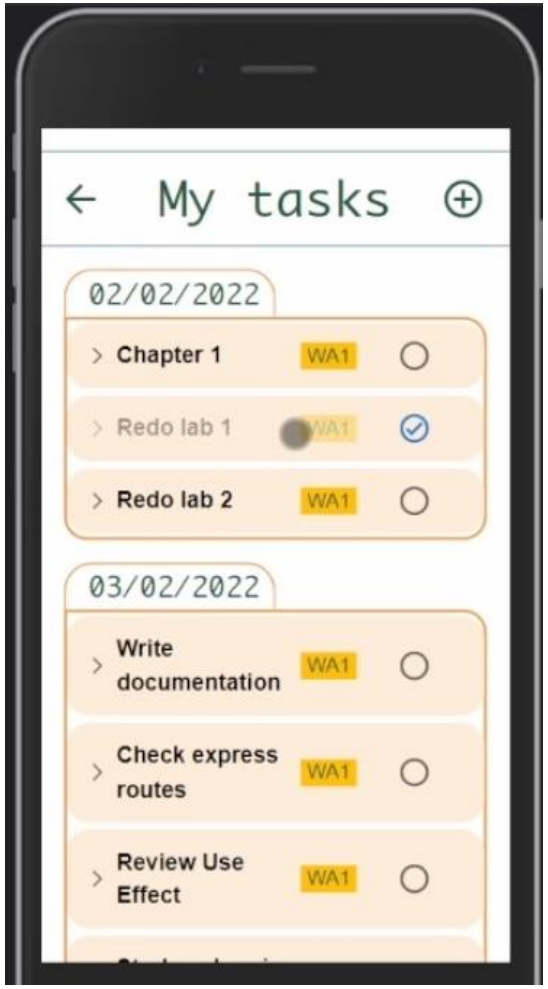
Low to High Fidelity Prototypes

Low-fi

- Lays out the *main* information, interactions, and design choices
- With many missing details



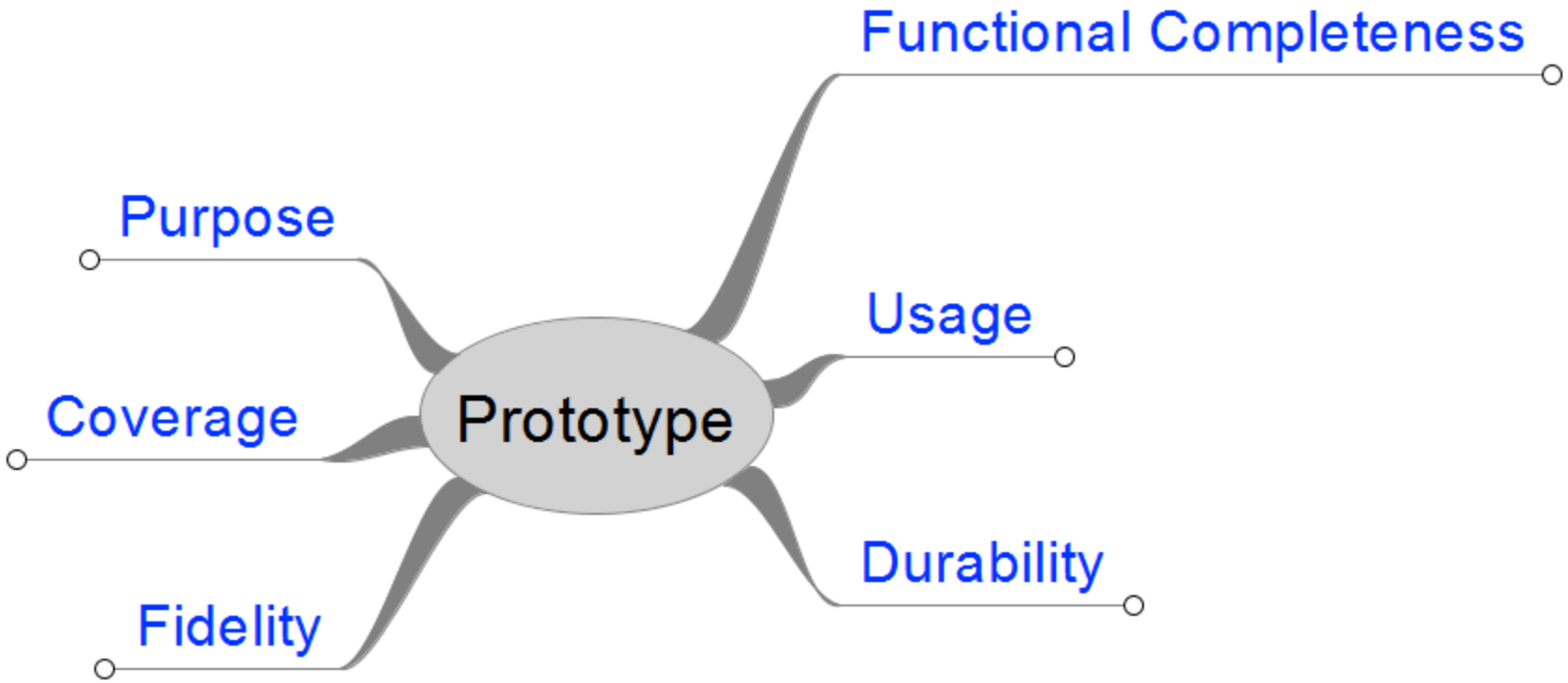
Low to High Fidelity Prototypes



Hi-fi

- It *looks like* the final product

Characteristics of Prototypes

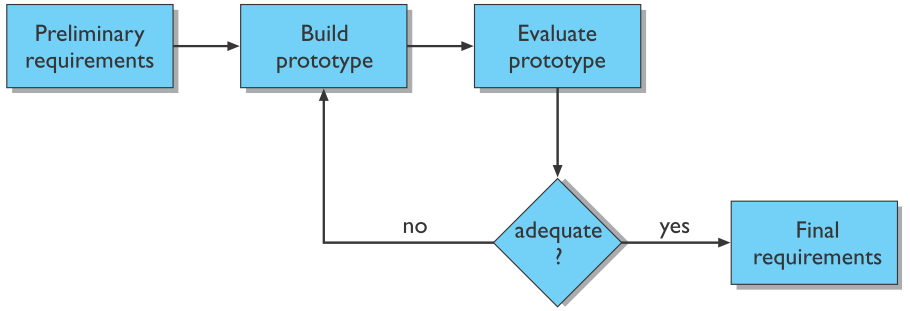


Possible Purposes For a Prototype

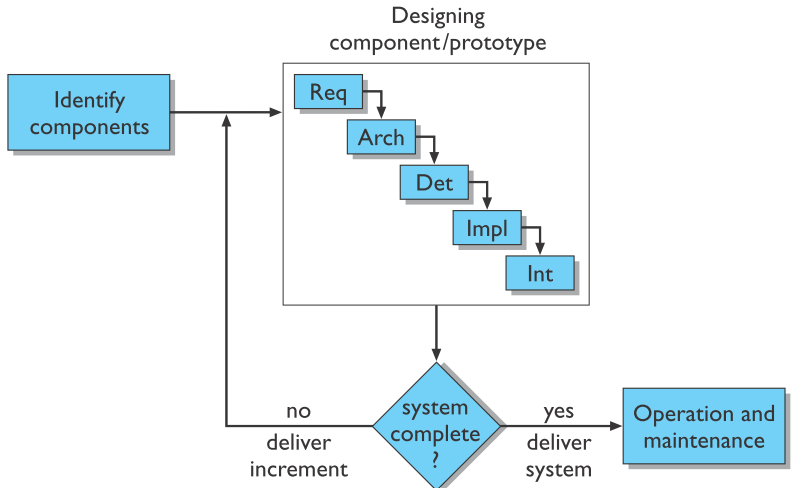
- Expert analysis
- Check with design rules and guidelines
- Involve users in a controlled experiment
- Involve users “in the wild”
- …

Durability (1)

- **Throw-away prototype:** used to assess some qualities of the system (gain knowledge), then discarded

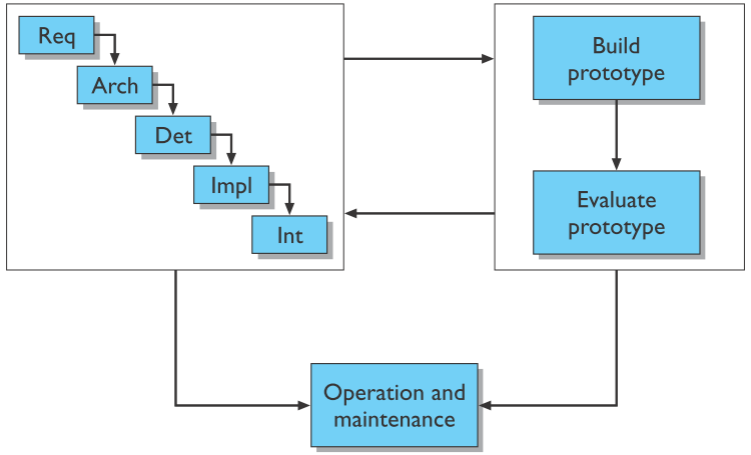


- **Incremental prototype:** the system is developed as incremental modules, each of them released in a separate step

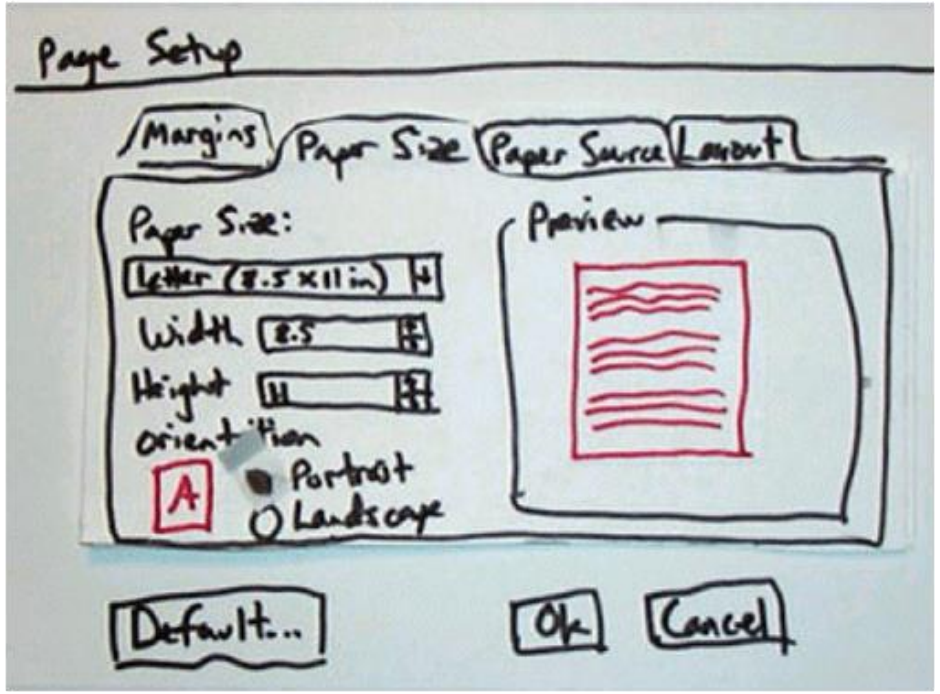


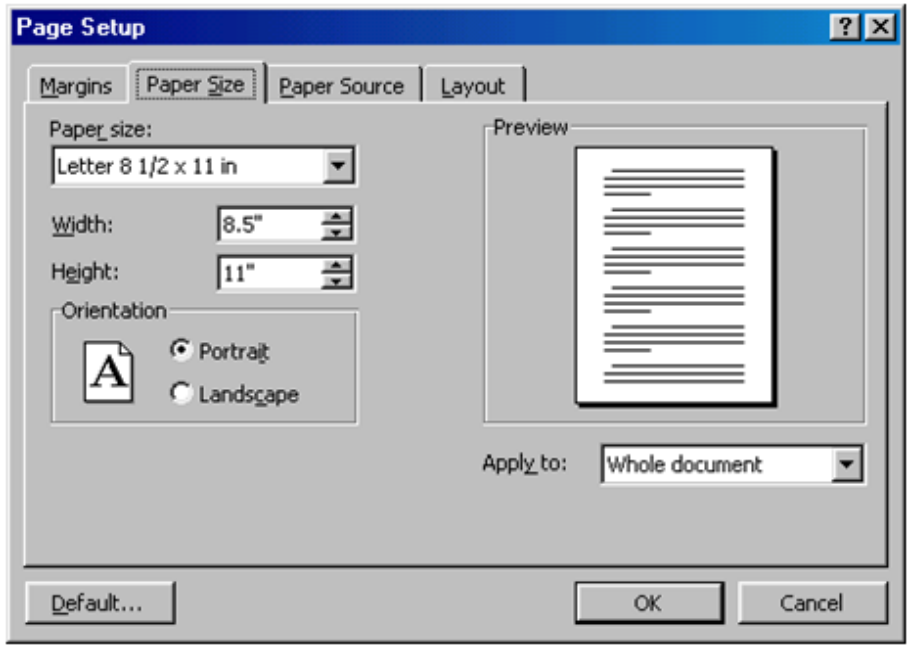
Durability (2)

- **Evolutionary prototype:** the prototype *becomes* the product; each product iteration builds upon the previous one



Fidelity: Different Information Is Conveyed



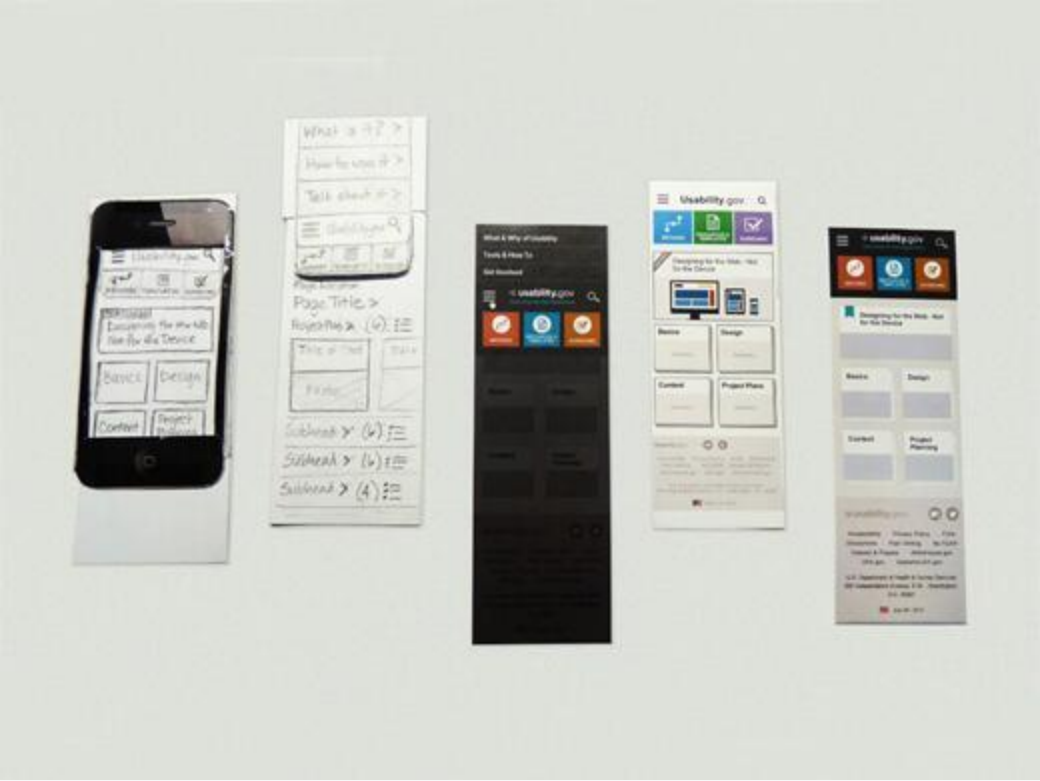


Low Fidelity Prototypes

How to start using an application, months before implementing it

Paper Prototypes

- A hand-drawn mock-up of the user interface (usually) on multiple sheets of paper of varying sizes



Key Features for Paper Prototypes

- Interactive paper mockup
  - Sketches of screen appearance
  - Paper pieces show windows, menus, dialog boxes
- Interaction is natural
  - Pointing with a finger = mouse click
  - Writing = typing
- A person simulates the computer's operation
  - Putting down & picking up pieces
  - Writing responses on the "screen"
  - Describing effects that are hard to show on paper
- Low fidelity in look & feel
- High fidelity in depth (person simulates the backend)

http://web.mit.edu/6.813/www/sp18/classes/11-prototyping/

Materials

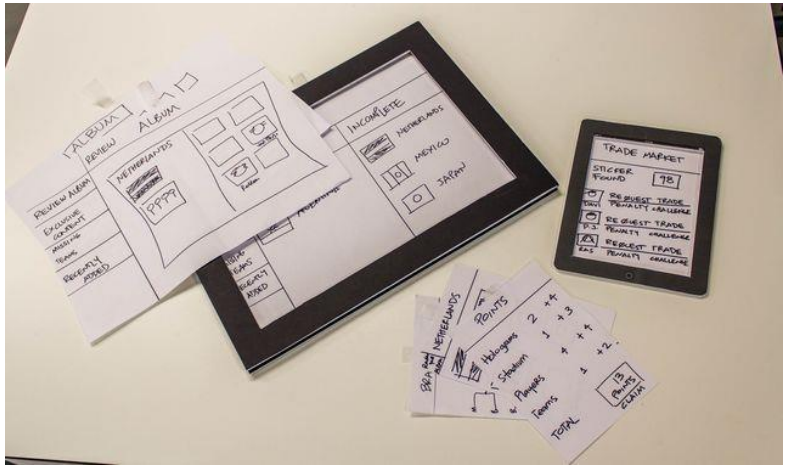
- Paper, Transparent paper
- Pens, Markers
- Post-It notes
- Glues, scotch tape, scissors
- Photocopies
- UI Stencils
- Reusable UI components
- Printouts of screenshots

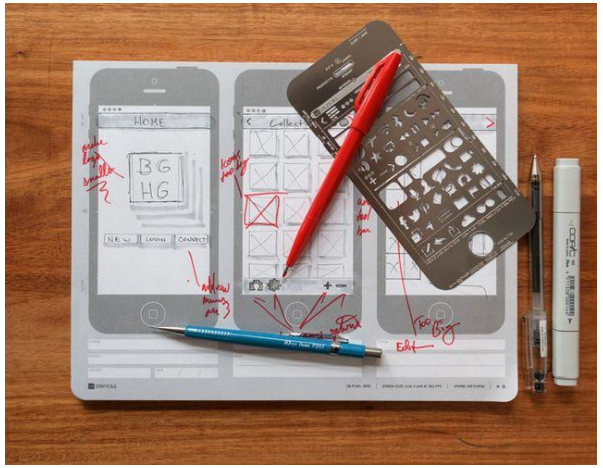
Why Paper Prototyping?

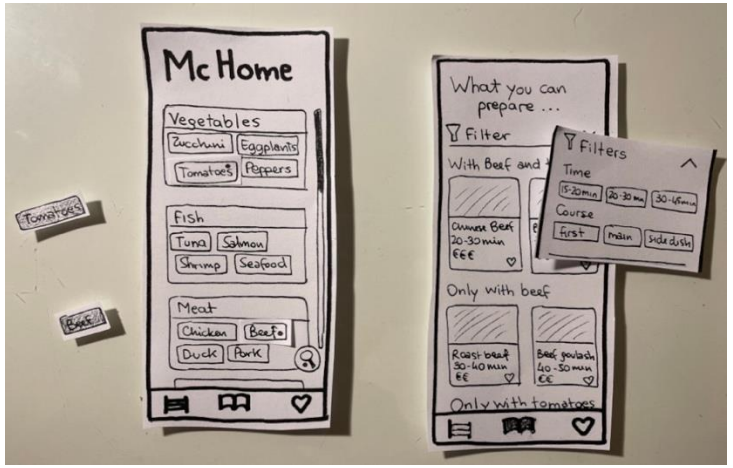
- Faster to build
  - Sketching is faster than programming
- Easier to change
  - Easy to make changes between user tests, or even **during** a user test
  - No code investment - everything will be thrown away (except the design)
- Focuses attention on big picture
  - Designer doesn't waste time on details
  - Customer makes more creative suggestions, not nitpicking
- Non-programmers can help
  - Only kindergarten skills are required

http://web.mit.edu/6.813/www/sp18/classes/11-prototyping/

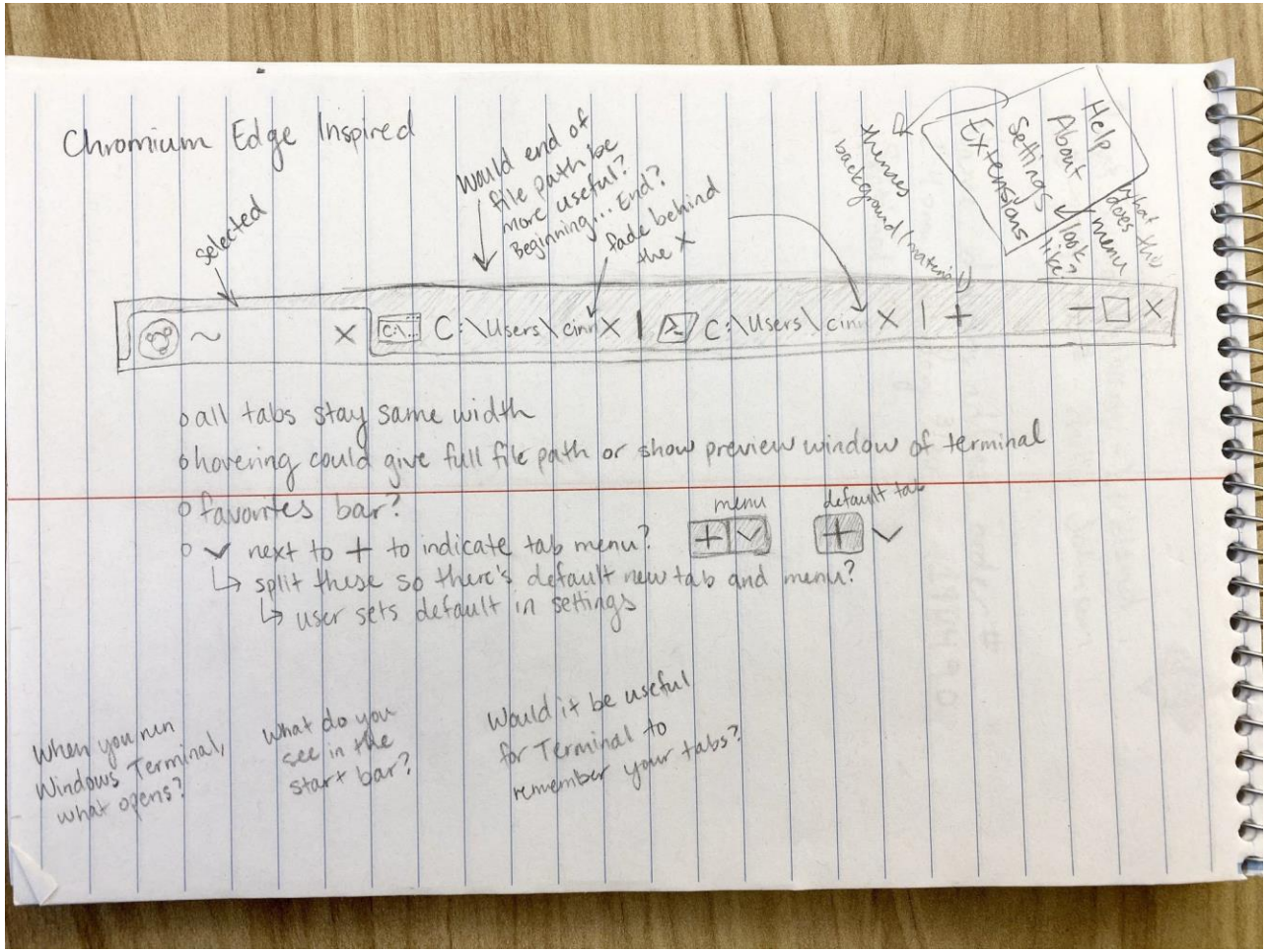
Paper Prototypes: Examples





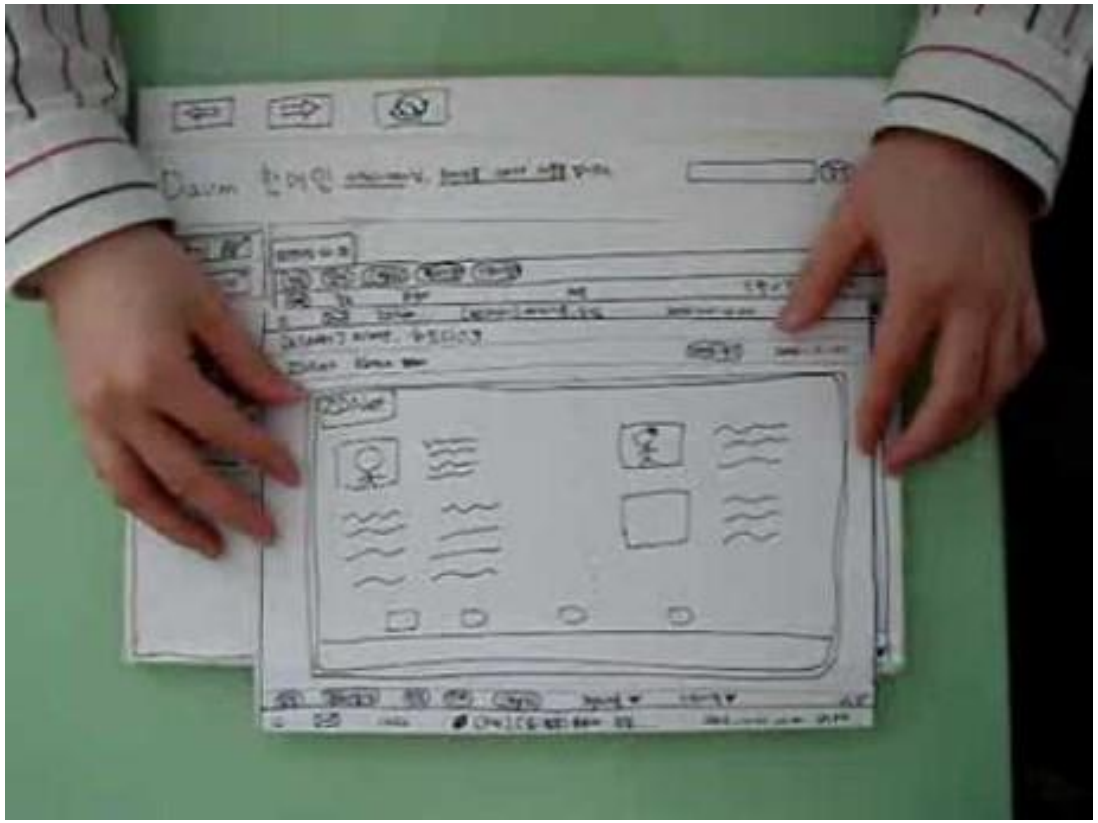


First Ever Mockup of the Windows Terminal Tab Bar



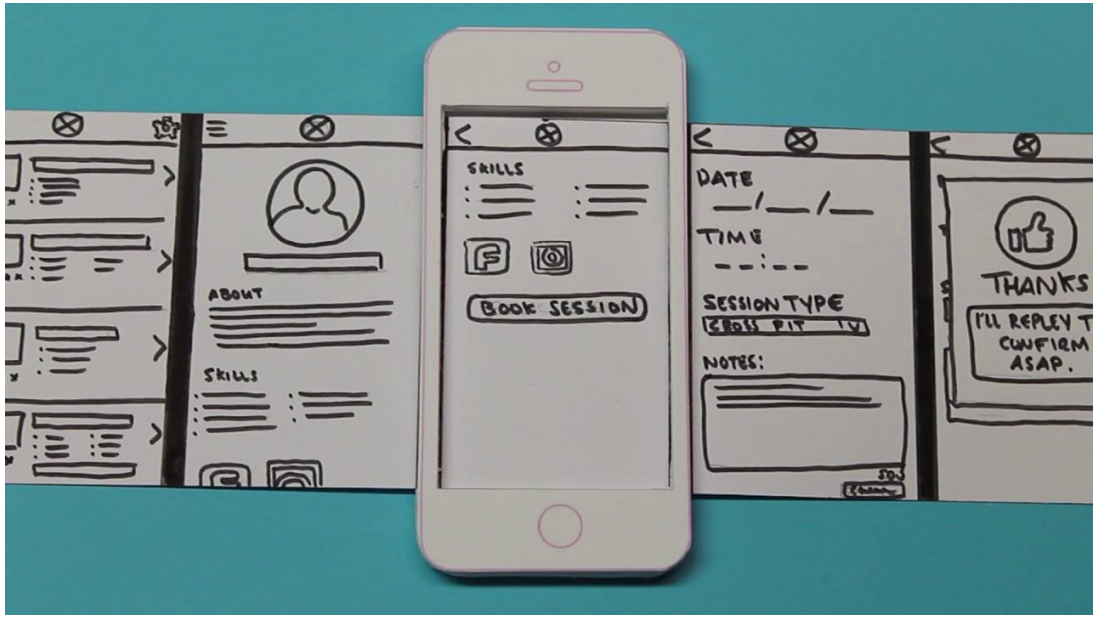
https://twitter.com/cinnamon_msft/status/1190015862201176065?s=20

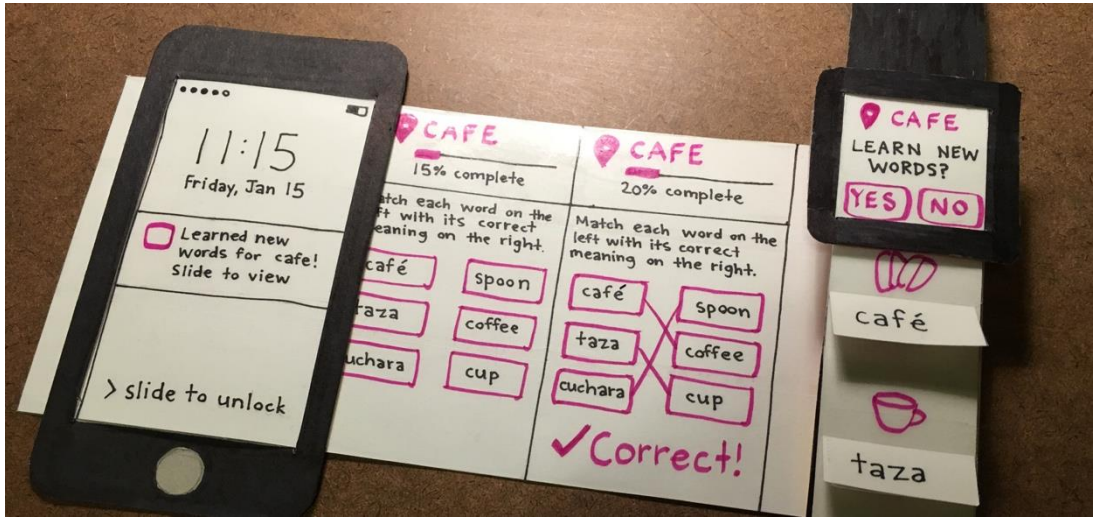
Creating Flows With Paper Prototypes



https://youtu.be/GrV2SZuRPvo

“Dynamic” Screens





How to Test a Paper Prototype

- The Design Team should cover these roles
- 'Computer' actor
  - Simulates prototype
  - Does not give any feedback that the computer would not
- Facilitator
  - Presents interface and tasks to the user
  - Encourages user to "**think aloud**" by asking questions
  - Keeps user test from getting off track
- Observer
  - Keeps mouth shut
  - Takes copious notes

Learnable Lessons From Paper Prototypes

Can Learn

- Conceptual model
  - Do users understand it?
- Functionality
  - Does it do what's needed? Missing features?
- Navigation & task flow
  - Can users find their way around?
  - Are information preconditions met?
- Terminology
  - Do users understand labels?
- Screen contents
  - What needs to go on the screen?

Cannot Learn

- Look: color, font, whitespace, etc
- Feel: efficiency issues
- Response time
- Are small changes noticed?
  - Even the tiniest change to a paper prototype is clearly visible to user
- Exploration vs. deliberation
  - Users are more deliberate with a paper prototype; they don't explore or thrash as much

References

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale: Human Computer Interaction, 3rd Edition, Chapter 15 “Task Analysis”
- David Benyon: Designing Interactive Systems, Chapter 11 “Task Analysis”
- http://www.usabilitybok.org/task-analysis
- https://www.usability.gov/how-to-and-tools/methods/task-analysis.html
- Google, Begin Today With Rapid prototyping, https://www.youtube.com/playlist?list=PL9KVIdeJ2K8NDpsiyYpcbB_qifd3y5CYZ
- MIT, http://web.mit.edu/6.813/www/sp18/classes/11-prototyping/#reading_11_prototyping
- Scott Klemmer, Storyboards, Paper Prototypes, and Mockups, https://youtu.be/z4glsttyxw8

Acknowledgements

- Some icons from https://icons8.com

- Some material by
  - http://www.inf.ed.ac.uk/teaching/courses/hci/0708/lecs/tasks.pdf
  - https://www.tutorialspoint.com/human_computer_interface/design_process_and_task_analysis.htm
  - https://www.slideshare.net/alanjohndix/hci-3e-ch-15-task-analysis

- Most of the slides are adapted from those used in the "Human Computer Interaction" course of Politecnico di Torino
  - http://bit.ly/polito-hci

License

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"

- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.

- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

- https://creativecommons.org/licenses/by-nc-sa/4.0/