



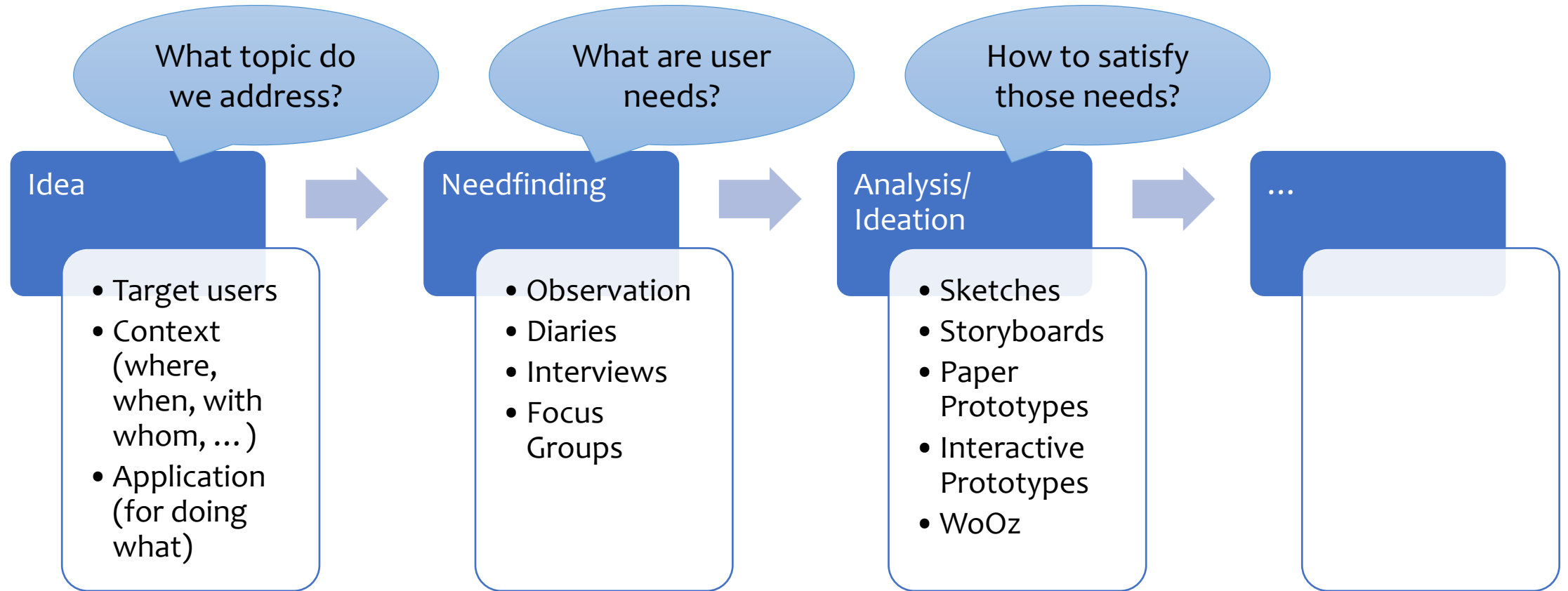
# Prototyping

**Human Computer Interaction**

Luigi De Russis, Fulvio Corno

Academic Year 2021/2022

# Process recap



# Previous steps

- After the project **idea**:
  - We want to help <<TYPE OF USER >>
  - When he is in <<LOCATION>> at <<TIME>> with <<OTHER USERS>>
  - While he is doing <<ACTION>>
  - To accomplish <<USER GOAL>>
  - Using <<DEVICES>>
- After **needfinding**:
  - All user needs emerging from observation and interviews
  - The most insightful 3-4 needs
  - Task analysis of those needs
  - How the project would address 1-2 deep user needs, and the strategies

# The goal

- **Envisionment:** making ideas visible
  - Generating new ideas
  - Evaluating new ideas (within the design group)
  - Testing new ideas (with users)
- Different tools and techniques, according to
  - The stage of design (early, ..., advanced, final)
  - The intended audience (designers, test users, clients, management, ...)
- **Error to avoid:** focusing on the user interface before focusing on the task that the user has to accomplish

# The method

- Techniques to **explore** different design **alternatives**
- Explore
  - Flows of action
  - Devices and their roles
  - Interfaces
- Alternatives
  - More than one possible design
  - Impossible to get it right the first time
  - Find the best possible solution

# Techniques

- Sketches
- Maps
- Scenarios
- Storyboards
- Paper prototypes
- Video Prototypes
- Hi Fidelity Prototypes

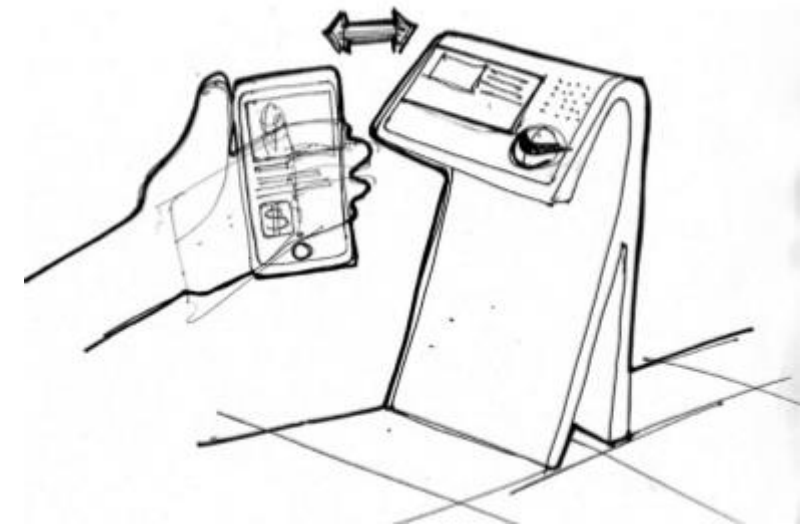
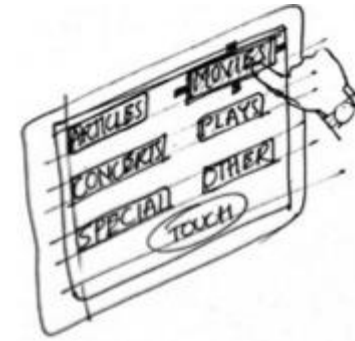
**“If a picture is worth a thousand words, a prototype is worth a thousand meetings” — IDEO**

# Sketches and Snapshots

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

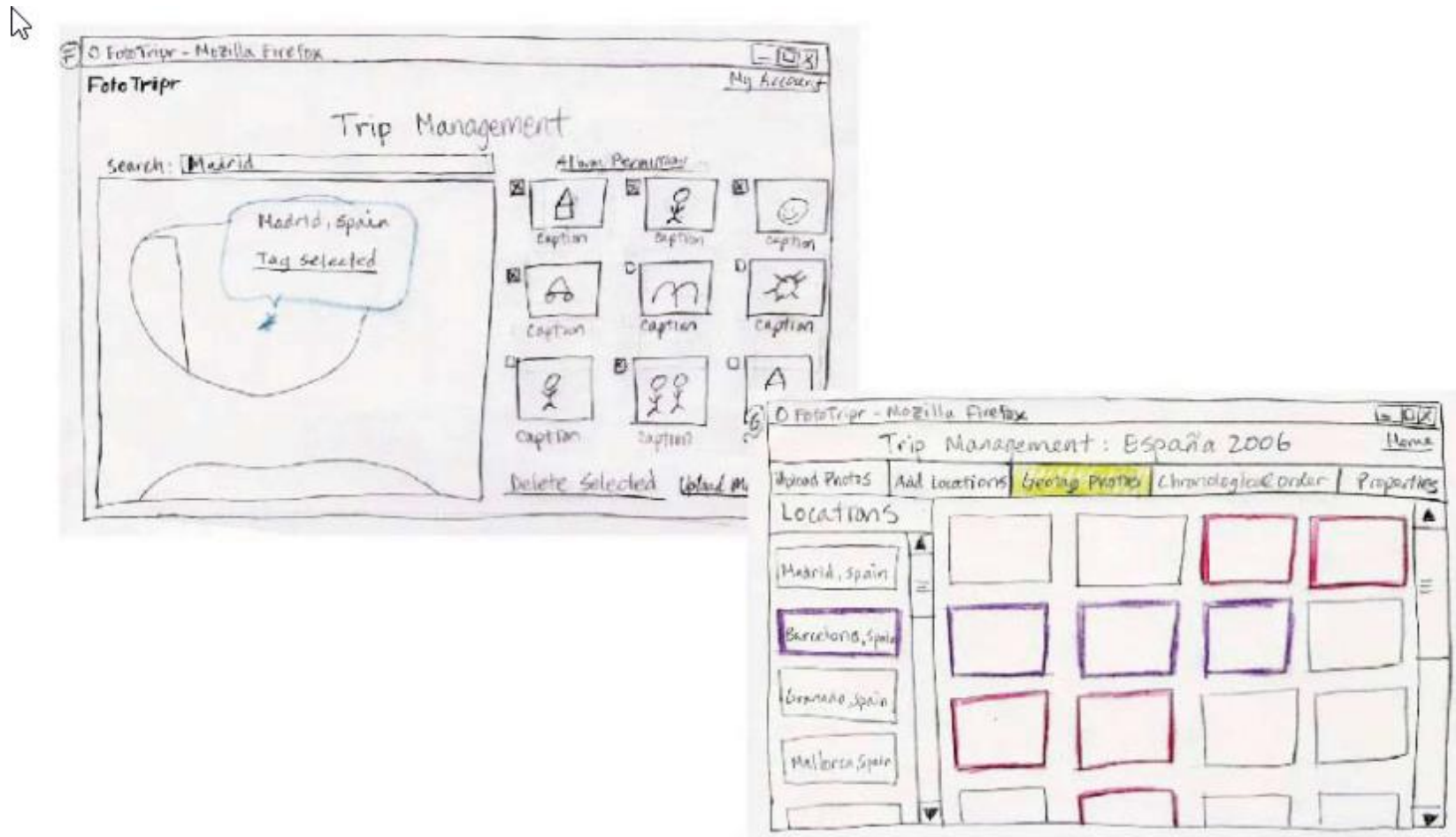
# Sketch

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





# Examples



# Maps

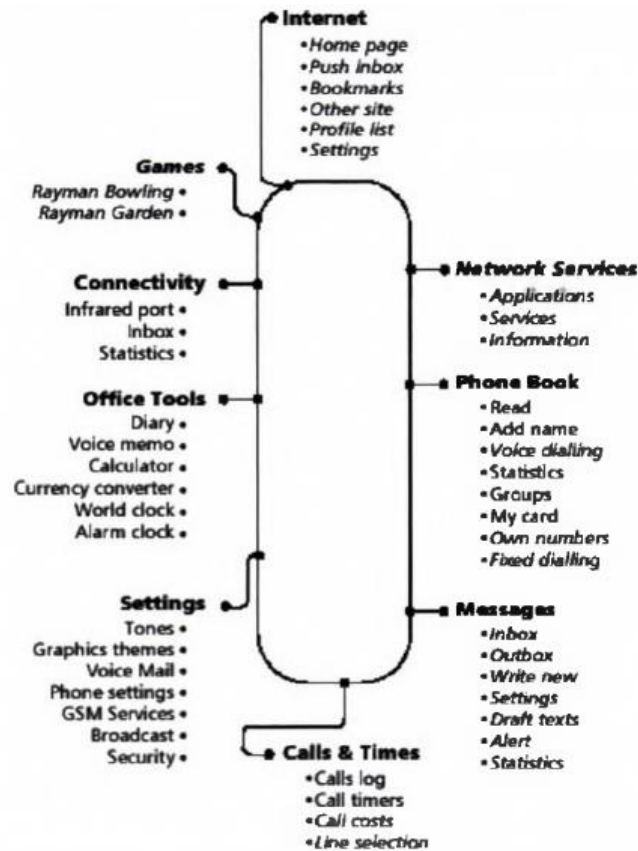
Visual overviews of navigation paths

# Navigation Map

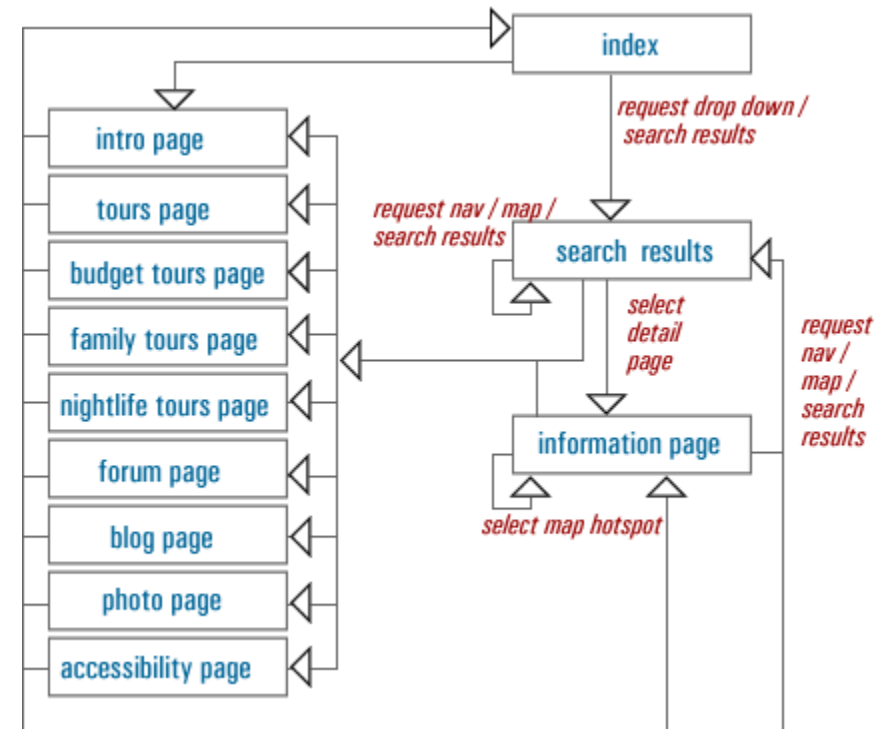
- A high-level view for the major structure of the interface
- Focus on how people move throughout the application
- Does not show the pages, only their organization and hierarchical relationship
- Related to the “information architecture” of the application

# Map examples

## Old-style mobile phone menus



## Website 'sitemap'



# 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
- Use for understanding what people do and what they want

## ▪ **Conceptual (abstract) Scenarios**

- Use for generating ideas and specifying requirements
- Abstracts tasks from stories
- No reference to technology
- May lead to different concrete scenarios

## ▪ **Concrete Scenarios**

- Use 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**

- Use 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



Stick People



Block People



Blob People



Star People



Triangle People



Use Your Imagination

Star man versatility



neutral



pointing



ballet



# 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)

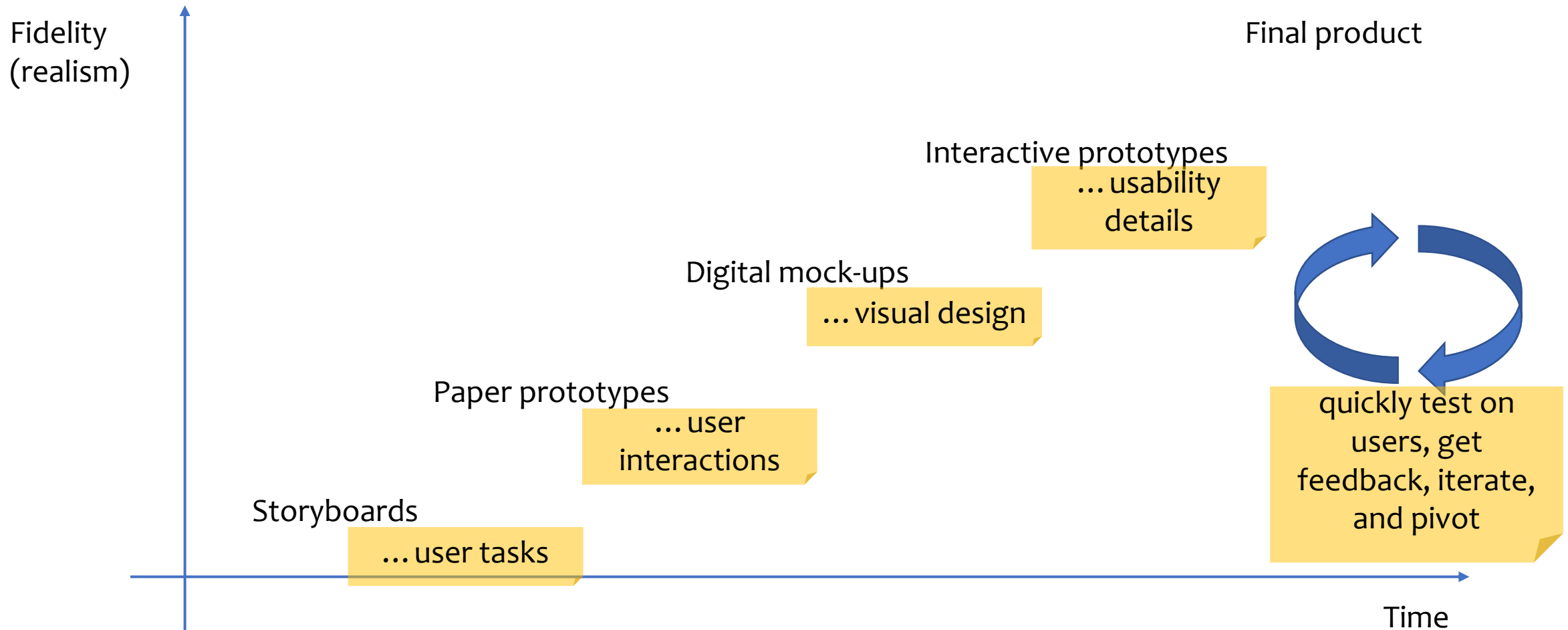
# Prototypes

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)

# Prototypes facilitate conversations about...

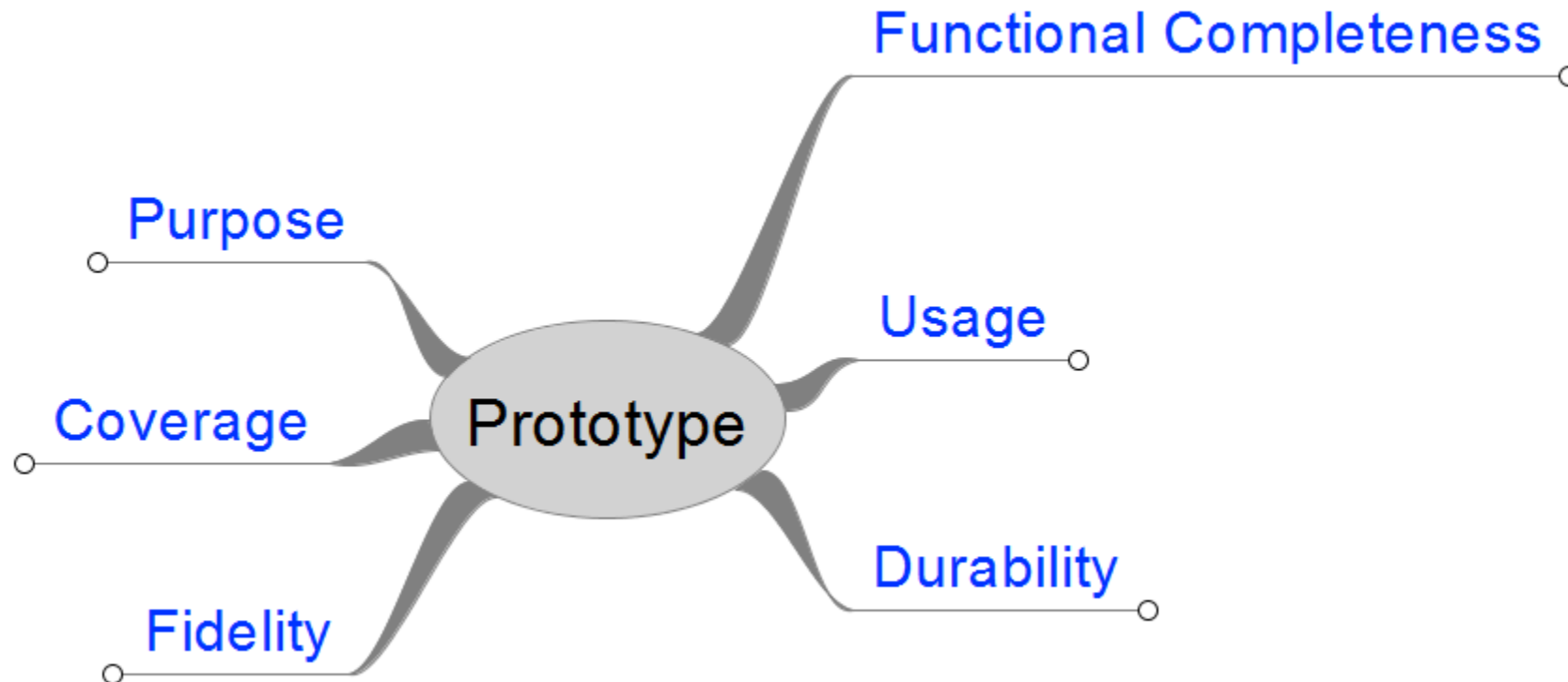


# Prototypes

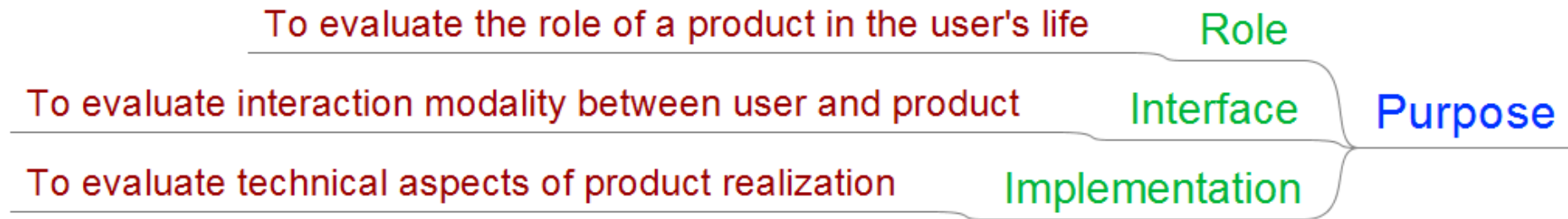
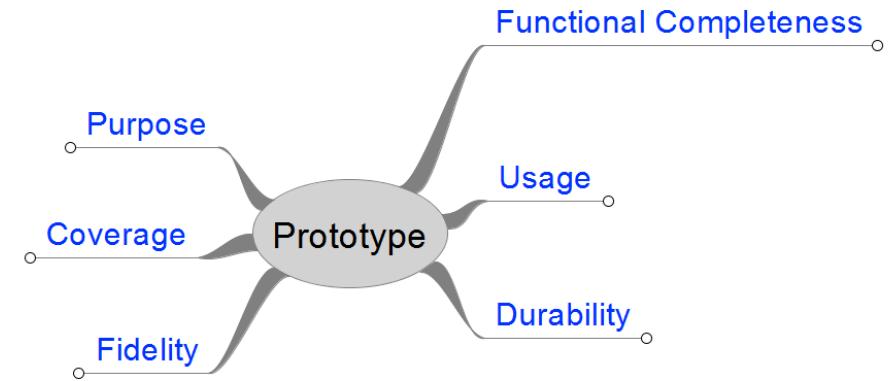
## What

- Paper designs
- Video prototypes
- Wizard-of-Oz
- Mockups
  - Low-fidelity
  - High-fidelity
- Preliminary versions

# Characteristics of Prototypes



# Characteristics of Prototypes

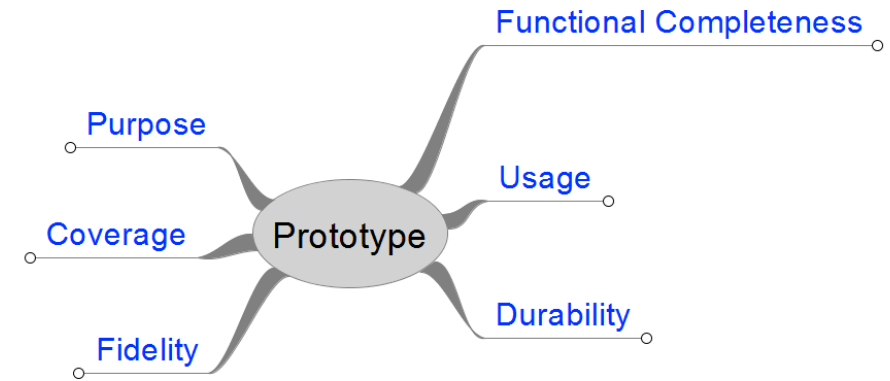


# Possible purposes for a prototype

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



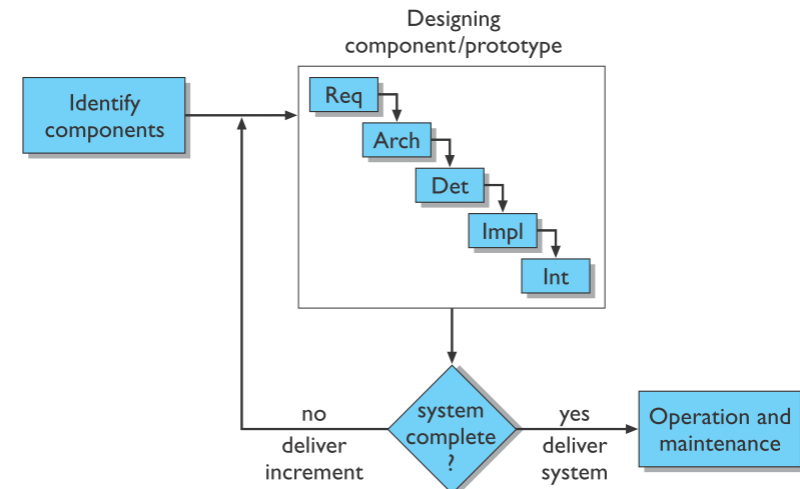
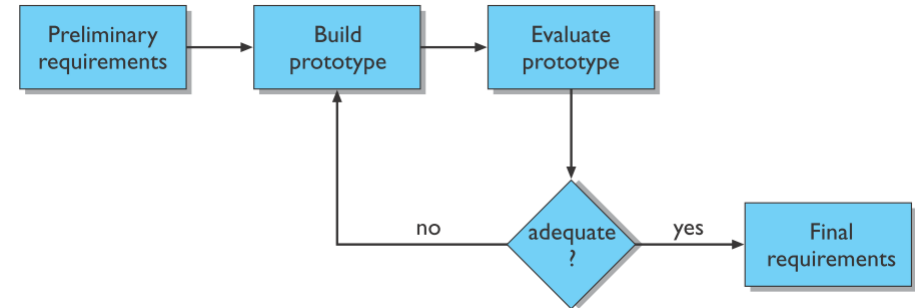
# Characteristics of Prototypes



Durability	Exploratory	A throw-away prototype used to clarify project goals, to identify requirements, to examine alternative designs, or to investigate a large and complex system
	Experimental	A prototype used to validate system specifications
	Operational	An iterative prototype that is progressively refined until it becomes the final system

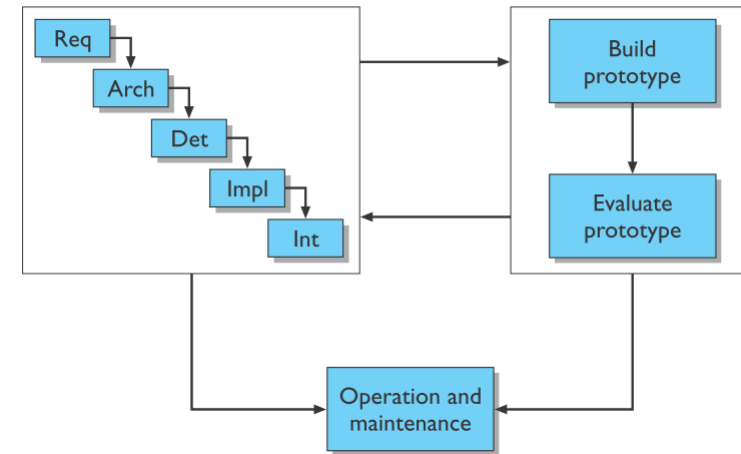
# Durability (1)

- **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

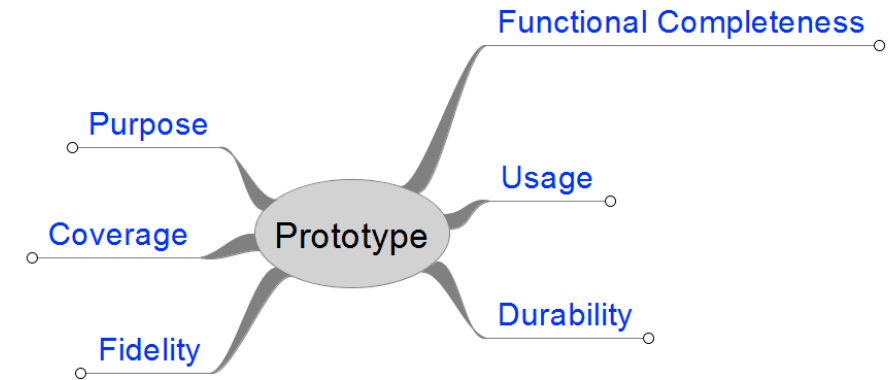


# Durability (2)

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



# Characteristics of Prototypes



A prototype of the entire system

- an expanded horizontal prototype
- models a greater number of features
- covers multiple levels of the system's structure chart
- useful throughout the design process

Global

Coverage

A prototype of a single usability-critical system component

- a vertical prototype that is focused on one feature
- useful at some specific stage of the design process

Local

Functional Completeness

Horizontal

A prototype that models many features but with little detail

- a horizontal slice of a system's structure chart from the top down to a specific depth
- most useful in the early stages of design
- purpose is to test the overall interaction metaphor, so includes common functions that the user is expected to perform frequently

Vertical

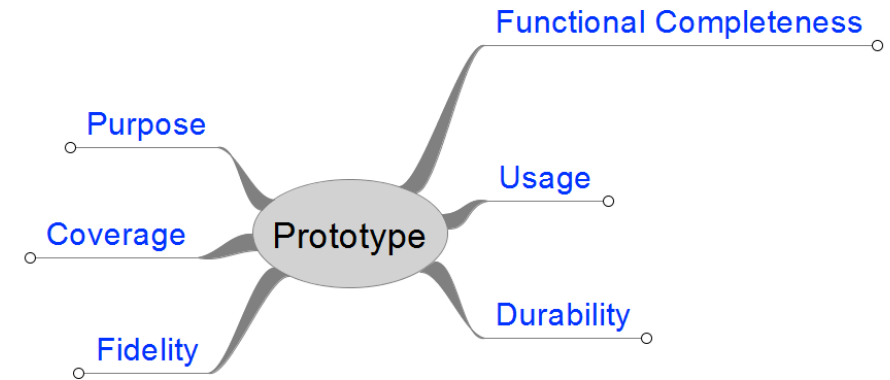
A prototype that models few features but with much detail

- a vertical slice of a system's structure chart from top to bottom
- most useful in the later stages of design
- purpose is to test details of the design

Diagonal

A prototype that is horizontal down to a particular level, then vertical below that point

# Characteristics of Prototypes



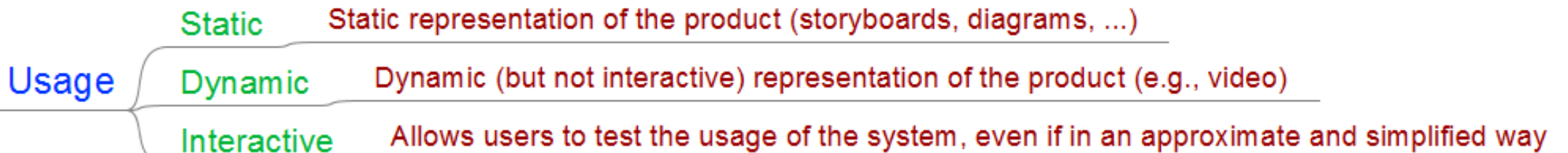
A set of drawings (e.g., storyboard) that provide a static, non-computerized, non-working mock-up of user interface for the planned system

Low

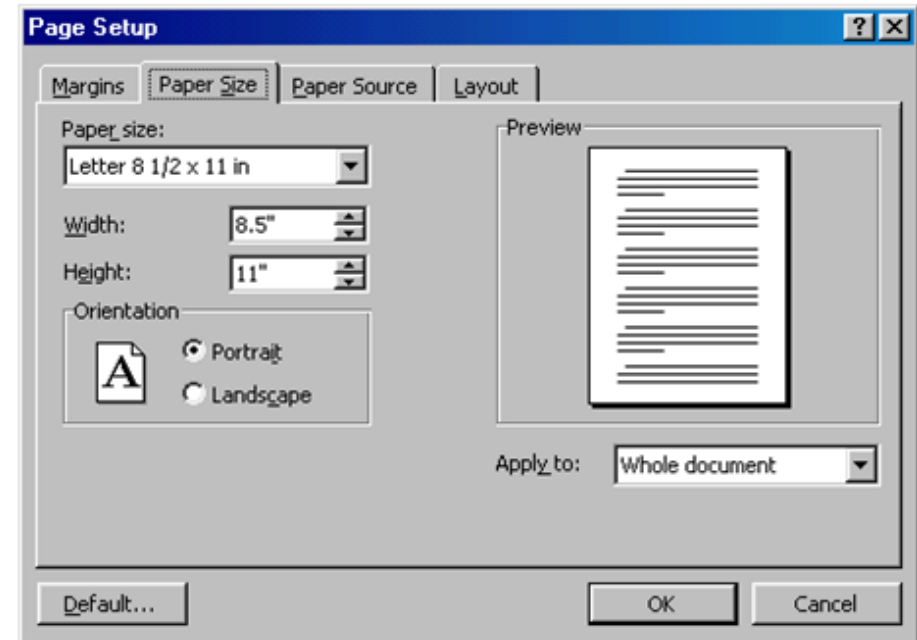
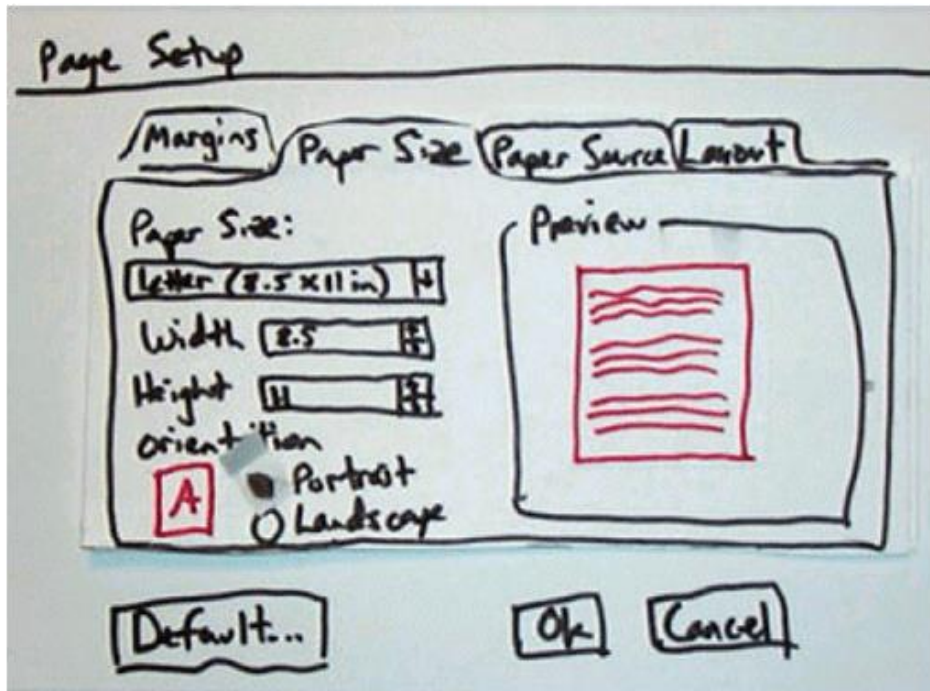
A set of screens that provide a dynamic, computerized, working model of the planned system

High

Fidelity



# Fidelity: different information is conveyed



# Paper 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)

# Materials

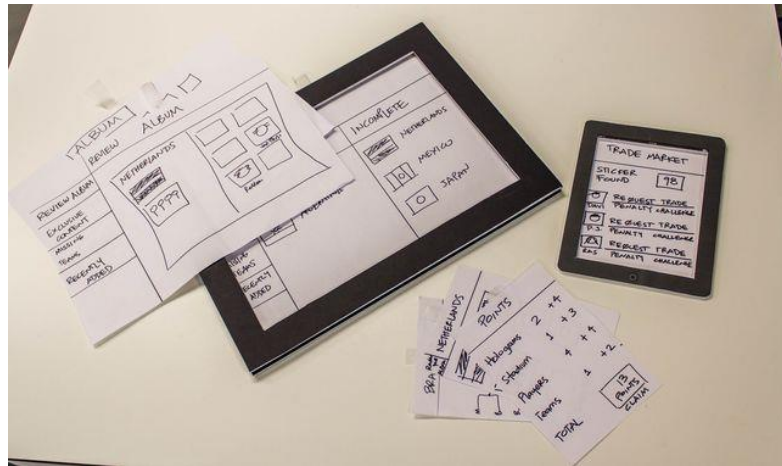
- Paper, Transparent paper
- Pens, Markers
- Post-It notes
- Glues, scotch tape, scissors
- Photocopies
- UI Stencils
- Reusable UI components
- Printouts of screenshots



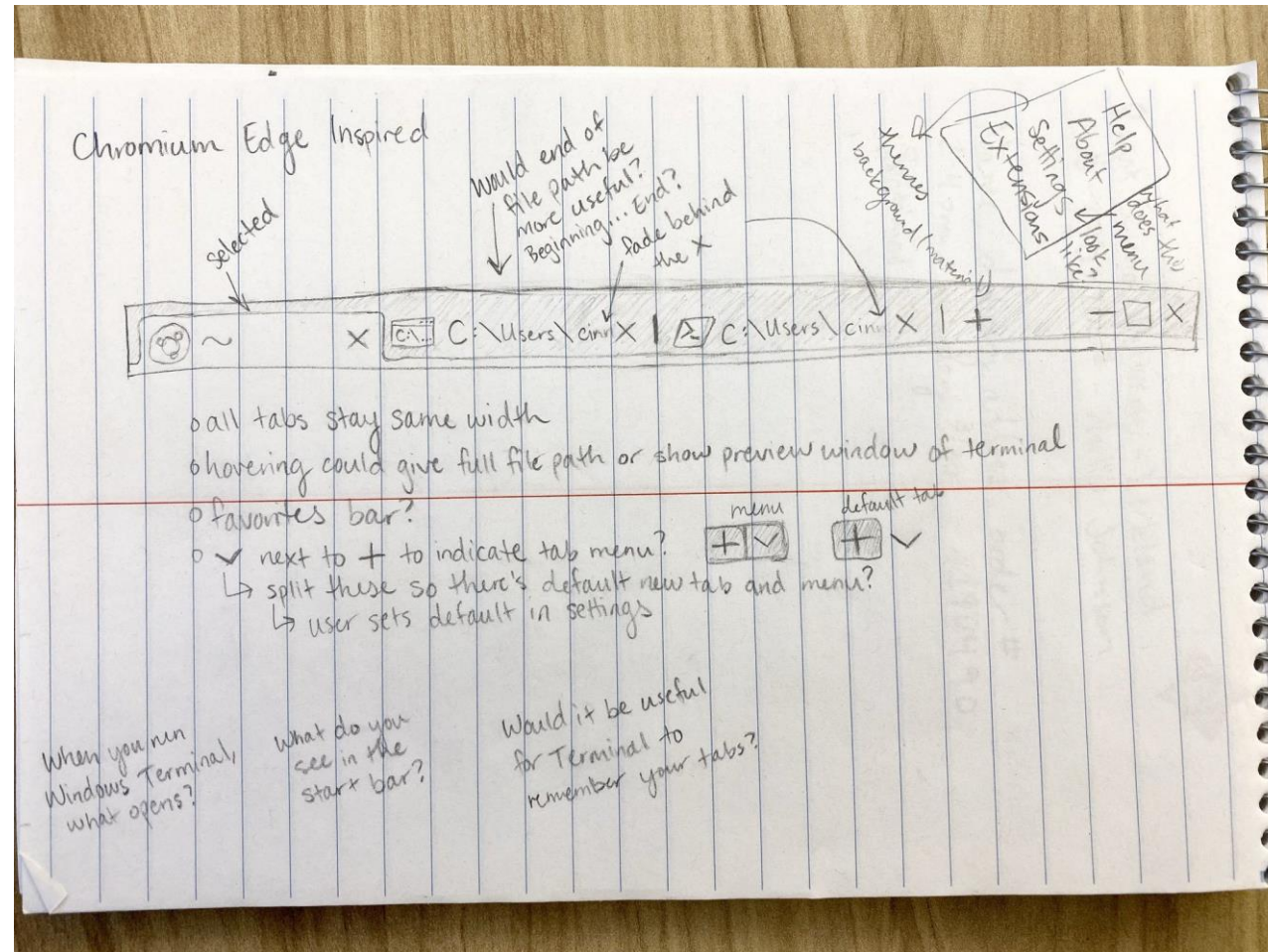
# Why Paper Prototyping?

- 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
- Nonprogrammers can help
  - Only kindergarten skills are required

# Paper prototypes: examples

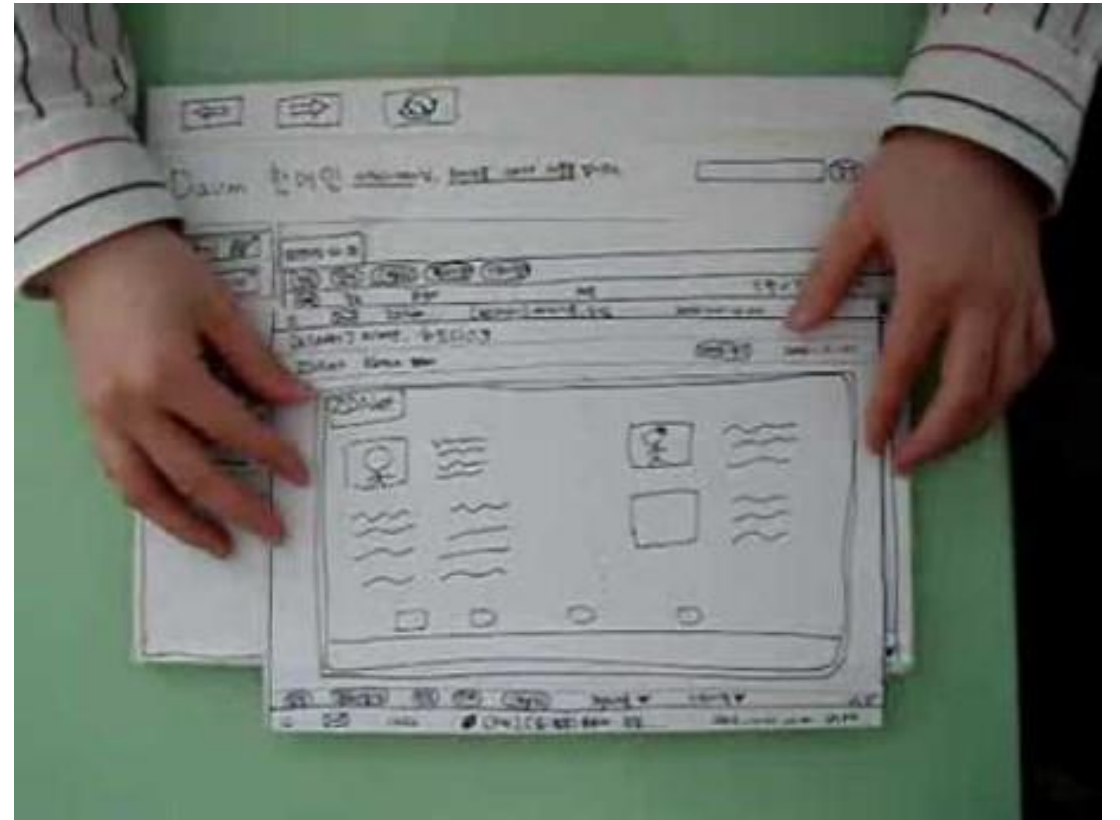


# First ever mockup of the Windows Terminal tab bar



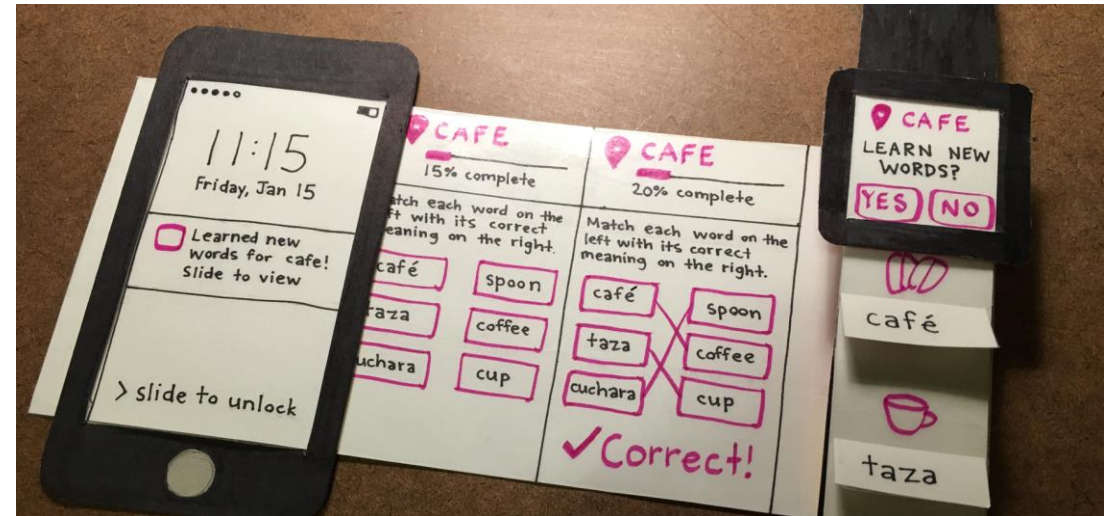
[https://twitter.com/cinnamon\\_msft/status/1190015862201176065?s=20](https://twitter.com/cinnamon_msft/status/1190015862201176065?s=20)

# 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
  - Doesn’t give any feedback that the computer wouldn’t
- 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?

## Can't 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

# Video Prototypes

Sharing a rich experience of your prototype

# Video Prototype

- A video that conveys your storyboard and/or paper prototype concepts.

# Example



<https://youtu.be/wbiYAqbZryA>

# Example



<https://youtu.be/kWsBvUnvCmg>

# Video prototype fidelity

- Informal, low fidelity
  - Just for brainstorming
  - A few minutes to create
- Medium fidelity
  - Starting with paper prototype
  - One-two hours to create
- High fidelity
  - Need to get support from organization or client
  - Expensive

# Required content

- Show the whole task (like a storyboard), including motivation and success
- Choose important tasks, that show cases when your system is performing really well
  - Tasks that you have observed
  - Key tasks in the application
- Defines the scope for an MVP: the shown tasks are the features of the first launch
- Defines the topics for the design team to argue discuss

# Creating a video prototype

- Define an outline
  - Or pick one of the storyboards
- Use minimum technology
  - High-quality equipment may become distraction
  - Reduce post-production and editing to a minimum
- Establish context
  - Choose representative users
  - Choose a meaningful location
- Focus on the message, not on the production quality



# Tips for production

- Bad audio is annoying and distracting
  - May also be a silent movie with “title cards” to explain what’s happening
- Choose the amount of interface you want to show in the video
  - Real-looking interactive prototype
  - Paper prototype
  - No interface at all (just users)
- Show both success and failure

# Benefits

- Cheap and fast
- Can more vividly inspire people's imagination – Great communication tool
- Clean & self-explanatory – just share a YouTube link
  - More portable than a paper prototype!
  - Good for “pitching” or “selling” to management
- Shows context of use: helps achieve common ground
- Can serve as a ‘spec’ for developers
- Ties interface design to user tasks
  - Ensure you develop all that is needed, nothing extra

# Medium Fidelity Prototypes

Wireframes, Powerpoints, Sketching tools

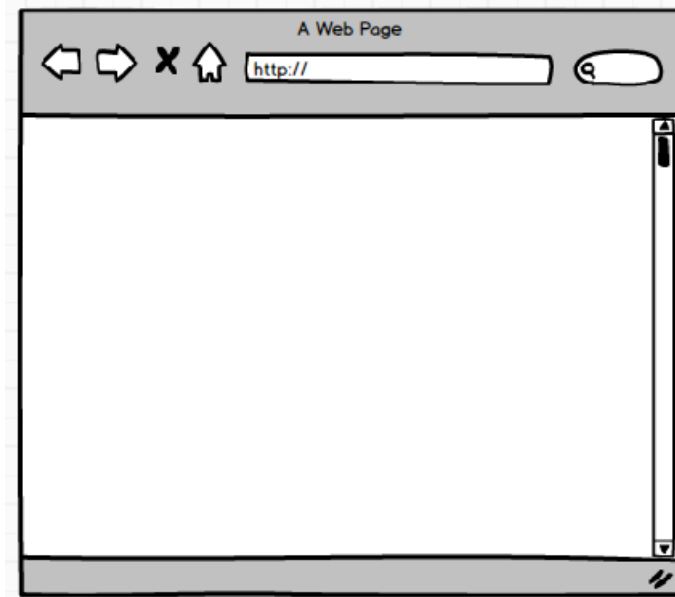
# Computer prototypes

- Interactive software simulation
  - Renders user interface
  - Accepts some user input
  - Responds by switching pages
- Medium-fidelity or High-fidelity in look & feel
- Low-fidelity in depth
  - The human operator in paper prototyping is aware of the algorithms

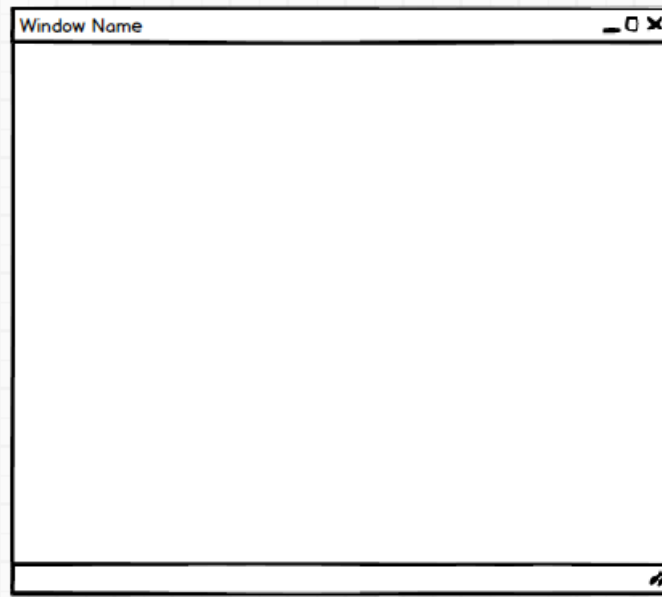
# Medium-fidelity

- Also known as “Mockups” or “Wireframe interface”
- Design of a single screen or a set of connected screens (following a task)
- “Wavy” or “imprecise” drawing (inspired by hand drawing)
  - Want to convey the impression that the design is still preliminary
  - Black and white
- Usually static information (predefined pages, only)
- May suggest user device

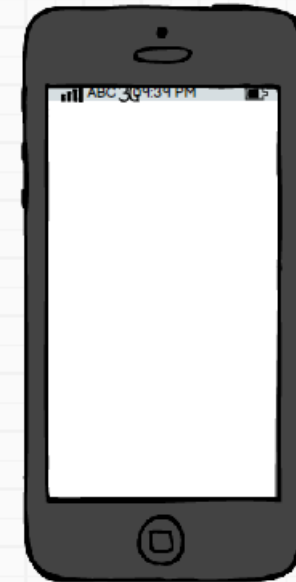
# Wireframes for the 3 interfaces



Web



Desktop



Mobile

moqups

Search stencils

Stencils Images

Horizontal Line Vertical Line

iPhone iPad

Are you sure? Okay NVM

iOS Alert iOS Picker

Settings Enabled Selected Sort Item

iOS Menu iOS Button

Quick Introduction to Moqups / Playground

Tweet 3,718

Moqzilla

http://moqups.com

Our Online store

Home Products About Contact

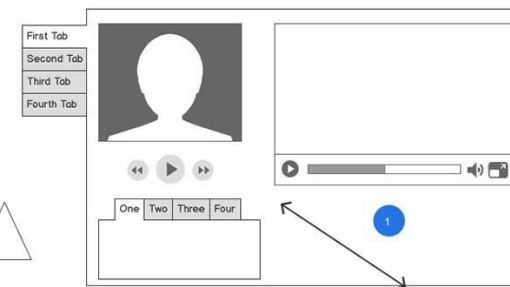
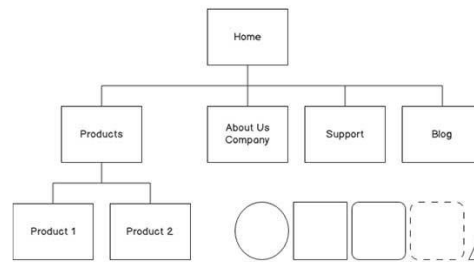
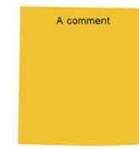
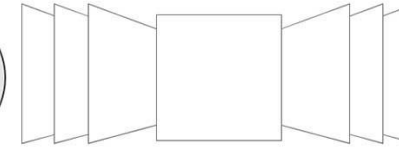
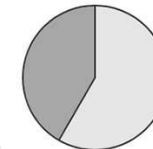
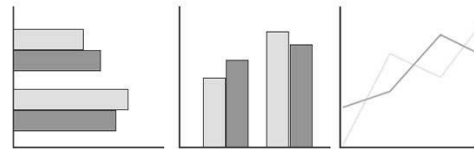
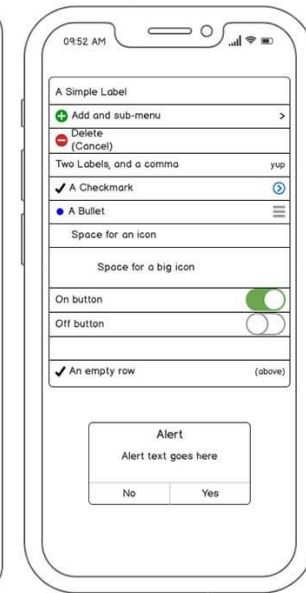
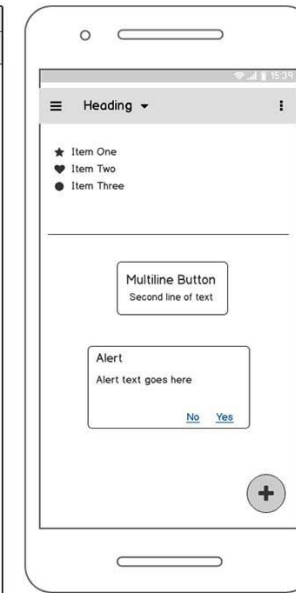
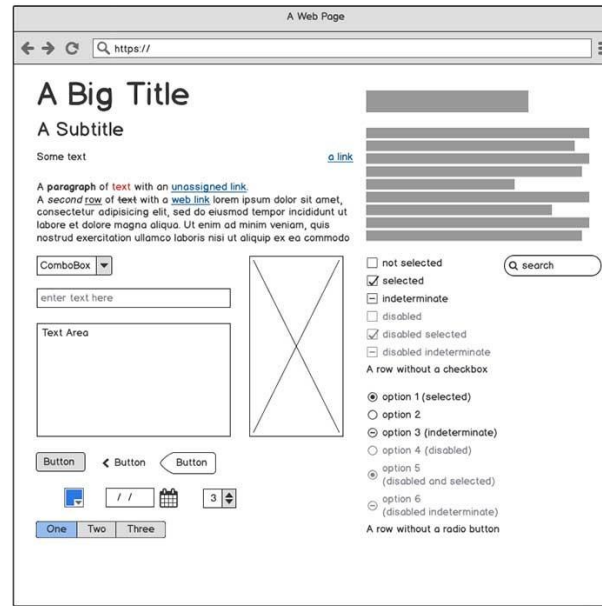
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus blandit elit tincidunt id.

Product rating: ♥♥♥♥♥

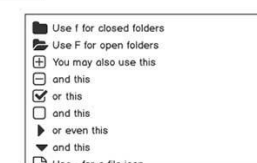
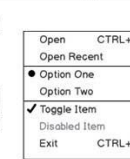
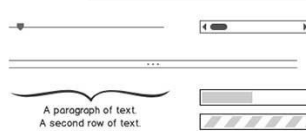
Add to cart

Some areas of the design may be "active" (link to a new page)

# UI Design libraries

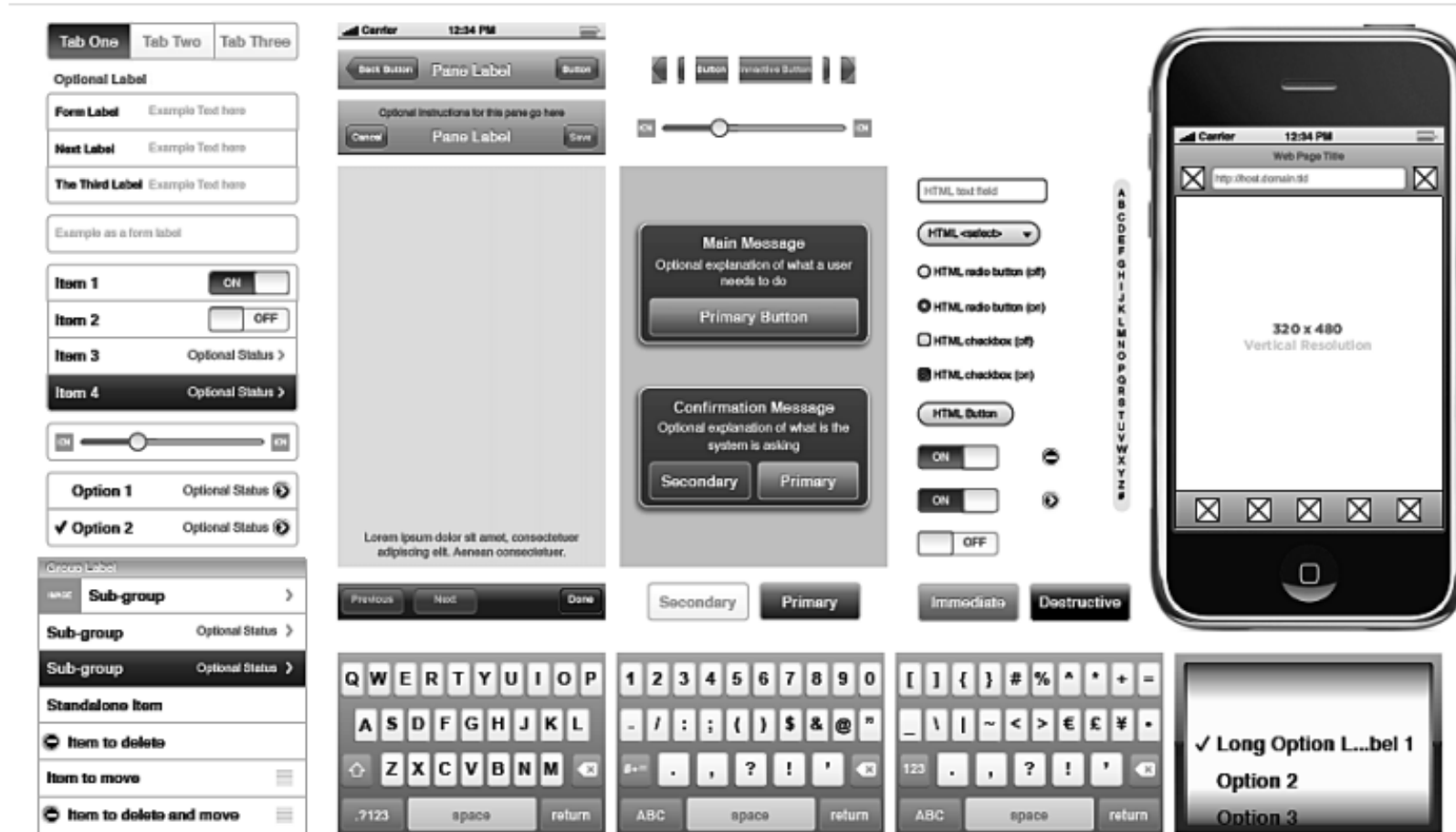


Name (job title)	Age	Nickname	Employee
Giacomo Gullizzoni Founder & CEO	40	Peldi	☑
Marco Bolton Tutore	38		☑
Mariah Macdoan Better Half	41	Patata	☑
Valerie Liberty Head Chef	35	Val	☑
<a href="#">Data Grid Docs</a>			☐





# Stencils for UI elements



# Some tools for wireframing



<https://balsamiq.com/wireframes/>  
<https://balsamiq.cloud/>



<https://moqups.com/>

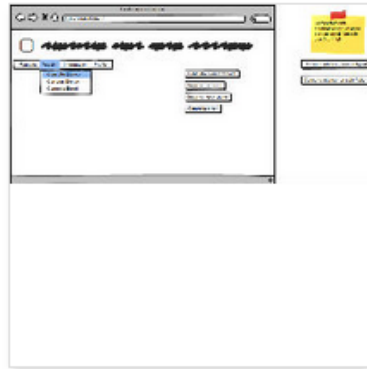


<https://www.mockplus.com/>

Mockingbird

<https://gomockingbird.com/>

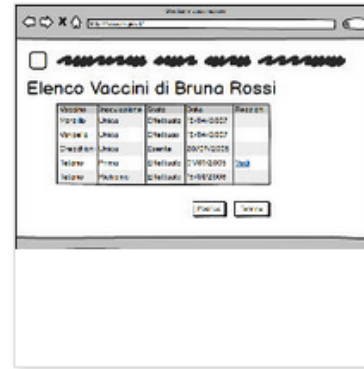
# Example



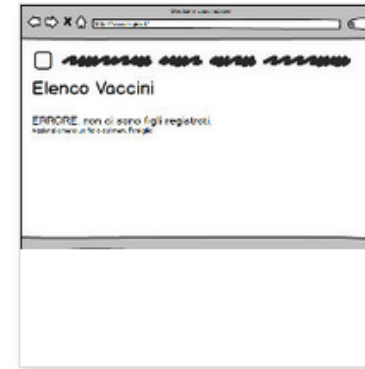
Step 1 e B-Step 1 ▾



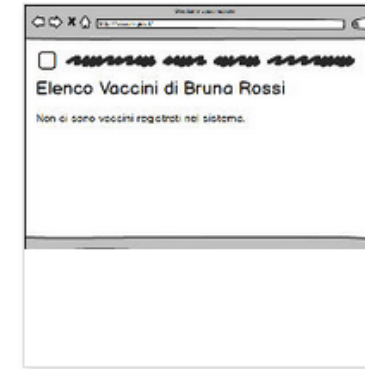
Step 2 3 ▾



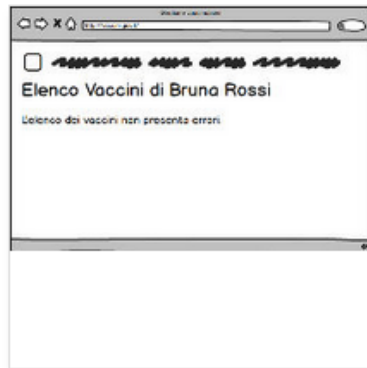
Step 4 5a ▾



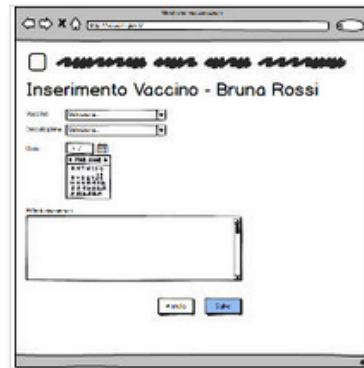
Step 2a ▾



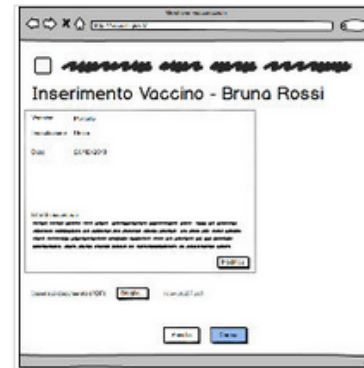
Step 4b ▾



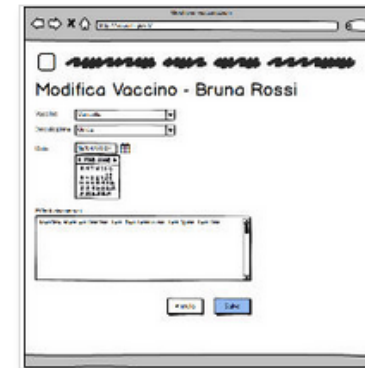
R-Step 6 ▾



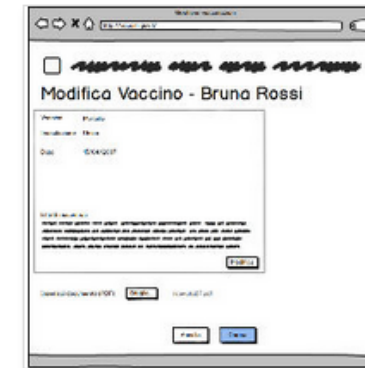
R-Step 5a1 5a2 ▾



R-Step 5a3 5a4 ▾



R-Step 5c1 5c2 ▾



R-Step 5c3 5c4 ▾

# Powerpoint-based Interactive mockups



# Wireframing tools: drawbacks

- Click, not interact
  - No text entry, no data entry, no real selection of listed data
  - Widgets aren't active
- Paths are static
- The tester is engaged in a “hunt for the hotspot”, to find the (few) only widgets that really clickable
  - Good for testing understanding of the UI and the workflow
  - Not good for testing the UI behavior

# Hi Fidelity Prototypes

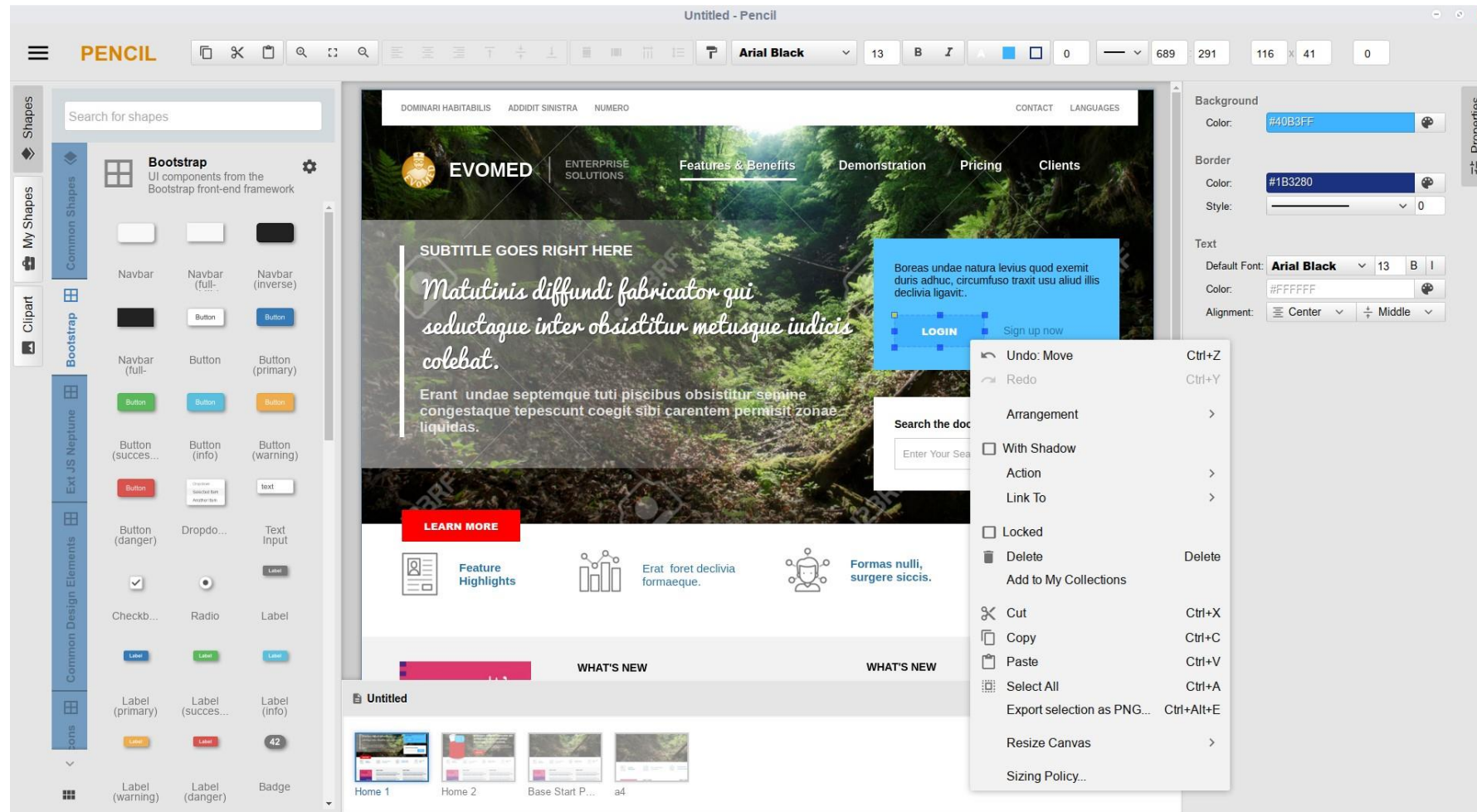
They look like the real thing. Widgets behave realistically. But it's still an illusion.

# Hi-Fi Prototypes (Digital Mock-ups)

- Actual computer application, with final-looking layout, colors, and graphics
  - May use design prototyping tools
  - May use real application code
- Much more expensive to build
- More time is spent with graphic design than interaction design

# High-fidelity computer prototypes

## Semi-interactive





# What can we learn from hi-fi interactive prototypes?

- Screen layout
  - Is it clear, overwhelming, distracting, complicated?
  - Can users find important elements?
- Colors, fonts, icons, other elements
  - Well-chosen?
- Interactive feedback
  - Do users notice & respond to status bar messages, cursor changes, other feedback
- Efficiency issues
  - Controls big enough? Too close together? Scrolling list is too long?

# Suggested video

- Prototyping fake it till you make it
- By Apple Design Team
- <https://youtu.be/3lqh-A5Jy4Q>

# Some tools for interactive hi-fi prototypes



<https://www.invisionapp.com/>



<https://www.figma.com>

**FROONT**

<https://froont.com/>

**webflow**

<https://webflow.com/>



<https://principleformac.com/>

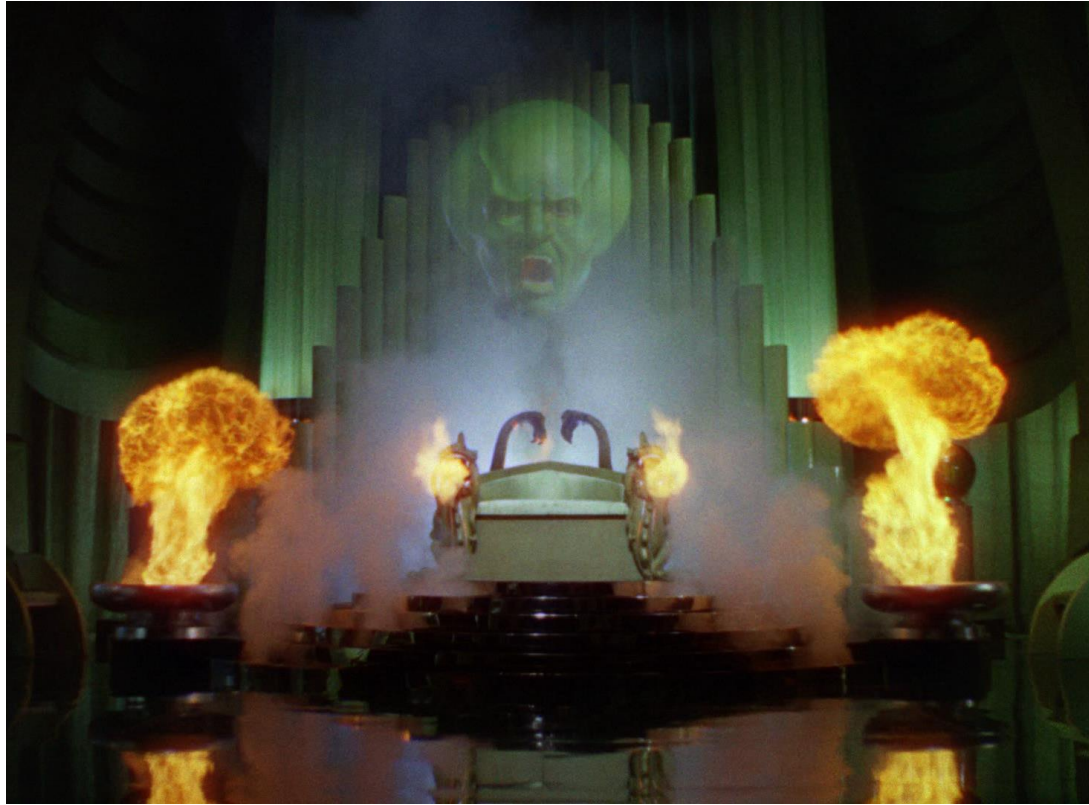
# Wizard-of-Oz techniques

Faking a technology, or filling-in for missing functionality

# Goal

- How to test an application that is really complete...
  - With finalized user interface
  - With finalized algorithms
  - Also including stuff that we still aren't able to implement
- ...but without actually writing the code
  - Except for a semi-interactive 'dumb' prototype

# Remember the main behind the curtain?



# Remember the Mechanical Turk?

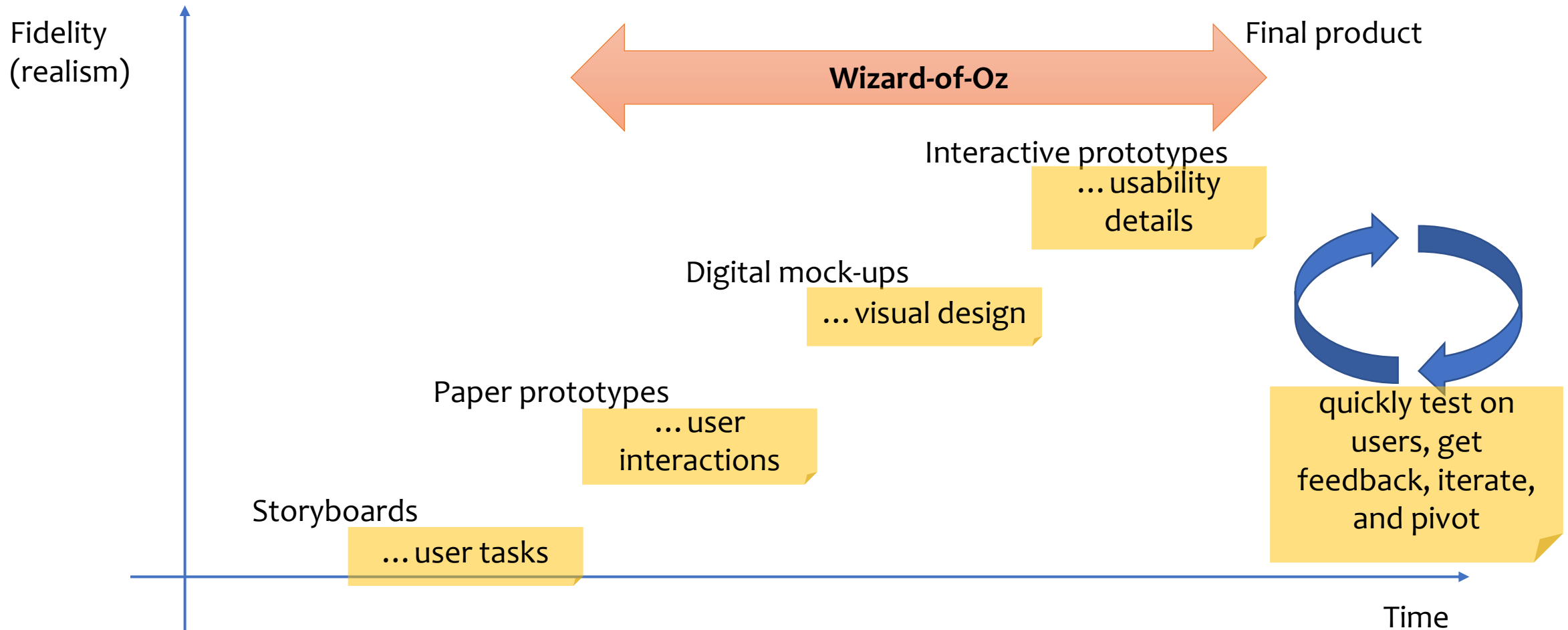


# Wizard-of-Oz

- Software simulation with a human in the loop to help
- “Wizard of Oz” = “man behind the curtain”
  - Simulates the machine behavior with a human operator
  - Wizard is usually but not always hidden
- Often used to simulate future technology
  - Speech recognition
  - Learning
- Wizard may be hidden or visible
  - Must always be revealed, at least at the end



# Prototypes facilitate conversations about...



# Implementing a Wizard-of-Oz prototype

- Choose supported tasks and scenarios
- Create User Interface mock-ups
  - Implement a part of the system
  - Leave “hooks” for the Wizard’s actions
- Implement a back-office interface for the Wizard
- Define “rules of behavior” for the Wizard
  - When he should respond
  - How it should respond (the “algorithm”)

# Benefits

- Faster and cheaper than most interactive prototypes
- More “real” than paper prototyping
- Creating multiple variations is easy
- Identifies bugs and issues with current design
- Can envision applications that are difficult to build
- Playing wizard allows a better understanding of algorithmic requirements

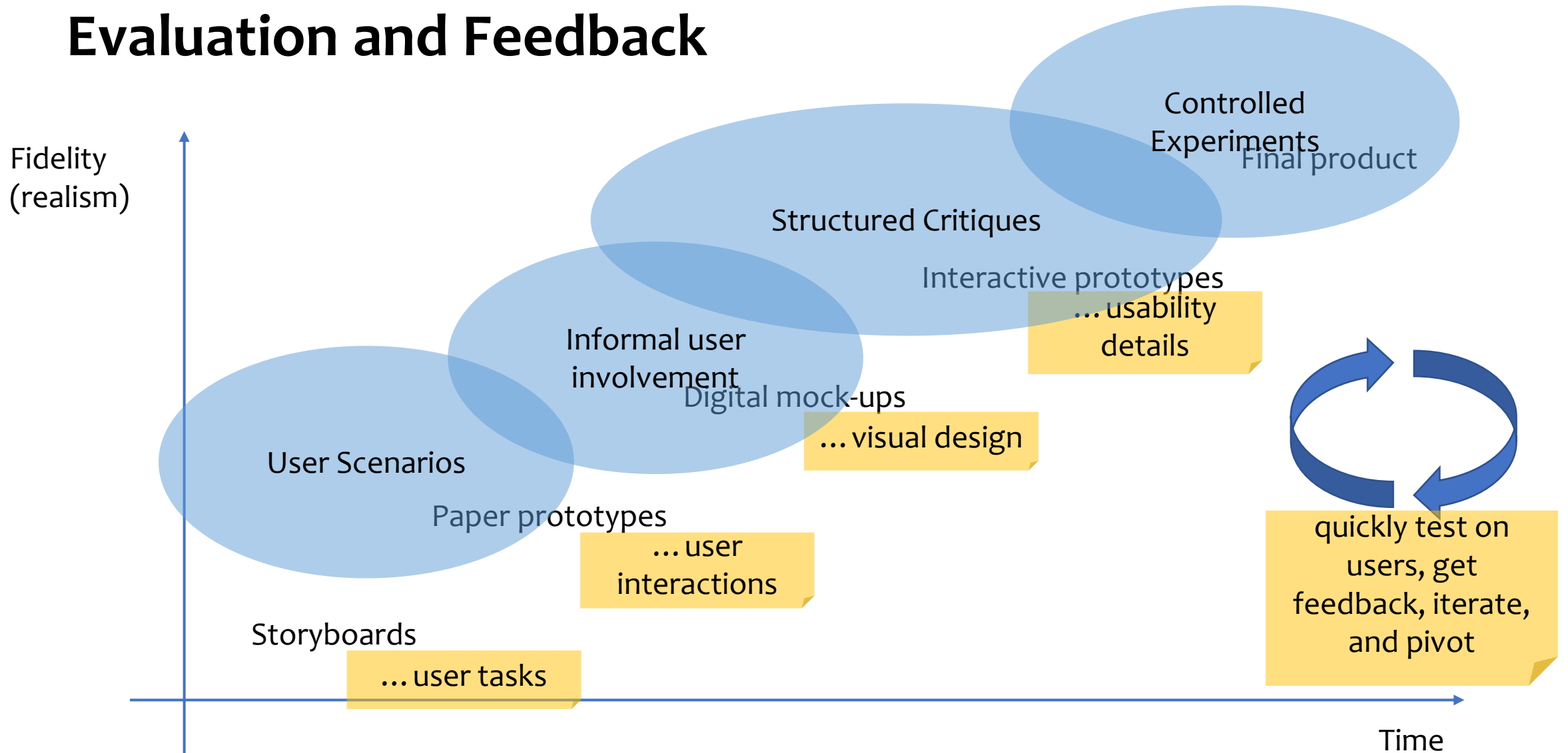
# Risks

- May be over-optimistic
  - Speech recognition that always works (instead of having an error rate)
  - Super-intelligence (that will never exist)
- Wizard behavior is difficult
  - Take into account system limitations
  - Emulate expected system response
  - Within acceptable timing
- Needs at least 2 researchers

# Wrap-up

Many different techniques, applicable to different goals and contexts

# Evaluation and Feedback



# References

- Google, Begin Today With Rapid prototyping, [https://www.youtube.com/playlist?list=PL9KVIdeJ2K8NDpsiyYpcbB\\_qifd3y5CYZ](https://www.youtube.com/playlist?list=PL9KVIdeJ2K8NDpsiyYpcbB_qifd3y5CYZ)
- MIT, [http://web.mit.edu/6.813/www/sp18/classes/11-prototyping/#reading\\_11\\_prototyping](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>



# 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/>

