

Interface Prototyping

Human Factors

Why prototype?

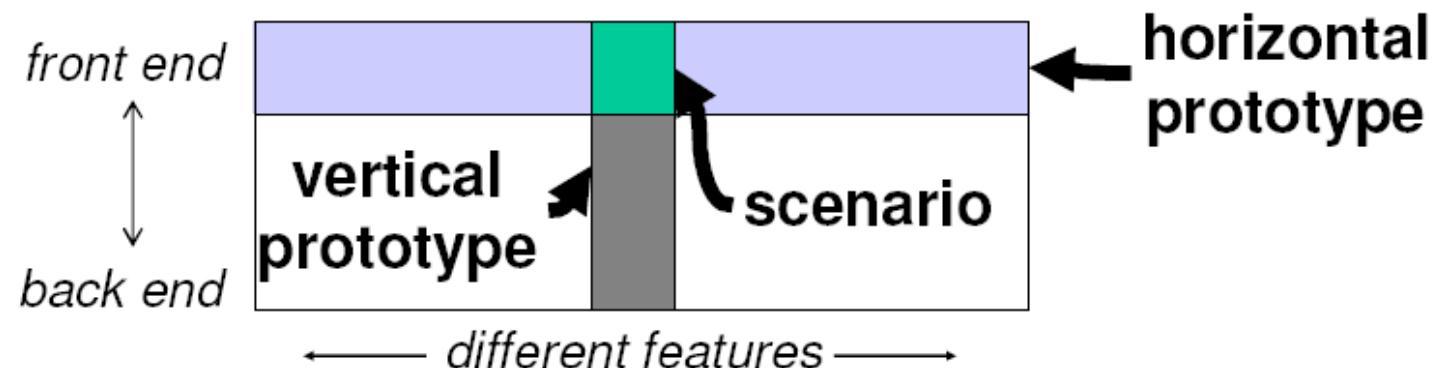
- Evaluation and feedback are central to interaction design
- Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing
- Team members can communicate effectively
- You can test out ideas for yourself
- Prototypes answer questions, and support designers in choosing between alternatives

Why prototype?

- Get feedback earlier, cheaper, much faster to build.
- Experiment with alternatives with multiple prototypes.
- Easier to change or **throw away** when we discover problems, more malleable.
 - Writing and debugging a lot of code creates a psychological sense of commitment which is hard to break. You don't want to throw away something you've worked hard on, even if it really should be scrapped.

Fidelity

- Breadth: % of feature set covered
 - Only enough features for certain tasks
- Depth: degree of functionality for each feature
 - Limited choices, canned responses, no error handling



What is a prototype?

In interaction design it can be (among other things):

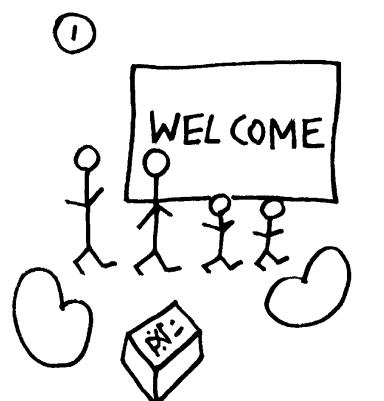
- a series of screen sketches
- a storyboard, i.e. a cartoon-like series of scenes
- a Powerpoint slide show
- a video simulating the use of a system
- a lump of wood (e.g. PalmPilot)
- a cardboard mock-up
- a piece of software with limited functionality written in the target language or in another language

What to prototype?

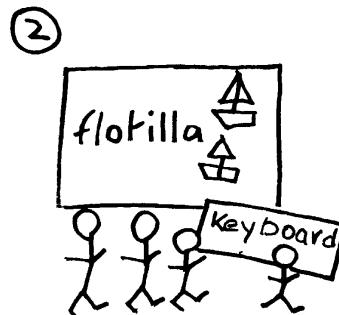
- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas



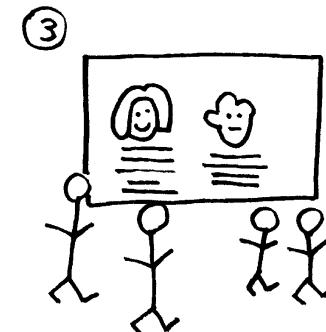
Generate storyboard from scenario



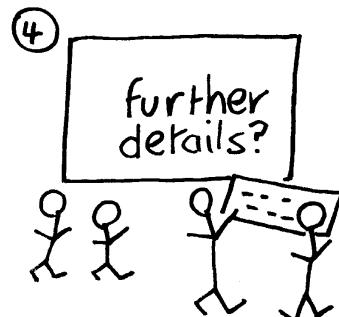
Thomson family gather around



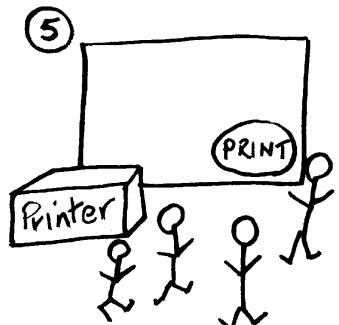
System suggests flotilla



System shows descriptions



System asks for details



Summary printed

Low-fidelity Prototyping

- Uses a medium which is unlike the final medium, e.g. paper, cardboard
- Is quick, cheap and easily changed
- Examples:
 - sketches of screens, task sequences, etc
 - ‘Post-it’ notes
 - storyboards
 - ‘Wizard-of-Oz’

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 (* handhelds?)
- A person simulates the computers 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

Why paper prototype?

- 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)
 - Architecture sketches
- Focuses attention on big picture
 - Designer doesn't waste time on details
 - Customer makes more creative suggestions, not nitpicking
- Nonprogrammers can help

The Team

- Roles for design team:
- Computer
 - 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, sits on hands if necessary
 - Takes copious notes

What you can learn from a paper prototype

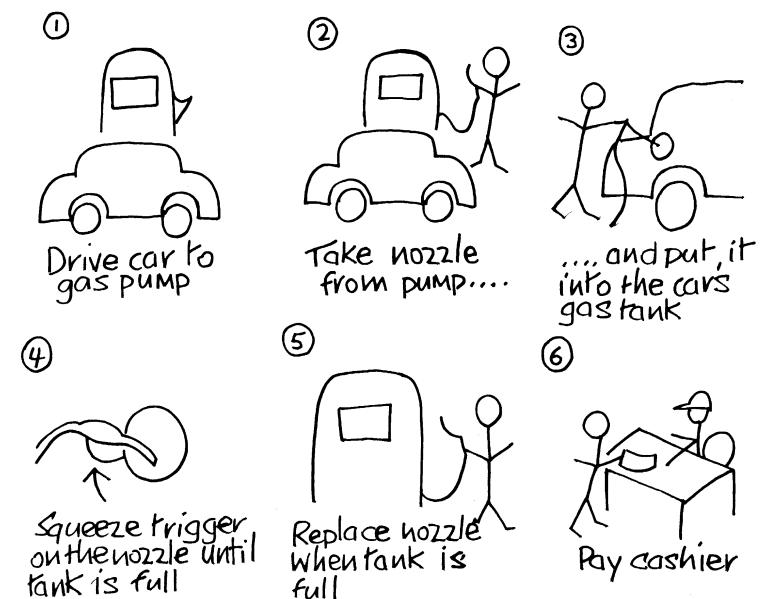
- Conceptual model
 - Do users understand it?
- Functionality
 - Does it do what is needed? Missing features?
- Navigation & task flow
 - Can users find their way around?
 - Terminology
 - Do users understand labels?
- Screen contents
 - What needs to go on the screen?

What you won't learn

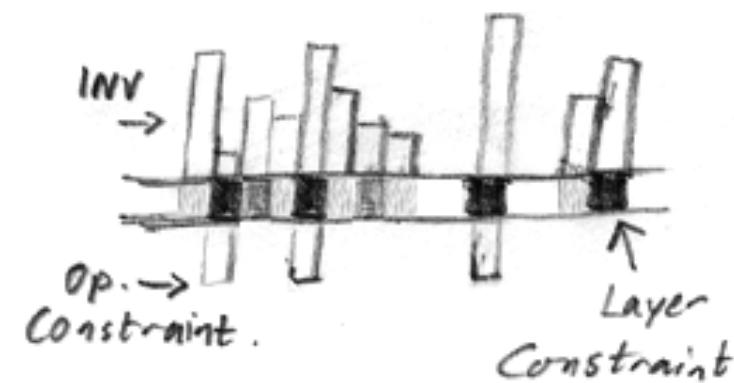
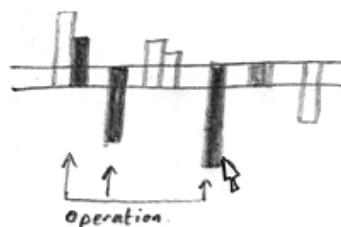
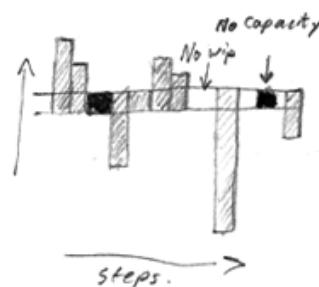
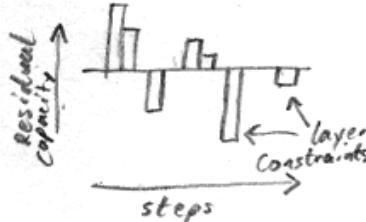
- Look: colour, font, whitespace, etc
- Feel: Fitts' Law issues
- Response time
- Are small changes noticed?
 - Even the tiniest change to a paper prototype is clearly visible to user (change blindness)
- Exploration vs. deliberation
 - Users are more deliberate with a paper prototype; they don't explore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

Storyboards

- Often used with scenarios, bringing more detail, and a chance to role play
- It is a series of sketches showing how a user might progress through a task using the device
- Used early in design

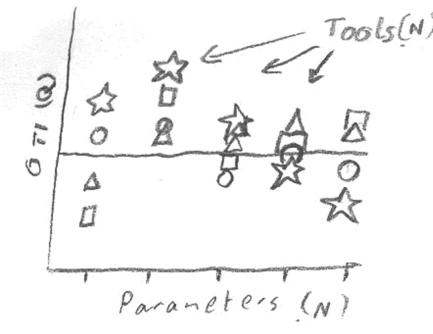


Sketching components

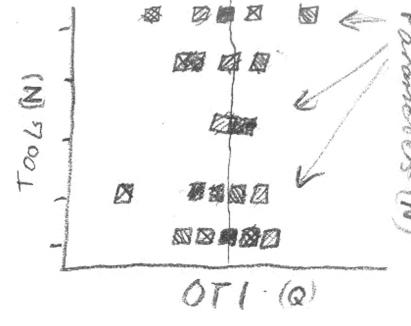


- Sketching is important to low-fidelity prototyping
- Don't be inhibited about drawing ability. Practice simple symbols

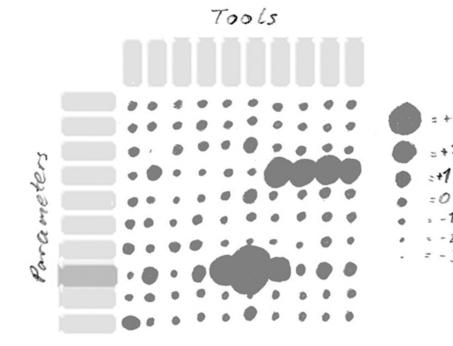
Sketching options



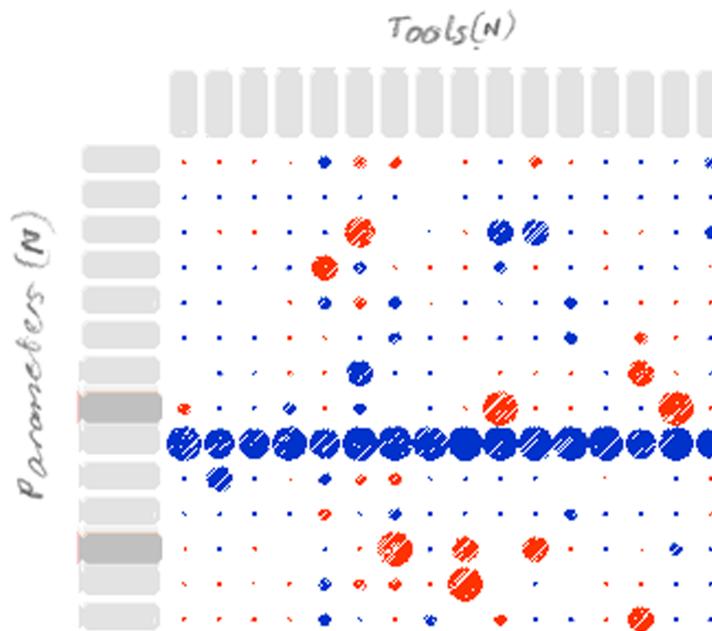
(a)



(b)



(c)



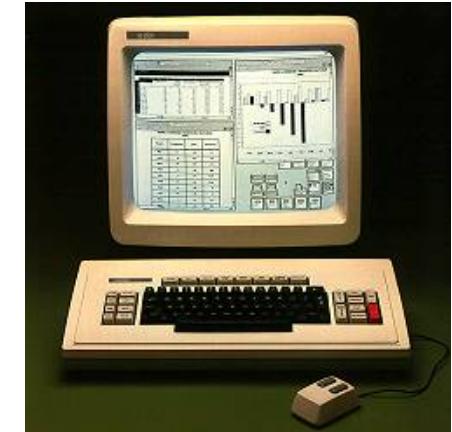
Conceptual design: from requirements to design

- Transform user requirements/needs into a conceptual model

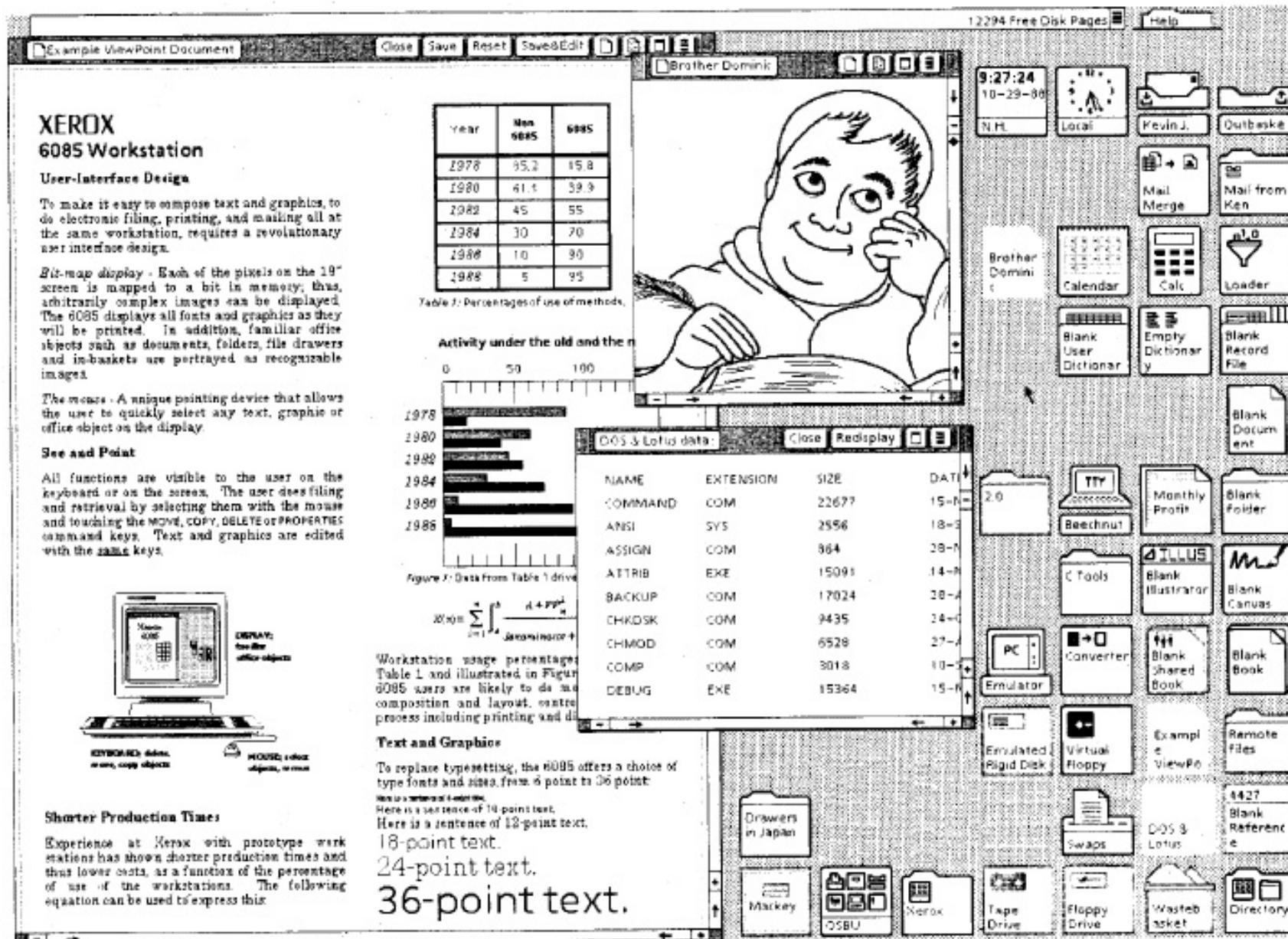
“a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended”
- Don’t move to a solution too quickly. Iterate, iterate, iterate
- Consider alternatives: prototyping helps.
- Interface metaphor and interaction style

Xerox Star

- Xerox Star interface hugely influential, used a conceptual model which users found easy to understand.
- Paper, folders, filing cabinets and mailboxes represented as icons on screen, possessing some of the properties of their physical counterparts.
- Dragging across the screen equivalent (in some sense) to physically moving piece of paper.
- Some parts of the metaphor are logical and consistent but not based on physical equivalent - e.g. dragging files onto printer to print.



Xerox Star



Expanding the conceptual model

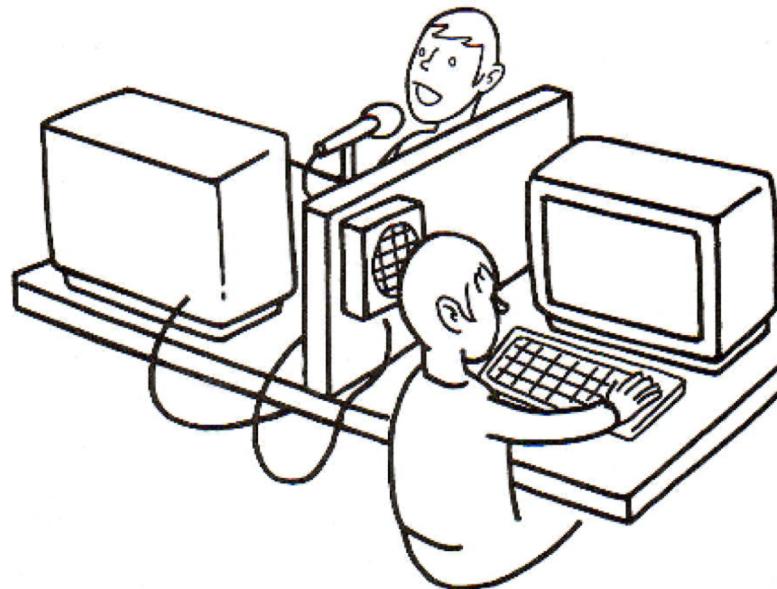
- What functions will the product perform?
What will the product do and what will the human do (task allocation)?
- How are the functions related to each other?
Sequential or parallel?
Categorisations, e.g. all actions related to telephone memory storage
- What information needs to be available?
What data is required to perform the task?
How is this data to be transformed by the system?

Using scenarios in conceptual design

- Express proposed or imagined situations
- Used throughout design in various ways
 - scripts for user evaluation of prototypes
 - concrete examples of tasks
 - as a means of co-operation across professional boundaries
- Plus and minus scenarios to explore extreme cases

‘Wizard-of-Oz’ prototyping

- The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.
- Usually done early in design to understand users' expectations.



Buxton, 2007

Wizard of Oz prototyping

- Software simulation with a human in the loop to help
- Wizard of Oz = man behind the curtain
 - Wizard is usually but not always hidden
- Often used to simulate future technology
 - Speech recognition
 - Learning
 - Intelligent contextual behaviour
- Issues
 - Two UIs to worry about: users and wizards
 - Wizard has to act mechanically - more “intelligent” applications.

High-fidelity prototyping

- Uses materials that you would expect to be in the final product.
- Prototype looks more like the final system than a low-fidelity version.
- Full graphical mockups with professional drawing tools, interactive prototyping tools.
- Evolutionary prototyping vs. ‘Throw-away’ prototyping
- Danger that users/managers think they have a full system.....

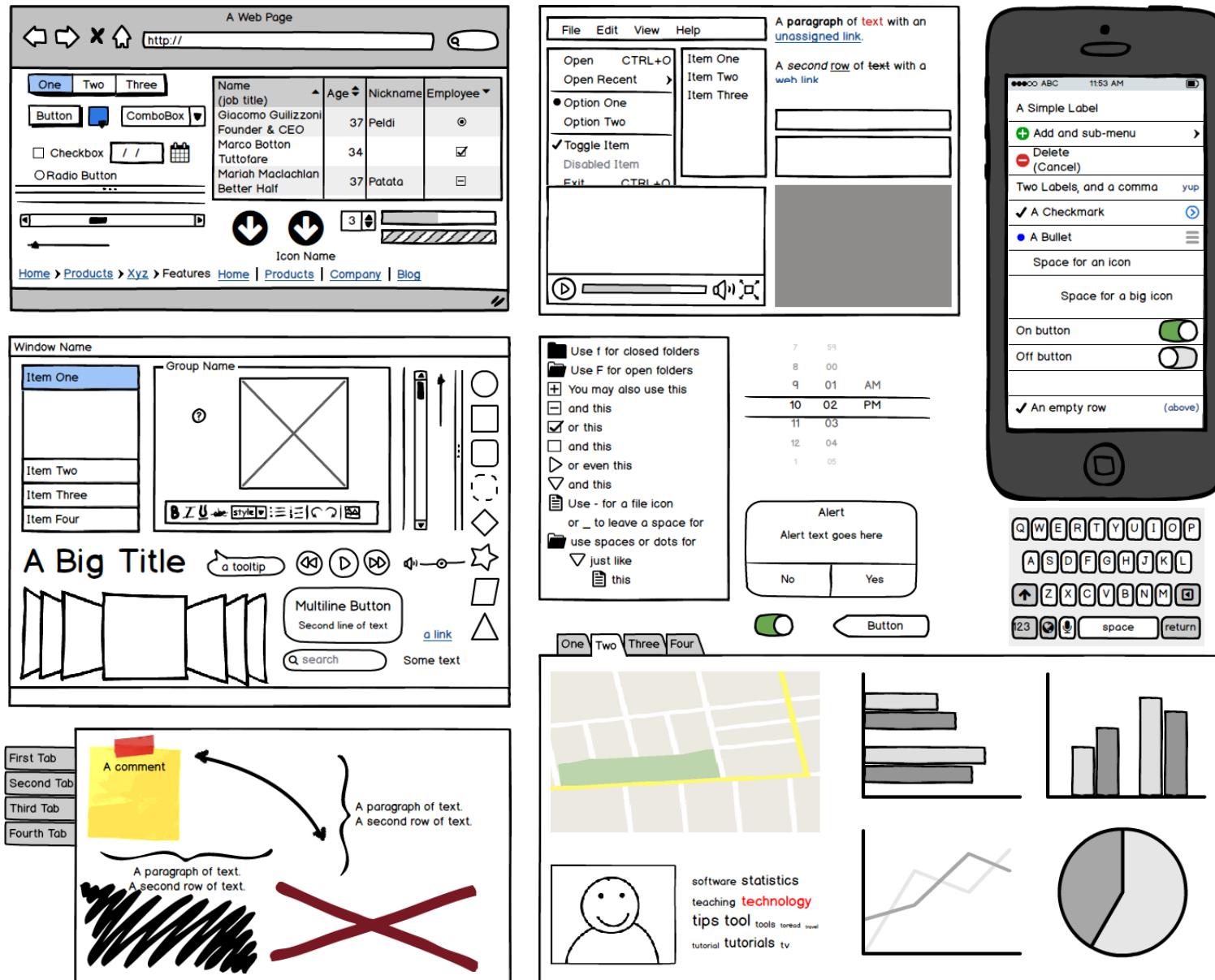
Cardboard prototype



High fidelity prototype



Prototyping tools



Summary

- Different kinds of prototyping are used for different purposes and at different stages
- Prototypes answer questions, so prototype appropriately
- Construction: the final product must be engineered appropriately
- Conceptual design (the first step of design)
- Consider interaction types and interface types to prompt creativity
- Storyboards can be generated from scenarios
- Card-based prototypes can be generated from use cases