Design Thinking & Prototyping

Ahmed Sabbir Arif
University of California, Merced
https://www.theiilab.com/

Lecture notes in this series are based on

- Ahmed Sabbir Arif. 2021. Statistical Grounding. *Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice*, ACM
- Ann Blandford, Dominic Furniss, Stephann Makri. 2016. Qualitative HCI Research: Going Behind the Scenes. Morgan & Claypool
- Jonathan Lazar, Jinjuan Feng, Harry Hochheiser. 2017. Research Methods in Human-Computer Interaction. Morgan Kaufmann
- I. Scott MacKenzie. 2013. Human-Computer Interaction: An Empirical Research Perspective, Morgan Kaufmann
- Interaction Design Foundation. 2022. Design Thinking
- Lecture notes of Amy Bruckman, Mark Dunlop, Niels Henze, I. Scott MacKenzie, Laura Moody, Albrecht Schmidt, Kami Vaniea

inclusive interaction lab

1

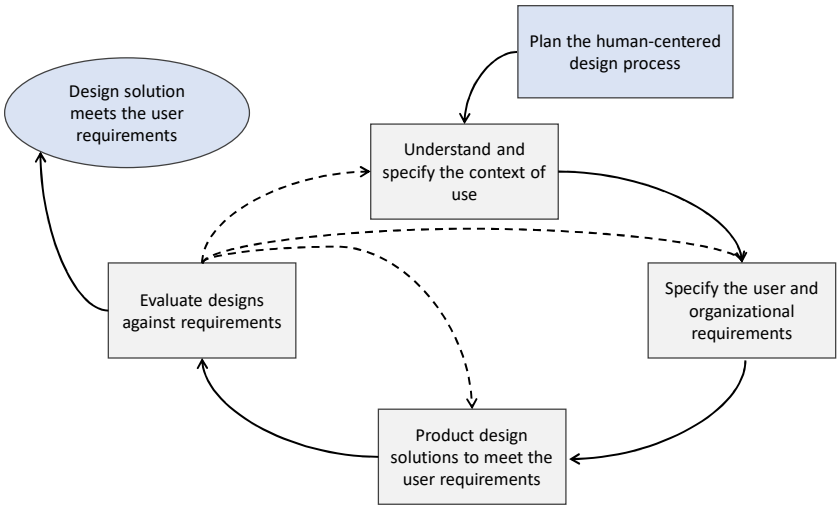
Design Thinking

Human-Centered Design

inclusive interaction lab

2

Human-Centered Design

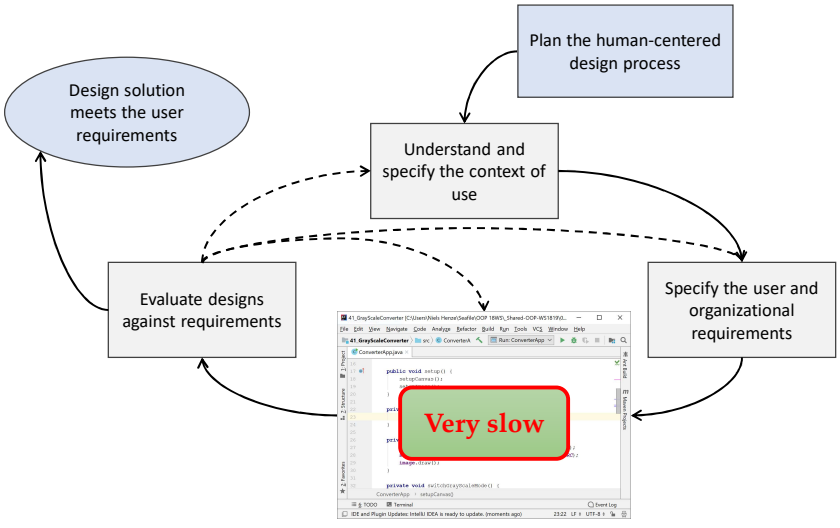


ISO 9241-210:2019 Ergonomics of Human-System Interaction

inclusive interaction lab

3

Human-Centered Design

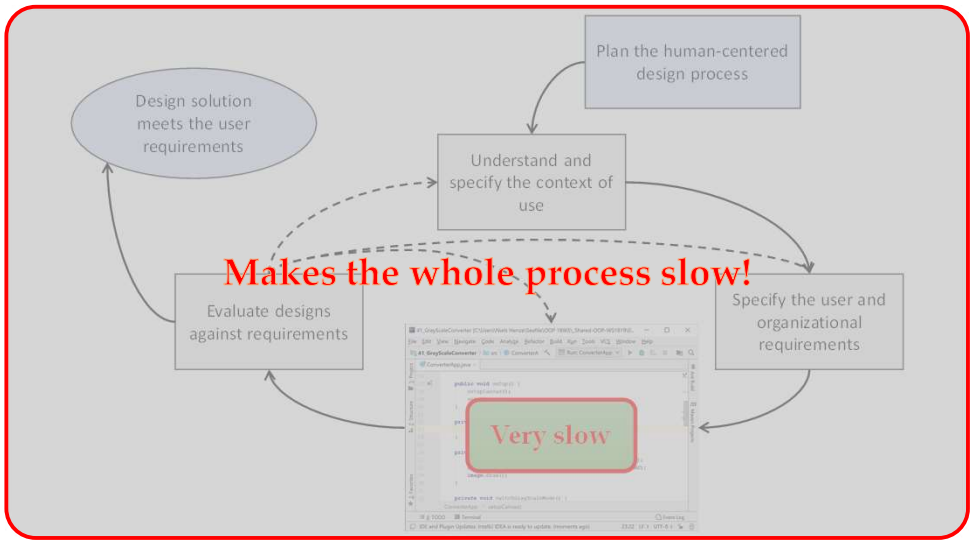


ISO 9241-210:2019 Ergonomics of Human-System Interaction

inclusive interaction lab

4

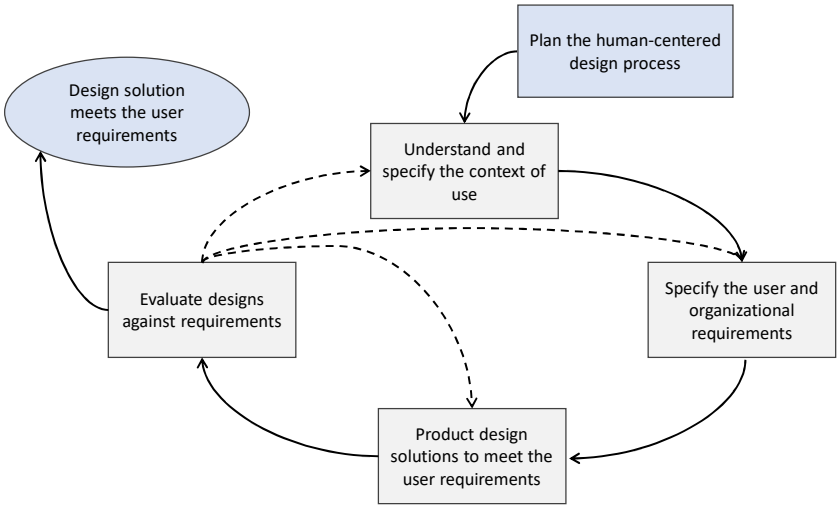
Human-Centered Design



ISO 9241-210:2019 Ergonomics of Human-System Interaction

5

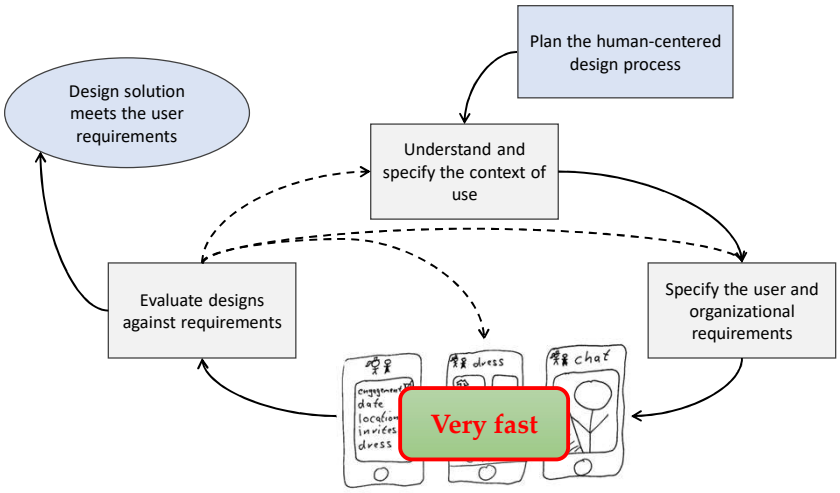
Human-Centered Design



ISO 9241-210:2019 Ergonomics of Human-System Interaction

6

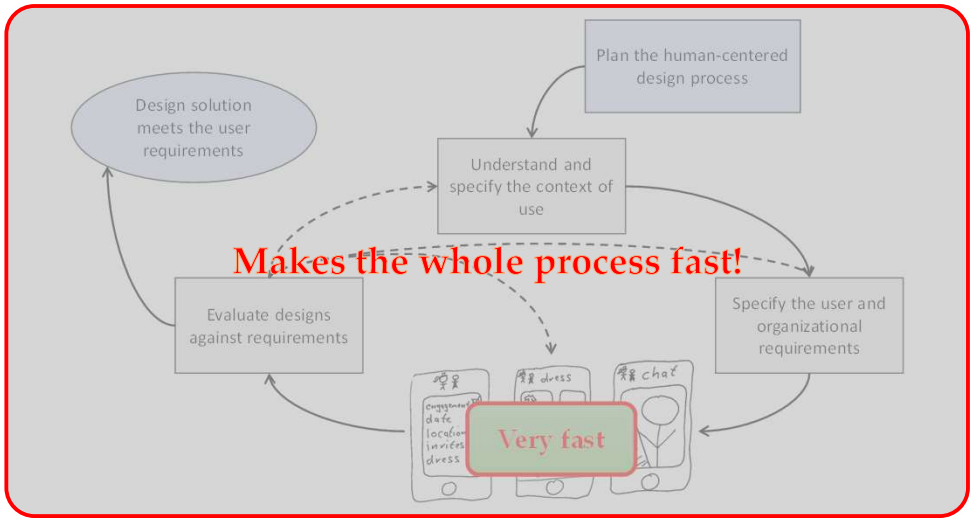
Human-Centered Design



ISO 9241-210:2019 Ergonomics of Human-System Interaction

7

Human-Centered Design



ISO 9241-210:2019 Ergonomics of Human-System Interaction

8

Design Thinking

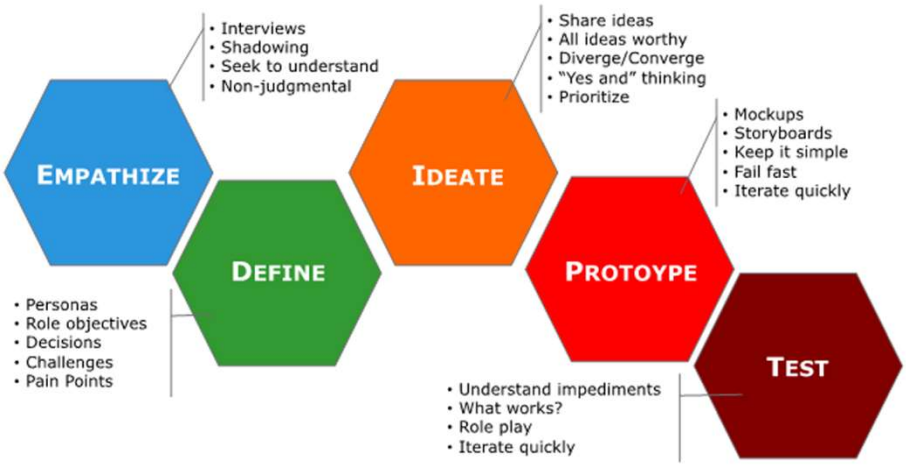
- Design thinking is a non-linear, iterative process that teams use to understand:
  - Users
  - Challenge assumptions
  - Redefine problems
  - Create innovative solutions to prototype and test
- Most useful to tackle problems that are ill-defined or unknown (wicked problems?)
- Used by professionals in a variety of fields

- First mentioned by Herbert A. Simon in his book: *The Sciences of the Artificial (1969)*

Design Thinking: Phases

- Not always sequential, you can run them:
  - In parallel
  - Out of order
  - Repeat them in an iterative fashion
- The Hasso Plattner Institute of Design at Stanford (the d.school)

Design Thinking: Phases

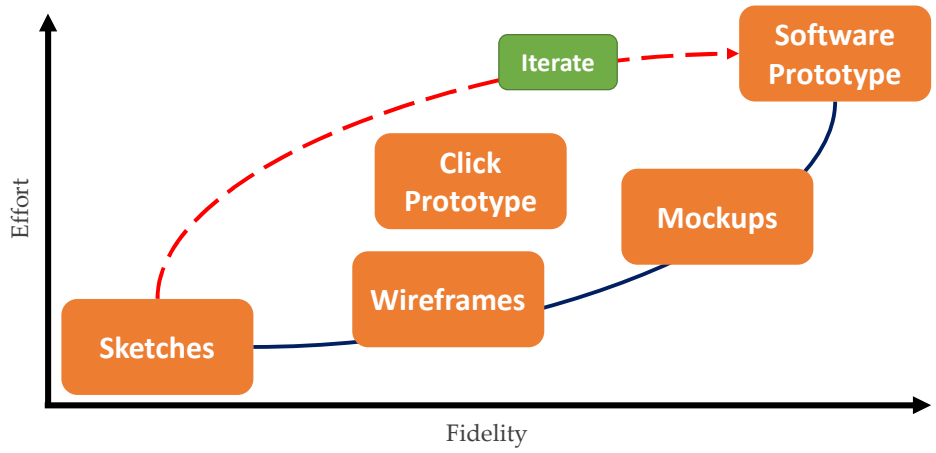


11

Prototyping

12

Prototyping: Effort vs. Fidelity

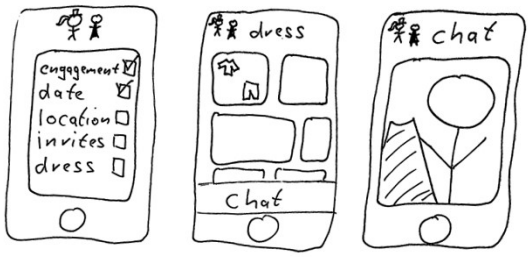


13

13

Prototyping: Sketches

- Explores design ideas
- Communicates interface design
- Used for early user tests



Video by Judith Amores, https://vimeo.com/76766231

14

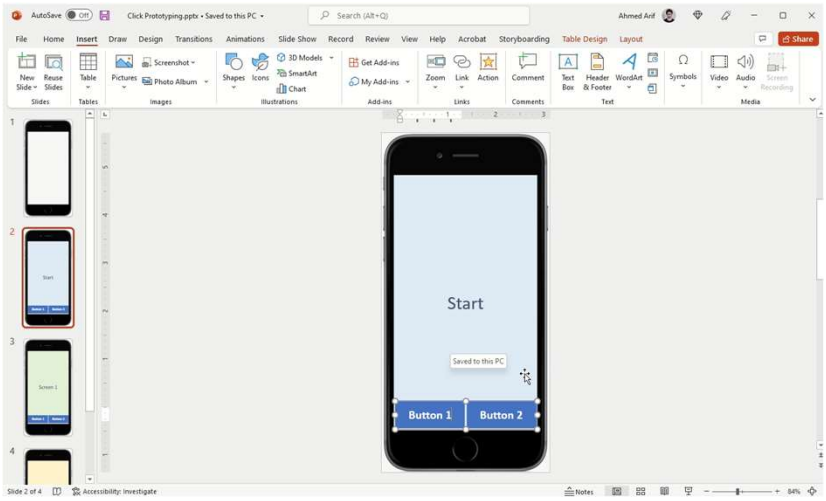
14

Prototyping: Templates Drafting Ruler

- Reusable, durable
- Can be faster than sketching
- Looks & feels more professional
- Reflects the effort put into design:
  - Can encourage users to take an evaluation (user study) seriously

Betsy Weber, https://www.flickr.com/photos/betsyweber/5516971798

Prototyping: Click Prototype



Prototyping: Click Prototype

inclusive interaction lab

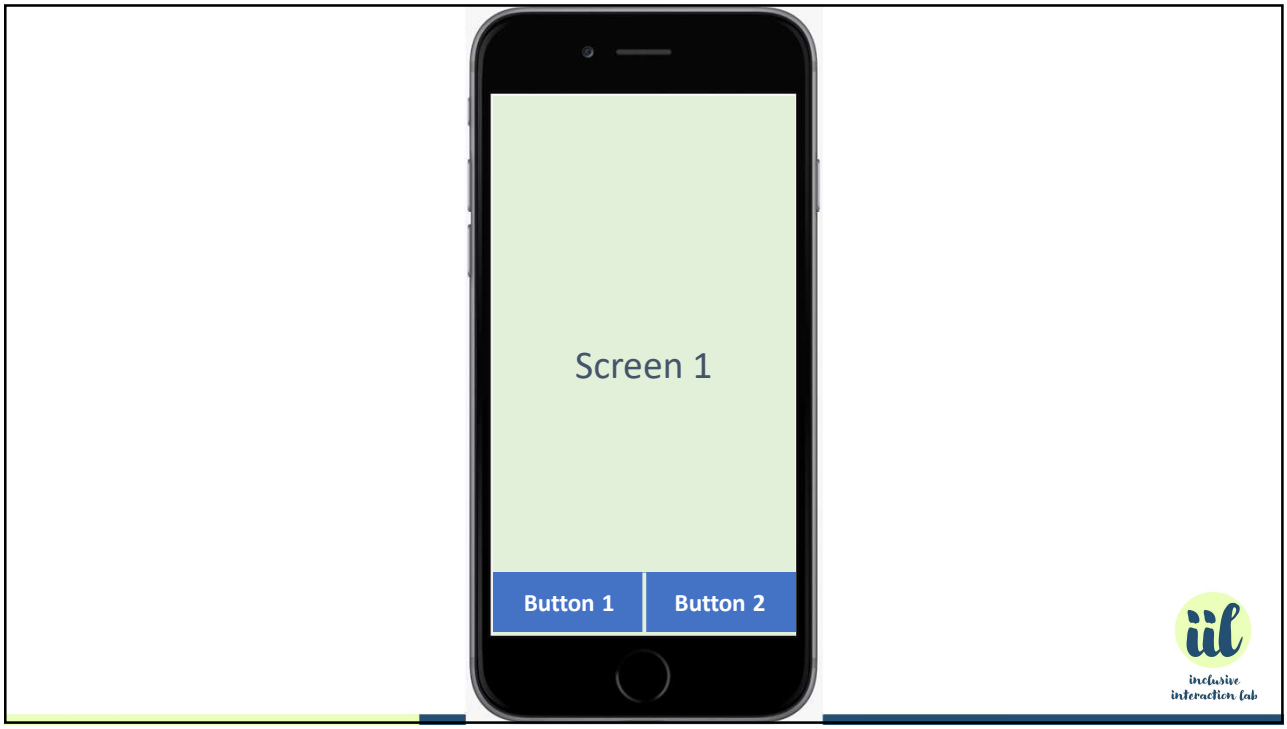
17

inclusive interaction lab

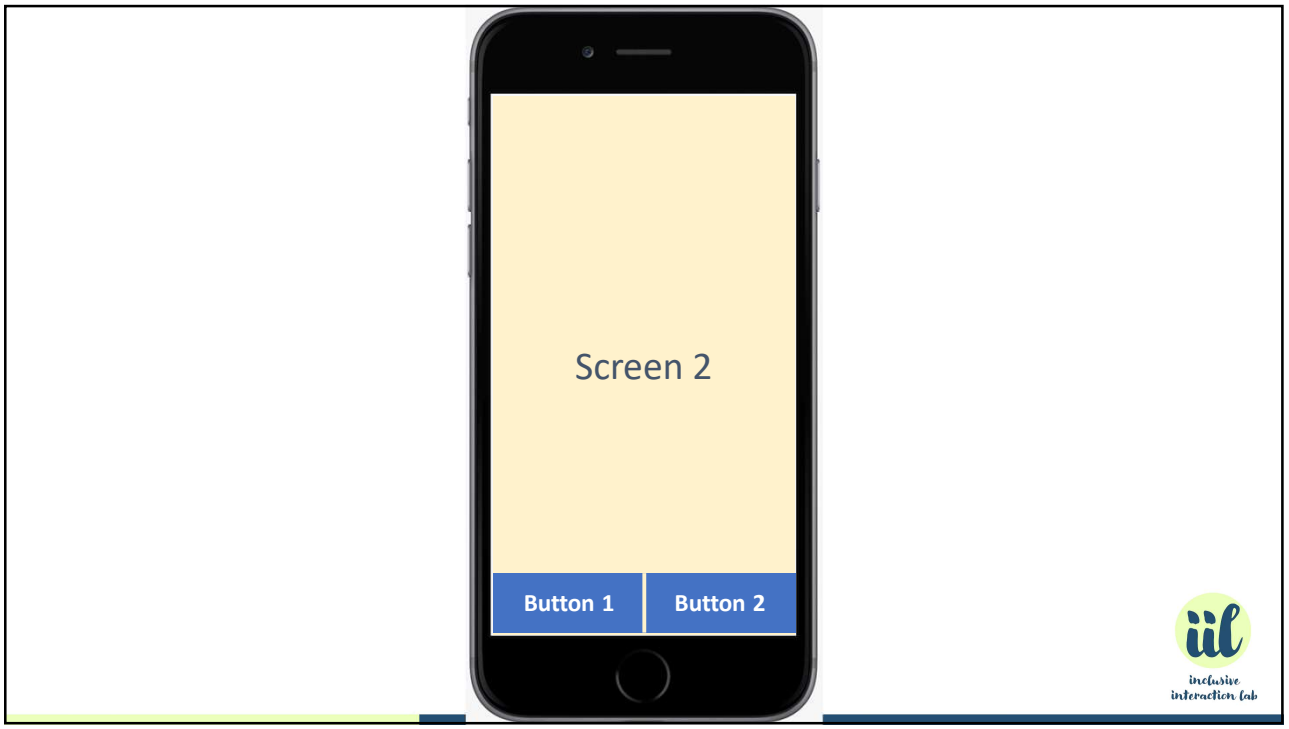
18


Start
Button 1
Button 2
inclusive interaction lab

19


Screen 1
Button 1
Button 2
inclusive interaction lab

20



21

Prototyping: Wireframe

- Used to arrange the content logically and within the browser window
- To map out the space we plan to use and determine in advance how we plan to use it
- There are no colours and few graphics in order to keep the focus on the content rather than the visual design

- Remember the lecture on eye tracking

22

inclusive interaction lab

22

Prototyping: Wireframe

23

inclusive interaction lab

23

Prototyping: Wireframe

24

inclusive interaction lab

24

Prototypes: Taxonomy

- Low-fidelity vs. high-fidelity prototypes
- Horizontal vs. vertical prototypes
- Non-functional vs. functional prototypes
- Throwaway vs. evolutionary prototypes

25

Low-Fidelity vs. High-Fidelity

26

Low-Fidelity Prototypes

- Check ideas and interaction flow
- Fast, cheap and easy to change

- Advantages:
  - Cheap, easy and quick to implement
  - Users are keen to criticise
- Disadvantages:
  - No real functionality, difficult to identify errors
  - Reuse and extending difficult to impossible
  - Not all ideas can be realized

High-Fidelity Prototypes

- Looks & feels like the final product
  - Colors, screen layout, fonts, … text used
  - Response time and interactive behavior

- Restricted functionality
  - Only certain functions work
  - Functionality is targeted towards the tasks
  - Invisible issues, e.g., security
- Standard technologies for prototyping
  - HTML, JavaScript
  - Axure, Director, Presentation programs
  - GUI Builder, e.g., Visual Basic, Delphi, NetBeans

Horizontal vs. Vertical Prototypes

29

Horizontal Prototyping

- Demonstrates the feature spectrum without implementing them
- Helps to evaluate/test
  - Navigation, e.g., finding a specific function or feature
  - Overall user interface concept
  - Feature placement
  - Accessibility
  - User preferences
- Applicable in low-fidelity prototyping and high-fidelity prototyping
- Used in early design stages
  - To determine the set of features to include
  - To decide on the user interface concept

30

Vertical Prototyping

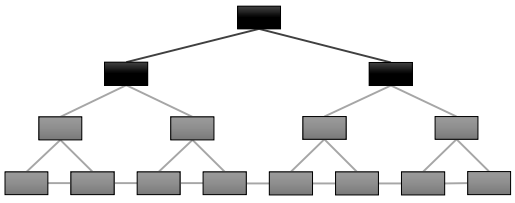
- Demonstrate a specific feature
- The details of the function/feature are shown/implemented

- Helps to evaluate/test:
  - The optimal design for a particular function
  - Optimize the usability of this function
  - User performance for this specific function

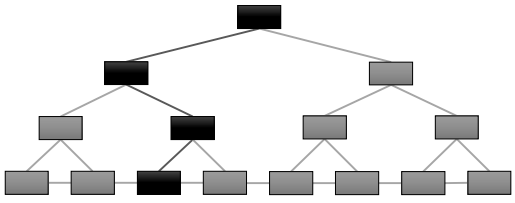
- Often used in high-fi prototyping but is also applicable to low-fi

Horizontal vs. Vertical Prototypes

Horizontal Prototype



Vertical Prototype



Throwaway vs. Evolutionary Prototypes

33

Throwaway vs. Evolutionary Prototypes

- Throwaway prototypes are developed from the initial requirements but not used for the final product
  - Built acknowledging all the requirements
  - It enables quick prototyping and throwing the prototype away
- Evolutionary prototypes are robust and constantly improved
  - Built only with well understood requirements

34

Non-functional vs. Functional Prototypes

35

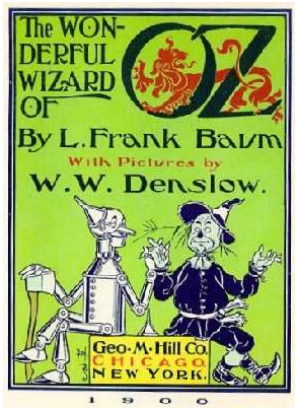
Non-functional vs. Functional Prototypes

- Functional prototypes imitate the functions of the actual product as closely as possible
  - Usually produced for products that are dependent on the function rather than the display
- Non-functional prototypes are low-fidelity prototypes, where users have to guess the results of an action/interaction

36

Wizard-of-Oz (WOz)

- An invisible "wizard" controlling parts of the functionality
- We implement the easy parts, leave the hard part to human operator
- Gives users with the experience without extensive implementation effort for the prototype
- *"Pay no attention to that man behind the curtain ..."* takes on a new meaning when the curtain is replaced by a computer terminal
- Typical areas: recognizers
- Important to account for the wizard's error



37

37

Wizard-of-Oz (WOz): Example

Ahmed Sabbir Arif, Benedikt Iltisberger, Wolfgang Stuerzlinger. 2011. Extending Mobile User Ambient Awareness for Nomadic Text Entry. In Proceedings of the 23rd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction (OzCHI 2011). ACM, New York, NY, USA, 21-30.

38

38