

From data to design (part 2)

«UCD: User-Centered Software Development»

Prof. Dr. Cl. Müller-Birn, Institute for Computer Science, HCC.lab

May 11, 2015

From data to design (part 2)

Introduction

Context for today

Last week

Primary noun analysis

Define the underlying structure of your software (how to guide users through the interface)

- For content-centered applications: information architecture
- For task-centered applications: interaction design

This week

Interaction design

- Specifying a flow chart and creating a storyboard

Interface design

- Learning about important design patterns

Bringing interaction and interface design together

- Using low-fi prototype

How to define your first design?

Outline

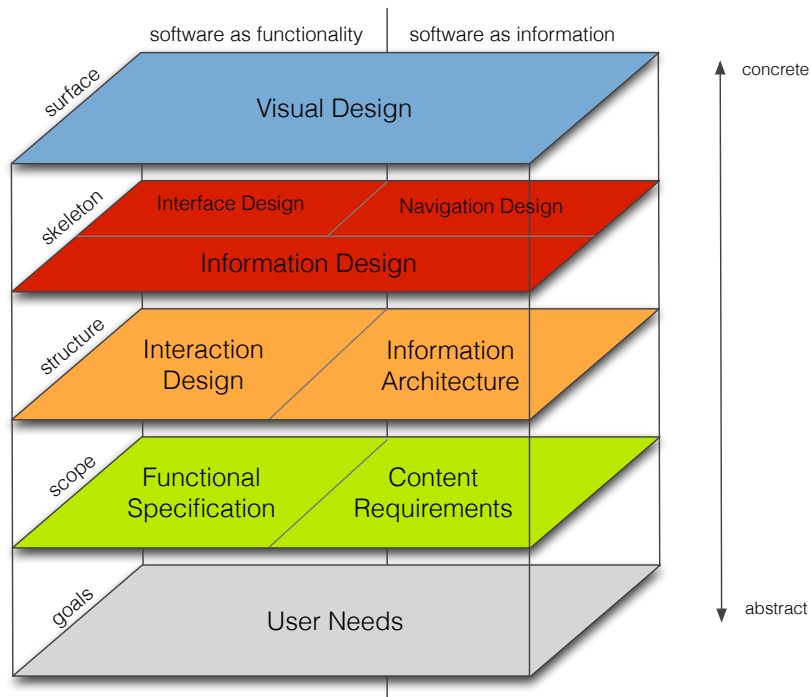
1. Recap: Information Architecture and Interaction Design
2. Specifying the structural aspects of your system: Interaction Design
3. Specifying the skeleton of your system: Interface Design
 - User Interface Design Patterns
 - Low fidelity prototyping

Learning goals for today

- You can differentiate all the terms with defined so far (interaction design, information architecture, information design, navigation design, interface design) and you know to which stage in the design process they belong to
- You know how to create a flow chart and a story board
- You know how to translate the data and ideas from the structural stage into a first design
- You can describe major interface design patterns
- You understand the advantages of using paper prototyping

From data to design (part 2)

Recap



Structural design for intuitive access to content

Developing application flows to facilitate user tasks

Defining data elements by primary nouns

Constructing major task flows

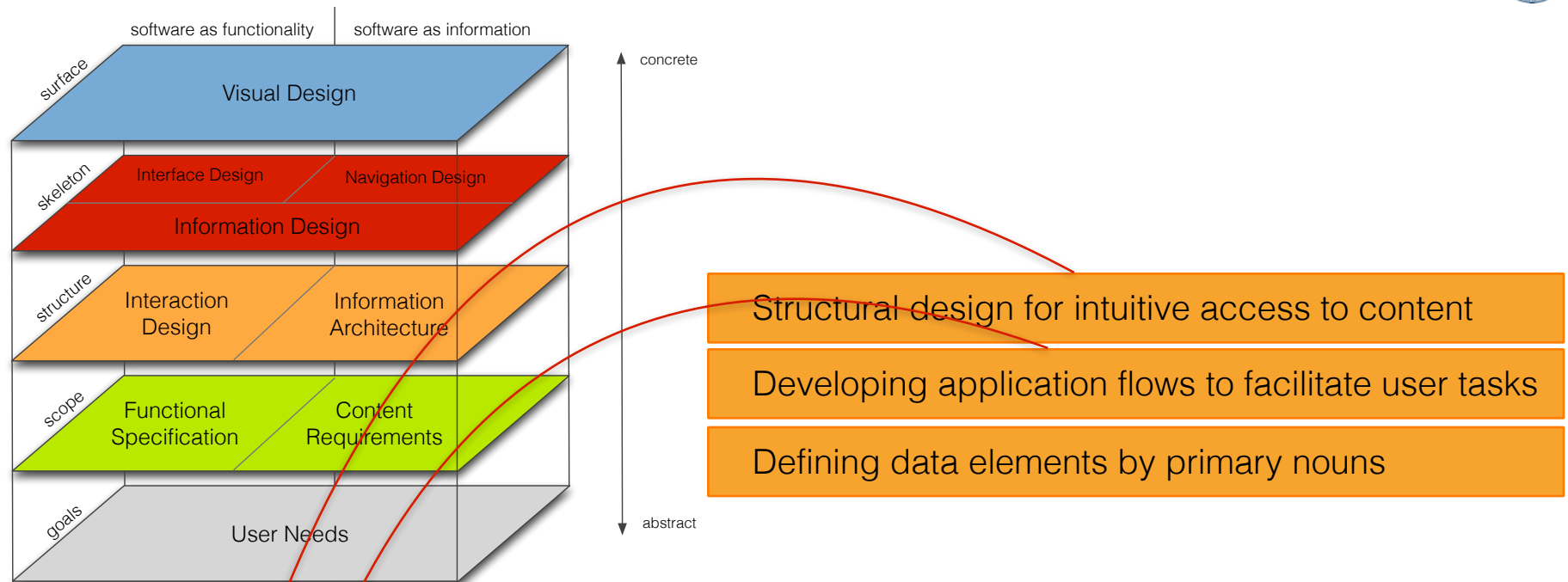
Describing a scenario

Defining a persona

Determining user groups

Carrying out interviews

Collecting information



Information Architecture

- Structural design of the information space to facilitate intuitive access to content

Interaction Design

- Developing application flows to facilitate user tasks
- Defining how the user interacts with site functionality

Information Architecture and Interaction Design

Content-centered applications:

Information Architecture

Structure is given through arrangement of content elements to facilitate human understanding.

Options involved in conveying information to a user.

Task-centered applications:

Interaction Design

Giving structure on the functionality: define how the system behaves in response to the user.

Options involved in performing and completing tasks.

Indirectly communicate conceptual model.

Information Architecture & Interaction Design

The user is not Cinderella!

Do not throw all information and functionality in the bucket (interface), all at once, and let the user select which ones are the „good“ (relevant) ones (for the current task).



(Garrett, 2012)

Information Architecture & Interaction Design

Reduce complexity and facilitate understanding

- Reduce visual and intellect burden by
 - only facing the user with relevant information and functionality for the respective moment (i.e. current task)
- Support the user to gain information. → **information architecture**
 - options to convey information to the user, i.e. name, organize, and interlink objects (contents) in away that they match the user's mental model
- Support the users to perform and complete their tasks → **interaction design**
 - define how the system responses to the user, i.e. sequences of options and available functionality

Specifying the structural aspects of your system

Interaction Design

Interaction Design

Requirements themselves don't describe how pieces fit together to form a cohesive whole.

- develop a conceptual structure for the site
- develop a conceptual model for the application

Conceptual Model: **Users' impressions of how the interactive components of the system will behave.**

- Knowing the user's conceptual model allows you to make consistent design decisions
- Example: the conceptual model for the shopping cart component of a typical e-commerce site is that of a shopping cart. This metaphorical concept influences both the design and the language we use in the interface

Conceptual Models

A good conceptual model allows us to predict the effects of our actions

[...] instead of designing software that works best for the machine, we could design software that works best for the people who use it [...]. (Garrett, 2011)

Using conceptual models people are already familiar with makes it easier for them to adapt to an unfamiliar site.

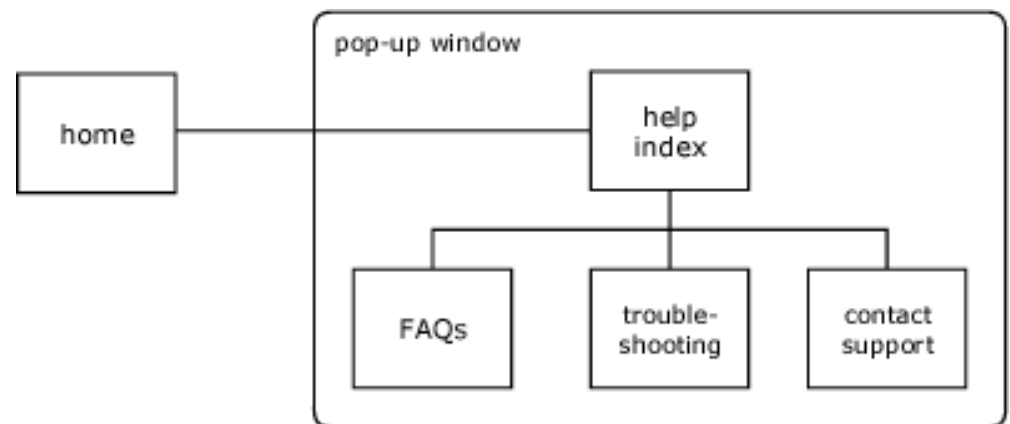
Of course, there's nothing wrong with breaking away from convention either—as long as you have a good reason for doing so and have an alternate conceptual model that will meet your users' needs while still making sense to them.

Unfamiliar conceptual models are only effective when users can correctly understand and interpret them.

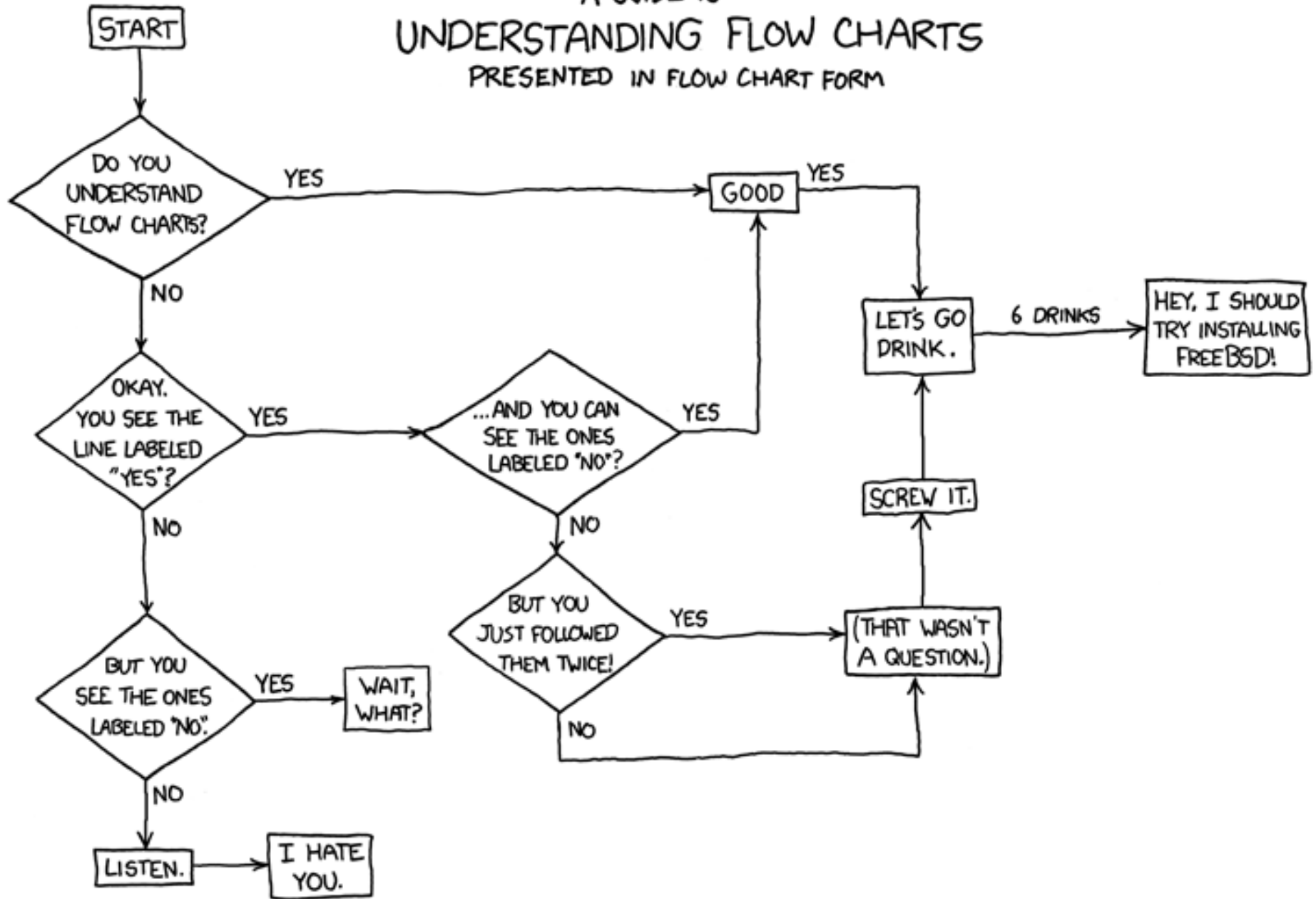
Flow Chart

The flow chart is a procedural representation of the different possible design flows at a very high-level

A flowchart can be seen as a decision tree for the user experience, i.e. if the user does "x" show screen "y"



A GUIDE TO UNDERSTANDING FLOW CHARTS PRESENTED IN FLOW CHART FORM



Derive a storyboard

The storyboard is a visual representation how users will accomplish a task in the system

The storyboard shows the steps the user will take AND how the system reacts to the user at each step

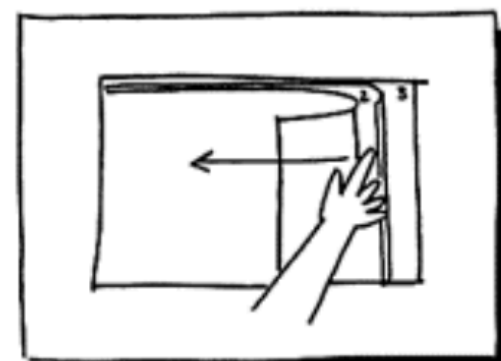
The task may be handed off between users, and may be supported by several systems operating together; the storyboard ensures the task remains coherent across these boundaries.



L. can write into her notebook.



She can flip over her pen and erase.



L. can flip to the next page.



L. marks her place.



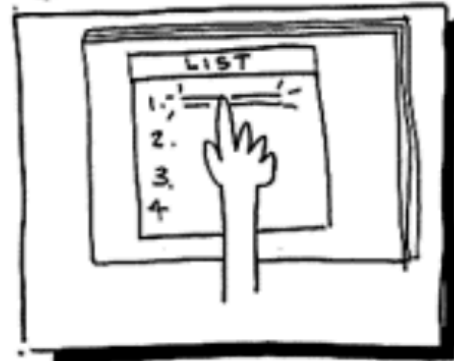
L. marks a phone number.



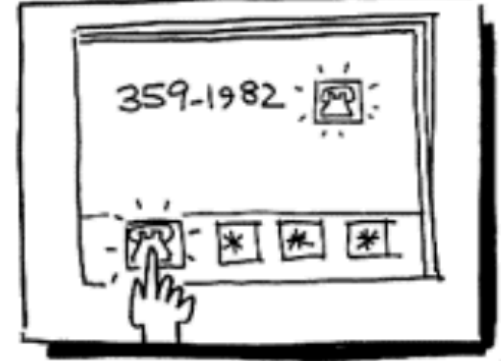
L. shifts to the InBox Section.



L. adds comments



L. brings up a list.



L. looks for phone numbers.

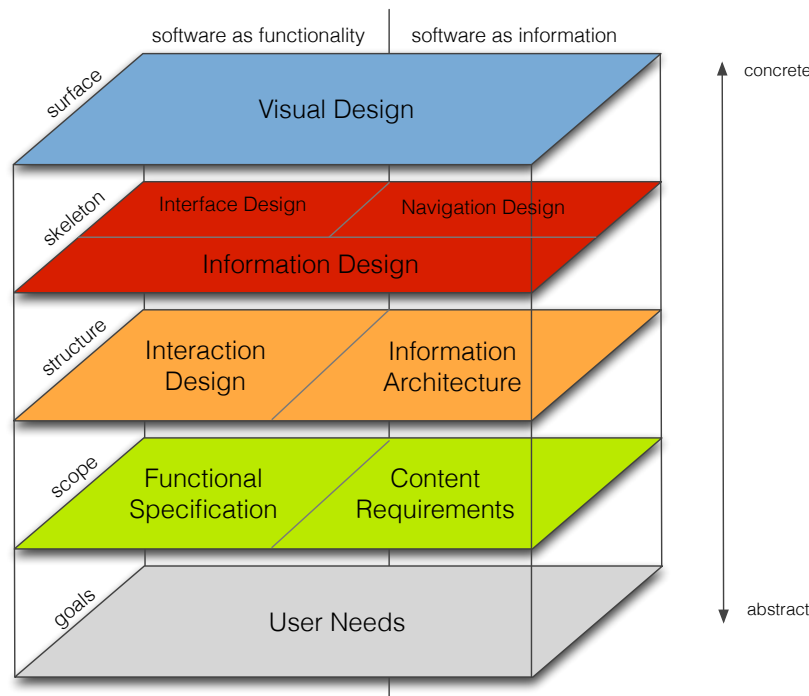
Hands-On: Designing interaction

Team working (30 minutes)

1. Specify the general interaction design via a flow chart based on the the user scenario of your persona (10 minutes).
2. Sketch within the next 15 minutes at least three (3) views / pages by
 - (i) providing the navigation and
 - (ii) providing specific actions.
3. Prepare to present your result. Use the last 5 minutes to pin everything on the wall.

Specifying the skeleton of your system

Interface Design



Designing the presentation of information

Designing interface elements for user's doing

Designing interface elements for user's moving

Structural design for intuitive access to content

Developing application flows to facilitate user tasks

Defining data elements by primary nouns

Constructing major task flows

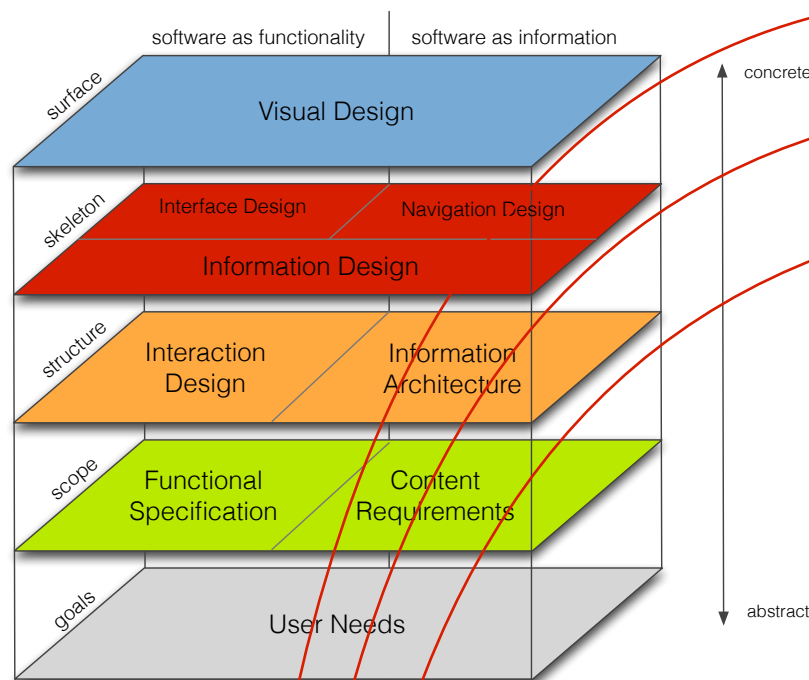
Describing a scenario

Defining a persona

Determining user groups

Carrying out interviews

Collecting information



Designing the presentation of information

Designing interface elements for user's doing

Designing interface elements for user's moving

Information Design

- Designing the presentation of information to facilitate understanding

Interface Design

- Designing interface elements to facilitate user interaction with functionality

Navigation Design

- Designing interface elements to facilitate the user's movement through the information architecture

Functional and Formal Aspects of Interaction

Interaction Design

- Process of interaction
 - Form: describes way to take when performing a task
- Structuring functionality
- How the system behaves in response to the user

Interface Design

- Functional presentation of interaction
- Selecting functional elements
- Communicates
 - WHAT is it
 - HOW does it work
 - HOW to control

What questions should you ask?

Affordance: does the visual design provide sufficient cues?

Intuitiveness: Is the design intuitive – does the user expect it?

Ease: Is the interaction easy to perform?

Integration: Do all of the interactions used in the design work elegantly together?

Controls: Are UI controls appropriate?

Feedback / Messages: Are error messages, feedback, and instructions meaningful and useful?

Interface Design Patterns

A pattern describes an optimal solution to a common problem within a specific context.

As such, patterns can be a description of best practices within a given design domain.

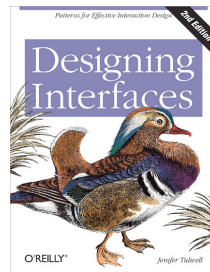
They capture common solutions to design tensions (usually called “forces” in pattern literature) and thus, by definition, are not novel.

They aren't off-the-shelf components; each implementation of a pattern differs a little from every other.

They aren't simple rules or heuristics either. And they won't walk you through an entire set of design decisions [...]

From „Designing Interfaces“ by Jenifer Tidwell“ <http://designinginterfaces.com/>

An extensive overview of pattern libraries: <http://developer.yahoo.com/ypatterns/about/libraries.html>



Hands-On: Patterns for Interaction Design

Round 1 (single student, 3 min):

Every student gets one pattern. Read and understand your pattern.

Round 2 (group of two, 5 min):

Create a group of two students. Exchange your patterns, read it and explain it to each other.

Round 3 (group of four, 8 min):

Create a group of four students. Exchange your patterns, read it and explain it to each other.

Round 4 (group of eight, 15 min):

Create a group of eight students. Exchange your patterns, read it and explain it to each other and design an exhibition by clustering your patterns in a meaningful way.

Round 5 (plenum, 15 min):

Each group gets five minutes to present the results.

Specifying the skeleton of your system

Low-fidelity prototyping

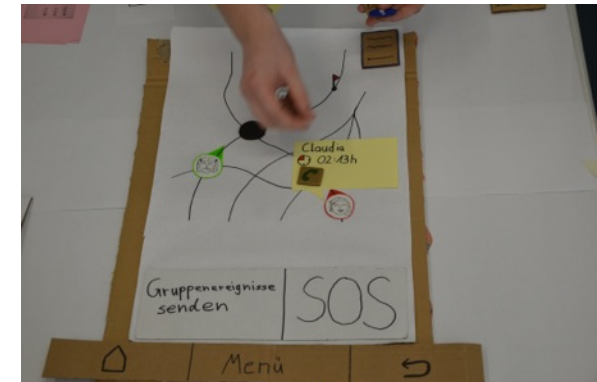
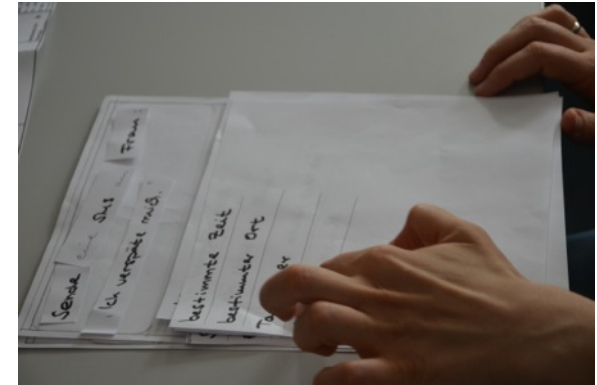
Designing a low-fi prototype

Early representation of a design idea

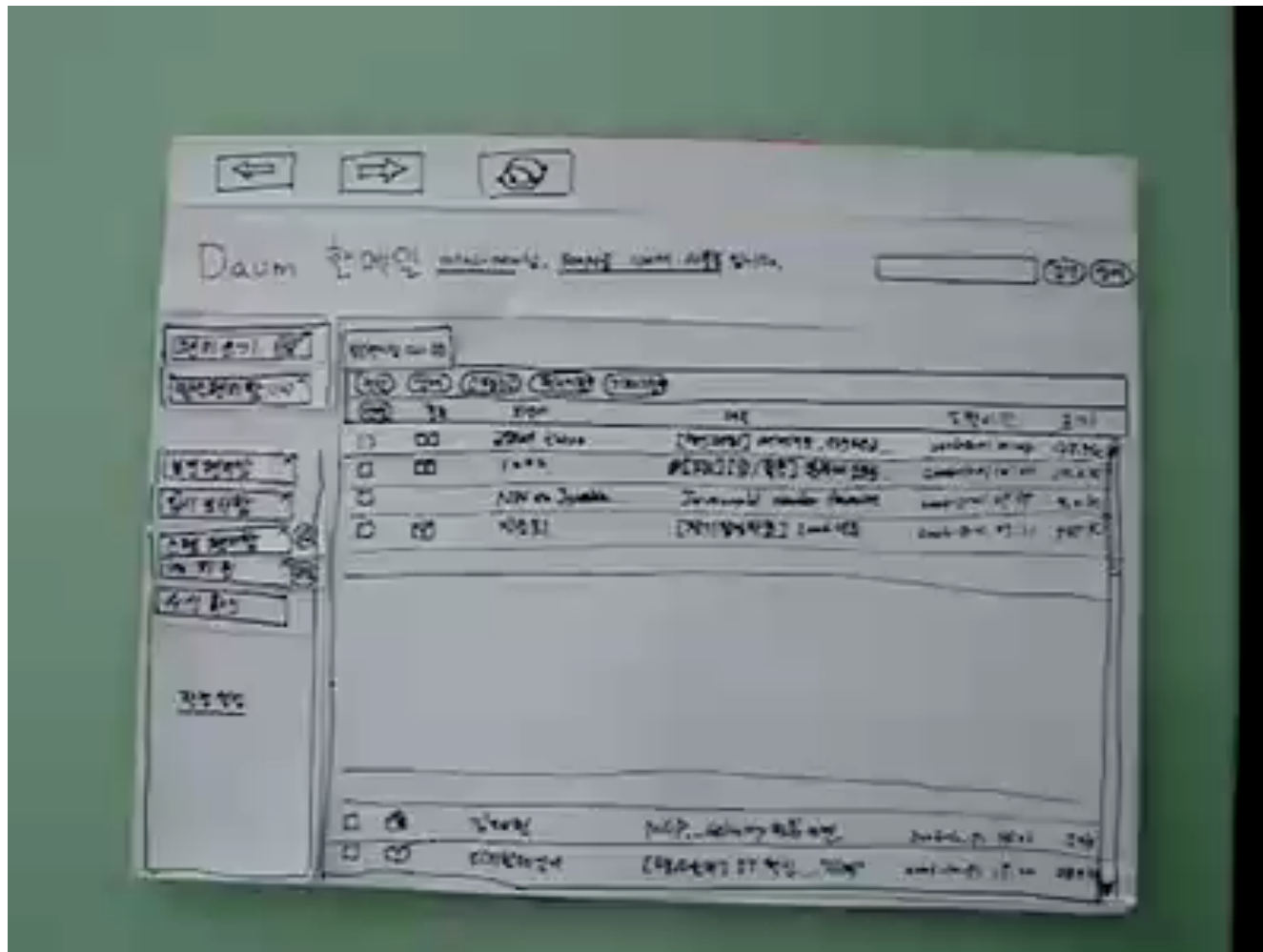
Educates developers to have a concern for usability and formative evaluation

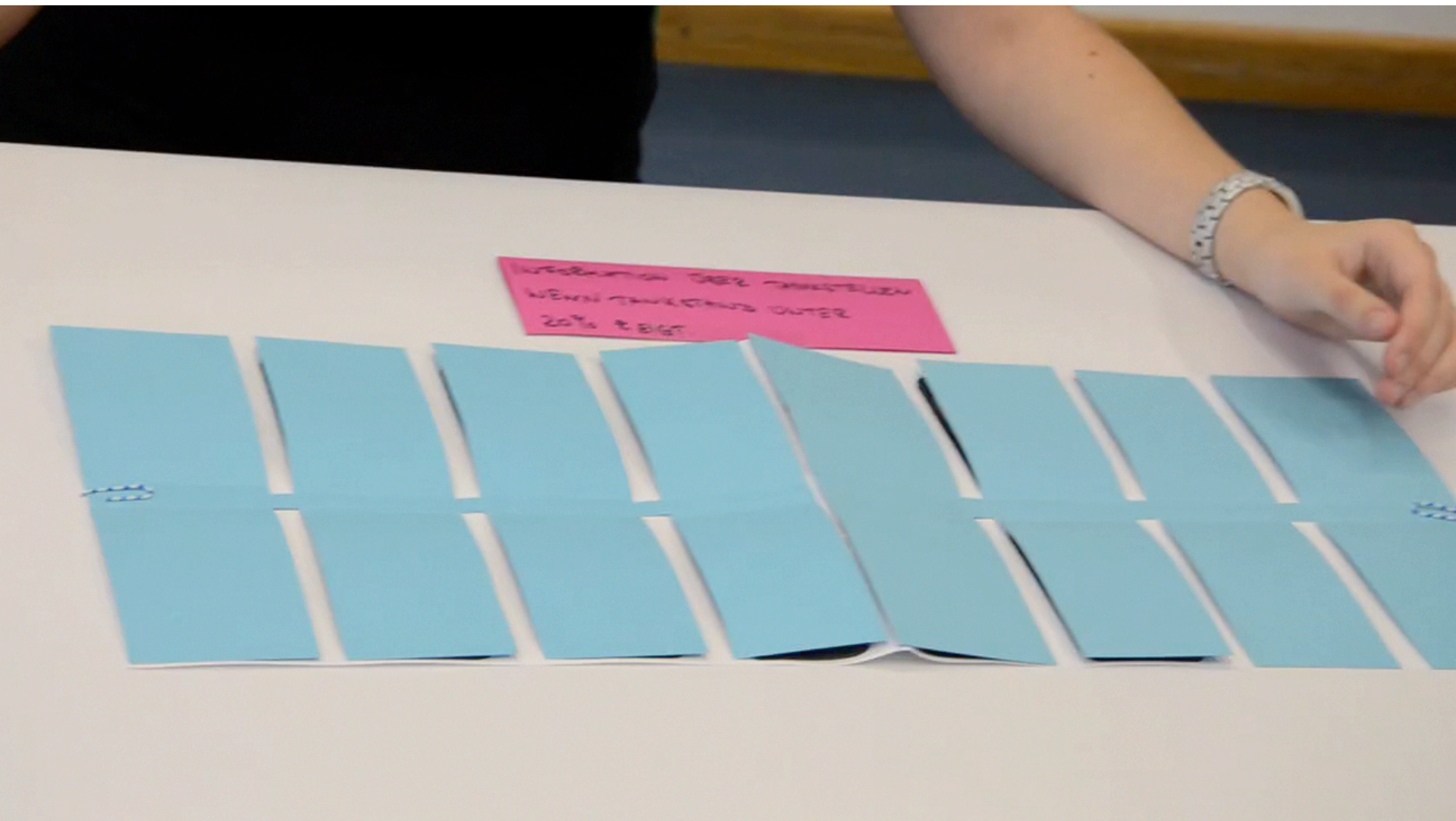
Maximizes the number of times you get to refine your design before you code

It is easy as well as relatively fast to make

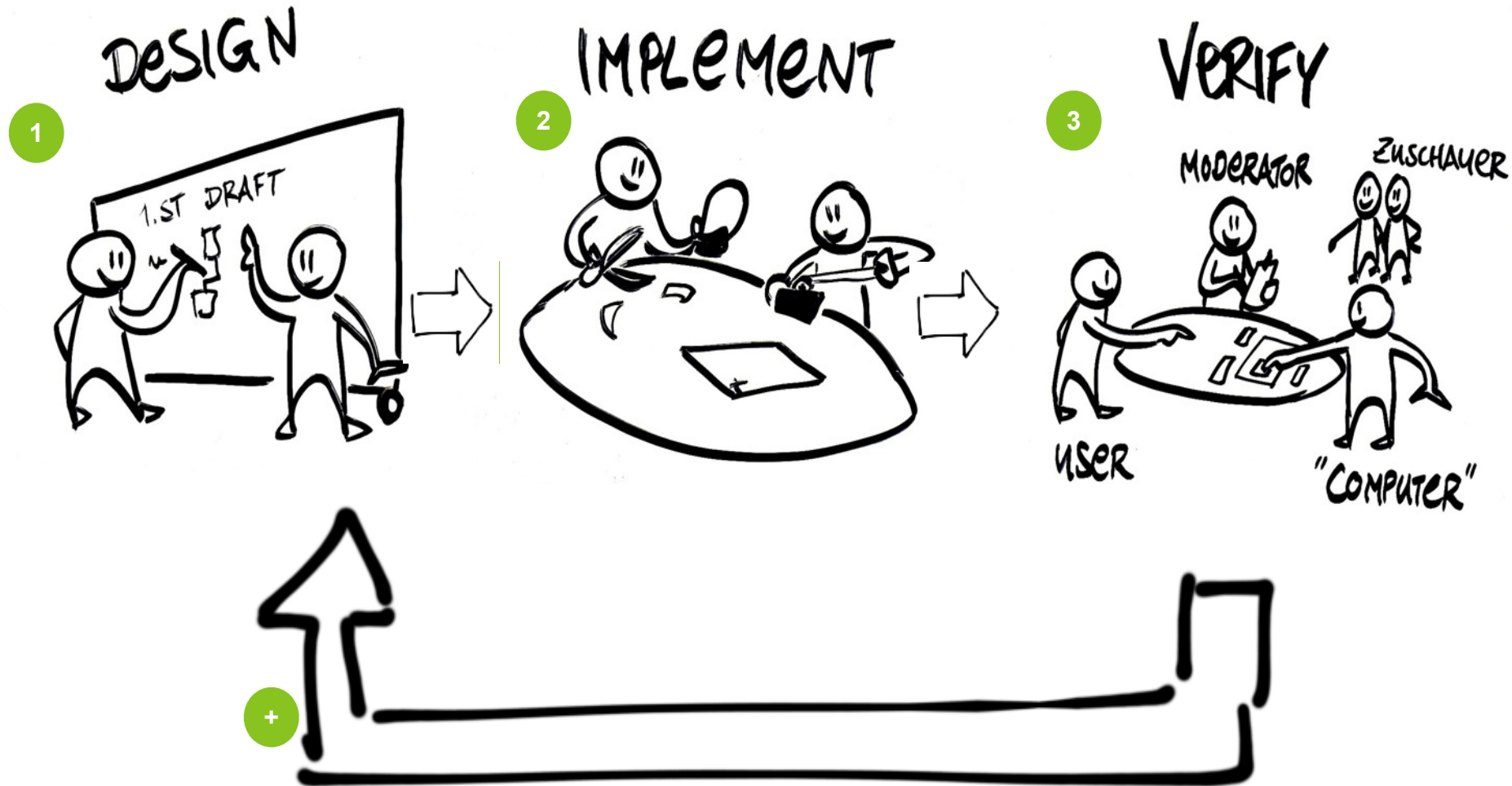


"Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person 'playing computer,' who doesn't explain how the interface is intended to work."





Process of paper prototyping



Building 1: Assemble a kit

- Large, (heavy) white paper
- Colored paper
- Thumbtacks
- Cardboard or foam core
- Index cards
- Tape or glue
- Pencils, pens, markers
- Overhead transparencies
- Scissors



Building 2: Design a first draft

- » Organize all user data (persona, scenario, task flow, primary noun, storyboard) and connect all insights
- » Review the primary screens you defined for your software in the flow chart and storyboard
- » Sketch the interface which should be extremely simple at first (“the rectangles phase”)
- » Label the rectangles, and illustrate and describe how one grouping or element affects others
- » Set a deadline

Building 3: Implement the paper prototype

Concentrate on the primary pathways through the interface that the persona takes with the greatest frequency

- » Primary pathways (i.e. key path scenarios) are defined in your flow charts/storyboards
- » Primary pathways are also described in the context scenarios

Describe in detail the precise behavior of each major interaction and provide a walkthrough of each major pathway.

- » Describe tasks a user can carry out with your interface
- » Use “real” data instead of placeholders (“John Smith” vs. “a person“)

Hands-On: Create a first paper prototype

Single person working (10 minutes)

Start with the rectangle phase.

The prototype has to be somewhat interactive (i.e., they have to have some moving parts). Just enough to prototype realistic interaction, but you do not need to animate every single menu, list, text field, etc.

Your goal for developing the paper prototype is to have an artifact that you can put in front of a user and learn whether your design is usable and complete.

Any time you are not sure how much effort you need to put in any part of your prototype, just ask yourself whether you expect to need it when you test it out.

Thank You!