

# Prototyping

«UCD: User-Centered Software Development»

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

May 18, 2015

# Prototyping Introduction

# Context for today

## Last 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

## This week

Bringing interaction and interface design together and evaluating your designs

***How to improve your design?***

# Outline

1. Basics of low fidelity prototyping
2. Verifying a lo-fi prototype
3. Basics of hi-fi prototyping

# Learning goals for today

- You understand the advantages of using paper prototyping
- You can carry out test sessions for paper prototypes

# Hands-On: Order the chaos

## **Single person working (10 minutes)**

On the next slide you will find all the methods and approaches we have discussed in this course so far. Unfortunately there are somehow mixed up.

Please bring them into the correct order based on their numbers.

Expected result:

Sequence of numbers such as, 5-7-1-...

(5) Collecting information

(2) Defining data elements by primary nouns

(7) Designing the presentation of information

(9) Designing interface elements for user's doing

(8) Defining a persona

(12) Carrying out interviews

(4) Designing interface elements for user's moving

(10) Constructing major task flows

(3) Describing a scenario

(1) Determining user groups

(11) Developing application flows to facilitate user tasks

(6) Structural design for intuitive access to content

(5) Collecting information

(12) Carrying out interviews

(1) Determining user groups

(8) Defining a persona

(6) Describing a scenario

(10) Constructing major task flows

(2) Defining data elements by primary nouns

(6) Structural design for intuitive access to content

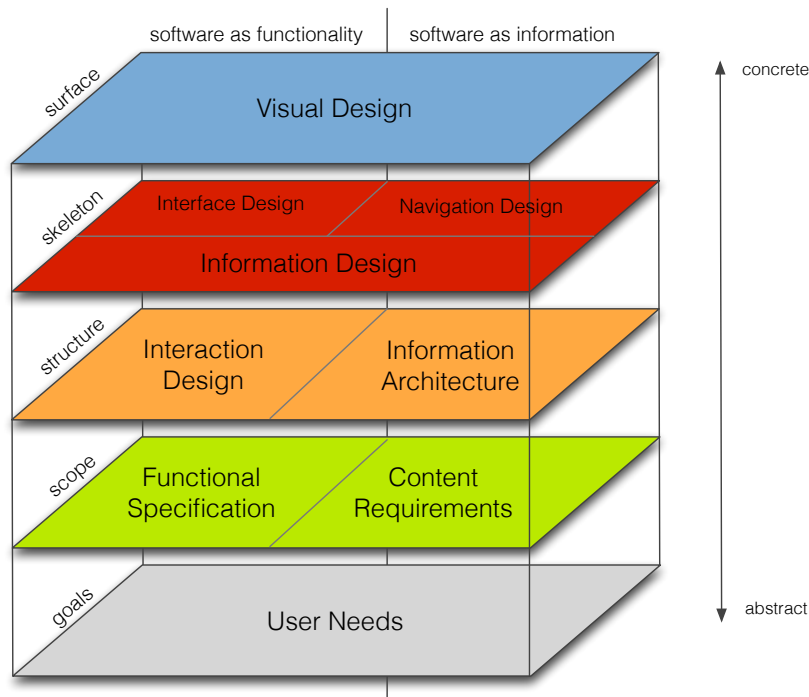
(11) Developing application flows to facilitate user tasks

(4) Designing interface elements for user's moving

(9) Designing interface elements for user's doing

(7) Designing the presentation of information





Designing the presentation of information

Designing interface elements for user's doing

Designing interface elements for user's moving

Developing application flows to facilitate user tasks

Structural design for intuitive access to content

Defining data elements by primary nouns

Constructing major task flows

Describing a scenario

Defining a persona

Determining user groups

Carrying out interviews

Collecting information

# Prototyping

## Basics of low fidelity prototyping

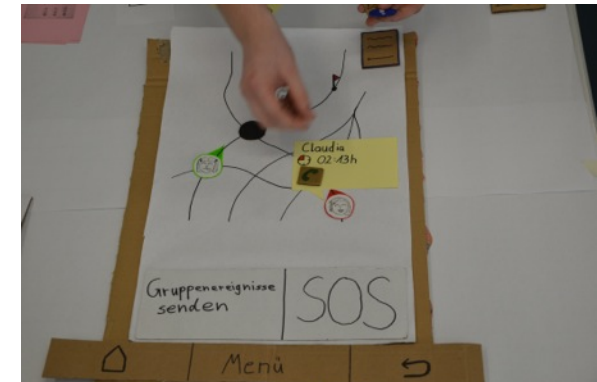
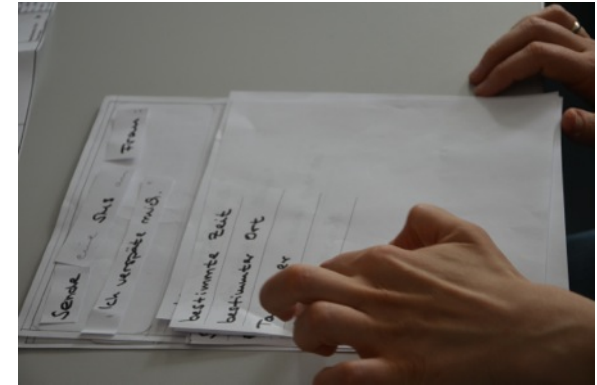
# Advantage a low-fi prototypes

Early representation of a design idea

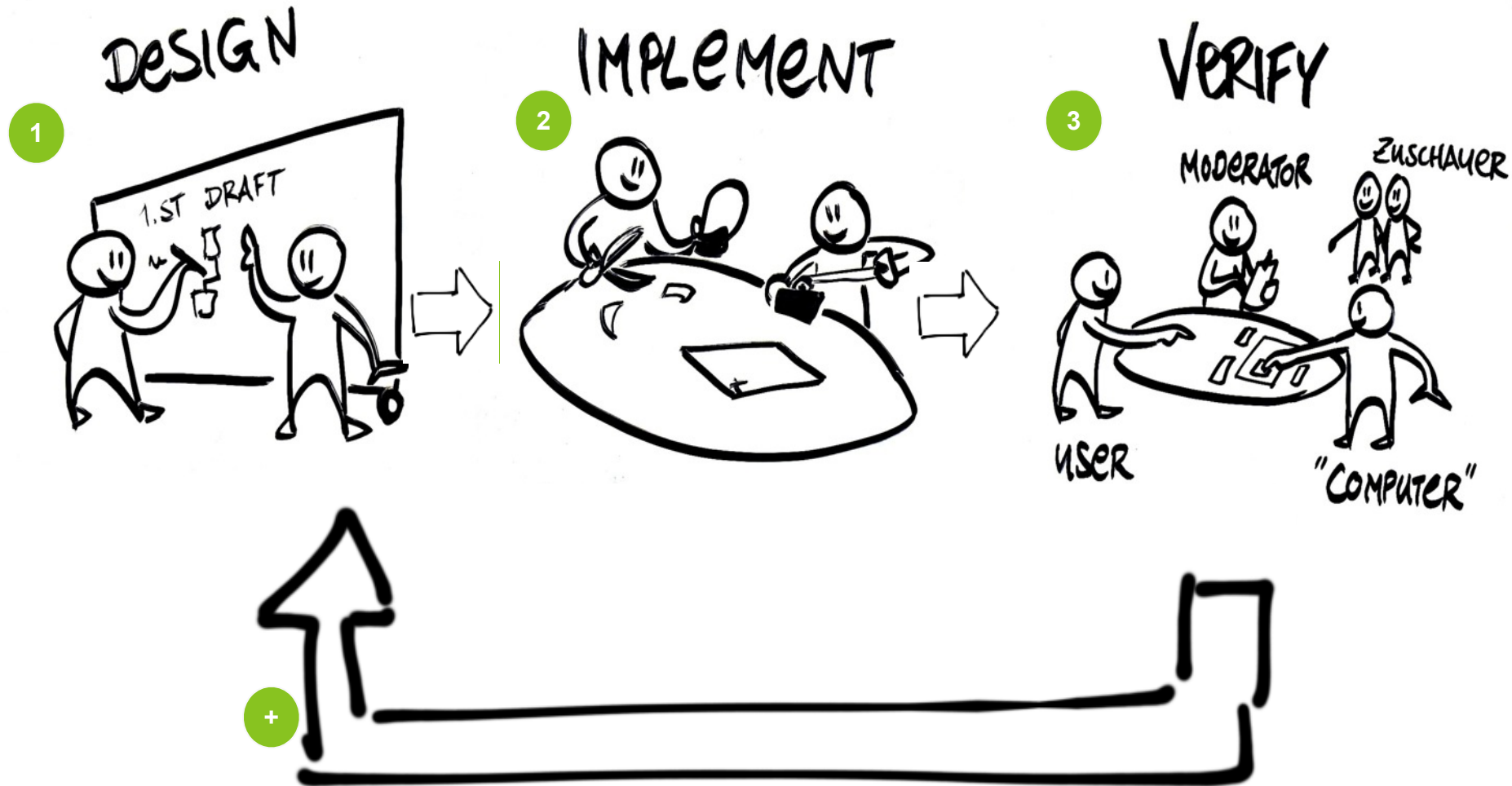
Educates developers to have a concern for usability and formative evaluation

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

It is easy as well as relatively fast to make



# Process of paper prototyping



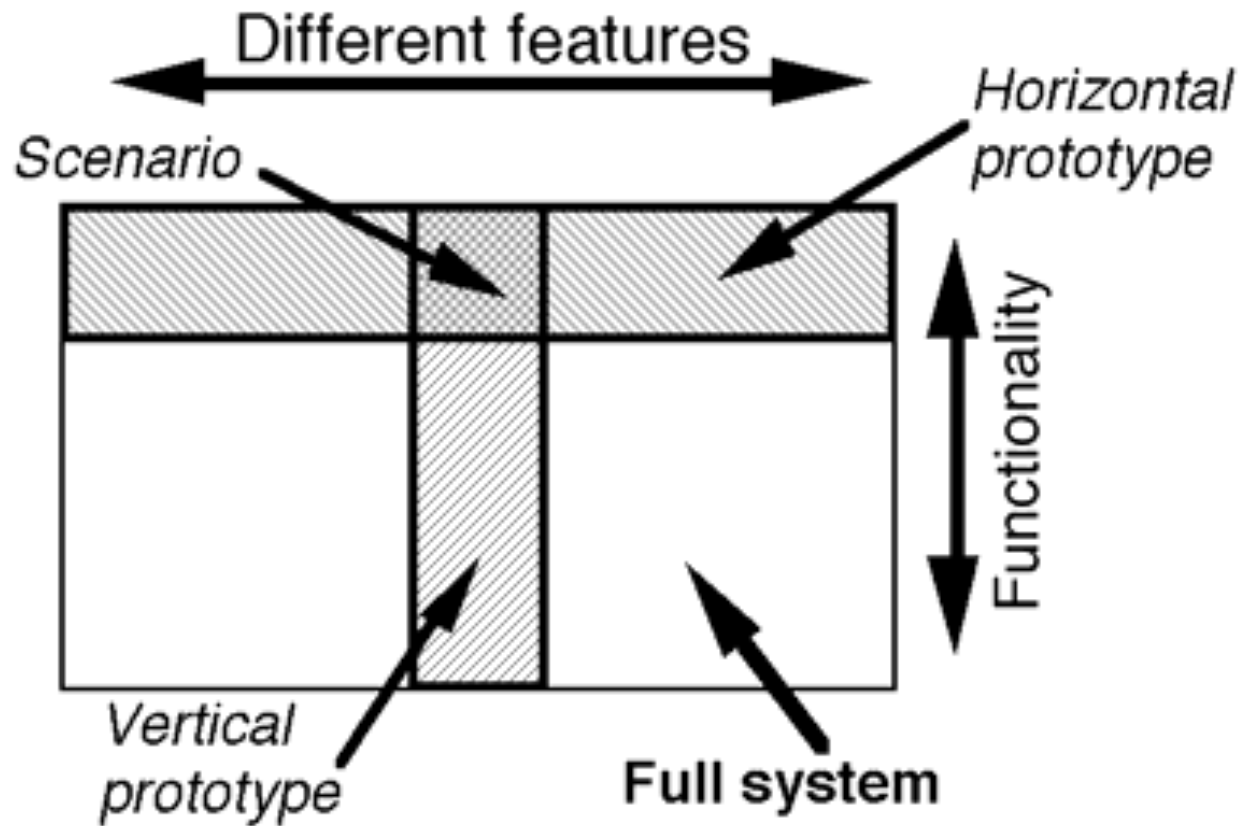
Low-fidelity prototyping

# Verifying a lo-fi prototype

# Formative evaluation

- Discover usability problems as part of an iterative design process.
- It can be used to test very early paper prototypes or partially developed products, the information architecture of the product, or a particular feature by itself
- Goal: Uncover as many problems as possible
- Paper prototyping
  - Is a variation of usability testing
  - Representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person 'playing computer'
  - Nobody explains the users how the interface is intended to work

# Vertical and horizontal prototypes



# Preparing your test

**1**

Select your users

**2**

Prepare test scenarios

**3**

Practice



# Test scenario and tasks

## **What is the scenario?**

- A good scenario will help your participants get into a frame of mind that is representative of the population and situation where your design will be used

## **What tasks will you ask the user to perform?**

- You need to define task that should answer questions you have regarding your user interface
- You should start with something simple (e.g., let the user explain the user interface first)
- Make sure that the tasks
  - are formulated in the language of the user (rather than the language of the system/interface)
  - do not imply a specific set of steps to be taken

# Example: test scenario and tasks

## *Scenario*

» “We’re interested in seeing how people use this web page to find information about parking fines.”

Task 1: “Find the fine for a Saturday ticket in a handicap zone”

Task 2: “Find the number of weeks you have to pay a fine before the fine increases”

Task 3: “Find the address to send fine payments to.”

# What to avoid by defining tasks

## **Language of the interface**

- » Example: “send a duplexed copy to the printer”
- » Instead: “get this document on paper, with content on each side”

## **Specific Instructions**

- » Example: “find the print dialog and set up your job, then print”
- » Instead: “print this”

## **Explore and critique (Takes the user out of the realistic task focus and into a world of critique.)**

- » Example: “Look at this web site and tell me what you think.”
- » Instead: “What are the most important things on this page. Why?”

# Preparing test sessions with actual tasks

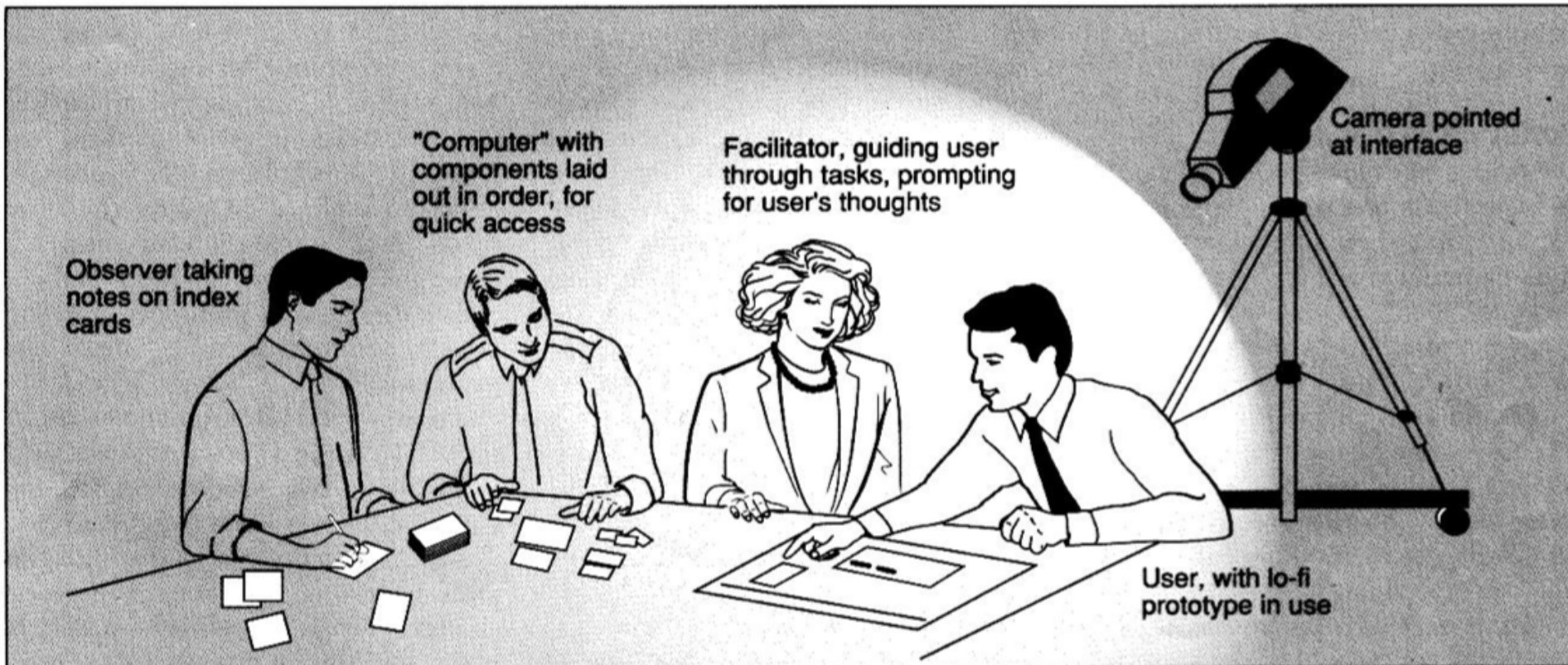
<b>Task#: &lt;task name&gt;</b>	
<b>Goal/result:</b>	
<b>Presumptions:</b>	
<b>Steps:</b>	
<b>Approximated Time:</b>	
<b>Hints for the user:</b>	
<b>Notes:</b>	

# Preparing test sessions with actual tasks

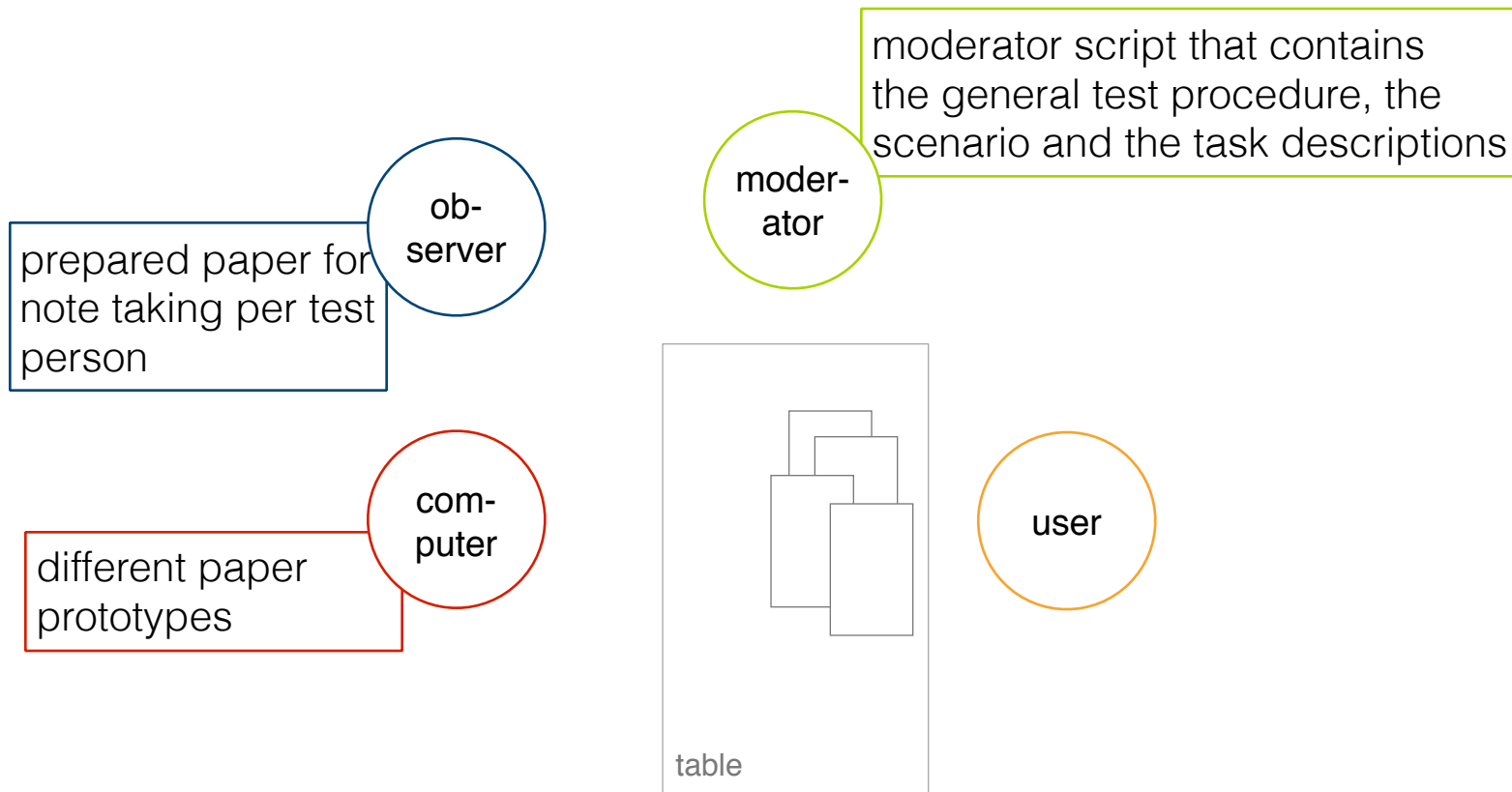
## Task 1: “Find the parking fine for a Saturday ticket in a handicap zone”

<b>Goal/result:</b>	The user should know the fine.
<b>Presumptions:</b>	The user should find the selector with the fine rate straight forward.
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Choose parking fine rates.</li> <li>2. Click on Saturday</li> <li>3. Click on handicafe zone</li> </ol>
<b>Approximated Time:</b>	1 minute
<b>Hints for the user:</b>	none
<b>Notes:</b>	—

# Conducting a test



# Needed documents for conducting a test



# Typical test session

Typical test session last about an hour.

## Phases

- » Getting ready
- » Conducting the test
- » Debriefing

In the getting ready phase you should begin with greetings, introductions, refreshments, and general ice-breaking. Make sure that the tester understands that the test is confidential and that the results will remain anonymous

In the debriefing phase you should ask questions, gathering impressions, and expressing your thanks



# Typical data to collect

## Performance data

- » Tabulate all of the breakdowns, workarounds (completion times)
- » How many participants finished each task? How many failed?

## Observations

- » What you see the participant do?
- » Where, when, how did the interesting events occur?

## Quotes

- » What do you hear the participant say?

## Questions

- » Actions or statements that were unclear or that you didn't capture completely

# Evaluating results

All collected problems should be reflected, summarized and prioritized

- » Identify areas where design failed to meet participant expectations
- » Identify areas where participants showed confusion
- » Identify tasks or actions participants were unable to complete

Try to be objective, don't write problems off

Discuss as a team and look for general patterns in user behavior

Let your insights guide redesigns for your next prototype iteration



Prototyping  
PAUSE (15 minutes)

# Hands-On: Develop a first test plan

## **Team working (30 minutes)**

Think about what you want to learn, and then design your test plan based on that.

Things that you will want to think about:

1. Think about the questions you have regarding your user interface design
2. Define a general test scenario (write it down!)
3. Specify two tasks the user will perform (they should help to answer your questions)

Expected results:

- scenario
- two tasks description
- general test procedure (including who is doing what)

# Hands-On: Let's test

**Audience (2 x 20 minutes)**

# Hands-On: Test your test plan

**Team working (20 minutes)**

Carry out one round of testing :)

# Prototyping

## Basics of hi-fi prototyping

# High-fidelity prototyping

Provide increasing amounts of functionality and refinement

Usually involve some amount of programming and interactivity

Can provide functionality that can be tested with users

Take more time and resources to build

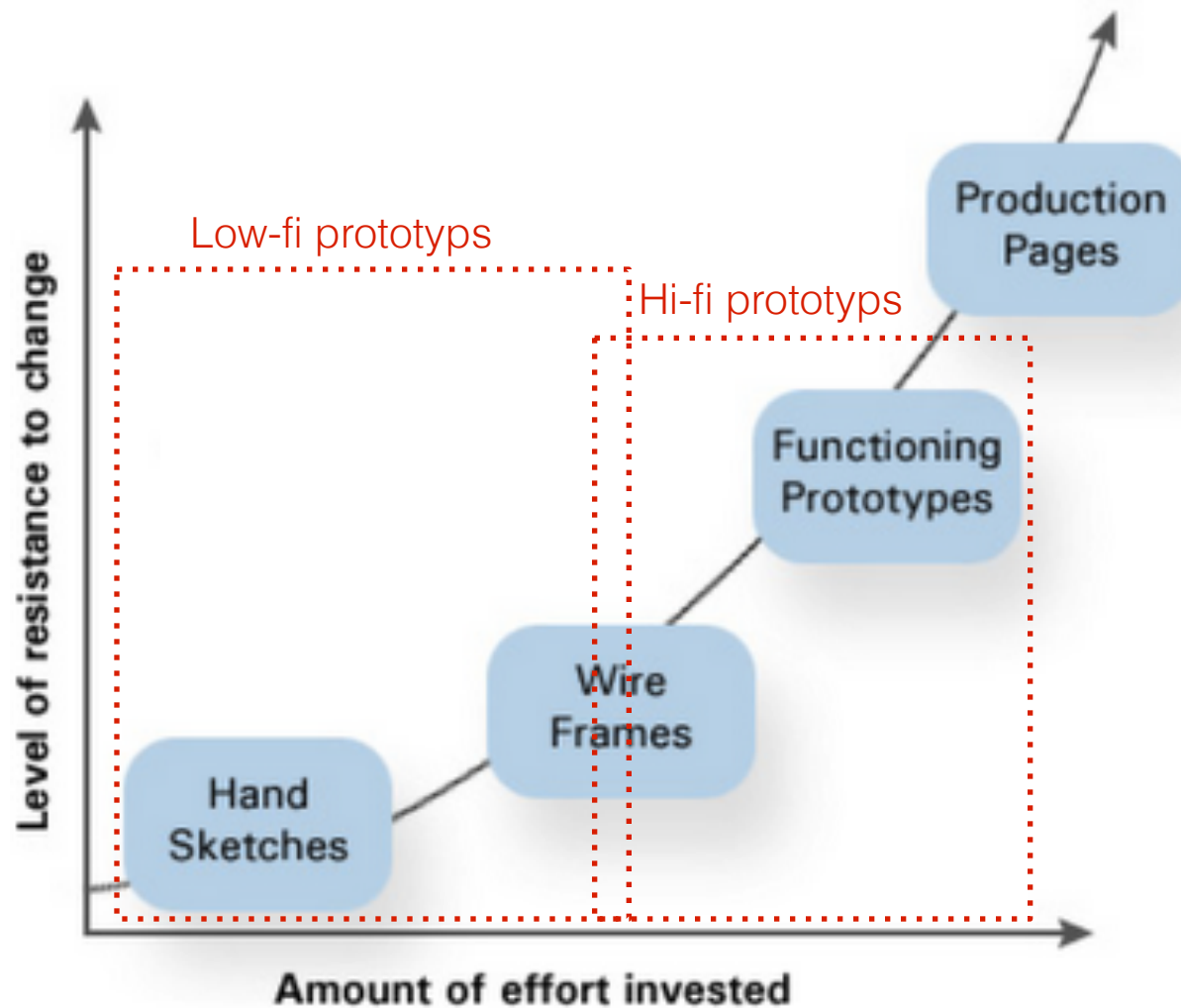
Can be distracted by limitations



# Problems with hi-fi prototypes

- Hi-fi prototypes take too long to build and change
- Reviewers and testers tend to comment on “fit and finish” issues
- Developers resists to change
- A prototype in software can set expectations that will be hard to change
- A single bug in a hi-fi prototype can bring a test to a complete halt

# Prototyping



Basics of hi-fi prototyping

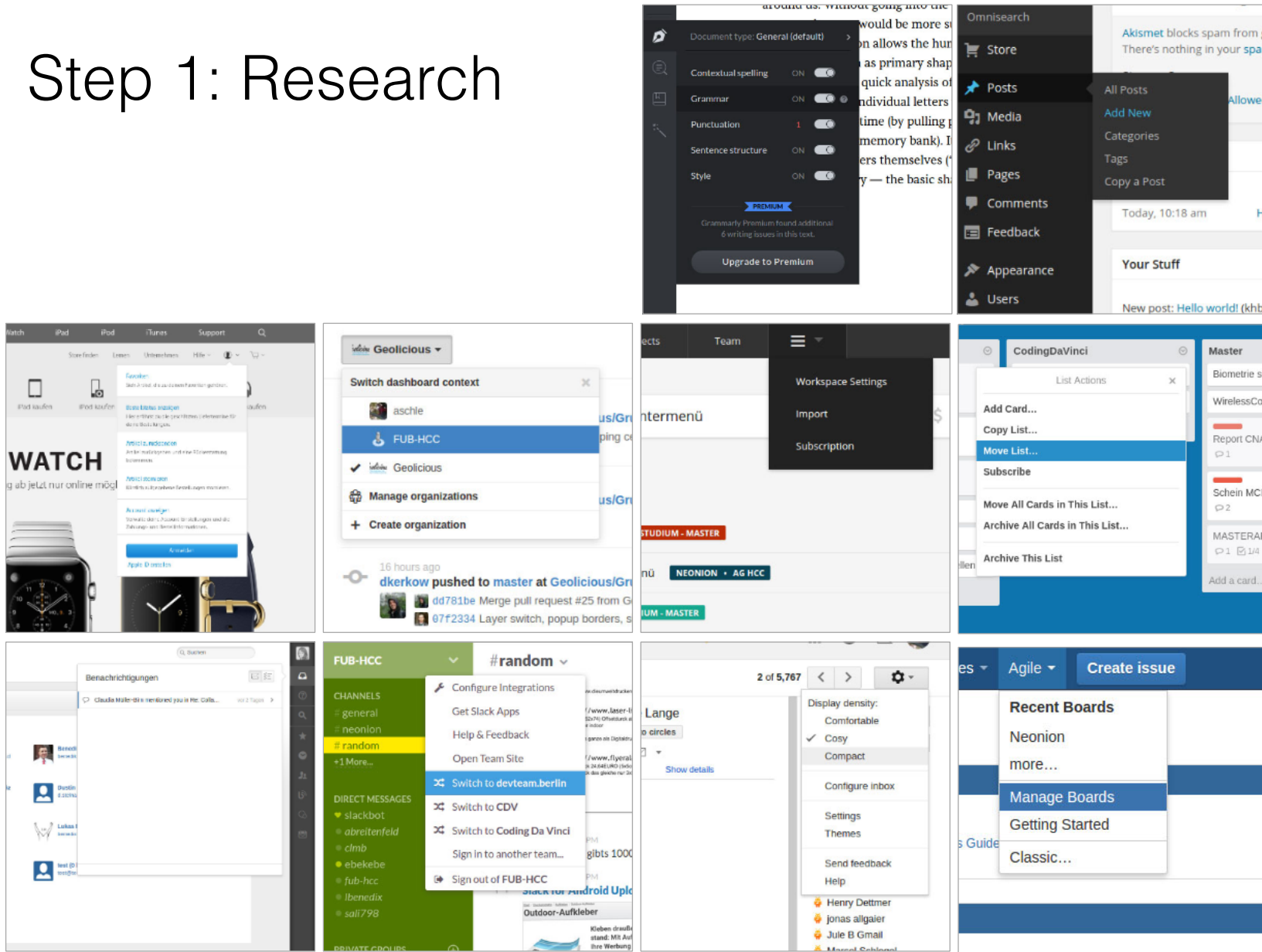
Example for combining paper prototyping and  
high fidelity prototyping

# Problem

Introducing a secondary menu for two new functions:

- “Switch Context” - user wants to change the annotation type
- “List of Contributors” - user wants to (un)check the various annotators to a document

# Step 1: Research

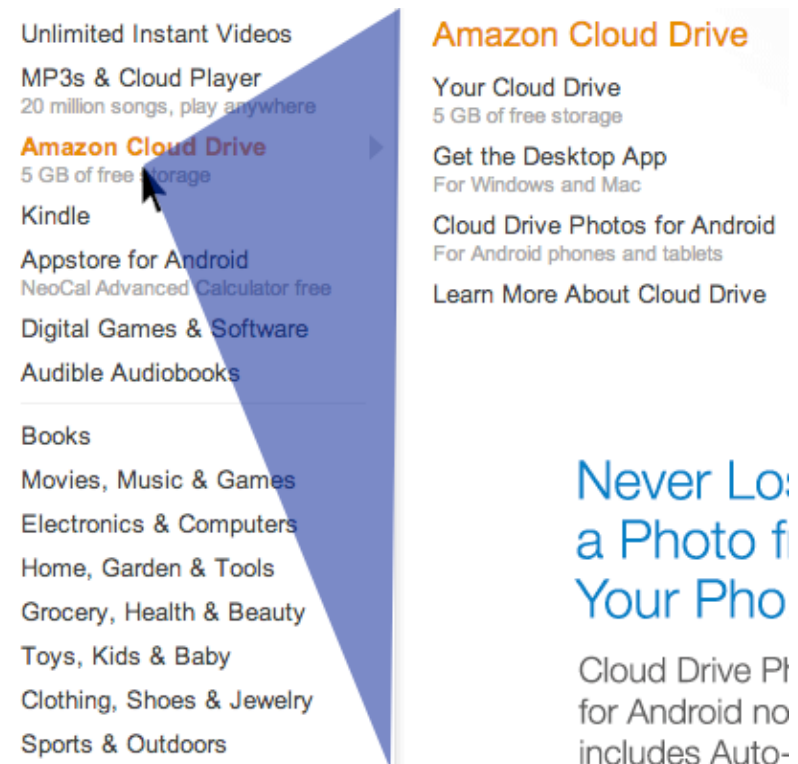


## Step 2: General design decisions

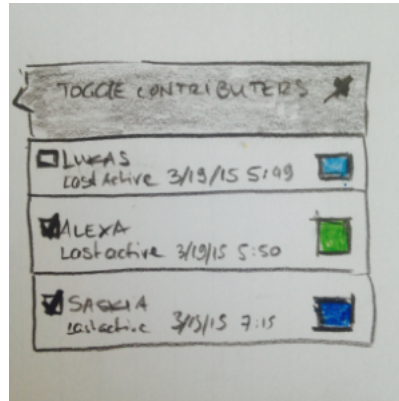
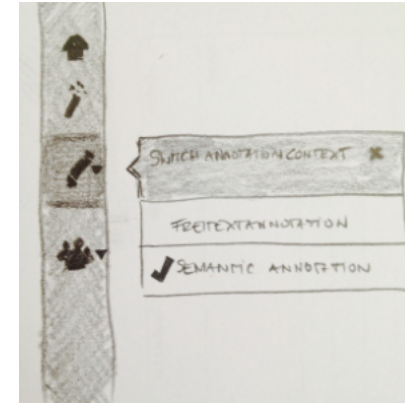
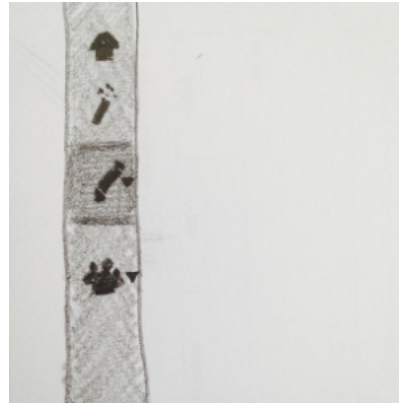
- Use a small arrow that indicates a fold out menu
- Open the menu via a mouse click (not hover!)
- Close the menu via a mouse click (close icon) or when staying with the mouse pointer outside the menu for some time
- Use a notable effect during hover
- Take into account the behavior of jQuery-menu-aim (<https://github.com/kamens/jQuery-menu-aim>)

# Detecting the direction of the cursor's path

- At every position of the cursor you can picture a triangle between the current mouse position and the upper and lower right corners of the dropdown menu
- If the next mouse position is within that triangle, the user is probably moving their cursor into the currently displayed submenu
- As long as the cursor stays within that blue triangle the current submenu will stay open
- If the cursor goes outside of the blue triangle, they instantly switch the submenu



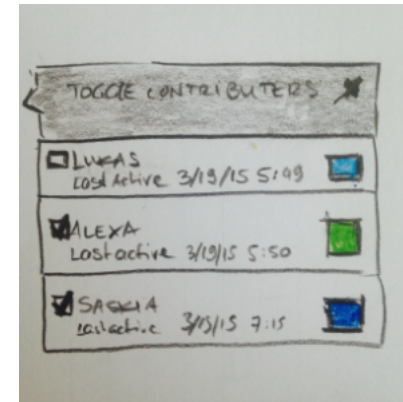
# Step 3: Paper Prototype



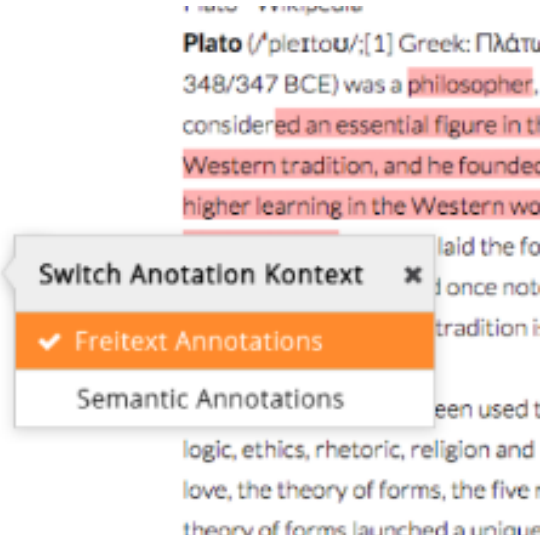
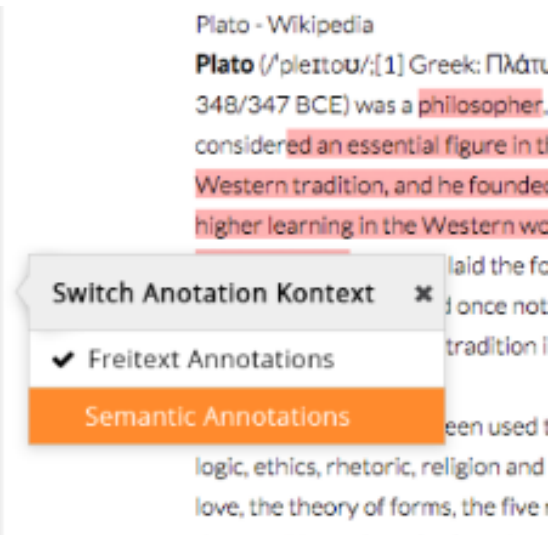
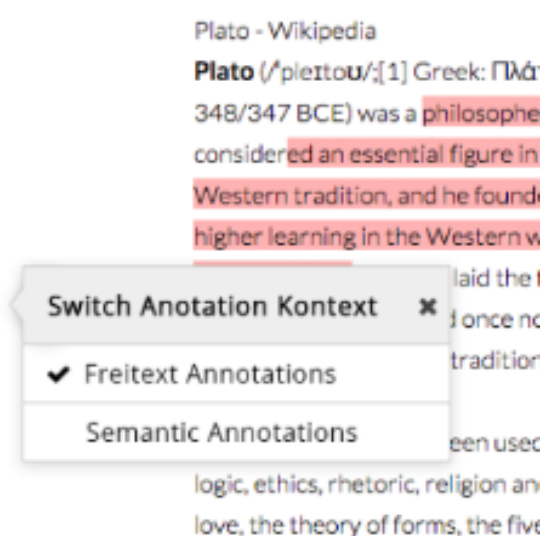
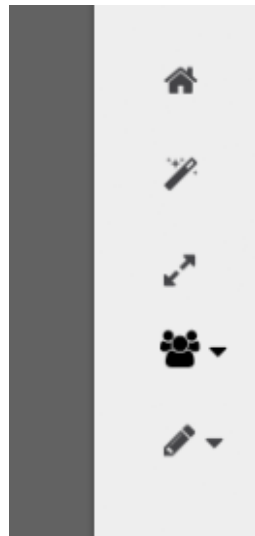


## Step 4: First team feedback

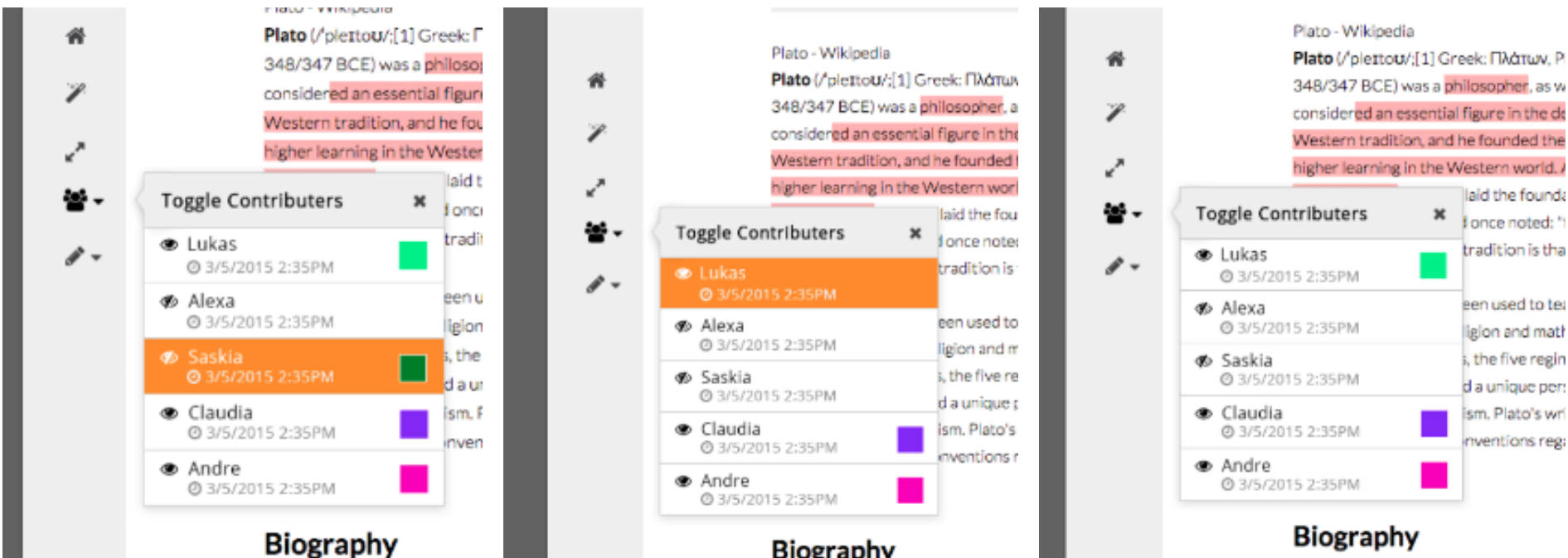
- Use icons for indicating the selected person instead of check boxes
- Align the design to the annotator view



# Step 5: Detailed Photoshop design “Switch Context”



# Step 5: Detailed Photoshop design “List of Contributors”



# What you can learn from hi-fi prototypes

Everything you learn from a paper prototype, and

Screen layout

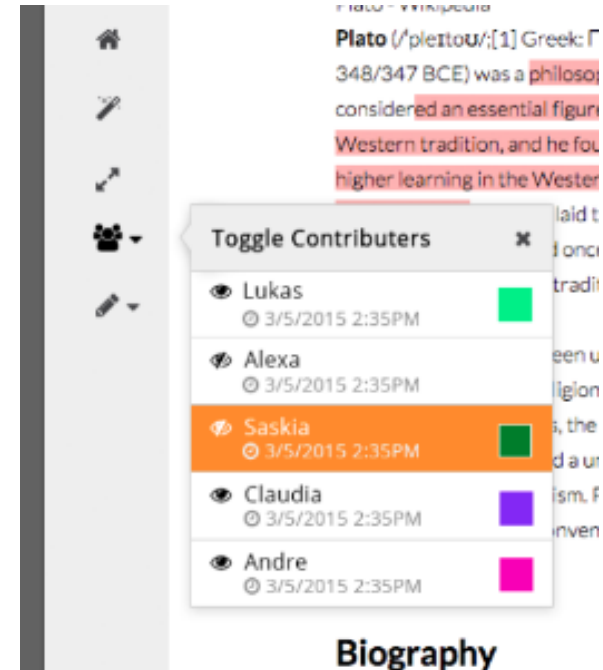
- » Is it clear, overwhelming, distracting, complicated?
- » Can users find important elements?

Colors, fonts, icons, other elements

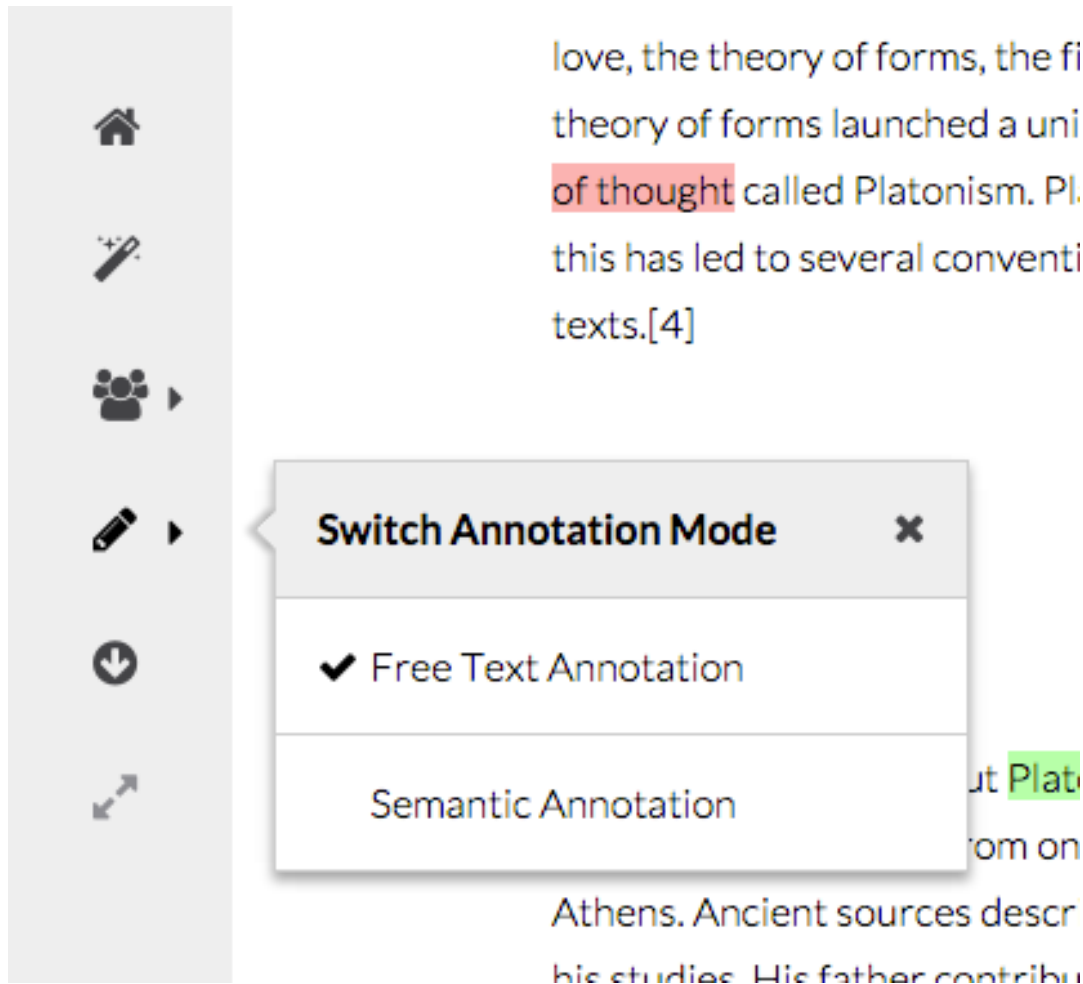
- » Well-chosen?

## Step 6: Second Team Feedback

- Align the color of activated and not-activated eyes in the menu

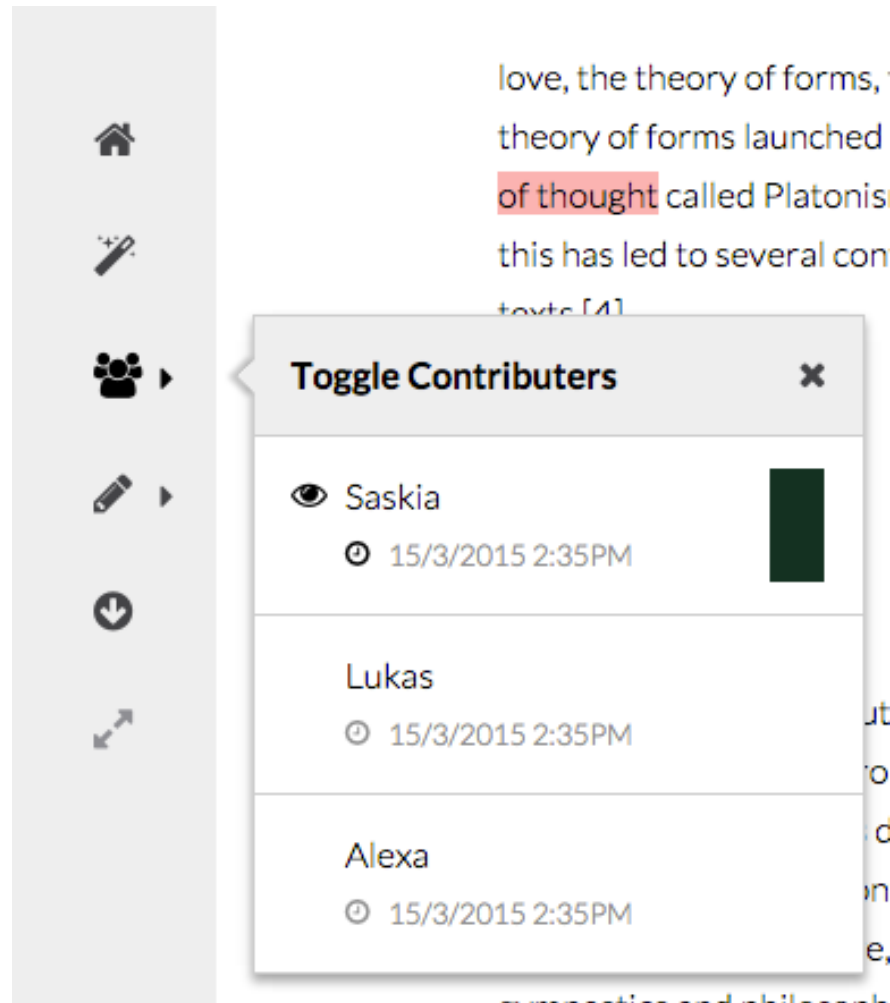


## Step 7a: Implementation “Switch Context”



video

## Step 7b: Implementation “List of Contributors”



**Thank You!**