



**POLITECHNIKA  
GDAŃSKA**

**Human Computer Interface**

Tomasz Kocejko



POLITECHNIKA  
GDAŃSKA

## Basic methods for interface evaluation



POLITECHNIKA  
GDAŃSKA

**WHY?**

To **check** whether the idea of interface  
is really **what user needs**

Whether the final product **works as  
expected**



POLITECHNIKA  
GDAŃSKA

## Usability

measures the **quality of a user's experience** when interacting with a product or system



- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction



## Why to do usability tests

- To identify product faults
- To identify users' needs
- To get data on performance
- To determine users' satisfaction



## Step 1: Create mockups

- Different type, different fidelity

## Step 2: Test mockups

- Cognitive walkthrough (.. specify task)
- Thinking aloud (... system and task)
- Heuristic approach (refers to the entire system)



POLITECHNIKA  
GDAŃSKA

# Step 1: Create mockups





Fuzzy layouts of general system requirements

Paper and/or digital based

- Sketches
- Stories
- mockups

Allow to gather information about basic functionalities or basic layout



## Advantages

- Cheap in production
- Help evaluate design ideas and design alternatives
- Focus on rapid iterative development
- GUI/layout presentation

## Disadvantages

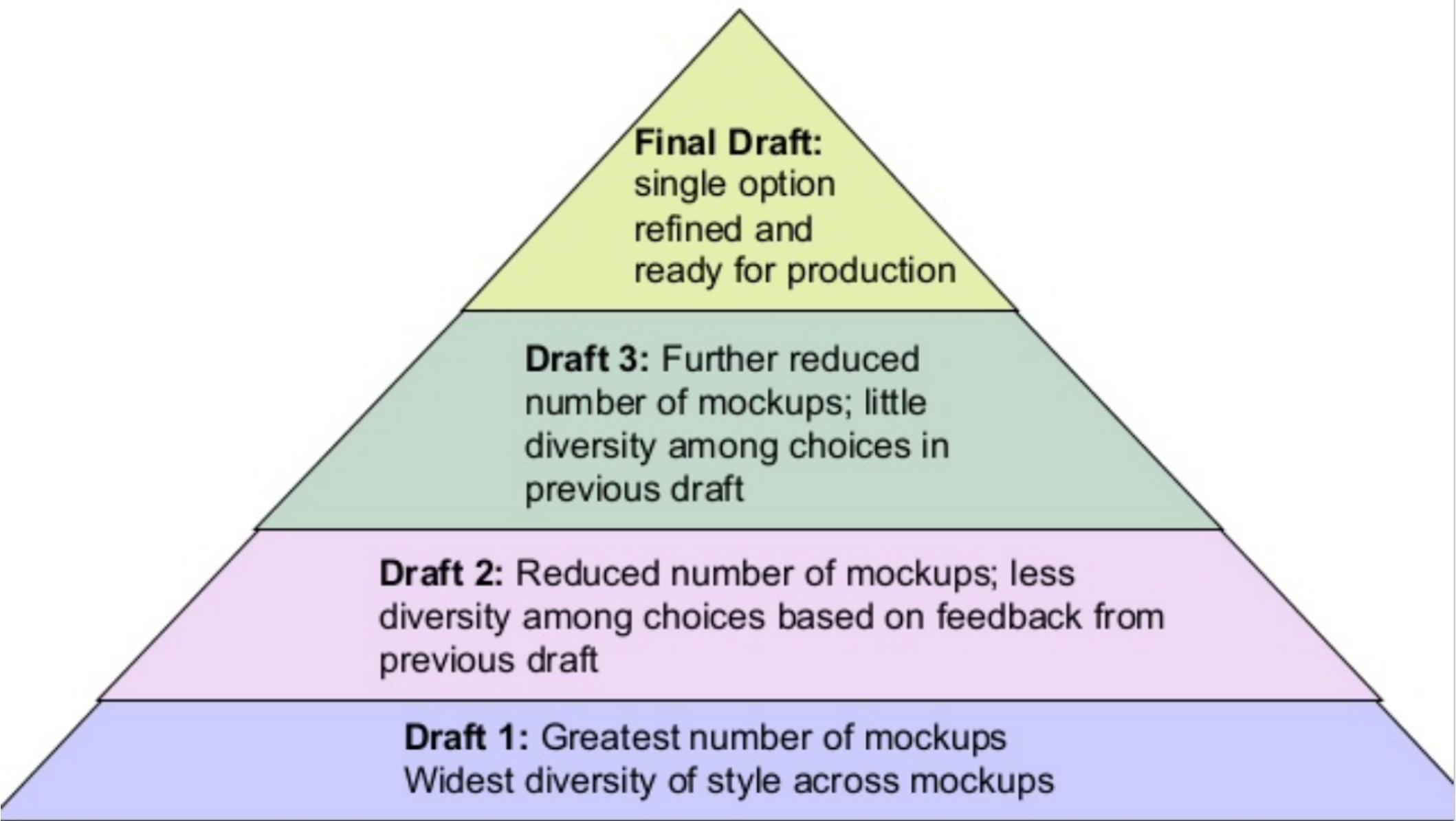
- Limited ability to check the errors
- Usually not documented. Difficult to program
- Human operator is needed to demonstrated the UI
- Not very useful after setting the general requirements for the system



- High quality
- Precisely reflects the digital version of the system
- Helps to gather the detailed information on the processes involved in moving between parts of the system

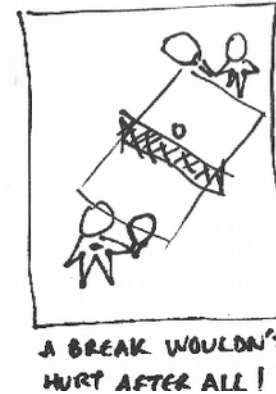
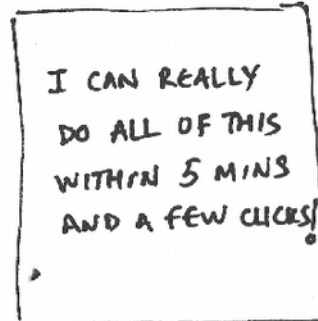
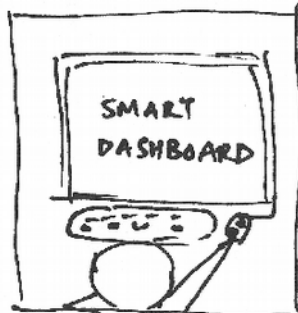
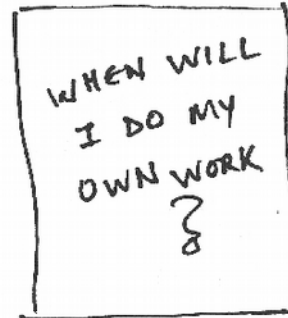
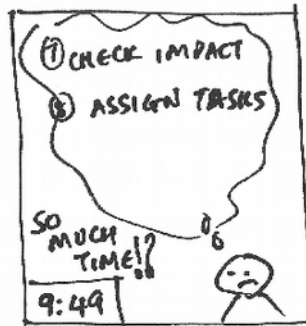
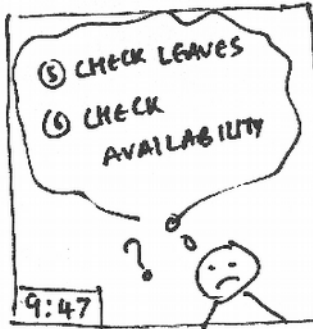
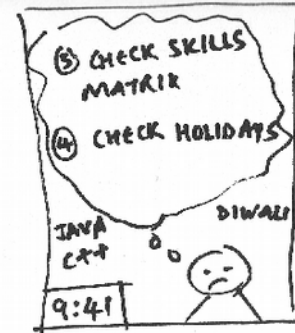
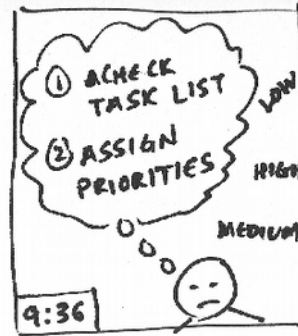


- Advantages
  - Demonstrates complete functionality
  - Shows the look and layout and behavior of the final product
  - Fully interactive, good for marketing
- Disadvantages
  - Takes time to create it
  - Difficult to change it during testing therefore not so effective during requirements gathering phase
  - Professional look sometimes discourage users to give comments





# storyboarding





- organizational tools that help create a rough visual representation of a final project (UI, app ...)
- is widely used in professional projects



## What should storyboards transfer?

- Setup
  - How many people should be involved
  - What is the environment
  - ...
- Order (what role UI has... not how many buttons it has)
  - What steps should be taken
  - Who should use the app
  - What task is presented
  - ...
- Satisfaction
  - What the app helps to accomplish
  - What makes the app „catchy”
  - What needs the app fulfills





## Benefits of storyboarding

- Helps to find out how the app/UI accomplishes a task
- Helps to present and check the idea/function of app/UI
- Helps to define the main goal of a project



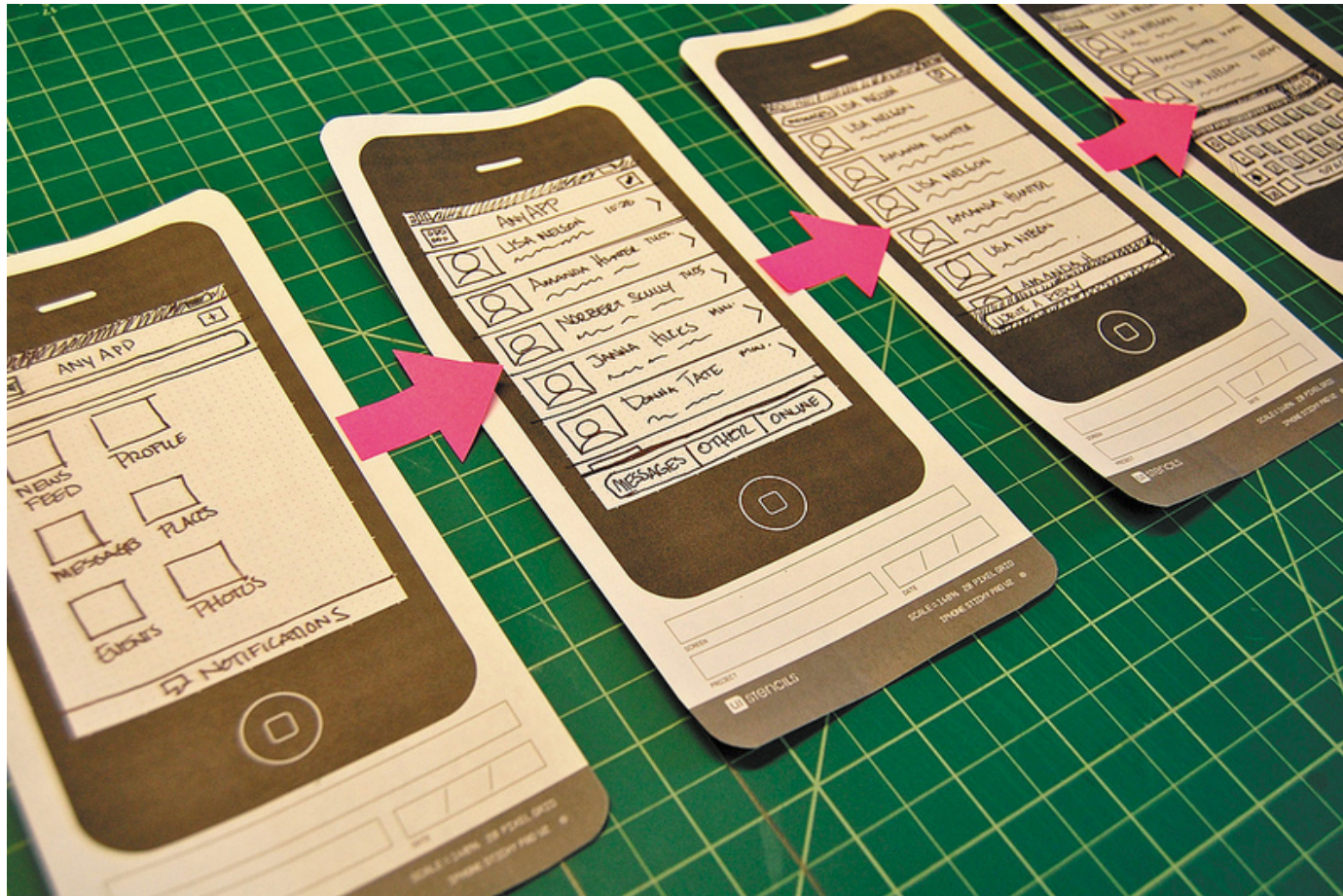
POLITECHNIKA  
GDAŃSKA

## Storyboards

- Do it fast
- Information (not design)
- Simplicity
- ...



# Paper prototype





POLITECHNIKA  
GDAŃSKA

## Paper mockups

DECO 7250

HUMAN-COMPUTER INTERACTION



## Paper prototyping tips

- Keep it organized
- Work quickly and make components
- If something is hard to simulate try to explain it to user verbally
- Mix (hardware+software)
- Add familiar components from your operating system



POLITECHNIKA  
GDAŃSKA

**Get creativ**

- Use different colors
- Use tape
- ...
- stickers



- Simulates users experience
- Cheap to create
- Easy to use
- Presents the idea of project design



- Cheap
- Fast
- Ties interface design to task
- Helps achieve common ground
- Video prototypes can be any fidelity





POLITECHNIKA  
GDAŃSKA





## What should it show

- Whole task (motivation and success)
- Illustrate important tasks of a system you are designing
- What can be changed



POLITECHNIKA  
GDAŃSKA

## Video prototype

- Focus on a message!!! ... not on a production value



- Set goals
  - Check to ensure that the final interface is consistent
  - Investigate how technology affects working practices
  - ...
- Choose the evaluation approach
  - Involve observation and interviews
  - Do not involve controlled tests in a laboratory
  - ...
- Evaluate...
- Analyze...
- Interpret and present the data
  - Reliability: can the study be replicated?
  - Validity: is it measuring what you expected?
  - Biases: is the process creating biases?
  - Scope: can the findings be generalized?



## Usability testing

- early
- Often
- Repeat...
- Repeat...
- ...

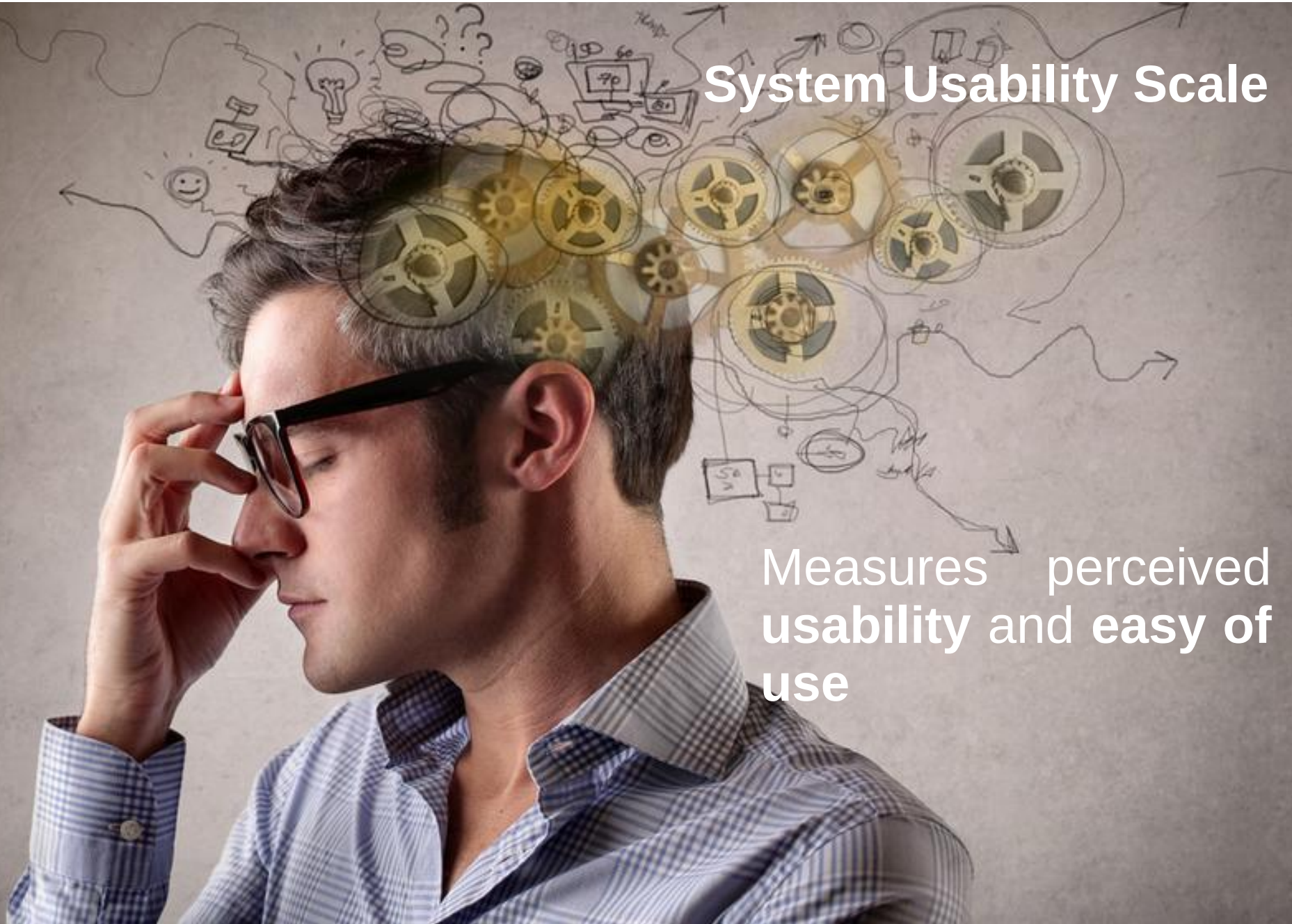




# Usability testing

## System Usability Scale

Measures perceived usability and easy of use





# Usability testing

Eight	2	( ) -1	2
Nine	4	5 - ( )	2
Ten	2	( ) -1	3
<b>TOTAL RAW SCORE</b>		5 - ( )	3
			3
			28

Total raw score *28* X 2.5 = *70* SUS score /100

Matching adjective  
= ..... *almost good* .....

- 92 = best imaginable
- 85 = excellent
- 72 = good
- 52 = Ok/fair
- 38 = poor
- 5 = worst imaginable

10 Statements  
five **positive** (odd numbers) and  
five **negative** (even numbers)  
Rate 1 (strongly disagree) to 5  
(strongly agree)



# Usability testing

Eight	2	( ) -1	2
Nine	4	5 - ( )	2
Ten	2	( ) -1	3
<b>TOTAL RAW SCORE</b>		5 - ( )	3
			3
			28

Total raw score *28* X 2.5 = *70* SUS score /100

Matching adjective  
= *almost good*

SUS is not enough  
**User-friendliness** was investigated with survey rated from 1 (Worst, totally unfriendly) to 7 (Best, super user-friendly)

- 92 = best imaginable
- 85 = excellent
- 72 = good
- 52 = Ok/fair
- 38 = poor
- 5 = worst imaginable



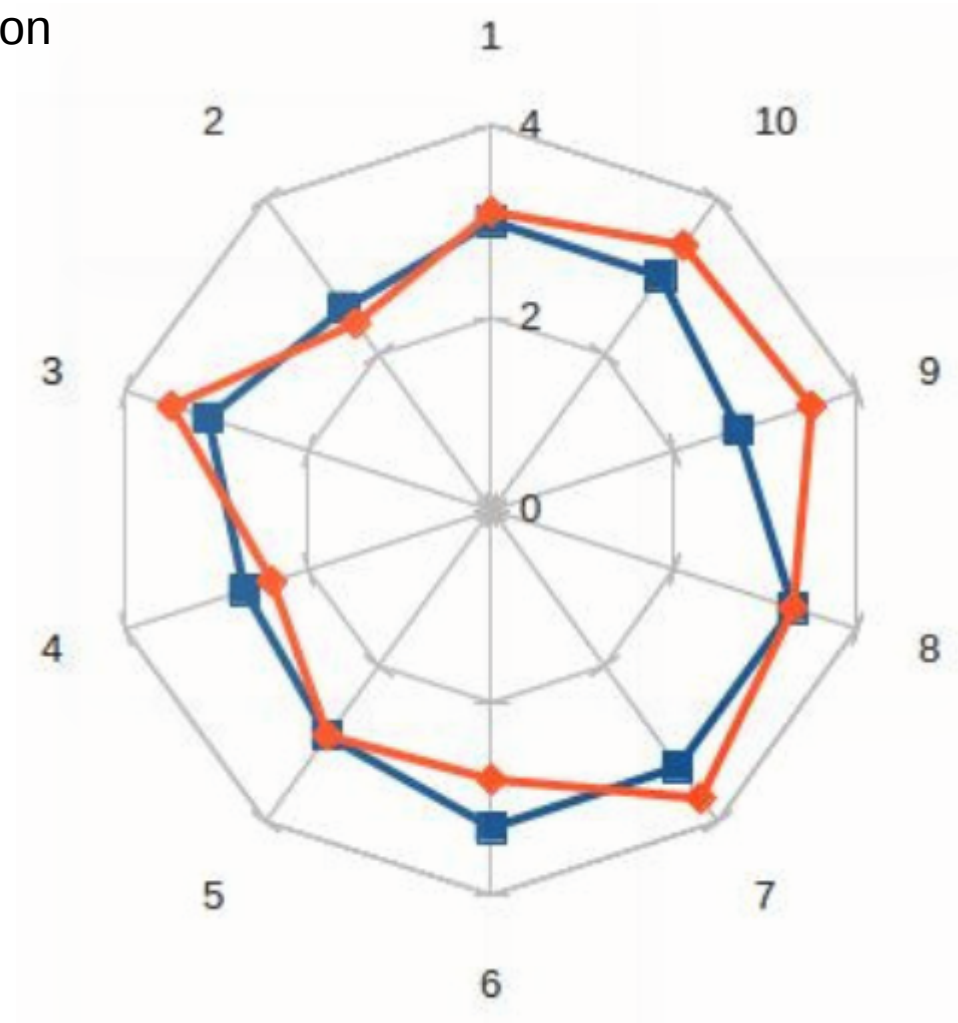


# Results

Blinking

Any gaze interaction

Participant	SUS score I	SUS score II
1	82.5	82.5
2	65	80
3	82.5	82.5
4	65	77.5
5	80	75
6	80	75
7	75	82.5
8	75	72.5
9	75	75
10	67.5	72.5





## Results – SUS interpretation

Participant	avg. SUS score	Friendliness
1	82.5	6
2	72.5	5
3	82.5	6
4	71.25	6
5	77.5	6
6	77.5	5
7	78.75	6
8	73.75	6
9	75	6
10	70	5





## Evaluation setting

- Laboratory based
  - Planned
  - Controlled
- Field study
  - Conducted in real situations
  - Typically less well controlled

## Data obtained

- Quantitative evaluation: typically objective
- Qualitative evaluation: typically subjective



## Usability inspection methods





- Heuristic evaluation
- Cognitive walkthrough
  - Group meetings
  - Simulates users's problem solving at each step through the dialogue
- Feature inspection
  - Experts knowledge required
  - Something users can't do
- Consistency inspection
- Standards inspection
- Formal usability inspection
  - procedurs with defined roles
  - Mix of heuristic evaluation and cognitive walkthroughs



**Find usability problems** in the design as early as it is possible

**Assign** each potential usability problem **to one or more heuristic**

**Assess** how much each usability issue could **impede user performance**



## Heuristics (by J.Nilsen)

1. Visibility of System Status
2. Match between system and real world
3. User Control and Freedom
  - „undo” „exit” „discard”
4. Consistency
  - Data should look the same every where in the system
5. Error prevention
6. Recognition over recall
7. Flexibility and Efficiency of use
  - Custom Installation vs Advance installation
8. Simple, aesthetic design
  - Avoid irrelevant info
  - Less content = more visibility
9. Help user diagnose and recover from errors
  - Suggest solution
  - Avoid error codes
10. Help and documentation
  - Call „F1” for help



## Heuristic evaluation benefits

- Inspection is usually less expensive than user observation
- During inspection, inspectors recommend solutions
- It prevents from discovering obvious design problems during user observations





Step by step explanation

Monitor the problems that user might experiencing

Discover mismatches between how the user thinks about the task vs how designer thinks about it



Basic idea (user have to verbalize the thoughts but not to express theirs ideas)

- „Click??? hmmm what that means!” (Good)
- „Click??? it should be click me” (Bad)



## Severity level of usability problems

Frequency (how often the problems occurs) - Is it common or rare

Impact of the problem if it occurs - easy or difficult for user to overcome?

Persistence of the problem - is it one time problem or will it appear repeatedly



## Severity level of usability problems

0 – not a problem at all

1 – cosmetic problem only (fix it if you have extra time in a project)

2 – minor usability problem (fix it, low priority)

3 – major usability problem (important to fix, high priority)

4 – catastrophe!!! (imperative to fix this before the release)



POLITECHNIKA  
GDAŃSKA

## Wizard of Oz

Simulates machine...  
... with human operators





# Make interactive applications

- Front end interface
- Wizard is in control of UI
- Only when it is faster/cheaper than real interface project



# Get feedback from users

- Hi-fidelity: users think it is real
- Low-fidelity...



Map out scenarios and app flow

Simulate unterface

Setup environment for wizard

Practice befor you run experiment

Recruit users

Users feedback... (how it should look like???)





**Fast**

**Easy** to create multiple variations

More **real** the paper prototype

User is in the center of development

Identifies bugs

Designer can play role of wizard



May simulate tech do not exist

Wizards needs training to be consistent

Playing wizard might be exhausting

... sometimes impossible to simulate



POLITECHNIKA  
GDAŃSKA

# Geo-space zoom techniques



POLITECHNIKA  
GDAŃSKA

