

session 6 - design

in the ucd cycle on the "determine design implications" step you use scenarios to go to the next step "design & prototyping"

Nielsen's heuristics

2: we're using conventions that a user can think of when they think about the action

3: user control and freedom: user has to have a choice to make a decision about what the software is supposed to do.

e.g. undo, redo, next, cancel buttons

6. you don't wanna rely on people's memory

5. restrict the format of phone numbers in a form, or double type your password

4. repeated things should act the same, color usage, placements, shapes

complete list:

aesthetics and minimalist design

consistency and standards

flexibility and efficiency

error prevention

recognition rather than recall

fail gently, help user to recognize, diagnose and recover from errors

error messages should be expressed in plain language (no codes). explain the problem and suggest a solution.

help and documentation

visibility of system status

match between system and the real world

the system should speak the user's language. things should look familiar to the user. follow real-world conventions.

user control and freedom

support redo, undo. allow for an emergency exit for the user at any time.

be decisive, but use **the smallest effective difference** (e.g. change the font just 1 level or 2 not 20 to imply a difference)

what are **design patterns**? it's a general way to solve a problem. e.g. navigation bars, scrollbars

why are they helpful? tested, efficient, people are familiar with it

the downsides of relying on design patterns? things won't match, **kills innovation** among designers,

breadcrumbs is used when there is a **hierarchy**

dropdown list: when **limited space**, categories

wizards: when there are **sub-tasks**

input feedback: the user expects to receive **feedback** on the result of a submission.

sketches/mockups are about **exploring design ideas**.

prototype: functional; can be tested as oppose to sketches/mockups

why?

cheaper and faster than actual building to test an idea

try out multiple designs

paper prototyping cons:

aesthetical elements **cannot be tested**

some **logistical issues**

problem of **professionalism**

low-fidelity: on paper, far from the final product

you don't fall in love with it

multiple alternatives

availability

cheap and fast

people are more comfortable give you feedback

mid-fidelity: still focus on **structure** and NOT "feel and look"

usually done with the use of a computer

use dummy text

use color sparingly (moderately)

our motto: "fake it before make it!"