Design heuristics

- Design heuristics is a set of rules and advise about good and bad design solutions, mainly based on practical experiences.
- The design aspects are divided into two sections. General and application specific aspects.
- The general aspects are possible to evaluate by a design aspect, without involvement of the users.
- The application specific aspects can only be evaluated with active participation by the users.
- We give very few practical examples here, which actually are needed to fully understand the aspects.

General design aspects

1. Disposition of the screen area
2. Orientation and navigation
3. Control and feedback
4. Input
5. Readability
6. Errors and help
7. Use of colours
8. Subjective look
1. Disposition of the screen area

• The screen area is a limited and valuable resource
• Define well separated main areas.
• Use fixed and well planed positions.
• Ovoid overlapping windows that hide information.
• Highlight important parts, e.g. data more then lines etc.
• Create a compact image, avoid large empty areas.
• It is seldom the amount of information that cause problems for the user, provided that the design is well adapted to the situation.

2. Orientation and navigation

• You must always know where you are and where you are going.
• You should always know where you are “without thinking”.
• You should always see where you are in a list och in a bundle.
• Show overview and details in parallel.
• Show if there are more information available, even if it is not visible just now.
• Show how the user can reach hidden information.
• Turning of pages is better then scrolling, especially when reading texts.
3. Control and feed-back

• The user should be in control of the dialogue, not the system.
• The user must be able to choose how, and in which sequence, tasks are performed.
• The user should always know what happens.
• Always provide feed-back to the user about everything that happens.
• Feed-back is important both for efficiency and for safety.
• Always show expected waiting time.

4. Input

• Input concerns data, parameters, menus, commands etc.
• Minimize all input. Input is time consuming, and requires attention.
• Avoid mixed key-board and mouse activities.
• Input via key-board is always faster then mouse.
• Show if data fields are input fields or not.
• Use broad instead of deep (many levels) menus.
• Use default values, short commands, shortcuts etc as often as possible.
5. Errors and help

- Allow the user to make errors without serious effects. Users will make errors if this is possible.
- Always provide “undo” possibilities.
- Give clear error messages when errors are made.
- Give help if needed.
- If help is needed often it is a sign of bad design.

6. Readability and layout

- Group data logically, so that they can be easily identified.
- Group using closeness, colour, fonts, frames etc.
- Group numbers in columns instead of in rows.
- Use readable fonts, large enough.
- Emphasize what is important, e.g. data instead of frames etc. The eye will focus on elements with the highest contrast.
- Use colours in a clever way (below).
7. Use of colours

- Design in grey scale first, then add colours.
- 5% of all users have defects in colour vision.
- Colours should transmit a message, not be cosmetics.
- Used unsaturated colours except for very important data.
- Use colours to code information. E.g. status, levels etc.
- Use few colours, max 5-7 different.
- Use an harmonic colours as background, e.g. grey, dark blue etc.
- Avoid interfering colours, e.g. red-green.
- Use maximal contrast when reading texts (black on white).
- If a grey background is used, both black and white text can be presented.

8. Subjective look

- Create a subjectively pleasing design!
- Use colours with sense.
- Place objects on the screen so that it does not look messy.
- The total image should have balance.
Application specific aspects

1. Process of user interaction
2. Simultaneous presentation
3. Emphasise what is important
4. Quick commands and shortcuts
5. Speak the language of the user

1. Process of user interaction

• Choose a metaphor that is functional for the users (desktop or room).
• Does the interface support the work activities, Are all necessary functionality available? Is there any unnecessary information?
• Is the information logically grouped? Does the interface support a good work flow, e.g. top to down, left to right?
• Emphasize for the tasks important information.
2. Simultaneous presentation

- All information needed for a decision must be visible simultaneously.
- To change view during a work task will use short term memory.
- Avoid overlapping windows.
- Forms for information presentation and writing must be visible simultaneously.
- It must be clear how hidden information can be reached.

3. Emphasise what is important

- Data are more important than headlines and should be emphasized.
- Code information using colour, form, font, size etc.
- Indicate when needed the status of the information (e.g. salary > € 50 000 coded red)
4. Quick commands and shortcuts

- Short commands are efficient for experienced users.
- Key-stroke is faster than mouse-click.
- It must be possible to go directly to a selected page, using shortcuts.
- Also going back must be able using shortcuts.

5. Speak the language of the user

- Use the terminology of the users.
- Avoid “computer language”.
- Design icons etc so that they relate to the work process.
- Select colours that relates to the work environment.