# USABILITY HEURISTICS GUIDE

# A User Interface Assesment Instrument for Novice Evaluators

### (Based on *Usability Engineering* by Jakob Nielsen)

## 1. Simple and Natural Dialogue

- Mapping
  - √ Is the mapping between computer concepts and user concepts as simple as possible? **(115)**
  - √ Is the navigation through the user interface minimized? **(115)**
  - √ Is the information provided exactly the information the user needs (and no more) at exactly the time and place where it is needed? **(116)**
  - √ Are the information objects and operations accessed in a sequence that matches the way users will most effectively and productively do things? **(116)**
- Graphic Design and Colors
  - √ Is the information that will be used together displayed close together or at least in the same screen? **(117)**

    *Wisdom: Grouping could be achieved with dividing lines, boxes, colors, etc.*
  - √ Do the most important dialogue elements stand out? **(118)**

    *Wisdom: Blinking should only be used in extreme cases (it is distracting and annoying).*
  - √ Does the interface looks like an angry fruit salad of wildly contrasting, highly saturated colors? **(119)**
  - √ Does the interface uses more than 7 colors? **(119)**
  - √ Could the interface be used without colors? **(119)**

    *Wisdom: Colors should be used to categorize, differentiate and highlight.*
- Less is More
  - √ Is irrelevant information presented in the interface? **(120)**
  - √ Is less important information left for auxiliary screens? **(120)**
  - √ Are too many choices provided? **(121)**

    *Wisdom: Use training wheels approach - provide multiple nested levels of increased complexity (for novice users to expert users).*

## 2. Speak the Users' Language

√ Is the terminology used based on the users' language (not on system oriented terms)? **(123)**

√ Are words in nonstandard meanings used? **(123)**

√ Are interaction messages stated from the users' perspective? **(124)**

√ Are there restrictions on the names the users assign to objects (i.e. limits on the number of characters that can be specified)? **(124)**

*Wisdom: Use vote winning terms (words).*

*Wisdom: Allow for rich use of synonyms in interpreting command languages and in documentation indexes.*

√ Is there a good mapping between the computer display of information and the user's conceptual model of the information (the one million dollar question)? **(126)**

*Wisdom: When using metaphors care should be taken to present the metaphor as a simplified model of a more detailed conceptual model of the system and not as a direct representation of the system.*

## 3. Minimize User Memory Load

*Wisdom: In general, people have much easier time at recognizing something that is shown to them than they have at having to recall the same information from memory without help.*

√ Are dialogue elements displayed to the users so that they can choose from them or edit them (menus, lists of objects, dialogue boxes, save as, etc.)? **(129)**

*Wisdom: Whenever a user is asked to provide input, the system should describe the required format and, if possible, provide an example of legal an sensible input, such as a default value (Left margin: 10 points [0-128]).*

√ Are generic commands used? **(131)**

*Wisdom: The system should be based on a small number of pervasive rules that apply throughout the user interface (cut & paste text/objects/etc., point and click, drag and drop, window management, select command from menu, etc.*

## 4. Consistency

*Wisdom: Users feel more confident in using the system if they know that the same command or the same action will always have the same effect.*

√ Is the same information presented in the same location on all screens and dialogue boxes? Is it formatted in the same way? **(132)**

√ Is the interface standard being followed? **(133)**

## 5. Feedback

√ Does the system continuously inform the user about what it is doing and how it is interpreting the user's input? **(134)**

√ Is feedback expressed in abstract and general terms? **(134)**

*Wisdom: Feedback should restate and rephrase the user's input to indicate what is being done with it (i.e. when overwriting a file).*

√ Is the degree of persistence of the feedback appropriate? **(134)**

√ Is feedback provided according to response time of operations? **(135)**

*Wisdom: No special feedback is needed for delays less than 1 second.*

*Feedback for events lasting more than 10 seconds should have a high degree of persistence (use percent done indicators).*

*Do not use percent done feedback for operations taking between 2 to 10 seconds.*

√ Is feedback provided in case of a system failure? **(137)**

## 6. Clearly Marked Exits

√ Does the system provides the user an easy way out of as many situations as possible (cancel, undo, etc.) **(138)**

*Wisdom: A basic principle for user interface design should be to acknowledge that users will make errors no matter what else is done to improve the interface, and one should therefore make it as easy as possible to recover from these errors.*

*Wisdom: In general, interfaces should show a high degree of responsiveness, to the extent that paying attention to the user new actions should get higher priority than finishing the user's old actions.*

√ Do the exit and undo mechanisms visible (not a special code or an obscure combination of keys)? **(139)**

## 7. Shortcuts

√ Are users allowed to jump directly to the desired location in large information spaces? **(140)**

√ Are users allowed to reuse their interaction history (i.e. list of documents previously accessed)? **(141)**

*Wisdom: The following are shortcuts to consider:*

- *function keys*
- *macros*
- *templates*
- *wizards*
- *double clicking*
- *type-ahead*
- *click-ahead*
- *repeat last command*
- *repeat last operation*
- *"comic strips"*
- *working groups*
- *list files by date*
- *default values*

## Good Error Messages

*Wisdom: Error messages should:* **(142)**

- *be phrase in clear language and avoid obscure codes*
- *be precise rather than vague or general*
- *constructively help the user solve the problem.*
- *be polite and should not intimidate the user or put the blame explicitly on the user*
- *avoid abusive terms such as fatal, illegal, etc.*

## Prevent Errors

√ Are there mechanisms to prevent errors due to: **(145-148)**

- spelling names of objects or operations
- case sensitive text
- operations with serious consequences
- use of modes
- other situations

## Help and Documentation

*Wisdom: The fundamental truth about documentation is that most users simply do not read the manuals.*

*The quality of help texts is far more important than the mechanisms by which those texts are accessed.*

√ Does the system provides documentation? **(148)**

√ Does the system provides online help? **(148)**

√ Is a "minimal manual" provided (one that only gives whatever information is absolutely necessary in order for the users to get started using the system with common tasks)? **(154)**