

Appendix A

Usability Guideline Sets

This appendix presents the list of the sets of guidelines that were collected and studied.

A.1 Ten Usability Heuristics (Nielsen, 2004)[7].

1. **Visibility of system status** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use** Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. Help and documentation Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

A.2 8 golden rules of Interface Design (Shneiderman, 1998)[9].

1. Strive for Consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer error prevention and simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

A.3 Usability Maxims (Lund, 1997)[6].

1. Know thy user and you are not thy user
2. Things that look the same should act the same
3. The information for the decision needs to be there when the decision is needed
4. Error messages should actually mean something to the user and tell the user how to fix the problem
5. Every action should have a reaction
6. Everyone makes mistakes, so every mistake should be fixable.
7. Don't overload the user's buffers.
8. Consistency, Consistency, Consistency (alternative version of 2)
9. Minimise the need for a mighty memory (alternative of 7)
10. Keep it simple
11. The user should always know what is happening.
12. The more you do something the easier it should be to do.
13. The user should control the system. The system shouldn't control the user. The user is the boss, and the system should show it.

14. Eliminate unnecessary decisions, and illuminate the rest.
15. The best journey is the one with the fewest steps. Shorten the distance between the user and the goal.
16. The user should be able to do what the user wants to do
17. If I made an error, let me know about it before I get into REAL trouble.
18. You should always know how to find out what to do next.
19. The idea is to empower the user, not speed up the system.
20. Things that look different should act different.
21. Don't let people accidentally shoot themselves
22. Even experts are novices at some point. Provide help.
23. Keep it neat. Keep it organized.
24. Provide a way to bail out and start over.
25. Design for regular people and the real world.
26. The fault is not in thyself, but in they system.
27. Colour is information.
28. If it is not needed, its not needed.
29. Everything in its place, and a place for everything.
30. The user should be in agood mood when done.
31. If I made an error, at least let me finish my thought before I have to fix it.
32. Let people shap the system to themselves and pain it with their own personality.
33. "Cute" is not a good adjective for systems.
34. To know the system is to love it.

A.4 User engineering principles for interactive systems (Hansen, 1971) [3].

1. *1 First principle: Know the user*
2. *2 Minimise memorisation*
3. *2a Selection not entry*
4. *2b Names not numbers*
5. *2c Predictable behavior*

6. *2d* Access to system information
7. *3* *Optimise operations*
8. *3a* Rapid execution of common operations
9. *3b* Display inertia
10. *3c* Muscle memory
11. *3d* Reorganize command parameters
12. *4* *Engineer for errors*
13. *4a* Good error messages
14. *4b* Engineer out the common errors
15. *4c* Reversible actions
16. *4d* Redundancy
17. *4e* Data structure integrity

A.5 Interface design for time-sharing systems (Cheriton, 1976)[1].

1. *Simple* Project a 'natural', uncomplicated 'virtual' image of the system.
2. *Responsive* Respond quickly and meaningfully user commands.
3. *User-controlled* All actions are initiated and controlled by the user.
4. *Flexible* Flexibility in common structure and tolerance or errors.
5. *Stable* Able to detect user difficulties and assist him in returning to correct dialog; never deadening' the user (ie, offering (no recourse)).
6. *Protective* Protect the user from costly mistakes or accidents (eg, overwriting a file).
7. *Self-documenting* The commands and system responses are self-explanatory and documentation, (explanations or tutorial material are part of the environment).
8. *Reliable* Not conducive to undetected errors in man-computer communication.
9. *User-modifiable* Sophisticated users are able to personalize environment.

A.6 Design guidelines for interactive systems (Gaines and Facey, 1975)[2].

1. Introduce through experience.
2. Immediate feedback.
3. Use the user's model.

4. Consistency and uniformity.
5. Avoid acausality.
6. Query-in-depth (tutorial aids).
7. Sequential - parallel tradeoff (allow choice of entry patterns).
8. Observability and controllability.

A.7 Ground rules for a 'well-behaved' system (Kennedy, 1974). [5].

1. Use terse 'natural' language, avoid codes, allow abbreviations.
2. Use short entries to facilitate error correction and maintain tempo.
3. Allow single or multiple entries to match user ability.
4. Maintain 'social element' to the communication.
5. Permit user to control length of cues or error messages.
6. Error messages should be polite, meaningful and informative.
7. Give help when requested or when errors are in difficulty.
8. Simple, logically consistent command language.
9. Control over all aspects of the system must appear to belong to the user.
10. Avoid redundancy in dialog.
11. Adapt to the user's ability.
12. Keep exchange rate in user's stress-free range; user can control rate.

A.8 Design guidelines for interactive systems (Pew and Rollins, 1975)[8].

1. Know the user population.
2. Respond consistently and clearly.
3. Carry forward a representation of the user's knowledge base.
4. Adapt wordiness to user needs.
5. Provide the users with every opportunity to correct their own errors.
6. Promote the personal worth of the individual user.

A.9 The design of idiot-proof interactive programs (Wasserman, 1973).[11].

1. Provide a program action for every possible type of user input.
2. Minimize the need for the user to learn about the computer system.
3. Provide a large number of explicit diagnostics, along with extensive online user assistance.
4. Provide program short-cuts for knowledgeable users.
5. Allow the user to express the same message in more than one way.

A.10 The human-machine interface in a computerized conferencing environment (Turoff, Whitescarver and Hiltz, 1977)[10].

1. *Forgiveness* ease in repairing errors.
2. *Segmentation* layered approach.
3. *Variety* choice of style.
4. *Escape* break out of danger.
5. *Guidance* direction and learning.
6. *Leverage* flexible, powerful features.

A.11 Human-computer interaction and CAI: A review and research prospectus (John G Hedberg and Neil R Perry, 1985)[4].

1. User characteristics.
2. Simplicity.
3. Flexibility.
4. User control and feedback.
5. Error messages.
6. Format of materials.