[1] Simple and Natural Dialogue

Dialogues should not contain irrelevant or rarely needed information. Every extraneous unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.

[2] Speak the User's Language

The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.

[3] Minimize the User's Memory Load

The user's short-term memory is limited. 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. Complicated instructions should be simplified.

[4] Be Consistent

Users should not have to wonder whether different words, situations or actions mean the same thing. A particular system action - when appropriate - should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user populations.

[5] Provide Feedback

The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.

[6] Provide Clearly Marked Exits

A system should never capture users in situations that have no visible escape. 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.

[7] Provide Shortcuts

The features that make a system easy to learn - such as verbous [verbose] dialogues and few entry fields on each display - are often cumbersome to the experienced user. Clever shortcuts - unseen by the novice user - may often be included in a system such that the system caters to both inexperienced and experienced users.

[8] Provide Good Error Messages

Good error messages are defensive, precise and constructive. Defensive error messages blame the problem on system deficiencies and never criticize the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.

[9] Error Prevention

Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

[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.

Appendixes A to F