

Mayhew's User Interface Design

SJSU
2005

Mayhew, D. (2002). Principles and Guidelines in Software User Interface Design

Deborah J. Mayhew

- Deborah J. Mayhew is President of Deborah Mayhew Associates.
- Developer of usability engineering, principles for multimedia software design.



1 User compatibility

- Perhaps the most fundamental principle, from which all others are derived, is to know the user. Designers should be familiar with cognitive psychology, or an understanding of the general strengths and weaknesses of the human mind so as to make the design acceptable to most users, not only a few of them.

2 Product compatibility

- Often the intended user of a new system is already a user of other systems, who has already invested a great deal of time and/or money in learning the existing systems. The across-product compatibility is important since it allows the user to adapt to a new system quickly.

3 Task compatibility

- The structure and flow of a system should match and support the task that is being carried out. The user doesn't need to navigate back and forth between applications in order to complete a task.

4 Work flow compatibility

- A system should be organized to facilitate transitions between tasks. For example, a windowed, multitasking system can support the nature of this type of user's work more effectively.

5 Consistency

- Consistency refers to similarities within a product, rather than across products. It allows people to reason by analogy and predict how to do things they have never done before.

6 Familiarity

- Concepts, terminology, and spatial arrangements that the user is already familiar with can be incorporated into the interface.

7 Simplicity

- Don't try to provide all the functionality that any user could possibly ever want or need. Instead, make the interface relatively simple.

8 Direct manipulation

- A direct manipulation interface is one in which users directly perform actions on visible objects. This is in contrast to interfaces in which users specify actions, parameters, and objects indirectly through language.

9 Control

- User prefer to feel a sense of mastery and control over any tool at their disposal, and the computer is no exception. The designer should be sensitive to this and present a tool-like interface.

10 WYSIWYG

- Try to make what you see on the screen is what you get on printed output or stored files.

11 Flexibility

- Allow more user control and accommodates variations in user skill and preferences.

12 Responsiveness

- Give users feedback as soon as possible and let users be aware of the progress.

13 Invisible technology

- Hide the technology from users. Only present the functionality that users need to know.

14 Robustness

- A system should tolerate common and unavoidable human error. System crashes should be minimized, and simple to understand and execute recovery measures should be presented. A robust system encourages users to learn new features and thus increases productivity.

15 Protection

- People make errors, especially when they are working quickly or under pressure. Users should be protected against the catastrophic results of common human error. "Undo" or other recovery measures should also be provided.

16 Ease of learning & ease of use.

- System should be both easy to learn for the novice and efficient and easy to use for the expert. Users should be able to get sufficient help information when they get confused.