Some Sets of Design Principles

Several authors have produced sets of design principles for different purposes, and with different kinds of systems in mind.

**Jakob Nielsen’s 113 Heuristics for Webpage Usability**

The most comprehensive publicly available useful resource is Jakob Nielsen’s collection of 113 heuristics for usable website homepages. Nielsen and colleagues have produced comparable sets of guidelines for other types of system, but charge for them.

<http://www.nngroup.com/articles/113-design-guidelines-homepage-usability/>

**Jakob Nielsen’s list of 10 basic heuristics**

Jakob Nielsen is primarily concerned with user interfaces for web based systems and consumer products. He claims that nearly all usability problems are caused by design failures that fall into one of these broad categories. But they are rather abstract and hard to map to design decisions and interface features.

**Visibility of system status**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

**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.

**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.

**Consistency and standards**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

**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.

**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.

**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.

**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.

**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.

**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.

**Ben Shneiderman’s Eight Golden Rules**

The classic list, from the first edition of *Designing the User Interface*.

**1 Strive for consistency.**
Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

**2 Enable frequent users to use shortcuts.**
As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

**3 Offer informative feedback.**
For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

**4 Design dialog to yield closure.**
Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

**5 Offer simple error handling.**
As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

**6 Permit easy reversal of actions.**
This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

**7 Support internal locus of control.**
Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

**8 Reduce short-term memory load.**
The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

**Weinschenk and Barker’s summary list of 20 basic heuristics**

Susan Weinschenk and Dean Barker (Weinschenk and Barker 2000) researched usability guidelines and heuristics from many sources (including Nielsen's, Apple and Microsoft) and did a massive card sort to generate the list of 20 below.

1. **User Control**: The interface will allow the user to perceive that they are in control and will allow appropriate control.
2. **Human Limitations**: The interface will not overload the user's cognitive, visual, auditory, tactile, or motor limits.
3. **Modal Integrity**: The interface will fit individual tasks within whatever modality is being used: auditory, visual, or motor/kinesthetic.
4. **Accommodation**: The interface will fit the way each user group works and thinks.
5. **Linguistic Clarity**: The interface will communicate as efficiently as possible.
6. **Aesthetic Integrity**: The interface will have an attractive and appropriate design.
7. **Simplicity**: The interface will present elements simply.
8. **Predictability**: The interface will behave in a manner such that users can accurately predict what will happen next.
9. **Interpretation**: The interface will make reasonable guesses about what the user is trying to do.
10. **Accuracy**: The interface will be free from errors
11. **Technical Clarity**: The interface will have the highest possible fidelity.
12. **Flexibility**: The interface will allow the user to adjust the design for custom use.
13. **Fulfilment**: The interface will provide a satisfying user experience.
14. **Cultural Propriety**: The interface will match the user's social customs and expectations.
15. **Suitable Tempo**: The interface will operate at a tempo suitable to the user.
16. **Consistency**: The interface will be consistent.
17. **User Support**: The interface will provide additional assistance as needed or requested.
18. **Precision**: The interface will allow the users to perform a task exactly.
19. **Forgiveness**: The interface will make actions recoverable.
20. **Responsiveness**: The interface will inform users about the results of their actions and the interface's status.

**Jill Gerhardt-Powals’ Cognitive Engineering Principles**

Jill Gerhardt-Powals (1996) presented a set of ten principles for cognitive engineering to improve human performance using computer systems; this list is oriented towards people doing relatively complicated information handling (her example domain was anti-submarine warfare. She attributes the first three principles to R.M. Taylor (1989) and principles four to nine to E.D. Murphy and C.M. Mitchell (1986).

Principle 1. **Automate unwanted workload.**

Eliminate mental calculations, estimations, comparisons, and any unnecessary thinking, to free cognitive resources for high-level tasks.

Principle 2. **Reduce uncertainty.**

Display data in a manner that is clear and obvious to reduce decision time and error.

Principle 3. **Fuse data.**

Bring together lower level data into a higher level summation to reduce cognitive load.

Principle 4. **Present new information with meaningful aids to interpretation.**

New information should be presented within familiar frameworks (e.g., schemas, metaphors, everyday terms) so that information is easier to absorb.

Principle 5. **Use names that are conceptually related to function.**

Display names and labels should be context-dependent, which will improve recall and recognition.

Principle 6. **Group data in consistently, meaningful ways.**

Within a screen, data should be logically grouped; across screens, it should be consistently grouped. This will decrease information search time.

Principle 7. **Limit data driven tasks.**

Use color and graphics, for example, to reduce the time spent assimilating raw data.

Principle 8. **Include in the displays only that information needed by the operator at a given time.**

Exclude extraneous information that is not relevant to current tasks so that the user can focus attention on critical data.

Principle 9. **Provide multiple coding of data.**

The system should provide data in varying formats and/or levels of detail in order to promote cognitive flexibility and satisfy user preferences.

**Principle 10. Practice judicious redundancy.**

Principles 6 and 8 may be in conflict, that is, in order to be consistent, it is sometimes necessary to include more information than may be needed at a given time. Allow redundancy to be consistent.HHHCH