

# Usability Heuristics

## Chris Nordling

### **Comfort and Control**

Offer users stable perceptual cues for a sense of "home." Make Actions reversible. Always allow "Undo." Always allow the user an 'out' but make it easier to stay 'in'. Ensure that users never lose their work as a result of error on their part, the vagaries of Internet transmission, or any other reason other than the completely unavoidable, such as sudden loss of power to the client computer. Communicate that level of safety.

### **Learnability**

Ideally, products would have no learning curve: users would walk up to them for the very first time and achieve instant mastery. In practice, all applications and services, no matter how simple, will display a learning curve. Ensure the interface hits the learning curve target for your audience.

### **Visibility of system status**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. Use status mechanisms to keep users aware and informed. Keep status information up to date and within easy view. Avoid invisible navigation.

### **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.

### **Consistency and standards**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. Inconsistency: It is just important to be visually inconsistent when things must act differently as it is to be visually consistent when things act the same. Make objects consistent with their behavior. Make objects that act differently look different.

### **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.

**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.

**Anticipate User Actions**

Applications should attempt to anticipate the user's wants and needs. Do not expect users to search for or gather information or evoke necessary tools. Bring to the user all the information and tools needed for each step of the process. Look at the user's productivity, not the computer's; Keep the user occupied. Wherever possible, use multi-threading to push latency into the background.

**Color**

Any time you use color to convey information in the interface, you should also use clear, secondary cues to convey the information to those who won't be experiencing any color coding today.

Text that must be read should have high contrast. Favor black text on white or pale yellow backgrounds. Avoid gray backgrounds.

**Defaults**

Defaults should be easy to "blow away." Fields containing defaults should come up selected, so users can replace the default contents with new material quickly and easily. Defaults should be "intelligent" and responsive.

Do not use the word "default" in an application or service. Replace with "Standard," "Use Customary Settings," "Restore Initial Settings," or some other more specific terms describing what will actually happen.

**Labeling**

Menu and button labels should have the key word(s) first. Use font sizes that are large enough to be readable on standard monitors. Favor particularly large characters for the actual data you intend to display, as opposed to labels and instructions.

**Editing**

Remove any element of the application that is not helping.