# Eight Golden Rules of Interface Design

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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

**More Guidelines**

**Aesthetics**

* **Principle: Aesthetic design should be left to those schooled and skilled in its application: Graphic/visual designers**
* **Principle: Fashion should never trump usabilit**y
* **Principle: User test the visual design as thoroughly as the behavioral design**

**Anticipation**

* **Principle: Bring to the user all the information and tools needed for each step of the process**

**Autonomy**

* **Principle: The computer, interface, and task environment all “belong” to the user, but user-autonomy doesn’t mean we abandon rules**
* **Principle: Enable users to make their own decisions, even ones aesthetically poor or behaviorally less efficient**
* **Principle: Exercise responsible control**
* **Principle: Use status mechanisms to keep users aware and informed**
* **Principle: Keep status information up to date and within easy view**
* **Principle: Ensure status information is accurate**

**Color**

**Color blindness**

* **Principle: 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 cannot see the colors presented.**
* **Principle: Test your site to see what color-blind individuals see**

**Color as a vital interface element**

* **Principle: Do not avoid color in the interface just because not every user can see every color.**
* **Principle: Do not strip away or overwhelm color cues in the interface because of a passing graphic-design fad.**

**Consistency**

**1) Levels of Consistency**

* **Principle: The importance of maintaining strict consistency varies by level.**

**1. level consistency**

• **Platform consistency:** Be generally consistent with [*de jure*As dictated by guidelines and standards](http://asktog.com/atc/category/human-computer-interaction-hci/hci-design/principles-of-hci-design/) & [*de facto*The unwritten rules to which the community adheres.](http://asktog.com/atc/category/human-computer-interaction-hci/hci-design/principles-of-hci-design/) standards

•**In-house consistency:** Maintain a general look & feel across your products/services

Communicates brand and makes adoption of your other products and services easier and faster

**2. Consistency across a suite of products, e. g., Microsoft Office**

General look & feel communicates family

**3. The overall look & feel of a single app, application or service–splash screens, design elements, etc.**

**4. Small visible structures, such as icons, symbols, buttons, scroll bars, etc.**

**5. Invisible structures**

**6. Interpretation of user behavior**

**2) Induced Inconsistency**

* **Principle: It is just important to be visually inconsistent when things act differently as it is to be visually consistent when things act the same**

**3) Continuity**

* **Principle: Over time, strive for continuity, not consistency**

**4) Consistency with User Expectation**

* **Principle: “The most important consistency is consistency with user expectations”—William Buxton**

**Defaults**

* **Principle: Defaults within fields should be easy to “blow away”**
* **Principle: Defaults should be “intelligent” and responsive**
* **Principle: Replace the word “default” with a more meaningful and responsive term**
* **Principle: Both your vocabulary and visual design must communicate the scope of a reversion**

**Discoverability**

* **Principle: Any attempt to hide complexity will serve to increase it**
* **Principle: If you choose to hide complexity, do so in the showroom only**
* **Principle: If the user cannot find it, it does not exist**
* **Use Active Discovery to guide people to more advanced features**
* **Principle: Controls and other objects necessary for the successful use of software should be visibly accessible at all times**
* **Principle: There is no “elegance” exception to discoverability**
* **Principle: With the exception of small mobile devices, controls do not belong in the middle of the content area**
* **Principle: Communicate your gestural vocabulary with visual diagrams**
* **Principle: Strive for Balance**
* **Principle: User-test for discoverability**

**Efficiency of the User**

* **Principle: Look at the user’s productivity, not the computer’s**
* **Principle: Keep the user occupied**
* **Principle: To maximize the efficiency of a business or other organization you must maximize everyone’s efficiency, not just the efficiency of the IT department or a similar group**
* **Principle: The great efficiency breakthroughs in software are to be found in the fundamental architecture of the system, not in the surface design of the interface**
* **Principle: Error messages should actually help**

**Explorable Interfaces**

* **Principle: Give users well-marked roads and landmarks, then let them shift into four-wheel drive**
* **Principle: Sometimes you do have to provide deep ruts**
* **Principle: Offer users stable perceptual cues for a sense of “home”**
* **Principle: Make Actions reversible**
* **Principle: Always allow “Undo”**
* **Principle: Always allow a way out**
* **Principle: Make it easy and attractive to stay in**

**Fitts’s Law**

* **Principle: The time to acquire a target is a function of the distance to and size of the target**
* **Multiple Fitts: The time to acquire multiple targets is the sum of the time to acquire each**
* **Principle: Fitts’s Law is in effect regardless of the kind of pointing device or the nature of the target**

**Human Interface Objects**

* **Principle: Human-interface objects can be seen, heard, felt, or otherwise perceived**
* **Principle: Human-interface objects have a standard way of being manipulated  
  Principle: Human-interface objects have standard resulting behaviors  
  Principle: Human-interface objects should be understandable, self-consistent, and stable**
* **Principle: Use a new object when you want a user to interact with it in a different way or when it will result in different behavior**

**Latency Reduction**

* Wherever possible, use multi-threading to push latency into the background
* Principle: Reduce the user’s experience of latency
  + Acknowledge all button clicks by visual or aural feedback within 50 milliseconds
  + Trap multiple clicks of the same button or object.

Because the Internet is slow, people tend to press the same button repeatedly, causing things to be even slower.

* **Principle: Keep users informed when they face delay**

Delay Feedback Times & Indicators

* **Principle: Make it faster to begin with**

Eliminate any element of the application that is not helping. Be ruthless.

**Learnability**

* **Principle:****Limit the Trade-Offs**
* **Principle: Avoid only testing for learnability**

**Metaphors, Use of**

* **Principle: Choose metaphors that will enable users to instantly grasp the finest details of the conceptual model**
* **Principle: Bring metaphors “alive” by appealing to people’s perceptions–sight, sound, touch, and**[**proprioception/kinesthesia**](http://en.wikipedia.org/wiki/Proprioception#Proprioception_and_kinesthesia)**–as well as triggering their memories**
* **Principle: Expand beyond literal interpretation of real-world counterparts**
* **Principle: If a metaphor is holding you back, abandon it**

**Protect Users’ Work**

* **Principle: Ensure that users never lose their work**

**Readability**

* **Principle: Text that must be read should have high contrast**
* **Principle: Use font sizes that are large enough to be readable on standard displays**
* **Principle: Favor particularly large characters for the actual data you intend to display, as opposed to labels and instructions.**
* **Principle: Menu and button labels should have the key word(s) first, forming unique labels**
* **Principle: Test all designs on your oldest expected user population**
* Principle: There’s often an inverse relationship between the “prettiness” of a font and its readability.

**Simplicity**

* **Principle: Balance ease of installation vs. ease of use**

**Principle: Avoid the “Illusion of Simplicity”**

**Principle: Use Progressive Revelation to flatten the learning curve**

**Principle: Do not simplify by eliminating necessary capabilities**

**State**

* **Principle: Because many of our browser-based products exist in a stateless environment, we have the responsibility to track state as needed**

Our systems should “know”:

* Whether this is the first time the user has been in the system
* Where the user was when they left off in the last session
* What the user has found of interest based on time spent with a pointing device moving, objects being touched, etc., in different areas
* Where the user has been during this session
* Where the user is right now and what they are doing
* **Principle: State information should be stored in encrypted form on the server when they log off**
* **Principle: Make clear what you will store & protect the user’s information**

**Visible Navigation**

* **Principle: Make navigation visible**
* **Principle: Limit screen counts by using overlays**