

UsabilityInstitute.com

User-Friendly Interface Requirements

Free Text to Use in Your Request for Proposals (RFPs)

- Copy all of this page to use in your own requests for proposals or even documents
 - Send us feedback
 - We make no stipulations whatsoever on how you must use this material or acknowledge us as its source, but if you are so inclined, we hope that by mentioning UsabilityInstitute.com you will speed the adoption these rules elsewhere. --Thanks, Jack Bellis/ webmaster@UsabilityInstitute.com
1. The software should be usable without reading a printed guide. If the complexity of the tasks being automated cannot feasibly be embedded into the software interface, reading a printed guide may be unavoidable. In this case, the most that any individual user should have to read for a particular role is 50 pages—short enough to read in one sitting.
 2. The interface should enable all interaction techniques and input to be discoverable and chosen from a browse-able, heirarchical structure, arranged in order of the functions the user needs to perform. Until recently this simply meant supplying an exhaustive menu or menu-like outline of choices, supported by dialogs with options and click-able choices. More recently, this requirement is being satisfied by multiple graphical choices, in the form of icons and segmented, click-able graphics known by various names, such as imagemaps. In either case, what this requirement specifically precludes is reliance solely on any of the following techniques: command line syntax; parameter (INI) file options not built into the interface; techniques supported only by combination keystrokes, mouse techniques, or combinations thereof; techniques requiring knowledge of special, manually entered values.
 3. Users should be able to accomplish every task and entry with the fewest possible keystrokes. For instance, dates should not necessitate typing four digits for the year unless the context of the given field leaves considerable doubt as to which millennium might be intended. In many cases, keying in any characters at all for the year may be an unnecessary expense of the users' time.
 4. The software should enable users to do things out of order without being penalized.
 5. The software should enable users to make outright mistakes without being terminated, executed, canceled, re-booted, or erased.
 6. The system should save all of what the users type, by default, meaning without extra steps.
 7. Users should be forewarned when any work is over-written, undone, or erased.
 8. The interface and messages should make it clear why the program does what it does.
 9. Most of the users' work should be retained after power interruptions.