Principles of Usability

* General Understanding.

Standards and Guidelines

* Direction for design.
* Standard is of Product, and Quality is measured by client.
* If the product fulfills all the requirements, then the product is quality product.
* **Standard:** A fan button should be of standard size; it doesn’t have to be of a size of a hand.

Design Patterns

* Capture and reuse design knowledge.

Types of Design Rules

Principles

* E.g. System should be easy to interact.
* Abstract design rules (All the rules discussed above should be implemented.)
* Low authority: Access to the database should be given to as minimum people as possible.
* High Generality: General standard of the application domain must be followed while developing the product.

Example: Test Cases -> X ID must be present in database, instead of 19k1048.

Standards

* Specific design rules.
* High Authority (ISO)
* Limited Application

Guidelines

* E.g. Use color to highlight links.
* Can guide on how to achieve a principle.
* Lower Authority
* More general application.

Principles to Support Usability

https://www.cs.uct.ac.za/mit\_notes/human\_computer\_interaction/htmls/ch06s06.html

LearnabIlity

* The ease with which new user can begin effective interaction and achieve maximal performance.
* **Predictability:** support for the user to determine the effect of future action based on past interaction history (can I ‘tell’ what will happen based on what I have gone through in past?)
* **Operation Visibility**: Perceived Affordance. Operation visibility refers to how the availability of operations which can next be performed is shown to the user. If an operation can be performed, then there may be some perceivable indication of this to the user. This principle supports the performance in humans of recognition over recall.

**Synthesizability:** support for the user to assess the effect of past operations on the current state (can I ‘tell’ why I am here based on what I have gone through in the past?)

**Familiarity:** the extent to which a user's knowledge and experience in other real world or computer-based domains can be applied when interacting with a new system.

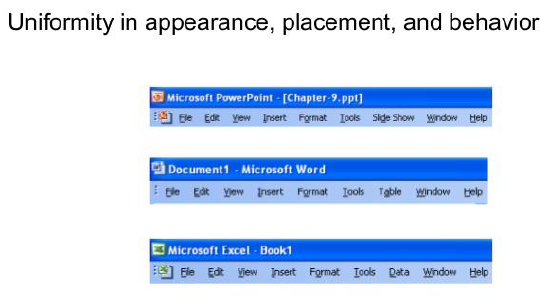
**EG:** MS Word -> Typewriter.

**Generalizability:** support for the user to extend knowledge of specific interaction within and across applications to other similar situations.

**EG:** Cut, copy, paste. Circle -> Ellipse, Square -> Rectangle.

**Consistency:** likeness in input-output behavior arising from similar situations or similar task objectives.

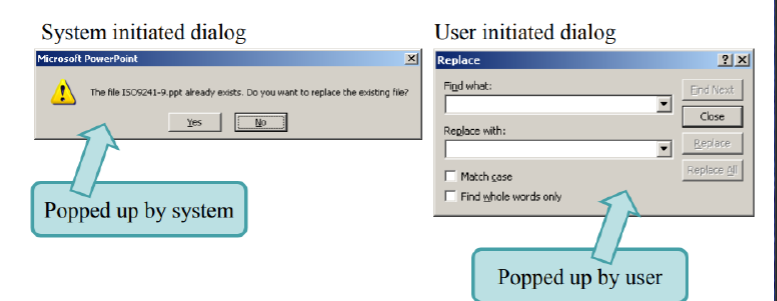
When keyboards for typewriters were first made, the designers laid out the keys in alphabetical order. Then it was discovered that such an arrangement of keys was both inefficient from the machine’s perspective (adjacent typewriter keys pressed in succession caused jams in the mechanism, so the likelihood of this occurrence had to be designed out) and tiring for the typist (a touch-typist would not have equal stress distributed over all fingers) The resulting QWERTY and DVORAK keyboards have since been adopted to combat the problems of the ‘consistent’ keyboard layout.



Flexibility

* The multiplicity of ways in which user and system exchange information.

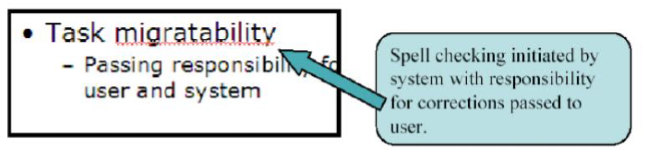
**Dialogue initiative:** user freedom from artificial constraints on the input dialog imposed by the system; user vs system - who has the initiative in the dialog?



**Multithreading:** The ability of the system to support user interaction for more than one task at a time.

E.g. Tabs, Windows.

**Task Migratability:** The ability to transfer control for execution of tasks between the system and the user (consider e.g., spell-checking task).



**Substitutivity:** The extent to which an application allows equivalent input and output values to be substituted for each other (values in input e.g. fractions/decimals, values in output e.g. both digital and analog, output/input e.g. output can be reused as input).

**Example: Line Coordinates example.**

**Customizability:** the ability of the user or the system to modify the user interface. (adaptability vs adaptivity)? - initiated modification.

Robustness

* The level of support provided to the user in determining successful achievement and assessment of goals.
* **Observability:** The extent to which the user can evaluate the internal state of the system from the representation on the user interface.
  + **Mail, Loading, Downloading.**
* **Recoverability:** The extent to which the user can reach the intended goal after recognizing an error in the previous interaction.
* **Responsiveness:** A measure of the rate of communication between the user and the system.
* **Task conformance:** The extent to which the system services support all the tasks the user would wish to perform and in the way the user would wish to perform.

Shneiderman’s 8 Golden Rules Of Interface Design

1. **Strive for consistency**

Example: Mac OS menu bar 🡪 Graphical Elements.

Else user will have to spend time learning new interface, User effort++ = Bad interface.

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 extremely helpful to an expert user.

**Example:** Keyboard Accelerators 🡪 Keyboard shortcuts.

1. **Offer informative feedback**

Every action performed should have a respective feedback, and in human understandable form.

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.

Instant or short time response.

1. **Design dialogs to yield closure**

Don’t keep the user guessing.

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 error prevention and 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.

Example: Online form filling, when a mandatory field is not filled, redirect the user to that field.

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.

Undo for every step.

Instead of starting from the starting point.

1. **Support internal locus of control.**

Make the user feel as if the whole system is in his 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 long term memory load.**

Recognition rather than recall.

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.

Norman’s 7 Princle

1. Use both knowledge in the world and knowledge in the head.
2. Simplify the structure of tasks.

Decompose the big task into smaller tasks.

1. Make things visible: bridge the gulfs of Execution and Evaluation.
2. Get the mappings right.

E.g. Buttons and Lights mapping.

Icons on Apps.

1. Exploit the power of constraints, both natural and artificial.
2. Design for error.

E.g. Undo Feature.

1. When all else fails, standardize.