GOLDEN RULES OF INTERFACE DESIGN

- **8 Golden Rules of Interface Design**
- Ben Shneiderman American computer scientist and professor at the University of Maryland Human-Computer Interaction Lab.
- **Strive for consistency** - in actions, sequences, layouts, commands, etc. users become familiar with the digital landscape of your product so they can achieve their goals more easily.
- **Enable frequent users to use shortcuts** – special keys sequences , macros to perform regular or familiar functions.

  Frequent / advanced user can navigate and operate the user interface more quickly and effortlessly.

GOLDEN RULES OF INTERFACE DESIGN

- **Offer Informative Feedback** – for every user action apt to the action's magnitude. The user should know where they are at and what is going on at all times
- **Design Dialogs to Yield Closure** – User Knows when the task is completed. Don't keep your users guessing. Tell them what their action has led them to.
- Closure the Famous ATM Insert and DIP Example!
- **Permit Easy Reversal of Actions** – relieves anxiety and promotes exploration – shud we introduce CONTROL Z!!!

GOLDEN RULES OF INTERFACE DESIGN

- **Offer Error Prevention or Error Handling** - users prevented from making errors or mistakes – clear feedback in case of errors to recover - when unavoidable errors occur, ensure users are provided with simple, intuitive step-by-step instructions to solve the problem - flag the text fields where the users forgot to provide input in an online form
- **Support Internal Locus of Control** – user has full control of the system
- **Reduce Short Term Memory Load** – Keep displays simple – consolidate multiple page displays - give time for learning actions

GOLDEN RULES OF INTERFACE DESIGN

- **five items in our short-term memory at one time.**
- (later we will revisit the 7+/-2 Digit Span Rule)
- interfaces should be as simple as possible with proper information hierarchy, and choosing recognition over recall.
- **Recognizing easier than recall** - perceiving cues that help us reach into our vast memory – easier retrieval
- multiple choice questions easier than short answer questions - recognize the answer rather than recall it from our memory

The principles	Questions to consider
1. Strive for consistency	Is the style of this element maintained across your site/app? Is this content placed in the correct location according to the site hierarchy? Does this follow the conventions for your chosen platform? How can you make your designs more consistent?
2. Enable frequent users to use shortcuts	Are there shortcuts available for your more experienced users? Who is this product designed for? Will there be a need to consider experienced users? How can you make it easier and quicker for experienced users?
3. Offer informative feedback	Does the user know where they are at in the process? Does the user know what they have done after performing this action? How are you communicating this feedback to your user?
4. Design dialogue to yield closure	Does the user have to do any guessing here? Is it clear and obvious enough for your intended audience? Are there any next steps for the user? How are you communicating the system status with the user?
5. Offer simple error handling	Have you done everything imaginable to prevent this error from happening on your end? Is this error avoidable in the first place? If the user does make an error, how easy is it for them to fix it?
6. Permit easy reversal of actions	How many steps does the user have to take to reverse their actions? Will the user quickly realize they need to reverse the action in the first place? How can you make your users detect the possibility of reversal?
7. Support internal locus of control	Will the user feel in control at this specific touch point in your app? Will they be surprised in an unpleasant manner? Does the site feel easily navigable? Does the user feel safe and in control? How can you make the user feel more safe and in control?
8. Reduce short-term memory load	Are there enough visual cues here for the user to find the functionality or item? Do they have to remember things to understand what's going on? How can you help the user recall?

NIELSEN'S RULES FOR INERFACE DESIGN

Visibility of system status - system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world - system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, ; information appear in a natural & logical order.

- **User control and freedom** - system functions by mistake ; clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

NIELSEN'S RULES FOR INERFACE DESIGN

- **Consistency and standards –** No confusion on 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 ! 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. Not 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.

NIELSEN'S RULES FOR INERFACE DESIGN

- **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 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** easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

NORMANS RULES FOR INERFACE DESIGN

- **Visibility** – The more visible functions are, the more likely users will be able to know what to do next.
- Incontrast, when functions are "out of sight," it makes them more difficult to find and know how to use.
- **Feedback** , allowing the person to continue with the activity. Types such as tactile, verbal, and combinations of these.
- **Constraints** restricting the kind of user interaction that can take place at a given moment.
- **Mapping** –relationship between controls and their effects in the world. up and down arrows used to represent the up and down movement of the cursor, respectively, on a computer keyboard.

- **Consistency** –interfaces to have similar operations and use similar elements for achieving similar tasks.
- using the same input action to highlight any graphical object at the interface, such as always clicking the left mouse button. Inconsistent interfaces, on the other hand, allow exceptions to a rule.
- **Affordance** –At a very simple level, to afford means "to give a clue" (Norman, 1988). When the affordances of a physical object are perceptually obvious ; it is easy to know how to interact with it.