Principles of GUI Design
In April 1981, Xerox announced the 8010 Star Information System, a new personal computer designed for offices.

The Star user interface adheres rigorously to a small set of principles designed to make the system seem friendly by simplifying the human-machine interface.
…a brainchild Xerox’s Palo Alto Research Center (PARC).

“The developers recognized the importance of large, highly detailed display screens with plenty of graphics; they gave the machine the ability to have several different documents on the screen at the same time; and they introduced a pointing device for the user to specify a work area on the screen. But the system was too expensive and too slow. Users liked the power and the ease of operation, but they needed better performance. The benefits of easy to use commands were completely outweighed by the slow response speed. The display could not always keep up with typing, and requests for explanation (“help”) sometimes took so long that a user could go for a cup of coffee while waiting… a common fate of pioneers: the spirit was willing but the implementation weak.”
The Foundation for Graphical Interfaces

• The illusion of manipulatable objects
• Visual order and viewer focus
• Revealed Structure
• Consistency
• Appropriate effect or emotional impact
• A match with the medium

Principles that guided the design of STAR
Design Goals

1. Aesthetically pleasing
2. Clarity
3. Compatibility
4. Comprehensibility
5. Configurability
6. Consistency
7. Control
8. Directness
9. Efficiency
10. Familiarity
11. Flexibility
12. Forgiveness
13. Predictability
14. Recovery
15. Responsiveness
16. Simplicity
17. Transparency
18. Trade-offs
Aesthetically Pleasing

• Provide visual appeal by:
  – Providing meaningful contrast between screen elements
  – Creating grouping
  – Aligning screen elements and groups
  – Providing 3-D representation
  – Using color effectively and simple
Clarity

The interface should be visually, conceptually, and linguistically clear, including:

– Visual elements
– Functions
– Metaphors
– Words and text
Compatibility

- Provide compatibility with the following:
  - the user
  - the task and job
  - the product

- Adopt the user’s perspective!
Comprehensibility

• A system should be easily learned and understood.
• The user should know the following:
  – what to look at
  – what to do
  – when to do it
  – why to do it
  – how to do it
• The flow of actions, responses, visual presentations, and information should be in a sensible order that is easy to recollect and place in context
Configurability

• Permit easy configuration and reconfiguration of settings
  – Enhance a sense of control
  – Encourage an active role in understanding
  – Allow for personal preferences
Consistency

• A system should look, act, and operate the same throughout. Similar components should:
  – Have a similar look
  – Have similar uses
  – Operate similarly
  – The same action should always yield the same result
  – The function of elements should not change
  – The position of standard elements should not change
Control

- The user must control the interaction
  - actions should result from explicit user requests
  - actions should be performed quickly
  - actions should be capable of interruption or termination
  - the user should never be interrupted for errors
- The context maintained must be from the perspective of the user
- The means to achieve goals should be flexible and compatible with the user’s skills, experiences, habits, and preferences
- Avoid modes since they constrain the actions available to the user
- Permit the user to customize aspects of the interface, while always providing a proper set of defaults
Directness, Efficiency, Familiarity

• **Directness**
  – Provide direct and intuitive ways to accomplish tasks

• **Efficiency**
  – Minimize eye, hand, and other control movements
  – Transitions between various system controls should flow easily and freely

• **Familiarity**
  – Employ familiar concepts and use a language that is familiar to the user
  – Keep the interface natural, mimicking the user’s behavior patterns
  – Use real-world metaphors
Flexibility

- A system must be sensitive to the differing needs of its users, enabling a level and type of performance based upon each user’s:
  - Knowledge and skills
  - Experience
  - Personal preference
  - Habits
  - The conditions at that moment
Forgiveness

• Tolerate and forgive common and unavoidable human errors
• Prevent errors from occurring whenever possible
• Protect against possible catastrophic errors
• When an error does occur, provide constructive messages
Predictability

• The user should be able to anticipate the natural progression of each task.
  – Provide distinct and recognizable screen elements
  – Provide cues to the result of an action to be performed
• All expectations should be fulfilled uniformly and completely
Recovery, Responsiveness

Recovery

– A system should permit:
  • commands or actions to be abolished or reversed
  • immediate return to a certain point it difficulties arise

Responsiveness

– The system must rapidly respond to the user’s requests
– Provide immediate acknowledgment for all user actions:
  • visual
  • textual
  • auditory
Simplicity

• Provide as simple an interface as possible
• Five ways:
  1) Use progressive disclosure, hiding things until they are needed
     • present common and necessary functions first
     • prominently feature important functions
     • hide more sophisticated and less frequently used functions
  2) Provide defaults
  3) Minimize screen alignment points
  4) Make common actions simple at the expense of uncommon actions made harder
  5) Provide uniformity and consistency
Transparency, Trade-Offs

• Transparency
  – Permit the user to focus on the task or job, without concern for the mechanics of the interface.
  – Workings, and reminders of workings, inside the computer should be invisible to the user.

• Trade-Offs
  – Final design will be based on a series of trade-offs balancing often conflicting design principles
  – People’s requirements always take precedence over technical requirements