USER INTERFACE DESIGN

almost every software product is highly interactive with users, so, almost every software product has a user interface. No wonder then that many users often judge a software product based on its user interface.

User interface design creates an effective communication medium between humans and computers.

Who does it?

A software engineer designs the user interface by applying an iterative process.

Why is it important?

If the software is difficult to use, if it forces you into mistakes, or if it decreases your efforts to accomplish your goals, you won't like it, regardless of the computational power or the functionality it offers. the interface has to be right because it forms the user's understanding of the software.

THE GOLDEN RULES

There are *three golden rules* that designers must follow to create an efficient user interface:

1. Place the user in control.
2. Reduce the user's memory load.
3. Make the interface consistent.

1. Place the user in control.

Below are a number of design principles that allow the user to maintain control:

- define interaction modes in a way that does not force a user into unnecessary or undesired actions.
- Hide technical internals from the casual user.
- Allow user interaction to be interruptible and undoable.

2. Reduce the user's memory load.

Below are a number of design principles that enable an interface to reduce the user's memory load:

- Reduce demand on short-term memory.
- Establish meaningful defaults.
- Define shortcuts that are intuitive.

3. Make the interface consistent.

Below are a number of design principles that help make the interface consistent:

- It is important to provide indicators (e.g., window titles, graphical icons, consistent color coding) that enable the user to know the context of the work at hand.
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

The User Interface Design Process

The design process for user interfaces is iterative and can be represented using a spiral model. Referring to Figure 6.1, the user interface design process encompasses four distinct framework activities:

1. interface analysis and modeling.

2. Interface design.

3. Interface construction.

4. Interface validation

The spiral implies that each of these tasks will occur more than once, with each pass around the spiral representing additional elaboration of requirements and the resultant design. In most cases, the implementation activity involves prototyping—the only practical way to validate what has been designed.

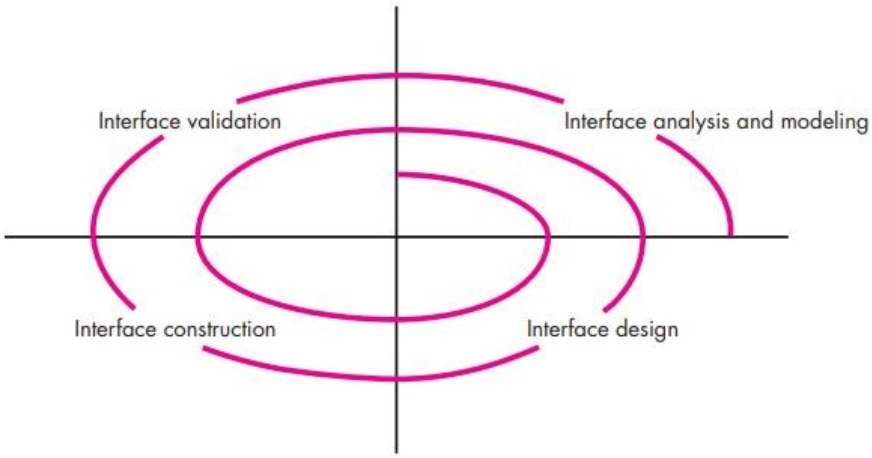

Figure 6.1: The user interface design process.

Design Evaluation

Once an operational user interface prototype has been created, it must be evaluated to determine whether it meets the needs of the user. Evaluation can range from an informal in which a user provides feedback, to a formally designed study that uses statistical methods for the evaluation.

The user interface evaluation cycle takes the form shown in Figure 6.2. After the design model has been completed, a first-level prototype is created. The prototype is evaluated by the user, who provides the designer with direct comments about the effectiveness of the interface. Design modifications are made based on user input and the next level prototype is created.

The evaluation cycle continues until no further modifications to the interface design are necessary.

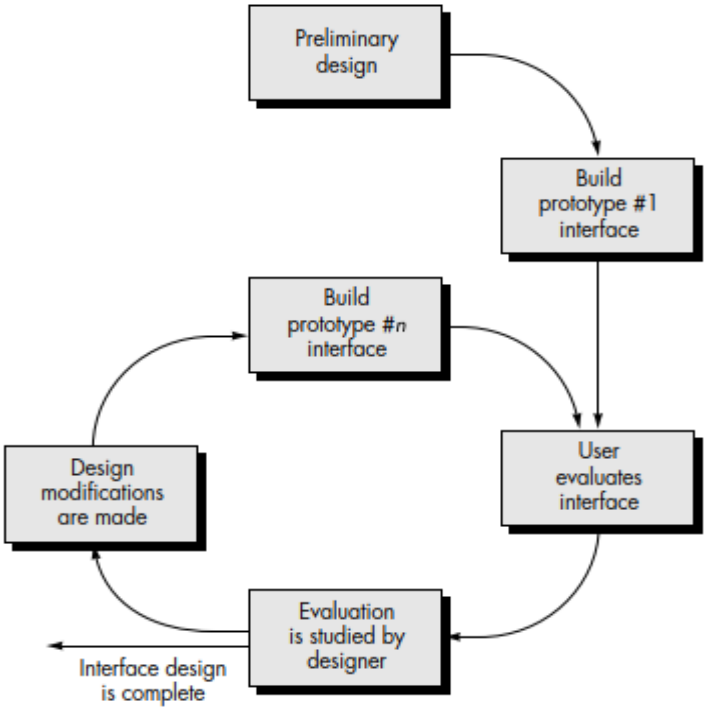

Figure 6.2: The interface design evaluation cycle.

TYPES OF USER INTERFACES

User interfaces can be classified into the following categories:

1. command line interface.

The user would have to learn a whole set of commands so that he could make use of the computer system. he would have to type a command to make anything happen.

The MS-DOS operating system and the command shell in the Windows operating system are examples of command-line interfaces (CLI).

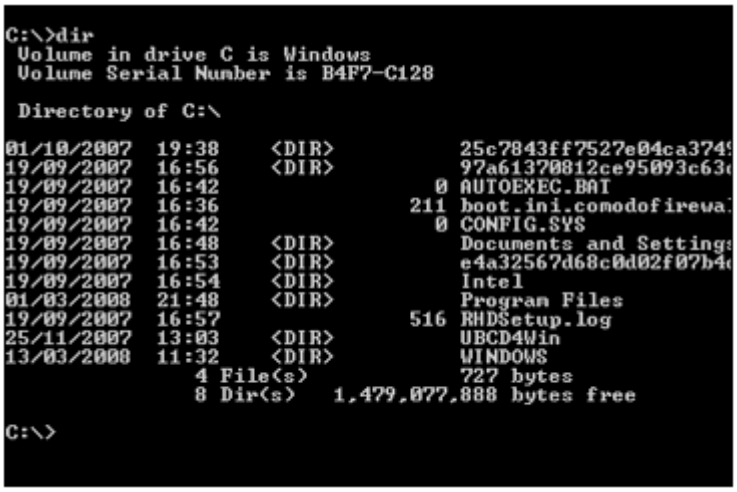

Figure 5.3: Command line interface.

2. Graphical user interface.

A GUI is an interface built around visual (graphical) things:

- *Windows* are regions of the screen used to display information
- *Icons* are small pictures that are used to represent folders, software, etc.
- *Menus* are lists of options the user can select from
- *A pointer* is an arrow that can be moved around the screen and is used to select things.

Therefore, a GUI is sometimes called a WIMP interface. Windows XP is an example of an operating system with a GUI.

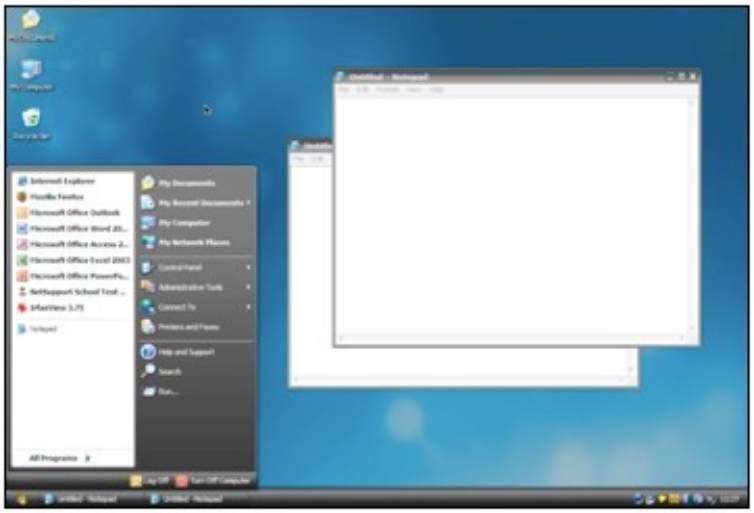

Figure 5.4: Graphical user interface.

Main differences between CLI and GUI

Various main differences between the CLI and GUI are as follows:

1. A graphical user interface enables users to interact with the operating system or application. On the other hand, a CLI is an interface that allows the user to perform tasks by issuing commands in successive lines of text or command lines.
2. The CLI needs the commands to be memorized, making it difficult for users. Professionals frequently prefer CLI. GUI is more user-friendly than CLI and may be utilized by both beginners and experienced professionals.
3. The CLI is ideal for dealing with difficult tasks. The difficult task may be handled by writing a few commands, whereas GUI requires some steps to be followed in order to function.
4. The GUI is slow than CLI. It took more time to complete tasks.
5. The interface in Command Line Interface is always consistent. The interface in GUI changes as the software is updated.