QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES
INFORMATION TECHNOLOGY DEPARTMENT

WEEKS 12
INTERFACE DESIGN

SE101-SOFTWARE ENGINEERING

LEARNING OUTCOMES:

- Explain the Golden Rules.
- Create a User Interface Design.
- Recognize design evaluation.

Topics:

Interface Design

- The Golden Rules
- User Interface Analysis and Design
- Design Evaluation

User Interface Design

User interface design **creates** an **effective communication medium** between a human and a computer.

Following a set of interface design principles, **design identifies interface objects and actions and then creates a screen layout** that forms the basis for a user interface prototype.

A software engineer designs the user interface by applying an iterative process that draws on predefined **design principles.**

- As technologists studied **human interaction**, two dominant issues arose.
- First, **a set of golden rules** were identified. These applied to all human interaction with technology products.
- Second, **a set of interaction mechanisms** were defined to enable software designers to build systems that properly implemented the golden rules.
- These interaction mechanisms, collectively called the **Graphical User Interface (GUI).**

The Golden Rules

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

1. Place the User in Control.

- Theo Mandel defines a number of design principles that allow the user to maintain control:

1.1 Define interaction modes in a way that does not force a user into unnecessary or undesired actions.

- An interaction mode is the current state of the interface.
- For example, if *spell check* is selected in a word-processor menu, the software moves to a spell-checking mode.

1. Place the User in Control. (cont.)

1.2 Provide for flexible interaction.

Because different users have different interaction preferences, choices should be provided.

- For example, software might allow a user to interact via keyboard commands, mouse movement, a digitizer pen, a multitouch screen, or voice recognition commands.

1. Place the User in Control. (cont.)

1.3 Allow user interaction to be interruptible and undoable.

- Even when involved in a sequence of actions, the user should be able to interrupt the sequence to do something else (without losing the work that had been done). The user should also be able to "undo" any action.

1. Place the User in Control. (cont.)

1.4 Streamline interaction as skill levels advance and allow the interaction to be customized.

Users often find that they perform the same sequence of interactions repeatedly. It is worthwhile to design a "macro" mechanism that enables an advanced user to customize the interface to facilitate interaction.

1. Place the User in Control. (cont.)

1.5 Hide technical internals from the casual user.

The user interface should move the user into the virtual world of the application. The user should not be aware of the operating system, file management functions, or other arcane computing technology.

In essence, the interface should never require that the user interact at a level that is "inside" the machine.

1. Place the User in Control. (cont.)

1.6 Design for direct interaction with objects that appear on the screen.

The user feels a sense of control when able to manipulate the objects that are necessary to perform a task in a manner similar to what would occur if the object were a physical thing.

For example, an application interface that allows a user to “stretch” an object (scale it in size) is an implementation of direct manipulation.

2. Reduce the User's Memory Load.

2.1 Reduce demand on short-term memory.

- When users are involved in complex tasks, the demand on short-term memory can be significant.
- The interface should be designed to reduce the requirement to remember past actions, inputs, and results.
- **This can be accomplished by providing visual cues that enable a user to recognize past actions, rather than having to recall them.**

2. Reduce the User's Memory Load. (cont.)

2.2 Establish meaningful defaults.

- The initial set of defaults should make sense for the average user, but a user should be able to specify individual preferences. However, a "reset" option should be available, enabling the redefinition of original default values.

2. Reduce the User's Memory Load. (cont.)

2.3 Define shortcuts that are intuitive.

When mnemonics are used to accomplish a system function (e.g., alt-P to invoke the print function), the mnemonic should be tied to the action in a way that is easy to remember (e.g., first letter of the task to be invoked).

2. Reduce the User's Memory Load. (cont.)

2.4 The visual layout of the interface should be based on a real-world metaphor.

- For example, a bill payment system should use a checkbook and check register metaphor to guide the user through the bill paying process. This enables the user to rely on well-understood visual cues, rather than memorizing an arcane interaction sequence.

2. Reduce the User's Memory Load. (cont.)

2.5 Disclose information in a progressive fashion.

- The interface should be organized hierarchically. That is, information about a task, an object, or some behavior should be presented first at a high level of abstraction

3. Make the Interface Consistent.

3.1 Allow the user to put the current task into a meaningful context.

- Many interfaces implement complex layers of interactions with dozens of screen images. 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.
- In addition, the user should be able to determine where he has come from and what alternatives exist for a transition to a new task.

3. Make the Interface Consistent.

3.2 Maintain consistency across a family of applications.

A set of applications (or products) should all implement the same design rules so that consistency is maintained for all interaction.

3. Make the Interface Consistent.

3.3 If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

Once a particular interactive sequence has become a de facto standard (e.g., the use of alt-S to save a file), the user expects this in every application he encounters. A change (e.g., using alt-S to invoke scaling) will cause confusion.

User Interface Analysis and Design

- The overall process for analyzing and designing a user interface begins with the
- creation of different models of system function (as perceived from the outside).
- You begin by defining the human- and computer-oriented tasks that are required to achieve system function and then considering the design issues that apply to all interface designs.
- Tools are used to prototype and ultimately implement the design model, and the result is evaluated by end users for quality.

Interface Analysis and Design Models

- **Four different models** come into play when a user interface is to be analyzed and designed.
- A human engineer (or **the software engineer**) establishes a **user model**, the software engineer creates a **design model**, the end user develops a mental image that
- is often called the user's mental model or the system perception, and the implementers.

- To build an effective user interface, **all design should begin with an understanding of the intended users, including profiles of their age, gender, physical abilities, education, cultural or ethnic background, motivation, goals and personality**.

- Users can be categorized as:
  - **Novices.** No syntactic knowledge of the system and little semantic knowledge of the application or computer usage in general.
  - **Knowledgeable, intermittent users.** Reasonable semantic knowledge of the application but relatively low recall of syntactic information necessary to use the interface.
  - **Knowledgeable, frequent users.** Good semantic and syntactic knowledge that often leads to the "power-user syndrome"; that is, individuals who look for shortcuts
  - and abbreviated modes of interaction.

- **The user's mental model** (system perception) is the image of the system that end users carry in their heads.

- For example, if the user of a particular word processor were asked to describe its operation, the system perception would guide the response. The accuracy of the description will depend upon the user's profile and overall familiarity with software in the application domain.

- **The implementation model** combines the outward manifestation of the computer-based system (the look and feel of the interface), coupled with all supporting information (books, manuals, videotapes, help files) that describes interface syntax and semantics.

- **When the implementation model and the user's mental model are coincident, users generally feel comfortable with the software and use it effectively.**

- To accomplish this "melding" of the models, the **design model must have been developed to accommodate the information contained in the user model**, and the **implementation model must accurately reflect syntactic and semantic information** about the interface.

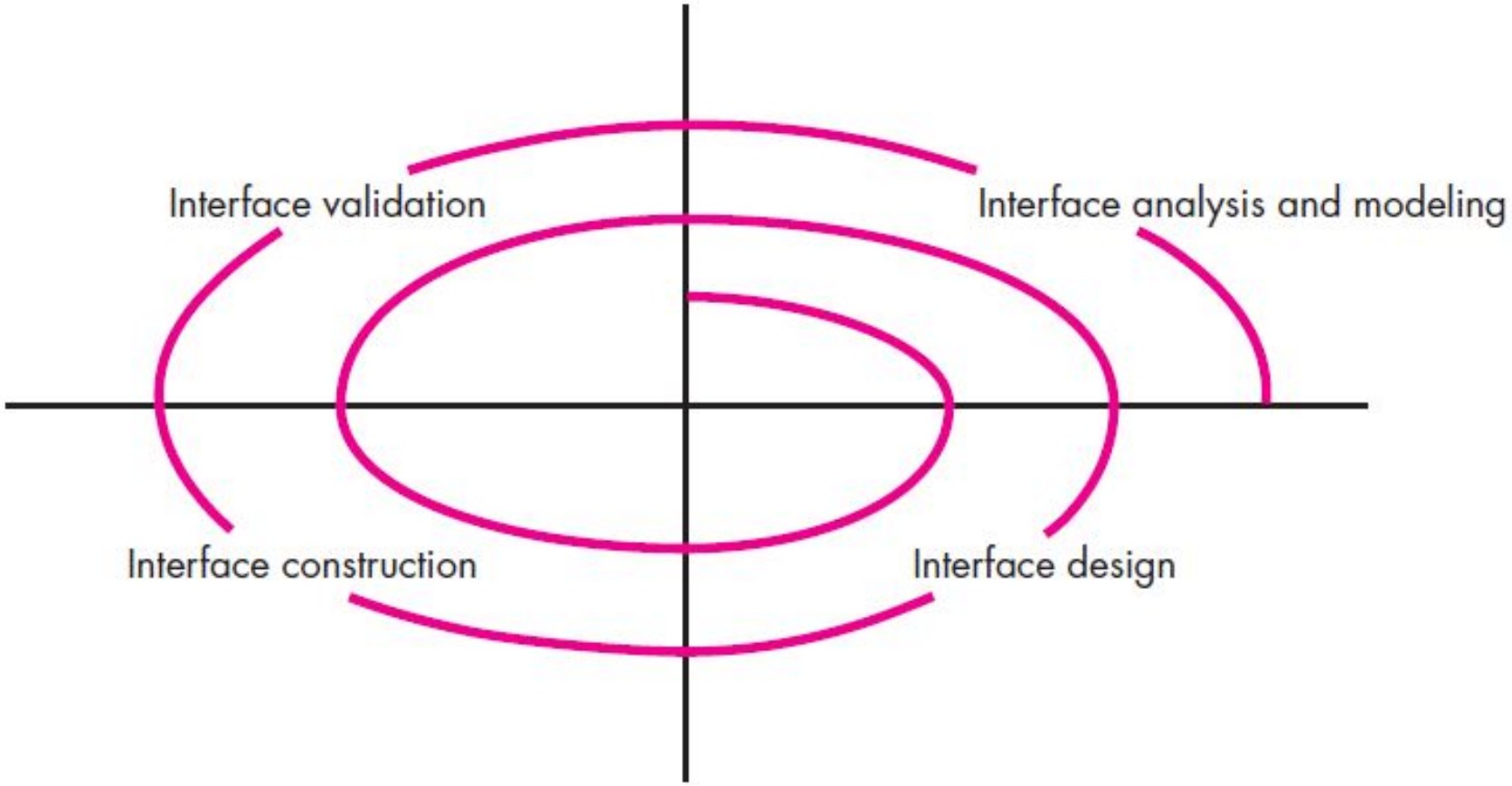

Figure 12.1 The User Interface Design Process

The goal of *interface design* is **to define a set of interface objects and actions** (and their screen representations) that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system.

Interface construction normally begins with the **creation of a prototype** that enables usage scenarios to be evaluated. As the iterative design process continues, a user interface tool kit may be used to complete the construction of the interface.

***Interface validation* focuses on**

(1) the ability of the interface to implement every user task correctly, to accommodate all task variations, and to achieve all general user requirements;

(2) the degree to which the interface is easy to use and easy to learn, and

(3) the users' acceptance of the interface as a useful tool in their work.

Use Cases

The use case **describes the manner in which an actor** (in the context of user interface design, an actor is always a person) **interacts with a system.**

When used as part of **task analysis,** the use case is **developed to show how an end user performs some specific work-related task.**

User Interface Design Steps

Although many different user interface design models have been proposed, all suggest some combination of the following steps:

1. Using information developed during interface analysis, **define interface objects and actions** (operations).
2. **Define events** (user actions) that will cause the state of the user interface to change. Model this behavior.
3. **Depict each interface state** as it will actually look to the end user.
4. **Indicate how the user interprets the state** of the system from information provided through the interface.

- Regardless of the sequence of design tasks, you should

  (1) always follow the golden rules,
  (2) model how the interface will be implemented, and
  3) consider the environment (e.g., display technology, operating system, development tools) that will be used.

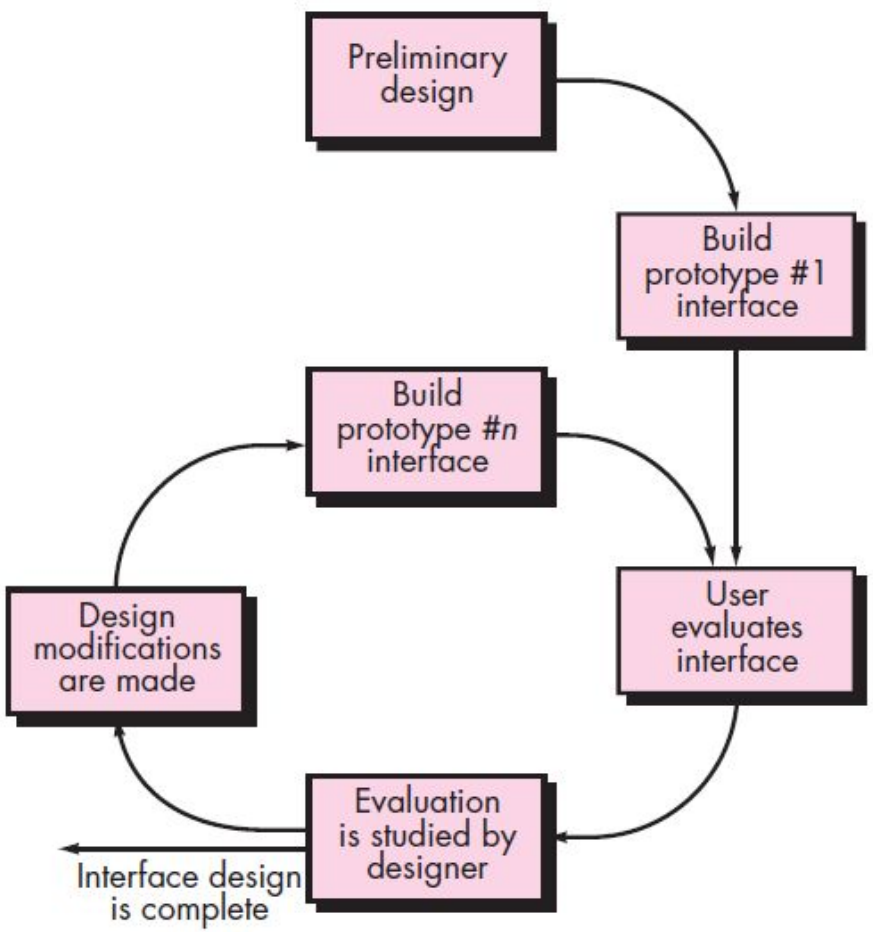

Figure 12.2 The Interface Design Evaluation Cycle

Design Evaluation

- Once you create an operational user interface prototype, it must be evaluated **to determine whether it meets the needs of the user.**

- Evaluation can span a formality spectrum that ranges from an **informal "test drive,"** in which a user **provides impromptu feedback** to a formally designed study that uses statistical methods for the evaluation of questionnaires completed by a population of end users.

Design Evaluation

- After the design model has been completed, a first-level prototype is created. The prototype is evaluated by the user, who provides you with direct comments about the efficacy of the interface.
- In addition, if formal evaluation techniques are used (e.g., questionnaires, rating sheets), you can extract information from these data (e.g., 80 percent of all users did not like the mechanism for saving data files).
- 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.

END OF PRESENTATION.
THANK YOU!