Chapter 15

Human-Computer Interaction

Design criteria for usability

- Efficiency
- Accuracy
- Learnability
- Memorability
- Satisfaction

Software design cycle

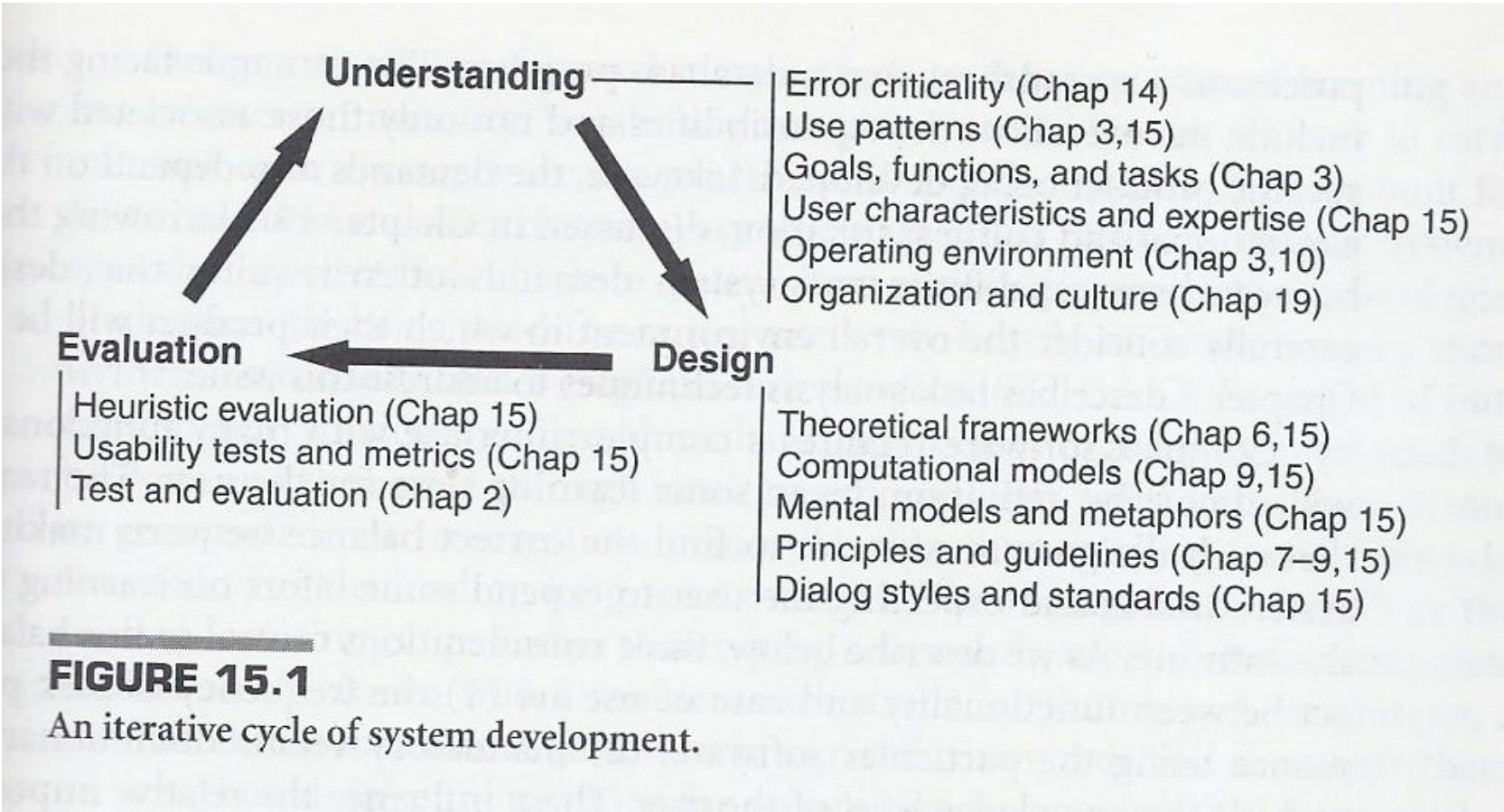

FIGURE 15.1
An iterative cycle of system development.

Balance between ease of use and functionality

- Three considerations central to balancing act between functionality and ease of use
  1. Frequency of use
  2. Mandatory vs. discretionary use
  3. Knowledge level of the user
     - Novice
     - Knowledgeable intermittent users
     - Expert frequent users

Seven stages of Action

1. Form goal
2. Form intent
3. Specify action
4. Execute
5. Process action
6. Evaluate feedback
7. Evaluate outcome

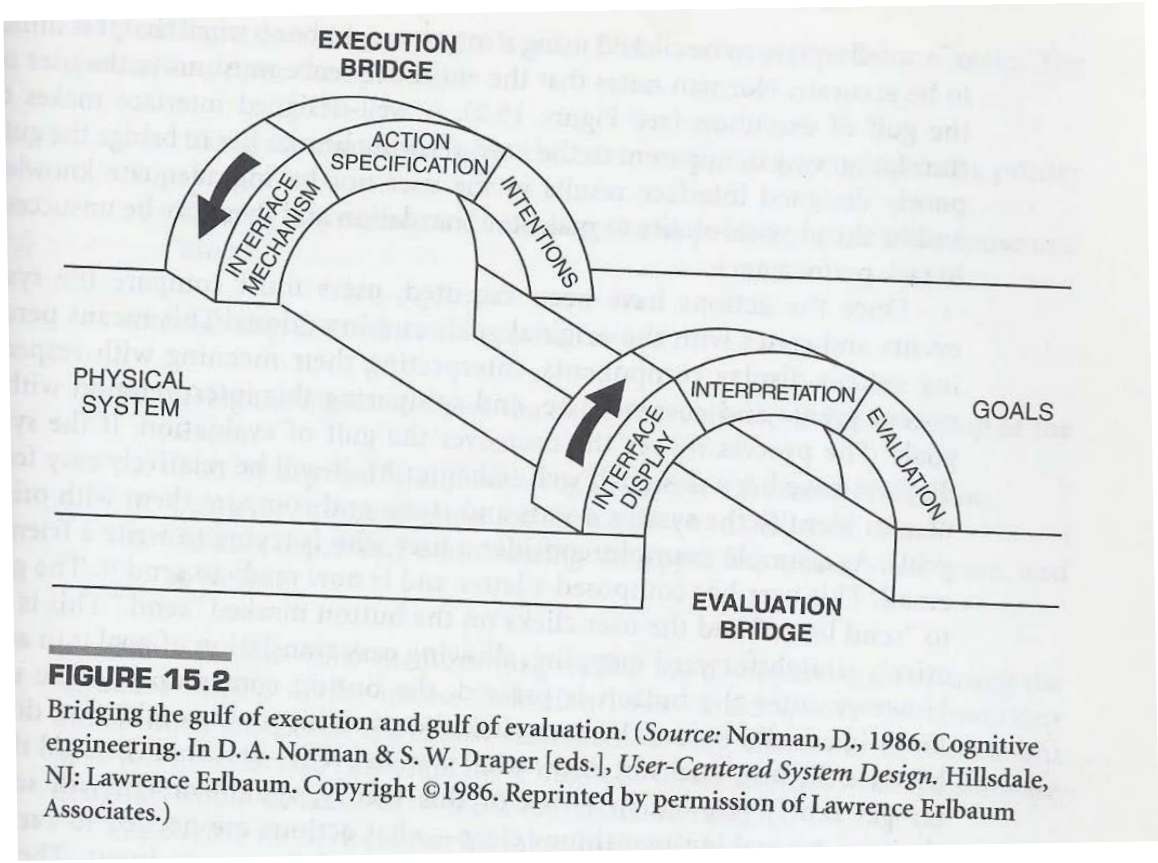

FIGURE 15.2
Bridging the gulf of execution and gulf of evaluation. (*Source:* Norman, D., 1986. Cognitive engineering. In D. A. Norman & S. W. Draper [eds.], *User-Centered System Design.* Hillsdale, NJ: Lawrence Erlbaum. Copyright ©1986. Reprinted by permission of Lawrence Erlbaum Associates.)

Goals, Operators, Methods, and Selection rules (GOMS)

- **Goals:** what the user is trying to accomplish
- **Operators:** actions that are performed to achieve the goal
- **Methods:** sequences of operators that accomplish the goal
- **Selection:** because several methods often can be used to accomplish a goal, selection rules must be specified to identify the conditions when a user will select one method over another

Supporting mental models

Conceptual model – framework through which the function of a system is presented

- Make the invisible visible
- Provide feedback
- Build consistency
- Present functionality through a familiar metaphor

Nielsen's Heuristic

TABLE 15.1 General Interface Design Principles

Match between system and real world
Speak the user's language.
Use familiar conceptual models and/or metaphors.
Follow real-world conventions.
Map cues onto user's goals.
Consistency and standards
Express the same thing the same way throughout the interface.
Use color coding uniformly.
Use a uniform input syntax (e.g., require the same actions to perform the same functions).
Functions should be logically grouped and consistent from screen to screen.
Conform to platform interface conventions.
Visibility of system status
Keep user informed about what goes on (status information).
Show that input has been received.
Provide timely feedback for all actions.
Indicate progress in task performance.
Use direct manipulation: visible objects, visible results.
User control and freedom
Forgiveness: Obvious way to undo, cancel, and redo actions.
Clearly marked exits.
Allow user to initiate/control actions.
Avoid modes when possible.
Error prevention, recognition, and recovery
Prevent errors from occurring in the first place.
Help users recognize, diagnose, and recover from errors.
Use clear, explicit error messages.
Memory
Use see-and-point instead of remember-and-type.
Make the repertoire of available actions salient.
Provide lists of choices and picking from lists.
Direct manipulation: visible objects, visible choices.
Flexibility and efficiency of use
Provide shortcuts and accelerators.
User has options to speed up frequent actions.
System should be efficient to use (also, ability to initiate, reorder, or cancel tasks).
Simplicity and aesthetic integrity
Things should look good with a simple graphic design.
Use simple and natural dialog; eliminate extraneous words or graphics.
All information should appear in a natural and logical order.

Source: Nielson, J. *Enhancing the explanatory power of visibility heuristics.* Chi '94 Proceedings. New York: Association for Computing Machinery.

Evaluating usability

- Heuristic evaluation
  - Several HCI specialists evaluate interfaces based on a set of principle or heuristics (such as Nielsen's heuristic)
- Prototypes
  - Low fidelity prototypes
  - Storyboards
  - Wireframes
- Usability metrics
  - Measures used for usability testing
  - Effectiveness, efficiency, and user satisfaction