



# Module 7

## Component-Level and User Experience Designs

---

### ▼ What are **Shneiderman's "Eight Golden Rules of Interface Design"**?

1. **Strive for consistency.** Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.
2. **Enable frequent users to use shortcuts.** As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
3. **Offer informative feedback.** For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
4. **Design dialog to yield closure.** Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.
5. **Offer simple error handling.** As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

6. **Permit easy reversal of actions.** This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
7. **Support internal locus of control.** Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
8. **Reduce short-term memory load.** The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

▼ What is the open-closed principle?

- **A module or component should be open for extension but closed for modification.**
- This is important because it means the component is specified in a way that allows it to be extended without needing to make internal modifications to the component itself.

▼ What is the Liskov Substitution principle?

- **Subclasses are substitutable for their base classes.**
- Specifically, a component that uses a base class should function properly if a class derived from the base class is passed to the component instead.

▼ What is the dependency inversion principle?

- **Dependency on abstractions rather than concretions.**
- The logic behind this is that the more a component depends on other concrete components, the more difficult it will be to extend.

▼ What is the interface segregation principle?

- **Many client-specific interfaces are better than one general-purpose interface.**

- With ISP, you should create a specialized interface to serve each major category of client.

▼ What is the reuse/release equivalency principle?

- When classes or components are designed for reuse, an understanding is established between the developer of the reusable entity and the people who use it.
- This is based on a smaller principle, which states that “the granule of reuse is the granule of release.”

▼ What is the common closure principle?

- **Classes that change together belong together.**
- This means classes should be **packaged** together.

▼ What is the common reuse principle?

- **Classes that aren't reused together should not be grouped together.**
- In other words, when one or more classes within a package change, the release number of the package changes, too.

▼ What does component-level design do?

Component-level design guides the construction phase (coding and testing) of the development process.

▼ What steps are involved in component-level design?

1. Identify all design classes that correspond to the problem domain.
2. Identify all design classes that correspond to the infrastructure domain.
3. Elaborate all design classes that are not acquired as reusable components.
  - a. Specify message details when classes or components collaborate.
  - b. Identify appropriate interfaces for each component.
  - c. Elaborate attributes and define data types and data structures required to implement them.
  - d. Describe processing flow within each operation in detail.

4. Describe persistent data sources (databases and files) and identify the classes required to manage them.
5. Develop and elaborate behavioral representations for a class or component.
6. Elaborate deployment diagrams to provide additional implementation detail.
7. Refactor every component-level design representation and always consider alternatives.

▼ What models are typically used in user interface design?

1. The user model, created by the software engineer
2. The design model, created by the software engineer
3. The mental model or system perception model, created by the end user
4. The implementation model, created by the implementers of the system

Note: It is the interface designer's role to reconcile these differences so that a consistent representation of the interface can be developed.

▼ What tasks are involved in the user interface design process?

- interface analysis and modeling
- interface design
- interface construction
- interface validation

▼ What should be understood in user experience analysis?

- The end users who will interact with the system
- The tasks the end users will perform to do their work
- The content presented as part of the interface
- The environment in which the tasks will be performed

▼ Google has defined a five-day sprint for doing User Experience (UX) design. What are the steps?

1. The team gathers information on the problem (in other words, the user needs and business goals).
2. Team members, including stakeholders, brainstorm solutions to the problem.
3. Solution sketches are presented, and the team votes on which to use for the following day's prototype.
4. A prototype is created based on the sketch selected the previous day.
5. Users try out the prototype to discover issues with its UX design. This leads to the next iteration of this process.