Human Computer Interaction

DESIGN PRINCIPLE & PROCESS

By: Nguyễn Công Hoan

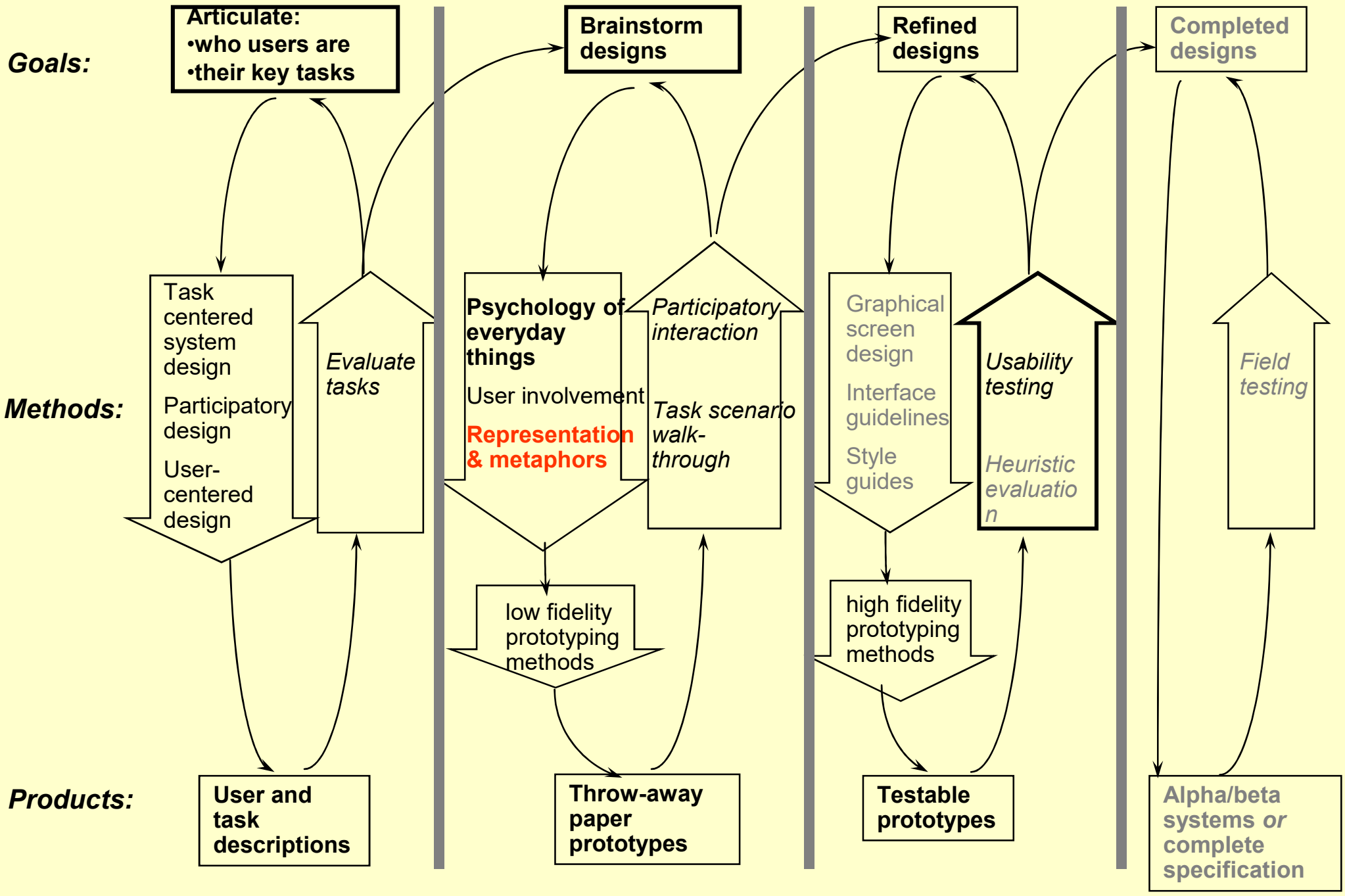
Reference

- DonaldNorman, **The Design of Everyday Things, MITPress, 23 Dec 2013**
- **Tutorial Teaching** of Prof. Dr. Keith Andrews, Graz University of Technology
- **Tutorial Teaching** of Lecture Nguyen Van Vu, University of Sciences
-

Agenda

- What is Interface?
- Interface Hall of Shame
- User Centered Design
- Principles of UI design
- Shneiderman’s Eight Golden Rules
- Design process

Interface Design and Usability Engineering



- What is Interface?

Interface is made up of a set of hardware devices and software tools from the computer side and a system of sensory, motor and cognitive processes from the human side.

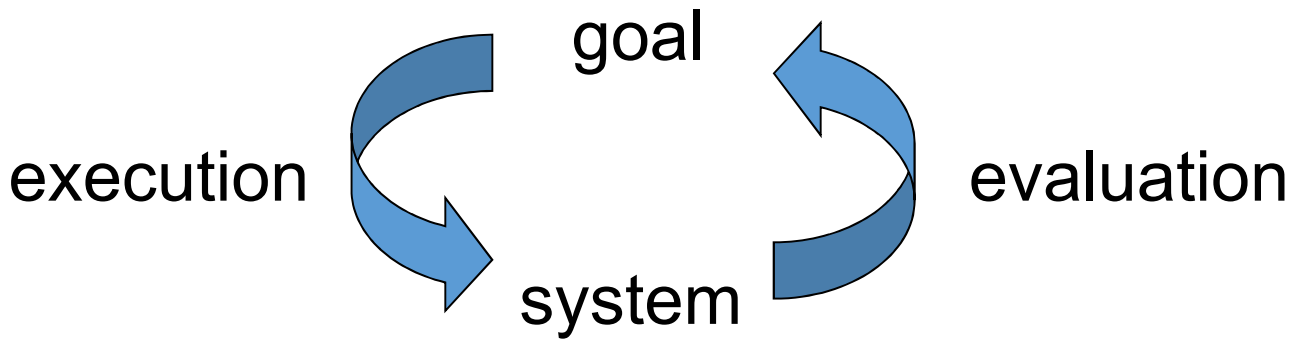
Interaction takes place at the ***Interface***,

Donald Norman's model

- Norman's model concentrates on user's view of the interface

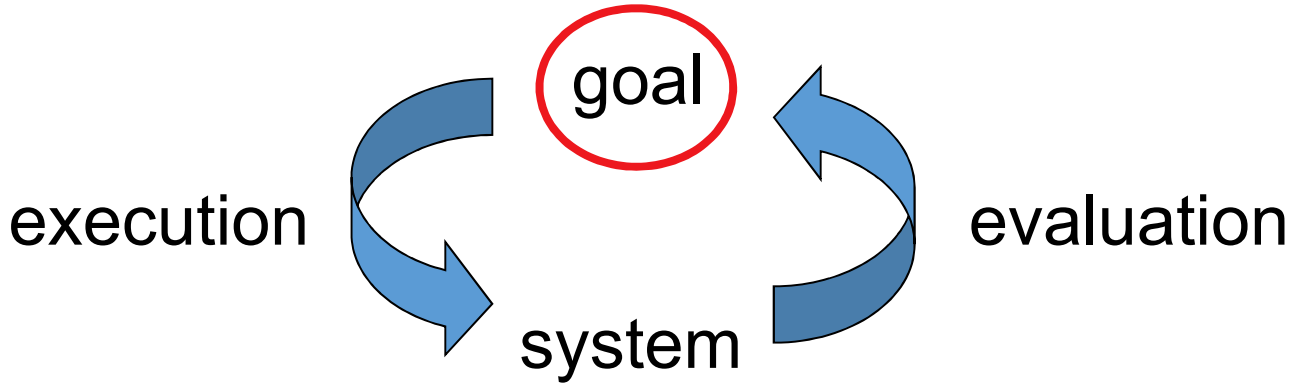
- Seven stages
  - User establishes the goal
  - Formulates intention
  - Specifies actions at interface
  - Executes action
  - Perceives system state
  - Interprets system state
  - Evaluates system state with respect to goal

Execution/Evaluation Loop



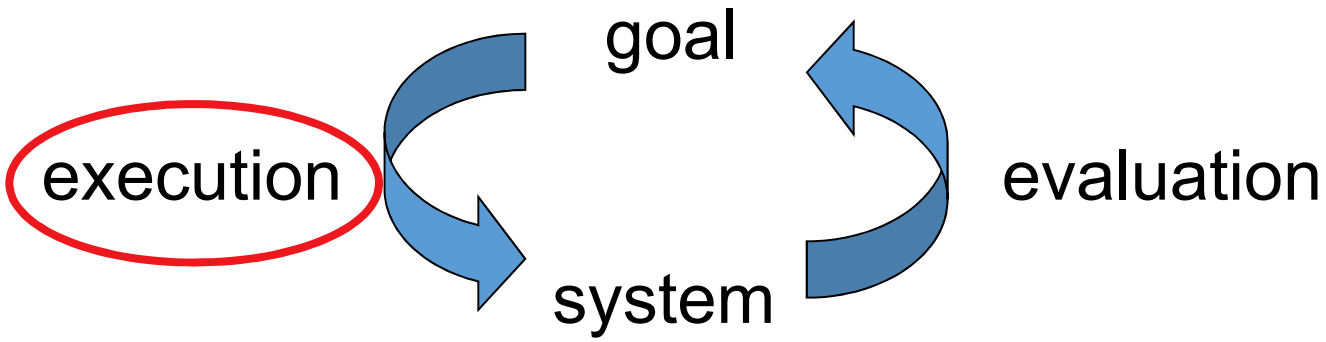
- User establishes the goal
- Formulates intention
- Specifies actions at interface
- Executes action
- Perceives system state
- Interprets system state
- Evaluates system state with respect to goal

Execution/Evaluation Loop



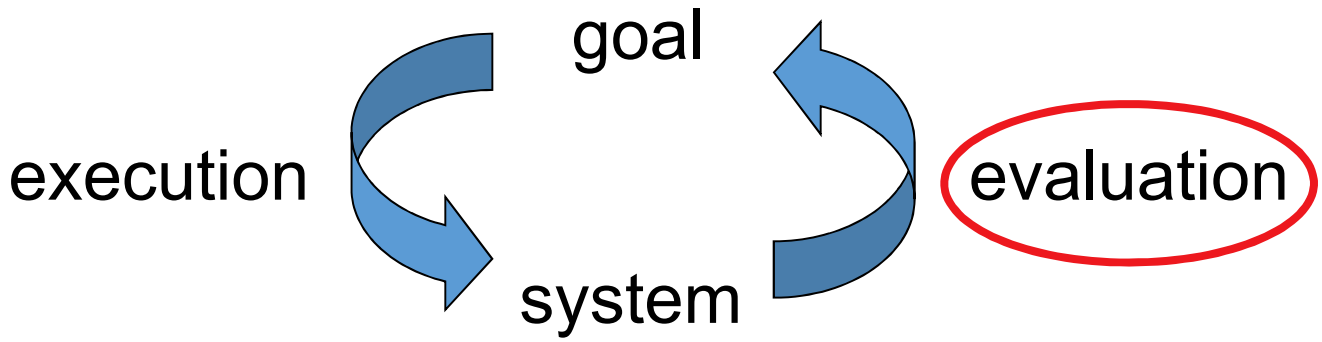
- User establishes the goal
- **Formulates intention**
- **Specifies actions at interface**
- **Executes action**
- **Perceives system state**
- **Interprets system state**
- **Evaluates system state with respect to goal**

Execution/Evaluation Loop



- **User establishes the goal**
- Formulates intention
- Specifies actions at interface
- Executes action
- **Perceives system state**
- **Interprets system state**
- **Evaluates system state with respect to goal**

Execution/Evaluation Loop



- **User establishes the goal**
- **Formulates intention**
- **Specifies actions at interface**
- **Executes action**
- Perceives system state
- Interprets system state
- Evaluates system state with respect to goal

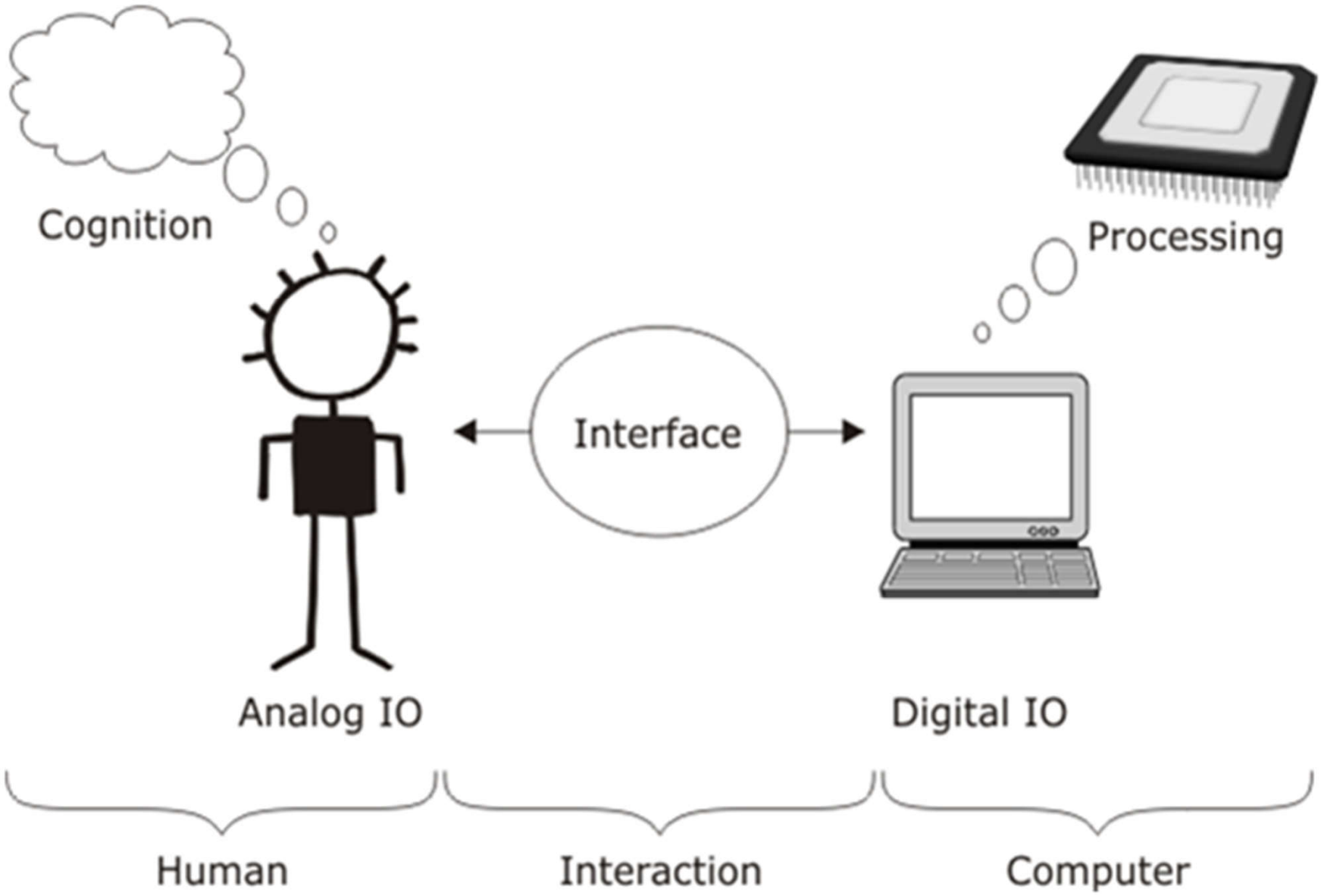
User interface (UI)

User interface: User interfaces mediate the interaction (**dialog**) between humans and computers.

The User Interface today is often one of the most critical factors regarding the success or failure of a computer system

Good UI design:

- Increases efficiency
- Improves productivity
- Reduces errors
- Reduces training
- Improves acceptance



User Interface

"Today, user needs are recognized to be important in designing interactive computer systems, but as recently as 1980, they received little emphasis."

J. Grudin

A balance of two key features is needed for an effective user interface

- **Expressive**: ability to achieve specific tasks efficiently
- **Intuitive**: ease of use, degree of effort required to learn

Basic user interface styles

- Five commonly encountered user interface paradigms:

Interface style

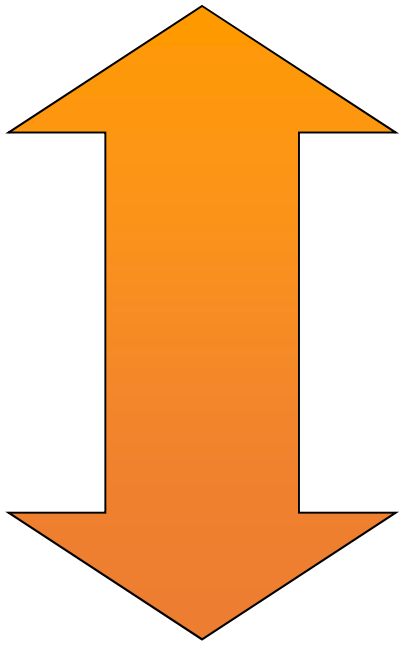
Command entry

Menu

Forms

WIMP

Natural language



Expressive

Intuitive

System-Centered Design

- What can be built easily on this platform?
- What can I create from the tools available?
- What do I as a developer find interesting to work on?
- What do I as a developer think users need?

User-Centered Design

- Also known as Participatory Design
- The design is based upon a user's:
  - Abilities and Needs
  - Context
  - Work
  - Tasks

User-Centered Design (cont.)

How to do?

- Focusing on users and tasks
  - User analysis: who uses the system
  - Task analysis: what users need to do
- Getting users involved in the process
  - Users as evaluators, consultants and designers (sometimes)
- Constant evaluation
  - Users evaluate prototypes and releases

User-Centered Design (cont.)

- Advantages
  - Accurate information and useful suggestions
  - Opportunity to argue over design decisions
  - Increased ego involvement in system success
- Potential problems
  - Users are not always available to participate
  - Their time maybe expensive
  - Users are not UI designers
  - Users have strong ego and preferences
  - UI designers overly obey users' preference

Principles of UI design

- Simple and natural dialogue
- Speak the user's language
- Minimize user's memory load
- Provide feedback
- Provide clearly marked exits
- Provide shortcuts
- Deal with errors in a positive manner
- Provide help

Principles of UI design (cont.)

How to apply?

- Determine users' skill levels
  - Novice/first-time users
  - Knowledgeable intermittent users
  - Experts and frequent users
- Identify the tasks
  - Frequent actions
  - Less frequent actions
  - Infrequent action

Principles of UI design (cont.)

How to apply?

- Choose appropriate interaction styles
  - Direct manipulation
  - Menu selection
  - Form fill in
  - Command language
  - Natural language
- Use Shneiderman's eight golden rules of interface design

Principles of UI design (cont.)

How to apply?

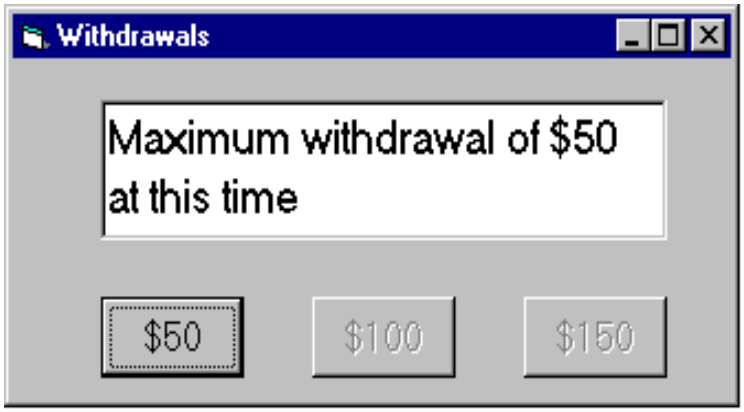
- Prevent errors
  - Constructive and informative error messages
  - Organizing screens and menus functionally
  - Providing feedback about the state of the interface
  - Correct actions
    - E.g., grayed menu items
  - Complete sequences
    - E.g., wizard windows often have both Next and Finish buttons
- Increase automation while preserving human control
  - Auto suggestion
  - Auto completion
  - Allowing users to change

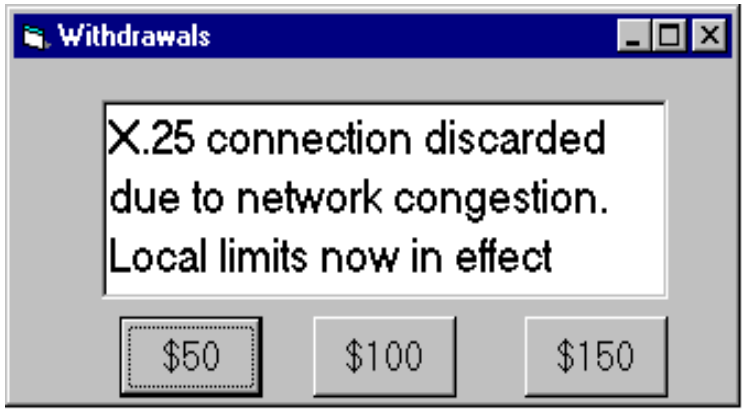
Shneiderman's Eight Golden Rules

- Strive for consistency
- Cater to universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors, rapid recovery
- Permit easy reversal of actions
- support user control
- Reduce memory load

Example: Speak the users' language

- Terminology based on users' language for task
  - e.g. withdrawing money from a bank machine

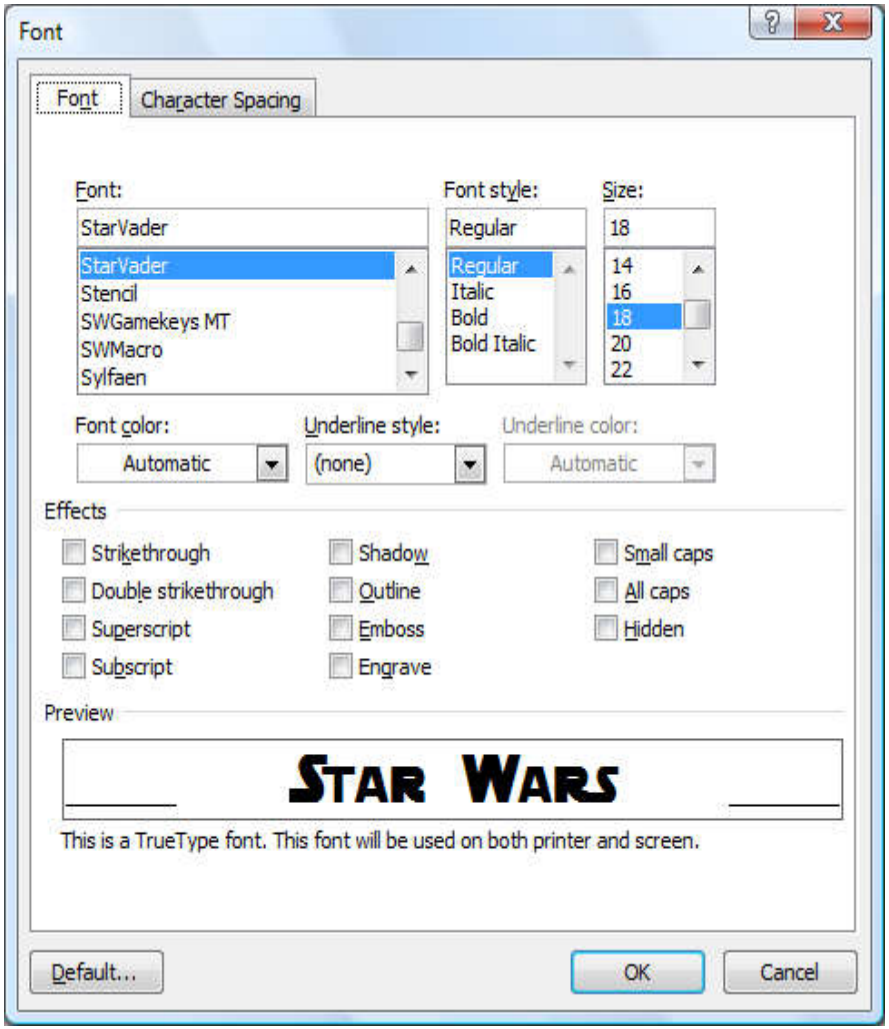




- Use meaningful mnemonics, icons & abbreviations
  - e.g File / Save

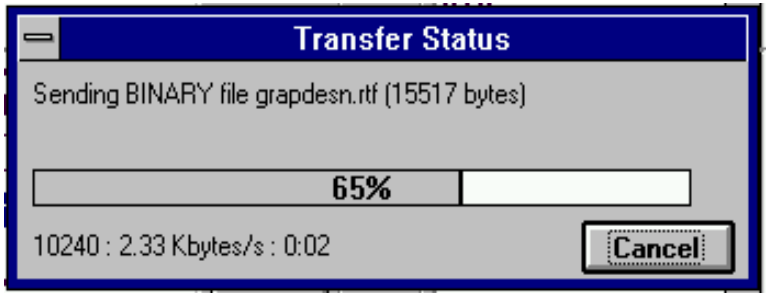
Example: Minimize user's memory load

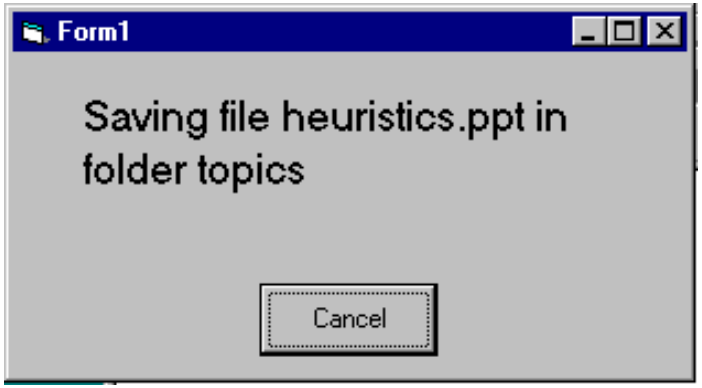
- Computers good at remembering, people are not!
- Promote Recognition over Recall
  - menus, icons, choice dialog boxes vs. commands, field formats
  - relies on visibility of objects to the user (but less is more!)



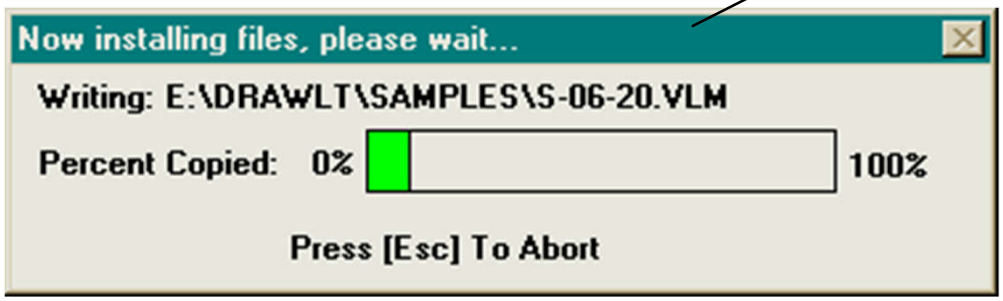
Example : Provide feedback

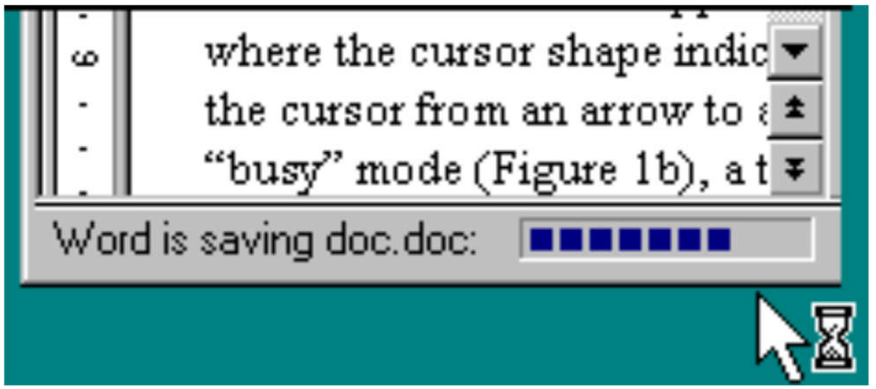
- Continuously inform the user about
  - what it is doing
  - how it is interpreting the user's input
  - user should always be aware of what is going on





Multiple files being copied,
but feedback is file by file.





Tips for Interface Designs

- HCI has traditionally been about designing efficient and effective systems.
- Well-designed interfaces can elicit good feelings in users.
- Expressive interfaces can provide comforting feedback.
- Badly designed interfaces make people angry and frustrated.
- Emotional interaction is concerned with how we feel and react when interacting with technologies.
- Emotional interaction is concerned with how interactive systems make people respond in emotional ways.
- Relaxed users will be more forgiving of shortcomings in design.
- Aesthetically pleasing and rewarding interfaces will increase positive affect.

Tips for Interface Designs (cont.)

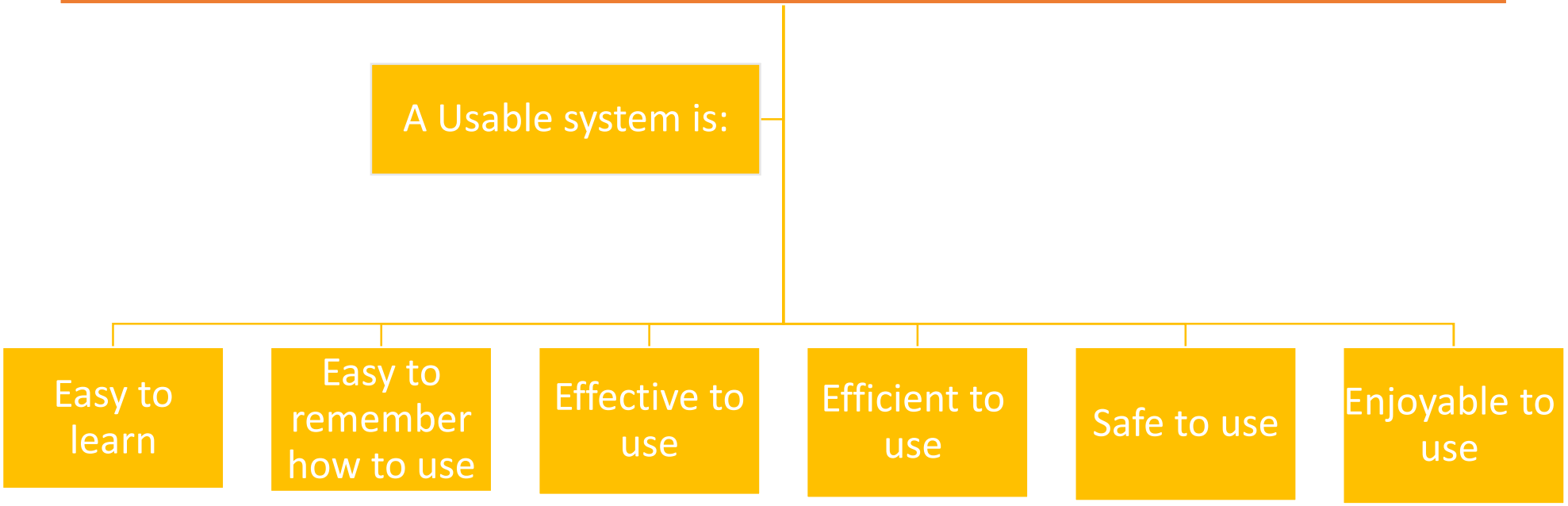
- User interfaces should be designed to match the skills, experience and expectations of its anticipated users.
- System users often judge a system by its interface rather than its functionality.
- A poorly designed interface can cause a user to make terrible errors.
- Poor user interface design is the reason why so many software systems are never used.
- Designers should be aware of people's physical and mental limitations (e.g. limited short-term memory) and should recognise that people make mistakes.

Basic Goal of HCI

USABILITY

One of the key concepts in HCI.

It is concerned with making systems easy to learn and use



In order to produce computer system with good usability;
Developers must attempt to

Understand	Develop	Achieve	Put People 1st
• The factors that determine how people use technology	• Tools and techniques to enable building suitable systems	• Efficient, effective, and safe interaction	• Their needs, capabilities and preferences for conducting various tasks should direct developers in the way that they design systems • People should not change their way they use the system to fit with it, instead system should match their requirements

The long term goal:
To design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task

Why is usability important?

- Poor usability results in
  - anger and frustration
  - decreased productivity in the workplace
  - higher error rates
  - physical and emotional injury
  - equipment damage
  - loss of customer loyalty
  - costs money

Principles of UI design (cont.)

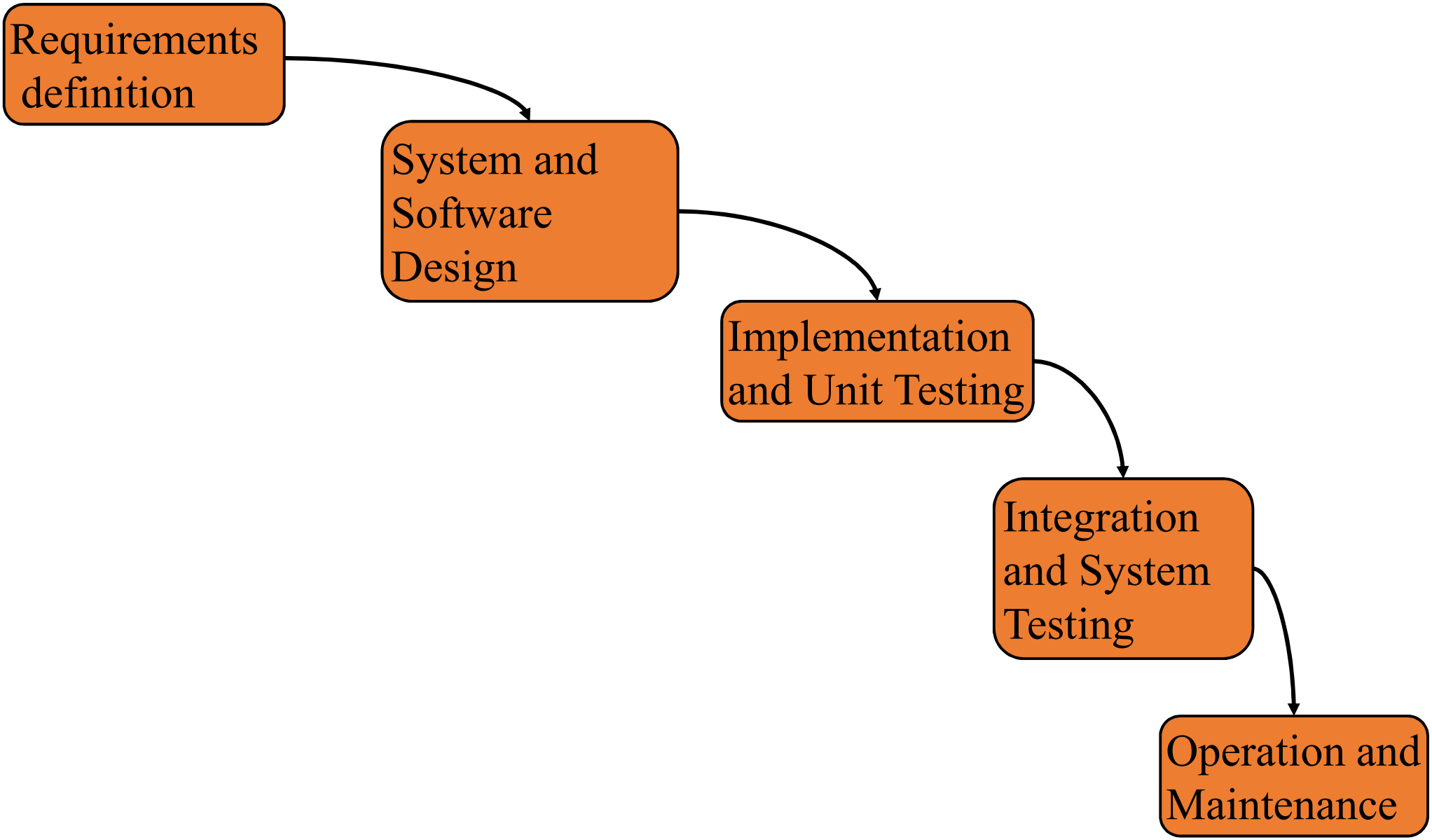
How to apply?

- Choose appropriate interaction styles
  - Direct manipulation
  - Menu selection
  - Form fill in
  - Command language
  - Natural language
- Use Shneiderman’s eight golden rules of interface design

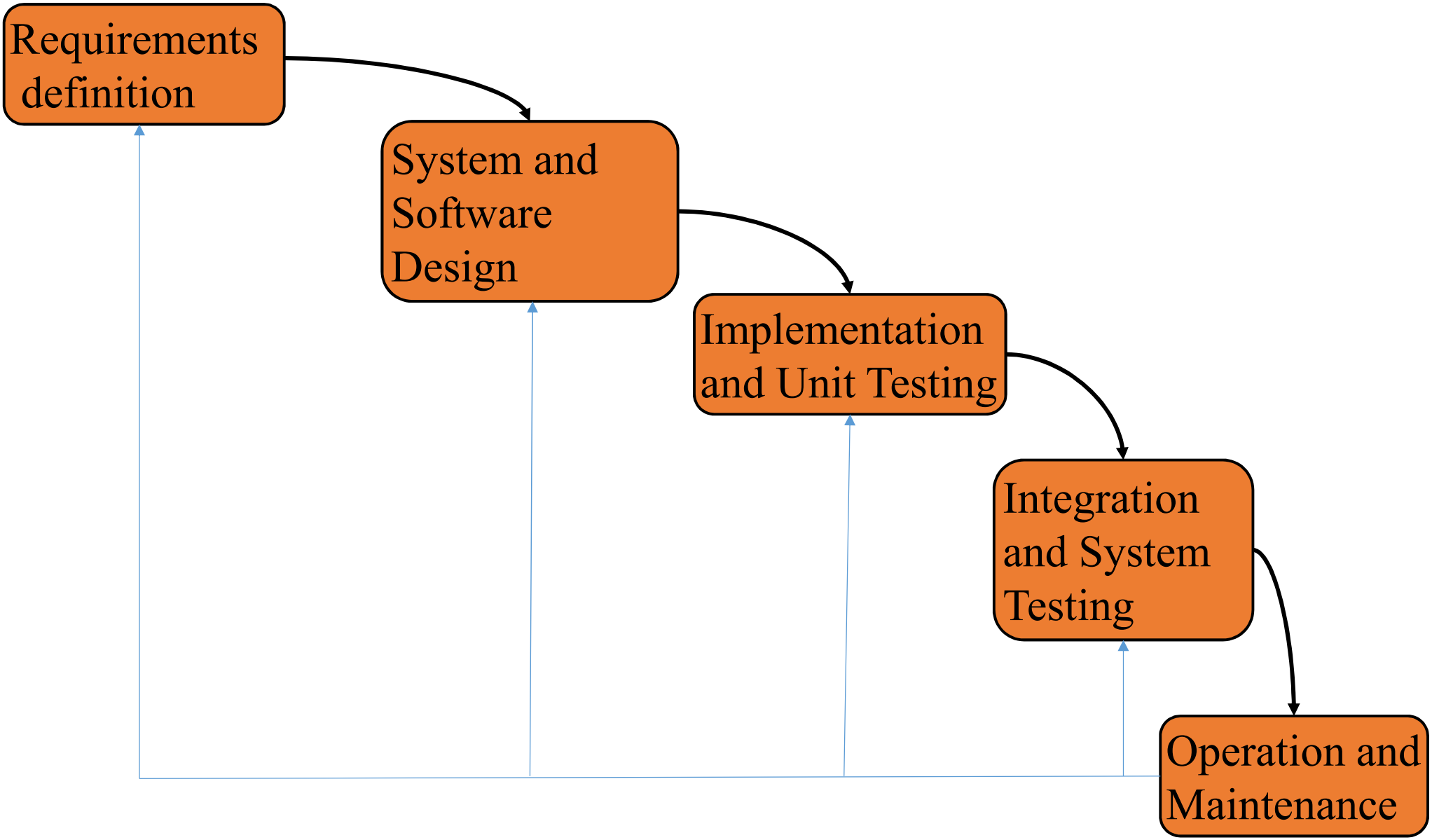
Design Process

- Waterfall model
- Iterative design
- Spiral model
- What to choose?

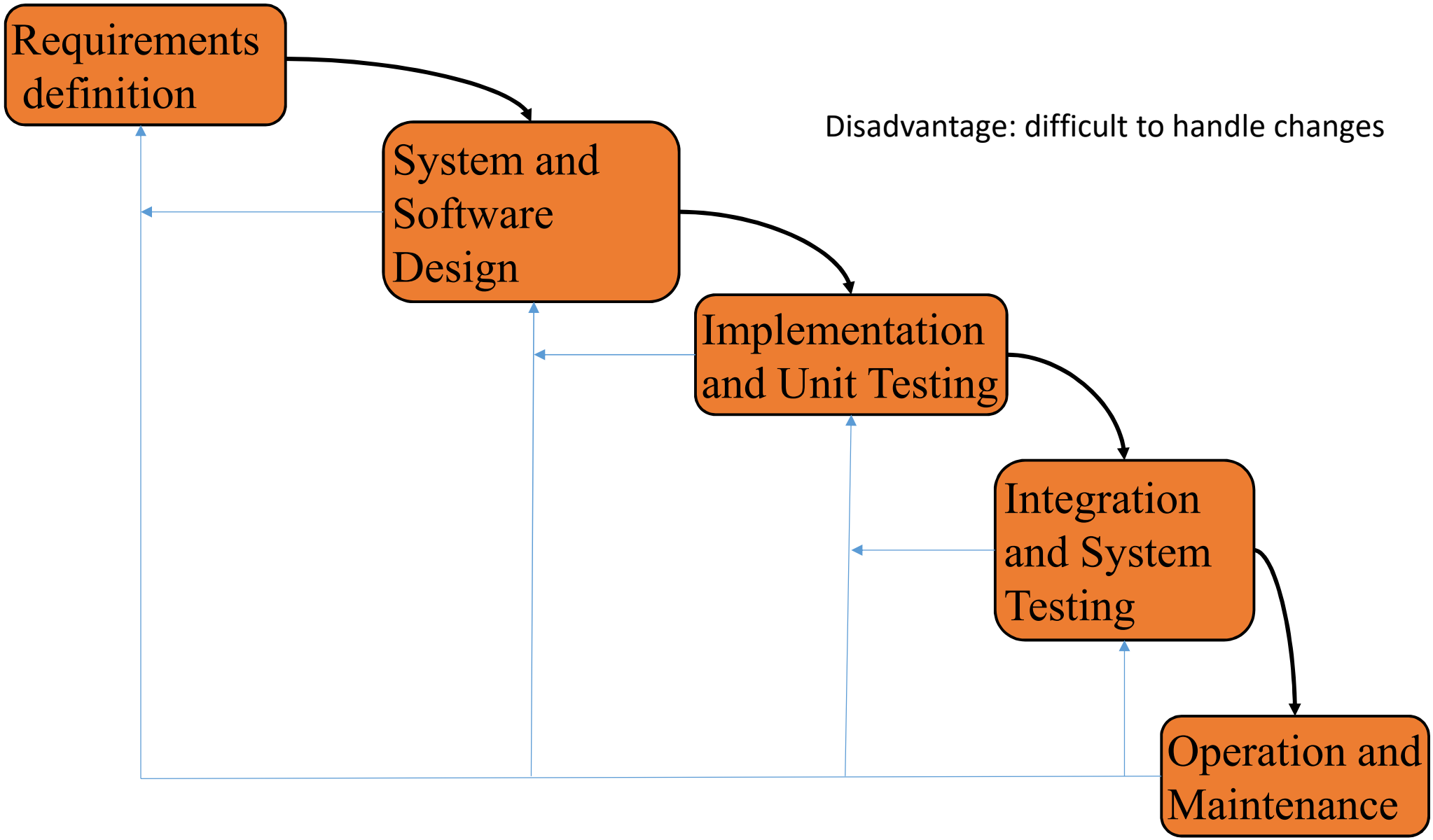
Waterfall model



Waterfall model (cont.)



Waterfall model (cont.)



Disadvantage: difficult to handle changes

Waterfall Model Problems

- Users are not involved in evaluation until acceptance testing
- UI problems result in changes in requirements and design
  - Waste of effort spent earlier
- Inflexible partitioning of the project into distinct stages
  - It is difficult to respond to changing customer requirements
- It is only appropriate when the requirements are well understood
  - Few business systems have stable requirements

Shneiderman's Interactive Systems Lifecycle

Software development lifecycle for interactive systems:

1) Collect Information
2) Define Requirements and Semantics
3) Design Syntax and Support Facilities
4) Specify Physical Devices
5) Develop Software
6) Integrate System and Disseminate to Users
7) Nurture the User Community
8) Prepare Evolutionary Plan

Collect Information

- Organize the design team
- Obtain management and customer participation
- Conduct interviews with users
- Submit written questionnaires to users
- Estimate development, training, usage, maintenance costs
- Prepare a schedule with observable milestones and reviews

Define requirements and semantics

- Define high-level goals and middle-level requirements
- Consider task flow sequencing alternatives
- Create task objects and actions
- Obtain management and customer agreement on goals,
- requirements, and semantic design

Design syntax and support facilities

- Compare alternative display formats
- Design informative feedback for each operation
- Review, evaluate, and revise design specifications
- Carry out paper-and-pencil pilot tests or field studies with an
- online mock-up or prototype

Specify physical devices

- Choose hard- or softcopy devices
- Select audio, graphics, or peripheral devices
- Consider work environment noise, lighting, table space, etc.
- Carry out further pilot tests and revise design

Develop software

- Use appropriate development tools
- Develop code
- Perform unit test

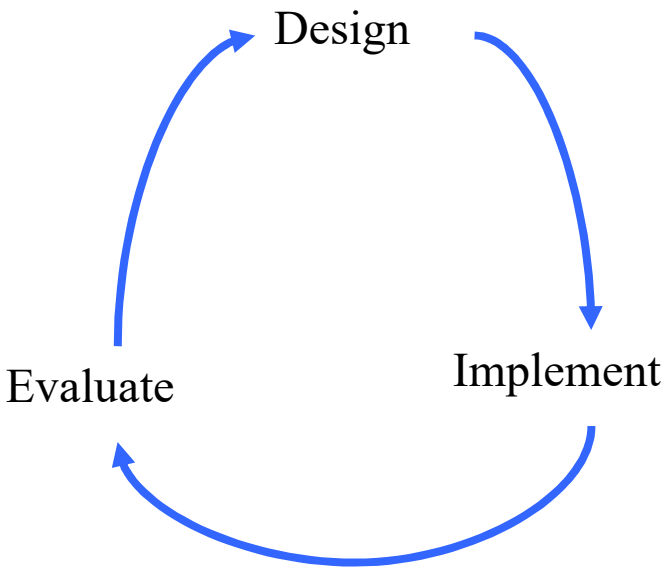
Integrate system and disseminate to users

- Assure user involvement at every stage
- Conduct acceptance tests and fine tune the system
- User documentation and training

Nurture the user community

- User support
- Monitor usage and measurement

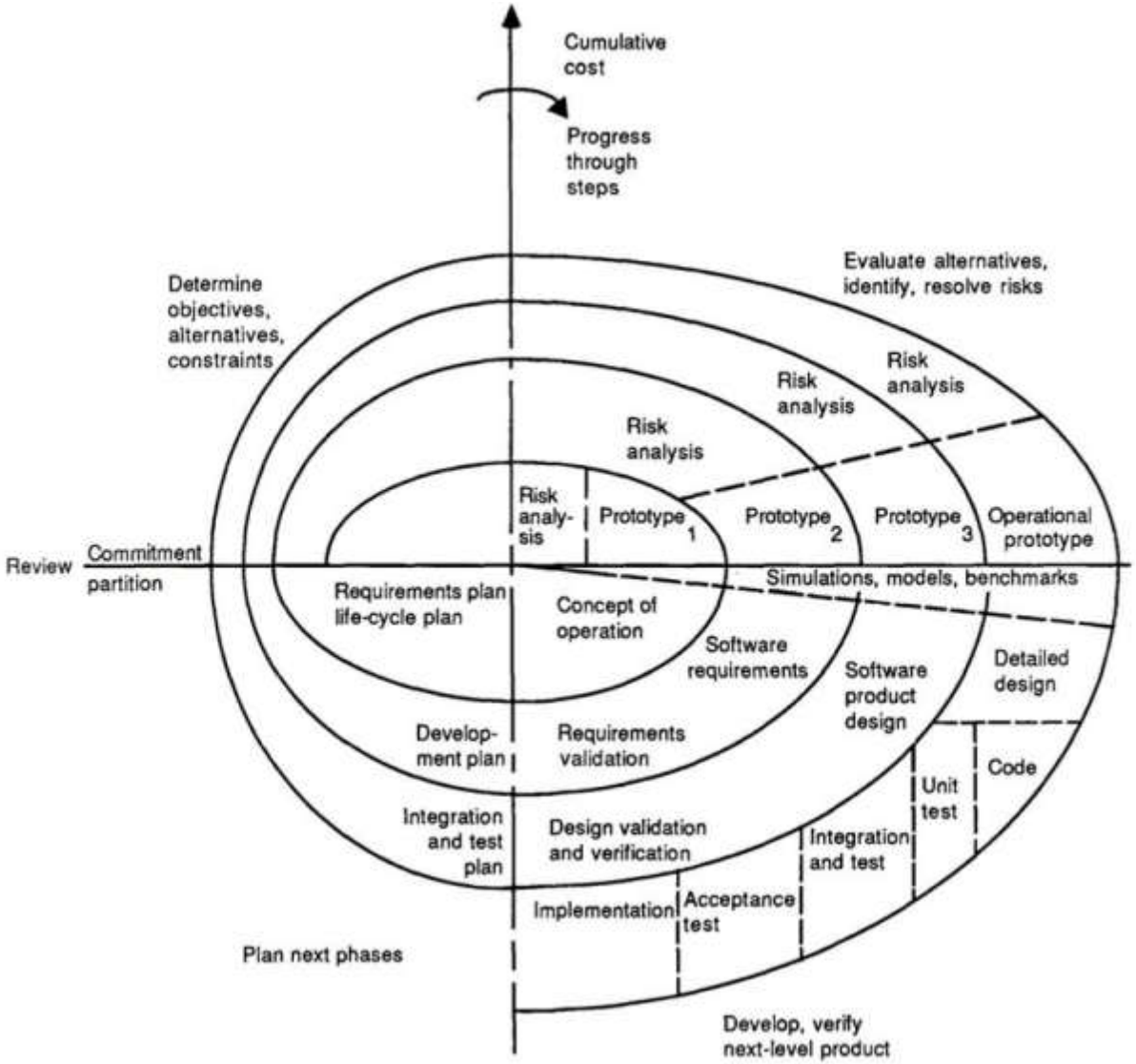
Iterative Design



Iterative Design (cont.)

- Each round is one iteration
  - Release is produced at the end of each iteration
  - Customer's feedback and evaluations can be
  - incorporated into next release
- Advantages:
  - Simple
- Problems
  - It's expensive to use customer's time to test
  - Customers may not be available
  - Customers don't like → they don't buy

Spiral Model



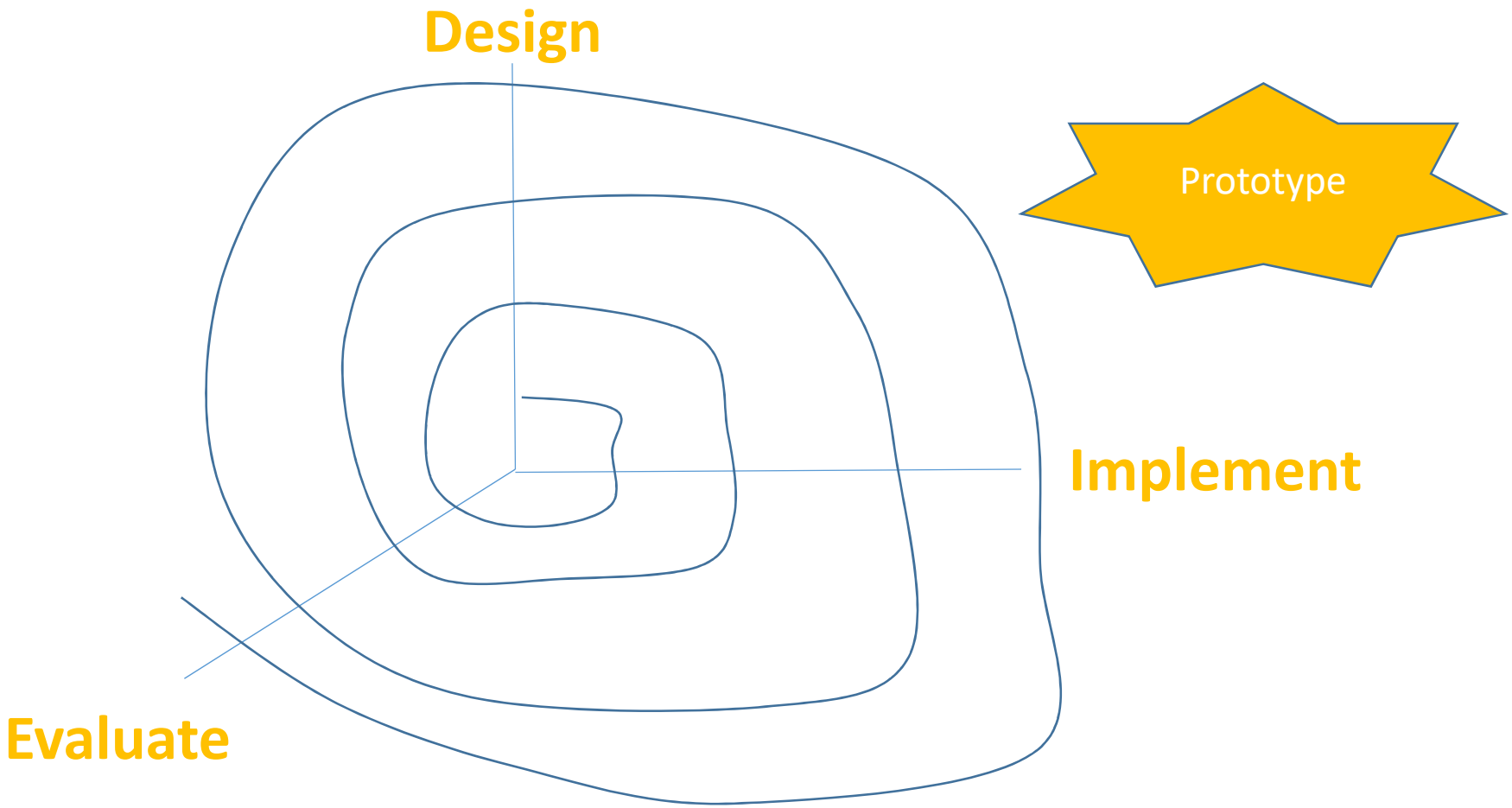
(Boehm 1988)

Spiral Model (cont.)

- Process is represented as a spiral rather than as a sequence of activities with backtracking
- Each loop in the spiral represents a phase in the process
- No fixed phases such as specification or design loops in the spiral are chosen depending on what is required
- Risks are explicitly assessed and resolved throughout the process

Increment Model

Increment Model = Iterative Design + Spiral Model



Increment Model for UI Design

- Early cycles use cheap prototypes
  - Paper prototypes
  - Sketches on computer
  - Quick prototyping tools
- Providing multiple prototype alternatives
  - Parallel prototyping
- Later cycles should be better than early ones
- Only mature releases of later cycles can be distributed to users
- Easy to integrate with Software Development process Scum