User Interface Design Principles

Lecture 5

Credits: These slides influenced by
Prof. Björn Hartmann's CS 160 lecture slides.

Human-Computer Interaction (HCI)

Human-Computer Interaction (HCI)

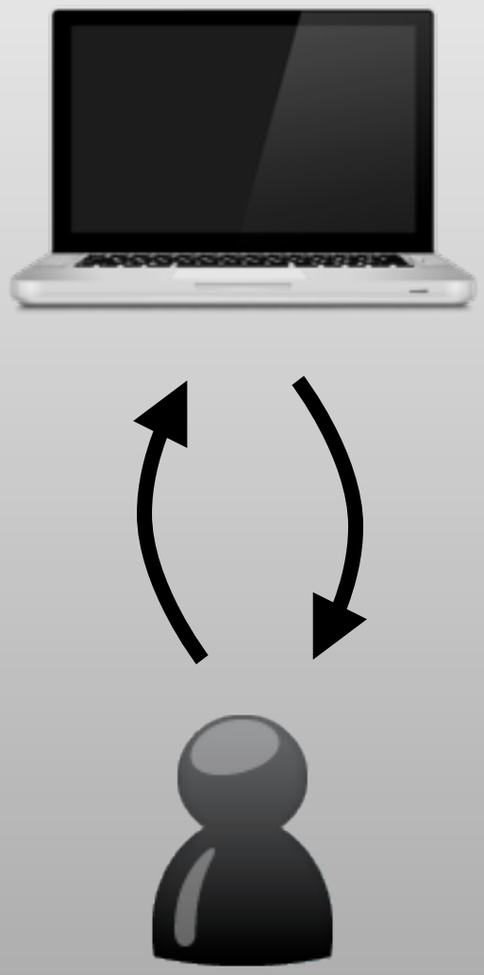
- Computer
  - Machine(s) the application runs on
- Often split between client & server

Human-Computer Interaction (HCI)

- Computer
  - Machine(s) the application runs on
    - Often split between client & server

- Human
  - The end-user of the application
  - It's somebody else!

Human-Computer Interaction (HCI)



- Interaction
  - Conversation between the human and the computer
- User gives instructions to the computer
- Computer communicates responses

What is a user interface?

What is a user interface?

- The part of an application where
  - The users tell the computer what they want
  - The computer gives results

What is a user interface?

- The part of an application where
  - The users tell the computer what they want
  - The computer gives results
- Not limited to the display!
  - Includes hardware, such as buttons, switches, sensors (accelerometers, gyroscopes, cameras)

What is usability?

What is usability?

- Ease of learning

What is usability?

- Ease of learning
- Recall

What is usability?

- Ease of learning
- Recall
- Productivity

What is usability?

- Ease of learning
- Recall
- Productivity
- Minimal error rates

What is usability?

- Ease of learning
- Recall
- Productivity
- Minimal error rates
- High user satisfaction

Design Cycle

Prototype

Design

Evaluate

Task Analysis & Contextual Inquiry

- Observe actual users doing tasks related to that you aim to solve
- Create scenarios where your product would be used
- Gain insight into existing work processes

Task Analysis & Contextual Inquiry

- Don't guess!
  - You don't know everything (sorry, it's true!)
- Observe *real users* in *real scenarios*
  - Ask them questions

Rapid Prototyping

Rapid Prototyping

- Mock up your design

Rapid Prototyping

- Mock up your design
- Begin with low-fidelity techniques
  - Paper sketches, Post-Its, etc.

Rapid Prototyping

- Mock up your design
- Begin with low-fidelity techniques
  - Paper sketches, Post-Its, etc.
- Interactive prototyping tools
  - HTML/CSS/JavaScript, Flash, Visual Basic

Rapid Prototyping

- Mock up your design
- Begin with low-fidelity techniques
  - Paper sketches, Post-Its, etc.
- Interactive prototyping tools
  - HTML/CSS/JavaScript, Flash, Visual Basic
- High-fidelity UI construction
  - Interface Builder, Visual Studio, NetBeans

Evaluate

- Analytically (no users)
  - Expert evaluation
    - "Rules of thumb" guidelines
- Test with target users
  - Informal user tests
  - Controlled usability studies

The Waterfall Model (Software Engineering)

Initiation

↓

Application Description

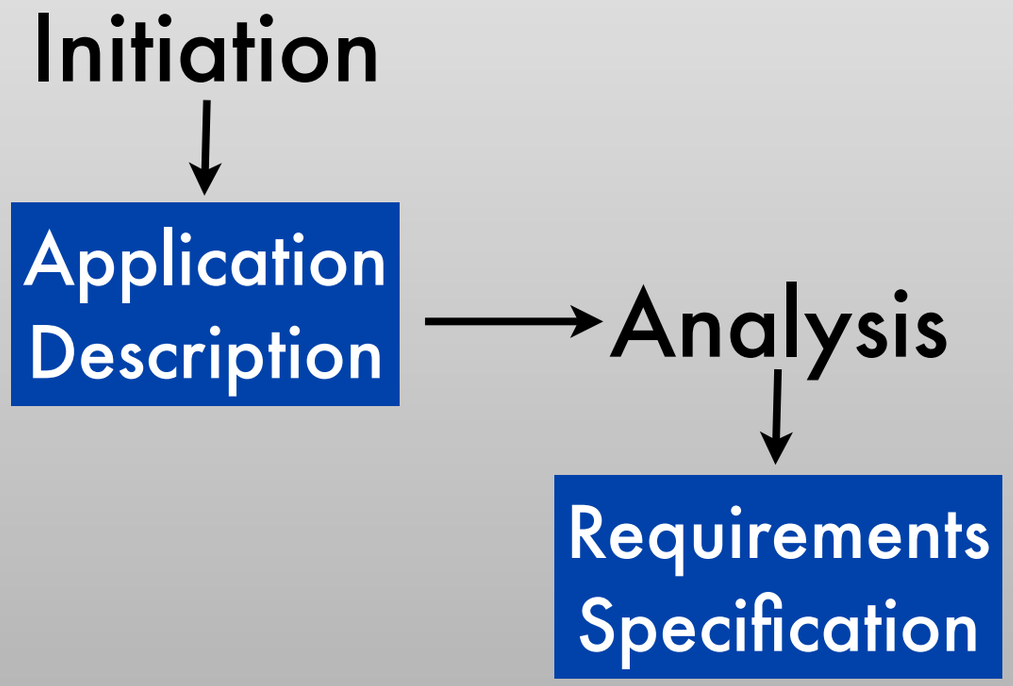
The Waterfall Model (Software Engineering)

Initiation

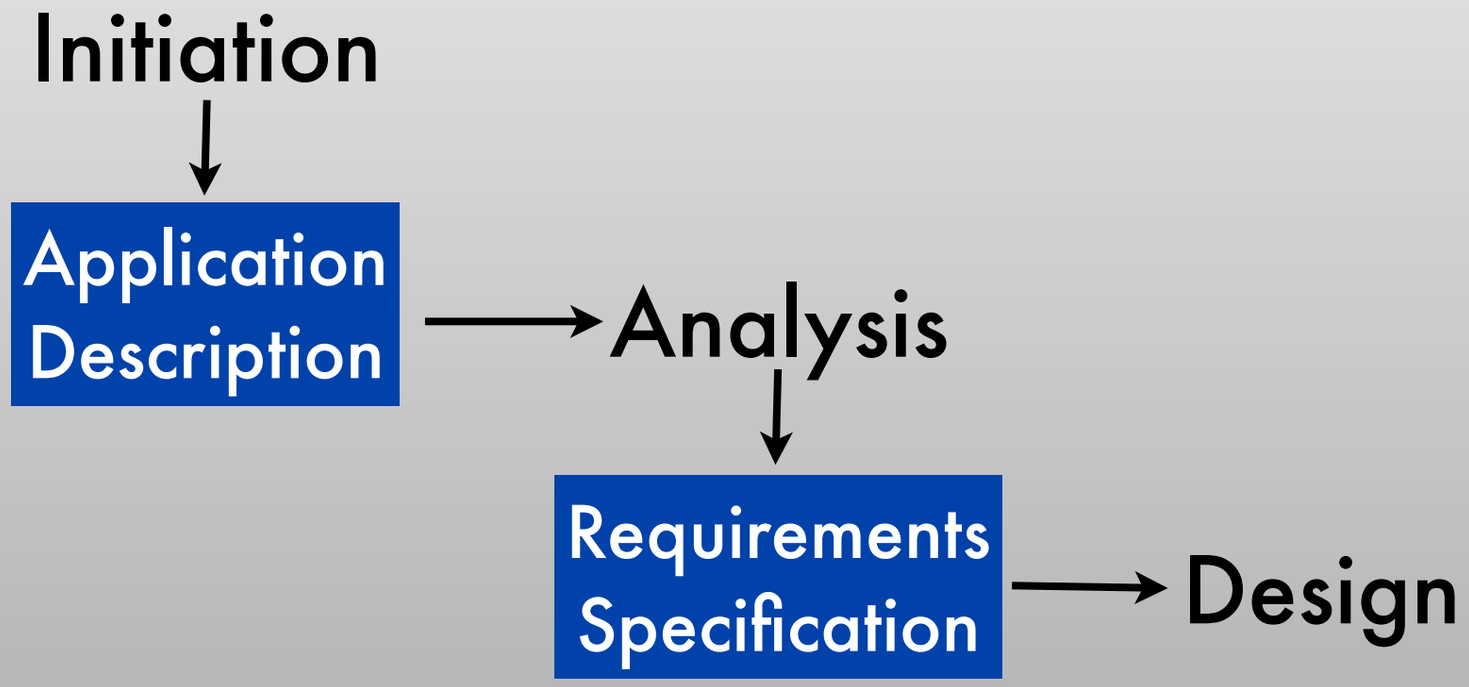
↓

Application Description → Analysis

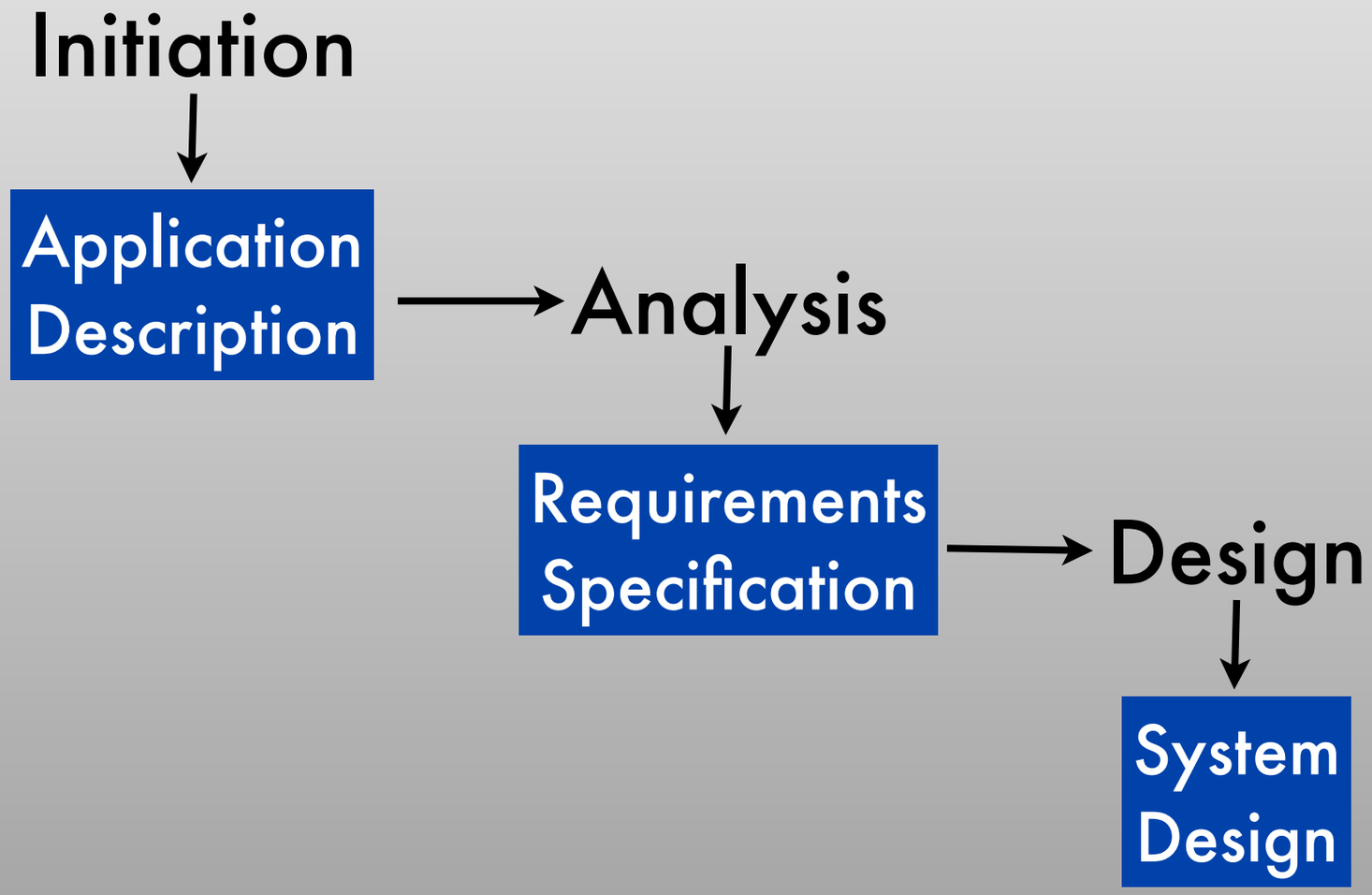
The Waterfall Model (Software Engineering)



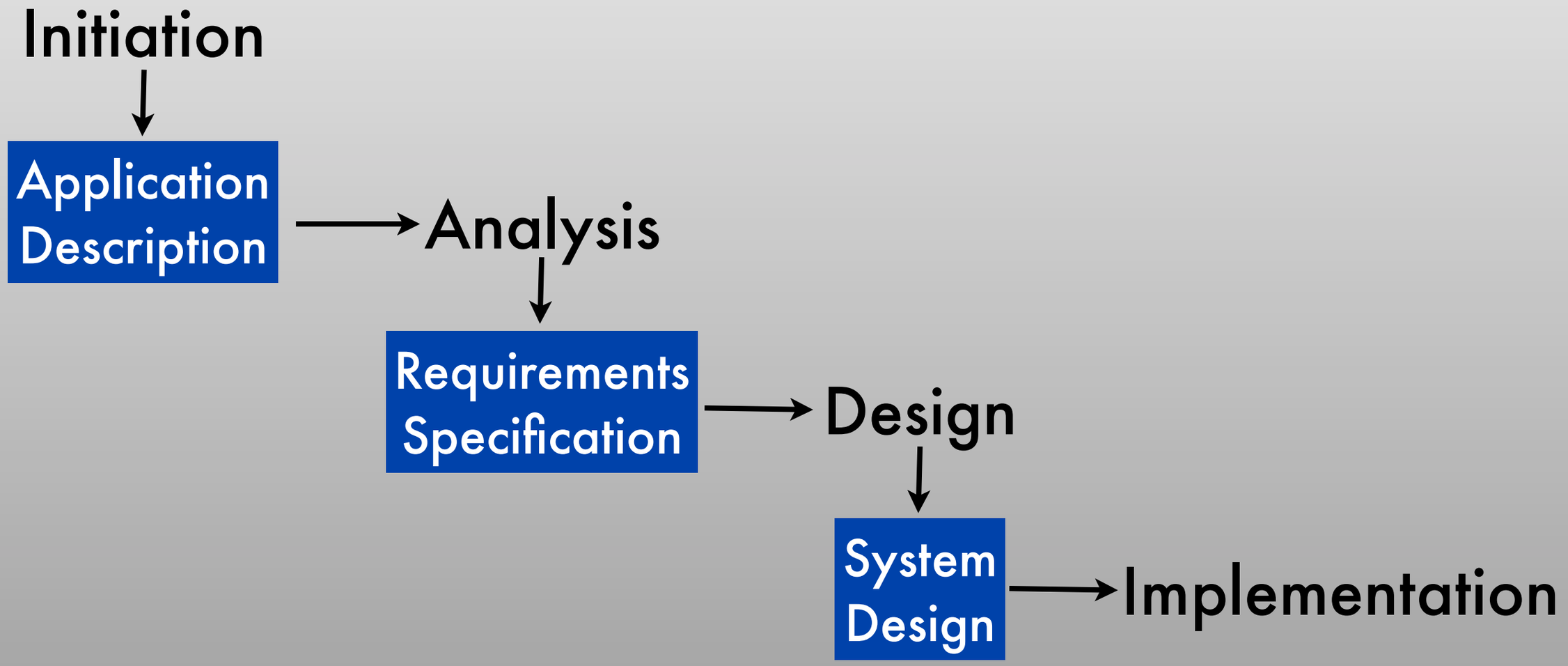
The Waterfall Model (Software Engineering)



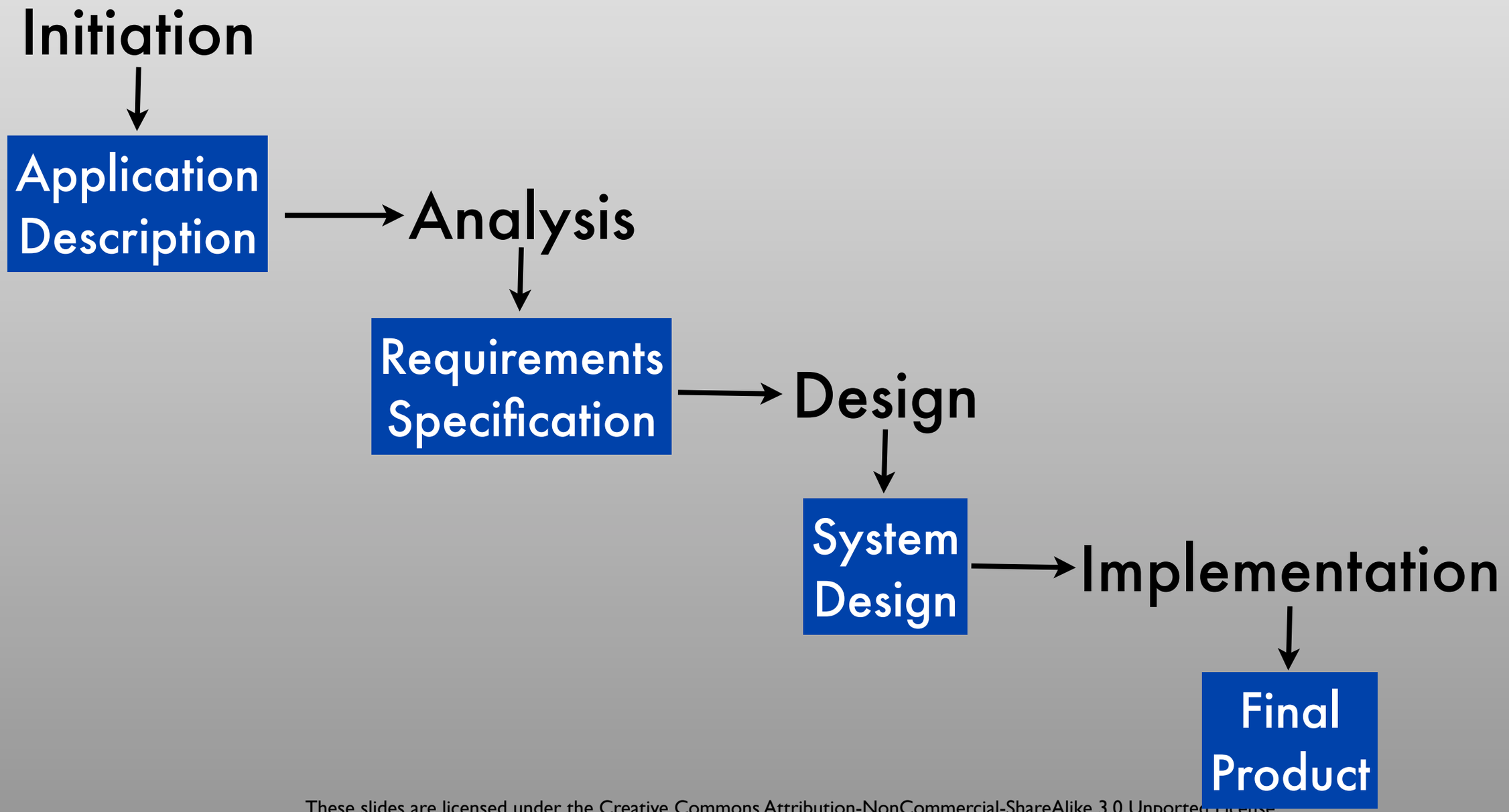
The Waterfall Model (Software Engineering)



The Waterfall Model (Software Engineering)



The Waterfall Model (Software Engineering)



(UI) Design Principles

From *The Design of Everyday Things*,
Don Norman

Make Things Visible

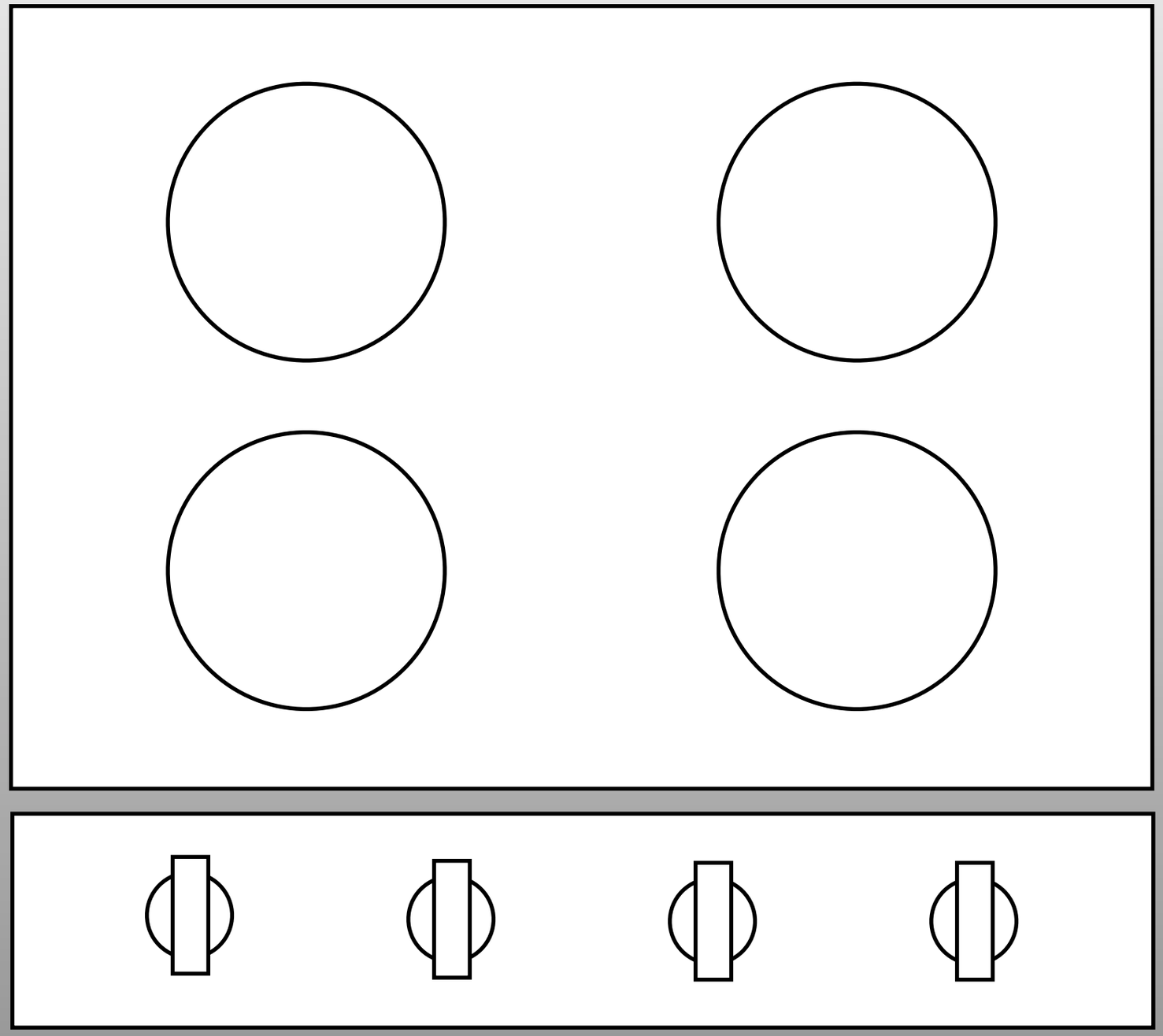
Make Things Visible

- "The correct parts must be visible, and they must convey the correct message."[1]
- "Whenever the number of functions and required operations exceeds the number of controls, the design becomes arbitrary, unnatural, and complicated."[2]

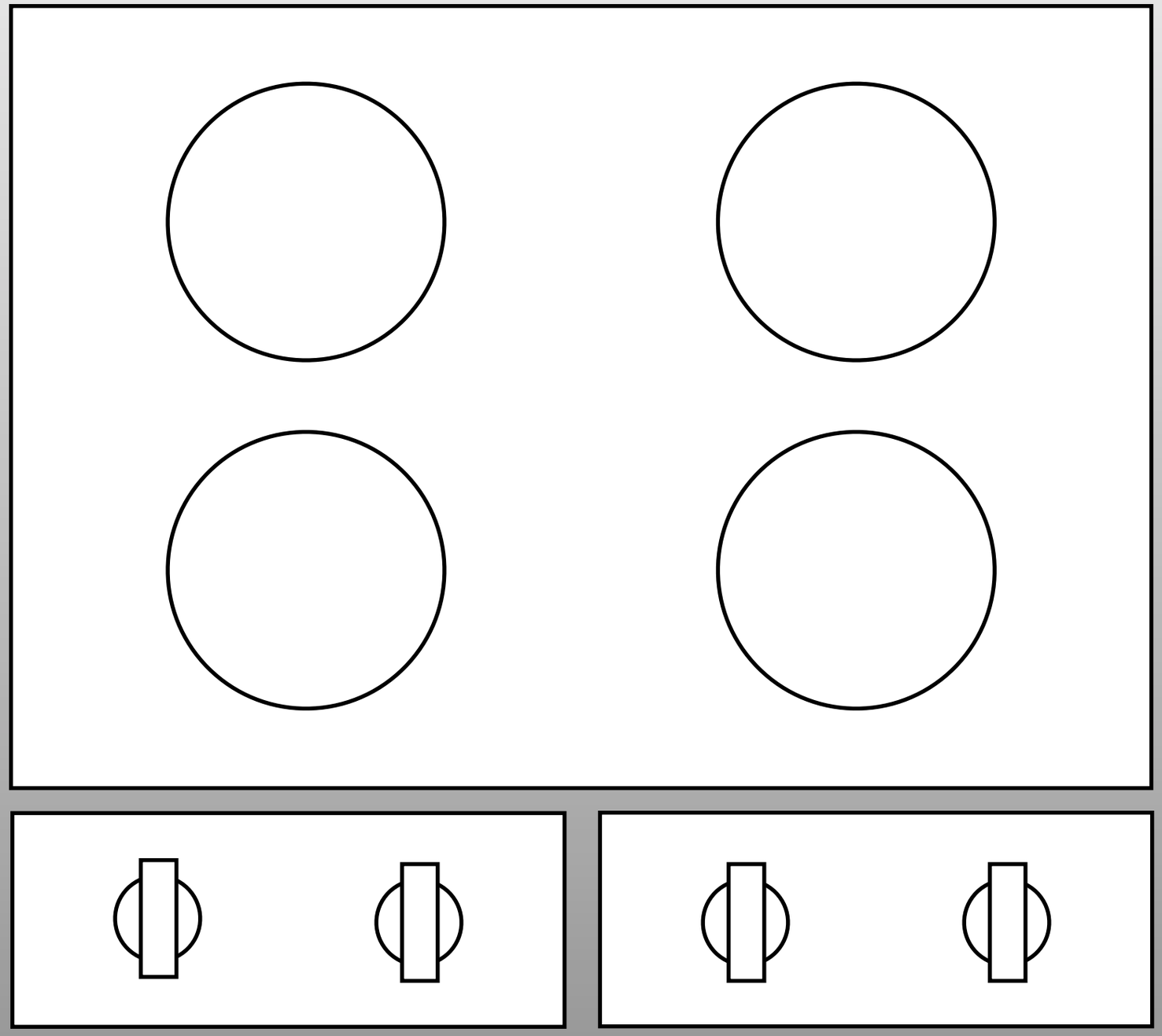
1. *The Design of Everyday Things*, page 4
2. *DOET*, page 31.

Establish a Clear Mapping

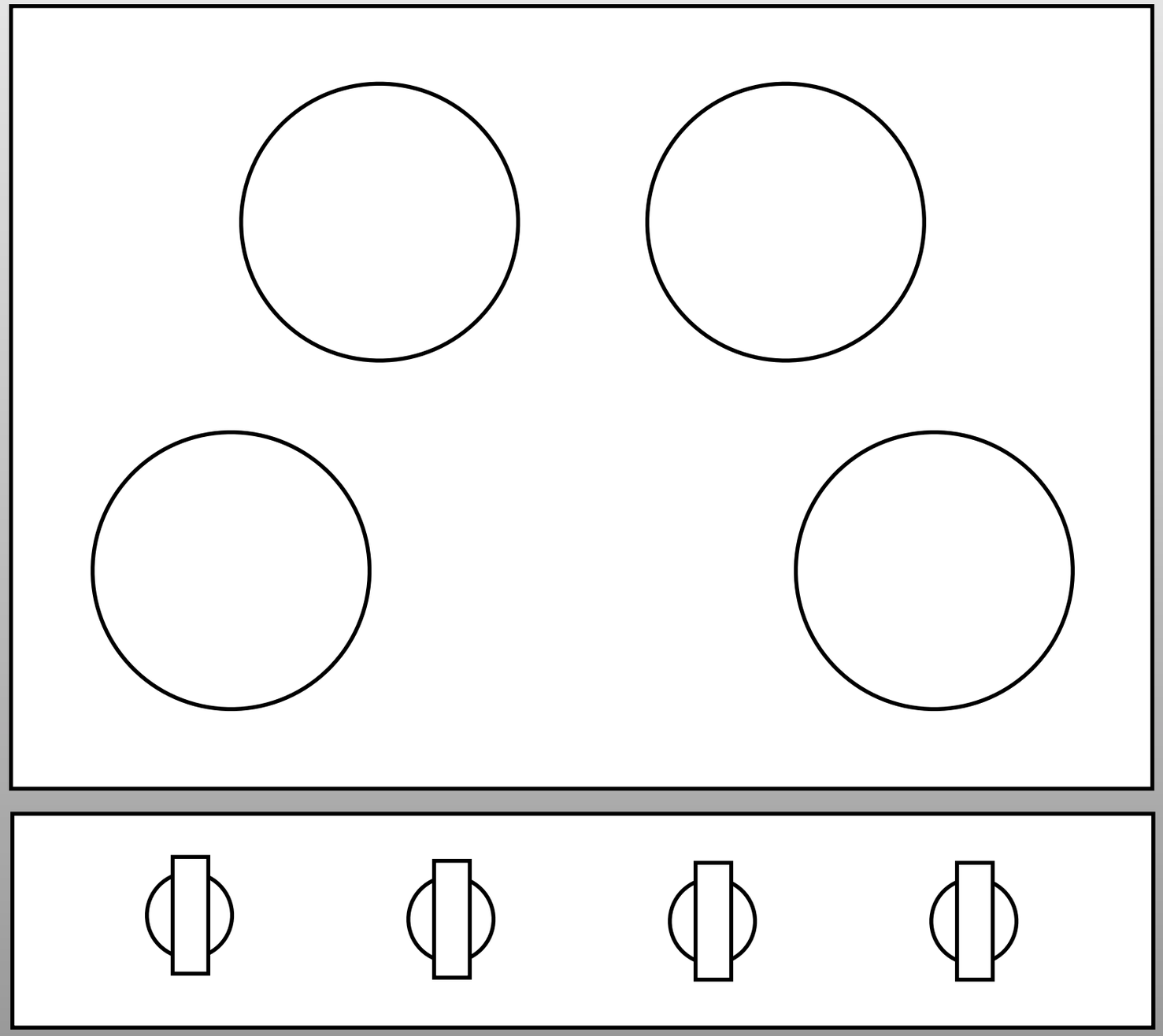
Which knob controls which burner?



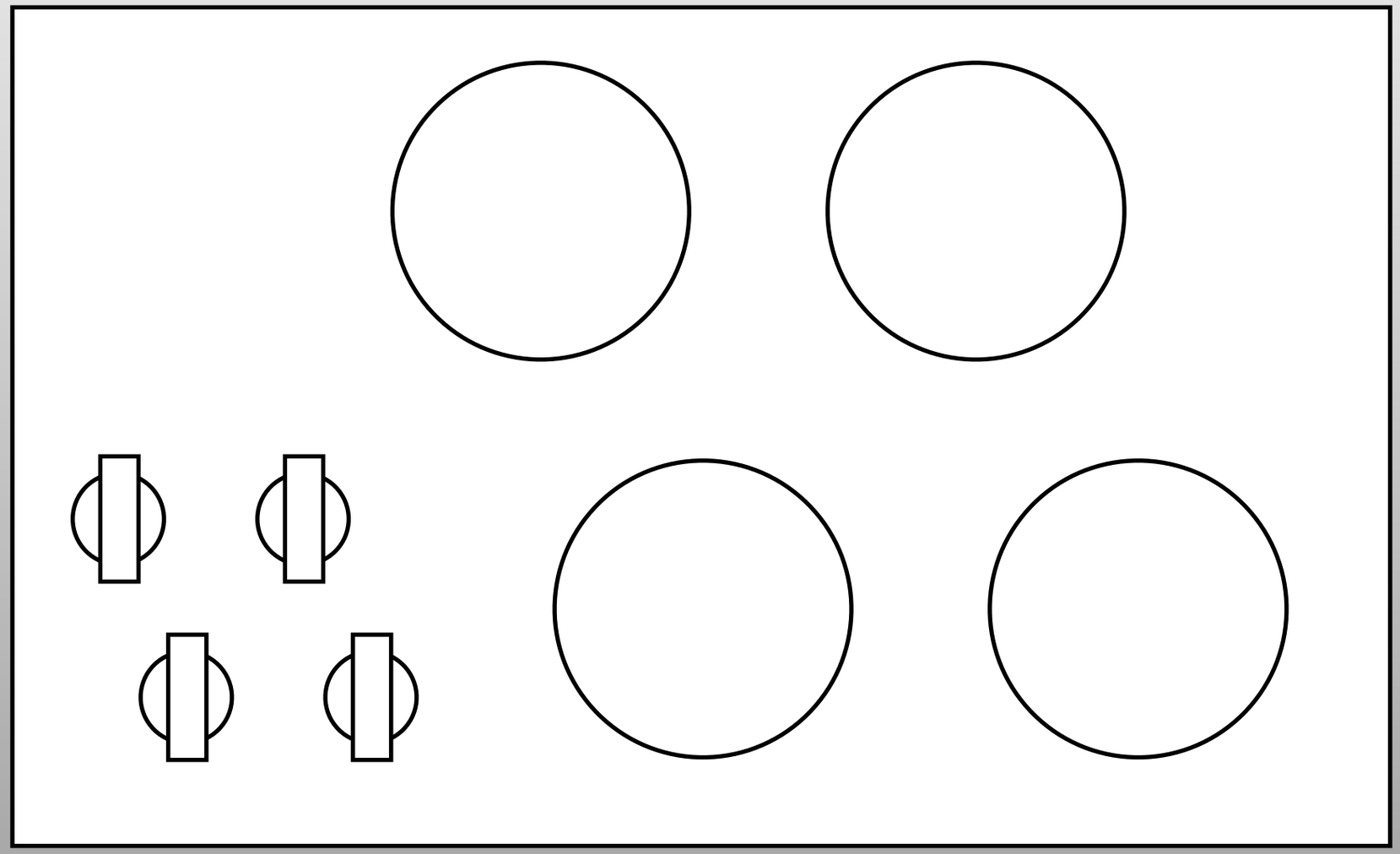
Which knob controls which burner?



Which knob controls which burner?



Which knob controls which burner?



Give Feedback

Give Feedback

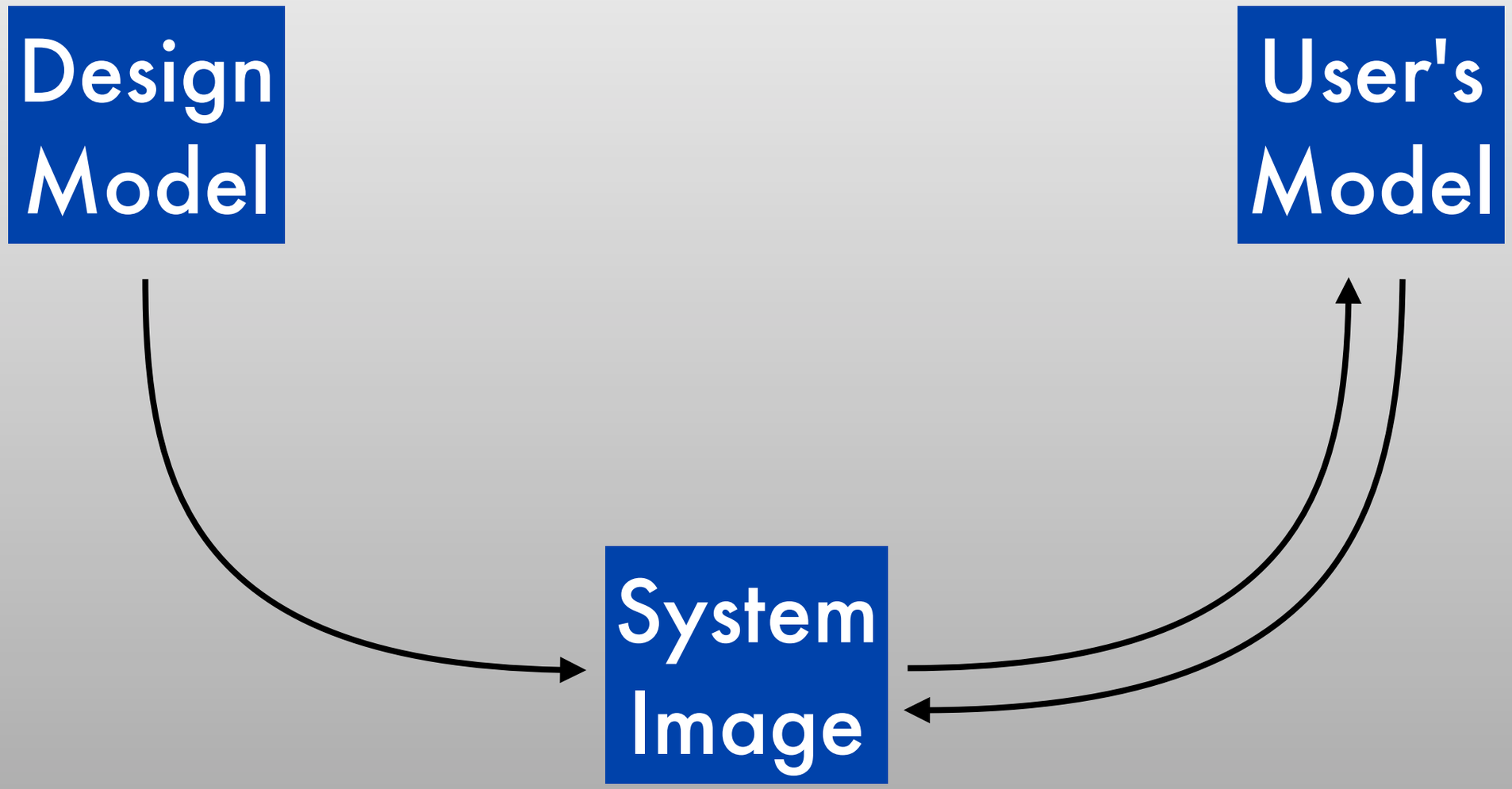
- The telephone
  - Dial tone tells the user the phone is connected
  - The user's own voice is fed back into the earpiece
- Crosswalk buttons (usually) don't give good feedback
  - People often press the button multiple times

Conceptual Models

Conceptual Models

- Humans create "mental models" of complex processes, including computer applications.
- It is important to encourage the user to form a mental model that closely matches your internal model.
  - Your application will be easier to use if users' actions do what they expect.

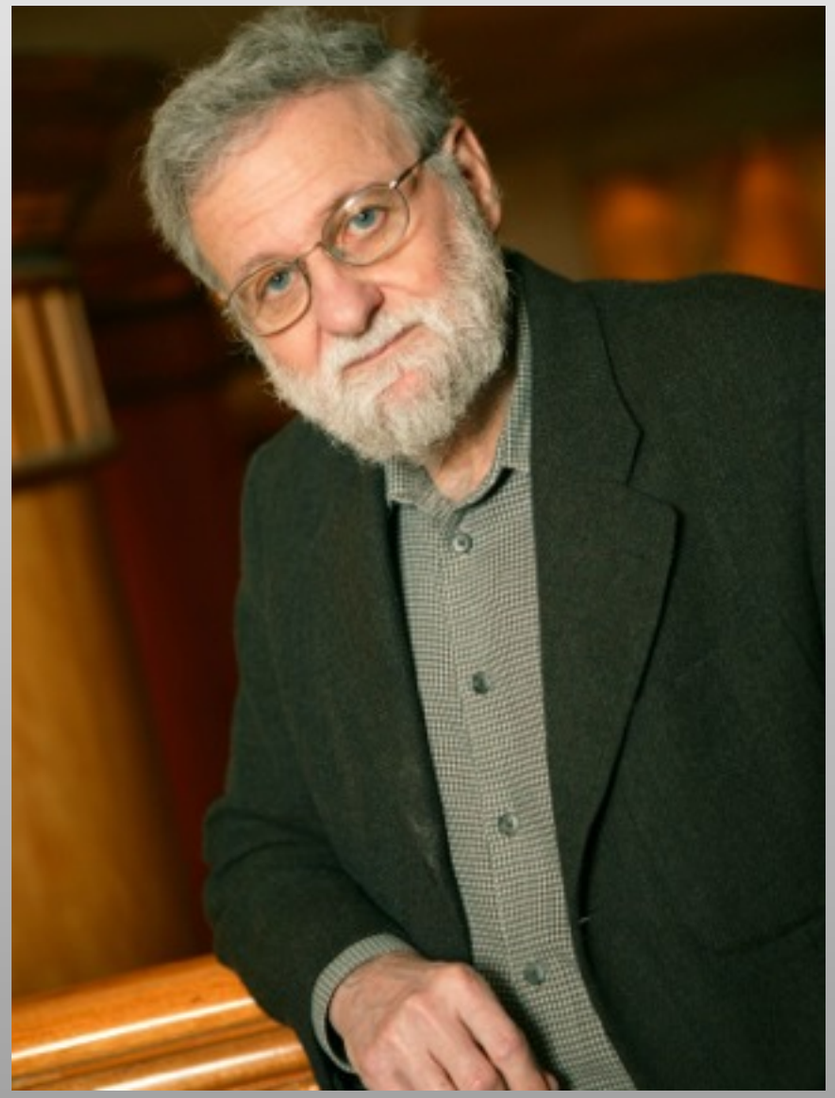
Conceptual Models



- Designer's model might not match the user's
- Users only work with the system image

Affordances

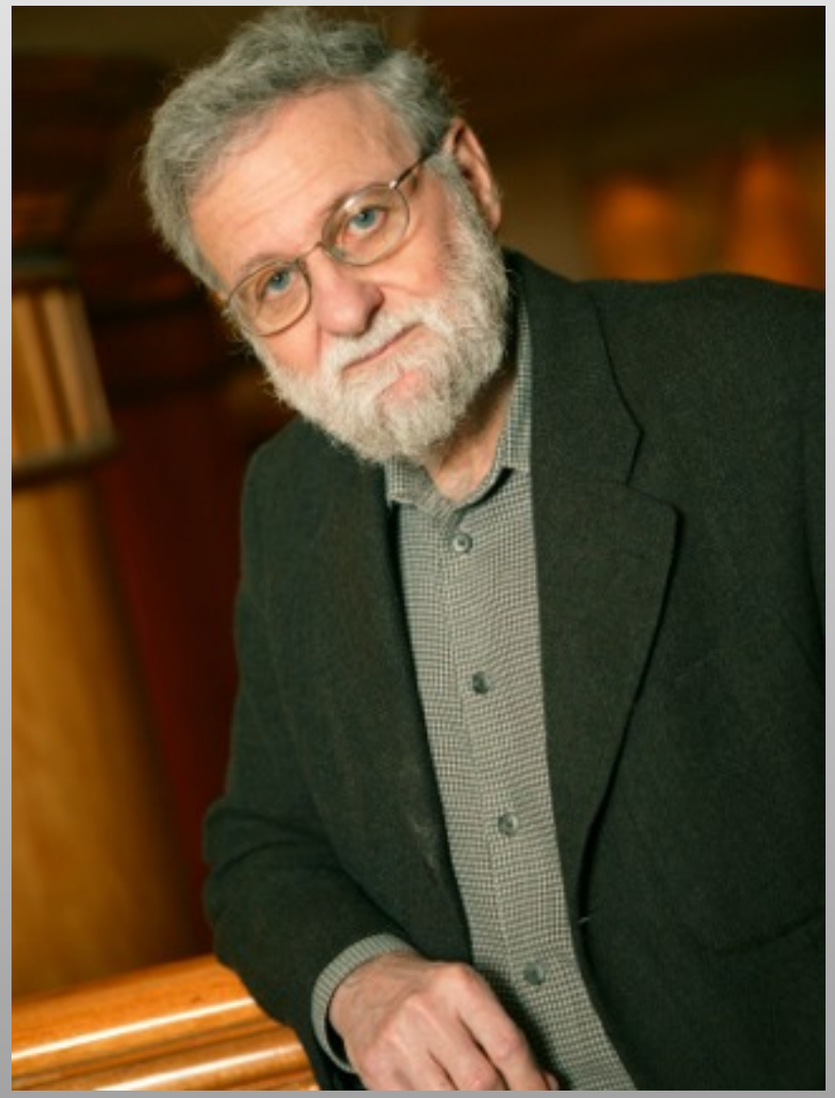
- "... the term *affordance* refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used."



The Design of Everyday Things
Don Norman

Affordances

- Knobs afford turning
- Buttons afford pushing
- Glass can be seen through



The Design of Everyday Things
Don Norman

Affordances

- Give clues about how the object/interface is supposed to work

Affordances

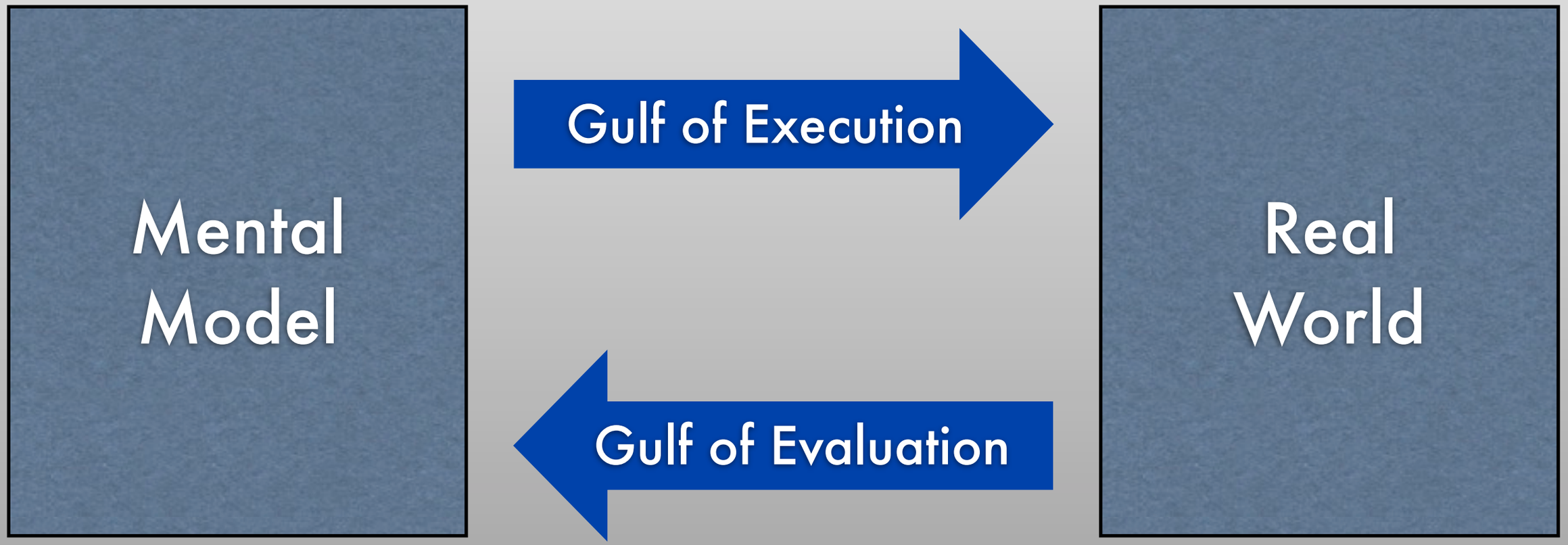
- What do screen-based interfaces afford?
- Screen, pointing device, physical buttons, keyboard
- These afford touching, pointing, clicking on every pixel

Affordances

Affordances

A little graphic design goes a long way

The Action Cycle



Interface Metaphors

- Designing around metaphors takes advantages of users' familiarity with real-life affordances (or other interfaces)
- Of course, if the metaphor is incomplete (or wrong), it can create an inaccurate conceptual model

Interface Metaphors

- Examples
  - Keynote/PowerPoint are like slide projectors
  - The Desktop metaphor

Problems with Metaphors

- Poorly-chosen metaphors might be:
  - Limiting: restrict interface possibilities
  - Too powerful: imply the system can do something it cannot
  - Mismatched: convey the wrong metaphor

Metaphor Guidelines

- A good metaphor
- Emphasizes the essential aspects
- Ignores/discards irrelevancies

Designing for (Human) Errors

Types of Errors

- Slips
  - "Slips result from automatic behavior, when subconscious actions that are intended to satisfy our goals get waylaid en route."
- Mistakes
  - "Mistakes result from conscious deliberations."

Types of Errors

- Not much can be done to prevent mistakes
  - Use design principles to make your interface easy to use
- Slips, by nature, are harder for users to detect (but can still be frustrating)
  - Norman identifies several kinds of slips

Modes

- Same action has different effect in different situations
- Examples
  - Caps Lock
  - Microsoft Word's "Overtype" toggle

Using Modes

Using Modes

- Temporarily restrict users' actions

Using Modes

- Temporarily restrict users' actions
  - Users must remember what mode they are in
  - Result: many errors

Alternative: Quasimodes

- Modes that require some conscious action to maintain
- Examples:
  - Shift key to capitalize (rather than Caps Lock)
  - Pull-down menus

Resources

- CS 160 Spring 2011 wiki:
  - http://husk.eecs.berkeley.edu/courses/cs160-sp11/index.php/Main_Page
- Don Norman's *The Design of Everyday Things*