

## Lec 4: Design Principles in HCI (Practical Industry Perspective)

---

### 1. The Pillars اعمدة of Design

In modern product design, especially in digital products (apps, websites, dashboards, etc.), the pillars provide a structured approach for creating intuitive, consistent, and user-friendly interfaces.

#### ◆ 1.1. Guidelines

- **Definition:**

Low-level recommendations or best practices for interface design.

Example: “Use consistent button shapes and colors for actions.”

- **Purpose:**

To ensure usability سهولة الاستخدام by following tested design conventions.

- **Industry Use:**

- Design systems like **Google’s Material Design** or **Apple’s Human Interface Guidelines** are examples.
- In Odoo ERP interfaces, guidelines define button positions, spacing, and font weights to ensure usability .

- **Example:**

- In a mobile banking app: the “Transfer” button is always in the bottom-right corner — following a guideline for task consistency.
- 

#### ◆ 1.2. Principles

- **Definition:**

Mid-level Strategies or rules for comparing and evaluating design alternatives.

They guide *why* you should design in a certain way — not just *how*.

- **Example:**

- When designing an e-commerce checkout flow, apply the principle of **error prevention** (Shneiderman) to disable the “Pay” button until all fields are validated.
- 

#### ◆ 1.3. Theories

- **Definition:**  
High-level frameworks explaining **why** users behave as they do when interacting with systems.
  - **Example:**
    - **Fitts's Law** informs why mobile touch targets (buttons) must be large enough and spaced reasonably.
    - **Norman's Action Model** helps UX designers identify where users may fail.
- 

## 2. Design Theories — Understanding the “Why”

These explain human capabilities (motor, cognitive) that directly impact usability.

### ◆ 2.1. Types of Theories

- **Descriptive:** Describe UI elements and how users perceive them.  
→ Example: A file “folder” icon visually maps to a real folder, helping mental models.
  - **Explanatory:** Describe *why* things happen (e.g., why users miss small icons).  
→ Example: “Users click the wrong button because both are blue and placed close together.”
  - **Prescriptive:** Suggest *what to do* to improve usability.  
→ Example: “Place confirmation buttons in the same location on all pages.”
  - **Predictive:** Allow performance estimation (e.g., how long it takes to click a target).  
→ Example: “A larger button reduces task completion time by 0.3s.”
- 

## 4. Norman’s Action Model (Theory)

- **Concept:** Describes how users form goals, act, and evaluate results.
- **Stages:**
  1. Forming a goal (“I want to save this file”)
  2. Intention to act (“I’ll click the save icon”)
  3. Execution (“I click the icon”)
  4. Perception (“The save icon grays out”)
  5. Interpretation (“It must be saved”)
  6. Evaluation (“now you **see “Report.docx” listed** among the other files in the folder—success”)

- **Design takeaway:**
    - Make **goals visible** → Use clear icons, labels (“Save,” “Download,” “Share”).
    - Provide **feedback** → Confirm success (“Saved ✓” toast message).
    - **Reduce the Gulf of Execution:** Users shouldn’t wonder how to do something.
    - **Reduce the Gulf of Evaluation:** Users shouldn’t guess what happened after doing it.
  - **Example:** Google Docs shows “Saving...” then “All changes saved” — perfect Norman feedback loop.
- 

## 5. Consistency Theories (Prescriptive)

- **Goal:** Reduce user learning time.
  - **Types of Consistency:**
    - **Visual consistency:** Fonts, colors, icons, button styles.  
*Example:* Apple’s system-wide rounded buttons.
    - **Functional consistency:** Same actions produce same results.  
*Example:* Swipe left = delete across all apps.
    - **Internal consistency:** Uniformity within the same product.  
*Example:* “Cancel” button always red; “Confirm” always blue.
    - **External consistency:** Aligns with familiar conventions across systems.  
*Example:* Shopping cart icons on all e-commerce sites.
  - **When to break consistency:** To draw attention (e.g., red “Delete Account” button in a gray form).
  - **Industry tip:** Consistency improves onboarding and reduces training costs.
- 

## 6. Human Processor Model (Predictive Theory)

- **Analogy:** Human brain works like a computer with input/output, memory, and processing.
- **Use:** Predict how long tasks take and how likely users are to remember things.
- **Applications in product design:**
  - Respect short-term memory: don’t make users recall previous steps.
- **Example:**
  - Auto-saved draft in Gmail reduces memory load (“Did I save it?” anxiety).

- A dashboard showing live data updates every 2 seconds to make the system responsive. **Why 2 seconds?**
  - According to HPM and **human attention limits**:
    - Updates faster than ~1 second might **overload perception** (too many changes to process).
    - Updates slower than ~10 seconds might feel **unresponsive** — the user's **cognitive** system starts to lose the sense of continuity.

- **Example:**

In an airline booking interface:

- Too many fields (departure, return, passenger info, seat selection) overload **working memory** → users make errors.
  - Solution: Break the form into **multi-step flows** with visible progress indicators.
- 

### ◆ 2.3. Memory in Design

- **Short-term memory (STM):**

- **Capacity:**  $7 \pm 2$  chunks.
- Rapid decay within seconds.
- **Design implication:** limit items per menu or list.
- **Example:** Keep main navigation menus under 7 items.

- **Long-term memory (LTM):**

- Holds learned patterns.
- Example: “Ctrl+S” always saves in productivity apps.

- **Practical takeaways:**

- Keep **instructions visible** (don't rely on memory).
  - Reuse **familiar icons** and labels to leverage learned behaviors from long-term memory.
- 

### ◆ 2.4. Fitts's Law

“The time to move to a target area is a function of the distance to and size of the target.”

- **Formula applied:**  
→ The larger and closer a button is, the faster users can click it.
  - **Industry Examples:**
    - **Touch UI:** Large “Add to Cart” buttons in shopping apps.
    - **Mobile:** Make primary buttons large and near the thumb zone.
    - **Web:** Keep “Submit” near where user’s eyes.
- 

### 3. Design Principles

Design principles translate theories into actionable, practical rules.

#### ◆ 3.1. Interaction Styles

Every interface falls under one (or a mix) of these:

1. **Direct Manipulation:**  
Dragging, dropping, resizing (e.g., Figma, Canva).
2. **Menu Selection:**  
Navigation via menus (e.g., Settings panels).
3. **Form Fill-in:**  
Input-heavy designs (e.g., online booking or tax filing).
4. **Command Language:**  
Terminal or CLI-based (e.g., developers in VS Code).
5. **Natural Language:**  
Chatbots, voice assistants, LLM-based UIs (e.g., ChatGPT).

Modern trend: **Hybrid interaction** — e.g., users can both click and speak in Google Assistant.

---

#### ◆ 3.2. Norman’s Principles of Good Design

Norman emphasizes the *cycle of action* — user forms a goal, executes it, and evaluates the outcome.

##### Key Industry Applications:

- **Visibility:**  
The degree to which **important parts** of the interface are **visible** — the user can easily see what actions are possible.

**In simple terms:** “Can I see what I can do here?”

→ Example: A **play button** (▶) on a music app — clearly visible and recognizable.

- **Good visibility** means the user doesn’t need to guess how to start or interact.

- **Good Conceptual Model:**

Align system behavior with user expectations.

→ Example: Dragging items into folders mimics real life.

- **Mappings:**

Controls match outcomes logically. **In simple terms:** “Do the controls behave as I expect?”

→ Example: Moving a **slider up** increases volume → intuitive mapping.

- **Good mapping** helps users predict what will happen before acting.

- **Feedback:**

Always show system response.

→ Example: “Uploading 80%...” progress bar.

---

## 4. Shneiderman’s 8 Golden Rules of Good Interface Design

### 1 Strive for Consistency

- **Definition:**

Similar tasks → similar design → reduces cognitive load.

- **Example:**

- Buttons always in the same position (e.g., “Save” bottom-right).
- Uniform icons across a system (e.g., gear = settings).

- **Inconsistency can be useful** only to draw attention (e.g., red “Delete Account” button).

---

### 2 Cater to Universal Usability

- **Definition:**

Design for diverse users (ages, disabilities, devices).

- **Examples:**

- Responsive layouts (desktop/mobile).
- Text-to-speech or dark mode for accessibility (e.g., blind, ...).
- Auto-adjusting font size for elderly users.

---

### 3 Offer Informative Feedback

- **Definition:**  
Every user action should produce an appropriate system response.
  - **Examples:**
    - Data saved → notification “Saved successfully.”
    - Payment failure → detailed error message.
- 

### 4 Design Dialogs to Yield Closure

- **Definition:**  
Every process should have a clear beginning, middle, and end.
  - **Example:**  
After completing an order → show “Order Confirmed” + “Track Order.”
  - **Why:**  
Helps users feel accomplished and ready for the next task.
- 

### 5 Prevent Errors

- **Definition:**  
Make it hard for users to make mistakes.
  - **Examples:**
    - Disable “Submit” button until form is complete.
    - Confirmation dialogs before deleting.
    - Highlight invalid input fields in red.
  - **Pro Tip:**  
Focus on *prevention*, not *correction*.
- 

### 6 Permit Easy Reversal of Actions (if error happened)

- **Definition:**  
Users should undo actions easily.
- **Examples:**

- Undo/Redo in Google Docs.
  - “Trash” folder before permanent deletion.
  - “Are you sure?” confirmations for destructive actions.
- **Impact:**  
مش خايف يغلط لانه يقدر يتراجع عن الخطأ  
Encourages exploration — users feel safe experimenting.
- 

## 7 Keep Users in Control

- **Definition:**  
The system shouldn't surprise the user. Avoid unexpected system behavior.
  - **Examples:**
    - No auto-refresh that loses typed data.
- 

## 8 Reduce Short-Term Memory Load

- **Definition:**  
Don't make users recall data between steps.
  - **Examples:**
    - Autofill previously entered data (addresses, skills from CV.pdf file).
- 

## 5. Design Principles by Benyon

### ◆ Learnability سهولة تعلم كيف تستخدم المنتج

- **Goal:** Make it easy for users to learn how to use the system.
  - **Key Practices:**
    - Familiar language (e.g., “cart” in e-commerce).
    - Predictable interactions (e.g., swipe left = delete everywhere).
- 

### ◆ Effectiveness

- **Goal:** Design the interface so that users **feel in control**, know **where they are**, **what they can do next**, and **how to undo mistakes** — without fear or confusion.
- **Key Practices:**

- Understand how to navigate (e.g., tabs).
  - Feel safe exploring (because they can always recover).
- 

## ◆ Safety & Security

- **Goal:** Protect users from errors and loss.
  - **Examples:**
    - Auto-save drafts.
    - Multi-step confirmation before destructive actions (e.g., Two-step “Delete account” flow).
- 

## ◆ Accommodation التكيف مع جميع أنواع المستخدمين

- **Goal:** Adapt to user diversity and context.
  - **Examples:**
    - “Dark mode” toggle.
    - Adjustable layout or text size.
    - Multilingual support icons.
- 

## ◆ Conviviality ودود

- **Goal:** Make the system friendly and polite.
  - **Examples:**
    - animations that create delight.
    - Respectful notifications — no intrusive مزعجة popups.
- 

## 8. Summary & Practical Insights

### ◆ Top Takeaways

1. **Consistency and feedback** form the foundation of usable design.
2. **Error prevention and recovery** directly reduce frustration and support trust.
3. **Memory load reduction** enhances flow and focus — crucial in mobile apps.

4. **Universal usability** widen your market and meet accessibility laws.

5. **Visibility + Mappings = Discoverability.**

---

### ◆ Common Design Mistakes

- Overuse of animations → cognitive overload.
- Inconsistent icons or color schemes.
- Hiding essential actions behind nested menus.
- Missing feedback after user actions.
- Not accounting for error recovery (e.g., irreversible delete).