

Lec 1: What is HCI?

- **HCI Definition:** Human-Computer Interaction (HCI) is the discipline focused on how humans interact with computers, systems, and technology.
 - Example: When you unlock your phone with Face ID, that's an HCI interaction.
 - **HCI Purpose:** It studies both the design of systems and the experience of users, ensuring technology is not only functional but also usable and enjoyable.
 - Example: Compare a complicated TV remote with 50 buttons vs. a simple Apple TV remote with just a few buttons—HCI explains why the simpler one improves usability.
 - Example: In ATMs, users must understand what button to press → the system must respond clearly with feedback.
-

Goals

- **Key Questions in HCI:**
 - What is HCI? → Understanding the field.
 - What is usability? → Ensuring systems are useful, easy to use, and attractive.
 - What are interaction design processes? → Steps to design user-friendly interfaces.
 - What is User-Centered Design (UCD)? → Putting users at the heart of design.
 - **Example:** When designing an online banking app, developers need to ask:
 - Is it easy to transfer money?
 - Is the language clear for non-technical users?
 - Are common mistakes (like sending to the wrong account) minimized?
 - **Example:** Designing medical software so that doctors can enter patient data quickly without confusion.
-

Human-Computer Interaction... In Brief

- **Multidisciplinary Nature:** HCI combines psychology, computer science, ergonomics, sociology, business, and design.
 - Example: Designing VR headsets requires psychology (how humans perceive 3D), ergonomics (comfort), computer science (software), and business (market adoption).
- **Dynamic Interaction:** Both the human and the computer influence each other over time.

- Example: Auto-correct adapts to the user's typing style, while the user adapts to how auto-correct behaves.
 - *Example:* A recommendation system (Netflix/YouTube) adapts based on what the user watches → user behavior changes because of recommendations.
-

HCI Is Multidisciplinary

- **Psychology & Cognitive Science:** Understanding how users perceive, think, and solve problems.
 - Example: Websites use color psychology (red for urgency, green for success).
 - **Ergonomics:** Designing for physical comfort and safety.
 - Example: Split keyboards reduce wrist strain for programmers.
 - **Sociology:** Considering cultural and social context.
 - Example: WhatsApp added voice notes because many cultures prefer speaking over typing.
 - **Computer Science/Engineering:** Build the actual hardware and software.
 - Example: Apple engineers coding iOS to run smoothly.
 - **Business:** Products must satisfy market needs.
 - Example: Food delivery apps like Talabat target the demand for quick meals.
 - **Graphic Design:** Making interfaces visually clear and attractive.
 - Example: Spotify's dark mode improves readability and reduces eye strain.
 - **Technical Writing:** Clear documentation and tutorials.
 - Example: Google Docs tooltips that explain features in one line.
-

The Goal of HCI

- **Focus on Three Ingredients:**
 - The User (needs, goals).
 - The Computer (hardware, software, interfaces).
 - The Task (the work to be completed).
- **Usability Requirements:**
 - Useful (solves a real problem).
 - Usable (easy to operate).

- Used (people actually adopt it).
 - **Example:** Zoom's success during COVID-19 → It was useful (remote meetings), usable (simple interface), and widely used (millions adopted it quickly).
-

The Ingredients

- **Human Factors:**
 - Sensory systems (visual, audio, touch).
 - Example: Blind users rely on screen readers (audio feedback).
 - Acting systems (hands, voice, body).
 - Example: Voice assistants like Alexa rely on speech.
 - Cognitive processes (memory, perception).
 - Example: Icons help memory recall—trash bin icon = delete.
 - **Computer Side:**
 - **Input devices:** keyboard, mouse, touch screen, sensors.
 - Example: Facial recognition sensors in smartphones.
 - **Output devices:** screens, audio, VR.
 - Example: Apple Watch giving vibration alerts.
-

What Is Interaction (in HCI)?

- **Interaction is not just “talking to a computer.”**
 - It's a process where the user and computer influence each other's behavior over time.
 - **Types of influence:**
 - Statistical (system tracks user patterns, e.g., Netflix recommendations).
 - Mechanical (system enforces strict rules, e.g., ATM won't dispense cash without correct PIN).
 - Structural (system changes user habits, e.g., TikTok reshaping how people consume video).
 - **Users' goals are the final metric.**
 - System success is measured by whether the user achieves their goal.
 - Example: A ride-hailing app is successful only if it actually gets the user from point A to point B.
-

Assumptions

- **Users operate in a domain with specific goals.**
 - Example: In medical software, “patient record” and “diagnosis” have precise meanings that doctors expect.
 - **Tasks are the operations that achieve goals.**
 - Example: Goal → “book a flight.” Task → choose date, select seat, enter payment.
 - **Interaction is bridging the two languages:**
 - User speaks in “task language” (“I want to transfer money”).
 - System speaks in “core language” (database updates, transaction logs).
-

Norman’s Model of Interaction

- **Seven Stages of Action:**
 1. Establishing the goal → WHAT to do.
 - Example: “I want to send an email.”
 2. Forming the intention → HOW to do it.
 - Example: “I’ll open Gmail.”
 3. Specifying the action sequence → step-by-step plan.
 - Example: “Click Compose → type recipient → write message → hit send.”
 4. Executing the action → physically doing it.
 - Example: Typing the message and pressing send.
 5. Perceiving system state → checking feedback.
 - Example: Gmail shows “Message Sent.”
 6. Interpreting system state → understanding feedback.
 - Example: Understanding that the email has reached the recipient’s server.
 7. Evaluating outcome → judging success.
 - Example: Checking if recipient replies or email bounces.
- **Two Gaps in Interaction:**
 - Gulf of Execution → difficulty translating goals into actions.
 - Example: A poorly designed photo editor makes it hard to find the “crop” tool.

- Gulf of Evaluation → difficulty understanding feedback.
 - Example: Error messages like “404” confuse users instead of explaining clearly.
-

Human Errors in Interaction

- **Slips:** Correct intention but wrong execution.
 - *Example:* Clicking “Delete” instead of “Save.”
- **Mistakes:** Wrong intention due to bad understanding.
 - *Example:* New user thinks “Zoom” button = magnify but actually starts a Zoom meeting.
- **Design must reduce error probability:**
 - Better spacing of buttons (prevents slip).
 - Clear labels and confirmations (prevents mistakes).
- **Norman’s View:** Errors are not the user’s fault → they are a symptom of poor design.
 - Example: If many users press the wrong elevator button, it’s a design flaw, not human fault.
- **System should compensate:**
 - Anticipate mistakes.
 - Allow undo & error recovery.
 - *Example:* Gmail “Undo Send” feature → allows correction.

Frameworks: Major UI Styles

- **Common UI styles:**
 - Command Line (efficient for experts).
 - Example: Linux terminal.
 - Menus (easy to navigate).
 - Example: Drop-down menus in Microsoft Word.
 - Natural Language Interfaces.
 - Example: Siri, Alexa.
 - WIMP (Windows, Icons, Menus, Pointer).

- Example: Desktop OS interfaces.
 - Mobile Interfaces.
 - Example: Swipe to close apps android
 - 3D Interfaces.
 - Example: VR design tools like “Tilt Brush”.
-

Design Processes and Frameworks

- **Purpose:** HCI provides approaches for shaping the design process.
 - **Why it matters:**
 - Avoid software project failure. **50% of projects fail due to poor developer-user communication.**
 - A framework ensures alignment and avoids “building the wrong system.”
 - **Example:** A healthcare app that looks beautiful but doesn’t meet hospital regulations → failure due to ignoring user (doctor, patient) needs.
-

User-Centered Design (UCD)

- **User at the center:** Users’ needs/wants considered from the start.
 - **Benefits:**
 - Easier to learn.
 - Fewer errors.
 - Higher user satisfaction.
 - *Example:* Banking apps tested with actual customers before launch.
 - **Challenges:**
 - Finding representative users.
 - Motivating them to participate.
 - Extracting needs in their own language.
 - **Industry Example:** Microsoft runs usability studies with thousands of users before releasing Windows updates.
-

Agile Interaction Design

- **Principle:** Iterative, incremental design inspired by Agile software development.
 - **Key Features:**
 - Rapid prototyping.
 - Continuous feedback.
 - Low-cost, fast usability inspections.
 - **Example:** Tesla pushes frequent over-the-air updates → users immediately test changes, and feedback loops inform future iterations.
-

Design Thinking

- **5 Stages (non-linear, iterative):**
 1. **Empathize** – Understand user needs (e.g., interview commuters about pain points).
 2. **Define** – Frame user problems (e.g., “People struggle with real-time bus updates”).
 3. **Ideate** – Brainstorm creative solutions (e.g., live GPS bus tracking app).
 4. **Prototype** – Build quick models (e.g., mockup نموذج of the app interface).
 5. **Test** – Evaluate with users and refine.
 - **Example:** Airbnb used Design Thinking workshops to redesign their app → improved booking flow and user trust.
-

Human-Centered Design Process

- **Main Steps:**
 - **Needfinding** – Discover user needs.
 - Example: Observe how elderly people struggle with tiny fonts in apps.
 - **Analysis** – Structure needs, create scenarios/tasks.
 - Example: Compare “current bus schedules” vs. “desired real-time GPS updates.”
 - **Design** – Apply rules, guidelines, principles.
 - Example: Use larger fonts, voice interfaces for elderly users.
 - **Iteration & Prototyping** – Validate ideas with mockups.
 - Example: A clickable Figma prototype tested by 10 users before coding.
 - **Implementation & Deployment** – Build final system and documentation.

- Example: Final release of WhatsApp voice messaging after beta testing.
 - **Key Point:** Iteration is central → prototypes save time & money.
 - It's cheaper to adjust a sketch or mockup multiple times than to redesign a finished product after launch.
-

Usability

- **ISO Definition:** The extent to which specified users achieve goals effectively, efficiently, and with satisfaction in a given context.
 - **Usability Dimensions:**
 - Usefulness → Does it solve a real need? (Google Maps replacing paper maps).
 - Learnability → Easy to learn? (Instagram's intuitive posting flow).
 - Memorability → Easy to recall? (Once you know CTRL+C, you remember forever).
 - Effectiveness → achieves goals? (ATM cash withdrawal).
 - Efficiency → Quick once learned (*Uber 1-click reordering*).
 - Visibility → System state clear? (Progress bars in downloads).
 - Error Handling → Few, recoverable errors. (Undo button in MS Word).
 - Satisfaction → Enjoyable to use. (Apple design aesthetics).
 - **Golden Rule:** “*Don't Make Me Think*” → users should navigate naturally, without effort.
-

HCI in the Software Process

- **When to apply UCD in software development:**
 - Before every design step.
 - Before every implementation step.
 - At every iteration/sprint.
- **Why it's cost-effective:**
 - Cheaper to test prototypes than full products.
 - Early user testing avoids costly redesigns later.
- **Non-functional requirements (NFRs):** Usability, safety, performance → must be part of system requirements.
- **Example:**

- NASA runs usability simulations before launching space mission software → impossible to fix after deployment.
- In startups, mockups and A/B testing reduce the risk of building irrelevant features.