

## Lec 2: Needfinding in Human-Computer Interaction

### Needfinding

- **Definition:**
    - *Needfinding* means discovering what users really need, not just what they say they want.
    - It is the foundation of **user-centered design** — creating solutions that fit real people's lives.
  - **Why it matters:**
    - Many products fail because designers build what they ***think*** users need, not what users actually need.
    - Example: Early smartwatches focused heavily on mini-apps, but users mainly wanted **notifications, health tracking, and long battery life**.
  - **Analogy:**
    - Think of a doctor diagnosing a patient. The patient may complain of “headache,” but the doctor must dig deeper to uncover the *underlying cause*.
    - Similarly, needfinding uncovers the *true pain points*.
  - **Industry case:**
    - **Netflix** didn't just ask people what they wanted. Users said they wanted more movies, but deeper research revealed the *need was quick and personalized entertainment*. This insight led to recommendation algorithms.
- 

### Main Needfinding Questions

- **What do users need?**
  - Goes beyond surface-level desires (what they say they want).
  - Example: A student might say they want a “better learning app,” but the real need may be **study reminders, offline access, and adaptive quizzes**.
- **Who are the users?**
  - Must define specific groups, not “generic users.”
  - Example: “Bank customers” is too broad. Needs differ between **retired seniors using branches, young professionals using mobile apps, and business clients needing corporate dashboards**.
- **What is the context?**

- Needs change by context: time, place, culture.
  - Example: Mobile banking in rural Africa differs from New York City — limited internet → **SMS-based banking (M-Pesa App)**.
  - **Why not just ask them?**
    - People often cannot say needs.
    - Example: Users didn't ask for the iPhone; they asked for "better phones with keyboards." Apple observed frustrations with buttons and revealed the need for **touchscreen simplicity**.
- 

## **Know Your Users**

- **Users come in categories**
    - Young vs old, junior vs expert, daily vs occasional.
    - Example: Photoshop is great for experts but overwhelming for normal users → Adobe later released **Photoshop Express**.
  - **Designers/developers are not the user**
    - Designers/developers think differently.
    - Example: Engineers at Microsoft initially overcomplicated early versions of Word; actual office workers just needed **basic formatting and spell check**.
  - **The client/manager is not the user**
    - Managers may think they know workers, but they see *tasks from above*, not daily struggles.
    - Example: A hospital IT manager may claim nurses just need a "better database," but nurses may actually need **fast chart access at patients' bedside**.
- 

## **Talking vs Watching Users**

- **Talking to users:**
  - Surveys, interviews.
  - Example: Asking employees what frustrates them about software.
  - *Risk:* They may answer what they think you want to hear → "App is *OK*? App is very good." بياخدوك على اد عقلك
- **Watching users:**

- Observation, diaries, work analysis.
  - Example: Seeing how teachers actually use attendance apps while teaching reveals **real workflow friction**.
  - Advantage: Captures *real behavior*, not filtered answers.
- 

## **Imagining Users (Personas)**

- **Personas** = fictional characters.
  - Example:
    - “Ahmed, 34, small business owner, tech-savvy, values quick payments, hates paperwork.”
    - This helps designers ask: “Would Ahmed find this feature useful?”
  - **Real-world use:**
    - Airlines design apps with personas like “Frequent flyer businesswoman” vs “Occasional Vacation traveler.” Each has different needs (speed vs price).
- 

## **Problem Framing** تحديد المشكلة الحقيقة

- **Why framing matters:**
    - Defining the problem wrong → wrong solution.
    - **Example:**
      - Problem framed as “*students forget deadlines*” → build a reminder app.
      - But deeper framing: “*students feel overwhelmed by too many platforms*” → build a single integrated course dashboard.
    - **Example:**
      - A company says “Our employees need more training.”
      - Framing may reveal the *real problem* = “The system is confusing → people make mistakes, so they look untrained.”
  - **Best practice:** Reframe until the **underlying cause** is clear.
- 

## **Needfinding Methods**

- **Observation:** Watch real use. Example: Studying how doctors enter patient info.
- **Dairies:** ask users to record tasks/problems daily. Example: Students record daily study habits → shows delay تسويف patterns.

- **Interviews:** ask structured/unstructured questions for deep exploration. Example: Gamers describing frustrations with unfair matchmaking توفيق بين اللاعبين .
- **Focus groups:** gather group opinions through shared discussion. Example: Parents group discussing e-learning platforms.
- **Surveys:** scale to large population, shallow detail. Example: 10,000 users rate satisfaction with food delivery.
- **Contextual Inquiry:** Observe + interview simultaneously. Example: Watching chefs cook while asking why they use certain tools.

**Best practice:** Combine methods to balance **breadth (surveys)** and **depth (observations/interviews)**.

---

## Observation

- **Observation**
  - Embed in user's environment.
  - Example: Airbnb founders lived with hosts to understand hosting challenges.
- **What to learn:**
  - Reality vs official processes.
  - Example: Official hospital process = Nurses should enter patient vitals directly into the **electronic medical record (EMR)**.
  - Reality = Because the system is slow or inconvenient (auto-logout, poor usability), nurses fall back on **handwritten notes** → then double work later.
- **Controlled observation (lab) vs Naturalistic (real-world):**
  - Controlled (lab) = Easier to repeat but artificial (e.g., testing an ATM in a controlled room).
  - Natural: More realistic insights (e.g., testing an ATM at a busy street corner).
- **Blending in:**
  - *Complete Observer* (watch silently) vs *Complete Participant* (join activity).
  - Example: A UX researcher/designer “worked” as a barista جرسون for weeks to understand coffee shop workflow.

---

## Interviewing

- **Structured Interviews**

- **Definition:** A set of pre-prepared questions asked in the same order to all participants.
- **Example:**
  - Asking 100 employees: “On a scale of 1–5, how often do you use the HR system?”
  - Great for comparison but might miss the **why** behind low usage.

- **Unstructured Interviews**

- **Definition:** A free conversation, open-ended questions, allowing users to guide the discussion.
- **Example:**
  - Sitting with a nurse and asking: “Tell me about your day at the hospital.”
  - The nurse may reveal problems you didn’t anticipate — e.g., problem of handwritten notes and electronic records because EMR system is slow.

- **Focus Groups**

- **Definition:** A discussion with 5–10 participants to explore opinions, and conflicts.
- **Example:**
  - A focus group on food delivery apps:
    - Some users say they want “fast delivery.”
    - Others complain “cheaper prices matter more.”
    - This conflict shows design trade-offs → should the product emphasize speed, price?

- **Challenges:**

- **Users don't always know what they want.**
- Example: Before Uber, riders said “I want safer taxis,” not “I want an app to book rides instantly.”
- Example: iPhone’s touchscreen keyboard → people said they preferred physical keys, but once experienced touchscreen keyboard, they shifted.

- **Choosing participants:**

- Recruit representative users, not just suitable ones.
- Example: Testing a food delivery app only with college students may miss insights from busy parents or office workers.

- **Good questions:**
    - Unstructured/open-ended Questions are preferred.
    - Example: “Tell me about a typical day using this software.”
    - Example: “What’s the most frustrating part of your current workflow?”
    - Example: “Tell me about the last time you had trouble booking travel.”
  - **Bad questions:**
    - Leading: “Do you like our new feature?”
    - Hypotheticals افتراضيات: “What would you do if...” → unreliable answers.
- 

## **Surveys**

- **Pros:**
    - Cheap, scalable, easy to analyze.
    - Example: Google Forms surveys about online learning.
    - Example: Netflix surveying 10,000+ users about recommended movies.
  - **Cons:** Shallow, no follow-up questions.
    - Example: A survey might say “80% of users are satisfied,” but doesn’t explain why.
  - **Risks:**
    - Inaccurate data (skip questions).
    - Example: People underreport لا يقولون الحقيقة how often they use social media.
  - **Example of (Likert) scale preferred in surveys:**
    - Scale: 1 (Strongly Disagree) → 5 (Strongly Agree) for statements like:
      - “This app is easy to use?”
  - **Warnings:**
    - Avoid complex or negative wording.
    - Example: Instead of “How much do you *not* dislike...?” كم لا تكره , ask simple positive questions.
- 

## **الاستفسار السياقي**

- **Definition:** Observe while interviewing.
    - Researcher/designer acts like a learner, user like a master craftsman **حرفي ماهر**.
  - **Why it's powerful:**
    - People can't recall all steps in interviews. But when *doing tasks live*, they explain naturally. → Contextual = real-time, deeper insights.
    - Example: Watching a doctor prescribe medicine → researcher/designer sees frustrations with software.
  - **Vs Observations:**
    - Observations = silent watching.
    - Contextual = interactive, asking “why” questions.
  - **Risks:**
    - User changes behavior because they feel “watched.”
  - **Best practices:**
    - Keep an open mind, validate findings with participants.
    - Example: After shadowing warehouse workers, confirm: “So you **tape boxes first then attach labels later because the printing labels is slow** — is that correct?”
    - This looks unusual because the “official” process is: print label → stick it → then tape.
- 

## Summary:

- Needfinding is about **discovering real needs (pain points), not just what users say.**
- Techniques include **observation, interviews, surveys, diaries, contextual inquiry.**
- Real-world product success = solving the *real*, often **unspoken**, user need.
- **Don't trust only what managers say** — validate with real users (workers).
- **Frame problems correctly to design the right solution** — misframing leads to wrong solutions.
- Combine methods for **comprehensive understanding.**