Designing Interactive Systems I

Evaluation

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

Winter Semester ’24/’25

https://hci.rwth-aachen.de/dis

RWTH AACHEN UNIVERSITY

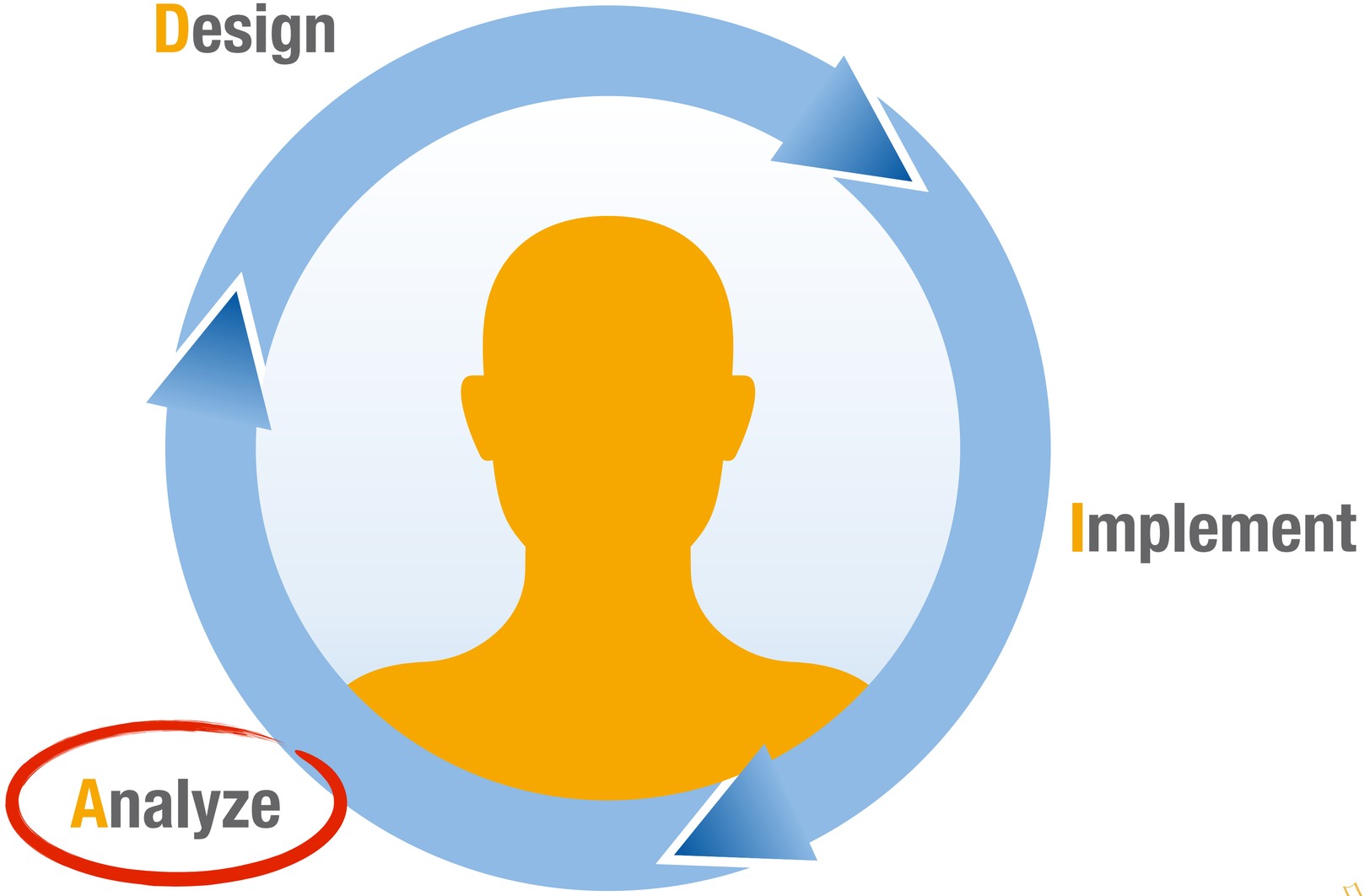
Review

- Types of paper prototypes
- Types of software prototypes?
  - Horizontal/vertical software prototypes?
  - Tools?
- When should you use a hardware prototype?
  - Wizard of Oz?

Evaluation

- When, why, where, and what?
- Concrete methods to evaluate designs and implementations
- How to deal with users

When to Evaluate



When to Evaluate

- Evaluation should happen throughout the entire software development process
  - Early designs are more often evaluated by the design team, analytically and informally
  - Later implementations are more often evaluated by users, experimentally and formally

Why Evaluate?

- To ensure that system matches user needs
  - Necessary even if design was already user-centered (interviews, …)!
- To judge system features
  - Does it facilitate users' tasks?
  - Does it offer the right features, easy to reach, and presented as expected?
- To judge effects on users
  - How easy is it to learn and use the system?
  - How do users feel about the system?
  - Are there areas that overload users?
- To discover specific problems
  - Do unexpected/confusing situations come up?

Where to Evaluate: Lab

+ Equipment (A/V, see-through mirrors, special computers)
+ No disruptions
+ Quiet
– Natural environment missing (shelves, wall calendar, …)
– Unnatural situation (relevance?)

- Preferable if
  - the real location is dangerous
  - remote (ISS), or
  - a controlled situation is needed

Where to Evaluate: In The Field

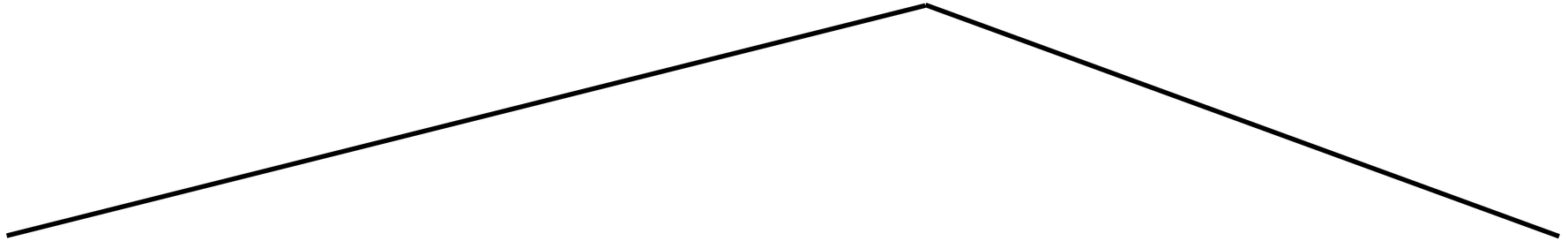
- Studies in the users' natural environment

+ More realistic (also *because* of disruptions)
+ Situations and behavior more natural
+ Better suited to long-term studies
– Noise, task interruptions
– Will still feel like a test situation

Participatory Design

- Involve users as part of design team throughout entire software process
- Originated in Scandinavia where it is the law for certain products
- Techniques for team communication
  - Brainstorming, storyboarding, workshops, interviews, role plays, paper prototypes
- Problems
  - High effort, conflicts with client hierarchies, user conversion

Evaluation Techniques



Evaluating Without Users

E1 Literature Review

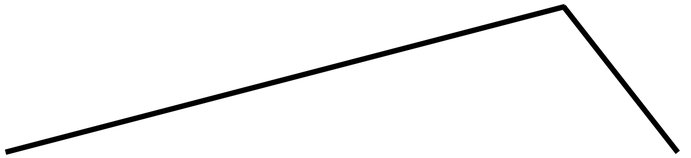
E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation

- GOMS, HCI Design Patterns, …

Evaluating With Users



Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

E1: Literature Review

- Many research results about user interface design have been published
- Idea: Search literature for evidence for (or against) aspects of your design
- Saves own experiments
- Results only carry over reliably if the context (users, assumptions) is very similar

E2: Cognitive Walkthrough

- Goal: Judge learnability and ease of use — without users
- Analytical method for early design or existing systems
- Requires an HCI expert (designer, cognitive psychologist), interface description, task description, user profile, and context description; takes time
- For each task, derive goal—intention—action sequence, and ask
  - Does system help the user to get from goals to intentions and actions?
  - What knowledge and cognitive processes will the user need for this decision process?
  - What problems could learning/doing this step have?
- Question forms can capture psychological knowledge to guide the user

E3: Heuristic Evaluation

- Variant of Cognitive Walkthrough
- Choose usability heuristics
  - General guidelines, e.g., Ten Golden Rules
- Step through tasks and check whether guidelines are followed
  - \+ Quick and cheap
  - – Subjective
    - Better done by several independent designers

E4: Model-based Evaluation

- Some models exist that offer a framework for design and evaluation

- Examples:
  - GOMS, KLM
  - Information efficiency

  } next lecture

  - Design Rationale (History of design decisions with reasons and alternatives)
  - HCI Design Patterns

16 social protocol ***

Figure 17: Passing on a mouse for a group display.

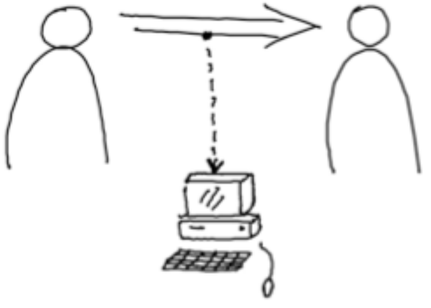
…you have picked your hardware to control the room and its services—ROOM CONTROLLER (15), and now need to decide how the technology is operated by the users.

♦♦♦

Interactive technology likes to be told when something happens or when it is supposed to do something. But people easily forget that extra step, especially when in the middle of a high-energy brainstorming session.

A research video by MIT once showed a group of researchers having a meeting around the table, and the room was "listening in" on the conversation going on. Whenever a certain point was reached, such as deciding to add a new item to the agenda, or delegating a task to a member in the room, everybody had to shut up, and the moderator would speak the corresponding commands for the computer to keep up with what was going on. It was the worst group support interface imaginable.
Good group support software follows what's going on in the room as good as it can, trying to detect from a variety of sensors, models, and other input what the current activity and actors are, and then takes initiative on a simple, reliable level to help the actors, without presuming to understand more than it can.
Computer scientists will argue that deriving this information from sensor values is not reliable, so the computer needs clear commands in order not to do something wrong. This is perfectly true in distributed settings with low bandwidth for human communication: If user A decides to pass control over the shared mouse cursor to remote user B in a shared application, he usually has to click a button to do so.
In a collocated setting of an AE, an enormous advantage comes to the help of the system: social protocol. The people in the room can see and hear each other. If one person is controlling the mouse cursor using their laptop, and someone else wants to take over with their own laptop, they will just say so. The computer does not need to understand this verbal command, nor does he need to lock the cursor for everybody else but one user at a time: It can simply accept cursor movement from everybody in the room; if there's a conflict of concurrent access, the users will quickly and easily notice and resolve it among themselves. This approach, on the other hand, saves the users having to send explicit messages each time they wish to pass control of that cursor to someone else, making the interaction much more fluid.
Examples include the design of the interaction for the iRoom's remote cursor control that allows "mouse fights" to occur, simply always using the last coordinate received; or its iClipboard feature that lets people cut and paste in a single shared clipboard for the room.
Winograd et al., in their chapter elsewhere in this book, reflect on this concept by suggesting room infrastructure in which "...users and social conventions in an environment take responsibility for actions, and the system infrastructure is responsible for providing a fluid means to execute those actions."
Therefore:

Do not put unnecessary protocols into place that are aimed at avoiding overlapping access to technology, if that collision can be easily noticed and fixed by the users through social interaction. If a user issues a social protocol act, such as passing a wireless mouse to someone else, never require an additional repetitive step from the user to tell the room what he just did for everyone else to clearly see.



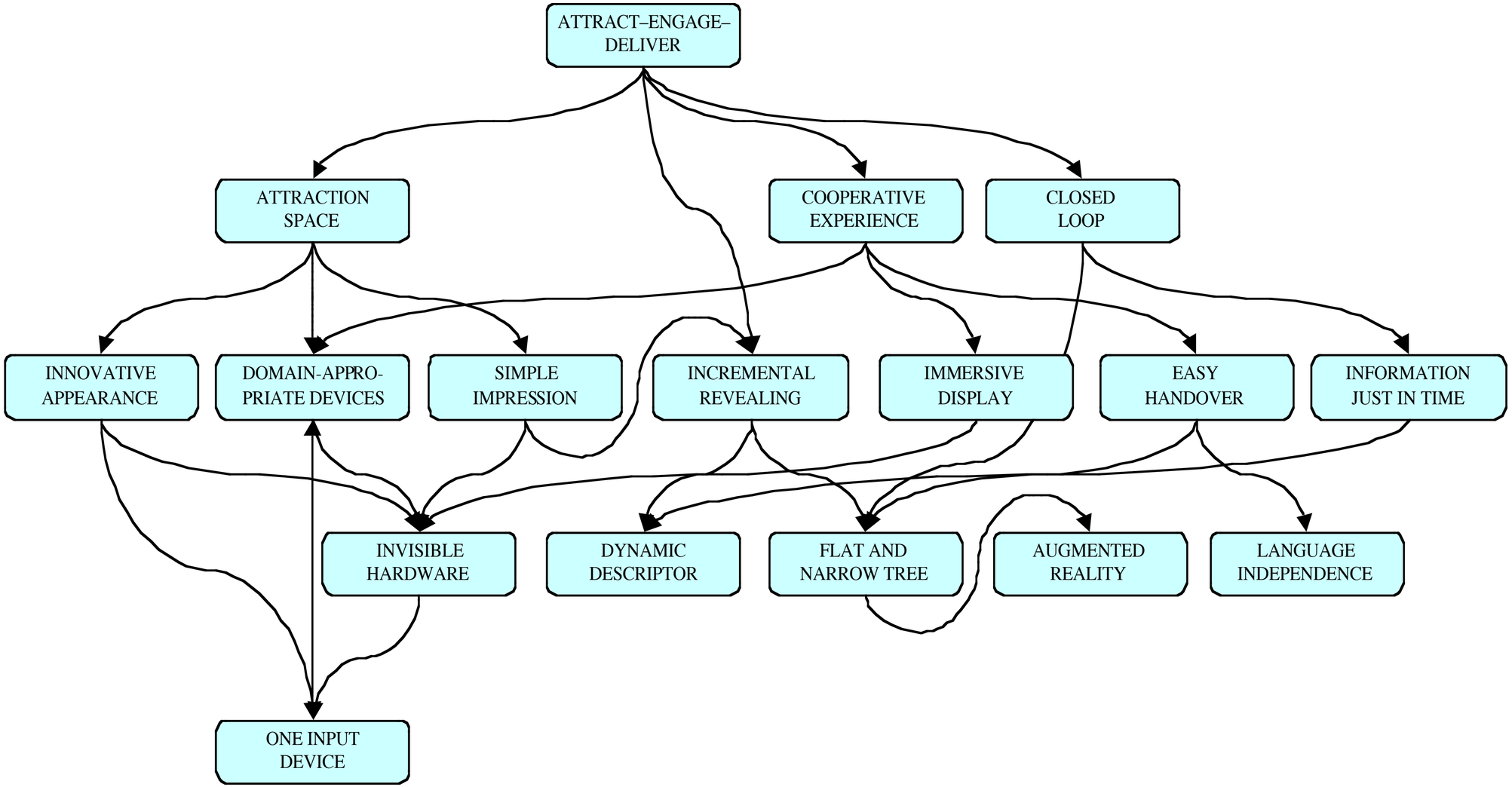
♦♦♦

This is a basic pattern with no further references within this language.

Pattern Languages in HCI

- Early references
  - Norman & Draper (1986): User-Centered System Design
    - Earlier than in SW-Eng!
  - Norman (1988): The Psychology (Design) of Everyday Things
    - "Fascinating to skim, frustrating to read" :)
  - Apple Macintosh Human Interface Guidelines (1992)
    - "seminal in the field of environmental design"

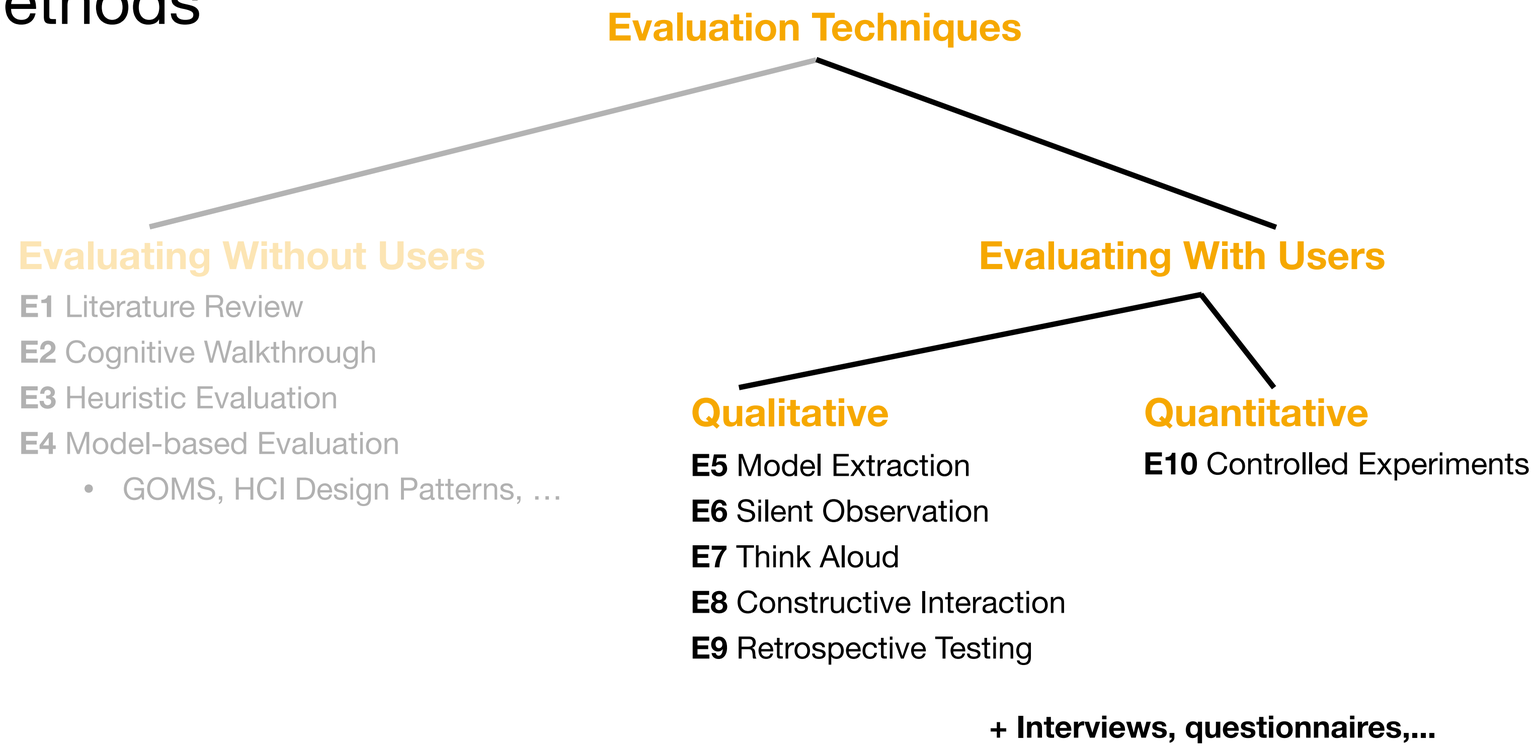
Interactive Exhibits: A Pattern Language



Evaluating with Users

- E1–E4 evaluate designs without the user

- As soon as implementations (prototypes) exist they should also be tested with users, using the following methods



E5: Model Extraction

- Designer shows a prototype or screen shots to the user
- The user tries to explain elements and their function

\+ Good to understand naïve user's conceptual model of the system

– Bad to understand how the system is learned over time

E6: Silent Observation

- Designer watches the user working on one of the tasks in a lab or natural environment

- No communication during observation

+ Helps to discover big problems

– No understanding of the decision process (that leads to problems) or user's mental model, opinions, or feelings

by Saul Greenberg

E7: Think Aloud

- As E6, but the user is asked to say aloud:
  - What she thinks is happening (state)
  - What she is trying to achieve (goals)
  - Why she is doing something specific (actions)
- Most common method in industry

+ Good to get some insight into user's thinking, but:
  - Talking is hard while focusing on a task
  - Feels weird for most users to talk aloud
  - Conscious talking can change behavior

by Saul Greenberg

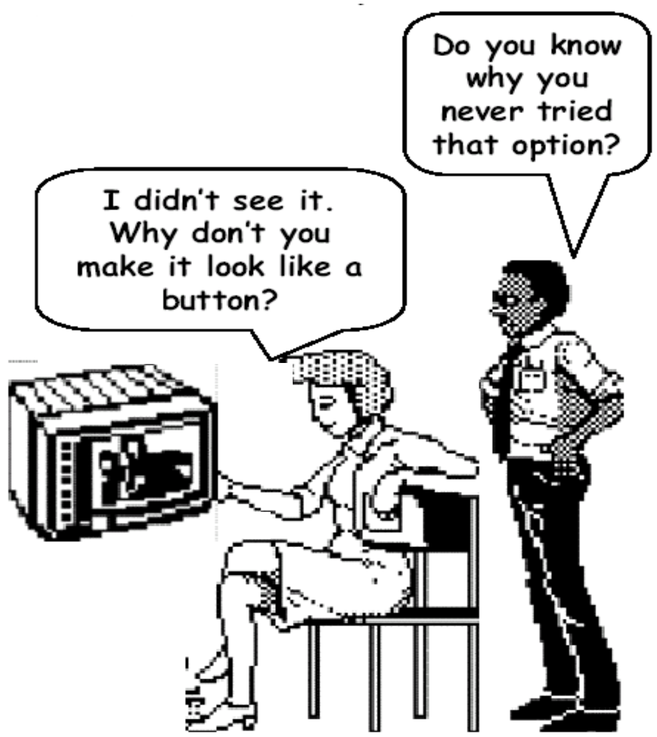
E8: Constructive Interaction

- Two people work on a task together
  - Normal conversation is observed (and recorded)
  - More comfortable than Think Aloud
- Variant of this: Different partners
  - Semi-expert as “trainer”,
    newbie as “student”
  - Student uses UI and asks, trainer answers
  - Good: Gives insight into mental models of beginner and advanced users at the same time!

by Saul Greenberg

E9: Retrospective Testing

- Additional activity after an observation
- Subject and evaluator look at video recordings together, user comments his actions retrospectively
- Good starting point for subsequent interview, avoids wrong memories
- Often results in concrete suggestions for improvement



by Saul Greenberg

Recording Observations

- Paper + pencil
  - Evaluator notes events, interpretations, other observations
  - Cheap but hard with many details (writing is slow). Forms can help.
- Audio recording
  - Good for speech with Think Aloud and Constructive Interaction
  - But hard to connect to interface state
- Video
  - Ideal: two cameras (user + screen) in one picture
    - Or use screen recording + user camera (synchronization!)
  - Best capture, but may be too intrusive initially
  - Some dedicated tools for analysis, e.g., MAXQDA (for labeling)

Evaluation Techniques

Evaluating Without Users

E1 Literature Review

E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation

- GOMS, HCI Design Patterns, …

Evaluating With Users

Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

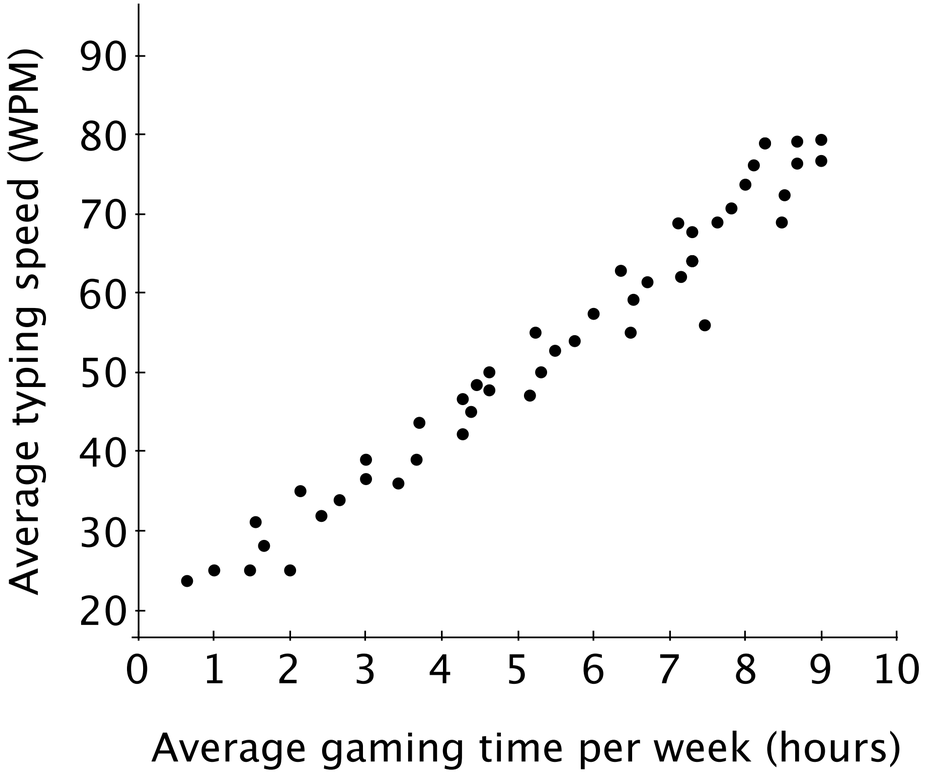
Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

E10: Controlled Experiments

- Quantitative, empirical method
- Used to identify the cause of a situation or set of events
  - “X is responsible for Y”
  - Directly manipulate and control variables
- Correlation does not imply causality
  - Example: relationship between typing speed and time spent playing games
- Use a controlled experiment to verify an observation, a correlation, or a “hunch”



E10: Controlled Experiments

A. Formulate hypothesis
B. Design experiment, pick variable and fixed parameters
C. Run pilot study
D. Choose and recruit subjects
E. Run experiment
F. Interpret results to accept or reject hypothesis

Controlled Experiments: Steps

A. Formulate hypothesis
    - Selecting menu items is faster with pie menus than with linear menus

B. Design experiment, pick variables and fixed parameters
    - Type of menu ⇒ target seek time

C. Run a pilot study to debug your experimental procedures
    - Improving distribution of menu targets

The examples are simplified from [Callahan et al., CHI'88]

Controlled Experiments: Steps

D. Recruit subjects

   - Undergraduate students with consistent experience
     (back in ’88: “little or no mouse experience”)

E. Run experiment

   - Each participant performs a menu selection (10 times for each type of menu)

F. Interpret results to accept or reject hypothesis

   a. Mean seek time: 2.26s (Pie), 2.64s (Linear)

   b. The difference is statistically significant (p =.0003)

The examples are simplified from [Callahan et al., CHI’88]

Controlled Experiments

- Subjects
  - Similar to real users in profile (age, education, computer and domain expertise, system knowledge, …)
  - Use at least 10 subjects
    - Use more if you need finer details
    - Statistical power analysis can tell you the exact number
- Variables
  - **Independent Variables (IVs)**: are varied under your control
    - E.g., number of menu entries
    - Each level of an independent variable is called a treatment
  - **Dependent Variables (DVs)**: are those you measure
    - E.g., execution time, error rates, subjective preferences

In-Class Exercise: Identifying Variables

Identify independent variables and dependent variables from each of the following scenarios. Indicate levels of each independent variable:

A. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training

B. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets

C. A research team examining whether virtual teams who use video chats are more productive than teams who use text-only chats

Hypothesis

- Predicts outcome of experiment
  - Usually: claims that changing independent variables influences dependent variables
- Experiment goal: confirm **research hypothesis (H_1)**
  - “No amount of experimentation can ever prove me right; a single experiment can prove me wrong.” —Albert Einstein
- Approach: Reject inverse **null hypothesis (H_0)**, i.e., “no influence”
  - If we can determine that H_0 is wrong, we can accept that H_1 is true (naïve view)
  - H_0 is usually a precise statement ⇒ we’ll know the probability that H_0 is incorrect
    - E.g., “Average WPM between gaming and non-gaming groups are equal”
  - The data should indicate that there is a very low probability that H_0 is correct
- Being unable to reject H_0 **does not imply** that you can accept **H_0**
  - E.g., your number of participants may just have been too small

In-Class Exercise: Identifying Hypotheses

Identify a **research hypothesis (H_1)** and **null hypothesis (H_0)** for each of our scenarios:

A. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training

B. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets

C. A research team examining whether virtual teams who use video chats are more productive than teams who use text-only chats

Basic Experimental Designs

- **Between-groups design**
  - Each subject only does one variant of the experiment
  - There are at least 2 groups to isolate effect of manipulation:
    - Treatment group and control group
  + No learning effects across variants
    - Good for tasks that are simple and involve limited cognitive processes, e.g., tapping, dragging, or visual search
  – But: requires more users
- **Within-groups design**
  - Each subject does all variants of the experiment
  + Fewer users required, individual differences canceled out
    - Good for complex tasks, e.g., typing ,reading, composition, problem solving
  – But: learning effects may occur

Experimental Designs

Which type of experimental design is appropriate for each scenario?

A. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training

B. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets

C. A research team examining whether virtual teams who use video chats are more productive than teams who use text-only chats

Examples from: Research Methods in HCI, Lazar et al. (2010)

Within-Group Design: Order Effect

- The order of presenting the treatments (IV levels) might affect the dependent variable
  - Learning effect
  - Fatigue effect
  - Contrast effect: the effect of the first treatment carries over to influence the response to the second treatment
- Solutions
  - Rest period between treatments
  - Counterbalancing: all possible orders of treatments are included — but: O(n!)
  - Latin Square: A limited set of orders, O(n)

Latin Square

- Each condition appears at each ordinal position
- Each condition precedes and follows each other condition once
- Example for six treatments (A, B, C, D, E, F)

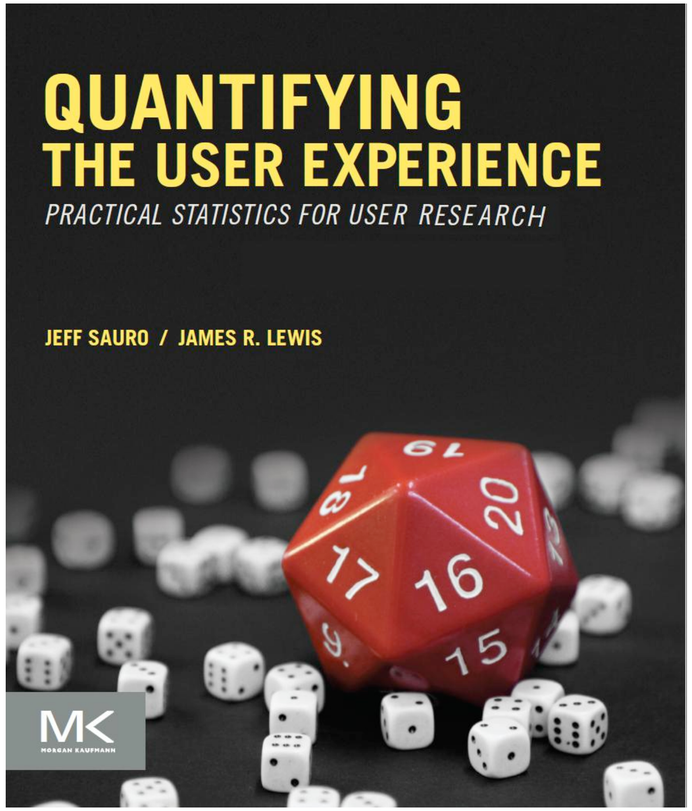
1	A	B	F	C	E	D
2	B	C	A	D	F	E
3	C	D	B	E	A	F
4	D	E	C	F	B	A
5	E	F	D	A	C	B
6	F	A	E	B	D	C

Randomization

- Randomly assign treatments to participants
- Prevents systematic bias
- But: randomization ≠ counterbalancing
  - With small numbers, randomization might not cover all combinations

Analyzing Results

- Do statistical analysis using well-defined test methods
  - E.g., Student's *t*-test, ANOVA (analysis of variance), regression analysis, Wilcoxon or Mann/Whitney test, χ^2 test
- Choice depends on number, continuity, and assumed distribution of variables, and the desired form of the result
  - Results can be simple "yes/no", size of difference, or confidence of estimate



Making Your Evaluation Valid and Reliable

- Validity: How accurate is your result?
  - **Internal validity:** Is the causal inference logical? How strong is it?
  - **External validity:** Can the result be generalized to other populations and settings?
- Reliability: How consistent or stable is your result?
  - Can the experiment be replicated by other research teams in other locations?
    - Clear procedure, avoid experimenter bias and influence of the environment,…
- These apply to *all* evaluations — not just controlled experiments

Other Evaluation Methods

- Before and during the design, with users:
  - Questionnaires
  - Personal interviews
- After completing a project:
  - Email bug report forms
  - Hotlines
  - Retrospective interviews and questionnaires
  - Field observations (observe running system in real use)

Dealing with Users

- Tests are uncomfortable for the participant
  - Pressure to perform, mistakes, competitive thinking

- So treat participants with respect at all times!
  - Before, during, and after the test

Before the Test

- Do not waste the users' time
  - Run pilot tests before
  - Have everything ready when users arrive
- Make sure users feel comfortable
  - Stress that the system is being tested, not them
  - Confirm that the system may still have bugs
  - Let users know they can stop at any time
- Guarantee privacy
  - Individual test results will be handled as private
- Inform user
  - Explain what is being recorded
  - Answer any other questions (but do not bias)
- Only use volunteers (consent form)

During the Test

- Do not waste the users' time
  - Do not let them complete unnecessary tasks
- Guarantee privacy
  - Never let users' boss (or others) watch
- Make sure users are comfortable
  - Early success in the task possible
  - Relaxed atmosphere
  - Breaks, coffee, …
  - Hand out test tasks one by one
  - Never show you are unsatisfied with what the user does
  - Avoid interruptions (cell phones, …)
  - Abort the test if it becomes too uncomfortable

After the Test

- Make sure the users are comfortable
  - Stress that the user has helped finding ways to improve the system
- Inform
  - Answer any questions that could have changed the experiment if answered before the test
- Guarantee privacy
  - Never publish results that can be associated with specific individuals
  - Show recordings outside your own group only with written consent from users

Summary

Evaluation Techniques

Evaluating Without Users

E1 Literature Review
E2 Cognitive Walkthrough
E3 Heuristic Evaluation
E4 Model-based Evaluation
- GOMS, HCI Design Patterns, …

Evaluating With Users

Qualitative

E5 Model Extraction
E6 Silent Observation
E7 Think Aloud
E8 Constructive Interaction
E9 Retrospective Testing

Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

- When, why, where, and what?
  - Lab vs. field
- Participatory Design
- How to deal with users?