ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

HUMAN COMPUTER INTERACTION GROUP

Evaluating Interactive Systems

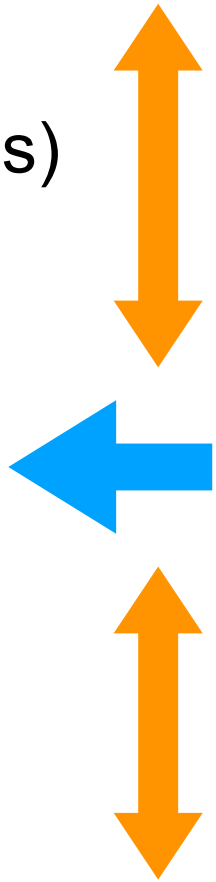
School of Computer and Communication Sciences

EPFL

Pearl Pu

A RANGE OF METHODS

- Heuristic Evaluations
- Walk-throughs (week 11 prototyping techniques)
- Inspection
- Formative Study
- Expert review
- Focus Group
- Survey
- A/B Testing



Quick methods, Formative, Summative

- **Discount evaluation:** assesses overall quality quickly
  - Heuristic evaluation

- **Formative Evaluation**: qualitative user evaluation
  - Learn why something went wrong, not just where it went wrong (breakdown points)
  - Collect data for ***qualitative*** observation
- **Summative Evaluation**: quantitative user evaluation
  - Collect bottom-line data for ***quantitative*** measurements of performance. e.g., How long did users take? Were they successful? How many errors did they make?
  - Test final products or compare alternative designs
  - E.g., A/B testing

Discount evaluation

DISCOUNT EVALUATION

- Identify usability problems following Nielsen's 10 principles (called heuristics)
- A number of experienced testers (or "evaluators") examine the interface, and judge its compliance with recognized usability principles

http://www.useit.com/papers/heuristic/

HEURISTIC EVALUATION

- Nielsen's rules are in grey box
- Additional interpretation in bullet points

Heuristic 1: Visibility of System Status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

- This rule is related to gulf of evaluation
- We add two more check points to cover the gulf of execution
  - "Can users find the controls?"
  - "Can users predict the system's reaction?"

H2: Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

- Can users understand the labels and messages in layman's terms?
- We add:
  - Is users' mental model reflected in the designer's model?

H3: User Control and Freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

- Can users back out of actions?
- We add:
  - Are users unknowingly given “intelligent software”?
  - Second guessing users’ intentions is really hard; make sure we get it right, or it backfires
  - To balance intelligent software with user control, it’s important to give them a way to opt out
  - E.g., the filter bubble story: people want to read news of opposite opinions, not just tailored to what they always read

H4: Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

- This principle means:
  - Use of language consistency
  - Visual and layout consistency
  - Screen consistency
- The same word, situation, or action should always have same meaning or effect
- We add:
  - external consistency, which is consistency with convention used elsewhere

H5: ERROR PREVENTION

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

- Error prevention
  - prevent problems from occurring in the first place (e.g., let users select file from menu rather than asking them to type the file names; confirm with users before dangerous actions; avoid using similar command names)
- We add: can users store intermediary results?
  - For major tasks, can users store intermediary results to avoid loosing them?

H6: MEMORY

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

- Memory (Recognition rather than recall) — is it faster to click the date in a visual calendar (recognition) or is it easier to type the date (recall)

H7: Flexibility & efficiency

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

- Accelerators -- abbreviations, command keys, type-ahead, edit and reissue previous commands, menu of most recently used files, macros
- We add:
  - allow users to tailor the use of frequent actions to their own key assignments

H8: Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

- This rule is related to visual hierarchy
- Remember using the following techniques to create hierarchy
  - grouping the elements to reduce complexity
  - use proximity to give structure to the interface
  - use size and value of fonts, spacing, and colors to emphasize

H9: Help users recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

H10: HELP DOCUMENTS

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

- We add:
  - use tooltips to explain purpose of user actions

Formative Study

What is formative study? Why is it called formative?

Formative studies let us observe users' responses to information and widgets on the UI as users perform their tasks;

Formative study is qualitative; it ***educates*** us about how users interact with a product;

"Educate us" thus it is called a formative study.

When to do a formative study?

- Evaluate a prototype during the development cycle

  - Develop lofi prototypes
  - Formative evaluation
  - Prototype improvement

  - Move to hifi prototypes
  - Formative evaluation
  - Prototype improvement

  - Move to final products

What are we looking for in a FS?

- Identify breakdown points
- A ***breakdown point*** refers to any ***friction*** in user's interaction with the interface
  - due to gulf of execution (e.g., mental model, poor visibility of controls)
  - due to gulf of evaluation (e.g., insufficient feedback)
- A breakdown point can be
  - self-reported by users
  - observed by you (designers or usability officers)
  - in each case, care must be taken to define what qualifies a breakdown (verbal frustrations, long delays in interaction, excessive user errors, lengthy interactions)

The structure of a formative study?

- Introduction
- Pre-study questionnaire
- Ask users to perform a set of tasks
- Post-study questionnaire

How to prepare, conduct the formative study?

1. Develop the Test Plan
2. Select and acquire participants
3. Get subjects' profile by questionnaires
4. Assign roles to testers
5. Prepare orientation scripts & training scripts
6. Design test environment
7. Prepare task list and tasks to be tested 
8. Conduct the test
9. Debrief the participants
10. Analysis and final report

Stage 1: Develop the test Plan

- Purpose, goals, and objectives of the test
  - *what kind of goals the software is to achieve: increase user satisfaction, increase conversion rate, increase task efficiency, reduce error rate*
- Participant characteristics
- Method (test design)
- Task list
- Test environment, equipment and logistics
- Test moderator role
- Data to be collected and evaluation measures
- Report contents and presentation

STAGE 2: GETTING PARTICIPANTS

- Select users based on Persona (you can develop a screening questionnaire to ensure users fit profile)
- Make sure they are first-time users
- Acquire test users via employment agency, students, existing customers, internal personnel
- Maintain a database of potential test users

How does the industry practice this?

- Testing is very expensive (budget won't allow many subjects)
- Nielson showed that after ***3*** subjects, the benefits go down
- After each formative study, improve design and update prototypes before using the next user
- In course project, you will test three users on the same product; observe how users' behaviors form patterns

Stage 3: Subject profiling

- What do they do
- Age
- Gender
- What OS and software they use
- How do they perform the tasks currently?
- What are their motivations for using the tested software?

BACKGROUND QUESTIONNAIRE

- Admin. Data: date, test number, user number or id
- General Data: age, sex, educational level
- Computer Experience: total time, frequency of use, types of software, having used a GUI before
- Application Experience: total time, frequency of use, brand

A SAMPLE QUESTIONNAIRE

Background questionnaire
Date: _____ Tast No.: _____ User No.: _____
General information
Age: ____ Sex: _____
Education
1. Please circle the highest educational grade you have achieved:
secondary school University degree Doctorate
2. If you are a student or graduate, pleast state your major area of study
_____
Computer Experience
1. How long have you been using personal computers (years and months)? _____
2. In a typical week, how many hours do you use a computer? _____
3. Please circle the types of system you have used, followed by approximately how many months experience you have with them.

Operating systems	**Months of experience**
DOS	
OS/2	
Windows(95,98)	
Windows NT	
Unix	
Mac	
Application	
Online travel system reservation	
online text search engine	
online shopping system	
java application	

Stage 4: Assigning testing roles

1. Test administrator
   - is in charge of test and responsible for all interaction with test user (introduction, test, debriefing)
2. Data logger
   - records activities and events of interest on paper, incl. time of occurrence
3. Technical expert
   - restart after system crash, unexpected hangups, etc.
   - may want to set screen resolution down to say 640 x 480, so icons and test are bigger and easier to recognize on video

Stage 4: Assigning testing roles

4. Video operator (only if users give consent)
   - responsible for recording entire test proceedings, incl. Initial instructions and debriefing
   - check camera angles so user and interface both clearly visible
   - use manual focus
   - ensure audio recording level is high enough
   - label, copy and edit tapes
5. Test participants or subjects
   - users taking part in the test

We recommend having max 2-3 people besides the user

Stage 5: orientation script

- Introduce yourself and others in the testing room
- Explain purpose of test (to improve product usability)
- Emphasize that system is being tested, not user
- Acknowledge software is new and may have problems

Stage 5: orientation script

- Do not mention any association you have with product (do mention if you are not associated with product)
- Explain any recording (reassure confidentiality)
- Say user may stop at any time
- Say user may ask questions at any time, but they may not be answered right away
- Invite questions

ORIENTATION SCRIPT – EXAMPLE (1)

Hi. My name is Paul. I'll be working with you to test a software prototype. It's an online travel planing system. We need your help to find out how user-friendly the system is.

I'll be asking you to perform some typical tasks with the system. Do your best, but don't be too concerned with results -- ***the system is being tested, not your performance***.

ORIENTATION SCRIPT – EXAMPLE (2)

Please ask questions at any time, but I may only answer them at the end of the session.

While you are working, I'll be taking some notes and timings. We will also be videotaping the session for analysis of testing results.

If you feel uncomfortable, you may stop the test at any time.

Do you have any questions?

If not, then let's begin by filling out a short background questionnaire and having you sign the non-disclosure agreement

Stage 5: Training script

Exact written description of prior training

- Demonstration of GUI
- Demonstration of special interaction styles: mouse keys, drag-and-drop, etc.
- Walk-through of sample task
- ***Make sure not to give away hints for performing the tasks you want to test***
- Demo of how to think aloud (for thinking aloud style tests)

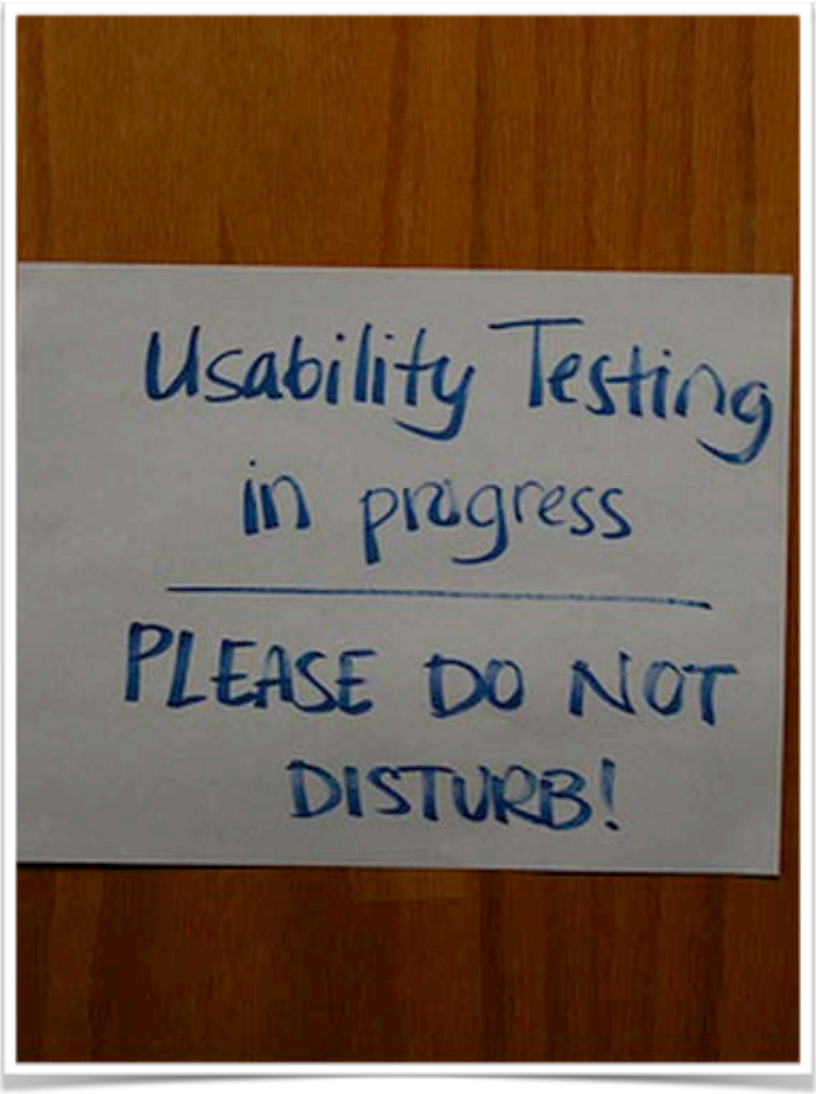
Stage 6: Design testing environment

- Environment can range from simple to sophisticated installations, e.g.,
  - 2-3 testers and one user in a quiet room where the user works
  - one tester and one user with video recording in a quiet room
  - A group of observers in one-way mirror room

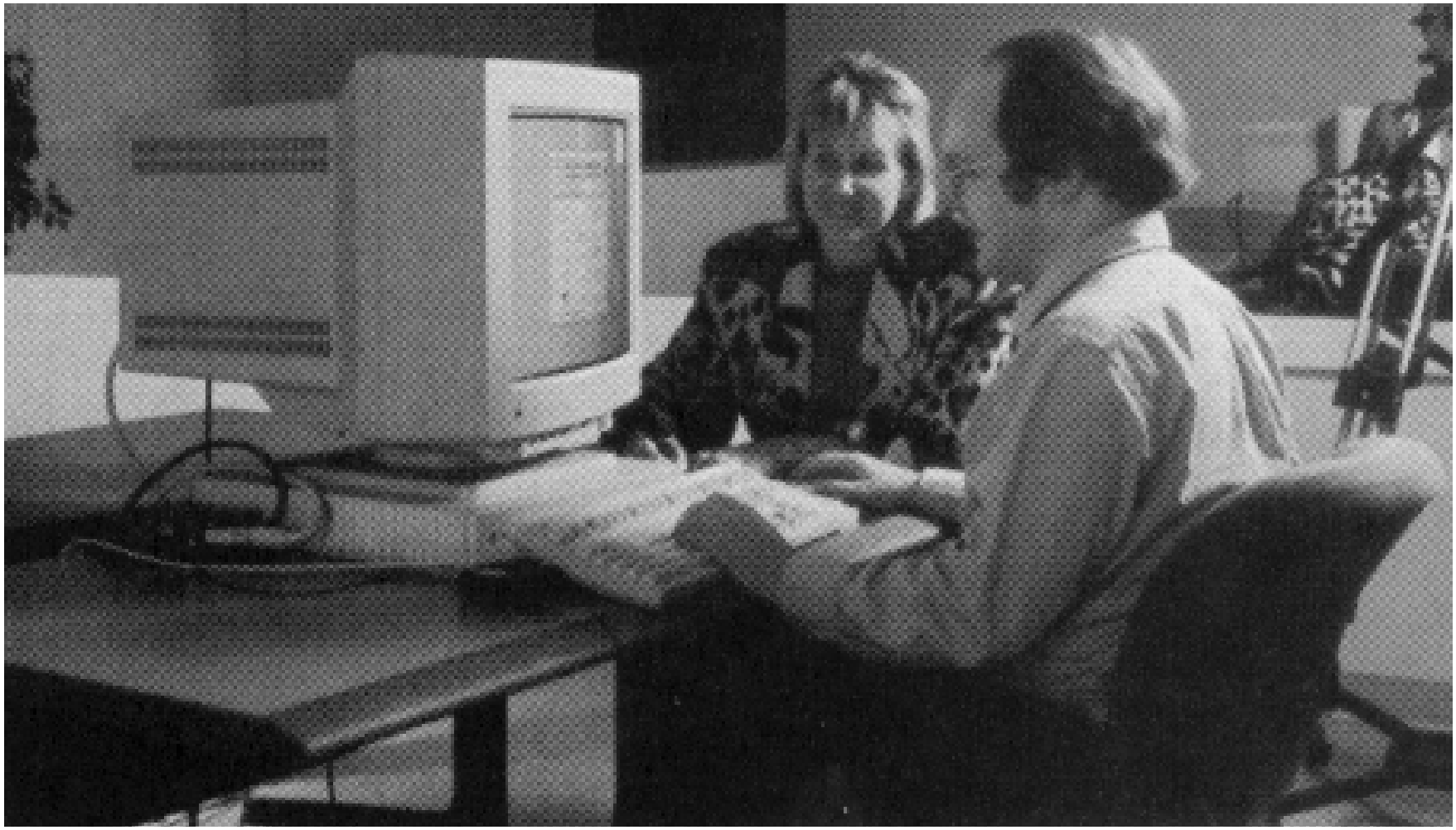
Simple testing room: for course project

Ensure comfortable test environment

- Organize a quiet room
- ***Put up sign "Don't disturb"***
- Disable telephones
- Provide refreshments



Option: use video and audio recording



Video captures the user's actions and the screen while microphones capture thinking-aloud comments (Shneiderman 1998, p129)

Used in industry, not in course project

Sophisticated testing center: Observers watch the subject through a one-way mirror window (Shneiderman 98, p130)

Stage 7: Prepare task list

- List all task scenarios (general and specific tasks)
- Prioritize tasks by frequency and criticality
- Choose those most frequent and critical to test first

Example: Online travel site

All specific tasks

- Retrieve & refine trips
- Define trips
- Browse & select trips
- Keep trips in basket
- Reserve trips
- Purchase trips (critical, but may not be as frequent as others)

How to turn a task into a test

For each task:

- Define any prerequisites
- Define successful completion criteria
- Specify maximum time to complete each task, after which help may be given
- Define what constitutes an error

Task Scenarios

Task descriptions given to the test users

- Simple introductory first task (early success)
- Realistic scenarios in typical order
- If sequential ordering is not crucial, randomnize the order
- Write each task scenario on a separate sheet (do not hand the user all the tasks at once, but one at a time)
- ***Do not guide participants through the task***
- Describe the goal rather than the individual steps
- Put the tester into his/her role (you are a 40-year mom planning a trip for your family)

PREPARE DETAILED TASK LIST FOR EACH USERS

- If behavior variables are important, personalize task scenarios
  - Participant A (primary persona) - plans trips for her self
  - Participant B (secondary persona): plans trips for family, but assistant plans the business trips
    - recruit an assistant to testing
    - participant B modifies business trips on the road
  - Participant C: plans trips for family, also plans trips for business on his/her own

A EXAMPLE OF TASK LIST

Task	Description	Criteria
1	Define itinerary for a trip from Geneva to Hamburg	**PreReq**: user is on the main page (sign-in taken care of)
		Completed: an itinerary is defined
		MaxTime: 10 min
2	Select a trip from the results that stops in Munich	**PreReq**: Results from Hamburg trip have been returned
		Completed: a trip via Munich is selected
		Max time: 5 min
3	Search for a trip that leaves between 8 and 10 am	**PreReq**: Results from Hamburg trip have been returned
		Completed: trips satisfying the preferences have been found
		Max time: 5 min
4	Search for a trip that arrives in Hamburg around noon	**PreReq**: results from Hamburg trip have been returned
		Completed: trips satisfying the preferences have been found
		Max Time: 3 min
5	Search for a trip that returns to Geneva around 9pm	**PreReq**: results from Hamburg trip have been returned
		Completed: trips satisfying the preferences have been found
		Max Time: 3 min

Stage 8: Conduct test

- Do you have all the test materials?
- To answer this question, it's best to use a checklist.

TEST CHECKLIST

1. Everything ready in test room
2. Prepare yourself mentally
3. Greet the participant
4. Read the orientation script and set the stage
5. Have participant sign consent forms
6. Administrator gives background questionnaires
7. Move to testing area
8. Provide any prior training
9. Record starting time
10. Distribute or read written task scenarios to participant one at a time
11. Observe, note interesting and critical events
12. Administrator gives post test questionnaires
13. Debriefing interview
14. Thank participant, provide any remuneration, show him out
15. Organize data sheets and notes
16. Summarize thoughts about test
17. Prepare for next participant

(Adapted from Andrews Keith)

POST STUDY QUESTIONNAIRE

- Collect feelings, opinions, suggestions
  - Interface organization matches real-world tasks?
  - Too much or too little information on screen?
  - Similar information consistently placed?
  - Problems with navigation?
  - Too much computer jargon used?
  - Appropriate use of color?

STAGE 9: DEBRIEFING SUBJECTS

- Let subject speak thoughts first: “So, what did you think?”
- Top-down: probe high-level issues from task list, then more detailed questions about actions
- Review answers to post-study questionnaires

STAGE 10: ANALYSIS RESULTS

- Identify breakdowns and pain points which arose
- Diagnose the source of each of them
- Prioritize them by their severity or criticality

Table of contents of test report

- Purpose
- Problem statement
- Gather user profile information using questionnaires
- Method employed (formative or summative)
- Test environment (describe your environment)
- Task list
- Data collected and analysis of data
- Summarize good points, identify problems
- Improvements or design guidelines

THE END