Evaluating the Design without Users

- This chapter illustrates the importance of evaluating the evolving design when no users are present.
- A good evaluation can catch problems that an evaluation with only a few users many not reveal.
- This type of testing won't uncover everything either but doing both kinds will significantly improve your chances for success.

Evaluating the Design without Users

- Cognitive Walkthrough
- Action Analysis
- Heuristic Evaluation

Cognitive walkthrough

- A formalized way of imagining people's thoughts and actions when they use an interface for the first time
  - you have a prototype or a detailed design description of the interface and you know who the users will be
  - you select one of the tasks that the design is intended to support
  - try to tell a believable story about each action a user has to take to do the task
  - to make the story believable you have to motivate each of the user's actions, relying on the user's general knowledge and on the prompts and feedback provided by the interface

Information required

1. A description or prototype of the interface
2. a task description
3. a complete, written list of the actions needed to complete the task with the interface
4. an idea of who the users will be and what kind of experience they'll bring to the job

Performing the walkthrough ...

- Keep four key questions in mind as you critique the story:
  - will users be thinking the designers assume they will be thinking?
  - Will users see the control (button, menu, switch, etc.) for the action?
  - Once users find the control, will they recognize that it produces the effect they want?
  - After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?

What you should look for during the walkthrough

- Will users be trying to produce whatever effect the action has?
- Will users see the control (button, menu, switch, etc.) for the action?
- Once users find the control, will they recognize that it produces the effect they want?

What you should look for during the walkthrough (Continued)

- After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?

- Additional things to look for

Documenting the walkthrough

- Important to document the cognitive walkthrough to keep a good record of what needs improvement
  - produce standard evaluation forms

  - for each action, a separate standard form is filled out
  - document all negative answers on a separate usability problem report sheet

Example walkthrough

1. A description or prototype of the interface
2. a task description
3. a complete, written list of the actions needed to complete the task with the interface
4. an idea of who the users will be and what kind of experience they'll bring to the job

#1 Are users thinking what the designers assume they will be thinking?

#2 Can users see the control (button, menu, switch, etc.) for the action?

#3 Once users find the control, can they recognize that it produces the effect they want?

#4 After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?

Who and when?

- Small piece of an interface
- larger parts of an interface

Common mistakes

- the “written list of the actions” is merged into the walkthrough
- the walkthrough is viewed as testing real users on the system

What to do with the results?

- Fix the interface!!
  - many of the fixes will be obvious
  - harder problem exists when the user doesn't have any reason to think that an action needs to be performed

Evaluating the Design without Users

- Cognitive Walkthrough
- **Action Analysis**
- Heuristic Evaluation

Action analysis

- Action analysis is used to examine the sequence of actions that a user has to perform to compete a task with an interface.
- keystroke-level analysis
- back of the envelope

Two fundamental phases

1. Decide what *physical* and *mental* steps the user will perform to complete one or more tasks with the interface
2. Analyze those steps looking for problems

Keystroke level analysis

- Sometimes referred to as a "formal action analysis"
- each task is broken down into unit tasks
- the time to perform each unit task (physical or mental) is estimated
- times are totaled to predict the time for the action

Average times for computer interface actions

Physical movements:	
• Enter one keystroke on a standard keyboard	.28 seconds
• Use mouse to point at an object on the screen	1.5 seconds
• move had to pointing device or function key	.3 seconds
Visual perception:	
• respond to a brief light	.1 seconds
• recognize a six letter word	.34 seconds
• move eyes to a new location on the screen	.23 seconds
Mental Actions:	
• retrieve a simple item from long-term memory	1.2 seconds
• learn a single "step" procedure	25 seconds
• execute a mental "step"	.075 second
• choose among methods	1.2 seconds

Back of the envelope analysis

- similar to keystroke analysis:
- difference:
- every action takes at least two or three seconds
- or observe a few users and time the actions
- useful for deciding whether or not to add features to an interface

GOMS Model

Goals, **O**perators, **M**ethods, **S**election

- Goals
- Operators

GOMS Model

- **G**oals, **O**perators, **M**ethods, **S**election
- Methods
- Selection

GOMS meets the phone company

- Small differences in time per call can result in large monetary savings
- NYNEX
  - each second reduction per call in work time for its Toll & Assistance Operators saves three million US dollars per year!
  - New workstations
- Initially used experimental studies and a GOMS analysis

Results

- Experimental data revealed that operators were SLOWER with the new workstations!
- Obvious possibilities
- The GOMS analysis revealed WHY the old workstation was faster than the new

Wednesday, June 16th

- Reminder:
  - Assignment 2, Part II is due on June 16th
- Assignment 3 will be handed out on June 16th. You will need to be in groups of 4.
- Readings:
  - Article 13: Usability Heuristics by Jakob Nielsen
  - Article 14: Designing for Error by Clayton Lewis and Donald A. Norman