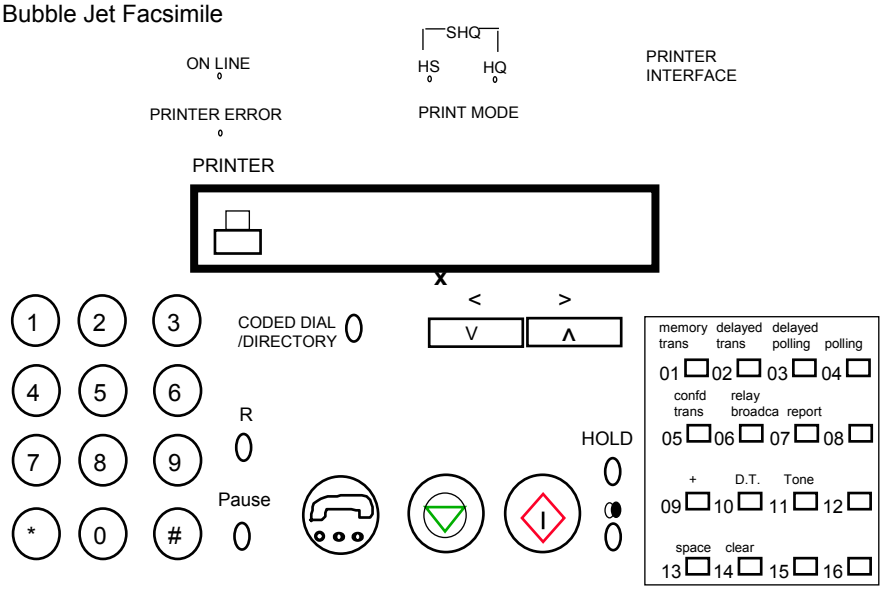
Qualitative Evaluation Techniques

Quickly debug and evaluate prototypes by observing people using them

Specific evaluation methods helps you discover what a person is thinking about as they are using your system

Canon
Fax-B320
Bubble Jet Facsimile



Discount Usability Evaluation

Low cost methods to gather usability problems

- approximate: capture most large and many minor problems

How?

- Quantitative
- Qualitative

Qualitative Approach For Usability Evaluation

Description of approach:

- Count, log, measure something of interest in user actions
- Speed, error rate, counts of activities

Qualitative Methods For Usability Evaluation

Description of approach:

- produces a description, usually in non-numeric terms
- may be subjective

Methods

- Introspection
- Extracting the conceptual model
- Direct observation
  - simple observation
  - think-aloud
  - constructive interaction
- Query via interviews and questionnaires
- Continuous evaluation via user feedback and field studies

The Introspection Method

Designer tries the system (or prototype) out

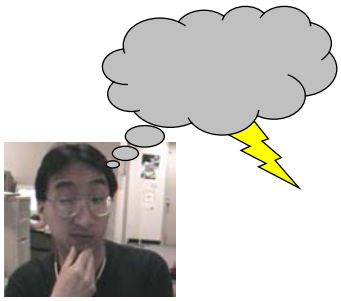
- does the system "feel right"?
- most common evaluation method

Benefits

- can probably notice some major problems in early versions during every day use

Problems

- not reliable as completely subjective
- not valid as introspector is a non-typical user
- intuitions and introspection are often wrong



Conceptual Model Extraction

Show the user static images of:

- the paper prototype *or*
- screen snapshots *or*
- actual system screens during use

Have the user try to explain

- what all elements are
- what they would do to perform a particular task

Initial vs. formative conceptual models

- **Initial:** how person perceives a screen the very first time it is viewed
- **Formative**: the same, except after the system has been used for a while

This approach is:

- Good for eliciting people's understanding before & after use
- Requires active intervention by evaluator, which can get in the way

Direct Observation

Evaluator observes and records users interacting with design/system

- in lab:
  - user asked to complete a set of pre-determined tasks
  - a specially built and fully instrumented usability lab may be available
- in field:
  - user goes through normal duties

This approach is:

•Validity/reliability depends on how controlled/contrived the situation is

•Excellent at identifying gross design/interface problems

Three general approaches:

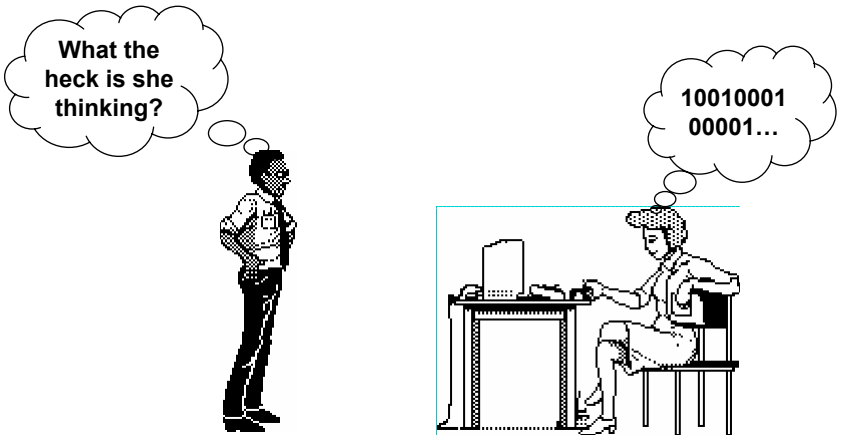
- simple observation
- think-aloud
- constructive interaction

Simple Observation Method

User is given the task, and evaluator just watches the user

Problem

- does not give insight into the user's decision process or attitude



The Think Aloud Method

Test users are asked to say what they are thinking/doing

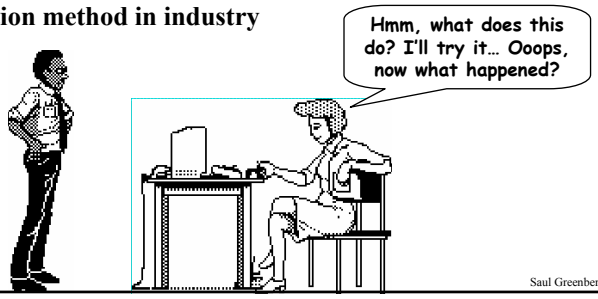
- what they believe is happening
- what they are trying to do
- why they took an action

- Gives insight into what the user is thinking

Problems

- awkward/uncomfortable for person (thinking aloud is not normal!)
- "thinking" about it may alter the way people perform their task
- hard to talk when they are concentrating on problem

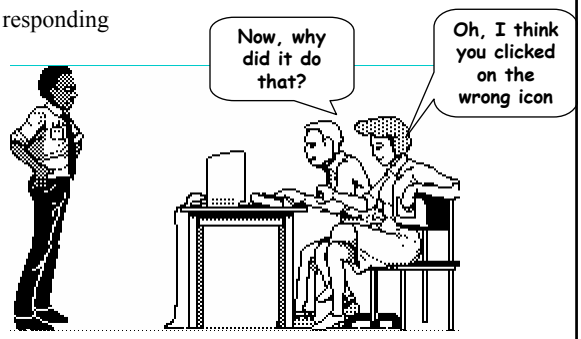
Most widely used evaluation method in industry



The Constructive Interaction Method

Two people work together on a task

- normal conversation between the two users is monitored
  - removes awkwardness of think-aloud
- Variant: Co-discovery learning
  - use semi-knowledgeable "coach" and novice user together
  - only novice uses the interface
- results in
  - novice user asking questions
  - semi-knowledgeable coach responding
  - provides insights into thinking process of both beginner and intermediate users



Recording Observations

How do we record user actions during observation for later analysis?

- if no record is kept, evaluator may forget, miss, or mis-interpret events

- paper and pencil
  - primitive but cheap
  - evaluators record events, interpretations, and extraneous observations
  - hard to get detail (writing is slow)
  - coding schemes help…

- audio recording
  - good for recording talk produced by thinking aloud/constructive interaction
  - hard to tie into user actions (ie what they are doing on the screen)

- video recording
  - can see and hear what a user is doing
  - one camera for screen, another for test user (picture in picture)
  - can be intrusive during initial period of use

Coding Scheme Example...

tracking a person's activity in the office

s = start of activity
e = end of activity

	Desktop activities			Absences		Interruptions	
Time	working on computer	working on desk	initiates telephone	away from desk but in room	away from room	person enters room	answers telephone
9:00	s						
9:02	e					s	
9:05					s	e	
9:10			s		e		
9:13							

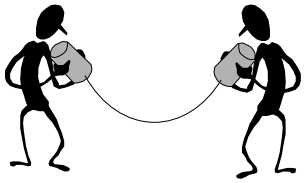
Querying Users Via Interviews

Excellent for pursuing specific issues

- vary questions to suit the context
- probe more deeply on interesting issues as they arise
- good for exploratory studies via open-ended questioning
- often leads to specific constructive suggestions

Problems:

- accounts are subjective
- time consuming
- evaluator can easily bias the interview
- prone to rationalization of events/thoughts by user
  - user's reconstruction may be wrong



How To Interview

Plan a set of central questions

- could be based on results of user observations
- gets things started
- focuses the interview
- ensures a base of consistency

Try not to ask leading questions

Follow interesting leads rather than bulldozing through question list

Group discussions

•start with individual discussions to discover different perspectives, and continue with group discussions

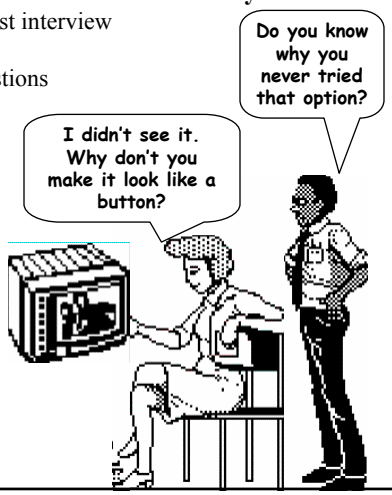
•Increasing group size may increase the universality of the comments

•May encourage cross discussions.

Retrospective Testing

Post-observation interview to clarify events that occurred during system use

- perform an observational test
- create a video record of it
- **have users view the video and comment on what they did**
  - excellent for grounding a post-test interview
  - avoids erroneous reconstruction
  - users often offer concrete suggestions



Querying Users Via Questionnaires And Surveys

Questionnaires / Surveys

- preparation "expensive," but administration cheap
  - can reach a wide test group (e.g. mail)
- does not require presence of evaluator
- results can be quantified
- anonymous
- but there are drawbacks
- See also the url below for a checklist on questionnaire design
  http://pages.cpsc.ucalgary.ca/~tamj/2002/hci_topics/assignments/usability/questionnaire_tips.html

Styles Of Questions

Open-ended questions

- asks for unprompted opinions
- good for general subjective information
  - but difficult to analyze rigorously

Can you suggest any improvements to the interfaces?

Styles Of Questions

Closed-ended questions

- restricts the respondent's responses by supplying alternative answers
- data is more narrow (less rich but can be easily analyzed)
- but watch out for hard to interpret responses!
  - alternative answers should be very specific

Do you use computers at work:

O often O sometimes O rarely

vs ✓

In your typical work day, do you use computers:

O over 4 hrs a day
O between 2 and 4 hrs daily
O between 1and 2 hrs daily
O less than 1 hr a day

✓

Styles Of Questions

Scalar

- ask user to judge a specific statement on a numeric scale
- scale usually corresponds with agreement or disagreement with a statement

Characters on the computer screen are:

hard to read easy to read

1 2 3 4 5

Styles Of Questions

Multi-choice

- respondent offered a choice of explicit responses

How do you most often get help with the system? (tick one)

O on-line manual
O paper manual
✓ ask a colleague

Which types of software have you used? (tick all that apply)

✓ word processor
O data base
✓ spreadsheet
O compiler

Styles Of Questions

Ranked

- respondent places an ordering on items in a list
- useful to indicate a user's preferences
- forced choice

Rank the usefulness of these methods of issuing a command
(1 most useful, 2 next most useful..., 0 if not used

__2__ command line
__1__ menu selection
__3__ control key accelerator

Styles Of Questions

Combining open-ended and closed-ended questions

- gets specific response, but allows room for user's opinion

It is easy to recover from mistakes:

disagree agree comment: *the undo facility is really helpful*

1 2 3 (4) 5

Querying Users Via Questionnaires / Surveys

How

- establish the purpose of the questionnaire
  - what information is sought?
  - how would you analyze the results?
  - what would you do with your analysis?
- do not ask questions whose answers you will not use!
  - e.g. how old are you?
- determine the audience you want to reach
  - typical survey: random sample of between 50 and 1000 users of the product
- determine how would you will deliver and collect the questionnaire
  - on-line for computer users
  - web site with forms
  - surface mail
    including a pre-addressed reply envelope gives far better response
- determine the demographics
  - e.g. computer experience

Interviews Vs. Questionnaires (Pros And Cons)

Preparation time

Unanticipated/unexpected events

Depth of information

Analysis time

Continuous Evaluation

Usually done in later stages of development

- (ie beta releases, delivered system)

Good for monitoring problems of system in actual use

Problems can be fixed in next release

a) User feedback via gripe lines

- users can provide feedback to designers while using the system
  - email
  - special built-in gripe facility
  - telephone hot line
  - help desks
  - suggestion box
  - bulletin board
- best combined with trouble-shooting facility
  - users always get a response (solution?) to their gripes

Continuous Evaluation...

b) Case/field studies

- careful study of "system usage" at the site
- good for seeing "real life" use
- external observer monitors behaviour or gets feedback via methods described above

What You Now Know

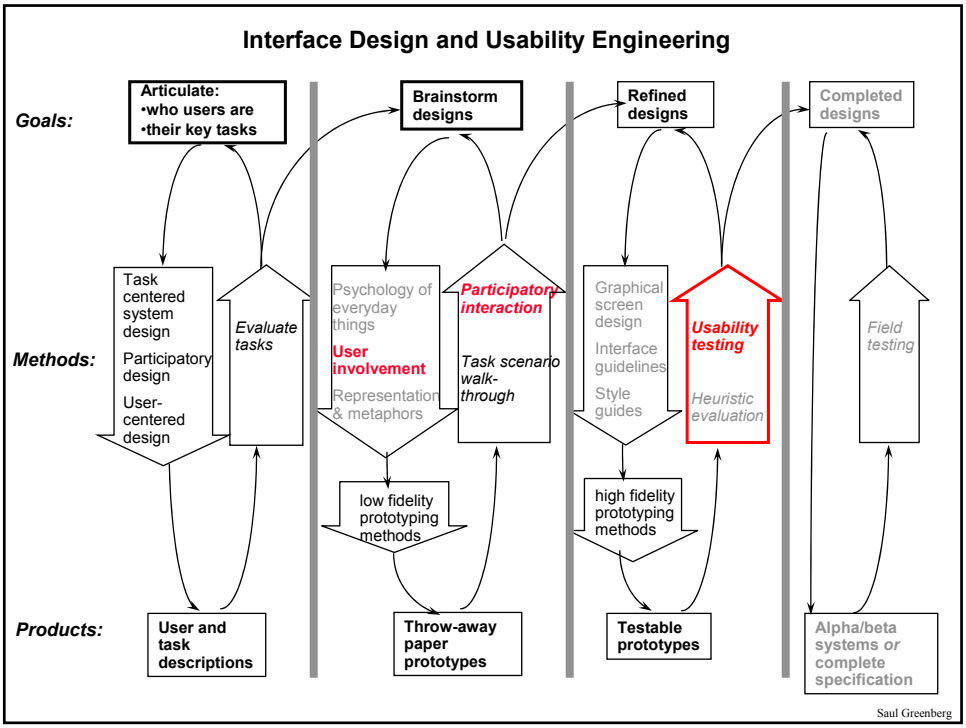
Observing a range of users use your system for specific tasks reveals successes and problems

Qualitative observational tests are quick and easy to do

Several methods reveal what is in a person's head as they are doing the test

Particular methods include

- Conceptual model extraction
- Direct observation
  - simple observation
  - think-aloud
  - constructive interaction (co-discovery learning)
- Query via interviews, retrospective testing and questionnaires
- Continuous evaluation via user feedback and field studies


Interface Design and Usability Engineering
Goals:
Articulate:
•who users are
•their key tasks
Brainstorm designs
Refined designs
Completed designs
Methods:
Task centered system design
Participatory design
User-centered design
Evaluate tasks
Psychology of everyday things
User involvement
Representation & metaphors
Participatory interaction
Task scenario walk-through
Graphical screen design
Interface guidelines
Style guides
Usability testing
Heuristic evaluation
Field testing
low fidelity prototyping methods
high fidelity prototyping methods
Products:
User and task descriptions
Throw-away paper prototypes
Testable prototypes
Alpha/beta systems or complete specification
Saul Greenberg