Evaluation Techniques

Gary Marsden
(gaz@cs.uct.ac.za)
2006
Humility

➤ There are lots of reasons not to conduct evaluation
➤ Often these have to do with pride or laziness

➤ There is no way out, however

➤ Evaluation is essential
Who, why, what, when, where...

- Different books look at different schemes for classifying evaluation
- In the spirit of this course, we will take a pragmatic approach based on prototyping life-cycles
  - On whom: Users; experts; mental models
  - Where: Lab; outside; in office
  - Results: Numbers; words; lots of words
A quick word on results

- All results are classified as:
  - qualitative:
    - observe user interactions
    - gather user explanations and opinions
    - produces a description, usually in non-numeric terms
    - anecdotes, transcripts, problem areas, critical incidents...
  - quantitative
    - count, log, measure something of interest in user actions
    - speed, error rate, counts of activities,
Quick and Dirty

- Designer tries the system (or prototype)
  - does the system “feel right”?
  - benefits
    - can catch some major problems in early versions
  - problems
    - not reliable as completely subjective
    - not valid as introspector is a non-typical user
    - intuitions and introspection are often wrong

- More formal versions discussed soon
Conceptual model extraction

➢ How?
  - show the user static images of
    • the prototype or screens during use
  - ask the user explain
    • the function of each screen element
    • how they would perform a particular task

➢ What?
  - Initial conceptual model
    • how person perceives a screen the very first time it is viewed
  - Formative conceptual model
    • How person perceives a screen after its been used for a while

➢ Value?
  - good for eliciting people’s understanding before & after use
  - poor for examining system exploration and learning
Example
Direct observations

- Evaluator observes users interacting with system
  - in lab:
    - user asked to complete a set of pre-determined tasks
  - in field:
    - user goes through normal duties

- Value
  - excellent at identifying gross design/interface problems
  - validity depends on how controlled/contrived the situation is
Simple observation method

- User is given the task
- Evaluator just watches the user

Problem
- does not give insight into the user’s decision process or attitude
Think aloud method

- Users speak their thoughts while doing the task
  - what they are trying to do
  - why they took an action
  - how they interpret what the system did
  - gives insight into what the user is thinking
  - most widely used evaluation method in industry
  - may alter the way users do the task
  - unnatural (awkward and uncomfortable)
  - hard to talk if they are concentrating

Hmm, what does this do? I'll try it... Ooops, now what happened?
Constructive interaction method

- Two people work together on a task
  - monitor their normal conversations
  - removes awkwardness of think-aloud
- Co-discovery learning
  - use semi-knowledgeable “coach” and novice
  - only novice uses the interface
    - novice ask questions
    - coach responds
    - gives insights into two user groups
Recording observations

How do we record user actions for later analysis?
  - otherwise risk forgetting, missing, or misinterpreting events

  - paper and pencil
    - primitive but cheap
    - observer records events, comments, and interpretations
    - hard to get detail (writing is slow)
    - 2nd observer helps...

  - audio recording
    - good for recording think aloud talk
    - hard to tie into on-screen user actions

  - video recording
    - can see and hear what a user is doing
    - one camera for screen, rear view mirror useful...
    - initially intrusive
    - Boring!
Coding sheet example...

- tracking a person’s use of an editor

<table>
<thead>
<tr>
<th>Time</th>
<th>text</th>
<th>General actions</th>
<th>Graph editing</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00</td>
<td>editing</td>
<td>×</td>
<td>node</td>
<td></td>
</tr>
<tr>
<td>09:02</td>
<td>error</td>
<td>×</td>
<td>node</td>
<td>×</td>
</tr>
<tr>
<td>09:05</td>
<td>image</td>
<td>×</td>
<td>node</td>
<td>×</td>
</tr>
<tr>
<td>09:10</td>
<td>new</td>
<td>×</td>
<td>node</td>
<td></td>
</tr>
<tr>
<td>09:13</td>
<td>delete</td>
<td>×</td>
<td>node</td>
<td></td>
</tr>
<tr>
<td></td>
<td>modify</td>
<td>×</td>
<td>node</td>
<td></td>
</tr>
</tbody>
</table>
Heuristic Evaluation

- As popularised by Jakob Nielsen
What is it?

- Have a bunch of experts look at an interface and compare with design heuristics
  - Not personal opinion
  - See hand-out for heuristics

- In practice
  - Highly interactive (not like walk-through)
  - 1 to 2 hours long
  - Multiple people (don’t talk to each other)
Output

- Written up (or not)
- List of usability problems [with explanation]
- Can’t always give fix
- Usually have debrief with development team
  - Can be a bit depressing
How many evaluators?

- Bane of my life – cheers Jakob
Types of problem found

- Tend to find more minor problems
  - Need to rate severity of each problem 0-4
  - **Frequency**
  - **Impact**
  - **Persistence**
  - **Marketing**

- Good to intersperse user testing and heuristic evaluation
Many HCI people are skeptical about heuristic evaluation

Prominent among these is Gilbert Cockton who thinks that HE works by accident

- Proposes an alternative scheme with no explicit heuristics, but a well designed feedback form
- We are trying it at the moment, but no hard data
Interviews

- Good 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
  - a few good questions gets things started
    • avoid leading questions
  - focuses the interview
  - could be based on results of user observations

➢ Let user responses lead follow-up questions
  - follow interesting leads vs bulldozing through question list
Retrospective testing interviews

Post-observation interview to

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

Do you know why you never tried that option?

I didn’t see it. Why don’t you make it look like a button?
Critical incidence interviews

> People talk about incidents that stood out
  - usually discuss extremely annoying problems with fervor
  - not representative, but important to them
  - often raises issues not seen in lab tests

Tell me about the last big problem you had with Word

I can never get my figures in the right place. It's really annoying. I spent hours on it and I had to...
Questionnaires and Surveys

➢ Questionnaires / Surveys
  – preparation “expensive,” but administration cheap
    • can reach a wide subject group (e.g. mail)
  – does not require presence of evaluator
  – results can be quantified

➢ But
  – only as good as the questions ask
Questionnaires and 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!

- determine the audience you want to reach

- determine how would you deliver / collect the questionnaire
  - on-line for computer users
  - web site with forms
  - surface mail
    - pre-addressed reply envelope gives far better response
Continuous Evaluation

- Monitor systems in actual use
  - usually late stages of development
    - ie beta releases, delivered system
  - fix problems in next release

- User feedback via gripe lines
  - users can provide feedback to designers while using the system
    - help desks
    - bulletin boards
    - email
    - built-in gripe facility
  - best combined with trouble-shooting facility
    - users always get a response (solution?) to their gripes
Continuous evaluation

➢ Case/field studies
  - careful study of “system usage” at the site
  - good for seeing “real life” use
  - external observer monitors behavior
  - site visits
Scientific experimentation

- This is what people often think is the sum total of evaluation
- Ben Schniederman in particular pushed this approach and did much to ensure a sound scientific footing for HCI
- Like all such experiments, results are constrained but convincing
Example

- Imagine you want to design a better system for text input on a mobile device. This will involve comparing the new system (NS) to the previous best system (PBS)
- We need to start with thinking about ‘better’
Steps in an experiment

- Hypothesis: state the question in a testable way.
- Assume that ‘better’ means faster entry
  - Happy to ignore accuracy issues (assume SMS application)
  - Assume we want to test whilst seated and walking
- Time now becomes a dependent variable
- System choice is independent
- Hypothesis given in terms of dependent and independent variables:
  - Users will be able to input text more quickly using NS rather than PBS either seated or walking
If only it were that easy

- To be statistically sound, we do not prove the hypothesis correct
- Instead we prove that the opposite of the hypothesis (null-hypothesis) is incorrect
- Null-hypothesis: takes same time to enter text on PBS and NS regardless of where they are
- If you are confused, think about it in terms of raw data and ignore context
- Usually we rent a tame psychologist
Who are the users

- Not students or friends!
- Choose representative sample of target users
- In our case, busy executives
  - Expensive!
- We also need enough of them to make our results meaningful
  - Finding one guy who likes the system is not enough
How many

- This is tricky, but 10 people per condition is close enough
- We are going to need 40 people
  - Actually, we can cheat a bit

<table>
<thead>
<tr>
<th></th>
<th>NS</th>
<th>PBS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text sitting</strong></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td><strong>Text walking</strong></td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
Tasks

- We need to get people to actually enter some text so they can be timed
- Could use a paragraph from a novel (not so good)
- Could use re-cycled notes from other PDA users (much better)
Experiment structure

- Everyone enters using PBS and then NS (within-groups)
  - Learning effects mean people get progressively better, so NS is always best
  - Or they get tired and slower, so PBS is better
- Or we just get 10 people for each condition (between groups)
  - Get effect from having all the weirdos in one group
- Use a mixture
  - Say 20 people, 10 walking, 10 sitting
  - For walking/sitting, 5 start with PBS and 5 with NS
  - Task order randomised
Interpreting results

- Really don’t want to get in to this

- Essentially you will have mean (average) timing results for each condition
  - You want to know if these changes could have occurred by random: statistics let you say to a given degree of certainty if the effect was random or due to the system
  - 95% certain is good enough (you cannot be 100% certain)

- Remember, the statistics only check the significance of the numbers you recorded, not the significance of the result
  - An improvement of 0.1 ms in NS is not really significant in a pragmatic sense
Mobile Systems

- All of the previous evaluation techniques apply to mobile systems
- The big difference is looking at context:
  - Physical environment
  - Social setting
  - Technology infrastructure
- These throw up a lot of issues
  - .....
Ethics

- Testing can be a distressing experience
  - pressure to perform, errors inevitable
  - feelings of inadequacy
  - competition with other subjects

Golden rule
  - subjects should always be treated with respect
Ethics – before the test

- Don’t waste the user’s time
  - use pilot tests to debug experiments, questionnaires etc
  - have everything ready before the user shows up

- Make users feel comfortable
  - emphasize that it is the system that is being tested, not the user
  - acknowledge that the software may have problems
  - let users know they can stop at any time

- Maintain privacy
  - tell user that individual test results will be completely confidential

- Inform the user
  - explain any monitoring that is being used
  - answer all user’s questions (but avoid bias)

- Only use volunteers
  - user must sign an informed consent form
Ethics – during the test

➤ Don’t waste the user’s time
  - never have the user perform unnecessary tasks

➤ Make users comfortable
  - try to give user an early success experience
  - keep a relaxed atmosphere in the room
  - coffee, breaks, etc
  - hand out test tasks one at a time
  - never indicate displeasure with the user’s performance
  - avoid disruptions
  - stop the test if it becomes too unpleasant

➤ Maintain privacy
  - do not allow the user’s management to observe the test
Ethics – after the test

- Make the users feel comfortable
  - state that the user has helped you find areas of improvement

- Inform the user
  - answer particular questions about the experiment that could have biased the results before

- Maintain privacy
  - never report results in a way that individual users can be identified
  - only show videotapes outside the research group with the user’s permission