Evaluating the User Interface

Michigan State University
E. Kraemer
CSE491, Fall 2007

Outline
- The Role of Evaluation
- Usage Data: Observations, Monitoring, User’s Opinions
- Experiments and Benchmarking
- Interpretive Evaluation
- Predictive Evaluation
- Comparing Methods

The Role of Evaluation
- What do you want to know and why?
- When and how do you do evaluation?

Evaluation
- Concerned with gathering data about the usability of
  - a design or product
  - by a specific group of users
  - for a particular activity
  - in a specified environment or work context
- Informal feedback … … controlled lab experiments

What do you want to know? Why?
- What do users want?
- What problems do they experience?
- Formative -- meshed closely with design, guides the design process
- Summative -- judgments about the finished product

Reasons for doing evaluations
- Understanding the real world
  - How employed in workplace?
  - Better fit with work environment?
- Comparing designs
  - compare with competitors or among design options
- Engineering towards a target
  - x% of novice users should be able to print correctly on first try
- Checking conformance to a standard
  - screen legibility, etc.
When and how do you do evaluation?

- **Early to**
  - Predict usability of product or aspect of product
  - Check design team’s understanding of user requirements
  - Test out ideas quickly and informally

- **Later to**
  - Identify user difficulties / fine tune
  - Improve an upgrade of product

Case Study: 1984 Olympic Messaging System

- Voice mail for 10,000 athletes in LA -> was successful
- Kiosks place around village -- 12 languages
- Approach to design (user-centered design)
  - Printed scenarios of UI prepared, comments obtained from designers, management prospective users -> functions altered, dropped
  - Produced brief user guides, tested on Olympians, families & friends, 200+ iterations before final form decided
  - Early simulations constructed, tested with users -> need ‘undo’
  - Toured Olympic village sites, early demos, interviews with people involved in Olympics, ex-Olympian on the design team -> early prototype -> more iterations and testing

Case Study: 1984 Olympic Messaging System

- Approach to design (continued)
  - “Hallway” method: put prototype in hallway, collect opinions on height and layout from people who walk past
  - Try to destroy it method: CS students invited to test robustness by trying to “crash” it
- Principles of User-Centered Design:
  - Focus on users & tasks early in design process
  - Measure reactions using prototype manuals, interfaces, simulations
  - Design iteratively
  - Usability factors must evolve together

Case Study: Air Traffic Control

- CAA in the UK, 1991
- Original system -- data in variety of formats
  - Analog and digital dials
  - CCTV, paper, books
  - Some line of sight, others on desks or ceiling mountings outside view
- Goal: integrated display system, as much info as practical on common displays
- Major concern: safety

Air Traffic Control, continued

- Evaluate controller’s task
  - Want key info sources on one workstation (windspeed, direction, time, runway use, visual range, meteorological data, maps, special procedures)
- Develop first-cut design (London City airport, then Heathrow)
- Establish user-systems design group
- Concept testing / user feedback
  - Modify info requirements
  - Different layouts for different controllers and tasks
  - Greater use of color for exceptional situations and different lighting conditions
  - Ability to make own pages for specific local conditions
  - Simple editing facilities for rapid updates

ATC, continued

- Produce upgraded prototype
- “Road Show” to five airports
- Develop system specification
- Build and install system
  - Heathrow, 1989
  - Other airports, 1991
- Establish new needs
Case Study: Forte Travelodge

- System goal: more efficient central room booking
- IBM Usability Evaluation Centre, London
- Evaluation goals:
  - identify and eliminate problems before going live
  - avoid business difficulties during implementation
  - ensure system easy to use by inexperienced staff
  - develop improved training material and documentation

The Usability Lab

- Similar to TV studio: microphones, audio, video, one-way mirror
- set up to resemble Travelodge reception area, attempt to be non-threatening

Particular aspects of interest

- System navigation, speed of use
- screen design: ease of use, clarity, efficiency
- effectiveness of onscreen help and error messages
- complexity of keyboard for computer novices
- effectiveness of training program
- clarity and ease-of-use of documentation

Procedure

- Developed set of 15 common scenarios, enacted by cross-section of staff
- eight half-day sessions, several scenarios per session
- emphasize that evaluation is of system not staff
- video cameras operated by remote control
- debriefing sessions after each testing period, get info about problems and feelings about system and doc

Results:

- Operators and staff had received useful training
- 62 usability failures identified
- Priority given to:
  - speed of navigation through system
  - problems with titles and screen formats
  - operators unable to find key points in doc
  - need to redesign telephone headsets
  - uncomfortable furniture
- New system: higher productivity, low turnover, faster booking, greater customer satisfaction

Evaluation Methods

- Observing and monitoring usage
  - field or lab
  - observer takes notes / video
  - keystroke logging / interaction logging
- Collecting users’ opinions
  - interviews / surveys
- Experiments and benchmarking
  - semi-scientific approach (can’t control all variables, size of sample)
Methods, continued

- **Interpretive Evaluation**
  - informal, try not to disturb user; user participation common
  - includes participatory evaluation, contextual evaluation

- **Predictive Evaluation**
  - predict problems users will encounter without actually testing the system with the users
  - keystroke analysis or expert review based on specification, mock-up, low-level prototype

- Pilot Study for all types!! -- small study before main study to work out problems with experiment itself

- Human Subjects concerns --

Usage Data: Observations, Monitoring, User’s Opinions

- Observing users
- Verbal protocols
- Software logging
- Users’ opinions: Interviews and Questionnaires

Direct Observation

- **Difficulties:**
  - people "see what they want to see"
  - "Hawthorne effect" -- users aware that performance is monitored, altering behavior and performance levels
  - single pass / record of observation usually incomplete
  - Example: Eurochange system, new multimedia system in school

- Useful: early, looking for informal feedback, want to know the kinds of things that users do, what they like, what they don’t

- Know exactly what you’re looking for -> checklist/count
- Want permanent record: video, audio, or interaction logging

Eurochange System

- Machine that exchanges one form of European currency for another and also dispenses currency for credit/debit cards -- like an ATM machine
- Intended for installation in airports and railway stations
- Prototype machine installed in Oxford Street
- Your goal: find out how long average transaction takes; note any problems with user’s experience
- Problems you might experience???

New school multimedia system

- Being tried out by groups of 13 year olds
- Don’t interfere with children’s activities – note the kinds of things they do and the problems they encounter …
- What problems might you encounter?

Indirect Observation: Video recording

- Alleviates some difficulties of direct observation
- Can be synchronized with keystroke logging or interaction logging
- Problems:
  - effort required to synchronize multiple data sources
  - time required to analyze
  - users aware they’re being filmed
    - set up and leave for several days, they get used to it
Indirect Observation: Morae

- Recorder captures screen video, camera video and audio
- Observers can view in real time or post-event
- Observers can annotate captured data
- Additional features for analysis and reporting
- Created by local company: TechSmith
  - Other related software as well (Camtasia, UserVue, …)

Analyzing video data

- Task-based analysis
  - determine how users tackled tasks, where major difficulties lie, what can be done
- Performance-based analysis
  - obtain clearly defined performance measures from the data collected (frequency of task completion, task timing, use of commands, frequency of errors, time for cognitive tasks)
  - classification of errors
  - repeatability of study
  - time (5:1) -- tools can help

Verbal protocols

- User’s spoken observations, provides info on:
  - what user planned to do
  - user’s identification of menu names or icons for controlling the system
  - reactions when things go wrong, tone of voice, subjective feelings about activity
- “Think aloud protocol” -- user says out loud what he is thinking while working on a task or problem-solving
- Post-Event protocols -- users view videos of their actions and provide commentary on what they were trying to do

Software Logging

- Researcher need not be present
- part of data analysis process automated
- Time-stamped keypresses
- Interaction logging-- recording made in real time and can be replayed in real time so evaluator can see interaction as it happened
- Neal & Simons playback system -- researcher adds own comments to timestamped log
- Remaining problems: expense, volume

Interviews and Questionnaires

- Structured interviews
  - predetermined questions, asked in a set way
  - no exploration of individual attitudes
  - structure useful in comparing responses, claiming statistics
- Flexible interviews
  - some set topics, no set sequence
  - interviewer can follow replies
  - less formal, for requirements gathering

Interviews, continued

- Semistructured interview
  - set of questions available for interviewer to draw on if interviewee digresses or doesn’t say much
- Prompted interview
  - draw out more information from interviewee
    - based on screen design or prototype
    - or “… and what do you mean by …”
Example: semi-structured using checklist

- Why do you do this? (To get the user’s goal.)
- How do you do it? (To get the subtasks -- ask recursively for each subtask)
- Why not do it this way instead? (Mention alternative - in order to get rationale for choice of method actually used.)
- What are the preconditions for doing this?
- What are the results of doing this?
- May we see your work product?
- Do errors ever occur when doing this?
- How do you discover and correct these errors?

Variations on interviews

- Card sorting
  - users asked to group or classify cards to answer questions, answers recorded on data collection sheet
- Twenty questions
  - interviewer asks only yes/no questions

Interviews -- summary

- Focus is on style of presentation and flexibility of data gathering
- More structured -> easier to analyze
- Less structured -> richer information
- Good idea: transcribe interviews to permit detailed examination (also true for verbal protocols)

Questionnaires and surveys

- Focus is on preparation of unambiguous questions
- Again, pilot study important
- closed questions:
  - respondent selects from set of alternative replies
  - usually some form of rating scale
- open questions:
  - respondent free to provide own answer

Closed question - simple checklist

Can you use the following text editing commands?

<table>
<thead>
<tr>
<th></th>
<th>Yes</th>
<th>No</th>
<th>Maybe</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPLICATE</td>
<td>[   ]</td>
<td>[   ]</td>
<td>[   ]</td>
</tr>
<tr>
<td>PASTE</td>
<td>[   ]</td>
<td>[   ]</td>
<td>[   ]</td>
</tr>
</tbody>
</table>

Closed question -- six-point scale

Rate the usefulness of the DUPLICATE command on the following scale:

very useful | | | | | | | | | | of no use
Closed question - Likert scale

Computers can simplify complex problems

<table>
<thead>
<tr>
<th>strongly agree</th>
<th>agree</th>
<th>slightly agree</th>
<th>neutral</th>
<th>slightly disagree</th>
<th>disagree</th>
<th>strongly disagree</th>
</tr>
</thead>
</table>

Closed question - semantic differential

Rate the Beauxarts drawing package on the following dimensions:

<table>
<thead>
<tr>
<th>extremely easy</th>
<th>quite easy</th>
<th>neutral</th>
<th>slightly easy</th>
<th>quite clear</th>
<th>neutral</th>
<th>slightly clear</th>
<th>extremely clear</th>
<th>difficult</th>
<th>extremely fun</th>
<th>quite fun</th>
<th>neutral</th>
<th>slightly fun</th>
<th>extremely boring</th>
</tr>
</thead>
</table>

Closed question - ranked order

Place the following commands in order of usefulness (use a scale of 1 to 4 where 1 is the most useful)

___ PASTE
___ DUPLICATE
___ GROUP
___ CLEAR

Questionnaires

- Responses converted to numerical values
- Statistical analysis performed (mean, std_dev, SPSS often used if more statistical detail required)
- Increase chances of respondents completing and returning:
  - short
  - small fee or token
  - send copy of report
  - stamped, self-addressed envelope
- Pre- / post- questionnaires

Example: Eurochange questionnaire

- Eurochange.pdf
- Identify strengths and weaknesses.
- How could this be improved?

How to write a good survey

- Write a short questionnaire
  - what is essential to know? what would be useful to know? what would be unnecessary?
- Use simple words
  - Don’t: "What is the frequency of your automotive travel to your parents' residence in the last 30 days?"
  - Do: "About how many times have you driven to your parent's home in the last 30 days?"
How to write a good survey

Relax your grammar
- if the questions sound too formal.
- For example, the word “who” is appropriate in many instances when “whom” is technically correct.

Assure a common understanding
- Write questions that everyone will understand in the same way. Don’t assume that everyone has the same understanding of the facts or a common basis of knowledge. Identify even commonly used abbreviations to be certain that everyone understands.

How to write a good survey

Start with interesting questions
- Start the survey with questions that are likely to sound interesting and attract the respondents’ attention.
- Save the questions that might be difficult or threatening for later.
- Voicing questions in the third person can be less threatening than questions voiced in the second question.

Don’t write leading questions
- Leading questions demand a specific response. For example: the question “Which day of the month is best for the newly established company-wide monthly meeting?” leads respondents to pick a date without first determining if they even want another meeting.

Avoid double negatives
- Respondents can easily be confused deciphering the meaning of a question that uses two negative words.

Balance rating scales
- When the question requires respondents to use a rating scale, mediate the scale so that there is room for both extremes.

Don’t make the list of choices too long
- If the list of answer categories is long and unfamiliar, it is difficult for respondents to evaluate all of them. Keep the list of choices short.

Avoid difficult concepts
- Some questions involve concepts that are difficult for many people to understand.

Don’t make the list of choices too long
- If the list of answer categories is long and unfamiliar, it is difficult for respondents to evaluate all of them. Keep the list of choices short.

Avoid difficult recall questions
- People’s memories are increasingly unreliable as you ask them to recall events farther and farther back in time. You will get more accurate information from people if you ask about the recent past (past month) versus the more distant past (last year).

Use closed-ended questions rather than open-ended ones
- Closed-ended are useful because the respondents know clearly the purpose of the question and are limited to a set of choices where one answer is right for them. Easier to analyze.
- An open-ended question is a written response. For example: "If you do not want a company picnic, please explain why". Can provide new ideas/info.

Put your questions in a logical order
- The issues raised in one question can influence how people think about subsequent questions.
- It is good to ask a general question and then ask more specific questions.

Pre-test your survey
- First test to a small number of people.
- Then brainstorm with them to see if they had problems answering any questions. Have them explain what the question meant to them.
How to write a good survey

Name your survey
- If you send it out by email, it may be mistaken for "spam". Also want to pique the interest of the recipients.
- Here are examples of survey names that might be successful in getting attention:
  - Memo From the Chief Executive Officer
  - Evaluation of Services of the Benefits Office
  - Your Opinion About Financial Services
  - Free T-shirt Win a Trip to Paris
  - Please Respond By Friday
  - Free Subscription
  - Win a notebook computer
- But some of these look like spam to me. Proceed with caution.

Cover memo or introduction
- If sending by US mail or email, may still need to motivate recipient to complete it.
- A good cover memo or introduction should be short and includes:
  - Purpose of the survey
  - Why it is important to hear from the respondent
  - What may be done with the results and what possible impacts may occur with the results.
  - Address identification
  - Person to contact for questions about the survey
  - Due date for response

Your job now …
Design/revise your survey for P2B.
- p.s. – you need to include this in your submission for human subjects approval
Exchange
Critique
Refine …

Example Usability Study:
See: ApiDB Workshop Study
And online materials at:
- www.apidb.org/workshop

Interpretive Evaluation
Contextual inquiry
Cooperative and participative evaluation
Ethnography
- rather than emphasizing statement of goals, objective tests, research reports, instead emphasizes usefulness of findings to the people concerned
- good for feasibility study, design feedback, post-implementation review

Contextual Inquiry
Users and researchers participate to identify and understand usability problems within the normal working environment of the user
Differences from other methods include:
- work context -- larger tasks
- time context -- longer times
- motivational context -- more user control
- social context -- social support included that is normally lacking in experiments
Why use contextual inquiry?
- Usability issues located that go undetected in laboratory testing.
  - Line counting in word processing
  - unpacking and setting up equipment
- Issues identified by users or by user/evaluator

Contextual interview: topics of interest
- Structure and language used in work
- individual and group actions and intentions
- culture affecting the work
- explicit and implicit aspects of the work

Cooperative evaluation
- A technique to improve a user interface specification by detecting the possible usability problems in an early prototype or partial simulation
- low cost, little training needed
- think aloud protocols collected during evaluation

Cooperative Evaluation
- Typical user(s) recruited
- representative tasks selected
- user verbalizes problems/ evaluator makes notes
- debriefing sessions held
- Summarize and report back to design team

Participative Evaluation
- More open than cooperative evaluation
- subject to greater control by users
- cooperative prototyping, facilitated by
  - focus groups
  - designers work with users to prepare prototypes
  - stable prototypes provided, users evaluate
  - tight feedback loop with designers

Ethnography
- Standard practice in anthropology
- Researchers strive to immerse themselves in the situation they want to learn about
- Goal: understand the ‘real’ work situation
- typically applies video - videos viewed, reviewed, logged, analyzed, collections made, often placed in databases, retrieved, visualized ....
Predictive Evaluation

- Predict aspects of usage rather than observe and measure
- doesn’t involve users
- cheaper

Predictive Evaluation Methods

- Inspection Methods
  - Standards inspections
  - Consistency inspection
  - Heuristic evaluation
  - “Discount” usability evaluation
  - Walkthroughs
- Modelling: The keystroke level model

Standards inspections

- Standards experts inspect the interface for compliance with specified standards
- relatively little task knowledge required

Consistency inspections

- Teams of designers inspect a set of interfaces for a family of products
  - usually one designer from each project

Usage simulations

- Aka - “expert review”, “expert simulation”
- Experts simulate behavior of less-experienced users, try to anticipate usability problems
- more efficient than user trials
- prescriptive feedback

Heuristic evaluation

- Usage simulation in which system is evaluated against list of “heuristics”, e.g.
- Two passes: per screen, and flow from screen to screen
- Study: 5 evaluators found 75% of problems
Sample heuristics
- Use simple and natural dialogue
- Speak the user’s language
- Minimize user memory load
- Be consistent
- Provide feedback
- Provide clearly marked exits
- Provide shortcuts
- Provide good error messages
- Prevent errors

Discount usability engineering
- Phase 1: usability testing + scenario construction (1-3 users)
- Phase 2: scenarios refined + heuristic evaluation
- “Discount” features
  - Small scenarios, paper mockups
  - Informal think-aloud (no psychologists)
  - Scenarios + think-aloud + heuristic evaluation
  - Small number of heuristics (see previous slide)
  - 2-3 testers sufficient

Walkthroughs
- Goal: detect problems early on; remove
- Construct carefully designed tasks from a system specification or screen mockup
- Walk-through the activities required, predict how users would likely behave, determine problems they will encounter
- See checklist for cognitive walkthrough

Modeling: keystroke level model
- Goal: calculate task performance times for experienced users
- Requires
  - Specification of system functionality
  - Task analysis, breakdown of each task into its components

Keystroke-level modeling
- Time to execute sum of:
  - Tk - keystroking (0.35 sec)
  - Tp - pointing (1.10)
  - Td - drawing (problem-dependent)
  - Tm - mental (1.35)
  - Th - homing (0.4)
  - Tr - system response (1.2)

KLM: example
- Save file with new name in wp that uses mouse and pulldown menus
  - (1) initial homing (Tth)
  - (2) Move cursor to file menu at top of screen (Tp + Tm)
  - (3) Select “save as” in file menu (click on file menu, move down file menu, click on “save as”) (Tm + Th + Tk)
  - (4) Word processor prompts for new file name, user types filename (Tr + Tm + Tk(filename) + Tk)
Experiments and Benchmarking

- Traditional experiments
- Usability Engineering

Traditional Experiments

- Typically narrowly defined, evaluate particular aspects such as:
  - menu depth v. context
  - icon design
  - tickers v. fade_boxes v. replace_boxes
- Usually not practical to include in design process

Example: Star Workstation, text selection

- Goal: evaluate methods for selecting text, using 1-3 mouse buttons
- Operations:
  - Point (between characters, target of move, copy, or insert)
  - Select text (character, word, sentence, par, doc)
  - Extend selection to include more text

Selection Schemes

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button 2</td>
<td>C</td>
<td>C, W, S, P, D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>Adjust</td>
<td>Adjust</td>
</tr>
<tr>
<td>Button 3</td>
<td>W, S, P, D</td>
<td>D</td>
<td>Adjust</td>
<td>Adjust</td>
<td>Adjust</td>
<td>Adjust</td>
<td>Adjust</td>
</tr>
</tbody>
</table>

Methodology

- Between-subjects paradigm
- six groups, 4 subjects per group
- in each group: 2 experienced w/mouse, 2 not
- each subject first trained in use of mouse and in editing techniques in Star w.p. system
- Assigned scheme taught
- Each subject performs 10 text-editing tasks, 6 times each

Results: selection time

Time:

- Scheme A: 12.25 s
- Scheme B: 15.19 s
- Scheme C: 13.41 s
- Scheme D: 13.44 s
- Scheme E: 12.85 s
- Scheme F: 9.89 s (p < 0.001)
Results: Selection Errors

- Average: 1 selection error per four tasks
- 65% of errors were drawthrough errors, same across all selection schemes
- 20% of errors were “too many clicks”, schemes with less clicking better
- 15% of errors were ‘click wrong mouse button’, schemes with fewer buttons better

Selection scheme: test 2

- Results of test 1 lead to conclusion to avoid:
  - drawthroughs
  - three buttons
  - multiple clicking
- Scheme “G” introduced -- avoids drawthrough, uses only 2 buttons
- New test, but test groups were 3:1 experienced w/mouse to not

Results of test 2

- Mean selection time: 7.96s for scheme G, frequency of “too many clicks” stayed about the same
- Conclusion: scheme G acceptable
  - selection time shorter
  - advantage of quick selection balances moderate error rate of multi-clicking

Experimental design - concerns

- What to change? What to keep constant? What to measure?
- Hypothesis, stated in a way that can be tested.
- Statistical tests: which ones, why?

Variables

- Independent variable - the one the experimenter manipulates (input)
- Dependent variable - affected by the independent variable (output)
- experimental effect - changes in dependent caused by changes in independent
- confounded -- when dependent changes because of other variables (task order, learning, fatigue, etc.)

Selecting subjects - avoiding bias

- Age bias -- Cover target age range
- Gender bias -- equal numbers of male/female
- Experience bias -- similar level of experience with computers
- etc. ...
## Experimental Designs

- **Independent subject design**
  - single group of subjects allocated randomly to each of the experimental conditions

- **Matched subject design**
  - subjects matched in pairs, pairs allocated randomly to each of the experimental conditions

- **Repeated measures design**
  - all subjects appear in all experimental conditions
  - Concerns: order of tasks, learning effects

- **Single subject design**
  - in-depth experiments on just one subject

## Critical review of experimental procedure

- **User preparation**
  - adequate instructions and training?

- **Impact of variables**
  - how do changes in independent variables affect users

- **Structure of the tasks**
  - were tasks complex enough, did users know aim?

- **Time taken**
  - fatigue or boredom?

## Critical review of experimental results

- **Size of effect**
  - statistically significant? Practically significant?

- **Alternative interpretations**
  - other possible causes for results found?

- **Consistency between dependent variables**
  - task completion and error scores versus user preferences and learning scores

- **Generalization of results**
  - to other tasks, users, working environments?

## Usability Engineering

- **Usability of product specified quantitatively, and in advance**

- **As product is built, it can be demonstrated that it does or does not reach required levels of usability**

## Usability Engineering

- **Define usability goals through metrics**

- **Set planned levels of usability that need to be achieved**

- **Analyze the impact of various design solutions**

- **Incorporate user-defined feedback in product design**

- **Iterate through design-evaluate-design loop until planned levels are achieved**

## Metrics

- **Include:**
  - time to complete a particular task
  - number of errors
  - attitude ratings by users
### Metrics - example, conferencing system

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mastery eval.</th>
<th>Task learning</th>
<th>Initial use</th>
<th>Task learning</th>
<th>Initial use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual use</td>
<td>3 (high)</td>
<td>2 (somewhat)</td>
<td>3 (high)</td>
<td>2 (somewhat)</td>
<td>3 (high)</td>
</tr>
<tr>
<td>Expert use</td>
<td>5 (high)</td>
<td>4 (very)</td>
<td>5 (high)</td>
<td>4 (very)</td>
<td>5 (high)</td>
</tr>
<tr>
<td>Error recovery</td>
<td>None prefer</td>
<td>Same as Q</td>
<td>None prefer</td>
<td>Same as Q</td>
<td>None prefer</td>
</tr>
<tr>
<td>Preference over product A</td>
<td>'much' better</td>
<td>Second half better</td>
<td>Two halves equal</td>
<td>1st half vs. 2nd half score</td>
<td>1st half vs. 2nd half score</td>
</tr>
<tr>
<td>Mastery level</td>
<td>0 errors</td>
<td>50% better</td>
<td>0 errors</td>
<td>50% better</td>
<td>0 errors</td>
</tr>
<tr>
<td>Infreq. Use</td>
<td>8-10 successful interxns / 30 min</td>
<td>Carefully constructed standard tests used to monitor users' performance in usability testing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Benchmark tasks
- Carefully constructed standard tests used to monitor users' performance in usability testing
- Typically use multiple videos, keyboard logging
- Controlled testing -- specified set of users, well-specified tasks, controlled environment
- Tasks longer than scientific experiments, shorter than “real life”

### Making tradeoffs
- Impact analysis - used to establish priorities among usability attributes. It is a listing of attributes and proposed design decisions, and % impact of each.
- Usability engineering reported to produce a measurable improvement in usability of about 30%.