FORMATIVE EVALUATION OF USER INTERACTION: AFTER EVALUATION SESSION

Chapter 10 (d)

TOPICS:

• Analyzing the data
• Cost-importance table
• Drawing conclusions/Redesigning
• Connecting back to UE management
• Team exercise on formative evaluation

Copyright © 2001 H. Rex Hartson and Deborah Hix.

All rights reserved. No part of this material may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, microfilming, recording, or otherwise, without prior written permission of its authors.
AFTER EVALUATION SESSION: ANALYZING THE DATA

• Major decision: Accept as is or redesign
  Decision must be made at a global (metaphor) as well as detailed (individual problem) level

• Compare usability specifications to observed results
  How close is design to meeting specs?

• If usability specifications are not met, identify interaction design problems and solve in order of cost and effect on usability
  * In most situations, finding usability problems is bad, but in formative evaluation, finding usability problems is good.

• Structured identification of interaction design problems and potential solutions
COST-IMPORTANCE ANALYSIS

- Cost-importance table

* Best to do in a spreadsheet

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
</table>

* Problem — something that has an impact on usability; observed as user interacts
COST-IMPORTANCE ANALYSIS

- Cost-importance table (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Importance to fix — effect on usability (independent of cost), based on best engineering guess (include risk of not fixing)

- Importance = M: must fix, regardless

- Importance = 5: If interaction feature is mission critical or usability problem has major impact on task performance or user satisfaction (e.g., user cannot complete key task), expected to occur frequently, causes costly errors, or major source of dissatisfaction
COST-IMPORTANCE ANALYSIS

- Cost-importance table (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Importance = 3: If user can complete task, but with difficulty (e.g., caused confusion and required extra effort), or problem was source of dissatisfaction

- Importance = 1: If problem did not impact task performance or dissatisfaction much (e.g., irritant or cosmetic), but is still worth listing

Dilbert: stadium -->->->

16.5 Eval After
COST-IMPORTANCE ANALYSIS

• Adjustment factors for "Importance"

* Probability of occurrence

- Over all affected user classes, how often will users encounter this problem?

- Example: If task cannot be completed (i.e., importance = 5) but usability problem represents situation that will arise rarely and not critical task, downgrade importance to 4 or 3
COST-IMPORTANCE ANALYSIS

- Cost-importance table (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Solution(s) — proposed changes to solve problem
  - Design principles and guidelines
  - Brainstorming
  - Studying other similar designs
  - Solutions suggested by users and experts
  - Typically not a good solution: More training and/or more documentation

Adjust the design, not the user!
COST-IMPORTANCE ANALYSIS

• Cost-importance table (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td>3</td>
<td>Allow user to click in time slot, then press delete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Cost — resources (time, money) needed for each proposed solution

- Changes to paper prototypes are minimal cost
- Significance is in computer-based prototypes, versions of real product

* Remove all "must fix regardless of cost" problems from table

These are EXCEPTIONS; e.g., extreme cost of a "must fix" problem could cause difficulty in process. Exceptions can also be dictated by corporate policy, management decision, etc.
COST-IMPORTANCE ANALYSIS

- Cost-importance table (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td>3</td>
<td>Allow user to click in time slot, then press delete</td>
<td>4 hrs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Priority ratio — metric for sorting by priorities:

(important rating/cost in person-hours) * 1000

Intuitively, should be high number; hi imp., but low cost

<table>
<thead>
<tr>
<th>Problem</th>
<th>Imp.</th>
<th>Solution(s)</th>
<th>Cost</th>
<th>Prio. Ratio</th>
<th>Prio. Rank</th>
<th>Resolution</th>
<th>Cumulative cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>User didn't know to select appointment before deleting</td>
<td>3</td>
<td>Allow user to click in time slot, then press delete</td>
<td>4 hrs</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Priority rank — sort remaining problems by descending order of ratios to determine which will give biggest improvement for least cost

Fill in after compute all priority ratios
COST-IMPORTANCE ANALYSIS

- Resolution — final decision about each observed problem

* Group by type of problem to find common design solution to share cost when possible

* Add up figures for "cumulative cost" column in sorted table
  
  To get hi imp/lo cost first; sort from highest ratio to lowest

* Draw cutoff line where cumulative cost meets or exceeds resources available

* Compile three lists

  - Problems that definitely will be fixed, starting with "must fix" list

  - High priority rank (high importance/low cost) problems (down to the cutoff line) that will be fixed next, time permitting

  - Problems tabled (problems below the cutoff line)
CONNECTING BACK TO USABILITY ENGINEERING LIFE CYCLE

- Implement chosen design decisions
- Realization of benefits of evaluation
- May have to re-evaluate some solutions
- Management of decision-making
- Engineering: achievement of specs, not perfection
- Control of iterative process
- Knowing when to stop iterating: when usability specifications are met
CONNECTING BACK TO USABILITY ENGINEERING LIFE CYCLE

• Goal is cost effectiveness

• If schedule to first release is too tight for thorough formative evaluation in lab, use:
  * Usability inspection
  * Rapid usability analysis

  - Isolate most severe problems, "show stoppers"
  - Leave rest until next release

  Don't follow up on problems with lower severity ratings (but note for future)

• Usability engineering tools for classification, reporting, storing and retrieving and re-using usability problems, analyses, decisions, etc.

• Videoclip: Analysis of Envision evaluation sessions
TEAM EXERCISE: FORMATIVE EVALUATION

• *Goal:*
  * To perform very simple formative evaluation, data collection, and analysis

• *Activities:*

  **PREPARATION**

  * Assemble in teams

  * Decide roles for your team members:

    - A prototype *executor*, to move transparencies and provide feedback (person who knows design best and can "play computer")

    - An evaluation *facilitator*, to keep experiment moving, to interact with participants, and to record critical incidents (qualitative data)

    - A user performance *timer*, to time participants performing tasks and/or count errors (quantitative data)
TEAM EXERCISE: FORMATIVE EVALUATION

* Decide roles for your team members (continued):
  - Two participants, to trade to another team
  - Anyone else can record critical incidents

* Then, trade the two participants for two from another team, forming new teams. The traded participants are now permanently on their new team.

* Stay with your new team while we hand out and explain forms.

* As new teams, excuse both your participants from the room momentarily.

RUN EXPERIMENT

* Get your prototype “booted up”.

* Bring first participant into "lab", greet them, and explain evaluation session to them.
TEAM EXERCISE: FORMATIVE EVALUATION

* Have first participant use your prototype to perform the benchmark tasks for your objective usability specifications. That is, have participant read first task aloud, then perform that task while thinking aloud. (Don't count reading of task in task timing.)

* Next have participant read second task aloud and perform it while thinking aloud.

* Have this participant complete questionnaire, and then give them their "reward".

* Now have the second participant perform the tasks and complete the questionnaire. The first participant should stay and observe.
TEAM EXERCISE: FORMATIVE EVALUATION

• *Cautions and hints:*

* Team members must not coach participants as they perform tasks.