

CS3724 Human-computer Interaction

Formative Evaluation of User Interaction: After Evaluation Session

Copyright © 2005 H. Rex Hartson and Deborah Hix.

Topics



- Analyzing the data
- Cost-importance analysis
- Drawing conclusions
- Connecting back to UE management
- Team exercise on formative evaluation

2

Eval after

Analyzing the Data



- Major design:
Accept as is or redesign
- Compare observed results to usability specifications
 - Remember, it's formative, not summative (therefore, statistical significance is not at issue)
- If usability specifications are not met, identify interaction design problems and solve in order of cost and effect on usability

3

Eval after

Analyzing the Data



- 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

4 Eval after

Cost-Importance Analysis

- Cost-Importance table
 - Best to do in a spreadsheet
 - *Problem*
 - Something that has an impact on usability; observed as user interacts
 - Deduced from critical incident data



5 Eval after

Cost-Importance Analysis

- For Calendar example

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Reso lution
User didn't know to select appt before trying to delete							

6 Eval after

Cost-Importance Analysis

- *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), expect to occur frequently, causes costly errors, or major source of dissatisfaction

7

Eval after

Cost-Importance Analysis

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

8

Eval after

Cost-Importance Analysis

- Adjustment factors for "Importance"
 - Probability of occurrence
 - Over all affected user classes, how often will user encounter this problem?
 - Example: if task cannot be completed (e.g., Importance = 5) but usability problem represents situation that will arise rarely and not critical task, downgrade importance to 4 or 3
 - Example: if impact is moderately significant (3), but occurs frequently, upgrade to 4
 - Learnability
 - If users can learn to solve it immediately, it won't affect subsequent usage (reduce Importance by 1)

9

Eval after

Cost-Importance Analysis

- For Calendar example, assume an Importance = 3

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Reso lution
User didn't know to select appt before trying to delete	3						

10 Eval after

Cost-Importance Analysis

- *Solution(s)*--proposed changes to solve problem
 - Design principles and guidelines
 - Brainstorming
 - Study other similar designs
 - Solutions suggested by users and experts
 - A given problem can have more than one possible solution; list them each on a separate line
 - Typically not a good solution: More training or documentation (adjust the design, not the user!)

11 Eval after

Cost-Importance Analysis

- For Calendar example:

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Reso lution
User didn't know to select appt before trying to delete	3	Allow user to click in time slot then press delete; add pop-up hint 'Click in appointment to select for deletion'					

12 Eval after

Cost-Importance Analysis



- *Cost*--resources (time, money) needed for each proposed solution
 - Changes to paper prototype are minimal cost
 - Cost more significant in computer-based prototypes, versions of real product
 - Usually units of cost are in person-hours
 - Usually round up fractional values to next integer (this is an engineering estimate, anyway)
 - For problems with multiple possible solutions, give cost for each on separate line (presumably importance is same)

13 Eval after

Cost-Importance Analysis

- For Calendar, assume a cost of 4 person-hours for recoding this feature

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Reso lution
User didn't know to select appt before trying to delete	3	Allow user to click in time slot then press delete; add pop-up hint 'Click in appointment to select for deletion'	4				

14 Eval after

Cost-Importance Analysis

- Priority ratio – metric to establish priorities
 - = (importance/cost) * 1000
 - Intuitively, high priority means high importance, low cost

15 Eval after

Cost-Importance Analysis

- For Calendar example, priority = $(3/4) * 1000 = 750$

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Reso lution
User didn't know to select appt before trying to delete	3	Allow user to click in time slot then press delete; add pop-up hint 'Click in appointment to select for deletion'	4	750			

16 Eval after

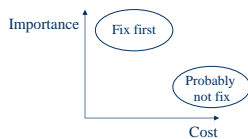
Cost-Importance Analysis

- Do priority ratios for all individual usability problems
- Group together related problems (that can share common solution)
 - Represent as single problem and recalculate importance, cost, priority ratio
 - Usually lower cost than total for individual problems
 - Often, for problem with more than one solution, the broadest solution (though more costly) groups better with others

17 Eval after

Cost-Importance Analysis

- Move all "must fix regardless of cost" problems to top of table
- Sort rest of table in descending order by priority ratio
 - This gets "must fix" problems first
 - The high importance, low cost problems are next
 - These are the problems that, when fixed, give biggest improvement in usability for least cost



18 Eval after

Drawing Conclusions



- Deal with “Must fix” problems
 - If enough resources, fix
 - Otherwise, someone (e.g., project manager) must decide
 - Sometimes fixing “must fix” problems means no resources left for any other problems
 - Extreme cost of a “must fix” problem could make it infeasible to fix in current version
 - Exceptions (with cost overruns) can be dictated by corporate policy, management decision, marketing, etc.
 - In our example, we have enough resources

22 Eval after

Drawing Conclusions

● Our resolution

Problem	Imp	Solution(s)	Cost	Prio ratio	Prio rank	Cum cost	Resolut ion
pqr	M	Blah blah	20	M	1	20	Fix
Abc	5	Yada yada	5	1000	2	25	Fix, if time
User didn't know to select appt before trying to delete	3	Allow user to click in time slot then press delete; add pop-up hint 'Click in appointment to select for deletion'	4	750	3	29	Table until next version
xyz	1	Sure, right	3	333	4	32	Probably never

23 Eval after

Drawing Conclusions

- Other considerations
 - Sometimes get ties on Priority Ratios
 - Must break ties by Importance, personal preference, etc.
 - Similarly, if Importance (which reflects severity, impact on user, etc.) is more significant factor in your environment, weight it a little more in the formula (for Priority Ratio)

24 Eval after

Connection Back to Usability Engineering Lifecycle

- Implement chosen design solutions
- Realize benefits of improved usability
- Cycle back through life cycle process and evaluate again
- Usability testing controls iterative process
 - Stop when achieve usability specifications
 - Goal is not perfection

Stop!

25 Eval after

Connection Back to Usability Engineering Lifecycle

- Cost-effectiveness
 - If schedule for first release is too tight for thorough testing in lab, use:
 - Usability inspection
 - Rapid usability analysis
 - Isolate most severe problems, "show stoppers"
 - Leave rest until next release
 - But, be careful not to let tight production schedule force you to release something that could embarrass your company
 - Quality is remembered long after schedules are forgotten!

26 Eval after

Team Exercise – Usability Data Analysis



- Goal: To perform the analysis part of a very simple formative usability evaluation
- Activities:
 - Assemble in same teams (including your new participants)
 - The team compiles results to determine whether usability specifications were met.

27 Eval after

Team Exercise – Usability Data Analysis

- Fill in the "Observed results" column on the usability specification table.
- Using cost-importance table, list 2 or 3 usability problems from critical incidents.
- Assign an "Importance" rating, 1 through 5, to some of the observed problems, based on severity.
- Propose "Solutions" (without doing all the work of re-design).

28 Eval after

Team Exercise – Usability Data Analysis

- Assign "Cost" values (in person-hours) to each solution. Values based on computer (not paper) prototype changes.
- Compute "Priority ratios".
- Group together related problems and list as single problem
- Move "Must fix" problems to the top

29 Eval after

Team Exercise – Usability Data Analysis

- Sort the rest by decreasing ratios to determine "Priority rank" of usability problems.
- Fill in cumulative cost column.
- Assume hypothetical value for available time resources (something to make this exercise work).
- Draw the cutoff line.
- Finalize your "management" decisions about which changes to make in the next version.

30 Eval after

Team Exercise – Usability Data Analysis

- Deliverables
 - Summary of quantitative results, written in "Observed results" column, on usability specification table on transparency (for comparison)
 - List of raw critical incidents
 - Cost-importance table, with "Resolutions" column completed in the form on transparency.
 - Choose someone to give brief report on evaluation results.

31 Eval after
