Usability Evaluation

• Any analysis or empirical study of the usability of a prototype or system
• Goal is to provide feedback in software development in support of an iterative development process
• Recognize problems, understand underlying causes, and plan changes

Goals of Usability Evaluation

Formative Evaluation: What and how to re-design?

“Cook tastes the soup”

Design → Construction
design

“Customer tastes the soup”

Summative Evaluation: How well did we do?

"Customer tastes the soup"
Formative and Summative Goals

Formative: *during* development, guides process
Summative: *after* development, or at a *checkpoint*

- What sorts of “test data” aid formative goals?
- What about summative?
- SBD relies on *mediated* usability evaluation
  - Claims analysis documents design features of concern
  - Users’ performance & reactions tied to these features, establishing a *usability specification*

Analytic and Empirical Methods

Analytic: theory, models, guidelines (*experts*)
Empirical: observations, surveys (*users*)

“If you want to evaluate a tool, say an axe, you might study the design of the bit, the weight distribution, the steel alloy used, the grade of hickory in the handle, etc., or you might just study the kind and speed of the cuts it makes in the hands of a good axeman."

- Which is more expensive? Why?
- Which carries more weight with developers? Why?
### Analytic Methods

- Usability Inspection
- Heuristic Evaluation
- Cognitive Walkthrough

### Usability Inspection

- Expert walkthrough based on usability guidelines, often working from a checklist
  - Generally want more than one expert (if affordable!)
- Guidelines (and walkthrough) can be at many levels
  - e.g., screen layout, detailed analysis of cognitive states
- May or may not use a standard set of tasks
  - Depends on how comparable you want judgments to be
- Summarize by listing problems identified in each category, also often rating them for severity
Guidelines

- Many published sources
  - Simple google search yields 10 pages of links
- “Official” sources
  - Yale style guide
  - Nielsen and Tahir, “Homepage Usability”

Heuristic Evaluation

- Use simple and natural language
- Speak the users’ language
- Minimize memory load
- Be consistent
- Provide feedback
- Provide clearly marked exits
- Provide shortcuts
- Provide good error messages
- Prevent errors
- Include good help and documentation

Multiple experts review with respect to these issues

Can also include different classes of stakeholders, e.g. developers, users
Large Screen Display Heuristics

- Use appropriate color schemes
- Layout should reflect information structure
- Judicious use of animation is necessary for effective design
- Use text banners only when necessary
- Show the presence of information, but not the details
- Cyclic displays are useful, but care must be taken in implementation
- Avoid the use of audio
- Eliminate or hide configurability controls

Why Heuristics?

- Take 3-5 experts and find 80% of the usability problems in a design
- Very fast, somewhat cheap
  - Especially if the experts are in house
- Very popular in industry
- Starts with common sense, but can really pinpoint details
Cognitive Walkthrough

- Form-based inspection of system/prototype
  - For general use systems (affordances, metaphors)
  - Careful task selection, answer questions at each step; e.g. How obvious is the next action? Must competing goals be ignored? Is knowledge assumed?

Cognitive Walkthrough

- Check-list approach attractive to practitioners
- Concerns with how to select the tasks for analysis, i.e. complexity/realism vs evaluation cost
- In practice, often can be used in more lightweight fashion, more of a “tuning” to issues
Excerpt From Cognitive Walkthrough Form

Step [B] Choosing the Next Correct Action:
[B.1] Correct Action: Describe the action that the user should take at this step.
[B.2] Knowledge Checkpoint: If you have assumed user knowledge or experience, update the USER ASSUMPTION FORM.
[B.3] System State Checkpoint: If the system state may influence the user, update the SYSTEM STATE FORM.
[B.4] Action Availability: Is it obvious to the user that this action is a possible choice here? If not, indicate why.
[B.5] Action Identifiability:
[B.5.a] Identifier Location, Type, Wording, and Meaning:
______ No identifier is provided. (Skip to subpart [B.5.d])
Identifier type: Label Prompt Description Other (Explain)
Identifier wording: ____________________________
Is the identifier’s location obvious? If not, indicate why.
[B.5.b] Link Between Identifier and Action: Is the identifier clearly linked with this action? If not, indicate why.
[B.5.c] Link Between Identifier and Goal: Is the identifier clearly linked with an active goal? If not indicate why.

GOMS Analysis

- Build predictive model using scientific knowledge about human memory and behavior
  - like HTA, can analyze for complexity, consistency
  - or build computational version, to estimate task times for different design alternatives
  - if successful, can provide huge benefit...why?
- Extends general techniques of HTA
  - goals, sub-goals, plans, actions
  - BUT adds model elements for mental activities such as goal creation and selection, memory retrieval, etc.
Downsides of Analytic Methods

- Usability inspections are rapid, relatively cheap
  - But may miss details only seen in realistic use contexts involving real users, say little about what caused the problems, or expected impact

- Model-based approaches have good scientific foundation, are credible, can be very powerful
  - But current theories have limited scope, and developing the models takes time/expertise

“Discount” Usability Evaluation

- Goal: get the most useful information for guiding re-design with the least cost
  - Pioneered by Jacob Nielsen (heuristic inspection)
- Do a little bit of each (analytic and empirical)
  - 3-4 experts find most of the guidelines issues
  - 4-6 users experience most of the actual use problems
  - Between the two, get a good sense of what to fix
- Not surprisingly, a popular strategy, pretty much what you find in practice
Running an Analytic Evaluation

- Overall organization
  - Who to test and how to coordinate?
  - Where to test?
  - How many to test at the same time?
  - How long should it take?
  - Do you need informed consent forms?

- Demo the prototype, concerns:
  - What functionality to demonstrate?
  - What fidelity to use?
  - How to ensure all evaluators understand the same things about the interface?

(Continued)

Running an Analytic Evaluation

- Obtaining evaluation results
  - How to balance participants using each evaluation type (project groups, gender, HCI experience)?
  - What types of results will heuristics and survey each provide?
  - What level of detail do you need & how can you ensure you’ll participants will provide this?

- Analyzing results
  - How will you compare the results you obtain with each method?
  - What will be the follow-on actions?
Results: Summarizing User Data

- How you summarize depends on variable type:
  - **Categorical**: responses are classified into groups
  - **Ordinal**: responses fall in groups, but natural order
  - **Interval**: a scale with equidistant values
  - **Ratio**: numerical scale with defined zero value
  - **Qualitative**: comments to organize and discuss

- Examples of each?
- What are appropriate summary treatments of these differing kinds of variables?

Statistics

- **t-test**
  - Compares 1 dep var on 2 treatments of 1 ind var

- **ANOVA**: ANalysis Of VAriance
  - Compares 1 dep var on n treatments of m ind vars

- Result: “significant difference” between treatments?
  - \( p = \) significance level (confidence)
  - typical cut-off: \( p < 0.05 \)
Statistics in Microsoft Excel

- Enter data into a spreadsheet
- Go to Tools..., Data Analysis... (may need to choose Analysis Toolpak from Addins first)
- Select appropriate analysis

---

t-tests in Excel

- Used to compare two groups of data
- Most common is “t-test: two-sample assuming equal variances”
- Other t-tests:
  - Paired two-sample for means
  - Two-sample assuming unequal variances
ANOVA in Excel

- Allows for more than two groups of data to be compared
- Most common is “ANOVA: Single factor analysis”
- Other ANOVAs:
  - ANOVA: Two-factor with replication
  - ANOVA: Two-factor without replication

\[ p < 0.05 \]

- Found a “statistically significant difference”
- Averages determine which is ‘better’
- Conclusion:
  - Vis Tool has an “effect” on user performance for task1
  - PerspWall better user performance than Lifelines for task1
  - “95% confident that PerspWall better than Lifelines”
  - Not “PerspWall beats Lifelines 95% of time”
- Found a counterexample to the null hypothesis
  - Null hypothesis: Lifelines = PerspWall
  - Hence: Lifelines \(\neq\) PerspWall
p > 0.05

- Hence, same?
  - Vis Tool has no effect on user performance for task1?
  - Lifelines = PerspWall ?

- Be careful!
  - We did not detect a difference, but could still be different
  - Did not find a counter-example to null hypothesis
  - Provides evidence for Lifelines = PerspWall, but not proof
  - Boring! Basically found nothing

- How?
  - Not enough users (other tests can verify this)
  - Need better tasks, data, ...

---

Reporting Results

- Often considered the most important section of professional papers
- Statistics NOT the most important part of the results section
- Statistics used to back up differences described in a figure or table
Reporting Means, SDs, t-tests

- Give means and standard deviations, then t-test

  - ... the mean number was significantly greater in condition 1 (M=9.13, SD=2.52) than in condition 2 (M=5.66, SD=3.01), t(44)=3.45, p=.01

What Are Those Numbers?

- ... the mean number was significantly greater in condition 1 (M=9.13, SD=2.52) than in condition 2 (M=5.66, SD=3.01), t(44)=3.45, p=.01
  - M is the mean
  - SD is the standard deviation
  - t is the t stat
  - the number in parentheses is the degrees of freedom (df)
  - p is the probability the difference occurred by chance
Reporting ANOVAs

• ... for the three conditions,
  \[ F(2,52)=17.24, \text{ MSE}=4528.75, \text{ p}<.001 \]
  – \( F(x,y) \) -- F value for \( x \) between groups and \( y \) within groups degrees of freedom (df)
  – MSE -- mean square error for the between groups condition
  – p -- probability that difference occurred by chance

Making Sense of the Results

• Relate to high-level goals: is the system useful, easy to learn and use, satisfying?
  – Which of these is hardest to judge in lab study?
• But also compare directly to usability specs:
  – Did you miss, meet, or surpass the target level?
  – More importantly, can you figure out why?
• Guidance on how to change design comes from the details of the testing, not the summary values
  – Why was user confused (or not), why was an interaction difficult or awkward, etc.
Usability Specifications

• Quality objectives for final system usability
  – like any specification, must be precise
  – managed in parallel with other design specifications

• In SBD, these come from scenarios & claims
  – scenarios are analyzed as series of critical subtasks
  – reflect issues raised and tracked through claims analysis
  – each subtask has one or more measurable outcomes
  – tested repeatedly in development to assess how well
    project is doing (summative) as well as to direct design
    effort toward problem areas (formative)

• Precise specification, but in a context of use

Activity, information, and interaction claims:
identify key design features to be tested

Design scenarios:
extract motivation and context for subtasks to be tested

Usability specifications:
a list of subtasks with performance and satisfaction parameters

Estimates of behavior:
published or pilot data of expected user behavior
What about Generality?

• Salient risk in focusing only on design scenarios
  – may optimize for these usage situations
  – the “successful” quality measures then reflect this

• When possible, add contrasting scenarios
  – overlapping subtasks, but different user situations
    (user category, background, motivation)
  – assess performance satisfaction across scenarios

• Motivation to construct functional prototypes as early as feasible in development cycle

A Sample Usability Specification

<table>
<thead>
<tr>
<th>Scenario &amp; Subtasks</th>
<th>Worst Case</th>
<th>Planned</th>
<th>Best Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction Scenario: Mr. King coaches Sally</td>
<td>2.5 on usefulness, ease of use, and satisfaction</td>
<td>4 on usefulness, ease of use, and satisfaction</td>
<td>5 on usefulness, ease of use, and satisfaction</td>
</tr>
<tr>
<td>1. Identify Sally’s view and synchronize</td>
<td>1 minutes, 1 error 3 on confusion</td>
<td>30 seconds, 0 error 2 on confusion</td>
<td>10 seconds, 0 error 1 on confusion</td>
</tr>
<tr>
<td>2. Upload desktop file from the PC</td>
<td>3 minutes, 2 errors 3 on familiarity</td>
<td>1 minute, 1 error 4 on familiarity</td>
<td>30 seconds, 0 error 5 on familiarity</td>
</tr>
<tr>
<td>3. Open, modify, try to save Excel file</td>
<td>2 minutes, 1 error 3 on confidence</td>
<td>1 minute, 0 errors 4.5 on confidence</td>
<td>30 seconds, 0 error 5 on confidence</td>
</tr>
<tr>
<td>4. Create nested exhibit component</td>
<td>5 minutes, 3 errors 3 on complexity</td>
<td>1 minute, 1 error 2 on complexity</td>
<td>30 seconds, 0 error 1 on complexity</td>
</tr>
</tbody>
</table>

• Where do targets come from? Serious, but not absolute
• Notice that we can also “test” overarching scenario