Evaluation (cont.):

Empirical Studies

CS352
Announcements

• Notice upcoming due dates (web page).

• Where we are in PRICPE:
  – Predispositions: Did this in Project Proposal.
  – RI: Research was studying users. Hopefully led to Insights.
  – CP: Concept and initial (very low-fi) Prototypes.
  – Evaluate throughout, repeat iteratively!!
Evaluation

• Analytical – based on your head
• Empirical – based on data
• Advantages/disadvantages of empirical
  – More expensive (time, $) to do.
  + Greater accuracy in what users REALLY do.
  + You’ll get more surprises.
  + Greater credibility with your bosses.
Empirical Work with Humans

• What do you want from it?
  – List of problems:
    • Usability study. (eg, 5 users in a lab).
  – List of behaviors/strategies/needs:
    • Think-aloud study or field observation.
  – Compare, boolean outcome (eg, A>B)
    • Statistical study.

• Note:
  – Impossible to “prove” no problems exist.
    • Can only find problems.
The Usability Study

• Returns a list of UI problems.
• Pick a task and user profile.
• Users do the task with your prototype
  – in here: using your CogTool-generated html
Usability Study: How

• How many: 5 users find 60-80% problems.
• How:
  – Be organized!! Have everything ready, ...
  – Test it first.
  – Users do task (one user at a time).
• Data IS the result (no stats needed).
Usability Study: How (cont.)

- Q: What, exactly, do you have to do during planning?
- A: Try to figure this out from the Steve Krug example:
  - Reminder of where we were in “3. The Home Page Tour” (7:00-7:45).
  - Doing a task “4. The Tasks” (8:13-11:50, 12:00-13:15)
- Discussion of Steve’s planning to do this.
Many Cheap Usability Studies: Steve Krug’s Recommendations

• Steve Krug recommends:
  – How often: one morning/month.
  – When: Continually, thru-out the dev. process.
  – # participants: 3 (to make once/month viable).
  – Participants: better to relax your standards about getting the “perfect” representative than to sacrifice the once/month frequency.
  – Where: in your lab, with a one-way mirror so that the dev team can watch.
    • Not in same room as participant – why?
Steve Krug’s recommendations (cont.)

– Who watches when: as many of the dev team as can come, for as long as they can stay.
– Who identifies the problems when: Everyone watching, while they watch.
  • Then at lunch, debrief and make decisions.
– Reporting: a brief email to the dev team summarizing the decisions made.
How to test sketches/concepts

• To spot problems with the concept.
• Like the “home page tour” from Steve Krug’s demo video.
• Can often be done in about 5-10 minutes.
How to test paper prototypes

• Here you need the user to do a task, as in the “wizard of oz” Excel demonstration.
• Usually 15-45 minutes will get you everything you need.
How to test a “working” prototype

• Such as your CogTool prototypes.
• Get it into executable form.
  – CogTool: export to html.
• Then do what Steve did:
  – the warm-up,
  – then present the task,
  – then do as in the part of the Steve Krug video we just watched.
# Things to Say

<table>
<thead>
<tr>
<th>When user does this:</th>
<th>Say this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Something surprises them (&quot;Oh!!&quot;)</td>
<td>“Is that what you expected to happen?”</td>
</tr>
<tr>
<td>Participant tries to get you to give them clues.</td>
<td>“What would you do if you were at home?” (Wait.) “Then why don’t you go ahead and try that?”</td>
</tr>
<tr>
<td></td>
<td>“I’d like you to do whatever you’d normally do.”</td>
</tr>
<tr>
<td></td>
<td>“How do you think it would work?”</td>
</tr>
<tr>
<td>Participants says something but you don’t know what triggered it.</td>
<td>“Was there something in particular that made you think that?”</td>
</tr>
<tr>
<td>Participant suggest that s/he’s not giving you what you need.</td>
<td>“No, this is very helpful.”</td>
</tr>
</tbody>
</table>
About the “think-aloud” part...

- Steve recommends having the participant “think aloud”, and this is a good idea.
Think-Aloud

- Most helpful with a working prototype or system,
  - but can also use it for early prototypes/concepts.
  - can be part of a usability study (eg, Steve Krug) or for more Research-y purposes.
- Returns a list of behaviors$strategies/impressions/thoughts...
- Pick a task and user profile.
- Users do the task with your prototype.
Think-Aloud: How

• How many: 5-10 users is usual in a research study, fewer if usability study.
  – If a research study, data analysis is time-consuming.
  – In here: 1-2 max!

• How:
  – Be organized!! Have everything ready, ...
  – Test it first.
  – Users do task (one user at a time).
Think-Aloud Training (Activity)

– “We are interested in what you say to yourself as you do the task. So we’ll ask you to TALK ALOUD CONSTANTLY as you work on the tasks. Say aloud EVERYTHING you say to yourself. Let’s practice talking aloud.”

– Add 234 + 456 in your head (aloud).
  • Discuss

– How many windows are in your parents’ house? (aloud).
  • Discuss

– What states can you ski in that start with “A”?
Think-Aloud Research Studies

• Analyze data for patterns, surprises, etc.
• No stats: not enough subjects for this.
• Sample think-aloud results:
  – From VL/HCC'03 (Prabhakaran Rao et al.)
Statistical Studies

• We will not do these in this class, but
  – you need to know some basics.
• Goal: answer a binary question.
  – eg: does system X help users create animations?
  – eg: are people better debuggers using X than using Y?
• Advantage: your audience believes it.
• Disadvantage: you might not find out enough about “why or why not”.
Hypotheses Testing

• Need to be specific, and provable/refutable.
  – eg: “users will debug better using system X than in system Y”
  – (Strictly speaking we use the “null” hypothesis, which says there won’t be a difference, but it’s a fine point...)
  – Pick a significance value (rule of thumb: 0.05).
    • If you get a p-value <=0.05, this says you’ve shown a significant difference, but there’s a 5% chance that the difference is a fluke.
Design the Experiment

• Identify outputs (dependent variables) for the hypotheses:
  – eg: more bugs fixed?
  – eg: fewer minutes to fix same number of bugs?

• Identify independent variables (“treatments”) we’ll manipulate:
  – eg: which system they use, X vs. Y?
Design the experiment (cont.)

• Decide on within vs. between subject.
  – “Within”: 1 group experiences all treatments.
    • In random order.
    • “Within” is best, if possible. (Why?)
  – “Between”: different group for each treatment.

• How many subjects?
  – Rule of thumb: 30/treatment.
  – More subjects -> more statistical power ->
    more likely to get p<=.05 if there really is a
difference.
Design the experiment (cont.)

- Design the task they will do.
  - Since you usually run a lot of these at once and you’re comparing them, you need to be careful with length.
    - Long enough to get over learning curve.
    - Big enough to be convincing.
    - Small enough to be do-able in the amount of time subjects have to spend with you.
  - Vary the order if multiple tasks.
Design the experiment (cont.)

• Develop the tutorial.
  – Practice it like crazy! (Must work the same for everyone!)

• Plan the data to gather.
  – Log files?
  – Questionnaires before/after?
  – Saved result files at end?
Designing the experiment (cont.)

• Water in the beer:
  – Sources of uncontrolled variation spoil your statistical power.

• Sources:
  – Too much variation in subject background.
  – Not a good enough tutorial.
  – Task not a good match for what you wanted to find out.
  – Etc.

• Result: no significant difference.
Finally, Analyze the Data

• Choose an appropriate statistical test. (There are entire courses on this.)
• Run it.
• Hope for $p \leq .05$.
• Summary:
  – Statistical studies are a lot of work, too much to do in this class!
  – Right choice for answering $X > Y$ questions.
Summary: For your project

• In here, you’ll be doing the Usability Study, with Think-aloud.
• 1-3 users.
• Your plan for doing this is part of what’s due on Friday!
  – To get you ready for actually doing the study the following week.