Identifying Critical Incidents for Large Scale Usability Analysis

David Akers
dakers@stanford.edu
Google SketchUp... Before
Google SketchUp... Now
Software Demo
Critical Incidents in HCI

“A negative critical incident is any event that causes errors, dissatisfaction, or negatively impacts effort or task performance.”

[Castillo 1997]
Traditional Usability Testing

- Identifying critical incidents
- Collecting additional detail on symptoms
- Reasoning from symptoms to causes
The Long Tail of Usability

problems often already known to the design team

problems often unknown to the design team
Broad Research Questions

How can we detect and characterize critical incidents in applications like SketchUp without requiring a usability expert’s attention?

... and ...

How would this compare to traditional usability testing?
Detecting Critical Incidents

A: Electrodes attached to chest

B: Electrocardiogram strip

C: Recording device
## Obvious Tradeoffs

<table>
<thead>
<tr>
<th>Self reporting</th>
<th>Event-based reporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Don’t need hypotheses or software instrumentation</td>
<td>+ Does not interfere with work</td>
</tr>
<tr>
<td>+ Very few “false positives”</td>
<td>+ Relies on objective judgments</td>
</tr>
<tr>
<td></td>
<td>+ Requires no training</td>
</tr>
<tr>
<td></td>
<td>+ Can identify problems unrecognized by the user</td>
</tr>
</tbody>
</table>
Environments for Usability Testing

Field or lab study
- Small scale
- High compensation
- Short duration
- Privacy not an issue
- Tasks usually provided

Instrumented panel
- Medium scale
- Some compensation
- Variable duration
- Privacy a minor issue
- Tasks sometimes provided

Real-world use
- Large scale
- No compensation
- Long duration
- Privacy a major issue
- Tasks not provided

Event-based reporting
Self-reporting
Self Reporting
Self Reporting Implementations

- Safari
- Bugzilla
- Others?
Self Reporting

“You can recognize that you are experiencing a negative critical incident when you are feeling confused, annoyed, fatigued, or frustrated.”

[Castillo 1997]
Report a Negative Critical Incident

Instructions
- Answer each of the following questions
- When you have completed the report, press the SUBMIT button
- Use this form to report ONE critical incident
- If you experience multiple critical incidents for a task, please file a separate report for each one
- If you decide not to submit the report you can return to the main reporting page

TASK DESCRIPTION

What was your overall objective?
What was the purpose of your task? What generally were you trying to do?
For example: add a footnote, insert a page number

What part of the interface were you using?
What menu, or window, or dialog box were you using?
For example: the main window, the reminder window, the file dialog box

How were you carrying out your task?
Please give a concise description of what you were doing, but complete enough that someone else could recreate the task context
- What was the sequence of actions you performed?
- What equipment did you use? (keyboard, mouse)
- What buttons, menus, pull-down items, etc. did you use?

CRITICAL INCIDENT DESCRIPTION

Describe what happened
What was the critical incident?
What was the feature or aspect of the interface that caused the critical incident?
During what part of the task did the critical incident occur?

How was your performance affected?
Some suggestions for information you might want to provide in this box:
What aspects of your performance were affected (e.g., speed, accuracy, ease, comprehension)
How were those aspects affected?
Why did this feature fail to meet your expectation or
Why do you consider it a poorly designed part of the program?

How did this make you feel?
Frustrated, confused, iritated, limited in productivity, physically fatigued, or any other adjectives or phrases that describe your reaction to the incident.

How did you recover from this critical incident?
Were you able to recover and complete your task?
What actions did you take to resolve or compensate for the problem?
Why did you choose those actions?

Rate the severity of the critical incident
- Unanswerable - can’t or don’t want to use this feature because of the way the software has been designed and implemented
- Severe - will probably continue to use this feature, but will be severely limited in my ability to do so. Will have great difficulty
  in circumventing the problem.
- Moderate - Will be able to use the program in most cases, but will have to undertake some moderate effort in getting
  around the problem.
- Minor - The problem occurs only intermittently, can be circumvented easily, or is dependent on a problem that is outside
  the product’s boundaries. Could also be a cosmetic problem.
Pilot Study

Embarrassed (12/15): “I felt self conscious about admitting my mistakes.”

Polite (8/15): “It felt kind of like pressing a flight attendant call button.”

Unaware (7/15): “When I was busy, I forgot about the button.”

Unmotivated (6/15): “I was more interested in completing the task.”

Unqualified (4/15): “I didn’t report problems unless I understood the cause.”

Calm (4/15): “I just didn’t get frustrated enough to press the button.”
Event-Based Reporting
Approach 1: Capture Everything
Approach 2: Hypothesize Behavior

Goal: Detect cases when developers’ expectations do not match users’ expectations.

Hilbert et al. 1997
But... Many Uses
And... Many Ways To Use
Approach 3: Look for Symptoms

On-line help is invoked
UNDO action invoked
Error message triggered
Warning message triggered
An action has no effect
DELETE invoked
Cancel button

Swallow et al. 1997
SketchUp Tasks
<table>
<thead>
<tr>
<th>Tool</th>
<th>Number of Tool Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencil</td>
<td>1200</td>
</tr>
<tr>
<td>Erase</td>
<td>900</td>
</tr>
<tr>
<td>Move/Copy</td>
<td>800</td>
</tr>
<tr>
<td>Rectangle</td>
<td>700</td>
</tr>
<tr>
<td>Push/Pull</td>
<td>600</td>
</tr>
<tr>
<td>RotateObject</td>
<td>500</td>
</tr>
<tr>
<td>Measure</td>
<td>400</td>
</tr>
<tr>
<td>Paint</td>
<td>200</td>
</tr>
<tr>
<td>Paste</td>
<td>100</td>
</tr>
<tr>
<td>Dimension</td>
<td>100</td>
</tr>
<tr>
<td>Arc</td>
<td>50</td>
</tr>
<tr>
<td>FollowMe</td>
<td>10</td>
</tr>
<tr>
<td>Offset</td>
<td>5</td>
</tr>
<tr>
<td>Circle</td>
<td>2</td>
</tr>
</tbody>
</table>
Results (Push/Pull)

Undos of Push/Pull are caused by:

- 2% exploration in SketchUp
- 20% known problems in SketchUp
- 67% previously unknown problems in SketchUp
- 11% we’ll never know!
The Long Tail of Usability

- Problems often already known to the design team
- Problems often unknown to the design team
Research Question

What types of usability problems does each technique reveal?

Self-reporting
- usability testing
  - button presses, screen capture, commentary
- symptom descriptions
  - usability expertise, domain expertise
- problem descriptions

Event-based reporting
- usability testing
  - log events, screen capture, commentary
- symptom descriptions
  - usability expertise, domain expertise
- problem descriptions

Traditional lab testing
- usability testing
  - expert observation, video, eye-tracking, probing questions
- symptom descriptions
  - usability expertise, domain expertise
- problem descriptions
Tasks
User Commentary

1. Please describe the events that led you to [undo/erase/self-report]. Focus your answer on recounting a “play-by-play” of what you were thinking and doing at the time. If you can’t remember, just say so and move on to the next episode.

2. During the episode, did the behavior of SketchUp surprise you? If yes, explain the difference between your expectations and what actually happened.

3. Did you find a way around the issue? If so, what did you do to get around it?
User Commentary (Erase/Undo)

4. Did you report this as an issue?

5. If you did not report this as an issue, why do you think that you didn’t?
Problem Typologies

Problem severity is a combination of three factors:

The frequency with which the problem occurs
The impact of the problem if it occurs
The persistence of the problem

Nielsen 1994
User Interaction Cycle

PLANNING (Determining what to do)

ASSSESSMENT (Determining, via feedback, if outcome was favorable)

TRANSLATION (Determining how to do it with physical actions)

PHYSICAL ACTION (Doing it)

See, Think

See, Think

See, Do

Feedback (visible representation of outcome)

Outcome: State change, internal to system, invisible to user

Andre et al. 2001
Discussion

Other problem typologies that might be useful?
How to encourage better retrospective think-aloud commentary?