Questions?
Overview of Today’s Lecture

Analytical Evaluation

Inspections

Performance modelling
Analytical Evaluations

Evaluations without involving users
Analytical Evaluations: Types

Inspections
  - Heuristic evaluation
  - Cognitive walkthrough

Performance modelling
  - Fitts’s Law
  - Hick-Hyman Law
  - Keystroke Level Modelling
  - GOMS
Inspection (aka “Asking Experts”)

User sometimes difficult to find or too expensive to get involved

Alternative would be to ask a group of experts to do the evaluation

Experts use their knowledge of users & technology to review software usability

Expert critiques (crits) can be formal or informal reports

Two flavors:

- Walkthroughs involve stepping through a pre-planned scenario noting potential problems
- Heuristic evaluation is a review guided by a set of heuristics
Cognitive Walkthroughs

“Cognitive walkthroughs simulate a user’s problem-solving process at each step in the human-computer dialog, checking to see if the user’s goals and memory for actions can be assumed to lead to the next correct action”
Cognitive Walkthroughs: Process

Characteristics of typical users are identified and sample tasks are developed

assumptions about user population, context of use, task details; clear sequence of actions needed

A re-use of user profiles, scenarios and task decompositions developed as part of the “Investigate” phase

One or more experts walk through the prototype with the scenario
Process: 3 questions

Experts place task actions in context of a scenario and answer the following:

Will the user know the correct action?
  i.e. will the user know what to do to achieve the task?

Will the user notice how to do the correct action?
  Can users see the button or menu item that they should use for the next action?
  Is it apparent when it is needed?

Will the user associate and interpret the response from the action correctly?
  Will users know from the feedback that they have made a correct or incorrect choice of action?
Cognitive walkthroughs: Process

As the experts work through the scenario, information is recorded on:

- Assumptions about what would cause problems and why
  - involves explaining why users would face difficulties
- Notes about side issues and design changes

Results are summarized

Eventually design is revised to fix problems identified
ex: find a book at Amazon.ca via search

**task**: to buy a copy of a textbook from amazon.ca using search

**typical users**: students who use the web regularly

step 1. find the book using search

Q: will users know what to do?
A: Yes – they know that they must find “books”

Q: will users see how to do it?
A: Yes – they have seen menus before and will know to look through them and select the appropriate items.

Q: will users understand from feedback whether the action was correct or not?
A: Yes – their action visually changes search ALL to search BOOKS.
A: Yes – their action brings them to a BOOKS page

Q: will users understand how to use the search, what to type in?
A: Yes – searching is common, and they know the title or author of the book

step 2. purchase the book

...
ex: find a book at Amazon.ca via navigation

**task:** to buy a copy of a textbook from amazon.ca using the menus

**typical users:** students who use the web regularly

1. step 1. find the book using navigation

Q: will users know what to do?
A: Yes – they know that they must find “books” somewhere

Q: will users see how to do it?
A: Yes – they have seen web links and understand menus and sections on a website

Q: will users understand from feedback whether the action was correct or not?
A: Yes – their action brings them to a textbook search page
A: Yes – their action brings them to a BOOKS page (repeat questions)

Q: will users know which category their book belongs to?
A: Yes – they are taking a course with such a title
A: No – some may be confusing, such as accounting vs business and finance

...
Heuristic Evaluation

Systematic inspection of an interface design to see if an interface complies with a set of usability heuristics, or usability guidelines.

General process:

» 3-5 inspectors (usability engineers, experts...)

» inspect interface in isolation (~1-2 hr for simple interfaces)

» results are aggregated afterwards
   single evaluator catches ~35% usability problems
   5 evaluators catch ~75%
Heuristic Evaluation

Proportion of Usability Problems Found

Number of Evaluators

- 25%
- 50%
- 75%
- 100%

- 5
- 10
- 15

25% 50% 75% 100%
Heuristic Evaluation: Background

Developed by Jacob Nielsen in the early 1990s

Based on heuristics distilled from an empirical analysis of 249 usability problems

These heuristics have been revised for current technology

Heuristics still needed for some emerging technologies (e.g., mobile devices, AR, etc).

Design guidelines form a basis for developing heuristics
Nielsen’s heuristics

Visibility of system status - Are users kept informed at all times?
Match between system and real world - Is the UI language simple?
User control and freedom - Are there easy escapes from unexpected locations?
Consistency and standards - Is performing similar action consistent?
Help users recognize, diagnose, recover from errors - Are error messages helpful?
Error prevention - Is it easy to make errors?
Recognition rather than recall - Are objects, actions and options always visible?
Flexibility and efficiency of use - Are there accelerators?
Aesthetic and minimalist design - Is any unnecessary and irrelevant information provided?
Help and documentation - Is help provided that can be easily searched?
Nielsen’s heuristics

Could be too general and might need to be tailored to the environment

E.g., HOMERUN suggested for corporate web site evaluation

- High-quality content
- Often updated
- Minimal download time
- Ease-of-use
- Relevant to users’ needs
- Unique to the online medium
- Netcentric corporate culture
Heuristic Evaluation: Advantages

Fewer practical and ethical issues to deal with

Best experts have knowledge of application domain & users

Can be difficult & expensive to find experts

Could have disagreement between experts
Heuristic Evaluation: Problems

principles may be too general

subtleties involved in use

designer may not be able to overcome being defensive / experts may disagree

you may actually have the wrong design altogether

can be hard to find experts

false positives: does the rule always apply?

not complete: will miss problems

not a replacement for user testing
Performance Modelling

Using mathematical models to generate quantitative predictions of certain interface actions or sequences of actions
Fitts’ Law

One of most tested, lasting models in HCI
Models target acquisition performance

\[ T = a + b \log_2 \left( \frac{A}{W} + 1 \right) \]

T = Time
A = Amplitude (distance)
W = Width (size) of target
a/b = empirically derived constants
Fitts’ Law

\[ T = a + b \log_2 \left( \frac{A}{W} + 1 \right) \]

What does the law tell us?

How can you use the law?
Hick-Hyman Law

Fitts’ Law predicts how long it will take users to acquire targets once they know which target to select.

What about decision time?

Hick-Hyman Law models the time it takes users to decide between $n$ alternatives.
Hick-Hyman Law

When items are equi-probable:

\[ DT = a + b \log_2 n \]

When certain items are more likely to be chosen than others:

\[ DT = a + b \left( \sum_{i=1}^{n} p_i \log_2 \left( \frac{1}{p_i} \right) \right) \]
Hick-Hyman Law

Applications?
E.g.,

Deciding on menu depth vs. breadth
Hick-Hyman Law

Models decision time, not searching time

If the user is not familiar with the interface elements:

Time to search through $n$ items is linear, not logarithmic
Keystroke Level Model

Given a task consisting of a sequence of steps

How long will it take the user to perform those steps given a specific interface?

Keystroke Level Model (KLM)

Models performance given a sequence of steps for an expert user
KLM

How is performance calculated?

Individual steps described using operators:

K = keystroking = 0.35s
P = pointing = 1.10s (Fitts’ Law for greater precision)
B = Button press or release (mouse) = 0.10 seconds (BB for mouse click = 0.20 seconds)
H = homing = 0.4s
D = drawing = variable with length of line
M = Mental operator = 1.35s
R = response operator by system = 1.2s

Sum up times for individual steps
Activity

How long would it take the user to replace all occurrences of a 4-letter word with a new 4-letter word.
# Activity

<table>
<thead>
<tr>
<th>Description</th>
<th>Operation</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reach for mouse</td>
<td>H</td>
<td>0.40</td>
</tr>
<tr>
<td>Move pointer to “Replace” button</td>
<td>P</td>
<td>1.10</td>
</tr>
<tr>
<td>Click on “Replace” button</td>
<td>BB</td>
<td>0.2</td>
</tr>
<tr>
<td>Home on keyboard</td>
<td>H</td>
<td>0.40</td>
</tr>
<tr>
<td>Type old word</td>
<td>M, K4</td>
<td>2.75</td>
</tr>
<tr>
<td>Reach for mouse</td>
<td>H</td>
<td>0.4</td>
</tr>
<tr>
<td>Move pointer to correct field</td>
<td>P</td>
<td>1.10</td>
</tr>
<tr>
<td>Click on field</td>
<td>BB</td>
<td>0.2</td>
</tr>
<tr>
<td>Home on keyboard</td>
<td>H</td>
<td>0.4</td>
</tr>
<tr>
<td>Type new word</td>
<td>M, K4</td>
<td>2.75</td>
</tr>
<tr>
<td>Reach for mouse</td>
<td>H</td>
<td>0.4</td>
</tr>
<tr>
<td>Move pointer to Replace All</td>
<td>P</td>
<td>1.10</td>
</tr>
<tr>
<td>Click Replace All</td>
<td>BB</td>
<td>0.2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>11.4</strong></td>
</tr>
</tbody>
</table>

According to this model, the time is 11.4 secs