CS 377E Spring 2016: Heuristic Evaluations
Instructor: James Landay, Julie Stanford, Jenny Mailhot

Heuristic Evaluation (Individual)

Due: 5/9/2016, in class

Overview

You have been hired as a consultant to another group in the class. They are building a new user interface for their course project, but they would like some outside assistance in finding some problems with their prototype interface. You can find your assigned project's medium-fi prototype and list of tasks on this Google Sheet.

Evaluation

You will perform a heuristic evaluation (individually) of their user interface using only the materials they turned in for their last project assignment (prototype and description of tasks). Using their tasks, task flows, interface design, screen shots, and medium-fi prototype you will apply Nielsen's heuristics to the user interface. You should be able to get all of this information from their last assignment. Read their tasks first and then run their prototype. Your evaluation will use both the information in the document and the prototype.

Please use the second set of heuristics from our lecture slides on heuristic evaluation (also described in Nielsen's chapter) and the numbering scheme from our lecture slides (e.g., 2-1, 2-2, etc.). You will produce a report showing the problems in the interface.

Report

Your report will **list each of the problems found** in the following format: problem# [heuristic violated] description of problem, reasoning why you think this violates the heuristic, & suggestion to fix.

For example:

1. **[H2-4 Consistency & Standards]**
   The interface used the string "Save" on the first screen for saving the user's information, but used the string "Store" on the second screen. Users may be confused by this different terminology for the same function. Use "Save" on all screens.
2. **[H2-3 User Control & Freedom]**
   The interface brings the user into a set of preference screens when they select "New User", but doesn't allow the user out of the dialog until they fill out all four screens. There is no way to cancel from any of the screens if a user came into the first screen by accident. Add a "Cancel" function to each screen in the sequence.

Your report will also **summarize the number of violations found in each of the ten heuristic categories** (make a table) and give the total number of violations in the entire interface. Finally, your report should close with some **overall recommendations** you have for improving the user interface given what you read in their presentation slides and what you experienced in testing their prototype (1 paragraph).

CS 377E Spring 2016: Heuristic Evaluations
Instructor: James Landay, Julie Stanford, Jenny Mailhot

Deliverables

You will turn-in your write-up (Word Processing document or PDF) directly in email to Sarah (snader2@stanford.edu) by the end of the 45-minute exercise in Monday's class. Make sure to bring your laptop with you to class on Monday as you will be working on this individually and then with a group. Please give your file a name that identifies you (e.g., johndoe-HE-prototype.doc). Your write-up should follow this outline with separate sections for the top-level items:

1. Prototype (one sentence description of the UI you are evaluating)
2. Violations Found (i.e., the list)
3. Summary of Violations (table)
4. Recommendations (1 paragraph)

Grading Criteria

You will be graded on how complete your HE report is in terms of coverage of the issues present in the user interface design, clarity of your violation descriptions, and quality of your recommendations. You should concentrate on the interface the group has *designed*, not only on what has been *implemented*. Reports that continually focus on features that are missing, but will clearly be added will be marked down (e.g., "there should be help on this screen... and this screen..." – if it is a globally missing feature like "help" you can report it once). Please focus on evaluating what they have designed so far.

Prototype (10 points)

Did you accurately and succinctly describe the prototype you are evaluating?

Violations (50 pts)

- Are you finding violations across all three tasks?
- Are you finding different kinds of violations, not just similar violations in many places? Make a note if something is frequent, but don't worry about citing every example.
- Do you have some less obvious violations (if they exist) in addition to the more obvious ones? Don't worry if it's hard to pick a category for a violation, it's more important that you spotted a difficult part of the interface. Many violations, even if they are small, will be helpful to the team.
- Are you focusing too much on missing features rather than giving helpful feedback on what you see?
- Is your feedback oriented around the tasks the team designed for?

Summary (15 pts)

- Have you summarized your results in a table? Make sure this is not just a laundry list of every violation, but a helpful and easy to read summary that gives the violations by category as well as overall.

Recommendations (25 pts)

- Is there any feedback you have that doesn't fit neatly into the violations?
- What are your general impressions when using the prototype? Do you have any additional feedback that you think would be helpful to the other team?
- Is there a larger trend or way of thinking that is spread across many of the violations you found?