Heuristic Evaluation Synthesis

Due: Nov 6 @ 11:59 PM

Objective

Synthesize individual heuristic evaluations of the same prototype so that the corresponding team can fix the most important problems found.

Learning Goal

Continue improving your understanding of Nielsen's adapted heuristics for evaluating and iterating on a user interface. Through discussion with your group members, you will refine your understanding of the heuristics as well as your ability to recognize different violation types.

Role of Assignment for the Project

While you are producing a report for a different team, you will in turn receive a report on the heuristic violations for your project. You will incorporate this feedback and the recommendations to improve your design before producing the high-fidelity prototype.

Assignment Overview

Make a copy of this spreadsheet. Throughout the report, ***use anonymized reviewer IDs***, not names. You will:

1. **Agree on a prototype description.** This should be a 1-sentence description capturing the UI you are evaluating and its purpose. Include the name of the product, the realization being used (e.g., "Mobile app"), a brief outline of the problem, and some explanation as to how the product provides a solution.
2. **Synthesize the violations found.** Each evaluator should copy and paste your individual heuristic evaluations from A7. Merge similar violations such that there are no duplicates. For such duplicates, you should list the issue only once in the spreadsheet ***using the best version of the description*** and suggested fix.
3. **Agree on a severity rating for each violation.** Use these ratings defined in lecture: 0 = not a usability problem, 1 = cosmetic, 2 = minor, 3 = major, 4 = usability catastrophe
4. **Format and organize your synthesized violations list.** Ensure the violations are organized by task flow (e.g., all violations for task 1 grouped together). If the violation occurs across all 3 tasks, label it using "All Tasks". If the violation doesn't fall under any of the specified tasks, label it using "Extra Violations". Similarly to A7, if the team has more than 3 tasks, add to the dropdown as needed. Within the list of violations for each task, feel free to sort the list in order of heuristic categories (H1 to H12) and/or by severity.

   Here is an example violation (formatting will look slightly different on the spreadsheet, but the idea is the same). The description should include what the violation is and where it

was found. When appropriate, consider using images to explain your rationale, grounding your recommendations based on the target user, and/or sketching potential fixes to provide even better feedback.
H4 Consistency & Standards | 1. Simple Task | 3 | Found by: A, C

Description: 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.
Rationale: Users may be confused by this inconsistent terminology for the same function.
Fix: Use "Save" on all screens.

5. **Review the summary of violation totals.** Review the 'Summary of Evaluations' tab. Note that all rows and columns in this table should add up as expected.

6. **Analyze how well each evaluator did by reviewing the evaluation statistics.** This will also show you how well the technique works. Review the 'Evaluation Statistics' tab and ensure that everything adds up as expected.

7. **Merge your summarizing recommendations.** Do this in the 'Summary Recommendations' tab. Group the recommendations logically (e.g., by theme, by task, etc). There should be *at least* 3 recommendations. For each recommendation, include what the recommendation is and a justification grounded in knowledge about the target user base, lecture slides, and/or outside research. This should be a few paragraphs.

Deliverables

Your group will submit a link to your synthesized spreadsheet through this Google Form. Make sure the permissions are set *so that your CA can view the file*.

1. **Synthesized report**
   Your file should be named "CS147 Group HE - [Project Name evaluated]"

Examples

*** Note: this assignment has been modified, so these examples are not perfect mappings to the deliverables (we required manual documentation for ALTiO and Cabana whereas the spreadsheet was required for Collabrio); however, the quality of the work stands.*
Collabrio (au25*)*, ALTiO (wi22), Localized (au18), Cabana (au17)

Grading Criteria

Report (100 pts)
Prototype Description (5)
___ One-sentence description capturing what the application seeks to achieve

Exceed expectations with:

_____ Conveying the problem and solution

List of Violations (65)

____ Organized correctly and avoids duplicate violations
____ Violation descriptions are detailed enough to be actionable
____ Gives good coverage of the actual problems in the prototype
____ Gives good coverage of the individual heuristic evaluations it synthesizes

Exceed expectations with:
____ Finding more and a greater variety of violations, across all the tasks
____ Use of images or sketches to convey issues or fixes
____ Very comprehensive coverage of the prototype's problems

Table Summaries (10)

____ Violations table is error free
____ Severities table is error free

Summary of Recommendations (20)

____ Recommendations merged in a coherent manner; provide value to the team

Exceed expectations with:
____ Considering project's values
____ Incorporating design patterns or specific graphic design principles
____ Grounding recommendations in details about the project (e.g., earlier assignments)
____ Multiple ideas per problem

CS 147 Autumn 2026: Assignment 9
Instructor: James Landay

12 Usability Heuristics

Adapted from Nielsen (2nd version). These are 12 general principles for user interface design including H11 and H12 created by the CS147 Teaching Staff. They are called “heuristics” because they are more in the nature of rules of thumb than specific usability guidelines.

H1. Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

H2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

H3. User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

H4. Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

H5. Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

H6. Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable when appropriate.

H7. Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

H8. Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

H9. Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

H10. Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

H11. Accessible design
Users can interact with the system using alternative input methods. Content is legible with distinguishable contrast and text size. Key information is upfront and not nested for screen readers. Purely visual or auditory content has text-based alternatives for users with low vision and low hearing.

H12. Value alignment and inclusion
The design should encode values that users can understand and relate to. It should make a diverse group of users feel included and respected. The design should prevent the reproduction of preexisting inequities and not create additional burdens for disadvantaged populations.