CS 147 Autumn 2023: Assignment 9
Instructor: James Landay

# Heuristic Evaluation Synthesis

## Due: Nov 10 @ 11:59 PM

### Goal

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

### Assignment Overview

You will receive a copy of the HE evaluation template from your studio CA. 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.
2. **Synthesize the violations found.** List each distinct problem with a unique number merging similar violations such that there are no duplicates. For such duplicates, you should list the issue only once in your report 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.** Organize the violations by task flow (e.g., all violations for task 1 grouped together). If the violation occurs across all tasks include these in an "All Tasks" section. If the violation occurs outside of a specific task, include this in a "Extra Violations" section. Your list of violations should be arranged in a logical order. For each violation, use the following format:

   *[#]. [H#] [Heuristic Name] / [Severity #] / Found by: [Reviewer IDs of those who found it]*
   *Task: [Where issue was found]*
   *Description: [What the issue is]*
   *Rationale: [Reason why it violates the heuristic]*
   *Fix: [Suggestion for fix]*

   For example:
   *1. H4 Consistency & Standards / Severity 3 / Found by: A, B*
   *Task: Specify your dietary preferences*
   *Description: The interface used the string "Save" on the first screen for saving the user's information, but used the string "Update" on the second screen.*
   *Rationale: Users might be confused by this different terminology for the same function.*
   *Fix: Use "Save" on all screens.*

5. **Summarize the violation totals in a table.** Fill in the table provided where each cell should contain the number of violations found that suffice the given row and column criteria (e.g., the first box in the upper left are all violations severity 0 and H1). A Google Sheet Template is provided here to help you calculate numbers.

| **Category** | **# Viol. (sev 0)** | **# Viol. (sev 1)** | **# Viol. (sev 2)** | **# Viol. (sev 3)** | **# Viol. (sev 4)** | **# Viol. (total)** |
|---|---|---|---|---|---|---|
| H1: Visibility of Status | 0 | 4 | 0 | 5 | 1 | **10** |
| H2: Match Sys & World | 1 | etc… | | | | |
| H3: User Control | 2 | | | | | |
| H4: Consistency & Standards | 1 | | | | | |
| H5: Error Prevention | 0 | | | | | |
| H6: Recognition not Recall | 1 | | | | | |
| H7: Efficiency of Use | 0 | | | | | |
| H8: Minimalist Design | 0 | | | | | |
| H9: Help Users with Errors | 0 | | | | | |
| H10: Help & Documentation | 2 | | | | | |
| H11: Accessible Design | 0 | | | | | |
| H12: Value Alignment & Incl | 3 | | | | | |
| **Total Violations by Severity** | **10** | | | | | |

*** Note that all rows and columns in this table should add up as expected.*

6. **Analyze how well each evaluator did by filling out the table below.** This will also show you how well the technique works. The same Google Sheet Template (provided here) will help you calculate numbers. The table should be filled out as follows:

| **Severity/Evaluator** | **Evaluator A** | **Evaluator B** | **Evaluator C** | **…** |
|---|---|---|---|---|
| **% Sev. 0** | $[\frac{\#\ severity\ 0\ found\ by\ A}{total\ \#\ severity\ 0\ found} \times 100]\ \%$ | etc. | | |
| **% Sev. 1** | $[\frac{\#\ severity\ 1\ found\ by\ A}{total\ \#\ severity\ 1\ found} \times 100]\ \%$ | | | |
| **% Sev. 2** | $[\frac{\#\ severity\ 2\ found\ by\ A}{total\ \#\ severity\ 2\ found} \times 100]\ \%$ | | | |
| **% Sev. 3** | $[\frac{\#\ severity\ 3\ found\ by\ A}{total\ \#\ severity\ 3\ found} \times 100]\ \%$ | | | |
| **% Sev. 4** | $[\frac{\#\ severity\ 4\ found\ by\ A}{total\ \#\ severity\ 4\ found} \times 100]\ \%$ | | | |
| **Total (sev 3 & 4)** | $[\frac{\#\ sev\ 3\ and\ 4\ found\ by\ A}{total\ \#\ sev\ 3\ and\ 4\ found} \times 100]\ \%$ | | | |
| **Total (all severity levels)** | $[\frac{total\ \#\ found\ by\ A}{total\ \#\ found\ by\ all} \times 100]\ \%$ | | | |

***Note that the bottom 2 rows are NOT calculated by adding the numbers above*

7. **Merge your summarizing recommendations.** Make sure that this segment is coherent and consistent. This should be a few paragraphs.

## Deliverables

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

1. **Synthesized report**
   Write up your report as a Google Doc using the provided template. Your file should be named "CS147 Group HE - [Project evaluated]"

## Examples

*** Note: this assignment has been modified, so these examples are not perfect mappings to the deliverables; however, the quality of the work stands.*

ALTiO, Cabana, Localized

## Grading Criteria

**Report (100 pts)**

*Prototype Description (5)*

____ One-sentence description capturing what the application seeks to achieve

*List of Violations (50)*

____ Formatted correctly, organized logically, 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

*Table Summaries (30)*

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

*Summary of Recommendations (15)*

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

## 12 Usability Heuristics

Adapted from Nielsen (2nd version). These are 12 general principles for user interface design. 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.