Heuristic Evaluation Workbook Template

Resources:

- Intro to Cognitive Walkthrough and Heuristic Evaluation (HCI WS 21/22)
- How to Conduct a Heuristic Evaluation by Jakob Nielsen
- Nielsen's 10 Usability Heuristics
- Watch Two Students Do Heuristics Evaluation (YouTube)

Heuristic Evaluation Workbook Template Guidelines:

- Make sure to follow the steps.
- Make sure the Comments, Severity Rating, Recommendations, and Effort Rating are aligned for each issue in the table below when documenting.
- Severity rating indicates the impact of a usability issue on the user experience. Common scales for severity rating include (use the numbers to provide ratings):
    - **No Issue (0):** Don't agree that this is a usability problem.
    - **Cosmetic (1):** Minor issues that don't significantly affect the user experience. Users may not even notice these problems.
    - **Minor (2):** Minor usability problem → Slightly noticeable issues that may cause some inconvenience but don't prevent users from completing tasks.
    - **Major (3):** Major usability problem; important to fix → Significant issues that seriously affect the user experience, potentially preventing users from completing important tasks.
    - **Critical (4):** Critical issues that render the system unusable or lead to severe data loss, posing a substantial risk to users (Usability catastrophe; imperative to fix).
- Effort rating refers to the level of difficulty or resources required to fix a usability issue. Common scales for effort rating include (use the letters to provide ratings):
    - **Low (L):** The issue is easy to fix and requires minimal time and resources.
    - **Moderate (M):** Fixing the issue requires some effort but is manageable within a reasonable timeframe.
    - **High (H):** The issue is complex and will require significant time, effort, and resources to fix.

Heuristic Evaluation Workbook Template

Evaluator: *name*	Product: NoteBot	Date
Task/Feature: *task/feature description*		

	Step 1 Focus on generating all the usability issues first	Step 2	Step 3 How can it improve? Jot down ideas freely	Step 4
Heuristics	**Issue(s)**	**Severity Rating**	**Recommendations**	**Effort Rating**
1. Visibility of System Status The design should always keep users informed about what is going on through appropriate feedback within a reasonable amount of time. • Does the design clearly communicate its state? • Is feedback presented quickly after user actions?	Issue: Description: (You can add more issues)			
2. Match Between System and the Real World The design should speak the users' language. Use words, phrases, and concepts familiar to the user rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order. • Will the user be familiar with the terminology used in the design? • Do the design controls follow real-world conventions?				
3. User Control and Freedom Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.				

• Does the design allow users to go back a step in the process? • Are exit links easily discoverable? • Can users easily cancel an action? • Is Undo and Redo supported?				
4. Consistency and Standards Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions. • Does the design follow industry conventions? • Are visual treatments used consistently throughout the design?				
5. Error Prevention Good error messages are important, but the best designs carefully prevent problems 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. • Does the design prevent slips by using helpful constraints? • Does the design warn users before they perform risky actions?				
6. Recognition Rather Than Recall Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g., field labels or menu items) should be visible or easily retrievable when needed. • Does the design keep important information visible so that users do not have to memorize it?				

• Does the design offer help in context?				
7. Flexibility and Efficiency of Use Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. • Does the design provide accelerators like keyboard shortcuts and touch gestures? • Is content and functionality personalized or customized for individual users?				
8. Aesthetic and Minimalist Design Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility. • Is the visual design and content focused on the essentials? • Have all distracting, unnecessary elements been removed?				
9. Help Users Recognize, Diagnose, and Recover from Errors Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution. • Does the design use traditional error message visuals, like bold, red text? • Does the design offer a solution that solves the error immediately?				
10. Help and Documentation It's best if the system doesn't need any additional				

explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks. • Is help documentation easy to search?? • Is help provided in context right at the moment when the user requires it?				