astro

USABILITY HEURISTICS

WHAT IS A HEURISTIC EVALUATION?

A heuristic evaluation is method of finding an application's usability problems.

WHAT DOES "HEURISTIC" MEAN?

A fast and practical way to solve problems or make decisions. *(From Interaction Design Foundation)*

GOALS

Evaluators will analyze each section or screen of the application against ten heuristic principles and document:

- Usability issues
- The usability principle(s) each issue violates
- Details and rationale for why it is a potential issue
- The severity of the issue

STEPS

- Identify either...
  - a typical usage scenario, listing specific steps an operator would take to perform a task
  - or, sections of the application to be evaluated
- Evaluate the first step or section of the application and compare it to the list of usability principles. Mark any principle(s) not in compliance.
- Write why the step breaks the marked principle(s). Be specific. These notes often lead to possible solutions.
- Mark how severe this usability principle is on a scale from Cosmetic to Critical.
- Repeat for every step of the usage scenario/part of the application.
- Compare notes with other evaluators to create a list of total usability issues and their severity.

Note: Creating recommendations to solve usability issues is not part of the evaluation.

NOTES/TIPS

- While a heuristic evaluation is performed individually, it's best to gather 3-5 total evaluators. This will aid in finding a range of usability issues that one evaluator may not notice.
- A full evaluation could take 1-2 hours. To simplify, the evaluators can focus only on specific parts of the application.
- It's recommended to go through the application twice. Once to familiarize the evaluator with the application and a second time to evaluate the usability.
- Recommending solutions is not part of the evaluation. The evaluation is to note the number of usability errors and their severity. Finding solutions is done after the evaluation with the notes from each evaluator.

10 HEURISTIC PRINCIPLES

1: VISIBILITY OF SYSTEM STATUS

Keep the user informed about what is happening behind the scenes in a reasonable timeframe.

Example: Progress indicator to show where the user is on a multi-step form information.

2: MATCH BETWEEN SYSTEM AND THE REAL WORLD

Use words, phrases, and concepts familiar to the user. Follow real-world conventions, making information appear in a natural and logical order.

Example: Forms should flow from left to right, top to bottom, as a person reads.

3: USER CONTROL AND FREEDOM

User 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.

Example: Cancel button or X to close modal window.

4: CONSISTENCY AND STANDARDS

Users should not have to wonder whether different words, situations, or actions mean the same thing – follow platform conventions.

Example: Navigation is typically located on the top and/or left of an application.

5: ERROR PREVENTION

Good error messages are important, but the best designs carefully prevent problems from occurring at all.

Example: Making input fields automatically format phone numbers and other similar types of information.

6: RECOGNITION RATHER THAN RECALL

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.

Example: Breadcrumbs or other elements easily seen.

7: FLEXIBILITY AND EFFICIENCY OF USE

Shortcuts — hidden from novice users — may speed up the interaction for the expert user.

Example: Hidden functionality that novice users may not know about.

8: AESTHETIC AND MINIMALIST DESIGN

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.

Example: Tooltips provide succinct information that could otherwise clutter a screen.

9: RECOGNIZE, DIAGNOSE, & RECOVER FROM ERRORS

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

Example: Error text in a form should provide information about what and where the error is as well as how to make the necessary corrections.

10: HELP AND DOCUMENTATION

It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.

Example: Info icons, Tooltips, Help Text.

SEVERITY SCALE

COSMETIC

Could easily be overcome by users or occurs relatively infrequently.

MINOR

Might be more difficult to overcome or occurs more frequently.

MAJOR

Users might be unable to fix or unaware of how to fix or occurs persistently.

CRITICAL

Seriously impairs the use of the product and cannot be overcome by users.

HEURISTIC Which principle(s) is broken?

1 2 3 4 5 6 7 8 9 10

SEVERITY How critical is this problem?

Cosmetic Minor Major Critical

SKETCHES

OBSERVATIONS What is the problem?

FEATURE/FUNCTIONALITY Where is the problem?

ISSUE Why is it a problem?

HEURISTIC Which principle(s) is broken?

1 2 3 4 5 6 7 8 9 10

SEVERITY How critical is this problem?

Cosmetic Minor Major Critical

SKETCHES

OBSERVATIONS What is the problem?

FEATURE/FUNCTIONALITY Where is the problem?

ISSUE Why is it a problem?