

Heuristic Evaluation

What is Heuristic Evaluation?

Heuristic evaluation is a method for finding usability problems in a user interface by reviewing it for compliance with a checklist of recognized usability principles called *heuristics*. It is conventionally performed by a group of trained evaluators who individually evaluate a system and then combine their results to come up with a prioritized list of problems to be fixed. However, we believe that most anyone can use a heuristic checklist to spot usability problems.

Heuristic evaluations are often used to evaluate an existing application, but you (and other members of your team) can use them during development to make sure you have taken these considerations into account in your system. It is generally best to have people who are not directly involved in the project evaluate the application using these heuristics. (You may want to trade heuristic evaluations with someone else who's working on a project at the same time.) These evaluators are able to look at it with "fresh eyes", allowing them to see potentially confusing things that frequent users of the application have learned to ignore, or that you may think make sense because you understand the implementation model.

Evaluators may review the interface by going through it in a systematic way that makes sense to them, or may follow previously defined user scenarios (e.g. task scenarios used in a [Cognitive Walkthrough](#)). This is usually determined by whether the goal of the heuristic evaluation is to do a general overview of the interface, or evaluate specific interactions, screens, or use cases.

As the evaluators find usability problems, they describe the problem, list the heuristic violated, and rate the severity of the problem, often on a scale of 0-4, from "not a usability problem at all" to "usability catastrophe; imperative to fix." (More information on ratings can be found at: [Severity Ratings for Usability Problems](#).) Usability problems can then be ordered by severity, the proportion of the users experiencing the problem, and the impact of the problem on users who experience it to determine which problems must be fixed immediately and which fixes can be deferred, if necessary, to the next round of development.

Heuristic Principles for Usability and Accessibility

Experts in usability and accessibility have compiled lists of principles - often expressed as questions - to be applied in performing heuristic analysis. The following are recommended by the Fluid project:

- [Heuristic Analysis - A System Checklist](#) is a useful table of principles, created by Deniese Pierotti of Xerox Corporation.
- [Usability Evaluation Questions](#) is a list developed by members of the Fluid project to address both usability and accessibility.
- [10 Nielsen and Molich Usability heuristics](#) is a well known set of principles, presented in a form refined by Jakob Nielsen.

For illustration purposes, here is an excerpt from the Nielsen and Molich list:

Visibility of system status

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

Match between system and the real world

The system should speak the user's 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.

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.

Consistency and standards

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

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.

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 whenever appropriate.

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.

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.

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.

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.

Heuristic Principles for Accessibility

The Fluid UX Accessibility Working Group has created a set of protocols for assessing accessibility:

- [Simple Accessibility Review Protocol](#): This is a set of heuristics for evaluating the general accessibility of a web application without the need for complex assistive technologies. It provides a simple technique that anyone can learn while doing a UX Walkthrough.

A paper from Claire Paddison and Paul Englefield provides a list of nine heuristic principles for accessibility evaluations:

- [Applying Heuristics to Perform a Rigorous Accessibility Inspection in a Commercial Context](#)
(Click on the Full Text PDF link and view pages 129-130.)

Paddison and Englefield include in their paper a general discussion of the heuristic approach. This is recommended reading for all reviewers.