

Introduction

The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces. Usability consultant Jakob Nielsen developed this method on the basis of several years of experience in teaching and consulting about usability engineering. Heuristic evaluations are one of the most informal methods of usability inspection in the field of human-computer interaction. There are many sets of usability design heuristics; they are not mutually exclusive and cover many of the same aspects of user interface design. Quite often, usability problems that are discovered are categorized—often on a numeric scale—according to their estimated impact on user performance or acceptance. Often the heuristic evaluation is conducted in the context of use cases (typical user tasks), to provide feedback to the developers on the extent to which the interface is likely to be compatible with the intended users' needs and preferences.

Credit: https://en.wikipedia.org/wiki/Heuristic_evaluation#Gerhardt-Powals.E2.80.99_cognitive_engineering_principles

Benefits

The simplicity of heuristic evaluation is beneficial at the early stages of design. This usability inspection method does not require user testing which can be burdensome due to the need for users, a place to test them and a payment for their time. Heuristic evaluation requires only one expert, reducing the complexity and expended time for evaluation. Most heuristic evaluations can be accomplished in a matter of days.

The time required varies with the size of the artifact, its complexity, the purpose of the review, the nature of the usability issues that arise in the review, and the competence of the reviewers. Using heuristic evaluation prior to user testing will reduce the number and severity of design errors discovered by users.

Although heuristic evaluation can uncover many major usability issues in a short period of time, a criticism that is often leveled is that results are highly influenced by the knowledge of the expert reviewer(s). This "one-sided" review repeatedly has different results than software performance testing, each type of testing uncovering a different set of problems.

The Heuristics

- 1. User control** heuristics that check whether the user has enough control of the interface.
- 2. Human limitations** the design takes into account human limitations, cognitive and sensorial, to avoid overloading them.
- 3. Modal integrity** the interface uses the most suitable modality for each task** auditory, visual, or motor/kinesthetic.
- 4. Accommodation** the design is adequate to fulfill the needs and behaviour of each targeted user group.
- 5. Linguistic clarity** the language used to communicate is efficient, clear and adequate to the audience.
- 6. Aesthetic integrity** the design is visually attractive and tailored to appeal to the target population.
- 7. Simplicity** the design does not use unnecessary complexity.
- 8. Predictability** users will be able to form a mental model of how the system will behave in response to actions.
- 9. Interpretation** there are codified rules that try to guess the user intentions and anticipate the actions needed.
- 10. Accuracy** There are no errors, i.e. the result of user actions correspond to their goals.
- 11. Technical clarity** the concepts represented in the interface have the highest possible correspondence to the problem domain **they are modeling.
- 12. Flexibility** the design can be adjusted to the needs and behaviour of each particular user.
- 13. Fulfillment** the user experience is adequate and the user feels good about the experience.
- 14. Cultural propriety** the user's cultural and social expectations are met.
- 15. Suitable tempo** the pace at which users works with the system is adequate.
- 16. Consistency** different parts of the system have the same style, so that there are no different ways to represent the same information or behavior.
- 17. User support** the design will support learning and provide the required assistance to usage.
- 18. Precision** the steps and results of a task will be what the user wants.
- 19. Forgiveness** the user will be able to recover to an adequate state after an error.
- 20. Responsiveness** the interface provides the user enough feedback information about the system status and their task completion.