

Running Head: HEURISTICS

Paper: Assignment

Style: Oxford

Pages: 8

Sources: 2

Level: Bachelor

Heuristics

[Name of the Writer]

[Name of the Institution]

## Heuristics

### Survey

The heuristic evaluation is a technical inspection interfaces that were analyzed, refined, and popularised by Nielsen and Molich the late 1980s and early 1990s (Nielsen and Molich, 1990, 250). The general idea is very simple and Nielsen and Molich are not the first to find it. An inspector examines the interface that evaluates against a list of properties (heuristics) that implements the interface wants. When the inspector identifies a violation of heuristics, it reports. A violation of a heuristic is actually a prediction of a usability problem. Inspection reports are used as inputs to the design process and re design.

The advantages claimed for heuristic evaluation in comparison with testing ease of use are: (Nielsen and Molich, 1990)

- It is less expensive
- it is easy to motivate developers to do
- It does not require much advance planning
- It can be used very early in the development process.

There is also evidence that among the many common inspection techniques, inspection heuristic is the easiest to learn. (Nielsen, 1994, 152)

Anyone can make a heuristic inspection. However, experiments have shown that normally usability experts to perform more efficient and economical than experts scope, system designers and users.

The ergonomic audit can be performed on every stage of design GUI. Indeed, we can already obtain interesting results from the early stages of prototyping to the final interface.

### Conduct and reporting

In the inspection, the inspector carefully examines the presentation and dynamic behavior of the interface with a comparison against a list of heuristics. Nielsen offers a list of ten specific heuristics, which was found by a rigorous analysis of 249 categories of usability problems (Nielsen, 1994, 155). You can also use other lists of heuristics.

Where an interface violates a heuristic, the inspector created a problem report. Normally, the report includes a tracking number, an indication of where the problem occurs interface (perhaps with a screenshot annotated), a description of the problem and an indication of how

heuristic was violated. The report may also include a severity index), but it is often better to determine the severity of each problem later in the process.

If the inspector is not an expert in the field of application, expert help is often useful to answer questions specific to the domain. This is in contrast with the normal practice tests for ease of use, where the participant is not allowed to access help. It may also be useful to give the inspector a set of scenarios that represent common tasks in the field.

### Multiple inspectors

Experiments showed that in an inspection, inspectors found only about the trireme heuristic violations that exist in the portion of the interface scrutinised. But, different inspectors will find different sets of violations. So, with one hole more inspectors vs. more problems, but at a high cost. In most circumstances, the benefit to cost ratio is optimised with between three and five inspectors.

If using multiple inspectors, each inspector working so individual. The inspectors' reports are combined and consolidated after.

### Activities after inspection

The reports are used as entered after the design process where re design. A rational process for reports should include:

### Consolidation of reports

The allocation of severity for each problem. For many inspectors, the normal process is to give each inspector a copy of the consolidated report and the inspector determined by individual work, the severity for each problem. We can find a way to determine the index. The average indexes are normally stable for three or more inspectors.

Decisions troubleshooting can be done taking into account the severity indices, the estimated level of effort as necessary to resolve the problem, or other considerations engineering or management.

### Audit usability: results fast, easy and cheap

The application of ergonomics audit a small number of evaluators assess the conformity of an interface to a list of ergonomic principles called heuristics. It is important to note that this method does not appeal to users, but to evaluators.

English speaking rather heuristic evaluation, but it seems that French is the preferred term ergonomics audit, which seems include several techniques. The method was formalised by Jakob Nielsen.

### Conduct an audit of ergonomics

The heuristic evaluation must be conducted by several people. As a first step, the evaluation is done by each person independently. Each evaluator should try to make a list of tasks. In carrying out these tasks repeatedly, it should be noted all the rules infringed by the interface.

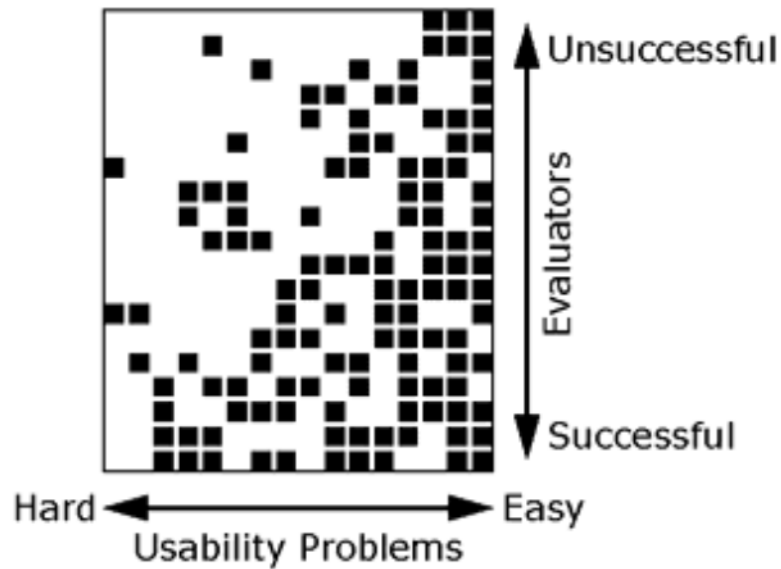
For each problem identified, the auditor should accurately describe the problem and how to arrive at the same position. The evaluator must also specify the rule is not respected and the severity of the violation.

An independent exploration by all auditors ensures maximum exploration of possible violations.

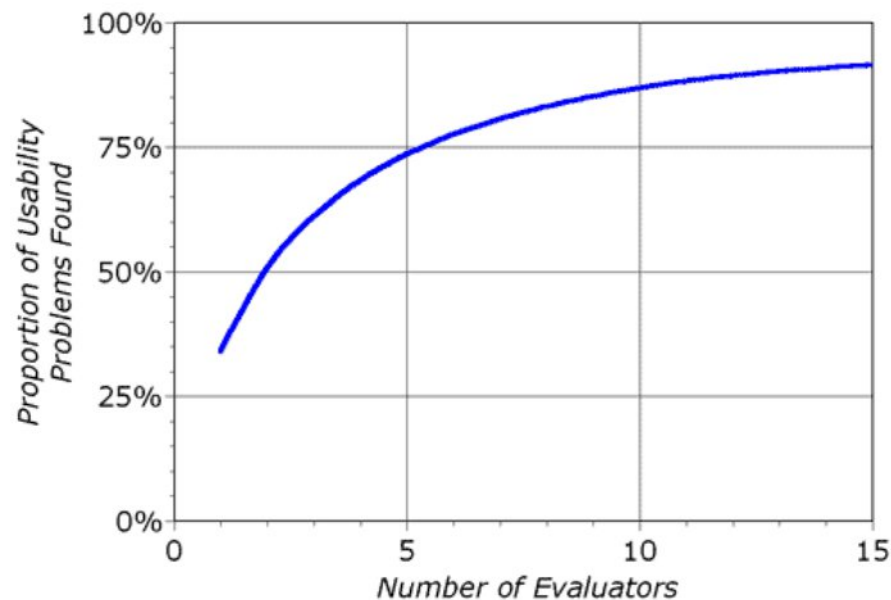
In a second step, the evaluators should share their work group identified several problems, and discuss the severity of each problem to determine the priorities in relation to the severity and cost of a possible correction.

### Three to five evaluators

An audit must involve multiple evaluators, who are obviously not the designers of the interface. It is impossible for a single evaluator, even very competent, to find without fail all the problems of an interface.



The above figure (by Jakob Nielsen), represents ergonomic problems found by evaluators. Each black square represents a problem is detected, and each white square problems undetected. The problems are ranked by frequency: the most commonly found are right. We see that in this case, the best evaluator did not find the third most common problem.



However useless to appeal to tens of evaluators. Nielsen showed that 3-5 evaluators suffice (figure above). This number can be sure of finding the most important problems, while

avoiding "false positives" identified by experts but not that impact users.

### Rules to check: heuristics

Jakob Nielsen has defined 10 rules that improve the usability of an interface. There are other lines of analysis, but it is simple, complete and this is the most common:

1. Visibility of system status: The system should always inform the user about what is happening in a reasonable time.
2. Use metaphors and colloquialisms: the system must use words, phrases, metaphors, concepts familiar to the user and not the technical terms. The system should follow the conventions of electronic and real world.
3. Control and freedom to the user: the user must be able to deceive and undo an action in error. There should be emergency exits.
4. Consistency and standards: two words, actions, metaphors, different image should not be used to mean the same thing. The standards of the platform should be respected.
5. Error Prevention: There is more than a beautiful and understandable error message. There is a design that prevents the error from occurring: the poka-yoke or actuator.
6. Identify rather than remember: make objects, actions and options visible. The user should not have to remember a hidden information in another screen. Similarly, the possible actions must be visible and identifiable.
7. Flexibility and efficiency accelerators allow advanced users to go faster and help meet the needs of novice and experienced users.
8. Aesthetics and minimalism: displays should contain only the information actually needed. The design should not be unnecessarily loaded, and must be properly organised.
9. Help to recognise, diagnose and correct errors: Error messages should be expressed in natural language, precisely indicate the problem and propose a solution.
10. Help and documentation: the system should offer help or documentation. This information should be easily accessible, research must be easy and it should be focused on user tasks.

### The results of a heuristic evaluation

An audit of ergonomics has several advantages. Firstly, it takes relatively little time and requires a limited number of evaluators, which makes it a low-cost method.

Then, a heuristic evaluation does not need to appeal to potential users of an application. This is an advantage because these users are often difficult to find and convince. Their time is valuable, and we cannot afford to ask them to test multiple design increments: we must recruit new users every time.

This type of audit can be carried out on a prototype, even in the early stages of prototyping. It can correct problems early in the design process, and therefore have a very low impact on the cost of development.

The audit, however, suffers from two important biases. The first is through the evaluators will probably find problems "false positives", that is to say, that did not bother the end user. Indeed, auditors spend an interface to the microscope through the prism of the heuristics, but did not ultimately used in a normal situation.

In addition, these evaluators will miss more serious problems. Imagine that lack essential functionality to your application. A real user will be quickly embarrassed because he seeks in vain this function. An auditor has reviewed the interface, verify that everything is consistent, that aid is present, etc. But he will not see what is missing.

It is for these reasons that the audit ergonomics seems an interesting technique, but appears complementary to a test with real users (a point which we will discuss later).

### Heuristics: how we could use

Today, this concept is foreign to us heuristics. But we realise an informal work that is similar to the audit.

As mentioned in the first part of the prototype, we work upstream models. Several designs are generally available, and the team believes is collected on each proposal. Everyone makes remarks indicating his preference.

If no proposal stands out clearly, another series of prototype is carried out taking into account the different returns.

But as these heuristics are not used as a reading grid, we certainly spend next important things. For each score points depending on the subjects to which it is most sensitive to the existing consistency, clarity of messages, the presence of aid, etc.

## References

- Nielsen, J. (1994) Enhancing the explanatory power of usability heuristics. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating interdependence (Boston, Massachusetts, United States, April 24 - 28, 1994). B. Adelson, S. Dumais, and J. Olson, Eds. CHI '94. ACM Press, New York, NY, 152-158.
- Nielsen, J. and Molich, R. (1990) Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People (Seattle, Washington, United States, April 01 - 05, 1990). J. C. Chew and J. Whiteside, Eds. CHI '90. ACM Press, New York, NY, 249-256.