Heuristic Evaluation and Usability Metrics

Guidelines can be used to help guide the design of a system.

Once a prototype system (or even a partial prototype) has been created, it can be analysed to see how usable it is.

The two main approaches to testing are **Heuristic Evaluation** and **Usability Metrics**.

Heuristic Evaluation

Heuristics:

- From the Greek word for "find" or "discover"
- Techniques based on experience that help in problem solving, etc.
- A means of rapidly arriving at a 'good enough' solution
- "rules of thumb", educated guesses, intuitive judgments, common sense

In Heuristic Evaluation, a number of evaluators examine an interface and assess its compliance with a set of recognised usability principles (the heuristics).

Heuristics are general rules - typically around 10 in number - which describe common properties of usable interfaces.

They are similar to guidelines, but framed so that they can be used in an ***analytical*** rather than ***generative*** manner.

For example, see Jacob's Nielsen's ***ten recommended heuristics.***

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

The process is as follows:

- Each evaluator is asked to assess the interface ***in the light of the heuristics*** - not their own likes/dislikes, etc..
- Each evaluator should work through the interface several times.
- Evaluators should either write-down their comments, or verbalise them, so that they can be recorded or noted by an observer.
- If an evaluator encounters problems with the interface the experimenter should offer assistance, but not until the evaluator has assessed and commented upon the problem.

Evaluators work alone, so that they cannot influence one-another.

Only when all the evaluators have assessed the system individually should the results be aggregated and the evaluators allowed to communicate with one another.

The number of evaluators is typically between 3 and 10, and they should have no prior knowledge of the interface or of the goals of the project, etc.

Using a single evaluator - even an experienced one - may not identify all the usability problems in an interface because different people identify different problems.

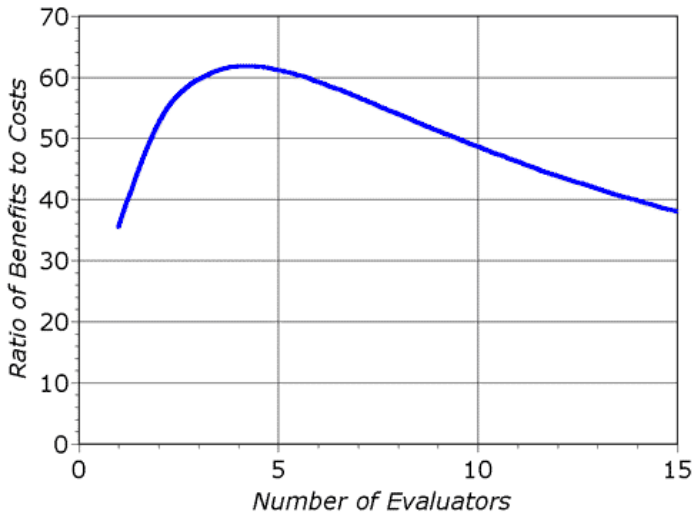
Nielsen (1992) conducted a study on Heuristic Evaluation:

- 19 evaluators were asked to assess an interface against a set of heuristics.
- Between them they identified 16 usability problems
- Some evaluators identified a far higher percentage of problems than others.
- However, a few problems were identified by only one or two evaluators, and these were not necessarily evaluators who found a high percentage of problems.

Using a large number of evaluators increases the likelihood of identifying problems but may also increase costs.

Nielsen compared effectiveness against cost in several large-scale projects and concluded that four evaluators is the best compromise.

Using a larger number of evaluators increased costs without greatly increasing the number of problems identified.



Usability Metrics

Heuristic Evaluation is often described as a ***discount*** method - a technique that costs relatively little to use but gives good results relative to its cost.

The term Usability Metrics refers to a range of techniques that are typically more expensive and time-consuming than Heuristic Evaluation but yield more detailed and/or more reliable results.

Techniques based on usability metrics involve asking a group of users to perform a specified task (or set of tasks).

Usability Metrics may gather either ***qualitative*** or ***quantitative*** data.

We'll consider an example of a quantitative metric.

Quantitative Metrics

Quantitative Metrics focus on measurable aspects of interaction such as:

- success rate (task completion/non-completion, % of task completed)
- time (e.g., time required to complete a specified task)
- errors (number of errors, time wasted by errors)
- use of help/documentation (number of instances, time spent)
- failed commands (number, how often repeated)
- user satisfaction (a subjective measure)
- etc.

Once gathered, the data may be analysed and used in a number of ways:

- Aggregated to yield either a set of scores, each reflecting a different aspect of usability, or a single overall usability rating.
- Analysed statistically to yield values that can be expressed to known level of uncertainty.

An example of a quantitative method is ***SUMI*** - the Software Usability Measurement Inventory.

To conduct a SUMI analysis, a number of subjects are asked to use a system and then complete a questionnaire about it.

At least 12 subjects are required, preferably far more.

The questionnaire typically contains 50 questions, of which the following are examples:

- This software responds too slowly to inputs.
- The instructions and prompts are helpful.
- The way that system information is presented is clear and understandable.
- I would not like to use this software every day.

The results are analysed to yield scores on the following scales:

- Efficiency
- Affect
- Helpfulness
- Control
- Learnability

The designers of SUMI claim that it has a high level of ***reliability***.

Reliability is measured by asking several different groups of subjects to fill in questionnaires for the same system.

- If the scores for each group vary significantly, it can be assumed that the questionnaire is mainly assessing the subjects.
- If the scores for each group DO NOT vary significantly, it can be assumed that the questionnaire is mainly assessing the system.

Thus a questionnaire's reliability is measured in terms of the amount of variability observed when different subjects assess the same system.

Automated Testing

Since web-pages follow an open-source standard, it's possible to test some of their features automatically.

Some automated testers only check the validity of the (X)HTML code, but others test for conformance with usability and accessibility guidelines.

Examples include paid-for services such as ***IBM's Rational Policy Tester Accessibility Edition*** and free, online checkers such as ***WAVE*** (http://wave.webaim.org/), ***TotalValidator***, ***Cynthia Says***, etc.

They automatically check many of the accessibility issues listed in the Web Content Accessibility Guidelines, e.g.:

- Inclusion of alt text, summaries, table header information, etc.
- Contrast between foreground and background colours
- etc.

Where a page is found to violate the guidelines, most testers identify the type of error and the line of HTML code on which it occurs.

Many accessibility issues cannot be checked automatically, so testers usually issue a number of standard warnings for all pages.

Designers should correct any specific errors identified and manually check the issues identified in the warnings.