# 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

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](http://web.archive.org/web/20100329062647/http%3A/www.useit.com/papers/heuristic/heuristic_list.html).

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..
* Evaluators work alone, so that they cannot influence one-another.
* 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.
* Only when all the evaluators have assessed the system individually should the results be aggregated and the evaluators be 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 reliable (usually *quantitative*) results.

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

The data gathered may include:

* success rate (task completion/non-completion, % of task completed)
* time
* 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 presented 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.

## Examples

A number of heuristics and metrics are currently in use, along with a number of techniques that do not strictly belong in either category but are often categorised as such.

We'll consider three examples:

* Cognitive Walkthrough
* SUMI (Software Usability Measurement Inventory)
* Watchfire WebXACT (online web-accessibility tester)

## Cognitive Walkthrough

The aim of the Cognitive Walkthrough (Wharton et.al., 1993) is to evaluate the steps required to complete a task and identify mis-matches between:

* the way the user thinks about the task
* the way the designer thinks about the task.

A Cognitive Walk-Through involves the following stages:

1. The user selects a task to be performed.
2. The user writes down all the steps required to complete the task
3. For each action in the task, the user:
	* explores the prototype, notes, or any available information that might enable him/her to perform the selected task.
	* selects the action that appears to match the required action most closely
	* interprets the system's responses and assesses if any progress has been made towards completing the task.

For each action in Step 3, the evaluators attempt to answer the following questions:

* How does the user know what to do next?
* Can the user connect the description of an action with what he/she is trying to do?
* Can the user tell if he/she has made the right choices on the basis of the feedback supplied by the system?

Attached is an [extract](http://web.archive.org/web/20100329062647/http%3A/www.cs.ucc.ie/~ianp/CS2511/HEM1_cogWalk.html) from a Cognitive Walkthrough analysing the steps involved in programming a video-recorder to record a television programme.

## SUMI (Software Usability Measurement Inventory)

To conduct a [SUMI](http://web.archive.org/web/20100329062647/http%3A/sumi.ucc.ie/) 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 are similar, it can be assumed that the questionnaire is revealing information about the system, not the subjects.

## Watchfire WebXACT

WebXACT works in a similar way to HTML validators, but analyses the target page for accessibility as well as for HTML code validity.

It automatically checks 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, WebXACT identifies the type of error and the line of HTML code on which it occurs.

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

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