

# What's the difference between a Heuristic Evaluation and a Cognitive Walkthrough?

Brought to you by: [lcraigwilson.com](http://lcraigwilson.com)

Heuristic Evaluations and Cognitive Walkthroughs belong to a family of techniques called Inspection Methods.

Inspection methods, like [Keystroke Level Modeling](#), are analytic techniques. They don't involve users and tend to generate results for a fraction of the time and cost as empirical techniques like usability testing.

They are also referred to as Expert Reviews because they are usually performed by an expert in usability or Human Computer Interaction. They are more than just some highly paid expert's opinion. Inspection Methods are based on principles derived from observing and analyzing user actions and intentions.

## Heuristic Evaluation

Heuristic evaluations were introduced by [Nielsen and Molich](#) in 1990 in the [influential paper](#) "Heuristic Evaluation of User Interfaces" and later assembled as part of a book (Nielsen 1994). A heuristic evaluation involves having an expert review an interface against a set of guidelines or principles. These heuristics provide a template to help uncover problems a user will likely encounter. [Nielsen's 10 heuristics](#) are the most popular but there are [actually many more](#).

Susan Weinschenk and Dean Barker (Weinschenk and Barker 2000) researched usability guidelines and heuristics from many sources (including Nielsen's, Apple and Microsoft) and did a massive card sort to generate the list of 20 below.

1. **User Control:** The interface will allow the user to perceive that they are in control and will allow appropriate control.
2. **Human Limitations:** The interface will not overload the user's cognitive, visual, auditory, tactile, or motor limits.

3. **Modal Integrity:** The interface will fit individual tasks within whatever modality is being used: auditory, visual, or motor/kinesthetic.
4. **Accommodation:** The interface will fit the way each user group works and thinks.
5. **Linguistic Clarity:** The interface will communicate as efficiently as possible.
6. **Aesthetic Integrity:** The interface will have an attractive and appropriate design.
7. **Simplicity:** The interface will present elements simply.
8. **Predictability:** The interface will behave in a manner such that users can accurately predict what will happen next.
9. **Interpretation:** The interface will make reasonable guesses about what the user is trying to do.
10. **Accuracy:** The interface will be free from errors
11. **Technical Clarity:** The interface will have the highest possible fidelity.
12. **Flexibility:** The interface will allow the user to adjust the design for custom use.
13. **Fulfillment:** The interface will provide a satisfying user experience.
14. **Cultural Propriety:** The interface will match the user's social customs and expectations.
15. **Suitable Tempo:** The interface will operate at a tempo suitable to the user.
16. **Consistency:** The interface will be consistent.
17. **User Support:** The interface will provide additional assistance as needed or requested.
18. **Precision:** The interface will allow the users to perform a task exactly.
19. **Forgiveness:** The interface will make actions recoverable.
20. **Responsiveness:** The interface will inform users about the results of their actions and the interface's status.

One of the biggest criticisms of HE is that it tends to uncover many low-severity problems or issues that aren't really problems (false positives). An additional practical problem is that multiple usability experts should be used. It can often be more expensive and difficult to find 3-5 usability professionals as it is to test 3-5 users.

## Cognitive Walkthrough

A cognitive walkthrough is also a usability inspection method like heuristic evaluation but the emphasis is on tasks. The idea is basically to identify users' goals, how they attempt them in the interface, then meticulously identify problems users would have as they learn to use an interface. The method was also

introduced at the same conference as Heuristic Evaluation (Lewis et al 1990). The cognitive walkthrough was an extension of earlier work by Polson and Lewis (1990).

For each action a user has to take to complete a task, a reviewer needs to describe the user's immediate goal and answer 8 questions:

1. First/Next atomic action user should take
2. How will user access description of action?
3. How will user associate description with action?
4. All other variable actions less appropriate?
5. How will user execute the action?
6. If timeouts, time for user to decide before timeout?
7. Execute the action. Describe system response.
8. Describe appropriate modified goal, if any.

It may come as no surprise that one of the biggest complaints about using the CW method is how long it takes to answer each question. In response to this, Spencer (2000) proposed a Streamlined Cognitive Walkthrough technique in which you ask only two questions at each user action.

1. Will the user know what to do at this step?
2. If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?

He found that by reducing the number of questions and setting up ground rules for the review team he was able to make CW work at Microsoft.

## What HE and CW have in Common

Despite their differences, the two techniques, like many inspection methods, have several things in common.

**Double Experts:** Having expertise in both Human Computer Action and the specific domain (e.g., Finance, Automotive Parts, etc.) will yield the best insights.

**Uncover Many Usability Problems:** In general both methods tend to find many of the problems in user-testing. The actual percentage of problems tends to vary from (30 to 90%) depending on the study (Hollingsed and Novick 2007). Interestingly enough, this percentage is similar to software inspection and walkthrough methods which tend to find between 30% and 70% of the logic-design and coding errors that are eventually detected (Myers 2004 p21).

**Not User Testing:** Neither method is a substitute for testing with actual users. Both offer a potentially cheaper way of identifying problems at all stages of the development process. It can be difficult to test users on a prototype and these inspection methods provide for early feedback, especially in an iterative design methodology.

**Users' Point of View:** Both methods require the usability experts to take the users' point of view as they inspect the interface.

**Multiple Evaluators are Best:** Each evaluator is only uncovering some of the usability problems (often around 30% for more obvious issues) so having multiple evaluators inspecting an interface will generate both more problems and identify overlapping problems.

**Inspections Can be more Thorough Than User Testing:** While usability testing is often looked at as the "gold-standard" for detecting problems, users are generally constrained to a small number of tasks. This limits their exposure in the interface and the opportunity to detect more problems. With HE/CW, a few evaluators can inspect all the "nooks and crannies" of an interface and provide more coverage of an interface.

## A Hybrid Approach: HE+CW

In my experience, having a focus on the tasks is valuable for diagnosing interaction problems and minimizing some of the peripheral problems HE tends to uncover. When I perform an Expert Review I use heuristics and a streamlined CW technique using these steps.

1. Define the users and their goals
2. Define the tasks they would attempt.

3. Walk-through the tasks step-by-step through the lens of the user (what terms they use, the things they'd look for and likely path's they'd take).
4. Look for and identify problems based on a set of Heuristics.
5. Specify where in the interface the problem is, how severe it is and possible design fixes.

I often perform an Expert Review prior to conducting a usability test and try and have at least two others independently review the interface with the same tasks. I typically see around 30-50% of problems we identified encountered by the users. This is encouraging because it shows these methods tend to uncover problems without testing users, while on the other hand it reminds us of the value of having actual users attempt the tasks.

Combining task scenarios with heuristics is nothing new. It's also a strategy recommended by Sears (1997) and even Nielsen (Nielsen 1993). So I suspect these two inspection methods will continue to blend to suit the need of practitioners.

For more information on how inspection methods compare see [the paper\[PDF\]](#) by Hollingsed and Novick which also contains one of the largest collections of references on Usability Inspection methods.

## References

- Hollingsed, T and Novick, D (2007). [Usability inspection methods after 15 years of research and practice\[PDF\]](#). In Proceedings of the 25th annual ACM international conference on Design of communication (SIGDOC '07). ACM, New York, NY, USA, 249-2 55.
- Lewis, C, Polson, P, Wharton, C and Rieman, J (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people (CHI '90), JACM, New York, NY, USA, 235-242.
- Myers, Glenford J (2004) [The Art of Software Testing\[PDF\]](#), Second Edition. Wiley.
- Nielsen, J (1993) [Usability Engineering](#). Wiley.

- Nielsen, J and Molich, R. (1990). [Heuristic evaluation of user interfaces](#)[pdf]. In Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people (CHI '90), ACM, New York, NY, USA, 249-256.
- Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen & R. L. Mack (Eds.), Usability inspection methods, (pp. 25-62). New York: John Wiley & Sons, Inc.
- Peter G. Polson and Clayton H. Lewis. 1990. [Theory-based design for easily learned interfaces](#). Hum.-Comput. Interact. 5, 2 (June 1990)
- Rick Spencer. 2000. [The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company](#). In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '00). ACM, New York, NY, USA, 353-359.
- Sears, A (1997) [Heuristic Walkthroughs. Finding the Problems without the Noise](#)[pdf]. Internation Journal of Human Computer Interaction 9(3) pp.213-234
- Weinschenk, S and Barker, (2000) [Designing Effective Speech Interfaces](#). Wiley