CHAPTER 4

Cognitive Walkthrough

Alternate Names: Simplified cognitive walkthrough, informal cognitive walkthrough.

Related Methods: Heuristic evaluation, perspective-based inspection, pluralistic walkthrough.

User Interface Inspection Methods. DOI: http://dx.doi.org/10.1016/B978-0-12-410391-7.00004-X

OVERVIEW OF THE COGNITIVE WALKTHROUGH

The cognitive walkthrough is a usability walkthrough technique that focuses primarily on the ease of learning of a product. A cognitive walkthrough can involve a single evaluator or a group of evaluators. In this author's experience, most cognitive walkthroughs involve a small group of usability experts and a few additional members from the product team.

The cognitive walkthrough is based on a theory that users often learn how to use a product through a process of exploration, not through formal training courses (Polson & Lewis, 1990). The cognitive walkthrough was originally designed to evaluate "walk-up-and-use" interfaces (e.g., museum kiosks, postage machines in public places, and ATM machines) but has been applied to more complex products (CAD systems, operating procedures, software development tools) that support new and infrequent users (Novick, 1999; Wharton, Bradford, Jeffries, & Franzke, 1992). The cognitive walkthrough is based on the concept of a hypothetical user and does not require any actual users, in contrast to the pluralistic walkthrough and think-aloud usability testing methods.

The cognitive walkthrough has gone through several versions and many extensions (Mahatody, Sagar, & Kolski, 2010). The original version, referred to here as CW Version 1 (Lewis, Polson, Wharton, & Riemen, 1990), was viewed as requiring substantial background in cognitive psychology (Wharton, Rieman, Lewis, & Polson, 1994) and cumbersome to apply in real-world environments. A second version, CW Version 2, tried to simplify the cognitive walkthrough method for practitioners who were not cognitive psychologists by using more detailed forms and instructions. However, these changes made the cognitive walkthrough procedure too laborious (and nearly as complex as Version 1) for most practitioners. The 1994 version (Wharton et al., 1994), CW Version 3, was written as "a practitioner's guide" and considered the primary reference for those who wanted to learn and conduct cognitive walkthroughs. However, even the practitioner's guide was sometimes viewed as too laborious for fast-paced commercial environments. Spencer (2000) proposed an even more simplified version, SCW, the "streamlined cognitive walkthrough" for fast-paced development. This chapter focuses primarily on Versions CW3 and SCW of the cognitive walkthrough. Later in this chapter, a variation on this method called the "informal cognitive walkthrough" (ICW) is described, which is adapted to agile development environments.

WHEN SHOULD YOU USE THE COGNITIVE WALKTHROUGH?

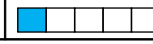
The cognitive walkthrough can be used during any phase of the development process but is most suited to the design stage where a functional specification and other design documents that provide enough information for the creation of action sequences (user inputs and system responses) for a set of tasks (Table 4.1). Cognitive walkthroughs can be conducted using textual descriptions of action sequences, sketches, paper prototypes, high-fidelity prototypes, and working products.

Table 4.2 illustrates the relative effort required, on average, to develop and use the cognitive walkthrough.

Table 4.1 Phases of Development When Cognitive Walkthroughs Are Useful

	✓	✓	✓	✓
Problem Definition	Requirements	Conceptual Design	Detailed Design	Implementation

Table 4.2 Relative Effort and Resources Required for Cognitive Walkthrough Development and Use

Overall Effort Required	Time for Planning and Conducting	Skill and Experience	Supplies and Equipment	Time for Data Analysis
				

STRENGTHS

The cognitive walkthrough has the following strengths:

- The cognitive walkthrough does not require a working product or even users.
- The cognitive walkthrough can be applied during any phase of development in which there is sufficient information to describe what users do and what the system does.
- The cognitive walkthrough provides detailed information based on cognitive theory that can be used to formulate specific solutions to problems.
- The cognitive walkthrough has well-defined procedures, is task-based, and focuses on one particular usability attribute, learnability.

Learnability Comes in Different Varieties

The cognitive walkthrough has a focus on initial usability—walk-up-and-use learning. Initial learning is one type of learnability (Grossman, Fitzmaurice, & Attar, 2009), but there are other types of learnability as well:

- **Learnability over time**. This deals with the impact of practice on errors and task performance.
- **First-time performance with instructions**. This type of learning is what people encounter with yearly national tax software programs or a voice-operated prescription refill service, which provides substantial embedded coaching to its users.
- **Expert learning**. How do experts in a domain learn new tools? How do experts on a tool learn how to use a new tool in the same domain? Many complex systems are not the ones that you can walk up and explore. Some systems (e.g., enterprise resource planning (ERP) and customer relationship management (CRM)) require days of training before anyone is allowed to touch a system.
- **Memorability**. How do users remember skills and knowledge over time?
- **Team learning**. With the future of computing quickly moving toward global collaboration, understanding how people learn to work in teams is becoming a critical issue in UX.

So, if you are asked to improve the learnability of a product, probe deeper with your team, client, or sponsor about just what kind or kinds of learning you need to consider.

WEAKNESSES

The cognitive walkthrough has the following weaknesses:

- The cognitive walkthrough has a narrower focus than other inspection methods such as heuristic evaluation. The focus is on learning and not efficiency or other usability attributes (although in practice, issues other than learnability are often captured).
- The method is laborious and slow, and coverage is often limited to a relatively small number of tasks. Rowley and Rhoades (1992) developed a method called the "cognitive jogthrough" in response to complaints from a development team that the cognitive walkthrough approach did not cover enough tasks to be cost-effective.
- Some complex products, such as the popular design tool, Photoshop™, or other content creation tools such as AutoCAD™,

allow many paths to the same goal. In the cognitive walkthrough, the walkthrough leader must choose the "appropriate" paths to achieve particular goals. Given the sheer number of tools and alternative ways to do things, there could be dozens of paths to accomplish the same goal in a complex product. Take the simple example of bolding text in Microsoft™ Word. You can bold with the keyboard, dialog box, styles, pop-up menus, find and replace, and icons in the ribbon. If your goal was to change all bold text to italics, your path to completing that goal could use any of these methods, although find and replace might be the best one to choose for making changes quickly.

WHAT DO YOU NEED TO USE THE COGNITIVE WALKTHROUGH METHOD?

This section provides a brief description of the basic resources needed to conduct a cognitive walkthrough.

Personnel

The cognitive walkthrough can be conducted by an individual or group. In a group evaluation, these are the important roles:

- **Facilitator**. The facilitator is generally the organizer and maestro of the walkthrough process. The facilitator is responsible for making sure that the walkthrough team is prepared for the session and follows the rules of good meeting management. Wharton et al. (1992) stress that the facilitator must decide when conversations can go beyond the narrow focus of the walkthrough and when those conversations must be reined in.
- **Notetake**r. The notetaker records the output of the cognitive walkthrough.
- **Product expert**. Because the cognitive walkthrough can be conducted early in the design stage (e.g., after requirements and a functional specification), a product expert is desired to answer questions that members of the walkthrough team may have about the system's features or feedback.
- **Evaluators**. Evaluators are representatives from the product team. These representatives can be usability practitioners, requirements engineers, business analysts, developers, writers, and trainers.

Table 4.3 Documents and Material Required for the Cognitive Walkthrough

Document or Item Name	Description	Required or Optional
User profile	Description of the primary users who perform the tasks that will be evaluated in a cognitive walkthrough.	Required
Task list	The task list should describe the task in a realistic and concrete manner.	Required
Action sequence for each task in the task list	Novick (1999) provides a good example. He includes the following information: • Date • Analysts • Users • Interface • Task • Action sequence • Comments.	Required
Problem reporting form	This is the form for recording the problems that emerge from the walkthrough.	Required
Representation of the UI	Flip charts, electronic display of outputs from cognitive walkthrough, prototypes, or working features.	Required

Documents and Materials

Table 4.3 lists the required and optional materials for the cognitive walkthrough.

PROCEDURES AND PRACTICAL ADVICE ON THE COGNITIVE WALKTHROUGH METHOD

The next section describes the basic procedures for the cognitive walkthrough method as well as practical tips for practitioners.

Planning a Cognitive Walkthrough

The basic steps in planning a cognitive walkthrough are listed here:

1. **Define the users of the product**. The 1994 practitioner's guide (Wharton et al.) and other articles provide little guidance on just how much you need to know about users before starting a cognitive walkthrough. Wharton et al. (1994, p. 109) use the example "Macintosh users who have worked with MacPaint," but you may want to consider a more robust description of the user. There is quite a difference between someone who has years of experience with various draw/paint tools and the eighty-five-year-old father

who was given a Mac and wants to sketch out the arrangement of flowers in his garden. You could use detailed user profiles or personas to provide a description of the primary users (Adlin & Pruitt, 2010). Different personas might take different paths through a system. If you have multiple personas, you can walk through the system with a focus on the attributes of each persona that could have an impact on how easily they can learn how to perform tasks with the target system.

2. **Determine what tasks and task variants (different ways to do the same task) are most appropriate for the walkthrough**. The theory behind the cognitive walkthrough does not address task selection, and the 1994 practitioner's guide provides little help on selecting the tasks that are most critical to users. The choice of tasks must balance complexity, realism, and the time allocated to the walkthrough. One strong recommendation is that the first task in the walkthrough be relatively simple so the team can learn the method before moving on to longer and more complex tasks. Wharton et al. (1992, pp. 383–384) suggest using realistic tasks that involve the use of several core features of the product. If there are multiple ways to do the same task, consider choosing the task variant that novices are most likely to encounter because the focus of this method is on initial learnability. Since much development is focused on new features, your tasks might be ones that take advantage of the new features in a product.
3. **Develop the ground rules for the walkthrough**. For example, your ground rules for conducting the cognitive walkthrough in a group might include:
   - Cell phones should be placed in silent mode.
   - No computers, smartphones, or tablets for the walkthrough participants. Only the notetaker and presenter will have a computer.
   - There will be no design discussions during the walkthrough. The purpose of the walkthrough is to elicit potential learning problems—redesign will generally take place at subsequent sessions. This can be a difficult rule, so you might provide cards where participants can jot down ideas for design solutions and collect those for later design activities.
   - Designers will not defend their designs during the walkthrough.
   - Participants will be professional and not use derogatory language (e.g., "this is a stupid design for a mobile system that is supposed to be learned in a few moments").

- The facilitator is in charge and will explicitly remind people of the ground rules if a violation occurs.

4. **Generate the action sequences for each task**. An action sequence is a detailed description of the actions the target users must take to complete a task and the associated system responses at each step. Here is a simple example of an action sequence that could be reviewed using a target user who is an elderly user with limited computer or web experience. Chrome was installed with the computer.

Task: Change the search engine to Bing from Google in the Google Chrome browser using the mouse.
Action Sequence:

1. Click on Chrome icon.
2. The browser appears.
3. Move the mouse pointer to the menu icon in the upper right of the browser.
4. A tooltip appears.
5. Click on the menu icon.
6. The Customize and Control Menu appears.
7. Move mouse pointer to Settings.
8. Click on Settings menu item.
9. Various controls appear in the browser area.
10. Move mouse to the Search drop-down.
11. Click on the drop-down.
12. Three menu items are shown: Google, Yahoo, and Bing.
13. Drag mouse to Bing.
14. Menu item highlights.
15. Click on highlighted Bing menu item.
16. Bing shows up in the drop-down menu field.

The granularity of the action sequence is an important consideration (Sears & Hess, 1998). For example, if the user must type some information in a text field, do you count each character as an action or the entire name as an action? The entire name might be considered as a single action on a desktop machine, but perhaps on a mobile device, each character might be considered an action. Another example dealing with task granularity is menu selection. Do you count the selection of a menu item from a pull-down menu as one action (1. Choose Print from the File menu.), two actions (1. Click on File. 2. Click on Print.), or three actions (1. Move

pointer to the File menu. 2. Click the File menu name. 3. Click the Print menu item.)? The granularity of actions can differ depending on the likelihood that the users will see an action as an aggregate or a set of subtasks.

5. **Provide a representation of the interface**. This representation can be a detailed text scenario, an operating procedure (Novick, 1999), a set of sketches, a storyboard, a paper prototype, a partially working prototype, or a fully working product.
6. **Assemble a group of evaluators for the cognitive walkthrough**. Candidates include usability practitioners, writers, trainers, product managers, quality engineers, and developers. This author has run cognitive walkthroughs with three to eight team members.

Conducting a Cognitive Walkthrough

To conduct the walkthrough, follow these steps:

1. **Walk through the action sequences for each task from the perspective of the "typical" users of the product**. For each step in the sequence, see if you can tell a credible story using one of three approaches:
   - The four-question approach of Wharton et al. (1994, p. 106):
     - Will the user try to achieve the right action?
     - Will the user notice that the correct action is available?
     - Will the user associate the correct action with the effect that the user is trying to achieve?
     - If the correct action is performed, will the user see that progress is being made toward solution of the task?
   - The ICW of Grigoreanu and Mohanna (2013) where user researchers played the role of users and asked themselves the questions:
     - As the user, would I know what to do at this step?
     - If I do the right thing, as the user, do I know that I have made progress toward this goal?

     After the user researcher walks through the tasks asking one of these sets of questions, other team members join in to provide their comments.

     The second part of the ICW process was a series of pluralistic walkthrough sessions (Chapter 5) where actual users commented on product task flows. So, the ICW is a hybrid method combining the cognitive walkthrough approach with the pluralistic walkthrough.

- The two-question streamlined approach of Spencer (2000):
  - Will the user know what to do at this step?
  - If the user does the right thing, will the user know that he or she did the right thing and is making progress toward the goal?

  For each action sequence in a task using the Spencer approach, the moderator (often the designer or usability practitioner) describes the action sequence and the state of the system after the user performs a correct action. Then, the evaluation team attempts to answer the two questions with plausible success or failure stories. If the team can come up with a plausible story for an action sequence, then nothing is recorded. However, if the team can't come up with a plausible success story, then the failure is recorded along with the knowledge that the user must know to progress. If you take the earlier example of an action sequence that required an elderly user to change a search setting, you might consider that there is no plausible success story for the action sequence "move the mouse pointer to the menu icon in the upper right of the browser" because an elderly user using the browser for the first time is not likely to know that the needed menu is represented by three horizontal bars in the upper right of the browser window. The elderly user might look for menus in the more traditional area under the title bar.

A "no" to any of the questions in the three approaches suggests a usability problem.

2. **Record success stories, failure stories, design suggestions, problems that were not the direct output of the walkthrough, assumptions about users, comments about the tasks, and other information that may be useful in design**. Use a standard form for this process so you can easily record and track the information.

After the Cognitive Walkthrough

After the walkthrough, follow these steps:

1. Bring the appropriate stakeholders together to review the results of the cognitive walkthrough.
2. Discuss potential solutions to the UI problems that were found in the walkthrough.
3. Determine which solutions you will apply to the product.
4. Evaluate the cognitive walkthrough process, and determine if there are improvements you could make for subsequent walkthroughs.

VARIATIONS AND EXTENSIONS TO THE COGNITIVE WALKTHROUGH

There are many variations and extensions to the cognitive walkthrough method in the literature. Some of these variations include:

- Heuristic walkthrough. See Chapter 1 for details on this method which is a cross between a heuristic evaluation and a cognitive walkthrough.
- Cognitive walkthrough for the web.
- Groupware walkthrough for highly collaborative systems.
- Cognitive walkthrough with users.
- Distributed cognitive walkthrough.

For a detailed review of eleven variations on the cognitive walkthrough, see the article "State of the Art on the Cognitive Walkthrough Method, Its Variants and Evolutions" by Mahatody et al. (2010). The article provides a description of each type of cognitive walkthrough and also discusses how well the methods support the finding of usability problems.

MAJOR ISSUES IN THE USE OF THE COGNITIVE WALKTHROUGH

Major issues that will face practitioners using the cognitive walkthrough are described in the section below.

How Do Evaluators and Teams Learn to Use the Cognitive Walkthrough Method?

The 1994 practitioner's guide recommends (p. 136) that one person on a walkthrough team has "some basic understanding of cognitive science," but it fails to define just what a "basic understanding" is. Wharton et al. (1994) state that the target audience for their practitioner's guide is "practicing software developers." Few practicing software developers have background in cognitive science so that would be a tough requirement. John and Packer (1995) present a case study of a single computer designer with little training in UX or cognitive science, learning and using the cognitive walkthrough method. They made the following conclusions from this detailed case study:

- The cognitive walkthrough method does not require deep experience with UCD or cognitive science.

- The practitioner's guide should be the primary reference for practitioners. The earlier papers may be too theoretical for the target audience of practicing software developers.

As a UX practitioner, follow this recommended training plan to learn the cognitive walkthrough method:

1. Read this chapter to get an overview.
2. Read the 1994 article and the streamlined method article by Spencer (2000) carefully with emphasis on the examples.
3. If possible, find a mentor who has conducted cognitive walkthroughs or a hands-on seminar or workshop.
4. Conduct several individual cognitive walkthroughs for practice using some simple tasks on a product you are familiar with, and make notes about points in the process where you had problems. Keep a diary and make notes about the process and questions that emerge during your first and subsequent experiences with the cognitive walkthrough (see John & Packer, 1995, for an example of what a diary might contain).

Task Definitions

The cognitive walkthrough does not provide much guidance about choosing tasks (Jeffries, Miller, Wharton, & Uyeda, 1991). The practitioner guide suggests that tasks be chosen on the basis of market studies, needs analysis, and requirements, which is not very helpful, especially at the design stage when there may be many such tasks to choose from. Wharton et al. (1992, p. 387) made some specific recommendations regarding tasks:

- Start with a simple task and move to more complex tasks.
- Consider how many tasks you can complete in a single walkthrough session. A common theme in the research and case study literature is that only a few tasks can be examined in any cognitive walkthrough session. A recommendation is to consider evaluating one to four tasks in any given session depending on task complexity.
- Choose realistic tasks that include core features of the product. Core features are those that form the underpinning of a product. For example, core features in Amazon.com are "search" and "shopping cart."
- Consider tasks that involve multiple core features so you can get input on transitions among the core features.

Other rationales for choosing tasks include the following:

- Whether the task uses new features that are considered high priority by the marketing and product development teams
- How important the task is to a user's first impressions of a product.

There are other factors to consider when choosing tasks for cognitive walkthroughs and other UX methods as listed in Table 4.4.

Table 4.4 Factors to Consider When Choosing Tasks for the Cognitive Walkthrough

Choose Tasks Based on	Description
Requests from your client	Although some clients will hand you a set of tasks to test, be wary because these tasks may only show the product in a good light without revealing core usability problems.
The constraints on your product	The state of the product will dictate possible tasks. Some tasks may not, for example, be possible with a paper prototype or medium-fidelity prototype.
Design team uncertainty	Are there parts of the product about which the design team has reservations or concerns (Wixon & Wilson, 1998)?
Verification of reputed problems	If you have feedback from technical support, articles about your product, or feedback from other sources about a problem, you may want to confirm and explore the problem in more depth in a think-aloud usability test.
Frequency	Tasks that people perform often are good candidates for think-aloud tasks. On the other hand, rare tasks that involve serious consequences also need to be considered.
Criticality	Infrequent tasks that can cause severe problems (lost revenues, catastrophic failure, and physical harm) are candidates for testing.
Important use cases	You can base your tasks on important use cases defined by the product team.
New features	Most product announcements tout new features that make products more "user friendly." Consider testing new features that are likely to have an impact on users or key features that are heavily promoted in the marketing and sales literature.
Avoidance of the confirmatory bias (Stacy & MacMillian, 1995)	The confirmatory bias is a tendency of those who create test cases for products to choose cases that are likely to show that a product works rather than fails. While the research has focused on software development, the same could hold true for usability personnel and is something to keep in mind. Usability personnel may be under subtle pressure to have results that are not too damning to the product.
Edge cases	Consider tasks that involve troubleshooting, error conditions, large databases, slow performance, and other edge cases. Edge cases can reveal problems that may not be evident under

(*Continued*)

Table 4.4 (Continued)

Choose Tasks Based on	Description
	"normal" conditions and be quite useful for finding problems that will tax your technical support lines.
Safety or liability concerns	There are examples in the human factors and engineering literature of safety-critical systems such as medical devices, backup software, military systems, and process control systems where a usability team should focus on tasks that are potentially injurious.
Accessibility	Does your system have to support users with various disabilities?

Tedious and Repetitive

A common complaint about the cognitive walkthrough is that the data collection requirements can be repetitive and tedious because you need to answer the same questions for each step in an action sequence. If your action sequence for a single task has twenty-five steps in the action sequence, you will need twenty-five pages to record your data collection. If you also collect design suggestions, assumptions about users, and other "side issues," you can accumulate a considerable quantity of online or paper forms. Here are recommendations for reducing the tediousness of the data collection:

- Record only those actions where there is a problem (a failure) (Wharton et al., 1992).
- Consider automating the data collection using a database with a simple form for input.
- Make any assumptions about the users publicly available during the walkthrough (e.g., record on a flip chart) so you don't have to continually refresh the memory of the evaluation team.
- Avoid "all-day" walkthroughs. Overly long walkthroughs are tiring and sometimes become an obstacle to the development process. Consider short sessions spread over several days.
- The facilitator of the walkthrough must balance the goals of the walkthrough with the goals of the product team. For example, cognitive walkthroughs are generally focused on evaluating the ease of learning of the product and not on solutions to design problems. However, if the discussion seems to be yielding a design solution to a problem that has been holding up the schedule, the facilitator might allow a long enough digression to capture the idea and then move on. And such a discussion might energize the team.

Suboptimal Solutions

The cognitive walkthrough method asks inspectors to record information on user assumptions and knowledge requirements but does not recommend how to use these records (Cockton et al., 2012).

The cognitive walkthrough emphasizes solutions for specific problems encountered in the action sequence of a task but does not deal with more general or higher level solutions that might be applicable across a number of tasks.

DATA, ANALYSIS, AND REPORTING

The primary data from a cognitive walkthrough are learnability problems based on knowledge or skill gaps that would prevent users from completing tasks successfully. Secondary data might include design flaws, or potential solutions (note that the goal is not to have a design session during the walkthrough, but good solutions should be captured). The problem data should be categorized by where in the action sequence it occurred, what user groups were considered, how severe the problem is, the expected frequency of the problem, and whether the problem is local (found in one place) or global (found in multiple places).

CONCLUSIONS

The cognitive walkthrough was designed to focus on exploratory learning and been the object of much research. A number of varieties of the cognitive walkthrough have been proposed to make the process more efficient, with recent "streamlined" and "informal" versions. The cognitive walkthrough is a technique for exposing assumptions about users and learning that can help designers create better first experiences.