Blackmon, M. H. (2004). Cognitive Walkthrough. In W. S. Bainbridge (Ed.), *Encyclopedia of Human-Computer Interaction*, 2 volumes (Vol. 1, pp. 104–107). Great Barrington, MA: Berkshire Publishing Group.

Cognitive Walkthrough

The cognitive walkthrough (CW) is a usability evaluation approach that predicts how easy it will be for people to learn to do particular tasks on a computer-based system. It is crucial to design systems for ease of learning, because people generally learn to use new computer-based systems by exploration. People resort to reading manuals, using help systems, or taking formal training only when they've been unsuccessful in learning to do their tasks by exploration. CW has been applied to a wide variety of systems, including automatic teller machines (ATMs), telephone message and call forwarding systems, websites, computerized patient-record systems for physicians, programming languages, multimedia authoring tools, and computer-supported cooperative work systems. HCI researcher Andrew J. Ko and associates innovatively applied CW (in lieu of pilot experiments) to predict problems that experimental participants might have with the instructions, procedures, materials, and interfaces used in experiments for testing the usability of a system (the system was a visual programming language).

Cognitive Walkthrough Methodology

The CW approach was invented in 1990 and has evolved into a cluster of similar methods with the following four defining features: (1) the evaluation centers on particular users and their key tasks, (2) the steps designers prescribe for doing the tasks are evaluated, (3) evaluators use theory-based, empirically verified predictions, and (4) software engineers can easily learn how to make CW evaluations. Each of the four defining features is elaborated below.

The evaluation centers on particular users and their key tasks.

Evaluators start a CW by carefully analyzing the distinctive characteristics of a particular user group, especially the relevant kinds of background knowledge these users can call upon when learning to perform tasks on the system. Next, CW evaluators select a set of key tasks that members of the user group will do on the system. Key tasks are tasks users do frequently, tasks that are critical even if done infrequently, and tasks that exhibit the core capabilities of the system.

The steps designers prescribe for doing tasks are evaluated.

For each key task, CW evaluators record the full sequence of actions necessary to do the task on the current version of the system. Then CW evaluators walk through the steps, simulating users' action selections and mental processes while doing the task. The simplest CW version asks two questions at each step: (1) Is it likely that these particular users will take the "right action"—meaning the action designers expect them to take—at this step? and (2) If these particular users do the "right action" and get the feedback the system provides (if any), will they know they made a good choice and realize that their action brought them closer to accomplishing their goal? To answer each question evaluators tell a believable success story or failure story. They record failure stories and have the option of adding suggestions for how to repair the problems and turn failures into successes. Anchoring the evaluation to the steps specified by designers communicates feedback to designers in their own terms, facilitating design modifications that repair the usability problems.

Evaluators use theory-based, research-supported predictions.

The foundation for CW is a theory of learning by exploration that is supported by extensive research done from the 1960s to the 1980s on how people attempt to solve novel problems when they lack expert knowledge or specific training. According to this theory, learning to do tasks on a computer-based system requires people to solve novel problems by using general problem-solving methods, general reading knowledge, and accumulated experience with computers. "The key idea is that correct actions are chosen based on their perceived similarity to the user's current goal" (Wharton et al., 1994, p. 126). For software applications, the theory predicts that a user scans available menu item labels on the computer screen and picks the menu item label that is most similar in meaning to the user's current goal. CW evaluators answer the first question with a success story if the "right action" designated by the designer is highly similar in meaning to the user's goal and if the menu item labels on the screen use words familiar to the user.

Software engineers can easily learn how to make CW evaluations.

It is crucial to involve software engineers and designers in CW, because they are the individuals responsible for revising the design to repair the problems. There is strong evidence that software engineers and designers can readily learn CW, but they have a shallower grasp of the underlying theory than usability experts trained in cognitive psychology and consequently find less than half as many usability problems. A group CW, including at least one usability expert trained in cognitive psychology, can find a higher percentage of usability problems than an individual evaluator—up to 50 percent of the problems that appear in usability tests of the system.

CW was one of the several evaluation methods pioneered in the early 1990s to meet a practical need, the need to identify and repair usability problems early and repeatedly during the product development cycle. The cost of repairing usability problems rises steeply as software engineers invest more time in building the actual system, so it is important to catch and fix problems as early as possible. For a product nearing completion the best evaluation method is usability testing with end users (the people who will actually use the system), but CW is appropriate whenever it is not possible to do usability testing. Early versions of CW were tedious to perform, but the 1992 cognitive jogthrough and streamlined CW of 2000, which still preserve all the essential CW features, are much quicker to perform. .

Transforming CW to Faster and More Accurately Predict User Actions

The cognitive walkthrough for the Web (CWW) has transformed the CW approach by relying on Latent Semantic Analysis (LSA)—instead of on the subjective judgments of usability experts and software engineers—to predict whether users are likely to select the “right action.” LSA is a computer software system that objectively measures semantic similarity—similarity in meaning—between any two passages of text. LSA also assesses how familiar words and phrases are for particular user groups.

While analyzing the distinctive characteristics of the particular user group, CWW evaluators choose the LSA semantic space that best represents the background knowledge of the particular user group—the space built from documents that these users are likely to have read. For example, CWW currently offers a college-level space for French and five spaces that accurately represent general reading knowledge for English at college level and at third-, sixth-, ninth-, and twelfth-grade levels.

CWW uses LSA to measure the semantic similarity between a user’s information search goal (described in 100 to 200 words) and the text labels for each and every subregion of the web page and for each and every link appearing on a web page. CWW then ranks all the subregions and link labels in order of decreasing similarity to the user’s goal. CWW predicts success if the “right action” is the highest-ranking link, if that link is nested within the highest-ranking subregion, and if the “right action” link label and subregion avoid using words that are liable to be unfamiliar to members of the user group.

Relying on LSA produces the same objective answer every time, and laboratory experiments confirm that actual users almost always encounter serious problems whenever CWW predicts that users will have problems doing a particular task. Furthermore, using CWW to repair the problems produces two-to-one gains in user performance. So far, CWW researchers have tested predictions and repairs only for users with college-level reading knowledge of English, but they expect to prove that CWW gives comparably accurate predictions for other user groups and semantic spaces.

Research by cognitive psychologist Rodolfo Soto suggests that CW evaluations of software applications would be improved by relying on LSA, but to date CW has consistently relied on subjective judgments of human evaluators. Consequently the agreement between any two CW evaluators is typically low, raising concerns about the accuracy of CW predictions. Many studies have tried to assess the accuracy and cost-effectiveness of CW compared to usability testing and other evaluation methods. The results are inconclusive, because there is controversy about the experimental design and statistics of these studies.

Relying on LSA opens the door to fully automating CWW and increasing its cost-effectiveness. If other CW methods start to rely on LSA they, too, could be automated. The streamlined CW is more efficient than earlier CW methods, but it still consumes the time of multiple analysts and relies on subjective judgments of uncertain accuracy.

Objectively Predicting Actions for Diverse Users

Relying on LSA makes it possible for CWW to do something that even usability experts trained in cognitive psychology can almost never do: objectively predict action selections for user groups whose background knowledge is very different from the background knowledge of the human evaluators. For example, selecting the sixth-grade semantic space enables LSA to “think” like a sixth grader, because the sixth-grade LSA semantic space contains only documents likely to have been read by people who have a sixth-grade education. In contrast, a college-educated analyst cannot forget the words, skills, and technical terms learned since sixth grade and cannot, therefore, think like a sixth grader.

Since CW was invented in 1990, the number and diversity of people using computers and the Internet have multiplied rapidly. Relying on LSA will enable the CW approach to keep pace with these changes. In cases where none of the existing LSA semantic spaces offers a close match with the background knowledge of the target user group, new semantic spaces can be constructed for CWW (and potentially for CW) analyses—in any language at any level of ability in that language. Specialized semantic spaces can also be created for bilingual and ethnic minority user groups and user groups with advanced background knowledge in a specific domain, such as the domain of medicine for evaluating systems used by health professionals.

Marilyn Hughes Blackmon, Ph.D.
University of Colorado at Boulder

See also User Modeling; Errors

Further Reading

Blackmon, M. H., Kitajima, M., & Polson, P. G. (2003). Repairing usability problems identified by the cognitive walkthrough for the web. In *CHI 2003: Proceedings of the Conference on Human Factors in Computing Systems*, 497–504.

Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C. (2002). Cognitive walkthrough for the Web. In *CHI 2002: Proceedings of the Conference on Human Factors in Computing Systems*, 463–470.

Desurvire, H. W. (1994). Faster, cheaper!! Are usability inspection methods as effective as empirical testing? In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 173–202). New York: Wiley.

Gray, W. D., & Salzman, M. D. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction, 13*(3), 203–261.

Hertzum, M., & Jacobsen, N. E. (2003). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human–Computer Interaction, 15*(1), 183–204.

John B. E., & Marks, S. J. (1997). Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology, 16*(4/5), 188–202.

John, B. E., & Mashyna, M. M. (1997). Evaluating a multimedia authoring tool. *Journal of the American Society for Information Science, 48*(11), 1004–1022.

Ko, A. J., Burnett, M. M., Green, T. R. G., Rothermel, K. J., & Cook, C. R. (2002). Improving the design of visual programming language experiments using cognitive walkthroughs. *Journal of Visual Languages and Computing, 13*, 517–544.

Kushniruk, A. W., Kaufman, D. R., Patel, V. L., Lévesque, Y., & Lottin, P. (1996). Assessment of a computerized patient record system: A cognitive approach to evaluating medical technology. *M D Computing, 13*(5), 406–415.

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *CHI '90: Proceedings of the Conference on Human Factors in Computing Systems,* 235–242.

Lewis, C., & Wharton, C. (1997). Cognitive walkthroughs. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., revised, pp. 717–732). Amsterdam: Elsevier.

Pinelle, D., & Gutwin, C. (2002). Groupware walkthrough: Adding context to groupware usability evaluation. In *CHI 2002: Proceedings of the Conference on Human Factors in Computing Systems,* 455–462.

Polson, P., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies, 36,* 741–773.

Rowley, D. E., & Rhoades, D. G. (1992). The cognitive jogthrough: A fast-paced user interface evaluation procedure. In *CHI '92: Proceedings of the Conference on Human Factors in Computing Systems*, 389–395.

Sears, A., & Hess, D. J. (1999). Cognitive walkthroughs: Understanding the effect of task description detail on evaluator performance. *International Journal of Human-Computer Interaction, 11*(3), 185–200.

Soto, R. (1999). Learning and performing by exploration: Label quality measured by Latent Semantic Analysis. In *CHI '99: Proceedings of the Conference on Human Factors and Computing Systems,* 418–425.

Spencer, R. (2000). The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. In *CHI 2000: Proceedings of the Conference on Human Factors in Computing Systems*, 353–359.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 105–140). New York: Wiley.