tification. Also, HE does not necessarily evaluate the complete design, since there is no mechanism to ensure the entire design is explored, and evaluators can focus too much on one section; and the validity of Nielsen's guidelines has been questioned [9].

A cognitive walkthrough (CW) is a task-oriented method by which the analyst explores the system's functionalities; that is, CW simulates step-by-step user behavior for a given task. CW emphasizes cognitive issues, such as learnability, by analyzing the mental processes required of the users. This can be achieved during the design by making the repertory of available actions salient, providing an obvious way to undo actions, and offering limited alternatives [5]. The background is derived from exploratory learning principles. Several versions of CW exist, including pluralistic walkthroughs wherein end users, software developers, and usability experts go through the system, discussing every single dialogue element.

Advantages include independence from end users and a fully functioning prototype, helping designers to take on a potential user's perspective, effective identification of problems arising from interaction with the system, and the ability to help to define users' goals and assumptions.

Disadvantages of CW include possible tediousness and the danger of an inherent bias due to improper task selection, emphasis on low-level details, and noninvolvement of the end user.

The action analysis method is divided into formal and back-of-the-envelope action analysis; in both, the emphasis is more on what the practitioners do than on what they *say* they do. The formal method requires close inspection of the action sequences a user performs to complete a task. This is also called keystroke-level analysis [2]. It involves breaking the task into individual actions such as move-mouse-to-menu or type-on-the-keyboard and calculating the times needed to perform the actions. Back-of-the-envelope analysis is less detailed and gives less precise results, but it can be performed much faster. This involves a similar walkthrough of the actions a user will perform with regard to physical, cognitive, and perceptual loading. To understand this thoroughly we must keep in mind that goals are external, and we *achieve* goals. Tasks are those processes applied through some device in order to achieve the goals, and we *perform* tasks. Actions are tasks with no problem-solving and no internal control structure. We *do* actions. The main problem of task analysis [3] is the difficulty in accommodating complicated tasks completed by more than one individual. Furthermore, the representation of a task analysis is complex, even when a simple task is studied, and tends to become very unwieldy very rapidly. Such representations can often only be interpreted by those who conducted the analysis.

Advantages include precise prediction of how long a task will take, and a deep insight into users' behavior.

Disadvantages of action analysis include it is very time-consuming and requires high expertise.

Usability Test Methods

Testing with end users is the most fundamental usability method and is in some sense indispensable. It provides direct information about how people use our systems and their exact problems with a specific interface. There are several methods for testing usability, the most common being thinking aloud, field observation, and questionnaires.

Thinking aloud (THA) [7] may be the single most valuable usability engineering method. It involves having an end user continuously thinking out loud while using the system. By verbalizing their thoughts, the test users enable us to understand how they view the system, which makes it easier to identify the end users' major misconceptions. By showing how users interpret each individual interface item, THA facilitates a direct understanding of which parts of the dialogue cause the most problems. In THA the time is very important, since the contents of the users' working memory contents are desired. Retrospective reports are much less useful, since they rely on the users' memory of what they had been thinking some time ago. A variant of THA called *constructive interaction* involves having two test users use a system together (co-discovery learning). The main advantage is that the test situation is much more natural than standard THA with single users working alone, since people are used to verbalizing their thoughts when trying to solve a problem together. Therefore, users may make more comments when engaged in constructive interaction than when simply thinking aloud for the benefit of an experimenter.

Advantages of THA include revealing *why* users do something; providing a close approximation to how individuals use the system in practice; provision of a wealth of data, which can be collected from a fairly small number of users; user comments of often contain vivid and explicit quotes; preference and performance information can be collected simultaneously; THA helps some users to focus and concentrate; and early clues can help to anticipate and trace the source of problems to avoid later misconceptions and confusion in the early stage of design.

Disadvantages include a failure to lend itself well to most types of performance measurement; the different learning style is often perceived as unnatural, dis-