

# 사용성 평가 방법들의 선택에 관한 연구 (Selecting Usability Evaluation Methods)

박 경 수, 임 치 환

한국과학기술원 산업공학과

## ABSTRACT

Usability evaluation can be a valuable tool for analyzing and improving computer software. Various usability evaluation methods have been developed in the fields of human-computer interaction and software ergonomics. Although there are several useful methods available, designers or engineers engaged in usability evaluations should really select the most suitable methods for each specific system being evaluated. Even different systems within the same organization may require different evaluation methods to maximize the quality of the data obtained from each evaluation. In this paper, the taxonomy of evaluation methods based on the development life cycle is presented and comparative analysis between methods is performed. In addition, the implications for selecting appropriate usability methods and using them collectively are discussed.

## 1. 서론

오늘날 사용성 평가 (usability evaluation)은 컴퓨터 소프트웨어를 분석하고 개선하기 위한 필수적인 도구로써 인식되고 있다. 사용성 연구 (usability study)에서 주요 이슈 중의 하나는 제품 사용시 발생하는 사용성 문제들 (usability problems)를 어떻게 측정하고 완화시키느냐 하는 것이다. 인간-컴퓨터 상호작용 (human-computer interaction)과 소프트웨어 인간공학 (software ergonomics) 분야에서는, 제품에 포함된 사용성 문제들이 다시 일어나지 않도록 보장하고 평가하기 위하여 다양하고 많은 방법들이 연구되어 왔다. 그러나 활용 가능한 평가 방법들의 다양함으로 인해서, 실제로 사용성 평가를 수행하는 설계자나 공학자들은 “평가될 특정 시스템에 가장 적합한 평가 방법은 무엇인가” 라는 문제에 자주 접하게 되었다. 일반적으로 제품의 사용성을 평가할 때는 가용한 시간과 주어진 예산 하에서 가능한 빠르고 저렴한 평가 방법이 제품의 주요한 문제점들을 발견하는데 사용되어야 할 것이다. 본 연구에서는 지금까지 연구되어 온 다양한 사용성 평가 방법들을 분류하고, 각 평가 방법들의 장단점을 비교하였다. 또한 현재 상태에서 가장 적합한 사용성 평가 방법을 선택하기 위하여 고려해야 될 사항들을 제시함으로써 주어진 예산과 시간 제약 하에서 제품 수명 주기에 적합한 평가 방법들의 전략 수립에 도움을 주고자 한다.

## 2. 제품 수명주기에 따른 사용성 평가 방법들

컴퓨터 소프트웨어 제품의 사용성을 보증하고 유용한 사용자 인터페이스를 설계하기 위해서는 제품의 수명주기에 따른 체계적인 사용성 노력을 기울여야 할 것이다. Mantei and Teorey (1988)

는 일반적인 제품 수명주기에 인간공학적 내용을 포함한 인간공학 수명주기 (human factors lifecycle)를 제시하였다. 한편 Nielsen (1993)은 제품의 설계 및 개발 단계에 쉽게 통합되어 체계적인 사용성 노력을 기울일 수 있는 사용성 공학 수명주기를 제시하였다. <표 1>은 앞에서 언급한 두 수명주기 동안 활용 가능한 여러 가지 사용성 평가 방법들을 보여준다. 일반적으로 사용성 연구의 성공을 보장하기 위해서는 초기에 어떠한 사용성 평가 방법들을 사용할 것인가에 관한 명백한 계획을 세우는 것이 중요할 것이다.

### 3. 사용성 평가 방법들의 분류

McGrath (1994)는 사용자 인터페이스 평가 방법들에 대한 간결하고 폭 넓은 틀 (framework)을 제공하기 위하여 <표 2>와 같은 네 가지 평가 전략들의 분류를 제시하였다. 그는 사용자들의 모집단에 대한 증거들이 얼마나 일반적인가 하는 일반화 (generalizability), 연구된 행태들의 측정이 얼마나 정확한가 하는 정밀도 (precision), 그리고 증거들이 수집되는 상황이 얼마나 현실에 가까운가를 의미하는 현실성 (realism) 등간의 질충 관계가 있음을 주장하였다. 이것은 바람직한 세 가지 속성들을 모두 포함하는 단 하나의 방법은 존재하지 않는다는 것을 의미한다. 일반적으로 평가 결과들은 평가 방법들에 의존하고, 모든 방법들은 제약을 갖고 있으므로 얻어진 결과들도 역시 제한적 일 것이다.

여러 가지 평가 방법들에 대한 다른 분류 방법은 자동적 (automatic), 실증적 (empirical), 정형적 (formal), 그리고 비정형적 (informal) 방법 등 네 가지로 요약해 볼 수 있다. 사용성은 사용자들, 직무들, 시스템, 그리고 환경 등 시스템이 사용되는 모든 상황 (context)를 고려해야 하므로, 현재 기술 수준 하에서는 자동화된 방법은 잘 사용되지 않는 듯하다. 시스템 운용상의 복잡도 (operational complexity)를 평가하기 위한 정형적 방법은 사용자 행태 (behavior)를 직접 측정하기 보다는 오히려 사용자가 시스템과 상호작용할 때 일어나는 것을 분석하고 예측하는데 이용하며, 숙련자의 수행도를 평가하는데 적합하다. 그러나 정형적 방법은 사용자에 대한 학습의 전이 (transfer), 사용시 범한 오류, 사용시 즐거움 (pleasant-to-use) 등을 잘 설명하지는 못한다. 또한 정형화된 분석을 수행하기 위해서는 직무에 대한 상세한 묘사 (description)를 필요로 하며, 직무의 작은 변동에도 많은 재분석이 요구되기도 한다. 따라서 정형화된 방법들은 규모가 큰 사용자 인터페이스에 적용하기 곤란하고 해석하기 어렵다.

실증적 방법들은 아마도 가장 일반적으로 사용되는 사용자 테스트 (user testing)과 함께 사용자 인터페이스를 평가하는 주요한 방법들이다. 종종 전개되는 디자인의 모든 버전들의 모든 측면들을 테스트하기에 충분한 실 사용자들 (피실험자들)을 모집하는 것이 어렵고 많은 비용을 수반할 수 있다. 이러한 제약을 보완하기 위해 전문가들에 의한 사용성 검사 (usability inspection) 방법이 있다. 정해진 프로젝트 스케줄이나 한정된 예산 때문에 사용성 검사와 같은 비정형적 평가 방법이 이용된다.

### 4. 평가 방법들의 비교

Jeffries et al. (1991)은 소프트웨어의 사용성 문제들을 발견하는 관점에서 휴리스틱 평가, 소프트웨어 가이드라인, 인지적 시연 (cognitive walkthrough)와 사용성 테스트 등 네 가지 방법들의 상대적인 효율성을 비교하였다. 그 결과 사용성 테스트 방법은 일반적인 문제점들 (global problems)는 잘 발견하였으나 국부적인 문제들 (local problems)는 잘 발견하지 못하였다. 오히려 휴리스틱 평가 방법이 특별하고 국부적인 문제들을 많이 발견하였으며, 두 가지 방법들을 상호 보완적으로 사용하는 것이 좋다고 결론지었다. 한편 Bailey et al. (1992)는 휴리스틱 평가에서 발견된 많은 국부적인 문제들의 변경을 통해서 소프트웨어 사용자 인터페이스의 사용성이 크게 개선되지 않았음을 밝혔다.

Karat et al. (1992)는 사용성 문제들을 발견하는 상대적 효율성의 관점에서 실증적 테스트(empirical testing)과 팀 시연(team walkthrough) 방법들을 비교하고, 실증적 테스트를 통해 시연 방법에서 발견하지 못한 상대적으로 중요한 문제들을 확인할 수 있음을 보였다. Desurvire et al. (1992)는 휴리스틱 평가, 인지적 시연, 사용성 테스트를 비교하였는데, 사용성 테스트 방법이 휴리스틱 평가보다 두배나 많은 문제들을 발견하였고, 휴리스틱 방법은 인지적 시연보다 많은 문제들을 발견하였다고 보고하였다.

Henderson et al. (1995)는 사용자에 의한 평가 방법들(user-based evaluation methods) 네 가지를 서로 비교하였다. 자동 기록법, 질문지, 인터뷰, 언어 프로토콜 분석 등이 세가지 소프트웨어들(스프레드 시트, 워드 프로세서, 데이터베이스)를 평가하기 위하여 사용되었다. 사용성 문제점들을 발견할 수 있는 능력의 관점에서 각 평가 방법들의 효율성을 비교한 결과, 언어 프로토콜 분석 방법이 가장 효율적인 것으로 나타났다. Nielsen et al. (1993)은 혜택-비용 분석을 통해서 휴리스틱 방법, 정형화된 방법(GOMS), 사용자 테스트 방법들을 비교하였다. 그들은 사용자 테스트 방법이 저렴한 휴리스틱 방법에 비해 4.9배나 비용이 소요되지만 더 나은 수행도 예측을 제공한다고 하였다. 이러한 연구 결과들을 종합해 볼 때 사용성 평가 방법들간에는 각각의 장단점이 있으며, 평가 방법의 선택에 신중을 기해야 함을 알 수 있다.

<표 3>은 사용자 인터페이스 사용성 평가시 이용되는 빈도가 높은 방법들의 장단점을 비교한 것이다. 각 방법들은 수명주기의 서로 다른 단계에 응용되며, 방법들의 장점 및 단점들이 부분적으로 각각의 방법에 대응될 수 있기 때문에, 사용성 방법들은 상호 보완적인 경향이 있다. 따라서 사용성을 평가할 때 하나의 방법만을 고수하는 것은 별로 바람직하지 않다.

한편 사용성 방법의 선택은 부분적으로 평가에 활용되는 사용자들의 수에 의존한다. 가능한 적은 수의 사용자들이 활용되어야 하는 경우에는 휴리스틱 평가(heuristic evaluation), 독백 기록(thinking aloud), 그리고 관찰(observation) 방법이 바람직하다. 약간 많은 수의 사용자들이 활용되는 경우에는 수행도 측정(performance measurement)와 초점 그룹(focus groups)를 이용하는 것이 가능하고, 매우 많은 수의 사용자들이 활용되는 경우에는 설문지(questionnaires), 상호작용 기록(interaction logging), 그리고 자연스러운 사용자 피드백(user feedback)의 체계적 수집 방법 등이 고려될 수 있다. 물론, 방법들을 약간 변형하는 경우 활용되어야 할 사용자의 수도 변한다. 예를 들어, 상호 협동적인 상호작용(collaborative interaction)은 독백 기록(thinking aloud) 방법보다 두 배의 사용자들을 필요로 한다. 그러나 사용성 평가시 사용자들에게 보다 자연스러운 작업 환경을 제시할 수 있는 장점이 있다. 한편 평가에 활용될 수 있는 사용성 전문가의 경험 수준 또한 방법의 선택에 큰 영향을 미친다.

일반적으로 사용성 평가시 하나의 방법만으로는 불충분하므로 여러 방법들을 결합해서 사용한다. 다양한 사용성 방법들을 결합하는 데는 여러 가지 가능한 조합 방법들이 있을 수 있으며, 각 프로젝트의 특성들에 따라 약간 다른 조합도 가능하다. 일반적으로 유용하다고 알려진 조합은 휴리스틱 평가와 독백 기록(thinking aloud) 또는 사용자 테스트(user testing)의 형태이다. 보통 처음에는 인터페이스에 존재하는 명확한 사용성 문제점들을 발견하고 제거하기 위하여 휴리스틱 평가를 수행한다. 인터페이스에 대한 재설계(redesign)가 이루어진 후에는, 휴리스틱 평가에 의하지적되지 않은 남아 있는 사용성 문제점들을 발견하고, 반복적 설계 단계에서 결과들을 점검하기 위하여 사용자 테스트(user testing)을 수행한다. 이러한 두 가지 방법들은 서로 다른 사용성 문제점들을 발견하는 것으로 보여지며, 상호 보완적인 관계를 가진다. 결론적으로 어떠한 평가 방법들을 이용할 것인가 하는 문제는 가용 시간과 주어진 예산과 같은 외부적 요인들에 의해 제약받는다. 또한 평가자들의 경험과 지식 수준, 제품의 수명주기의 단계, 사용자 특성 등과 같은 요인들도 평가 전략을 결정하는데 영향을 미친다.

## 5. 평가 방법 선택시 고려 사항들

제품 설계 및 개발시 상호 경쟁적인 상황에서 두 번째 주자가 되어 시장 기회를 놓치는 것보다는 오히려 적은 문제들을 포함하는 제품을 출하하는 것이 더 나을 수 있다. 이것은 주어진 조건들 하에서 가능한 빠르고 저렴한 평가 방법들이 제품의 사용성 평가에 이용되어야 한다는 것을 의미한다. 따라서 어떠한 평가 방법들을 이용할 것인가에 대한 방법의 선택 문제는 다음과 같은 요인들을 고려하여 신중하게 결정되어야 할 것이다.

- 평가를 수행하는데 필요한 시간과 가용한 시간;
- 평가 비용 (장비, 요원과 시간 등);
- 평가가 수행되는 수명주기의 단계;
- 평가에 필요한 사용자들 (users)의 수;
- 필요한 평가자들 (evaluators)의 수;
- 평가자나 디자인 팀의 경험과 지식 수준;
- 얻고자 하는 자료의 유형 (정성적 자료와 정량적 자료).

## 참 고 문 헌

- [1] Bailey, R. W., Allan, R. W., and Raiello, P., Usability Testing vs. Heuristic Evaluation: A Head-to-Head Comparison, Proceedings of the Human Factors Society 36th Annual Meeting, pp. 409-413, 1992.
- [2] Desurvire, H., Kondziela, J., and Atwood, M., What is Gained and Lost When Using Evaluation Methods other than Empirical Testing, Proceedings of HCI '92, pp.1-16, 1992.
- [3] Henderson, R. D., Smith, M. C., Podd, J. and Varela-Alvarez, H., A Comparison of the Four Prominent User-based Methods for Evaluating the Usability of Computer Software, Ergonomics, 38(10), pp. 2030-2044, 1995.
- [4] Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M., User Interface Evaluation in the Real World: A Comparison of Four Techniques, In Proceedings of CHI '91, ACM, pp.119-124, 1991.
- [5] Karat, C., Campbell, R., and Fiegel, T., Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation, In proceedings of CHI'92, ACM, pp.397-404, 1992.
- [6] Mantei, M. M. and Teorey, T. J., Cost/benefit Analysis for Incorporating Human Factors in the Software Lifecycle, Communications of the ACM, 31, pp. 428-439, 1988.
- [7] McGrath, J., Methodology Matters: Doing Research in the Behavioral and Social Sciences. Original paper, 1994.
- [8] Nielsen, J., Usability Engineering, Academic Press, 1993.
- [9] Nielsen, J. and Phillips, V. L., Estimating the Relative Usability of Two Interfaces: Heuristic, Formal, and Empirical Methods Compared, In proceedings of CHI'93, ACM, pp.214-221, 1993.

<표 1> 인간공학적 수명주기에 따른 사용성 평가 방법들

Human Factors Lifecycle (Mantci & Teorey, 1988)	Usability Eng. Lifecycle stages (Nielsen, 1993)	Methods for Usability Measurement and Evaluation
<ul style="list-style-type: none"> <li>• <b>Market Analysis</b></li> <li>• Feasibility Study</li> <li>• Requirement Definition</li> <li>• <b>Product Acceptance Analysis</b></li> <li>• <b>Task Analysis</b></li> <li>• Global Design</li> <li>• Prototype Construction</li> <li>• <b>User Testing and Evaluation</b></li> <li>• System Implementation</li> <li>• Product Testing</li> <li>• <b>User Testing</b></li> <li>• Update and Maintenance</li> <li>• <b>Product Survey</b></li> </ul>	<ul style="list-style-type: none"> <li>• Consider the larger context</li> <li>• Know the user</li> <li>• Competitive analysis</li> <li>• Setting usability goals</li> <li>• Participatory design</li> <li>• Coordinated design of the total interface</li> <li>• Guidelines and heuristic analysis</li> <li>• Prototyping</li> <li>• Empirical testing</li> <li>• Empirical testing</li> <li>• Iterative design</li> <li>• Collect feedback from field use</li> </ul>	<ul style="list-style-type: none"> <li>• concept test,   • interviews</li> <li>• psychological scales</li> <li>• factor analysis</li> <li>• comparative analysis</li> <li>• usability specification</li> <li>• HTA, GTN, OSD, GOMS,</li> <li>• TAG, CCT, link analysis etc.</li> <li>• Standards</li> <li>• expert review, checklist</li> <li>• heuristic evaluation</li> <li>• simulation trials</li> <li>• iterative laboratory experiments</li> <li>• thinking aloud</li> <li>• constructive interaction</li> <li>• walkthroughs with actual users</li> <li>• autologging</li> <li>• user's attitude, questionnaire</li> <li>• physiological analysis</li> <li>• TLX</li> <li>• impact analysis</li> <li>• design rationale</li> <li>• follow-up study</li> <li>• field study</li> <li>• hot line</li> </ul>

<표 2> 평가 전략들의 분류 (McGrath, 1994)

1. Fields strategies	2. Experimental strategies
<ul style="list-style-type: none"> <li>▸ Field studies                             <ul style="list-style-type: none"> <li>Ethnography and interaction analysis</li> <li>Contextual inquiry</li> </ul> </li> <li>▸ Field experiments                             <ul style="list-style-type: none"> <li>Beta testing of products</li> <li>Studies of technological change</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▸ Experimental simulations                             <ul style="list-style-type: none"> <li>Usability testing</li> <li>Usability engineering</li> </ul> </li> <li>▸ Laboratory experiments                             <ul style="list-style-type: none"> <li>Controlled experiments</li> </ul> </li> </ul>
3. Respondent strategies	4. Theoretical strategies
<ul style="list-style-type: none"> <li>▸ Judgment studies                             <ul style="list-style-type: none"> <li>Usability inspection methods, e. g., heuristic evaluation</li> </ul> </li> <li>▸ Sample surveys                             <ul style="list-style-type: none"> <li>Questionnaires and interviews</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▸ Formal theory                             <ul style="list-style-type: none"> <li>Design theory, e.g., Norman's 7 stages</li> <li>Behavioral theory, e. g., color vision</li> </ul> </li> <li>▸ Computer simulations                             <ul style="list-style-type: none"> <li>Human information processing theory</li> </ul> </li> </ul>

<표 3> 사용성 평가 방법들의 장단점 비교

Advantages of task analytic models	Disadvantages of task analytic models
<ul style="list-style-type: none"> <li>▸ can limit designs to psychologically plausible alternatives;</li> <li>▸ can be used to decide between two alternative interface designs;</li> <li>▸ can estimate performance times for specified tasks;</li> <li>▸ can indicate "hot spots" where interface errors or delays can be expected;</li> <li>▸ can contribute to the development of training programs.</li> </ul>	<ul style="list-style-type: none"> <li>▸ Assume error-free interaction;</li> <li>▸ Most are time consuming;</li> <li>▸ Most require specialist psychological knowledge.</li> </ul>
Advantages of heuristic evaluation	Disadvantages of heuristic evaluation
<ul style="list-style-type: none"> <li>▸ quick to perform;</li> <li>▸ relatively inexpensive;</li> <li>▸ evaluators can be used to suggest possible solutions;</li> <li>▸ uncover lots of potential usability defects.</li> </ul>	<ul style="list-style-type: none"> <li>▸ several evaluations must usually be performed for maximum benefit;</li> <li>▸ ideally done by experts;</li> <li>▸ evaluators can have biased views;</li> <li>▸ can be difficult to distinguish between trivial and serious problems.</li> </ul>

<표 3> 사용성 평가 방법들의 장단점 비교 (계속)

<b>Advantages of cognitive walkthroughs</b>	<b>Disadvantages of cognitive walkthroughs</b>
<ul style="list-style-type: none"> <li>• allows detailed usability insight early in the project;</li> <li>• allows evaluation of competing solutions before development begins;</li> <li>• forces the evaluators to consider the background knowledge and environment of the intended users;</li> <li>• provides a method of mediation between the requirements and the development sides of the design team.</li> </ul>	<ul style="list-style-type: none"> <li>• only a small sub-set of tasks can realistically be evaluated;</li> <li>• requires knowledge of cognitive science terms, concepts, and skills;</li> <li>• task selection can be problematic;</li> <li>• can lead to suboptimal or erroneous solutions;</li> <li>• does not provide a global view of the interface.</li> </ul>
<b>Advantages of thinking-aloud protocols</b>	<b>Disadvantages of thinking-aloud protocols</b>
<ul style="list-style-type: none"> <li>• requires little training so designers can perform evaluations;</li> <li>• provides qualitative data about the users' cognitive processes;</li> <li>• data collection is fast.</li> </ul>	<ul style="list-style-type: none"> <li>• thinking-aloud is unnatural to most people;</li> <li>• analysing results can be difficult and time consuming;</li> <li>• value depends on analyst's skill and knowledge;</li> <li>• may not provide accurate descriptions of what is happening.</li> </ul>
<b>Advantages of video recording</b>	<b>Disadvantages of video recording</b>
<ul style="list-style-type: none"> <li>• powerful method for persuading developers that usability problems do exist;</li> <li>• useful as a backup when events become too quick to record in real-time;</li> <li>• can be shown to users (retrospective think-aloud protocols) or to independent observers;</li> </ul>	<ul style="list-style-type: none"> <li>• data analysis is time consuming;</li> <li>• technique is highly intrusive;</li> <li>• difficult to synchronise video sources with other sources such as audio and software logging data.</li> </ul>
<b>Advantages of software logging</b>	<b>Disadvantages of software logging</b>
<ul style="list-style-type: none"> <li>• completely unobtrusive to subjects;</li> <li>• data is highly accurate (e.g. time stamps);</li> <li>• cheap (except in terms of disk storage);</li> <li>• useful for recording longitudinal data about user performance.</li> </ul>	<ul style="list-style-type: none"> <li>• generates large quantities of data;</li> <li>• data events recorded are low-level;</li> <li>• sometimes difficult to work out user actions and motives from recorded events.</li> </ul>