

## Software 품질 평가에 대한 Human - Computer Interaction 관점에서의 접근

한국전자통신연구원 합동한 · 장우현 · 유지원\*

### 1. 소프트웨어 품질 개요

사람이 설계하는 모든 시스템은 그 실체가 가시적이건 비가시적이건 항상 품질이라는 속성을 지니게 된다. 품질은 설계된 시스템이 총체적으로 요구되는 여러 기준을 만족하면서 본래의 목적을 제대로 달성할 수 있는가에 대한 능력이라 얘기할 수 있다[1,2]. 소프트웨어의 품질도 이와 같은 맥락에서 바라볼 수 있다. 그런데 Gillies는 소프트웨어 품질이 여러 측면에서 일반 공산품의 품질과는 다름을 지적하고 있다 [1]. 대표적인 차이점은 소프트웨어는 물리적 형태로 존재하지 않고 사용자의 요구가 개발과정을 거치면서 끊임 없이 변화할 수 있으며 정보기술의 발전 속도가 다른 분야에 비해 무척 빠르다는 것이다. 그러나 소프트웨어의 품질이 이러한 특징을 보이는 근본적인 이유는 아마도 소프트웨어가 사람의 복잡하고 다양한 정신작업의 대표적인 산물이면서 최종 사용자의 인지적 정보처리 작업에 사용되는 제품이라는 데에 있을 것이다. 이러한 점에서 본 논문에서는 소프트웨어의 품질을 그것을 사용하는 사람으로 하여금 안전하고 능률적으로 정보처리 작업을 하도록 해 줄 수 있는 소프트웨어의 종합적이면서 잠재적인 능력이라 정의하고자 한다.

소프트웨어의 품질을 설계하고 평가하기 위해 여러 연구자들이 주로 계층적인 모델을 이용해 품질 속성을 얘기해 왔다[3-5]. 대표적인 모델이 ISO/IEC 9126에서 제공하는 모델일 것이다[6-9]. 국내 유일의 품질시험인증 기관인 ETRI S/W시험센터에서도 이 모델을 기본 모델로 활용하고 있다.

이 모델에서 소프트웨어 품질이 여섯 개의 주 품질특성[기능성(functionality), 신뢰성(reliability), 사

용성(usability), 효율성(efficiency), 유지보수성(maintainability), 이식성(portability)]으로 이루어지고 각 특성은 다시 세부적인 부 품질특성으로 나누어진다. 각 부 품질특성은 그 것을 정량적으로 측정할 수 있는 품질 척도로 이루어져 있다. 각 품질특성을 소프트웨어의 요구 분석 관점에서 생각해 볼 때, 기능성은 기능적 요건과 밀접한 관련이 있고, 나머지 특성들은 비기능적 요건과 관련이 있다[10]. 시스템 이론 관점에서 볼 때 기능적 특성은 하향식(top-down) 분석으로 그 구조와 특징을 파악할 수 있지만 비기능적 특성은 시스템의 출현적 속성(emergent features)으로 간주되기 때문에 전체적/단일적(holistic) 관점에서 분석이 가능하다 할 수 있다. 또한 소프트웨어의 종류에 따라 다를 수 있지만 현재 품질인증을 위한 시험을 하면서 경험하게 되는 것 중의 중요한 한 가지는 기능성, 사용성, 신뢰성 등이 상대적으로 평가하기 어려운 반면에 나머지 속성들은 비교적 용이하다는 점이다.

한편 소프트웨어 생명 주기에 따라 품질을 각각 다른 관점에서 바라볼 필요가 생긴다. ISO/IEC 14598-1에서는 그림 1과 같이 소프트웨어 생명 주기의 관점에서 여러 품질 개념을 기술하고 있다[11]. 각 품질 개념의 정확한 의미는 참고문헌을 참조하기 바란다.

그림 1에서 중요한 사실은 소프트웨어의 품질 향상의 최종 목표는 사용 품질(quality in use)과 관련 있다는 점이다. ISO/IEC 9126-4에 의하면 사용 품질이 효과성(effectiveness), 생산성(productivity), 안전성(safety), 만족성(satisfaction)의 네 특성으로 이루어진다[9]. 사용 품질은 6개의 주 품질특성에 의해 모두 영향을 받으면서 다른 여러 상황적 요소에 의해서도 결정지어지는 품질이다. 이런 의미에서 6개 주 품질특성 중의 하나인 사용성(usability)과는 약간 다른

\* 정희원

의미를 지닌다고 할 수 있다. 사용성은 소프트웨어 자체의 문제이고 사용 품질은 보다 큰 정보처리 상황에서 이해해야 하는 개념이다. 이에 대해서는 2장에서 보다 상세히 기술하도록 한다.

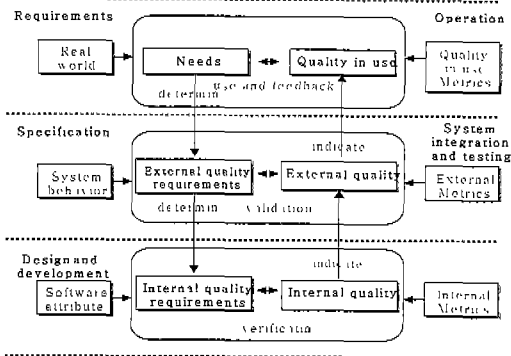


그림 1 소프트웨어 생명주기와 품질

## 2. HCI 관점에서의 소프트웨어 품질과 사용성

앞에서 기술한 바와 같이 소프트웨어의 품질은 최종적으로 사용 품질로 측정되고 평가된다. ISO/IEC 9126에서 정의하는 주 품질특성 중에서 사용성이 소프트웨어 자체만의 문제로 볼 수 있다면 사용 품질은 소프트웨어를 사용자가 실제 상황에서 사용할 때의 사용성 문제로 차이를 두어 생각할 수 있다[14-16]. 사용 품질이 6개의 주 품질특성 중에서 사용성과 가장 밀접한 관련이 있음은 분명하고, 또한 5개의 나머지 다른 특성들은 궁극적으로 사용성에 영향을 미치는 관련성을 갖고 있다. 이렇듯 사용성이 점점 중요해지면서 ISO/IEC에서도 사용 품질이라는 새로운 개념에 대한 표준을 개발하고 있는 것이다[9]. 그림 2는 여러 품질 개념과 정보처리 시스템의 범위와의 관계를 개념적으로 설명해준다.

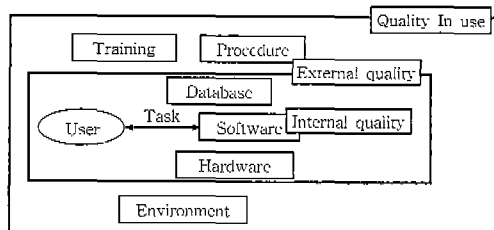


그림 2 각 품질 개념과 정보처리시스템 범위와의 관계

본 논문에서 사용 품질과 사용성을 위에 설명한 것처럼 의미상 구별하고는 있으나 실제적으로 두 개념을 구별한다는 것이 쉽지 않고 또 구별할 필요성도 없는 경우가 많다(앞으로 본 논문에서 소프트웨어 사용성이라 하면 사용 품질의 개념을 포함한 것으로 생각해도 무방할 것이다). 중요한 문제는 소프트웨어의 품질을 설계하거나 평가할 때 소프트웨어 자체만의 문제로 간주해서는 안되고 소프트웨어가 최종적으로 사용되는 실제 환경에서의 문제로 고려되어야 한다는 점이다[17,18]. 이 점은 소프트웨어가 일상 생활뿐만 아니라 전문적인 상황에서 그 비중을 더해가면서 더 강조될 것이다. 이러한 이유로 HCI가 소프트웨어 품질 평가에서 핵심적인 역할을 할 것으로 판단된다.

널리 알려진 바와 같이 HCI는 사람과 컴퓨터간의 상호작용을 향상시키기 위한 도구를 개발하고, 상호작용의 특성을 정보처리 관점에서 분석해서 사람과 컴퓨터로 이루어지는 전체 정보처리 시스템의 성능을 제고하기 위해 출현한 학문이다[19]. 도구 개발에 관한 연구로는 음성인식 인터페이스, 가상현실, 자연어 인터페이스, 포인팅 디바이스 등이 포함된다. 반면 후자의 시스템 분석 및 통합에 관한 연구에는 인지적 직무분석, 사용자의 지식 구조 및 사고모형, 사용자의 학습 패턴, 전문가와 초심자의 특성 및 차이, 사용자와 컴퓨터간의 상호작용 모형화, 정보의 구조와 시각화를 중심으로 한 사용자 인터페이스의 설계 및 평가 등이 포함된다. 또한 이러한 연구들에 기반해서 시스템 사용성에 영향을 미치는 요소가 무엇이고 이 요소들이 어떤 연관성을 갖고 있고 사용성을 높이기 위해 이 요소들을 어떻게 분석, 설계, 평가할 것인가에 대한 연구 결과들이 HCI분야에서 많이 이루어져 왔다[20]. 그러므로 소프트웨어의 품질 평가와 관련이 있는 것은 후자에 속하는 HCI 연구분야들 일 것이다.))

1) 후자의 HCI 기술의 핵심 역할을 하는 학문이 인지시스템공학(cognitive systems engineering)이다[20,21]. 인지시스템공학은 자연적인 정보처리 시스템인 사람과 인위적인 정보처리 시스템인 컴퓨터로 이루어지는 사람-컴퓨터 시스템(human-computer system)을 분석, 설계, 평가하기 위해 필요한 개념, 방법론, 기법, 도구 등을 연구하는 시스템 공학에 근간을 두는 학문이다[22,23]. 그러나 여기서는 편의상 인지시스템공학의 개념을 포함하는 후자의 HCI 기술을 그냥 HCI로 부르코자 한다. 이렇듯 HCI는 소프트웨어의 사용 품질 내지는 사용성 평가를 위해 유용한 개념적 혹은 방법적 배경지식을 제공해 준다.

소프트웨어 사용성 내지는 사용 품질을 이해하는데 가장 기본이 되는 것이 사용 상황(use context)에 영향을 주는 요소들을 파악하고 분류체계 (taxonomy)를 갖는 것이다[15]. 이 요소들은 소프트웨어 품질을 설계하거나 평가하는데 관심의 대상이 되지만 이러한 다양한 요소들이 무엇이 있는가를 아는 것 만으로는 큰 도움이 되지를 않는다. 이 요소들이 어떤 상황에서 어떤 구조로 연결되어 있으며 사용성과 안전성 높은 소프트웨어 내지는 시스템을 설계하기 위해 어떤 순서로 분석이 이루어져야 하는가와 평가할 때의 평가 범위와 방법의 선정 및 결과 해석을 이 요소들에 기반해서 어떻게 할 것인가에 대한 방안이 강구되어야 한다. 이에 대한 개념적인 연구가 HCI 분야에서 많이 이루어져 왔다[22-25]. 이 연구들에 기반해서 소프트웨어 품질 설계 및 평가에 활용할 일종의 template을 개발해 활용하는 것도 큰 의미가 있을 것으로 판단된다.

### 3. 기존의 HCI 연구 결과들과 소프트웨어 품질평가에의 활용

이 장에서는 소프트웨어 품질 평가와 관련이 있는 HCI 분야의 연구들을 소개하고자 한다. 우선적으로 소프트웨어 사용성에 관련된 ISO 국제표준을 설명하고자 한다. 사용성에 관한 표준은 설계부터 평가에 이르는 개발과정 동안 사용성을 어떻게 다룰 것인가에 대한 표준(Process oriented standards)과 소프트웨어 속성의 사용성 혹은 소프트웨어의 성능 요건에 관한 표준(Product oriented standards)으로 나누어진다[26]. 소프트웨어 품질 평가와 보다 밀접한 관련이 있는 것은 후자에 해당하는 표준들일 것이다. 대표적으로 ISO/IEC 9241 시리즈에서는 GUI 소프트웨어의 사용자 인터페이스를 구성하는 요소들에 대한 설계 원칙 및 가이드라인을 제공한다. 물론 이 원칙 및 가이드라인이 평가 목적으로도 활용될 수 있다. 소프트웨어 사용성에 관련해 현재 개발되어 널리 사용되고 있는 국제표준이 어떤 것이 있는가는 참고문헌 [26]을 참조하기 바란다.

사용자 인터페이스에 관한 설계 혹은 평가 가이드라인은 현재 상당히 많이 존재하며 현장에서 많이 사용되고 있다[27-29]. 그러나, 여러 연구들이 이 가이드라인이 현실적으로 많은 도움을 주지 못하고 있다는 보고를 하고 있다[30,31]. ETRI의 S/W시험센터

의 시험원들도 실제적으로 소프트웨어 사용성을 평가할 때 일반적인 가이드라인 형태로 되어 있는 품질 검사항목을 이용함에 큰 불편함을 호소하고 있다. 이러한 현상으로부터 현재의 가이드라인의 정보 구조가 평가자들이 작업할 때 사용하는 지식의 구조와 잘 맞지 않음을 짐작할 수 있다. 이러한 문제점을 해결하고자 현재 이 분야에 활발한 연구가 이루어지고 있으며 조만간 현재와는 다른 형식의 새로운 설계/평가 가이드라인이 나올 것으로 예상된다.

품질 인증기관 같은 곳에서는 여러 부류의 소프트웨어를 평가해야 한다. 이 경우 가장 어려움을 겪는 것이 소프트웨어를 평가하기 위해 소프트웨어가 사용되는 직무 혹은 작업영역(work domain)을 이해할 필요가 생긴다는 점이다. 예로 화학공장 제어용 소프트웨어라든가 금융산업 관련 소프트웨어 등을 평가할 때는 화학공장이 운영되는 메커니즘을 이해해야 하고 금융산업에서 행해지는 주요 직무 등을 이해해야 한다. 평가를 위한 이러한 분석 과정이 결코 쉬운 일이 아님은 주지의 사실이다. 흔히들 직무분석 내지는 기능분석이라고 부르는 과정이 필요하다는 것인데, 이를 위해 필요한 기법 내지는 방법론에 대한 연구도 HCI 분야에서 활발하게 진행되고 있다[23,24].

이와 관련해 가장 중요하게 언급되는 개념이 시스템의 추상화 계층(abstraction hierarchy)이다. 추상화 계층은 시스템의 목적을 달성하기 위해 어떠한 기능들이 구체적으로 수단-목적(means-ends) 관계에 의해 구현이 되었는가를 설명해주는 것으로 동일한 시스템을 여러 단계 내지는 관점에서 바라볼 수 있도록 해준다[32]. 다시 말해서 시스템의 설계 지식을 수단-목적의 계층화된 구조로 분석할 수 있도록 해주는 개념적인 지식표현의 틀이다. Rasmussen은 추상화 계층을 다음의 다섯 단계로 구분하였다[32]: 소프트웨어 시스템의 존재 이유를 대변하는 기능적 목적(functional purpose), 소프트웨어의 기능을 정보나 에너지와 같은 추상적 개념의 흐름과 균형의 관점에서 바라보는 추상화된 기능(abstract function), 소프트웨어의 표현방식과는 무관하게 제공되는 기능과 동작들에 관한 일반적 기능(generalized function), 일반적 기능의 물리적 표현 형태를 규정하는 물리적 기능(physical function), 실제 보이는 그대로의 물리적 형태(physical form).

그러나 이러한 추상화 계층의 개념을 이용할 때 꼭 다섯 단계로 할 필요는 없다. 중요한 것은 그 속에

내재되어 있는 수단-목적의 관점으로 시스템 평가에 활용해야 한다는 점이다. 추상화 계층이 중요한 이유 중의 하나는 실제로 사용자들이 시스템을 접하고 그것을 다룰 때 이러한 추상화 관점에서 정보처리를 한다는 점이다. 이러한 사실은 소프트웨어 사용성을 평가할 때 추상화 계층이 시스템이 얼마나 복잡도를 줄여가며 체계적으로 설계가 되었는가를 평가하는데 좋은 개념적 도구가 됨을 의미한다.

소프트웨어와 사람이 인지적인 상호작용을 할 때의 정보처리 특성도 또한 소프트웨어 평가자들이 알고 있어야 할 중요한 지식이라 생각된다. 여기에는 인간의 장단기 기억의 기능, 지식의 형태 및 구조, 시스템에 대한 사고 모형(mental models), 복잡도에 대응하는 문제해결 기술 및 인지적 편향, 직무처리 숙련 정도에 따른 정보처리의 차이, 인적 오류 등이 포함된다[33-35]. 여기서 모두 다 구체적으로 설명할 수는 없고 하나의 예를 들어 보겠다.

사람이 갖고 있는 지식구조에서 중요한 특징 중의 하나는 사물 혹은 작업순서의 주관적 복잡도를 줄이고자 스키마(schema)라는 것을 갖고 있다는 점이다. 스키마는 비슷한 사물들의 공통된 속성에 대해 갖고 있는 일종의 이미지라고 보면 된다. 우리가 서로 다른 회사의 동일한 종류의 소프트웨어(가령, 워드 프로세스)를 사용할 때 무슨 기능이 어떠한 메뉴 구조로 이루어져 있겠구나라고 짐작하고 작업을 하는 것도 사실은 워드프로세스라는 소프트웨어에 대해 갖고 있는 스키마 때문이다. 이 현상과 밀접한 관련이 있는 소프트웨어 설계 원칙이 일관성이다. 소프트웨어의 기능을 표시해주는 명칭(label), 기능의 동작 등이 일관성이 있을 때 그만큼 사용자는 작업하는데 수월함을 느끼게 된다. 한 제품 내에서도 비슷한 작업 절차를 갖게 되는 기능들이 있다(예로, 워드 프로세스에서 각주 및 미주 삽입, 캡션 삽입, 상호참조 삽입 등). 이러한 기능들의 작업 절차(삽입 위치 지정-> 해당 삽입 기능 선택 -> 삽입내용 작성)의 일관성이 깨진다면 사용자는 불편함을 느끼게 될 것임은 자명하다.

HCI에서 연구되어온 내용 중에 실제 소프트웨어 품질 평가자들이 가장 관심을 갖는 것이 사용성 평가 방법일 것이다. 일반적으로 소프트웨어의 품질을 평가할 때 주 품질 특성별로 평가를 하게 되는데 가장 어렵고 시간을 많이 요하는 것이 사용성이다. 다음 장에서 이에 관해 구체적으로 설명하기로 한다.

## 4. 사용성 평가 방법

### 4.1 평가 방법의 유형

일반적으로 사용성 평가의 방법을 다음의 3가지 유형으로 분류할 수 있다[36].

- 시험(testing): 대표성을 지닌 사용자로 하여금 실제 시스템 혹은 프로토타입을 대상으로 모의 환경에서 소프트웨어를 사용하는 대표적인 직무를 수행하게 해서 그 수행도를 관찰하는 방법

- 분석적 조사(inspection): 사용성 전문가(그러나 경우에 따라서는 개발자 내지는 사용자가 될 수도 있음)가 일정한 사용성에 관한 기준을 갖고 분석적으로 평가하는 방법

- 사용자 조사(inquiry): 평가자가 실제 사용자의 주관적 만족도를 사용자와의 인터뷰를 통해서 혹은 실제 환경에서 사용자가 소프트웨어를 사용하는 것을 관찰해서 혹은 사용성의 특징을 잘 정리한 설문지를 통해 조사하는 방법

표 1은 각각의 유형에 어떠한 방법들이 포함되는가를 정리한 것이다[37]. 여기서는 각각의 방법을 구체적으로 설명할 수 없으므로 관심이 있는 각각의 방법에 대해서는 참고문헌 [37]을 참고하기 바란다. 현재 ETRI의 S/W시험센터와 같이 제 3자 입장에서 다양한 종류의 소프트웨어 품질인증을 위해 사용성을 평가해야 하는 곳에서는 현실적으로 분석적 조사에 의존할 수 밖에 없다(구체적으로는 Feature inspection).

표 1 소프트웨어 사용성 평가의 방법들

시험	분석적 조사	사용자 조사
-Coaching method	-Cognitive	-Field observation
-Co-discovery learning	walkthroughs	-Focus groups
Performance measurement	Feature inspection	-Interviews
-Question-asking protocol	-Heuristic evaluation	-Logging actual use
-Remote testing	-Pluralistic evaluation	Proactive field study
-Retrospective testing	Perspective-based inspection	-Questionnaires
Shadowing method		
-Teaching method		
-Thinking aloud protocol		

평가자는 표 1에 나와 있는 다양한 방법의 장단점 및 특성을 모두 숙지한 상태에서 실제의 사용성 평가 상황을 고려하여 적합한 방법을 선택해야 할 것이다. 각 방법의 장단점 및 선택 기준에 대해서는 참고문헌 [38,39]에 상세히 기술되어 있다. Zhang[37]은 위와 같은 요소에 기반해서 가장 적합한 평가 방법을 제안하는 자동화된 프로그램을 web상에서 제공하고 있다.

한편 사용성은 사용자가 소프트웨어를 사용해서 얼마나 잘 직무를 수행하는가에 대한 객관적인 수행도를 실험적 방법에 의해 측정해서 평가하거나, 사용자의 주관적인 만족도를 조사해 평가할 수 있다. 대별되는 두 척도는 그 측정 방법이 다를 수 밖에 없으므로 나누어 설명을 한다.

## 4.2 사용자의 수행도 측정

앞에서 설명한 것처럼 사용 품질은 효과성, 생산성, 안전성, 만족성의 4개의 품질특성으로 이루어진다. 이 중에서 사용자의 직무 수행도 관점에서 실험적 방법에 의해 객관적으로 정량적인 결과를 얻을 수 있는 것은 효과성과 생산성일 것이다. 안전성은 어느 정도 정량적인(정확히는 확률적인) 결과를 얻어낼 수 있지만, 실제 사용에서는 인간 신뢰도 및 오류와 소프트웨어 안전성을 통합적으로 고려해야 하는 만큼 정성적으로 평가해야 하는 부분도 상당 부분 존재한다. 만족성의 경우는 주로 정성적인 결과를 낼 수 밖에 없음을 당연한 사실일 것이다.

사용성에 대한 실험적 평가는 그 중요한 인자로 사람을 포함하고 있기 때문에 실험방법부터 결과의 해석까지 상당히 철저하게 계획되고 일관되게 이루어져야 한다[25]. 사람이라는 다양성을 지닌 요소로부터 가급적 실험의 발생 가능한 잡음을 배제시키고 실험의 순도를 높이기 위해서는 사람에 대한 지식과 더불어 통계적 방법에 의한 실험계획의 지식을 모두 갖춘 사용성 전문가로 하여금 사용성 실험을 하도록 하는 것이 바람직하다. 이는 그만큼 사용성 평가가 어렵다는 것을 의미하기도 하며 동시에 HCI 전문가가 앞으로 소프트웨어 품질 평가에 앞으로 더 큰 역할을 하게 될 것으로도 해석된다. 그러나 실험에 의한 평가가 보다 의미가 있고 결정적인 결과를 제공한다는 데에는 이견이 없으나 수반되는 비용 및 여러 제약으로 인해 어려운 것이 현실이다. 더욱이 앞서 언급한 것처럼 품질 인증을 위한 시험기관과 같은 곳

에서는 더욱 어렵다 할 수 있다. 따라서 분석적 방법과 실험적 방법을 절충한 방법을 고안해서 활용하는 것도 매우 바람직할 것으로 생각된다.

## 4.3 사용자의 주관적 만족도 측정

사용자의 수행도 측정 방법에 비해 상대적으로 사용자의 주관적 만족도를 측정하는 방법에 대해서는 많은 연구가 이루어지지 않았다[40]. 그러나 사용자의 주관적 만족도를 측정하는 것은 두 가지 관점에서 의미가 있다. 첫째, 실제적으로 최종 소프트웨어를 사용하는 주체인 소비자가 자신의 경험에 근거한 품질 자료를 제공해준다는 점에서 다른 기타 방법보다 종합적인 품질 척도를 얻을 수 있는 기회를 제공한다. 둘째, 소프트웨어의 품질 특성들에 대한 타당성 검증 및 그 중요도에 관한 유용한 자료를 제공해줄 수 있다.

표 1에서 설명한 바와 같이 주관적 만족도를 측정하기 위해 여러 방법이 활용 가능하나 품질 인증 및 척도의 타당성 검증 측면에서 설문에 의한 방법(questionnaire)이 가장 관심의 대상이 된다. Perlman[41]은 현재까지 주관적 만족도를 측정하기 위해 개발된 방법들을 정리하고 그 것에 기반해서 주관적 만족도를 웹상에서 평가할 수 있는 기능을 제공하고 있다. 가장 대표적인 방법으로는 SUMI[17](Software Usability Measurement Inventory), PUEU[42](Perceived Usefulness and Ease of Use), CSUQ[43](Computer System Usability Questionnaire), PUTQ[44](Purdue Usability Testing Questionnaire) 등이 있다.

현재까지 개발된 각 방법들에 포함된 설문 항목들이 모두 다르고 그 수준에 있어서도 차이를 보이고 있다. 각 방법들에서 이용하는 항목들을 모두 종합하여 실제적인 측정 결과를 이용해 항목들간의 관련성을 규명해 보다 계층적으로 구조화된 설문 체계를 만드는 것이 추후 연구로 필요할 것이다. 또한 ETRI의 S/W시험센터와 같은 품질인증 기관에서는 앞으로 사용자를 대상으로 하여 주관적 품질 평가의 결과를 소프트웨어 품질 인증여부에도 활용할 뿐만 아니라 시험 과정에서 이용되는 평가 항목 선정에도 활용해야 할 것으로 판단된다.

한편, 만족도는 아니지만 소프트웨어의 기능내지는 구조적 복잡도가 얼마나 사용자로 하여금 인지적

부담(cognitive workload)을 느끼게 하는가를 사용자의 주관적 관점에서 측정하기도 한다. 이를 위해 많이 사용되는 것이 NASA에서 개발된 TLX(Task Load Index)이다[45]. 복잡도에 기인하는 요소를 6가지(Mental demand, Physical demand, Temporal demand, Effort, Performance, and Frustration level)로 분류해서 각각의 부하도에 근거해 종합적인 인지적 부하도를 산출하게 된다.

## 5. 결 론

본 논문에서는 소프트웨어 품질 모델 및 품질 특성을 고찰하고 여러 특성 중에서 사용성이 점점 중요해지고 있음을 알 수 있었다. 또한 사용성이 소프트웨어 자체의 설계문제가 아니라 사용자와 직무, 소프트웨어 및 하드웨어, 그 외 환경적 요소를 포함하는 인간-컴퓨터 정보처리 시스템의 맥락에서 이해되어야 하는 요소임을 설명하였다. 이러한 관점에서 HCI 분야의 기존의 연구 결과들을 고찰해보고 이들이 어떻게 소프트웨어 품질 및 사용성 문제에 연결되어 활용될 수 있는가를 토의하였다. 소프트웨어의 품질 특성 중에서 사용성에 대한 평가를 실제 평가자들이 가장 어려워한다는 점을 고려해서 현재까지 이용되고 있는 다양한 사용성 평가 방법을 고찰과 비교해보았다.

본 논문에서 소개된 HCI 분야의 중요 연구 결과는 다음과 같다: 소프트웨어 사용성에 관련된 ISO 국제 표준, 사용자 인터페이스의 설계 원칙 및 가이드라인의 구조 및 활용 방안, 시스템의 추상화 계층, 시스템과의 상호작용 관점에서의 인간의 보편적인 정보처리 특성들(장/단기 기억의 기능, 지식의 형태 및 구조, 사고모형, 복잡도에 대응하는 문제해결 기술 및 인지적 편향, 직무처리 숙련정도에 따른 정보처리의 차이, 인적오류 등), 사용성 평가 방법. 이 연구들은 소프트웨어 품질 평가의 문제를 전체 정보처리시스템의 성능 계고의 측면에서 바라보고 접근하는데 직접적인 개념적 혹은 방법적인 배경지식을 제공해 줄 것으로 여겨진다. 그렇지만 어떻게 HCI 분야의 연구 내용들을 소프트웨어 품질 평가에 보다 구체적으로 활용할 것인가에 대해서는 실제적인 품질 평가 활동을 통해 더 강구되어야 할 것으로 보인다.

소프트웨어 품질의 향상은 제품자체의 평가활동만으로는 지속적이고 체계적인 향상을 이룩하기 어렵다. 소프트웨어의 설계를 위한 분석단계에서부터

소프트웨어를 실제적으로 사용할 사용자 및 그들의 직무, 소프트웨어 및 하드웨어의 기술적 능력, 국내 외적인 규제 및 표준화 요건 등에 대한 종합적인 안목을 갖고 소프트웨어 품질 문제가 다루어져야 한다. 이를 위해 소프트웨어 개발기술에 관한 소프트웨어 공학, 소프트웨어 사용의 최적화를 통한 전체 인간-컴퓨터 정보처리 시스템의 성능 향상에 관한 HCI 및 인지시스템공학을 유기적으로 연결해서 소프트웨어 생명주기에 걸친 전 활동에 활용해야 한다.

## 참고문헌

- [1] Gillies, A., Software Quality: Theory and Management, Chapman & Hall Computing, 1992.
- [2] Juran, J.M., Quality Control Handbook (3rd edition), McGraw-Hill, 1979.
- [3] Boehm, B., Brown, J.R., Kasper, H., Lipow, M., McLeod, G., and Merritt, M., Characteristics of Software Quality, North-Holland, 1978.
- [4] Stephen, H.K., Metrics and Models in Software Quality Engineering, Addison-Wesley, 1995.
- [5] Kitchenham, B. and Pfleeger, S.L., "Software quality: the elusive target", IEEE Software, January, pp. 12-21, 1996.
- [6] ISO/IEC DTR 9126-1: Software engineering-Software product quality-Part 1: Quality model, 1998.
- [7] ISO/IEC DTR 9126-2: Software engineering-Software product quality-Part 2: External metrics, 2001.
- [8] ISO/IEC DTR 9126-3: Software engineering-Software product quality-Part 3: Internal metrics, 2001.
- [9] ISO/IEC DTR 9126-4: Software engineering-Software product quality-Part 4: Quality in use metrics, 2001.
- [10] Sommerville, I., Software Engineering (6th edition), Addison-Wesley, 2001.
- [11] ISO/IEC 14598-1: Information technology-Software product evaluation-Part 1: General overview, 1999.
- [12] Bollox, G. and Robillard, P.N., "A software

- system evaluation framework”, IEEE Computer, December, 1995.
- [13] Raman, S., “It is software process stupid: next millennium software quality key”, IEEE AES Systems Magazine, June 2000.
- [14] Bevan, N., “Quality in use: incorporating human factors into the software engineering lifecycle”, in Proceedings of the third international symposium and forum on software engineering standards, August 1997.
- [15] Bevan, N and Macleod, M., “Usability measurement in context”, Behavior and Information Technology, Vol. 13, pp. 132-145, 1994.
- [16] Bevan, N., “Quality in use: meeting user needs for quality”, The Journal of Systems and Software, Vol. 49, pp. 89-96, 1999.
- [17] Bevan, N., “Measuring usability as quality of use”, Journal of Software Quality, Vol. 4, pp. 115-140, 1995.
- [18] Bevan, N., “Usability is quality of use”, in Proceedings of 6th international conference on human computer interaction, July 1995.
- [19] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T., Human-Computer Interaction, Addison-Wesley, 1994.
- [20] Helander, M. (eds.), Handbook of Human-Computer Interaction, North-Holland, 1988.
- [21] 윤완철, HCI에 관한 인지공학적 연구의 방향, 한국정보과학회지, 제 9권, 제 5호, pp. 15-25, 1991.
- [22] Rasmussen, J., Pejtersen, A.M., and Goodstein, L.P., Cognitive Systems Engineering, John Wiley & Sons, New York, 1994.
- [23] Vicente, K.J., Cognitive Work Analysis, Lawrence Erlbaum Associates, New Jersey, 1999.
- [24] Hackos, J.T. and Redish, J.C., “User and Task Analysis for Interface Design”, John Wiley & Sons, 1998.
- [25] Rubin, J., Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, John Wiley & Sons, 1994.
- [26] Bevan, N., “Human-computer interaction standards”, in Proceedings of 6th international conference on human computer interaction, July 1995.
- [27] Mayhew, D.J., Principles and Guidelines in Software User Interface Design, Prentice-Hall, 1992.
- [28] Apple Computer Inc., Aqua Human Interface Guidelines (Macworld draft), Apple Computer, 2001.
- [29] Schumacher, R.M., Ameritech Graphical User Interface Standards and Design Guidelines, Ameritech Services, 1996.
- [30] Tetzlaff, L. and Schwartz, D.R., “The use of guidelines in interface design”, in Proceedings of the Conference on Human Factors in Computing Systems CHI’91, New York: ACM Press, pp.329-333. 1991.
- [31] Thovtrup, H. & Nielsen, J., “Assessing the usability of a user interface standard”, in Proceedings of CHI ’91. Boston: MA, pp.335-341. 1991.
- [32] Rasmussen, J., The role of hierarchical knowledge representation in decision making and system management, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, pp. 234-243, 1985.
- [33] Wickens, C.D., Engineering Psychology and Human Performance (2nd edition), HarperCollins Publishers, New York.
- [34] Wickens, C.D., Gordon, S.E., and Liu, Y., An Introduction to Human Factors Engineering, Longman, 1998.
- [35] Norman, D.A, The Psychology of Everyday Things, Basic Books Inc., 1988.
- [36] <http://www.best.com/~jthom/usability/>
- [37] <http://www.cs.umd.edu/~zzj/UsabilityHome.html>
- [38] [http://www.userdesign.com/docs/uem\\_compare.html](http://www.userdesign.com/docs/uem_compare.html)

- [39] <http://www.dcs.napier.ac.uk/marble/Usability/Table.html>
- [40] Han, S.H., Yun, M.H., Kim, K-J, and Kwahk, J, Evaluation of product usability: development and validation of usability dimensions and design elements based on empirical models, International Journal of Industrial Ergonomics, Vol. 26, pp. 477-488, 2000.
- [41] <http://www.acm.org/~perlman/question.html>
- [42] Davis, F.D., Perceived usefulness, perceived ease of use, and user acceptance of information technology, MIS Quarterly, Vol. 13(3), pp. 319-340, 1989.
- [43] Lewis, J.R., IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use, International Journal of Human-Computer Interaction, Vol. 7(1), pp. 57-78, 1995.
- [44] Lin, H.X., Choong, Y.Y., and Salvendy, G., "A proposed index of usability: a method for comparing the relative usability of different software systems", Behavior and Information Technology, Vol 16, pp. 267-278, 1997.
- [45] <http://iac.dtic.mil/hsiac/products/tlx/toc.html>



**함 등 한**

1993 인하대학교 산업공학과 공학사  
 1995 KAIST 산업공학과 공학석사  
 2001 KAIST 산업공학과 공학박사  
 1997~2001 산업공학과 연구조교  
 현재 ETRI 정보화기술연구본부 S/W 시험센터, 선임연구원  
 관심분야 : cognitive systems engineering, human-computer interaction, systems and software engineering

E-mail: dhham@etri.re.kr



**장 우 현**

1982.2 중앙대학교 전자계산학과 졸업(학사)  
 1991.8 중앙대학교 컴퓨터공학과 대학원 졸업(석사)  
 2001.2 대전대학교 컴퓨터공학과 대학원 졸업(박사)  
 1982.3~현재 한국전자통신연구원 (ETRI) 책임연구원, S/W 평가기술팀장  
 관심분야 : 소프트웨어공학, 컴포넌트 기술, e-Business

E-mail: whjang@etri.re.kr



**유 지 원**

1998.8 아주대학교 산업공학과 졸업(학사)  
 2001.2 한국정보통신대학원대학교(ICU) 공학부 졸업(석사)  
 1993.3~1999.12 한국통신 통신망연구소 위촉연구원  
 2001.3~현재 한국전자통신연구원 (ETRI) S/W 시험센터 연구원  
 관심분야 : 소프트웨어시험, 프로토타입 시험, 차세대 인터넷

E-mail: jwryu@etri.re.kr