

# UI 중심의 소프트웨어 사용성 평가 방법

이하용\*, 양해솔\*\*

서울벤처대학원대학교 융합산업학과\*, 호서대학교 벤처전문대학원 정보경영학과\*\*

## The Evaluation Method of Software Usability based on UI

Ha-Young Lee\*, Hae-Sool Yang\*\*

Dept. of Fusion Industry, Seoul Venture University\*

Dept. of Information Management, Graduate School of Venture, Hoseo University\*\*

**요약** 소프트웨어의 품질을 바라보는 관점이 다양한 특성으로 나타날 수 있으나, 소프트웨어의 내부적인 면에는 관심이 없는 엔드유저 입장에서는 소프트웨어의 사용자 인터페이스(User Interface)가 소프트웨어의 품질을 표현하는 전부일 수도 있다. 그러나 사용자 인터페이스에 기반을 둔 상세화된 평가기준은 다소 미흡한 실정이므로 소프트웨어의 사용자 인터페이스는 사용성 품질 특성의 주요 대상이라 할 수 있다. 본 논문에서는 소프트웨어의 품질평가에 관련된 국제표준의 품질특성들과 사용자 인터페이스를 구성하는 요소들과의 관련성을 명확히 분석하여 소프트웨어의 사용자 인터페이스로부터 측정할 수 있는 품질특성 요소들을 추출하고 평가하는 방법에 대한 체계를 구축하였다.

**주제어** : 사용자 인터페이스, 품질특성, 사용성, 품질평가, 점검표

**Abstract** Viewpoint about the quality of software is various. But, from the end-user point of view, user interface of software may be all to express the quality of software. But the detailed evaluation criteria were not established about usability evaluation method based on user interface so far, though user interface of software is the main object of usability quality characteristics. In this paper, we established a system about the method extracting and evaluating some quality characteristics elements measurable form user interface of software by analyzing the relationship between the elements constructing quality characteristics and user interface related to quality evaluation of software. We expect that this result of study will be a fundamental study adapting and evaluating to various types of user interface.

**Key Words** : User Interface, Quality Characteristics, Usability, Quality Evaluation, Check Table

### 1. 서론

소프트웨어의 품질은 소프트웨어의 유용성과 활용성을 좌우하는 핵심적인 요소라고 할 수 있다. 국내에서도

소프트웨어의 품질 향상을 위한 노력의 일환으로 한국정보통신기술협회(TTA)에서 1998년 11월에 국제규격인 ISO/IEC 9126을 기반으로 TTAS.IS-9126.1(소프트웨어 품질특성 및 매트릭 - 품질특성 및 부특성)을 제정하고

Received 1 May 2013, Revised 20 May 2013

Accepted 20 May 2013

Corresponding Author : Hae-Sool Yang(Graduate School of Venture, Hoseo University)

Email: lhyazby@suv.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1738-1916

2001년 4월에 KS X 2216-1(정보기술-소프트웨어 제품 품질)을 제정하는 것을 시작으로 지속적인 품질평가 분야의 발전을 위해 노력하고 있다.

소프트웨어 품질특성에 관해 ISO/IEC 9126의 국제표준에서는 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성 등으로 정의하고 각 품질특성에 관한 부특성을 정의하여 총 27개의 부특성을 정의하고 있다.

소프트웨어의 품질을 바라보는 관점은 다양한 특성으로 나타날 수 있으나, 소프트웨어의 내부적인 면에는 관심이 없는 엔드유저 입장에서는 소프트웨어의 사용자 인터페이스(User Interface)가 소프트웨어의 품질을 표현하는 전부일 수도 있다. 소프트웨어의 사용자 인터페이스는 사람과 정보매체의 접점 또는 채널이라 표현하며 그 개념은 보다 사용하기 편한 시스템을 만들기 위해 사용자의 인지적 측면에서 디자인하고 사용 편리성을 평가하는 것이다[1].

다시 말해서 정보를 제공하기 위해 물리적 컨트롤러, 콘텐츠 표현을 보다 친근하게, 쉽게, 모두가 잘 사용할 수 있게 하는 것이 사용자 인터페이스의 영역이라 할 수 있으며 이러한 사용도구 및 환경 사이에서 일어나는 물리적, 인지적 문제들을 해결함으로써 사용자와 시스템 사이의 간격을 줄이는 행위라 할 수 있다[2].

소프트웨어 품질평가 분야는 소프트웨어 품질특성에 관한 국제표준 제정을 기점으로 하여 품질평가 체계를 구축하는 면에서는 많은 발전을 이루었으나 평가항목에 대한 구체적인 측정방법 측면에서는 아직까지 미흡한 점이 많이 남아 있는 것이 현실이다.

본 논문에서는 소프트웨어의 품질평가에 관련된 국제표준의 품질특성인 ISO/IEC 9126[3]과 ISO/IEC 12119[4]의 사용성 품질특성과 부특성 체계를 근간으로 소프트웨어의 사용자 인터페이스로부터 측정할 수 있는 품질특성 요소들을 추출하고 평가하는 방법에 대한 체계를 구축하고자 한다.

## 2. 사용자 인터페이스 관련 동향

### 2.1 S/W 사용자 인터페이스 발전 과정

#### 2.1.1 명령 줄 인터페이스

명령 줄 인터페이스(CLI, Command line interface) 또는 명령어 인터페이스는 텍스트 터미널을 통해 사용자와

컴퓨터가 상호 작용하는 방식이다. 즉, 작업 명령은 사용자가 컴퓨터 키보드 등을 통해 문자열의 형태로 입력하며, 컴퓨터로부터의 출력 역시 문자열의 형태로 주어진다. 명령 줄 인터페이스는 1950년대 텔레타이프라이터가 컴퓨터에 연결되어 사용된 때부터 유래된다. 즉각적인 상호작용과 피드백을 할 수 있었기 때문에 천공카드에 비해 진보한 방식으로 여겨졌다.

#### 2.1.2 GUI의 역사[5]

GUI는 하이퍼링크를 기반으로 마우스를 사용하는 NLS(On-line System)를 기원으로 하여 1973년에 개발된 제록스 알토 컴퓨터에서 최초로 구현되었다.

이후 애플 컴퓨터는 1983년에 GUI에 기반한 컴퓨터 리사(LISA)와 매킨토시를 출시했고 마이크로소프트사는 마이크로소프트 윈도를 개발하였다. 마이크로소프트는 1995년 윈도 95를 출시하여 본격적인 GUI 시대를 열어 지금에 이르고 있고 CUI(Character User Interface) OS인 유닉스(UNIX)와 그 계열의 리눅스에서는, X 윈도 시스템을 통해 그래픽 사용자 인터페이스를 구현하고 있다.

## 2.2 최신 사용자 인터페이스 기술동향[6]

최신 사용자 인터페이스는 인간의 지각능력을 직관적으로 활용할 수 있는 다양한 형태로 발전하고 있다.

#### 2.2.1 터치인식 기술

터치 인터페이스는 디스플레이 화면을 직접 터치하여 문자를 입력하거나 아이콘을 선택하거나 화면을 통해 직접 제어하는 기술이다[7].

#### 2.2.2 음성인식 기술

음성인식 기술은 1960년대부터 많은 연구가 진행되어 왔으나 아직 대중화가 미흡하고 실제 환경에서의 인식률이 떨어지는 단점이 있다.[8].

#### 2.2.3 시각인식 기술

시각정보는 사람이 받아들이는 모든 정보의 약 80% 이상을 차지한다고 한다. 시각정보를 인지하는 시각센서는 색, 모양, 조도, 거리 등 다양한 정보를 한꺼번에 인지할 수 있어 각종 인식기술 중 가장 많은 정보를 얻을 수 있다.

### 2.2.4 AR(Augmented Reality) 기술

증강현실(Augmented Reality) 기술은 카메라를 통해 들어오는 현실의 영상에 가상의 그래픽 정보를 실시간으로 결합하여 보여주는 기술로서 현실감을 증대시킨 기술이다.

### 2.2.5 모션인식 기술

가속도 센서를 이용하여 휴대폰 등에서 사람의 동작을 인식하여 반응할 수 있는 기술이다.

### 2.2.6 햅틱 기술

햅틱 기술은 시각적, 청각적 정보가 갖지 못하는, 촉각 정보만이 갖는 독특한 감성을 전달할 수 있는 기술이다.

### 2.2.7 상황인지 기술

상황인지 기술은 기계와 사람과의 커뮤니케이션에서도 사람과 대화하는 것처럼 위치, 방향, 날씨, 주변광의 세기, 몸의 흔들림, 소음 등 주변 상황을 인식하여 효과적인 인터페이스 수단을 제공하는 기술이다.[9].

## 3. 사용자 인터페이스의 품질 요구사항

### 3.1 사용자 인터페이스의 요건

#### 3.1.1 사용자 인터페이스 팩터

가. 관용성, 유연성

##### 1) 초보자와 전문가 입장 수용

초보자의 전문가의 입장을 모두 고려한 UI의 설계가 필요하다. 예를 들어 전문가를 위해 다양한 설정값을 입력할 수 있도록 하되 초보자를 위해 디폴트(default)값을 제공하는 것과 같은 식이다.

나. 숙련도에 따른 대처

##### 1) 다양한 기능 수행 방법 제공

소프트웨어에서 결과적으로 동일한 기능을 수행하더라도 소프트웨어에 대한 사용자의 사용경험이나 숙련도에 따라 사용자 인터페이스를 사용하는 방법은 다를 수 있다. 초보자의 경우 마우스를 이용하여 아이콘을 클릭하는 방법을 쓰거나 풀다운(pull-down) 메뉴를 이용하여 해당 기능을 선택하기도 하고 숙련자라면 단축키를 많이 사용하여 빠른 처리가 가능할 수도 있다.

##### 2) 전문가를 위한 단축키 준비

소프트웨어의 사용 경험이 많은 전문가와 초보자의 업무수행 능력이 차이는 크다. 이러한 부분을 고려하여 빠른 기능수행을 위한 단축키를 제공해야 한다.

다. 사용자 보호

##### 1) 초기화면의 구성

좋은 UI는 첫 화면에 사용자 정의 옵션을 할 필요가 없도록 만들어야 한다. 즉, 처음 사용하는 사용자 위주로 첫 화면을 구성해야 한다. 소프트웨어를 처음 사용하는 사람은 어떤 기능의 UI에서 첫 화면을 구성하는 기능 요소들을 사용하는 것으로 끝날 수 있다. 사용자의 소프트웨어에 대한 이해도가 높아지면 더 깊은 레벨은 인터페이스로 접근하여 다양한 옵션을 사용할 수 있도록 하는 것이 좋다.

라. 유니버설 디자인

UI는 꼭 필요한 경우가 아니라면 일반적으로 통용되는 디자인에 따라 구성하는 것이 바람직하다. 대부분의 소프트웨어 UI는 메뉴 구성이나 유사 기능에 관련된 메뉴의 위치, 종료, 새문서 작성, 인쇄 등 대부분의 소프트웨어에 공통적으로 나타나는 기능들에 대해 동일한 위치에서 찾을 수 있도록 되어 있다.

마. 다른 문화 대응

소프트웨어가 한 국가 또는 문화권에서만 사용되는 것이 아니라면 사용자 인터페이스는 다른 나라나 문화권에 대해 고려해야 한다. 아이콘의 경우 문화적 차이 때문에 의미를 제대로 전달하지 못할 수 있다.

바. 만족도

소프트웨어의 UI에 대한 만족도를 개인적으로 측정할 경우 주관적인 대답을 얻게 되지만 다수를 대상으로 측정하여 평균을 내린다면 객관적인 자료가 될 수 있다.

사. 간결성

UI는 가능한 한 간결하게 구성해야 한다. 컴퓨터 상의 용어나 개념과 사용자가 인지하는 개념이 연관성이 있도록 하여 사용자가 UI를 사용할 때 혼란스럽지 않도록 구성해야 하며 사용자가 작업을 자연스럽게 수행할 수 있도록 해야 한다.

아. 알립성(피드백)

UI는 사용자의 액션에 대해 그 액션이 적절히 수행되었거나 또는 문제가 발생했다는 적절한 액션을 주어야 한다. 사용자의 액션에 대해 아무런 반응이 없을 경우 사용자는 자신의 액션이 제대로 수행되지 않았다고 오해할 수 있다.

자. 맵핑

어플리케이션을 사용하여 업무를 처리하는 과정에서 사용자들에게 공통적으로 적용될 수 있는 업무 처리 순서에 관한 시나리오가 존재한다면 UI는 이러한 시나리오의 처리 절차에 맞게 맵핑되어야 한다.

차. 식별성

1) 종료 방법의 제공

어떤 UI 상태에서라도 종료하는 방법에 대한 명확한 길을 제시해야 한다.

2) 중요 대상 식별

주목해야 할 부분으로 시선을 이끌기 위해 경계나 색, 폰트 등을 강조하여 표현할 수 있다. 그러나 지나치게 산만하거나 혼란스러운 방법은 피해야 한다.

카) 일관성

1) 사용자의 경험

사용자는 기존에 해오던 작업에 대한 편견을 가지고 있어서 그것을 계속 사용하려는 경향이 있다. 새롭고 혁신적인 사용자 인터페이스라고 하더라도 익숙한 기존의 방식을 고집하고 새로운 방식에 적응하려는 노력을 하지 않으려 할 수도 있다.

2) 일관성의 범위

UI의 일관성은 어플리케이션 화면에서 뿐만 아니라, 온라인 도움말이나 매뉴얼 등 모든 경우에 해당한다. 일관성은 한 시점에서 적용되는 것이 아니라 연속적인 제품의 릴리즈에도 적용되며 새로운 버전의 제품은 이전 버전의 제품과 일관성을 가지고 있어야 한다.

3) 용어사용

동일한 의미를 갖는 두 개 이상의 용어를 혼용해서 쓰지 않아야 한다.

다. 정보의 제공 레벨

한꺼번에 많은 양의 정보를 제공하는 것은 사용자에게 부담을 줄 수 있다. 일단 이해하기 쉬운 간략한 정보를 제공하고 제사한 추가 정보는 사용자의 선택에 따라 제공받을 수 있도록 하는 것이 좋다.

과. 적절한 용어, 정보 사용

대화상자에는 적절한 내용만 포함되어야 한다. 필요 없는 사항들은 배제하고 중요한 내용이 우선적으로 표현되어야 한다.

하. 기억 부담의 경감

1) 정보 기억의 부담

UI를 사용함에 있어 사용자의 기억에 부담을 주지 않아야 한다. 필요한 경우 다양한 형태의 도움말을 제공할 수 있도록 하여 사용자가 필요 이상의 정보를 기억하고 있을 필요가 없도록 해야 한다.

2) 과도한 컬러의 사용

과도하게 많은 색상을 사용하여 UI를 표현하는 것은 바람직하지 않다. 훈련된 사람도 11컬러 이상을 기억하기 어려우므로 5-7색 이상의 색을 쓰지 않는 것이 좋다.

3) 디폴트값의 사용

대화상자에서 사용자의 입력이 필요한 부분에 대해 디폴트(default)값을 제공하여 어떤 형태의 값이 들어야 하는지를 알려주면서 값 입력의 부담을 경감시켜 주는 것이 좋다.

거. 조작의 흐름

UI의 배치에 따른 조작의 흐름은 소프트웨어를 사용한 업무의 흐름에 따라 구성되어야 한다.

너. 강조

UI에서 exit나 undo 명령 같이 사용자가 작업 중에 빈번하게 또는 최소한 한 번 이상은 사용할 가능성이 있는 명령들에 대해서는 눈에 쉽게 띄는 곳에 배치하거나 강조하여 쉽게 인식할 수 있게 하는 것이 좋다.

더. 도움

1) 도움말

도움말은 이해가능한 선에서 최소한의 상세한 수준을 유지해야 한다. 그러나 복잡한 UI를 만들어 놓고 도움말을 자세히 하는 방법은 부적절하다.

2) 에러메시지

에러메시지는 평범한 언어로 표현하여 문제를 확실하게 알려주는 형태여야 하며 문제의 해결책을 유도해야 한다. 그렇지 못한 경우라면 에러로 인한 데이터 손실을 막을 수 있도록 데이터를 보존할 수 있도록 해야 한다. (예 : 저장 유도)

러. 직관성

1) 직관적인 이해

기능 사용방법을 직관적으로 이해할 수 있어서 Help 문서를 참조하지 않고도 사용가능한 것이 가장 바람직하다.

2) 기억성

소프트웨어를 처음 보았을 때 쉽다고 느끼는 것은 학습성에 관련되며 다시 접했을 때 쉽다고 느끼는 것은 기억성에 해당한다.

3.1.2 일반적인 요인

가. 조정

단축키는 사용자가 작업을 빠르게 수행할 수 있도록 하는 UI라 할 수 있다. 그러나 제품마다 다른 단축키는 사용자에게 혼란을 줄 수 있고 빠른 작업 수행에 지장을 줄 수 있으므로 최소한 빈번하게 사용되는 주요 단축키에 대해 조정할 수 있게 할 필요가 있다.

나. 실수에 대한 관용

1) 실수 방지를 위한 기능

파일을 지울 때 뜨는 경고창 같은 경우 숙련된 사용자가 실수가 적은 사용자에게는 불편을 주는 요인이 될 수 있으나 실수로 인해 파일을 삭제하게 됨으로써 발생하는 손실에 비하면 감당할 수 있는 수준이라고 할 수 있다.

다. 정보의 이해, 판단

1) 단순화

일반적으로 복잡한 시스템은 사용하기 어렵다. 그러나 필요한 정보를 제공하는데 있어서는 상세화할 수록 바람

직하다고 볼 수 있다.

2) 상세화

에러 발생시 에러메시지는 간결한 것이 오히려 바람직하지 않다. 메시지의 간결함은 사용자에게 혼란을 줄 수 있으므로 최소한의 필요한 정보를 담고 있어야 한다.

3) 배우기 쉬움

Infoworld 라는 소프트웨어 리뷰 회사의 경우, 소프트웨어 특성 중 “배우기 쉽다”는 요인에 대해 4~10%의 가산점을 부여하고 있다.

4) 친숙성

표현되는 문장은 명확해야 하고 사용자에게 친숙한 언어로 쓰여져야 한다.

3.2 사용성의 특성에 따른 인터페이스 요건의 분류

이 절에서는 앞 절에서 분석한 사용성 인터페이스의 요건을 바탕으로 사용성 품질특성에 관한 부특성인 이해가능성, 학습성, 운영성, 선호도를 중심으로 사용자 인터페이스를 분류하였다.

3.2.1 이해가능성에 관한 인터페이스의 요건

이해가능성은 소프트웨어가 적합한지, 그리고 특정 작업과 사용 조건에서 어떻게 사용될 수 있는지를 사용자가 이해할 수 있도록 하는 소프트웨어의 능력을 의미하며 이해가능성에 관한 인터페이스의 요건에 대해 정리하면 <Table 1>과 같다.

<Table 1> Requirements of Interface about Understandability

Requirements	Related Items
configuration of initial screen	- So that you do not need to configure the Custom option on the first screen - to make the first screen based on first-time users
universal design	- Configuration in accordance with generally accepted UI design - Be configuration so that you can find in the same position for the features that are common to most software

corresponding to other cultures	- Consider the user interface of the other countries or cultures
simplicity	- Make the UI, as concisely as possible - Configure so that the user can perform the task naturally - display the information user want in the appropriate time and location
Scope of the consistency	- Consider the consistency of the application screen - Consider the consistency of the online help or manual - Consider the consistency of continuous release of the product - Consider the consistency of the family products
the use of terms	- Do not mix more than two terms - it's good to use in one terminology
...	...

**3.2.2 학습성에 관한 인터페이스의 요건**

학습성은 사용자 하여금 소프트웨어가 제공하는 기능을 학습할 수 있도록 하는 소프트웨어의 능력을 의미하며 학습성에 관한 인터페이스의 요건에 대해 정리하면 <Table 2>와 같다.

<Table 2> Requirements of Interface about Learnability

Requirements	Related Items
memorability	- Configuration so that you can easily remember how to use the software
Learnability	- Configure the UI to make it easier to learn

**3.2.3 운영성에 관한 인터페이스의 요건**

운영성은 사용자가 소프트웨어를 운영하고 제어할 수 있도록 하는 소프트웨어의 능력을 의미하며 운영성에 관한 인터페이스의 요건에 대해 정리하면 <Table 3>과 같다.

<Table 3> Requirements of Interface about Operability

Requirements	Related Items
Flexibility	- UI design considering both the novice and expert - variety of settings for experts and default for beginners
methods that can perform a variety of functions.	- menu, icon, mouse, shortcuts

shortcuts for experts	- Consider the difference between expert and novice's ability to do the job a lot of experience with software
consistency	- the UI must do appropriate action on the user's actions
mapping	- The UI should be mapped according to the procedure of the business process scenario.
shutdown method	- Present a clear way on how to terminate at any UI.
Identifying the main target	- Lead the eye to highlight boundaries, colors, fonts, etc., to be noted. - Should be avoided too distracting or confusing.
Reduce the burden of memories	- Do not put a burden on the memory of the user when using the UI. - Give the various forms of help so that people do not need to remember more information
use of the default value	- Give the default values for the parts that require user input in the dialog box. - Reduce the burden to enter values.
...	...

**3.2.4 선호도에 관한 인터페이스의 요건**

선호도는 사용자에 의해 선호되는 소프트웨어의 능력을 의미하며 선호도에 관한 인터페이스의 요건에 대해 정리하면 <Table 4>와 같다.

<Table 4> Requirements of Interface about Attractiveness

Requirements	Related Items
Satisfaction	- Satisfaction about the UI of the software shall be objective measurement results measured by many people
Familiarity	- Sentence should be written in a clear and friendly language.
Tendency of the user	- Consider the familiar UI rather than a new and innovative UI.
Preclude the excessive use of color;	- It is not desirable to use a lot of colors to represent the UI excessively.

**4. 사용자 인터페이스의 사용성 평가 방법 구축**

이 장에서는 사용자 인터페이스를 중심으로 사용성을 평가하기 위한 방법에 대해 기술하였다.

#### 4.1 사용성에 대한 품질특성 체계

사용성에 관한 품질특성 체계는 소프트웨어 제품의 품질특성에 관한 국제표준인 ISO/IEC 9126과 소프트웨어 패키지의 품질요구사항 관련 국제표준인 ISO/IEC 12119를 근간으로 한다. 두 표준에서는 <Table 5>와 같은 사용성에 관한 체계를 구축하고 있다[3][4].

<Table 5> Quality Characteristics System of Usability

Quality Characteristics	Quality Subcharacteristics	Concept
Usability	Understandability	The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.
	Learnability	The capability of the software product to enable the user to learn its application.
	Operability	The capability of the software product to enable the user to operate and control it.
	Attractiveness	The capability of the software product to be attractive to the user.
	Compliance	The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.

본 논문에서는 ISO/IEC 9126과 ISO/IEC 12119의 사용성에 관한 품질특성과 부특성의 체계 및 개념을 근간으로 사용자 인터페이스 중심의 사용성 평가를 위한 요소를 도출하고 평가방법을 구축하였다.

#### 4.2 사용자 인터페이스의 평가를 위한 평가항목 구축

이 절에서는 사용자 인터페이스를 중심으로 사용성 품질특성의 부특성인 이해가능성, 학습성, 운영성, 선호도에 따른 척도(measure)에 대해 구축하였다.

##### 4.2.1 이해가능성에 관한 척도

이해가능성은 소프트웨어가 적합한지, 그리고 특정 작업과 사용 조건에서 어떻게 사용될 수 있는지를 사용자가 이해할 수 있도록 하는 소프트웨어의 능력을 의미하며 이해가능성에 관한 인터페이스의 요건에 대해 정리하면 <Table 6>과 같다.

<Table 6> Understandability Measure

Quality Subcharacteristics	item	Related Items
Understandability	Concise initial screen	- Configure the initial screen of the dialog box, - set selection items
	universal design	- UI design - configuration and location of UI items
	Correspond to different cultures	- the form of UI - understanding of icon and localization.
	simplicity	- Configuration of UI - The terminology used - The procedure for performing the work and the UI - When and where to present the elements that make up the UI.
	consistency(UI)	- UI consistency
	consistency (on-line document)	- on-line help - on-line manual
	consistency (Continuity of version)	- Older version of the product - Family suite
	Information offer level	- dialogue box - on-line help
	Stress	- The function elements to configure the UI
	help understandability	- on-line help
	Interface Understandability	- UI components - Icon
	...	...

##### 4.2.2 학습성에 관한 척도

학습성은 사용자로 하여금 소프트웨어가 제공하는 기능을 학습할 수 있도록 하는 소프트웨어의 능력을 의미하며 학습성에 관한 인터페이스의 요건에 대해 정리하면 <Table 7>과 같다.

<Table 7> Learnability Measure

Quality Subcharacteristics	item	Related Items
Learnability	Memorability	- the degree of memory of learning contents after a period of time after learning how to use
	ease of learning function	- Learning to use - Perform work tasks in accordance with the contents of the learning
	Ease of Access to Help	- on-line help
	...	...

4.2.3 운영성에 관한 척도

운영성은 사용자가 소프트웨어를 운영하고 제어할 수 있도록 하는 소프트웨어의 능력을 의미하며 운영성에 관한 인터페이스의 요건에 대해 정리하면 <Table 8>과 같다.

<Table 8> Operability Measure

Quality Subcharacteristics	item	Related Items
operability	Operating procedures, adjustment possibilities (flexibility)	- configuration of UI - default setting items of dialogue box - menu, icon - setting for expert(shortcuts)
	possibility to know the progress	- action of user - reaction of UI
	mapping	- work order - work order of UI
	possibility of end	- dialogue box - Various functions of time-consuming
	Identifying the main target	- configuration item of UI - color, font, icon etc.
	Reduce the burden of memories	- configuration item of dialogue box - on-line help - balloon help - Provided various messages
	use of the default value	- default value of dialogue box - user setting items of dialogue box - unit of input values
	The flow of operations	- work flow - work flow of UI - Placement of UI components
	Adjust shortcuts	- Function item set as a shortcut. - Shortcuts that can be set
	Tolerance for mistakes	- File delete function - The major features that can not be reversed after performing
	Ease of understanding a message (detailed message)	- error message - Various current situation notification message
	Ease of error recovery	- the situation which error occurs - recoverable or not
	Error prevention	- Possible Error Conditions - Occurrence of the error
...	...	

4.2.4 선호도에 관한 척도

선호도는 사용자에 의해 선호되는 소프트웨어의 능력을 의미하며 선호도에 관한 인터페이스의 요건에 대해 정리하면 <Table 9>와 같다.

<Table 9> Attractiveness Measure

Quality Subcharacteristics	Item	Related Items
Attractiveness	Possibility of interface changes	- Relocation of the toolbox - Relocation of the main elements used - Remove the non-use elements
	Interface preferences	- Popular form UI - Arrangement form of UI components
	...	...

4.3 사용성에 관한 품질검사표의 구축

품질검사표는 시험모듈에 정의된 메트릭을 기준으로 실제 품질 시험을 수행하는 과정에서 편리하게 활용할 수 있도록 필요한 핵심적인 사항들을 추출하여 정리한 표이다.

품질검사표는 메트릭명과 개념, 측정항목, 메트릭 계산식, 결과의 영역, 결과값, 문제점 기술 부분 등으로 구성되어 있다. 본 연구에서 개발한 주요 품질검사표를 정리하였다.

사용성에 관한 부특성인 이해가능성에 관련하여 초기 화면의 간결화, 유니버설 디자인, 다른 문화 대응 등의 품질검사표를 구성할 수 있다. <Table 10>은 유니버설 디자인에 관한 품질검사표를 보여주고 있다.

<Table 10> Quality Table of Understandability

Measure name	Is UI a common design?	
universal design		
Measurement items	A	the number of the dialog box to configure software
	B	the number of the dialog box with general design
expression	universal design = B/A	
The range of results	$0 \leq \text{universal design} \leq 1$	result value
problem		

5. 사용자 인터페이스의 사용성 평가 방법

이 장에서는 사용자 인터페이스를 중심으로 사용성의 부특성인 이해가능성, 학습성, 운영성, 선호도에 관련하여 평가한 사례를 예시하였다.



### 5.1 점검표의 작성

점검표는 척도(measure)를 측정하기 위해 필요한 구체적인 세부항목을 구성한 것이다. 여기에서는 주요 점검표를 선정하여 정리하였다.

〈Table 11〉 Checklist of Interface Consistency

No	Item name	Test result
1	Is it possible to visually determine the window control elements with different functions each other, and consistently located in the same location within each window?	Y
2	Is menu or window title consistently expressed in the form aligned on the top?	N/A
3	Is the label format consistent?	Y
4	Does UI provides a simple and consistent means for moving to the next higher level in the menu structure?	Y
5	Is there the basic menu or button in the same place on all screens?	Y
6	Is the meaning of vocabulary clear on the all screen which is used for displaying an icon?	Y
7	As a whole, is the interface consistent and unified?	Y
...	...	...
The number of Y		7
The number of N		0
Result		1.00

〈Table 12〉 Checklist of Understanding Help

No	Item name	Test result	
		Y/N	%
1	Is Help Available?	Y	
2	Are terms and content which is provided in the help configured easy to understand?	Y	
3	Is it possible to accurately match the actual function and the description provided by the Help?	Y	
4	After using the Help, can you perform the job correctly?	Y	
5	Does the information provided on screen explain the information related to the operation and its purpose clearly?	Y	
6	When you ask for help about a job, does software provide the narrative approach and the use stage method?	Y	
7	When the online training and tutorials is provided together, is the Links about relevant information provided?	Y	
...	...	...	
The number of Y		9	
The number of N		0	
Result		1.00	

〈Table 13〉 Checklist of IO Data

No	Item name	Test result
1	Is the areas represented clearly that data is entered?	Y
2	Is the suitable tables being used to represent the input data clearly?	Y
3	Depending on the size of the input data, are appropriate boundaries or underlines being used?	Y
4	In the data input dialog box, are pre-specified values provided and is text default edited easily?	Y
5	Are the basic routines being provided that calculate value without the data input of sum, average?	N/A
6	When a data entry field requires a specific format, is the format of the input field specified clearly?	Y
7	Is long information item expressed by dividing into groups with a specified number of characters?	N/A
8	Is each text field sufficient enough to accommodate the majority of the expected input without scrolling?	Y
9	If the number of input options acceptable is limited and is set in advance, are the mechanisms that can select and use some useful options provided?	Y
10	In the read-only field, are visual hint provided that represent the area can not be accessed?	Y
11	If there is no change in the previous input data, is there any input or key to use the input without change?	N/A
12	Is the function provided that can check the input field according to the established criteria before receiving the input data?	N/A
...	...	...
The number of Y		14
The number of N		1
Result		0.93

### 5.2 사용성에 대한 집계표

점검표로부터 도출된 결과를 품질부특성 수준과 품질 특성 수준에서 집계할 수 있다.

#### 5.2.1 품질부특성에 대한 집계표

품질부특성에 대한 집계는 점검표에 따른 각 척도의 점수를 기록하여 각 품질부특성별로 평균을 구한다.

<Table 14> Summary Sheet of Quality Subcharacteristics

Summary Sheet of Quality Subcharacteristics			
No	Quality Subcharacteristics	Evaluation Module	Score
12.	Understandability	Concise initial screen	0.90
		universal design	1.00
		Correspond to different cultures	N/A
		simplicity	0.95
		consistency	1.00
		Information offer level	1.00
		Stress	0.90
		help understandability	1.00
		Interface Understandability	1.00
		I/O data Understandability	0.93
		User guidance	0.95
		Total : 0.96	
13.	Learnability	Memorability	0.90
		ease of learning function	0.90
		Ease of Access to Help	0.88
		Total : 0.89	
14.	Operability	Operating procedures, adjustment possibilities (flexibility)	0.80
		possibility to know the progress	0.90
		mapping	0.95
		possibility of end	1.00
		Identifying the main target	1.00
		Reduce the burden of memories	0.95
		use of the default value	1.00
		The flow of operations	0.95
		Adjust shortcuts	N/A
		Tolerance for mistakes	0.90
		Ease of understanding a message	1.00
		Ease of error recovery	0.90
		Error prevention	0.95
		Total : 0.94	
13.	선호도	Possibility of interface changes	0.80
		Interface preferences	1.00
		Total : 0.90	

5.2.2 품질특성(사용성)에 대한 집계표

품질특성에 대한 집계는 품질부특성의 점수를 기록하여 평균을 구하여 도출한다.

<Table 15> Summary List of Quality Characteristics

Summary Sheet of Quality Characteristics		
Quality Characteristics	Evaluation Module	Score
Usability	Understandability	0.96
	Learnability	0.93
	Operability	0.94
	Attractiveness	0.90
Total : 0.93		

5.3 평가 방법의 비교

소프트웨어 제품에 대한 평가에는 다양한 방법이 있다. 소스 코드 자체를 정적으로 분석하여 코드의 복잡도를 평가하는 방법과 소프트웨어 라이프사이클 전반에 걸쳐 단계별로 체크리스트를 활용하여 산출물에 대한 평가를 수행하여 품질 수준을 추정하는 방법, 그리고 최종 소프트웨어 제품을 구성하는 요소인 제품설명서, 사용자 문서, 프로그램과 데이터에 대해 평가모듈을 적용하여 평가를 수행하는 형태가 있다.

5.3.1 코드 복잡도 평가

코드 복잡도 평가는 소프트웨어 품질특성 중 유지보수성과 이해 용이성에 관련된 것을 주로 고려한다. 소프트웨어 생명주기중 가장 문제가 되고 있는 유지보수성 문제는 개발 단계별로 표준화된 요구명세서, 구조적 프로그래밍 기법, 문서화 등을 갖춘다면 다소 해결될 수 있다. 그러나 이러한 단계별로 요구되는 항목들에 대한 표준화된 방안의 구축이 쉽지 않다.

따라서 코드 복잡도 평가방법은 유지보수성 품질을 향상시키기 위해 프로그램 개발자가 개발 중에 보다 구조적이고, 이해하기 쉽고, 기능을 쉽게 파악할 수 있는 소프트웨어를 개발할 수 있도록 한다는 목적에 집중한다.

코드의 복잡도를 측정하기 위해 기능 사이즈 모델, 이해 용이성 모델, 복잡성 모델 등을 구성한다.

가. 기능 사이즈 모델

기능 사이즈 모델은 모듈 하나를 유지하는 기능의 수를 측정하여 기능 사이즈로 정량화한 것이며, 소스 프로그램의 각각의 모듈이 유지하고 있는 기능이 작고, 하나의 모듈당 하나의 기능을 가지고 있는 것이 바람직하다는 점을 고려한 모델이다. 모듈당 기능이 작으면, 보수할 때 다른 모듈에 영향을 적게 주며, 각 모듈의 기능 추출

이 간단하여 소스 프로그램의 수정과 갱신이 용이하다는 점에 착안한 모델이다.

나. 이해 용이성 모델

이해 용이성 모델은 소스 프로그램의 읽기 쉬움을 정량화한 것으로 작성된 소스 프로그램이 시각적으로 읽기 쉽다면 프로그램을 빠르게 이해할 수 있음을 고려한 것이다. 프로그램을 쉽게 이해할 수 있다면 보수시 영향을 주는 범위를 명확히 알 수 있어, 프로그램의 이해가능성과 보수성을 높일 수 있다는 점에 착안한 모델이다.

다. 복잡성 모델

복잡성 모델은 논리의 복잡성을 정량화한 것으로, 소스 프로그램의 논리가 단순 명쾌한 것이 바람직하다는 생각을 기초로 한다. 복잡성 모델에서는 각 모듈에서의 실행문의 수, Halstead의 어휘 사이즈, Halstead의 프로그램 길이, Halstead의 프로그램 볼륨, McCabe의 척도, 정보 bit, D 척도 등의 요인항목이 있다.

5.3.2 체크리스트를 활용한 평가

<Table 16> Checklist about Consistency of Design Description

Check-List of Internal Characteristics	Evaluation Factor Items	Detailed Items	Evaluation				
			1.0	0.7	0.5	0.3	0.1
1. Are the standards/protocols of design descriptions being followed consistently?		Is there a consistency on the design descriptions? How is the design description provided with some degree of consistency in the standards/conventions?					
		...					
		Total :					
2. Is there a consistency on the name of modules and data?		Is standard guidelines being provided to give the names of variables / module name?					
		...					
		Total :					
3. Is there a consistency on the way of comment description?		Is comment provided faithfully in the design description?					
		...					
		Total :					
Analysis of Problems :							

체크리스트를 활용한 평가방법은 소프트웨어 라이프사이클 전반에 대해 체크리스트를 구축하여 최종 소프트웨어 제품이 나오기 전에 라이프사이클 단계별로 품질수준을 확인할 수 있는 방법이며 이를 통해 최종 소프트웨어 제품에 대한 품질수준을 추정한다. 체크리스트를 활용한 방법은 소프트웨어 제품평가에 관한 표준인 ISO/IEC 9126의 부분 3을 근간으로 하여 소프트웨어 개발과정에서 산출되는 중간산출물의 평가를 목적으로 한다. <Table 16>에 설계서의 일관성을 평가하는 체크리스트의 예를 나타내었다.

5.3.3 품질평가 모듈을 활용한 평가

품질평가 모듈을 활용하는 방법은 최종 소프트웨어 제품을 구성하는 제품설명서, 사용자 문서 그리고 프로그램과 데이터를 대상으로 하여 ISO/IEC 9126의 부분 2인 외부메트릭을 근간으로 한다.

품질평가 모듈을 활용하는 평가방법에서는 평가 대상 소프트웨어 제품의 특성에 따른 품질 요구사항을 분석하여 품질특성에 따라 분류하여 품질특성별 평가모듈을 평가 대상 소프트웨어 제품에 적합하게 하는 과정을 거쳐 완성된 평가모듈을 활용함으로써 각 소프트웨어의 특성을 고려한 최적화된 평가를 수행할 수 있다는 장점이 있다.

본 논문에서는 사용성에 관련하여 사용자 인터페이스를 중심으로 품질검사표를 구축하고 점검표의 세부 항목을 도출하여 적용함으로써 체크리스트를 활용하는 방법에 비해 소프트웨어 제품의 특성을 좀 더 명확히 반영하는 것이 가능하다고 하겠다. 즉, 사용성을 평가하기 위한 품질평가 모듈을 사용하는 방법을 통해 소프트웨어 제품에 대해 국제표준의 품질특성 중 사용성에 관한 부특성의 관점에서 소프트웨어 제품의 전반적인 품질수준을 판단할 수 있다는 장점이 있다.

6. 결론

국내외적으로 소프트웨어 품질평가 관련 체계의 표준화를 위한 다양한 활동과 연구가 수행되어 왔다. 국내에서도 소프트웨어 품질평가에 관해 국제표준에 근간을 둔 국내표준을 제정하고 이에 따른 인증제도가 구축되어 정착되어 왔으며 국내 소프트웨어 제품의 인증 활동을 통해 소프트웨어의 품질에 대한 인식을 높이고 실질적인

품질향상에 이바지해 왔다.

소프트웨어 품질평가를 수행하기 위해서는 소프트웨어 품질특성과 부특성 체계를 구축하고 부특성에 관한 척도(measure)를 구축해야 한다. 그 중에서 소프트웨어의 사용성에 관련된 품질특성에 관해 이해가능성, 학습성, 운영성, 선호도, 준수성 등의 부특성이 정의되어 있으며 이에 관한 측정항목에 대해 정립하기 위한 연구가 수행되어 왔다.

UI는 엔드유저가 소프트웨어를 사용함에 있어 사용성을 결정하는 가장 핵심적인 요소이므로 UI를 제외하고 사용성을 논할 수는 없을 것이다. 그럼에도 불구하고 UI를 구축하는 기준이나 권고안 등에 입각한 사용성 평가 방법 및 측정항목의 구축에 대한 연구는 미흡했던 것 같다.

소프트웨어 품질평가 분야는 소프트웨어 품질특성에 관한 국제표준 제정을 기점으로 하여 품질평가 체계를 구축하는 면에서는 많은 발전을 이루었으나 평가항목에 대한 구체적인 측정방법 측면에서는 아직까지 미흡한 점이 많이 남아 있는 것이 현실이다.

본 논문에서는 소프트웨어의 품질평가에 관련된 국제표준의 품질특성들과 사용자 인터페이스를 구성하는 요소들과의 관련성을 명확히 분석하여 소프트웨어의 사용자 인터페이스로부터 측정할 수 있는 품질특성 요소들을 추출하고 평가하는 방법에 대한 체계를 구축하였다.

본 연구를 통해 소프트웨어의 UI를 기반으로 한, 소프트웨어에 대한 품질평가 체계와 품질검사표를 구축하고 평가방법을 적용한 사례를 통해 다른 평가방법과의 차이점을 분석하였다.

개발된 사용성 관련 UI를 대상으로 한 시험 및 평가방법을 적용하여 시험사례를 구축함으로써 사용성 품질특성에 관한 국제표준의 품질특성과 부특성에 근거한 소프트웨어의 품질측정 방법에 대한 적용 가능성을 확인해 볼 수 있었으며 향후 지속적인 연구를 통해 다양한 유형의 UI를 기반으로 한 사용성 평가방법에 대해 품질측정 방법을 구축하는 연구를 수행할 필요가 있다.

사용성 품질특성에 대한 품질 시험 및 평가방법의 개발과 향후 실질적인 활용을 통해 고품질의 사용성 품질을 갖는 UI의 개발을 통한 품질향상으로 엔드유저의 편리한 소프트웨어 활용에 기여할 수 있기를 기대한다.

## REFERENCES

- [1] Eun-young Kim, Jae-ha You, Se-jin Jung, Young-seon Kweon, Study on Interface for enhancing customer satisfaction in Internet Shopping Malls, Proceedings of The Korea Society of Management Information Systems, pp.532-540, 2003.
- [2] Won-seok Jang, Yong-ku Ji, Usability evaluation methods for user interface of smartphone augmented reality application, The Journal of Society for e-Business Studies 16(1), 2011. 1.
- [3] ISO/IEC 9126, Information Technology - Software Quality Characteristics and metrics.
- [4] ISO/IEC 12119, Information Technology - Software Package - Quality requirement and testing.
- [5] Software Review Plan: Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, NUREG-0800 BTP GICB-14, U.S. NRC, Rev. 4, 1997.
- [6] Dong-Min Kim, Chul-woo Lee, Technology trends of Smart phone user interface, Journal of Korea Information Science Society, 2010. 5.
- [7] Sang-youn Lim, Yung-hee Kim, Jung-hyeok Yun, Mobile phone application UI technology trends, Journal of Informational Processing Society 15(4), pp.59-64, 2008.
- [8] Sang-Yeop Nam, Eun-hee Jun, In-Jung Park, Real-time embedded speech recognition system, The Institute of Electronics Engineers of Korea 40(1), pp.74-81, 2003.
- [9] Tae-Jun Kim, Biocensor, KSBB Journal 22(6), pp.421-425, 2007.

### 이 하 용(Ha-Yong Lee)



- 1993년 2월 : 강원대학교 전자계산학과 졸업(이학사)
- 1995년 2월 : 강원대학교 대학원 전자계산학과 SW공학전공(이학석사)
- 2005년 2월 : 호서대학교 벤처전문대학원 컴퓨터응용기술학과졸업(공학박사)

- 1996년 3월 ~ 2005년 2월: 경희대, 경원대, 선문대, 호서대 컴퓨터공학부강사
- 1995년 6월 ~ 2002년 12월 : 한국SW품질연구소 선임연구원
- 2005년 9월 ~ 현재 : 서울벤처대학원대학교 교수
- 관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 객체지향 프로그래밍, 객체지향 분석과 설계, 컴포넌트기반 S/W 개발방법론, 품질평가)
- E-Mail : lhyazby@suv.ac.kr

### 양 해 술(Hae-Sool Yang)



- 1975년 2월 : 홍익대학교 전기공학과 졸업(학사)
- 1978년 8월 : 성균관대학교 정보처리학과 졸업(석사)
- 1991년 3월 : 日本 오사카대학 정보공학과 S/W공학 전공(공학박사)
- 1975년 5월 ~ 1979년 6월 : 육군중앙경리단 전자계산실 시스템분석장교
- 1980년 3월 ~ 1995년 5월 : 강원대학교 전자계산학과 교수
- 1986년 12월 ~ 1987년 12월 : 日本 오사카대학 객원연구원
- 1995년 6월 ~ 2002년 12월 : 한국소프트웨어품질연구소 소장
- 2010년 3월 ~ 2011년 2월 : 호서대 글로벌창업대학원 원장
- 1999년 11월 ~ 현재 : 호서대학교 벤처전문대학원 교수
- 관심분야 : S/W공학(특히, S/W품질보증과 품질평가, 품질감리 및 컨설팅, SI), S/W프로젝트관리, 품질경영.
- E-Mail : hsyang@hoseo.edu