

운영체제 품질 평가 모형 A Study on the Quality Evaluation Model of Operating Systems

윤민석

전남 여수시 둔덕동 산 96-1 여수대학교 정보기술학부

한상호

전남 여수시 둔덕동 산 96-1 여수대학교 교육대학원

Abstract

The multiplicity of operating systems brings about a choice problem to the end-user and a multi-fold intensity to the competing developer. This study intends to provide guidelines solving the problems for both the end-user and the developers of operating systems. For the purpose, we adopt an appropriate evaluation model in terms of user-perceived quality through the applicable criteria and an evaluation method. Briefly, reliability and portability are key factors to contribute to the perceived quality. The empirical study also illustrates a comparative analysis on the dominant products. Microsoft Windows and Linux, in Korea.

1. 서론

컴퓨터 운영체제(operating system: OS)는 컴퓨터의 메모리, CPU(central processing unit) 등과 같은 제한된 각종 자원을 관리 및 제어 제어함으로써 사용자의 컴퓨팅을 돕는 사용자와 컴퓨터의 인터페이스(interface)를 담당하는 소프트웨어이다. 운영체제의 전형적인 역할은 프로세스 관리, 기억장치 관리, 입출력 서비스로 크게 구분해 볼 수 있다. 좀 더 구체적으로 프로세스의 생성 및 실행, CPU 스케줄링, 기억장치의 할당 및 회수, 인터럽트(interrupt) 관리, 데이터 관리 통신관리 등의 역할과 함께 어셈블, 컴파일, 연결 등의 기능을 제공하여 프로그램을 개발을 지원하는 역할도 수행한다[5].

운영체제의 중요성은 컴퓨터 운영의 기본적인 시스템 소프트웨어라는 점 이외에, 운영체제의 사양은 응용 소프트웨어, 컴퓨터 시스템, 컴퓨터 주변장치 구동기(driver)의 설계에 지대한 영향을 미친다는 데 있다. 따라서 운영체제의 시장 지배는 자체 시장경쟁에서의 지위뿐만 아니라 관련 소프트웨어 및 하드웨어 시장으로부터도 절대적인 지위를 보장받는 계기가 될 수 있다[7].

1980년대 이후 추구되어 온 개방형 시스템(open system)의 중심은 운영체제 표준이었으며, ANSI(American National Standards Institute), IEEE(Institute of Electrical and Electronic Engineers) 등 세계적인 표준 제정 기관에서도 OS 표준을 제정하고 있다. 그러나 이러한 개방형 운영체제는 Unix와 최근의 리눅스(Linux) 및 공개 소스 운영체제를 중심으로 이루어져 왔으며, IBM OS/2, AS/400, Microsoft NT 및 2000 계열 등 32비트 윈도우즈(Windows) 계열, Novel NetWare 등은 전용 시스템의 성격이 상대적으로 강한 것으로 분류된다. 전용 시스템의 환경에서 운용되는 소프트웨어 및 하드웨어들은 해당 운영체제의 규격(특히 인터페이스와 관련하여)을 따를 수밖에 없게된다[1, 7].

지난 20여년간의 PC(personal computer) 보급에는 Microsoft의 도스(disk operating system) 및 윈도우즈 역할이 지대하였음은 사실이다. 그러나 이를 통하여 Microsoft는 PC 운영체제뿐만 아니라 각종 사무자동화, 인터넷 브라우저, 개발 툴(tool) 등의 여러 분야에서 자사제품의 독점적인 지위를 확보하기에 이르러 독점의 폐해를 노출시키고 있다. 또한 1990년대 중반부터 서버용 운영체제를 출시하여 시장점유율이 증가 일로에 있어 독점 폐해 우

러를 높이고 있다.

운영체제를 포함하는 소프트웨어 시장에 반(Anti-) Microsoft 정서가 팽배해진 가운데 최근 수년간 주목을 받은 운영체제로 리눅스를 꼽을 수 있다. 1990년대 초 핀란드의 Linus Torvalds에 의해 개발되기 시작한 리눅스는 Unix를 기반으로 한 소스 공개로 문제 및 해결을 공유하게 되어 시스템 개발이 급진전되었다. 더욱이 프리웨어(freeware)로 배포되어 급속한 보급속도를 나타내고 있으며[4, 6], Dell Computer, Hewlett-Packard 그리고 IBM과 같은 다수의 IT 선두 공급업체들이 자신들의 워크스테이션과 서버상에 리눅스를 제공하고 있다. 따라서 현재 범용 서버용 운영체제는 크게 유닉스, 윈도우즈 경쟁체제에 리눅스가 포함될 가능성이 상당한 것으로 전망된다[7].

이에 본 연구는 최종 사용자의 품질 관점에서 운영체제를 평가할 수 있는 모형 개발에 연구의 목적을 두어, 최종 사용자에게는 운영체제 선택에 지침이 될 수 있는 정보의 제공과 함께 운영체제 제품개발자 또는 관련사업자에게는 자사제품의 개발 전략수립에 활용될 수 있는 유용한 정보를 제시하고자 한다.

2. 운영체제 품질평가 모형

2.1 소프트웨어 품질 평가에 관한 고찰

운영체제는 대표적인 시스템 소프트웨어이므로 소프트웨어 평가에 관한 선행연구를 고찰하고 이로부터 운영체제의 특수성을 고려한 운영체제 품질 평가 기준을 수립하고자 한다.

Brownstein & Lemer는 기능적 요구사항(functional requirements), 설계 개념(design concept), 비용(cost), 공급자 및 지원 서비스 등을 소프트웨어 평가 및 선택 기준으로 제시한 바 있다[13]. Zahedi[24]는 DBMS(database management system) 평가에 있어 기능적 관점, 물리적 관점, 비용(cost), 이익(benefit)으로 평가 기준을 대분류 하였다. Frankel 및 NASA는 사용자 인터페이스 또는 사용 용이성을 주요 요소로 고려하기 시작하였다[13]. Davis[10]는 소프트웨어 선택의 행위적 측면에서 유용성과 사용 용이성을 중심으로 한 실증 연구 결과를 제시한 바 있다. 이외에도 Kekre et al[18]는 신뢰성 이외에도 설치 용이성 및 유지 보수성 등을 고려한 점이 주목할 만하다.

이들 선행 연구로부터 소프트웨어 평가를 기능적 요구사항과 성능, 사용용이성 등 비기능적 요구사항으로 대별됨을 알 수 있으나 세부적 사항에서는 평가 목적, 대상 등에 따라 상이한 평가기준이 제시됨을 알 수 있다.

이에 본 연구는 보다 종합적이고 체계적인 접근을 시도한다는 차원에서 품질 지향적 소

프트웨어 평가를 추구하고자 한다. 소프트웨어 품질은 “명시적이거나 묵시적인 필요를 만족시키는 소프트웨어 제품의 능력과 관련된 소프트웨어 제품의 특징(features) 및 특성(characteristics)의 전체”로 정의된다[15]. 지금까지 소프트웨어 품질의 이해 및 측정은 소프트웨어 ‘품질모형’을 통하여 논의되어 왔다. 소프트웨어 품질모형은 “품질 요구사항 및 품질 평가에 기초를 제공하는 품질특성 및 그들간의 관계들의 집합”이며, 이 때 품질특성은 “소프트웨어 제품의 품질이 기술되고 평가될 때 적용되는 소프트웨어 제품 속성들의 집합”으로 정의되며 각 품질특성은 여러 개의 하위 특성으로 더 세분화 될 수 있다[16].

<표 1>와 같이 제안된 여러 품질모형 중 ISO 9126 모델이 소프트웨어가 실제 작동되는 환경에서 고려되는 사용자 관점을 잘 반영한다. Boehm[8]과 McCall[19]의 모형에서는 소프트웨어 사용상의 품질 요인과 소프트웨어 내부 속성(internal properties)이 혼재하고 있다. 반면, ISO 9126[16]은 양자를 구분하여 내부 속성들은 소프트웨어 제품 개발과정에 적용되는 품질특성으로 규정한다. Dromey[11]의 모형은 개발자 관점으로부터 출발하는 품질모형이므로 사용자 관점을 포괄적으로 반영하기에는 한계가 있는 것으로 판단된다.

< 표 1> 품질 모델에 관한 선행 연구

품질 모델	특 징
Boem's Model (1978)	-품질로서의 일반적 유용성은 3개의 범위로 범주화 -15개의 기초구성요소와 7개의 중간구성요소와 관련
McMall's Model (1977)	-3개의 S/W 발전 단계 -11개의 사용자에게 계획 되어진 요소 -23개의 소프트웨어에 순응하는 품질 분류 기준
Evans' Model (1987)	-소프트웨어 사용주기에 의한 -3단계와 12 요소
ISO 9126 Model (1991)	-외부 품질 모델 -6개의 주특성과 21개의 부특성
IEEE Model (1992)	-ISO 9126 모델과 비슷 -6개의 주특성과 21개의 부특성
Chidung & Raffy's Model (1992)	-3단계 : 사용자관점, 개발자 관점, 물리적 측정 -8가지 요소와 12가지 분류기준
Dromey's Model (1995)	-상대적으로 아래로부터 접근 -3단계 : 구조적 형태, 품질 수행 특성, 고품질의 속성
Tervonen's Model (1996)	-기술적 개념 지향적 모델 -개발자의 관점 플러스 목표 지향적 모델

자료: Yoon M.S.(1997)

2.2 운영체제 품질 평가 기준

본 연구에서는 ISO 9126 품질모델을 기반으로 하고 이를 운영체제에 적합하도록 수정하여 운영체제 품질 평가기준으로 이용하고자 한다. ISO 9126 품질모델을 기반으로 최종사용자 시각에 의한 운영체제 품질 평가를 수행하기 위해서는 다음 두 가지 사항이 선행되어야 한다. 첫째, ISO 9126 품질모델은 범용 목적이므로 본 연구의 대상인 운영체제를 평가하는데 적용되는 품질특성이 확인되어야 한다. 둘째, 각 품질특성은 다수의 소프트웨어 공학 용어로 기술되어 있으므로 이를 최종사용자가 이해하기 쉽게 표현하여야 한다[23].

첫 번째 과제와 관련하여 ISO 9126 품질모델은 소프트웨어 사용상의 품질을 반영하는 6개의 품질 특성을 제시하고 있다. 이들 중 유지보수성을 제외한 5개 품질 주특성 및 하위 17개의 품질 부특성으로 운영체제 품질 평가 기준을 구성하였다. 유지보수성은 분석성, 변경성, 안정성, 시험성을 포함하는 개념인데 내용상 다른 품질 특성에 비하여 최종사용자 computing과는 관련성이 적다고 판단하여 제외하였다.

한편, ISO 9126의 각 품질특성에 대한 정의는 내용의 모호성을 피하기 위하여 소프트웨어 공학용어로 기술되어 그 내용을 이해하는데 소프트웨어 공학에 상당한 수준의 지식이 요구된다. 이러한 사실은 본 연구의 최종사용자에 의한 평가 방법에 배치된다. 이를 해결하기 위하여 본 연구는 ISO 9126에서 제시하고 있는 각 품질특성과 관련된 Metrics & Indicators를 충분히 검토하여 최종 사용자가 이해할 수 있도록 최대한 운영체제와 관련된 용어로 전환하였다[그림 1].

2.3 계층분석과정(Analytic Hierarchy Process)을 이용한 평가 모형

만약 품질평가 기준 계층에서 최하위 단계(2 단계) 기준들 각각의 중요도($w_j^{(2)}$)와 각 요소에 대한 k번째 대안의 평가점수($Q_{jk}^{(2)}$)를 측정하면 k번째 대안의 종합점수(Q_k)는 식(1)과 같은 가중평균 형태로 간단히 구할 수 있다.

$$Q_k = \sum_{j=1}^n w_j^{(2)} \cdot Q_{jk}^{(2)}, \text{ 여기서 } \sum_{j=1}^n w_j^{(2)} = 1 \quad (1)$$

이 때 각 요소의 중요도는 계층적 분화 원리에 따라 차상위 요소의 중요도가 이에 속하는 차하위 요소들의 중요도에 계층적으로 전파되도록 한다. 이는 차상위 요소의 중요도와 이에 속하는 차하위 요소들의 중요도 합이 같도록 하면 간단히 해결된다. 따라서 계층상에

서 중요도 산출은 상위단계 요소들간 중요도를 산출하고 차례로 하위단계 요소들의 중요도를 산출하는 하향식 분화를 이루는 것이 일반적이다.

한편, 대안의 평가는 계층상에 존재하는 각 요소를 따라 상향식으로 종합되며 이를 계층 전체 요소에 대하여 일반화하면 식(2)와 같다.

$$Q_{ik}^{(h)} = \sum_{j=1}^{j_{\max}} w_j^{(h+1)} \cdot Q_{jk}^{(h+1)}, \text{ for } h=0, 1 \quad \left| \right.$$

$$j \in ID(C_i^{(h)}), \quad j_{\max} = n(C_i^{(h)})$$

여기서 $\sum_{j=1}^{j_{\max}} w_j^{(h+1)} = 1, \quad (2)$

$Q_{ik}^{(h)}$: 계층 수준 h의 i번째 기준에 대한 k제품의 품질 수준,

$w_j^{(h)}$: 계층 수준 h의 j번째 기준의 중요도,

$C_i^{(h)}$: 계층 수준 h의 i번째 기준에 속하는 차하위 계층 기준들의 집합,

$ID(*)$: 집합 *를 구성하는 원소들의 인덱스(index) 집합,

$n(*)$: 집합 *를 구성하는 원소들의 수.

위 식에서 $Q_{ik}^{(0)} = Q_k$ 는 k 운영체제의 품질지수가 된다. 만약 대안 평가에 있어서도 대안간 평가의 합이 1이 되도록 정규화 한다면 계층의 각 기준에 따른 운영체제 대안의 품질수준들 간에는 다음과 같은 조건이 성립하게 된다.

$$\sum_k Q_{jk}^{(h)} = 1, \quad k \in ID(P), \text{ for } \forall h, j,$$

여기서 $ID(P)$ 는 제품 대안들의 인덱스 집합.

본 연구의 모형과 같은 계층적으로 구조화된 문제해결에 적절한 기법으로 알려진 계층분석과정에 의한 대안 평가는 ① 의사결정을 위한 요소들을 계층적으로 분해하는 단계 ② 요소들을 쌍대비교(pairwise comparison)하는 단계 ③ 고유치(eigenvalue) 산출 및 일관성(consistency) 검정 단계 ④ 산출된 가중치를 종합하여 각 대안들에 대한 복합가중치를 구하는 단계 등 네 단계로 구성된다[20]. 계층분석과정은 지금까지 광범위한 의사결정분야에 적용되어 왔으며 소프트웨어 분야에도 여러 차례 적용된 바 있어[2, 3, 17] 상술은 생략하기로 한다.

특성	부특성	내 용
기능성	적절성 (suitability)	<input type="checkbox"/> 운영체제의 목적상 필요한 기능의 구비 정도 <input type="checkbox"/> 사용자가 원하는 기능의 제공 정도 <input type="checkbox"/> 제공되는 기능 수준의 적당성
	정밀성 (accuracy)	<input type="checkbox"/> 매뉴얼의 설명과 프로그램 동작의 일치 정도 <input type="checkbox"/> 수행된 작업 결과의 정확성·정밀성 정도
	상호 운영성 (interoperability)	<input type="checkbox"/> 다른운영체제 를 클라이언트로 사용하여 네트워크를 구성할 수 있도록기본 지원정도 <input type="checkbox"/> 다른 운영체제와 적용되는 능력
	순응성 (compliance)	<input type="checkbox"/> 요구되는 표준이나 관례, 또는 규정에 대한준수정도
	보안성 (security)	<input type="checkbox"/> 권한없는 사용자가 불법적으로 시스템에 접근하여 데이터를 조작하지 못하게 하는 능력 (해킹 등) <input type="checkbox"/> 암호화 기능의 존재화 수준
신뢰성	성숙성 (maturity)	<input type="checkbox"/> 프로그램의 결함에 따른 고장/장애 빈도와 심각성 <input type="checkbox"/> 프로그램의 결함에 영향을 받는 작업 결과물의오류 정도
	고장 허용 (fault tolerance)	<input type="checkbox"/> 오조작/오입력의 경우에도 프로그램과 데이터를 보존하는 능력 <input type="checkbox"/> 오조작/오입력의 검출 능력 또는 안내 메시지 제공 수준
	회복성 (recoverability)	<input type="checkbox"/> 고장/장애의 경우 원래 성능 수준으로 회복하는 능력 및 그 수준 <input type="checkbox"/> 고장/장애로 인해 직접적으로 영향을 받는 데이터를 본 상태로 되돌려 놓는 능력과 그 수준 <input type="checkbox"/> optional install, undo, shut-down 경우 자동 back-up 등
사용성	이해성 (understandability)	<input type="checkbox"/> 각 기능과 제반 관련사항의 개념 및 적용과 관련된 명확성 <input type="checkbox"/> 사용자가 운영체제를 운영하는 절차 및 방법을 쉽게 이해할수 있는 능력 <input type="checkbox"/> 프로그램메뉴,메시지,입력항목 그리고 결과 등은 이해하기 쉬운 가?
	학습성 (learnability)	<input type="checkbox"/> 쉽게 배울 수 있는 가?
	운용성 (operability)	<input type="checkbox"/> 조작하기 쉬운가?
효율성	시간행동 (time behavior)	<input type="checkbox"/> 프린터 출력시간, 파일 입·출력 시간 <input type="checkbox"/> 프로그램 loading/unloading 시간 <input type="checkbox"/> 다른프로그램 loading/unloading 시간
	자원행동 (resource behavior)	<input type="checkbox"/> CPU 또는 MPU 요구사항 <input type="checkbox"/> 메모리 요구사항 및 프로그램의 크기(Hard disk 점유량)
이식성	적용성 (adaptability)	<input type="checkbox"/> 적용 가능한 H/W
	설치성 (installability)	<input type="checkbox"/> 프로그램을 설치하기에 필요한 노력의 정도
	부합성 (conformance)	<input type="checkbox"/> 이식성에 관련된 표준이나 협정에 따라 만들어졌느냐의 정도
	대체성 (replaceability)	<input type="checkbox"/> 동일 환경에서 운영체제 교체시 데이터 및 기능을 계속적으로 사용할 수 있는 능력 <input type="checkbox"/> 다른 운영체제 로 대체할 수 있는 능력

[그림 1] 운영체제 평가 기준 체계 및 내용

3. 모형의 적용 및 중요도 평가

본 연구의 설문 대상은 운영체제 윈도우즈와 리눅스를 모두 사용해본 경험이 있는 사용자로 한정하였다. 이는 본 연구에서 다루어지는 소프트웨어 품질 평가 항목이 운영체제에 대한 전반적인 지식을 요구함이고 부가적으로 표본 크기에 비하여 신뢰성 있는 응답비율을 높이고자 하는 조사의 효율성 때문이기도 하다.

3.1 운영체제 품질 주특성의 중요도

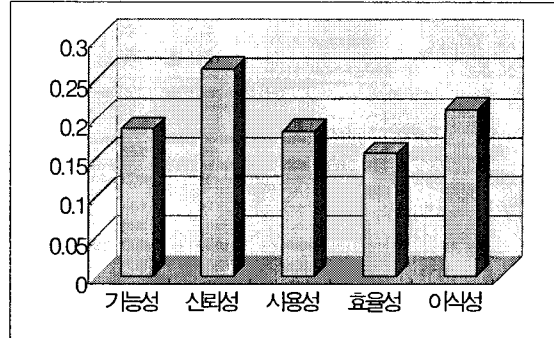
수집된 자료 중 응답의 신뢰성이 있다고 판단되는 자료만을 분석에 사용하였다. 이 때 신뢰성의 기준은 계층분석과정에서 제공하는 계층일관성 지수(consistency ratio of an entire hierarchy)[21]의 범위가 0.2 이하인 경우로 하였다. 또한 자료 요약은 계층분석과정의 척도이론에 따라 기하평균을 사용하였다.

계층 1단계 품질 주특성의 중요도를 분석한 결과 <그림 2>와 같다. 먼저 품질 주특성들을 중요도 순으로 나열하면 신뢰성(0.2626), 이식성(0.2113), 기능성(0.1867), 사용성(0.1839), 효율성(0.1554) 순으로 나타났다.

신뢰성은 일정 기간동안 정해진 조건 아래서 정해진 성능수준을 유지하도록 하는 소프트웨어 능력에 영향을 미치는 속성의 집합으로 정의하였다. 신뢰성이 가장 중요하게 나타난 것은 운영체제의 고장/장해가 시스템 운영에 미치는 영향력이 크기 때문이다. 운영체제 결함은 곧 각종 응용프로그램 및 장치 관리에 문제점을 유발시킬 수 있으며, 고장/장해시 데이터 회복 능력은 운영체제 자체 데이터뿐만 아니라 응용프로그램의 데이터에도 관계가 있다. 같은 맥락에서 운영체제는 사용자의 사소한 실수를 검출하고 관리할 능력도 요구된다.

환경 변화와 관련된 소프트웨어의 특성과 관련된 속성의 집합인 이식성 또한 상당히 중요한 것으로 나타났다. 이는 운영체제 프로그램을 설치하거나 다른 운영체제 프로그램 교체시 현 상태의 데이터를 계속해서 사용할 수 있느냐 없느냐의 문제가 사용자에게는 크게 작용한 것으로 판단된다.

기능 집합의 존재와 관련된 속성의 집합 및 성질의 명세인 기능성과 사용자가 사용하는데 필요한 노력 및 학습에 관련된 속성들의 집합인 사용성의 중요도는 비슷하게 나타났다. 그리고 일정한 조건 아래서, 소프트웨어 성능 수준과 사용된 자원의 양 사이의 관계와 관련된 속성의 집합인 효율성이 다소 낮게 중요한 것으로 평가되었다. 효율성의 중요도가 낮게 평가된 이유는 최근 컴퓨터 하드웨어 사양이 현저하게 좋아져 운영체제가 요구하는 시스템 사양이 상대적으로 큰 문제가 되지 않음을 반영하는 것으로 판단된다.



[그림 2] 품질 주특성의 중요도

3.2 운영체제 품질 부특성의 중요도

품질 부특성별 중요도 평가 결과는 <표 2>에 제시되었다. 본 연구에서 파악된 품질 부특성의 중요도에 관한 특징적인 면만을 살펴보기로 한다.

기능성의 부특성인 적절성(0.0279), 정밀성(0.0324), 상호운영성(0.0304), 순용성(0.0330), 보안성(0.0630) 중 권한이 없는 사용자가 허가 없이 시스템에 접근하여 데이터를 조작하지 못하게 하는 능력인 보안성이 가장 중요한 것으로 나타났다. 이는 좀더 편리한 기능이나 작업결과의 정확성보다는 해킹 위험이 더 큰 문제를 야기시키기 때문인 것으로 판단된다.

1단계 품질 특성에서 가장 중요한 것으로 평가된 신뢰성의 하위 특성인 회복성은 그 중요도가 0.1368로 전체 17개 부특성 중 가장 높은 것으로 나타났다. 이와 함께 성숙성(0.0738)도 상당히 중요한 것으로 나타났다. 이를 정리한다면 운영체제 자체의 결함이 없어야 하며 예기치 않은 고장/장해로부터 운영체제 자체와 데이터를 보호하는 능력이 대단히 중요하다.

사용성의 하위 특성들인 이해성(0.0677), 학습성(0.0558), 운용성(0.0604)은 모두 비슷하게 중요한 것으로 나타났다. 효율성의 부특성인 시간행동(0.0836)과 자원행동(0.0719) 모두 중요한 품질 특성으로 나타났으며 특히 시간행동이 더 중요한 것으로 나타났다.

그리고 마지막으로 동일 환경에서 운영체제 교체시 데이터 및 기능을 계속적으로 사용할 수 있는 능력인 이식성의 부특성에서는 적용성(0.0539), 설치성(0.0476), 부합성(0.0480), 대체성(0.0617) 모두 비슷 정동의 중요성이 있는 것으로 나타났다.

<표 2> 품질 부특성의 중요도

주특성	부 특성	중요도(그룹내)
기능성		0.1867
	적절성	0.0279 (0.1496)
	정밀성	0.0324 (0.1734)
	상호운영성	0.0304 (0.1628)
	순용성	0.0330 (0.1768)
신뢰성	보안성	0.0630 (0.3375)
		0.2626
	성숙성	0.0738 (0.2812)
사용성	고장허용	0.0520 (0.1980)
	회복성	0.1368 (0.5208)
		0.1839
효율성	이해성	0.0677 (0.3683)
	학습성	0.0558 (0.3034)
	운용성	0.0604 (0.3283)
이식성		0.1554
	시간행동	0.0836 (0.5375)
이식성	자원행동	0.0719 (0.4625)
		0.2113
	적응성	0.0539 (0.2550)
	설치성	0.0476 (0.2255)
	부합성	0.0480 (0.2273)
	0.0617 (0.2922)	

4. 운영체제 비교

앞서 제시한 품질 모형과 밝혀진 중요도를 가지고 Microsoft 윈도우즈와 최근 보급이 확산되고 있는 리눅스를 대상으로 품질 평가를 수행하였다. 평가 결과는 <표 3>과 같다.

두 제품의 상대적 품질 지수를 산출한 결과, 윈도우즈가 리눅스에 비하여 56% : 44%로 다소 우수한 것으로 평가되었다.

윈도우즈는 사용성과 이식성에서 리눅스보다 우수한 것으로 나타났고 리눅스는 효율성 면에서 차별적 우위가 존재하는 것으로 나타났다. 신뢰성 및 기능성에서는 서로 비슷한 것으로 나타났다.

좀 더 구체적으로 살펴보면, 기능성의 하위 특성 중 적절성, 정밀성, 상호운영성에서 윈도우즈가 리눅스보다 윈도우즈가 우수한 것으로 나타났다. 반면 리눅스는 순용성과 보안성에서 우수한 것으로 나타났다. 특히 보안성에서 리눅스가 우수한 것으로 평가된 점이 주목되는데 이는 리눅스는 공개된 소스코드이기 때문에 수많은 사용자들에 의한 검증 과정을 거쳤기 때문인 것으로 판단된다.

신뢰성의 부특성 중 성숙성과 회복성에서

윈도우즈가 고장허용에서는 리눅스가 약간 우수한 것으로 조사되었다.

사용성은 윈도우즈가 user interface 장점에 기반하여 이해성, 학습성, 운용성 모두에서 우수한 것으로 나타났다. 여기에는 많은 사용자가 PC를 다루면서 이미 윈도우즈 환경에 익숙하다는 점이 작용했을 개연성이 인정된다.

효율성면에서는 시간행동과 자원행동 모두 리눅스가 윈도우즈 보다 우수하게 나타나 이는 기존 문헌에서 주장된 내용과도 일치하여 본 연구결과의 신뢰성을 가능케 볼 수 있다.

마지막으로 이식성의 부특성인 적용성, 설치성, 부합성, 대체성 등에서 윈도우즈가 약간 앞서는 것으로 평가되었다.

본 연구의 제품 비교에는 상대적 비율척도에 의한 자료와 이들의 기하평균을 이용한 관계로 제품간 차이에 대한 t-검정과 같은 통계적 검정은 이론적으로 성립되지 않는다. 따라서 단순한 우열만을 중심으로 결과를 해석하였음을 밝혀둔다.

<표 3> 윈도우즈와 리눅스 비교

품질 특성	중요도	상대적 우수성	
		윈도우즈	리눅스
기능성	0.1867		
적절성	0.0279	0.6555+	0.3445
정밀성	0.0324	0.6282+	0.3718
상호 운영성	0.0304	0.6555+	0.3445
순용성	0.0330	0.4603	0.5397
보안성	0.0630	0.3741	0.6259+
신뢰성	0.2626		
성숙성	0.0738	0.5435	0.4565
고장 허용	0.0520	0.3762	0.6238+
회복성	0.1368	0.5084	0.4916
사용성	0.1839		
이해성	0.0677	0.6792+	0.3208
학습성	0.0558	0.7424+	0.2576
운용성	0.0604	0.7165+	0.2835
효율성	0.1554		
시간행동	0.0836	0.4581	0.5419
자원행동	0.0719	0.4536	0.5464+
이식성	0.2113		
적응성	0.0539	0.6851+	0.3149
설치성	0.0476	0.6611+	0.3389
부합성	0.0480	0.6337+	0.3663
대체성	0.0617	0.5313	0.4687
품질지수 (Qk)		0.5597	0.4402

+ : 상대적으로 우수하다고 판단됨

5. 결론

본 연구에서는 사용자에 의해 지각된 운영체제 품질 비교 방안에 관한 연구를 수행하였다. 이를 위하여 우선 ISO 9126 품질모델에 근거한 운영체제 소프트웨어 평가기준을 계층적으로 분류하고, 계층분석과정을 통한 평가모형을 제시하였다. 또한 제시한 평가모형을 실제로 윈도우즈와 리눅스에 적용하여 예시적으로 분석하였다.

본 연구의 결과 및 의의는 다음과 같이 제시할 수 있겠다.

첫째, 사용자의 운영체제에 대한 평가기준의 중요도가 신뢰성과 이식성에 있음이 밝혀졌으며 그 다음으로 기능성, 사용성, 효율성 순서로 중요한 것으로 밝힌바 있다.

둘째, 사용자 측면에서 볼 때, 본 연구의 결과는 향후 운영체제 사용자에 대해 제품선택의 지침이 될 수 있다는 점에서 의의가 있다. 따라서 세부적 평가 항목의 중요도 변화는 최종 사용자로 하여금 미래 지향적인 제품선택을 가능하게 하며 각 평가 기준에 따른 제품간 우열은 사용목적에 보다 적절한 제품을 선택하는데 도움이 될 것이다.

셋째, 공급자에게 제공하는 의미로서는 현재 시장에서 운영체제에 대한 품질 측면의 만족을 높이기 위한 핵심요소 및 향후 자사 제품의 개선점 파악에 본 연구의 결과가 도움이 될 것으로 사료된다.

참고문헌

- [1] 김해진, "리눅스 연구 개발 현황 및 방향," 정보처리학회지 6권 6호, 1999.
- [2] 윤민석, 이영, 성삼경, "유집수성 목표하의 소프트웨어 개발방법 평가에 관한 실증연구: ANP 기법을 중심으로", 한국경영과학회지 24권 4호, 1999.
- [3] 이상석, 윤민석, "통계처리용 소프트웨어 패키지의 품질 비교에 관한 연구", 품질경영학회지 27권 1호, 1999.
- [4] 이철원, 김홍근, 박태규, "리눅스 보안 연구개발 동향," 정보처리학회지 6권 6호, 1999.
- [5] 조순복, 주상호, 『운영체제』, 세화, 2000.
- [6] 진대제, "우리나라의 리눅스 발전방향", 정보처리학회지 6권 6호, 1999.
- [7] 『OS 기술시장보고서』, 한국전자통신연구원, 2001.
- [8] Boehm, B.W., J.R. Borwn, M. Lipow, G.J. MacLeod, and M.J. Merritt, Characteristics of Software Quality, North-Holland, New York, 1978.

- [9] Chidung Lac, Jean_Luc Raffy, "A Tool for Software Quality", IEEE, pp. 144-150, 1992.
- [10] Davis, F.D., "Perceived Usefulness, Ease of Use, and User Acceptance of Information Technology", MIS Quarterly vol.13, no. 3, pp. 319-339, 1989.
- [11] Dromey, R.G., "A Model for Software Product Quality," IEEE Transactions on Software Engineering, vol. 21, no.2, pp.146-162, 1995.
- [12] Evans, M. and J. Marviniak, Software Assurance and Management, John Wiley & Sons, New York, 1987.
- [13] Fritz, C.A. and B.D. Carter, *A Classification, and Summary of Software Evaluation and Selection Methodologies*, Computer Science Technical Report No. 940823, Mississippi State University, 1994.
- [14] IEEE Standard for a Software Quality Metrics Methodology, pp.19-20, 1992.
- [15] ISO 8402, Quality Management and Quality Assurance - Vocabulary, ISO, 1994.
- [16] ISO 9126, Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use, ISO, 1991.
- [17] Jung H.W. and M.S. "A Software Quality Evaluation and Resource Allocation Model", Proceedings of 5th. European Conference on Software Quality, 1996.
- [18] Kekre, S., M.S. Krishnan and K. Srinivasan, "Drivers of Customer Satisfaction for Software Products: Implications for Design and Service Support", *Management Science*, Vol. 41, No. 9, pp. 1456-1470, 1995.
- [19] McCall, J.A., Richards, P.K. and Walters, G.F., Factors in Software Quality (3 Vol.), RADC-TR-77-369, 1977.
- [20] Saaty, T. L., *The Analytic Hierarchy Process*, McGraw-Hill, 1980.
- [21] Saaty, T. L., *The Analytic Network Process*, RWS Publications, 1996.
- [22] Tervonen, I., "Support for Quality-Based Design and Inspection", Inc., Singapore, 1992.
- [23] Yoon, M.S., S/W Quality Evaluation Model Using the AHP, Thesis for The Degree of Doctor, Korea University, 1997.

- [24] Zahedi F., "Database Management System Evaluation and Selection Decisions", Decision Science vol. 34, no. 1, pp. 91-109, 1985.