

應用論文

QFD 기법을 이용한 패키지 소프트웨어의 품질 요구사항 획득*

유영관 · 이종무

한라대학교 경상학부

Acquiring Quality Requirements of Package Software using Quality Function Deployment Technique

Young Kwan Yoo · Jong Moo Lee

School of Business and Economics, Halla University

Abstract

In this paper, the QFD(quality function deployment) technique is used to acquire the quality requirements of package software which is sold in commercial software shop. To select the principal quality characteristics of package software, an user survey was conducted for a quality model developed based on the international standards and the previous research. The software user is divided into three groups: the general user, the expert, and the developer. Based on the survey results, the top 10 quality characteristics are selected including accuracy, suitability, installability, de-installability, and maturity. Then, the QFD technique is applied to choose the software quality attributes which have strong correlation with the selected quality characteristics, some of which are traceability, accuracy, modularity, completeness, and self-descriptiveness. The developed procedure can be used in evaluating the quality of package software with lower costs and efforts.

1. 서론

고품질의 소프트웨어를 개발하기 위한 노력은 소프트웨어 품질에 관한 명확한 이해와

함께 먼저 소프트웨어 사용자의 품질요구사항(quality requirements)이 구체적으로 무엇인지를 파악해야만 성공할 수 있다. 소프트웨어의 품질에 관한 이해는 Boehm (1981)을 비롯한 여러 기존의 연구와 함께 ISO/IEC 9126(2000) 품질모형과 그 특성에 관한 표준 등을 통하여 가능하지만, 실제 우리 산업현

* 본 논문은 한국전자통신연구원의 연구비 지원에 의해 수행되었음.

장에서 적용과 그 이해 정도는 미미한 수준이다.

소프트웨어개발주기 상에서 사용자 요구사항의 분석은 설계단계 이후 시험 등의 각 단계별 품질 검증의 기본이 되며, 또한 배포 및 설치, 유지보수로의 이행을 위한 품질 확인의 기준선이 되므로, 대부분의 소프트웨어 개발 프로젝트에서는 최종 성패를 좌우하는 시발점을 여기에 두고 있다. 이처럼 중요한 사용자 요구사항의 파악 및 분석은 향후 개발 프로세스 상에서 품질검증 및 확인을 하기 위한 기본 항목들을 체계적으로 도출하는데 필수적이며, 특히 품질평가를 통한 소프트웨어 인증을 목표로 한 품질인증 항목의 도출에는 반드시 요구되는 것이다.

소프트웨어 품질에 관한 사용자들의 다양한 요구는 여러 관점에 따라 정의되고 이해되어 왔다. 소프트웨어 품질모형을 비롯한 품질특성의 체계적인 틀과 그 적용을 위한 국제적인 표준은 ISO/IEC 9126(2000)과 ISO/IEC 14598(1999) 등을 통해 알 수 있다. 이 문서들에는 ISO/IEC 12207(1994)에서 제시하는 소프트웨어 개발 프로세스 품질이 어떻게 최종 소프트웨어 제품의 사용 품질(quality in use)에 관련되며, 특히 각기 다른 사용 경험과 다양한 기술 수준을 가진 사용자들의 품질요구사항을 내·외부품질(internal and external quality)과 사용 품질 등을 통하여 평가해 낼 수 있는가를 설명하고 있다.

소프트웨어 사용자의 품질요구사항의 명세화와 정형화 방법은 소프트웨어공학의 요구사항 관련 연구 결과들을 참고할 수 있다. 예를 들면, Deutsch와 Willis(1988)의 소프트웨어 품질공학에 관한 기법연구와 미 국방성의 소프트웨어 요구 명세(DI-MCCR-80025)

를 위한 DOD-STD- 2167에서 제시하는 품질요소와 기준, 그리고 이들 요구사항의 정량적인 구현 항목으로의 전환 예를 들 수 있다.

소프트웨어 사용자의 품질 요구사항에 관한 조사 연구의 대표적 예로는 노르웨이의 PROFF 프로젝트(Stalhane et al. 1997)를 들 수 있다. 이 연구에서는 서비스 품질에 관한 ISO/IEC 9004와 소프트웨어 제품 품질에 관한 ISO/IEC 9126에 근거해 도출된 품질요소들을 대상으로, 소프트웨어 사용자들의 요구사항에 관한 중요도를 정량적으로 측정 수집하여 소프트웨어의 제품 유형별로 분석 제시하였다.

그러나 기존의 이론적인 연구결과와 일부 프로젝트의 수행 결과를 직접적으로 우리 현실에 적용하기에는 여러 가지 문제점을 가질 수밖에 없다. 우선 ISO/IEC 9126과 ISO/IEC 14598 표준에는 구체적인 적용 방법의 제시가 없다. 따라서 주어진 환경에 따라 개별적인 현장 조사와 전문가의 주관적 판단에 따른 분석 연구가 수행되어야 한다. PROFF 프로젝트의 경우, 사용자의 품질 요구사항에 관한 연구 결과는 국제적인 벤치마킹 차원의 가치는 높다고 할 수 있으나 국내에 그대로 적용하기에는 적절한 검증이 필요하다. 또한 이 연구는 소프트웨어의 구입에 영향을 미치는 품질요소(quality factor)가 무엇인가만을 조사하였으며, 소프트웨어의 품질 측정과 인증을 위한 품질속성(quality attributes/criteria)의 선정은 시도하지 않고 있다.

본 연구에서는 시중에서 일반인들에게 시판되고 있는 상용 패키지 소프트웨어 즉, COTS(Commercial Off The Shelf Software) 중에서 응용 소프트웨어(application software)에 대하여 사용자의 품질 요구사항

을 조사하고, 이를 바탕으로 COTS 제품의 품질 측정 및 인증에 효율적으로 활용될 수 있는 주요 품질 속성들을 선정하고자 한다. 연구 방법을 요약하면 다음과 같다. 먼저 기존의 여러 연구와 표준을 바탕으로 COTS의 특성을 고려한 품질 모형을 도출한다. 다음에 소프트웨어의 사용자를 일반인, 전문가 집단, 개발자로 나누어 품질 특성의 상대적인 중요도에 대해 설문 조사한다. 조사 결과를 바탕으로 주요 품질 특성들을 선정한 후, 품질특성과 품질속성의 상관성을 분석하고 품질기능전개(QFD) 기법을 이용하여 주요 품질속성들을 도출한다.

2. 소프트웨어 품질모형과 품질특성

소프트웨어의 품질 모형은 여러 연구자와 단체에 의해 제시되어 왔다. Boehm(1981)은 최초로 소프트웨어 제품의 품질을 정량적으로 측정 평가할 수 있는 품질모형을 제시하였으며, 이후 McCall, Deutsch와 Willies(1988) 등에 의해 품질 모형이 개발되었다. 또한 IEEE 1061(1992)에서는 표준은 아니지만 정보 참조로서 사용 가능한 품질모형을 예시하고 있다.

학자와 단체마다 다르게 제시되는 품질 모형의 표준화를 위해 ISO/IEC/JTC1/SC7/WG6에서는 품질 특성과 품질 척도를 정의한 ISO/IEC 9126 시리즈를 개발하였다. 이 중 ISO/IEC 9126-소프트웨어 품질특성과 메트릭(2000)에서는 소프트웨어의 품질을 측정 평가하기 위한 품질모형을 계층구조로 세분화되어 표현하였다. 최상위 계층은 사용자 관점에서 소프트웨어의 품질목표를 정의하

고, 제2계층은 품질목표를 달성할 수 있는 품질특성(quality characteristics)으로, 제3계층은 상위특성을 측정하는 구체적인 부특성(sub-characteristics)으로 구성된다. 그리고 최하위 계층에는 부특성을 측정할 수 있는 품질속성(quality attributes)과 메트릭(metric)이 위치하게 된다. 품질 특성은 기능성(functionality), 신뢰성(reliability), 사용성(usability), 효율성(eficiency), 유지보수성(maintainability), 이식성(portability) 등 6개이고, 이들은 다시 적절성(suitability), 성숙성(maturity) 등 29개의 부품질특성으로 세분되며, 품질속성은 완전성, 추적성, 일관성 등 40개의 속성으로 구성된다.

본 연구에서는 사용자의 품질요구를 반영하는 주요 품질특성을 선정하기 위해 품질모형에 관한 선행 연구와 COTS 제품의 특성을 고려하여 ISO/IEC 9126의 품질 모형을 확장하였다. 즉, ISO/IEC 9126을 바탕으로 COTS 제품의 특성 상 필요하다고 생각되는 양립성, 제거성, 재사용성을 부품질특성으로 부가하였다(<표 1>). 이 세 가지 부품질특성의 의미는 다음과 같다. (나머지는 ISO/IEC 9126을 참조하기 바람)

양립성: 인간의 기대심리에 부응할 수 있는 소프트웨어의 능력(인간공학적 설계 등)

제거성: 소프트웨어가 규정된 환경에서 제거될 수 있는 능력

재사용성: 소프트웨어의 일부를 다른 응용 프로그램에 사용할 수 있는 능력

양립성은 소프트웨어의 사용성에 관계된 부품질특성으로써 특히 소프트웨어의 사용법을 처음 익힐 때 필요한 특성이다. 제거성은

<표 1> 본 연구에서 사용된 품질 모형

| 주품질 특성 | 부품질특성 | Reference |
|--------|-----------------------------|---------------------------|
| 기능성 | 적절성(suitability) | ISO 9126 |
| | 정밀성(accuracy) | ISO 9126 |
| | 상호운용성(interoperability) | ISO 9126 |
| | 보안성(security) | ISO 9126 |
| | 기능성 준수성(compliance) | ISO 9126 |
| 신뢰성 | 성숙성(maturity) | ISO 9126 |
| | 고장허용성(fault tolerance) | ISO 9126 |
| | 회복성(recoverability) | ISO 9126 |
| | 신뢰성 준수성 | ISO 9126 |
| 사용성 | 이해성(understandability) | ISO 9126 |
| | 학습성(learnability) | ISO 9126 |
| | 운용성(operability) | ISO 9126 |
| | 호감성(attractiveness) | ISO 9126 |
| | 양립성(communicativeness) | IEEE 1061 |
| | 사용성 준수성 | ISO 9126 |
| 효율성 | 시간행동성(time behavior) | ISO 9126 |
| | 자원이용성(resource utilization) | ISO 9126 |
| | 효율성 준수성 | ISO 9126 |
| 유지보수성 | 분석성(analyzability) | ISO 9126 |
| | 변경성(changeability) | ISO 9126 |
| | 안정성(stability) | ISO 9126 |
| | 시험성(testability) | ISO 9126 |
| | 유지보수성 준수성 | ISO 9126 |
| 이식성 | 적용성(adaptability) | ISO 9126 |
| | 설치성(installability) | ISO 9126 |
| | 제거성(de-installation) | |
| | 공존성(co-existence) | ISO 9126 |
| | 대체성(replaceability) | ISO 9126 |
| | 재사용성(reusability) | Dutsch & Willis McCall |
| | 이식성 준수성 | ISO 9126 |

소프트웨어를 쉽게 제거할 수 있는 정도로서 설치성과 더불어 소프트웨어 사용자들의 주관심사이다. 재사용성은 많은 학자들이 언급하고 있는 주요한 특성이지만 ISO/IEC 9126

에는 제외되어 있다.

3. 품질 특성의 중요도 설문 조사 및 결과

도출된 품질 특성들의 상대적인 중요도를 알아보기 위해 사용자들을 대상으로 설문 조사를 수행하였다. 본 연구에서는 일반적으로 시중에서 구할 수 있는 상용 소프트웨어의 사용자를 일반 사용자, 전문 사용자, 개발자로 나누었는데, 일반 사용자로는 대학생, 전문 사용자로는 품질 평가 관련 경험이 있는 PC통신의 컴퓨터 관련 동호회원, 그리고 개발자로는 국내 소프트웨어 개발 업체의 연구원을 선정하였다.

각 설문 항목은 품질특성에 대한 표준 용어를 사용하여 설명하였으며, 바로 옆에 중요도를 표시하도록 하였다. 중요도는 5점 척도를 사용하였는데, 중요도가 높을수록 점수가 커지도록 하였다(<표 2>).

대학생의 경우에는 강의실에서 직접 설문 조사를 하고, 전문 사용자와 개발자는 우편과 이메일을 이용하여 조사하였다. 총설문 중에서 성의 없는 답변을 제외하면 일반 사용자(대학생) 집단은 54부, 전문 사용자(컴퓨터관련 동호회원) 집단은 8부, 그리고 개발자 집단은 21부가 회수되었다.

설문 조사의 결과는 <표 3>과 같이 정리되었다. 이 표에 의하면 평가 집단에 따라 중요하게 생각하는 부 품질특성에 많은 차이가 있음을 알 수 있다.

각 집단 별 중요도 상위 10개 부품질 특성을 정리하면 다음의 <표4>와 같다. <표 4>에서 굵은 글씨로 쓴 것은 모든 사용자 집단에서 상위 10위 내에 선발된 것들이다. 즉,

<표 2> 부품질특성의 평균 중요도

| 특성 | 전체 | 일반인 | 개발자 | 전문가 |
|--------------|------|------|------|------|
| 적절성 | 4.16 | 4.13 | 4.38 | 3.75 |
| 정밀성 | 4.28 | 4.39 | 4.00 | 4.25 |
| 상호운용성 | 3.73 | 3.89 | 3.38 | 3.63 |
| 보안성 | 3.70 | 3.83 | 3.43 | 3.50 |
| 기능성 준수성 | 3.44 | 3.46 | 3.33 | 3.63 |
| 성숙성 | 3.84 | 3.85 | 3.86 | 3.75 |
| 고장허용성 | 3.71 | 3.85 | 3.43 | 3.50 |
| 회복성 | 3.94 | 4.17 | 3.33 | 4.00 |
| 신뢰성 준수성 | 3.36 | 3.35 | 3.43 | 3.25 |
| 이해성 | 3.59 | 3.69 | 3.60 | 2.88 |
| 학습성 | 3.49 | 3.52 | 3.57 | 3.13 |
| 운용성 | 3.67 | 3.63 | 3.86 | 3.50 |
| 호감성 | 3.19 | 3.17 | 3.43 | 2.75 |
| 양립성 | 3.19 | 3.06 | 3.48 | 3.38 |
| 사용성 준수성 | 3.28 | 3.30 | 3.24 | 3.25 |
| 시간행동성 | 3.81 | 3.82 | 3.81 | 3.75 |
| 자원이용성 | 3.81 | 3.83 | 3.91 | 3.38 |
| 효율성 준수성 | 3.47 | 3.48 | 3.52 | 3.25 |
| 분석성 | 3.69 | 3.96 | 3.24 | 3.00 |
| 변경성 | 3.52 | 3.61 | 3.38 | 3.25 |
| 안정성 | 3.78 | 3.80 | 3.67 | 4.00 |
| 시험성 | 3.66 | 3.76 | 3.38 | 3.75 |
| 유지보수성 준수성 | 3.34 | 3.39 | 3.19 | 3.38 |
| 적응성 | 3.73 | 3.82 | 3.71 | 3.25 |
| 설치성 | 3.99 | 4.04 | 4.00 | 3.63 |
| 제거성 | 3.96 | 4.02 | 3.91 | 3.75 |
| 공존성 | 3.65 | 3.69 | 3.55 | 3.63 |
| 대체성 | 3.53 | 3.67 | 3.33 | 3.13 |
| 재사용성 | 3.41 | 3.63 | 3.00 | 3.00 |
| 이직성 준수성 | 3.32 | 3.36 | 3.24 | 3.25 |

정밀성, 적절성, 설치성, 제거성, 성숙성은 모든 사용자 집단에서 높은 중요도 점수를 받은 특성들으로써 상용 소프트웨어의 품질 평가 시 반드시 평가되어야 할 특성들이라고 할

<표 3> 설문 항목의 일부 예

| 부품질 특성 | 설 명 | 중요도 | | | | |
|-----------|---|-----|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 상호운 용성 | 소프트웨어 제품이 하나 이상의 시스템과 상호작용할 수 있는 능력. 예를 들면, 스프레드 시트 프로그램을 워드 프로세서 내에서 연결 사용하는 기능이 보장되는 것. | | | | ✓ | |

수 있다. 이 밖에 회복성, 시간행동성, 자원 이용성, 안정성, 상호운용성 등도 주목해야 할 특성들이다.

상위 10개 품질부특성 중 제거성을 제외하면 모두 ISO/IEC 9126의 품질모형에 있는 부품질특성들이다. 제거성은 COTS의 특성을 고려하여 연구에 포함시킨 것으로, 많은 사용자들이 소프트웨어의 설치뿐만 아니라 제거의 용이성도 거의 동등한 중요도로 요구하고 있는 것을 알 수 있다. 양립성은 의외

<표 4> 사용자 별 상위 10개 부품질 특성

| 순위 | 전체 | 일반인 | 개발자 | 전문가 |
|----|--------|--------|--------|--------|
| 1 | 정밀성 | 정밀성 | 적절성 | 정밀성 |
| 2 | 적절성 | 회복성 | 정밀성 | 안정성 |
| 3 | 설치성 | 적절성 | 설치성 | 회복성 |
| 4 | 제거성 | 설치성 | 제거성 | 적절성 |
| 5 | 회복성 | 제거성 | 자원 이용성 | 제거성 |
| 6 | 성숙성 | 분석성 | 성숙성 | 성숙성 |
| 7 | 시간 행동성 | 상호 운용성 | 운용성 | 시간 행동성 |
| 8 | 자원 이용성 | 성숙성 | 시간 행동성 | 시험성 |
| 9 | 안정성 | 고장 허용성 | 적응성 | 설치성 |
| 10 | 상호 운용성 | 자원 이용성 | 안정성 | 공존성 |

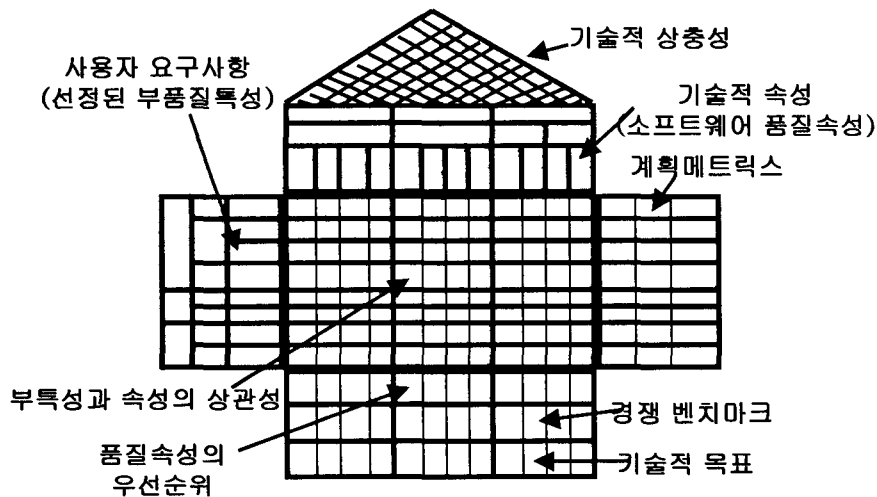
로 개발자 집단 외에는 모두 하위권에 랭크됨으로써 아직까지는 소프트웨어의 기본적인 기능에 대한 요구가 상대적으로 강한 것으로 나타났다. 재사용성도 모두 하위권에 랭크되었는데, 이 특성은 사실상 소프트웨어의 사용자보다는 개발자 입장에서 중요한 특성이므로 예측할 수 있는 결과라고 할 수 있다. 상위 10개 품질부특성들을 품질특성 별로 구분해 보면, 기능성(정밀성, 적절성, 상호운용성), 이식성(설치성, 제거성), 신뢰성(회복성, 성숙성), 효율성(시간행동성, 자원이용성), 유지보수성(안정성)의 순으로 나타났으며, 사용성은 해당되는 것이 없는 것으로 나타났다.

4. QFD 기법을 이용한 품질속성의 선정

4.1 QFD 기법의 응용

1960년대 후반 일본의 Misuno와 Akao에 의해 제안된 QFD(Quality Function Deployment) 기법은 70년대 일본의 제품 경쟁력을 높이는데 크게 기여하였으며, 이후 여러 나라에서 다양한 분야에 적용되고 있다 (Cohen 1995). QFD 기법은 소프트웨어 공학 분야에도 널리 활용되고 있으며(Elboushi와 Sherif 1997, Haag et al. 1996, Zultner 2000), Deutsch와 Willis(1988)에서 제시하고 있는 사용자의 품질요구에 대한 정형화 방법론은 그 내용상 QFD 기법의 응용으로 설명될 수 있다.

본 절에서는 Deutsch와 Willis(1988)에서 제시한 개념적인 방법론을 바탕으로 이를 QFD 기법의 틀로 재구성하여, 패키지 소프트웨어의 품질 평가에 사용될 품질속성을 선정한다. QFD 기법의 일반적인 방법과 본 연구에의 적용을 살펴보면 <그림 1>과 같다. 고객의 품질요구는 사용자의 품질 요구 즉, 설문조사로 선정된 부품질특성으로 주어지며



<그림 1> QFD 기법의 응용

(왼쪽부분), 기술적 품질요소는 소프트웨어 품질속성으로 주어진다(위부분). 부품질특성과 속성간의 상관관계는 가운데에 위치하며, 최종적으로 구하는 항목인 속성들의 우선 순위는 아래에 위치하게 된다. 본 연구는 부품질특성으로부터 품질속성을 선정하는 것이 목적이므로, 품질속성들간의 기술적 상충성을 나타내는 품질의 집(house of quality)의 최상단과 우측의 품질계획, 그리고 하단의 벤치마킹 등은 본 연구에서는 제외하였다.

4.2 부품질특성과 품질속성의 상관

품질속성 또는 내부 품질특성(internal quality characteristics)은 사용자의 요구를 나타내는 품질특성이나 부특성이 직접적인 측정과 평가가 어려우므로, 이러한 외부 품질특성(external quality characteristics)들을 소프트웨어의 개발 과정에서 내부적으로 측정할 수 있는 방법이 필요하게 됨에 따라 도입되었다. 품질모형과 마찬가지로 품질속성의 종류와 내용도 연구자나 단체에 따라 차이가 있다. 품질특성과 품질속성간의 상관관계는 대표적으로 ISO/IEC 9126(2000), Deutsch와 Willis(1988), 그리고 Abel과 Rout(1993) 등에서 찾아 볼 수 있다.

본 연구에서는 이상의 세 가지 연구 결과를 참고하여 공통적인 품질속성을 선정하고 상관관계를 정리하였다. 세 연구에서 제시하고 있는 품질속성들 중 공통적으로 제시된 27개의 품질속성들을 추출하고, 이 품질속성들과 앞 절에서 선정한 부품질특성과의 상관관계를 강, 중, 약의 세 가지 유형으로 구분하였다. 이 때, 세 연구에서 공통적으로 상관성을 중시하는 경우는 강의 상관성, 일부에

<표 5> 주요 부특성과 품질속성의 상관관계 (강=9, 중=3, 약=1)

| 부특성 품질속성 | 적절성 | 정밀성 | 상호운용성 | 성숙성 | 회복성 | 시간행동성 | 자원이용성 | 안정성 | 설치성 | 제거성 |
|-------------|-----|-----|-------|-----|-----|-------|-------|-----|-----|-----|
| 정밀성 | | 9 | 1 | 1 | | | | | | |
| 예외관리 | | 1 | | 1 | 9 | | | | | |
| 중대성 | | | | | | | | 3 | | |
| 자율성 | | | | | 3 | | | 1 | 3 | 3 |
| 공통성 | 1 | | 3 | | | | | | 3 | 3 |
| 완결성 | 9 | 3 | | 3 | 1 | | | | | |
| 일관성 | 3 | 3 | | 3 | 3 | | | 1 | | |
| 분배성 | | 1 | | | 3 | | | | | |
| 문서품질 | | | | | 1 | | | | | |
| 통신효율 | | | | | | 9 | 1 | | | |
| 처리효율 | | | | | | 9 | 1 | | | |
| 저장효율 | | | | | | 1 | 9 | | | |
| 기능범위 | 1 | | 1 | | | | | 1 | | |
| 일반성 | | | | | | | | 1 | 1 | 1 |
| 독립성 | | | 1 | | | | | 1 | 3 | 3 |
| 모듈성 | | | 1 | 3 | 3 | | | 3 | 9 | 9 |
| 운용성 | | | | | | | | | | |
| 안전관리 | | 1 | | | | | | | | |
| 자기기술성 | 3 | 3 | | 3 | 3 | | | 1 | 3 | 3 |
| 간결성 | | | | 9 | 9 | | | 1 | 1 | 1 |
| 지원성 | | | | | 1 | | | | | |
| 시스템접근성 | | | | | | | | | | |
| 시스템호환성 | 1 | | 1 | | | | | | | |
| 추적성 | 9 | 9 | | 3 | 3 | | | 3 | | |
| 훈련성 | | | | | | | | | | |
| 가상성 | | | | | | | | 1 | | |
| 가시성 | | | | | | | | | | |
| 합계 | 27 | 30 | 8 | 26 | 39 | 19 | 11 | 17 | 23 | 23 |

서 상위, 또는 중간 정도의 상관성으로 평가한 경우는 중의 상관성, 그리고 일부에서 중

4.3 부품질특성의 순위 가중치 결정과 정규화

품질속성과의 상관성을, 각 부품질특성 별로 정규화하기 위해서는 선정된 부품질특성의 순위에 따라 가중치를 부여해야 한다. 순위에 따른 가중치 부여 방법에는 여러 가지가 제시되고 있지만(Edwards 1982), 본 연구에서는 상위 10개 부특성에 대해 역수합계 순위법에 따라 가중치를 부여하였다. 이 방법은 순위의 차이를 분명히 나타내는 방법으로 널리 사용되고 있다. 다음의 <표 6>에는 부특성들의 순위와 순위 가중치, 그리고 이들과 <표 5>로부터 계산된 품질속성의 순위 값이 정리되어 있다. 계산 과정은 다음과 같다. 먼저 <표 5>에서 부특성별로(열별로) 상관 점수의 합계를 구한 후 그 열의 각 점수에 대해 나누면 부특성별로 정규화된 상관 점수를 얻는다(<표 6>의 중간 부분). 다음에 <표 6>에서 각 부특성의 순위 가중치를 이 상관 점수들에 곱한 후, 이 값들을 각 품질속성별로(행별로) 더하여 최종적으로 품질속성의 순위 값을 구한다(맨 오른쪽 열).

4.4 품질속성의 선정

<표 6>에서 품질속성들을 순위에 따라 정리하면 다음의 <그림 2>를 얻는다. 이 그림으로부터 품질속성의 순위 값은 추적성, 정밀성, 모듈성, 완결성, 자기기술성, 일관성, 간결성, 공통성, 저장효율성, 자율성 등의 순서로 나타났다. 특히 상위 5개 품질속성이 전체의 57.4%, 그리고 상위 10개 품질속성이 전체의 80.1%를 차지함으로써, 27개 품질속성들 중 일부의 품질속성 만으로도 요구되는 품질특성들에 대한 평가가 상당 부분 수행될 수 있음을 알 수 있다.

선정된 품질속성들을 바탕으로 소프트웨어 제품에 대한 품질인증을 수행하기 위해서는 실제 항목 측정을 위한 메트릭(metric)의 개발이 필요하다. 메트릭은 ISO/IEC 9126 (2000) 등에 품질속성 별로 제시되어 있으므로 이를 참조하면 될 것이다.

5. 결론

객관적인 소프트웨어의 품질 인증을 위해서는 소프트웨어의 품질을 정량적으로 측정할 수 있어야 한다. 그러나 품질 인증을 위한 시간과 비용의 제약으로 인해 수많은 소프트웨어의 품질 특성들과 관련 속성들을 모두 측정한다는 것은 불가능하며, 따라서 효율적인 품질 측정 절차가 요구된다. 기존의 많은 연구들이 소프트웨어의 품질에 대한 사용자의 요구 사항들을 정형화하여 소프트웨어의 개발에 반영하는 일반적인 방법론은 제시하고 있으나, 개발된 소프트웨어의 품질을 사용자의 요구 사항을 반영하여 평가할 수 있는 구체적인 절차는 제시하지 않고 있다.

본 연구에서는 COTS 중 응용 소프트웨어를 대상으로 사용자들의 요구를 반영하여 소프트웨어의 품질을 평가할 수 있는 구체적인 절차를 개발하고 그 결과를 제시하였다. 국제 표준적인 품질모형인 ISO/IEC 9126 (2000)을 바탕으로 COTS의 특성을 고려한 몇 개의 부품질특성들을 부가한 품질 모형을 사용하여 설문을 실시하였으며, 설문 결과를 바탕으로 중요도가 높은 부품질특성들을 도출해 내었다. 그리고 QFD 기법을 이용하여 선정된 부품질특성들과 상관 관계가 높은 품질 속성들을 도출해 내어 이들을 중심으로 평가함으로써 많은 시간과 노력을 절약하면

서 효율적인 평가 작업이 될 수 있도록 절차를 제시하였다.

본 연구에서 제시된 절차에 의해 선정된 품질 속성들은 소프트웨어의 품질 평가뿐만 아니라, 사용자의 품질요구를 반영한 새로운 소프트웨어의 개발에도 사용될 수 있을 것이다. 또한 다른 유형의 소프트웨어에 대해서도 비슷한 절차에 의해 품질 평가를 수행할 수 있을 것이다.

본 연구는 소프트웨어의 품질 측정 및 평가를 위한 구체적인 절차의 개발에 중점을 두었다. 따라서 품질속성들에 대한 좀더 정확한 순위 값을 구하기 위해서는 좀더 많은 수의 표본에 대한 설문이 수행되어야 하며, 설문에 대한 좀더 진지한 답변을 유도할 수 있는 방법이 요구된다. 그러나 적은 수의 표본이지만 전문적인 답변을 기대할 수 있는 전문가와 개발자가 설문에 포함된 본 연구의 결과가 크게 벗어나지는 않을 것으로 기대된다.

감사의 글

논문 심사위원의 세심한 조언에 감사드립니다.

참고문헌

[1] Abel, D. and Rout, T.(1993), "Defining and specifying the quality attributes of software products", The Australian Computer Journal, Vol.25, No.3, pp105-112.
 [2] Boehm, B. W.(1988), Software Engineering Economics, Prentice -Hall,

NJ.
 [3] Cohen, L.(1995), Quality Function Deployment, Addison-Wesley.
 [4] Deutsch, M. and Willis, R.(1988), Software Quality Engineering, Prentice Hall, NJ.
 [5] Edwards, W. and Newman, J. R., (1982), Multi-attribute Evaluation, Sage Publishing.
 [6] Elboushi, M. and Sherif, J.(1997), "Object-Oriented Software Design Utilizing Quality Function Deployment", J. of Systems Software, Vol.38, pp.133-143.
 [7] Haag, S., Raja, M., and Schkade, L.(1996), "Quality Function Deployment Usage in Software Development", Communications of the ACM, Vol.39, No.1, pp41-49.
 [8] IEEE Computer Society(1992), IEEE-STD-1061: IEEE Standard for a Software Quality Metrics Methodology.
 [9] ISO(1994), ISO/IEC 12119: Information Technology - Software Packages - Quality Requirements and Testing.
 [10] ISO(2000), ISO/IEC 9126: Information Technology - Software Quality Characteristics and Metrics.
 [11] ISO(1999), ISO/IEC 14598: Information Technology - Software Product Evaluation.
 [12] Stalhane, T., Borgersen. P., and Arnesen. K.(1997), "In Search of the Customer's Quality View", J. of

Systems Software, Vol.38, pp.85-93.

- [13] Zultner, R.(2000), "Software QFD for Very Rapid Development", Proceedings of 2nd World Conference on Software Quality, pp.67-73.