

응용논문

분석단계에서의 소프트웨어 품질 평가에 관한 연구
- A Study on Software Quality Evaluation in Analysis Phase -

장 영 숙*
Jang, Young Sook
권 영 식**
Kwon, Young Sik

Abstract

It is booming to use computer and information owing to the information society. Therefore, software demand is increasing rapidly. Enhancing utility value depends upon software quality. Software quality measurement and evaluation are necessary to satisfy the users who are using computers. In the present time, ISO/IEC 9126 define 6 quality characteristics and 21 sub-characteristics including 40 inter quality characteristics. We developed software evaluation technique using the check list in accordance with the ISO/IEC 9126 in analysis phase.

We evaluated the software quality in order to verify developing technique at S college. As a result of the evaluation, this college has found it necessary to maintain the software as soon as possible because academic systems are not adequate for users at this present time

1. 서 론

정보화 사회가 도래함에 따라 품질관리 또는 품질보증 대상 분야도 종전의 하드웨어(Hardware)에서 소프트웨어(Software) 중심의 활동을 하지 않으면 안 되는 국면에 있다 [3,5,6]. 그러나 소프트웨어 품질에 대한 우리의 관심은 그 중요성에 비해 아직도 부족한 실정이며, 저질 소프트웨어로 인한 유지보수 업무의 부담과 비용을 가중시키는 것이 현실이다. 이러한 유지보수 비용을 줄이기 위한 방법은 산업공학(IE : Industrial Engineering)의 관점에서 말하는 품질은 다 만들어 놓은 제품을 합격, 불합격 판정을 내리는 행위가 아니라 “제품을 처음부터 잘 만들어라(do it right the first time)” [8] 혹은 “제작 공정에서 품질을 주입하라(build quality into process)” [10]라는 원칙으로 소프트웨어를 개발하지 않으면 안 된다. 이렇게 함으로써 사용자 즉 고객이 만족할 수 있는 소프트웨어를 생산할 수 있다.

소프트웨어 품질의 특수성, 즉 말 그대로 ‘Soft’ 면을 극복하고 Yourdon[22]이 말한 “측

* 신성대학 품질관리과 교수

** 동국대학교 산업시스템공학부 교수

정할 수 없는 것은 관리될 수 없다(You can't manage what you can't measure)” 는 원칙에 입각하여 소프트웨어 품질평가를 시도해 보았다.

연구범위는 대학의 학사업무 전산화 일환으로 사용자 부서에서 요구한 내용을 해당 전산 부서에서 자체 개발한 응용 소프트웨어(application software)를 대상으로 하며, 소프트웨어 개발 주기 상에서 보면 유지보수 비용에 지대한 영향을 미치는 분석단계의 산출에 대한 품질평가 부분으로 제한한다.

본 논문 구성은 제2장에서 소프트웨어 품질평가, 제3장 적용사례, 제4장 결론 및 향후 연구 과제 순으로 전개했다.

2. 소프트웨어 품질평가

2.1 소프트웨어 품질 및 특성

ISO/IEC 9126[14]에서는 소프트웨어(software)를 “컴퓨터 프로그램과 절차, 규칙, 그리고 컴퓨터 시스템의 운용에 관련된 문서와 자료” 라고 정의하고 있으며, IEEE[12]에서는 소프트웨어 품질(software quality)을 “소프트웨어가 요구하는 속성들(desired combination of attributes)을 갖고 있는 정도” 라고 정의하고 있다. 여기서 요구하는 속성들이란 신뢰성, 유지보수성, 가용성 등을 말한다. 미 국방성[20]은 “소프트웨어의 명세 된 최종 아이템 사용을 수행하는데 가능한 소프트웨어의 속성 정도” 라고 정의하였다. 본 논문에서는 국제표준에 맞춰 ISO 8402[15]에서 정의한 “명시적(明示的) 이거나 묵시적(默示的)인 요구를 만족시키는 소프트웨어 제품의 능력과 관련된 소프트웨어 제품의 특성 전체” 라는 정의를 따른다.

소프트웨어 품질특성으로 Boehm[7]은 초기운용(as-is utility), 유지보수(maintainability), 이식성(portability)의 3차원으로 분류하고 7개의 intermediate constructs와 15개의 primitive constructs를 제공하고 있다. McCall[18] 11개, Evans[9]는 12개의 품질특성으로 구분하였다.

국제표준화기구 ISO(International Organization for Standardization)에서는 소프트웨어 품질보증 규격으로 제시한 ISO 9000-3[18]에 기반을 둔 ISO/IEC 9126[16]에서는 소프트웨어 품질평가에 사용할 수 있는 틀을 제공하고 있으며, 품질특성으로 기능성(functionality), 신뢰성(reliability), 사용성(usability), 효율성(eficiency), 유지보수성(maintainability) 및 이식성(portability)의 6개 품질주특성을 제시하고 있다. <표 1>은 이상에서 언급된 각 학자 및 기구(기관)에서 제시한 소프트웨어 품질특성에 관한 비교를 나타낸 것이다.

<표 1> 각 학자들의 소프트웨어 품질특성 비교

학자(기구)	수	소프트웨어 품질특성
Boehm	15	Device Independence, Self Containedness, Completeness, Robustness, Integrity, Consistency, Accountability, Device Efficiency, Accessibility, Communicativeness, Self Descriptiveness, Structureless, Conciseness, Legibility, Augmentability
McCall	11	Correctness, Efficiency, Flexibility, Integrity, Interoperability, Maintainability, Portability, Reliability, Reusability, Testability, Usability
Evans	12	Correctness, Efficiency, Expandability, Flexibility, Integrity, Interoperability, Maintainability, Portability, Reliability, Reusability, Usability, Verifiability
ISO/IEC	6	Functionality, Reliability, Usability, Efficiency, Maintainability, Portability

따라서 본 논문에서는 국제표준규격인 ISO/IEC 9126에서 제시하는 <표 1,2,3>의 품질주특성, 품질부특성 및 내부특성을 기준으로 평가를 실시하고자 한다.

<표 2> ISO/IEC 9126 품질주특성과 부특성

품질주특성 품질부특성	정의
기능성(functionality) 합목적성(suitability) 정확성(accuracy) 상호운용성(interoperability) 표준적합성(compliance) 보안성(security)	일련의 기능 존재와 규정된 기능 특성과 관련된 속성의 집합 지정된 업무를 수행하는데 필요한 기능이 존재하고 그 기능이 적절한 정도 소프트웨어가 올바른 결과 또는 합의된 결과를 출력할 수 있는 정도 소프트웨어가 지정된 시스템과 연결되어 상호 작용할 수 있는 정도 소프트웨어가 관련 표준, 지침, 규정 등을 따르는 정도 프로그램과 자료에 대해 인가되지 않은 접근을 방지할 수 있는 정도
신뢰성(reliability) 성숙성(maturity) 장애허용성(fault tolerance) 회복성(recoverability)	명시된 기간동안 명시된 조건에서 성능 수준을 유지하는 소프트웨어 능력과 관련된 속성 집합 소프트웨어 결함으로 인해 발생하는 고장의 빈도 소프트웨어 결함이나 인터페이스 상의 문제가 있더라도 지정된 수준의 성능을 유지하는 정도 고장에 의해 영향을 받는 데이터를 복구하고 성능을 회복할 수 있는 정도
사용성(usability) 이해성(understandability) 습득성(learnability) 운용성(operability)	사용을 위한 노력과 그러한 사용에 대한 개개인의 심사와 관련된 속성의 집합 사용자가 소프트웨어의 개념과 응용성을 인식하는 데 필요한 노력의 정도 사용자가 연산 제어, 입출력 등과 같은 적용 범을 배우는데 드는 노력의 정도 운용자가 소프트웨어를 적재, 초기화, 운용 및 통제, 종료하는데 드는 노력의 정도
효율성(efficiency) 시간효율성(time behavior) 자원효율성(resource behavior)	규정된 조건에서 소프트웨어의 성능 수준과 사용된 자원의 양 사이에 관련된 속성의 집합 소프트웨어 기능 수행에 따른 응답 시간, 처리 시간 소프트웨어의 기능 수행에 따른 자원 소요량 및 자원 사용 기간 등의 수준
보수성(maintainability) 분석성(analyzability) 변경성(changeability) 안정성(stability) 시험성(testability)	규정된 수정을 수행하기 위하여 필요한 노력과 관련된 속성의 집합 소프트웨어의 고장 원인이나 취약점을 진단하고 변경 부분을 식별하는데 드는 노력의 정도 소프트웨어의 환경 변화로 인해 드는 또는 결함 제거 및 변경에 따라 드는 노력의 정도 소프트웨어의 변경으로 인해 발생하는 예상치 못한 위협의 정도 변경된 소프트웨어를 검증하는데 드는 노력의 정도
이식성(portability) 환경적응성(adaptability) 설치성(installability) 규격적합성(conformance) 치환성(replaceability)	다른 환경으로 이전되는 소프트웨어 능력과 관련된 속성의 집합 소프트웨어를 다른 환경에 적응하는데 있어 고려된 사항 외에 다른 사항이 추가되지 않는 정도 소프트웨어를 지정된 환경에 설치하는데 필요한 노력의 정도 소프트웨어가 이식성과 관련된 표준이나 관례를 따르는 정도 다른 소프트웨어 환경에서 그를 대신해서도 사용될 수 있는 가능성 및 그에 따른 노력의 정도

2.2 외부특성과 내부특성

ISO/IEC 14598[13]에서는 품질모형으로 외부품질(external quality)과 내부품질(internal quality)로 구분하고 있다. 외부품질이란 특정 조건하에서 사용된 소프트웨어 제품이 “명시된 그리고 묵시적 요구(stated and implied needs)”를 만족시키는 품질정도를 의미한다. 이에 반하여 내부품질은 “명시된 그리고 묵시적 요구”를 만족시킬 수 있는지를 결정하는 소프트웨어 제품의 자체 속성의 집합을 의미한다.

따라서 외부품질은 <표 1,2>와 같이 ISO/IEC 9126에서 언급하는 기능성, 신뢰성, 사용성, 효율성, 보수성 및 이식성 6개 항목의 품질주특성 및 합목적성로부터 치환성까지 21개 항목의

품질부특성 집합이라고 볼 수 있다. 여기서 기능성은 5개, 신뢰성 3개, 사용성 3개, 효율성 2개, 보수성 4개 및 이식성 4개 항목의 품질부특성으로 구성되어 있다[1,3].

반면에 내부품질은 완전성부터 전달성까지 40개 항목의 내부특성 집합이며, <표 3>은 내부특성에 대해서 정의를 한 것이다[3].

평가 관점 측면에서 보면 외부특성은 사용자 또는 관리자 입장에서, 내부특성은 개발자 혹은 유지보수자 입장에서 평가되는 항목이라 말할 수 있다.

<표 3> 내부특성의 정의(일부분)

코드	내부특성	정 의
01	완전성(完全性) completeness	계획 또는 요구된 기능이 충분히 실현되었는지를 평가
02	추적가능성(追跡可能性) traceability	개발 공정에 따라 요구 명세로부터 실현된 것(기능)으로의 관련을 추적할 수 있는 성질
03	일관성(一貫性) consistency	설계기법이나 표기법, 용어, 기호 등이 통일되었는지를 평가
04	자기기술성(自己記述性) self descriptiveness	기능 및 기능과 기능의 연관을 설명하는 성질
05	무모순성(無矛盾性) coherence	시스템과 프로그램이 가지는 동일 기능에 모순이 없는 성질. 프로그램과 도큐먼트 사이에 모순이 없는 성질.
06	계산정확성(計算正確性) Accuracy	계산결과 및 출력이 요구하는 정도를 달성하는 성질
:	:	:
35	확장성(擴張性) Expandability	사양의 추가, 변경에 대하여, 용이하게 대응 가능하도록 준비되어 있는 성질
36	제품관리성(製品管理性) product management	소프트웨어의 개개 부품, 도큐먼트에 대한 구성, 이들의 정합성과 이력 등 제품의 관리가 쉽도록하는 성질
37	소프트웨어 독립성 s/w independence	소프트웨어가 특정의 소프트웨어 환경(os compiler, 언어, utility 등)에 의존하지 않고 동작 가능한 성질
38	기계독립성(機械獨立性) machine independence	시스템이 특정의 하드웨어 환경(시스템구성, 기종, 장치, 단말기 등)에 의존하는지를 평가
39	데이터독립성(獨立性) data independence	시스템이 특정 데이터 환경(데이터,DB,DBMS 등)에 의존하는지를 평가
40	전달성(傳達性) communicativeness	시스템의 입출력 형식이나 내용이 어느 정도 사용이 용이하게 통일되어 있는지를 평가

소프트웨어 개발주기별 내부특성간의 연관성은 <표 4>과 같이 분석단계에서는 완전성(completeness)을 포함하여 23개 항목, 설계단계 21개, 코딩단계 38개 및 테스트/유지보수단계 32개 항목이 있다. 이중 자기기술성, 무모순성, 견고성, 무결성, 설명성, 주목성 및 선택성 7개 항목은 소프트웨어 개발 모든 단계에 해당되므로 중요한 항목이라 하겠다.

<표 4> 소프트웨어 개발주기별 품질내부특성 관계

구 분 코드 내부특성	개발 주기			
	분석 단계	설계 단계	코딩 단계	테스팅 / 유지보수 단계
01 완전성	•	•	•	
02 추적가능성	•	•	•	
03 일관성	•	•	•	
04 자기기술성	•	•	•	•
05 무모순성	•	•	•	•
06 계산정확성	•		•	•
07 데이터공통성	•			•
08 통신절차공통성	•	•	•	
09 액세스가능성	•	•	•	
10 액세스제어성	•	•	•	
11 액세스감사성	•	•	•	
12 견고성	•	•	•	•
13 무결성	•	•	•	•
14 모듈성	•	•	•	
15 단순성		•	•	•
16 계층성		•	•	•
17 자기포함성			•	•
18 통일성		•	•	•
19 표현성		•	•	•
20 계층성	•	•		
21 설명성	•	•	•	•
22 비유성	•		•	•
23 완비성			•	•
24 주목성	•	•	•	•
25 적시성	•			•
26 적량성(기억성)	•		•	•
27 간결성		•	•	•
28 선택성	•	•	•	•
29 유도성			•	•
30 환경적합성	•		•	•
31 성력성			•	•
32 환경적합성			•	•
33 실행효율성			•	•
34 자원효율성			•	•
35 확장성	•		•	•
36 제품관리성			•	•
37 S/W독립성			•	•
38 기계독립성			•	•
39 데이터독립성			•	•
40 전달성			•	•

내부특성과 품질주특성 및 부특성과의 관련성과 관련 정도는 <표 5>와 같이 나타낼 수 있다[1,3].

<표 5> 외부특성과 내부특성의 관계(일부분)

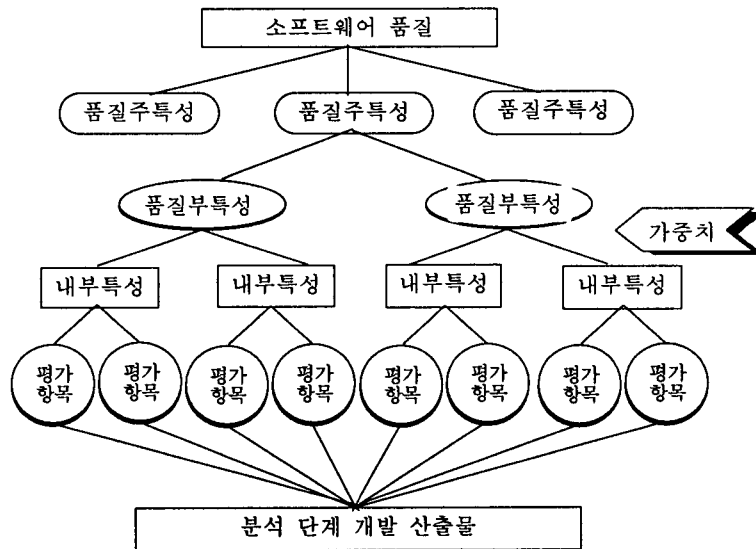
품질주특성	기능성					신뢰성			사용성			효율성		보수성				이식성					
	품질부특성	합목적성	정확성	상호운용성	표준적합성	보안성	성숙성	장애허용성	회복성	이해성	습득성	운용성	시간효율성	자원효율성	분석성	변경성	안정성	시험성	환경적응성	실치성	규격적합성	치환성	
코드 내부특성																							
01 완전성	●	○				●	△	△															
02 추적가능성	●					●	△	△							●	●	○	△					
03 일관성	○	○				●	○	●							●	●	○	○					
04 자기기술성	●	●				●		●							●	●		○		○			
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
37 S/W독립성															○	○		○	●	●	○	●	
38 기계 독립성															○	○		○	●	●	○	●	
39 데이터 독립성																			●	●	●	●	
40 전달성																			△	△		△	

● : 강한관계 ○ : 보통관계 △ : 약한관계

2.3 소프트웨어 품질평가

2.3.1 품질 특성 체계 및 평가 절차

소프트웨어 품질은 이상에서 말한 6개 품질주특성, 21개 품질부특성 및 40개 항목의 내부특성으로 구성되어 있다. (그림 1)은 이러한 특성을 중심으로 소프트웨어 품질특성 체계를 나타낸 것이다. 먼저 분석단계에서의 개발 산출을 대상으로 내부특성 값을 산출하기 위해서 개발된 평가 점검표에 따라 평가항목을 먼저 측정·평가한다. 이 값을 기초로 품질부특성 값을 얻을 수 있고, 나아가 품질주특성 값을, 최종적으로 소프트웨어 품질 값을 산출해 낼 수 있다.



(그림 1) 품질특성 체계

2.3.2 평가 점검표 설계

평가 점검표(check list)는 내부특성을 평가할 수 있도록 평가항목을 구분 기술하며, 이때 정량적, 정성적으로 평가할 수 있도록 항목은 상세하게 기술해야 한다. 평가 등급은 평가자의 주관성을 가능한 한 배제하기 위하여 5단계 평가등급(우수/양호/보통/미약/불량)으로 하고, 평가항목에 따라 평가 결과를 평가등급 난(□)에 표시(√)를 하면 된다. 만약 평가항목에 해당이 없으면 「해당 없음」 난에 표시하고, 이는 평가점수에 들어가지 않는다. 평가등급별로 획득한 점수를 해당 소계 난에 기록하고, 등급별 점수를 합산하여 총점 난에 기록하면 된다. <표 6>는 이러한 내용을 반영한 평가 점검표 양식이 되겠다[4].

<표 6> 평가 점검표

내부특성 :		해당□에 √표시를 한다					해당 없음
		우수	양호	보통	미약	불량	
No.	평가 항목	10	8	6	4	2	
:	:	□	□	□	□	□	□
소계							
총점							

품질의 공정성과 객관성을 확보하기 위해서는 평가자의 자격 기준이 대단히 중요하다. 그래서 현장 실무 경험과 전산 개발 경험이 풍부한 최소한 10년 이상의 전문가가 실시하는 것이 좋다. 또한 소프트웨어 평가를 위한 점검표 개발은 평가하고자 하는 대상에 따라 다양하게 개발됨으로써 평가 모델 적용의 일반화를 기할 수 있다고 본다.

2.3.3 내부특성 값, 품질부특성 및 품질주특성 값

내부특성 값 산출은 품질평가 점검표에서 측정된 총점을 가지고 식 (2.1)에 따라 백분율 값으로 나타내면 된다. 품질부특성 값 산출은 부특성 항목과 관련이 있는 내부특성을 도출하여 평가점검표에서 얻은 평가치를 기록하고, <표 5>에서 품질주특성, 품질부특성 및 내부특성간의 연관 관계에서 관련 정도에 따라 가중치를 부여한 다음 식 (2.2)에 따라 업무별로 산출하면 된다. 품질부특성 값을 이용하여 식 (2.3)에 의거 품질주특성 값을 산출해 낼 수 있으며, 소프트웨어 최종 값은 식 (2.4)과 같이 품질주특성 값을 평균함으로써 얻을 수 있다[2]. <표 7>은 이러한 품질특성 값 산출공식을 정리해 놓은 것이다.

<표 7> 품질특성 값 산출공식

구분	산출공식	기호설명	비고
내부특성 값	$A = \frac{\sum_{i=1}^n P_i}{10 \cdot n} \times 100$ (2.1)	A : 내부특성 값 P : 평가점수 n : 평가항목수	
품질부특성 값	$B = \sum_{i=1}^n (A_i \cdot W_i)$ (2.2)	B : 품질부특성 값 A : 내부특성 값 n : 내부특성 항목수	가중치 적용
품질주특성 값	$C = \frac{\sum_{i=1}^n B_i}{n}$ (2.3)	C : 품질주특성 값 B : 품질부특성 값 n : 품질부특성 항목수	
소프트웨어 최종 값	$D = \frac{\sum_{i=1}^n C_i}{n}$ (2.4)	D : 소프트웨어 최종 값 C : 품질주특성 값 n : 품질주특성 항목수	

본 연구에서는 내부특성 값에서 품질부특성 값을 산출할 때는 <표 5>의 관계정도에 따라 가중치를 적용하고 그 외 품질부특성 값에서 품질주특성 값, 품질주특성 값에서 최종 소프트웨어 값을 산출할 때 가중치 적용을 았기로 하며 향후 연구 과제로 남겨 놓는다. 그러나 기본적인 원칙은 소프트웨어 사용 용도에 따라 요구되는 품질특성이 다르므로 가중치 부여도 달라져야 한다고 본다.

2.3.4 결과 분석 및 프로젝트 관리방법

<표 8>는 식 (2.4)에 의해 산출된 최종 소프트웨어 품질 값으로 소프트웨어 개발 시 등급에 따라 프로젝트관리방법을 제시한 것으로, 원리는 제조 현장에서 흔히 사용되는 공정능력 지수를 소프트웨어 개발 프로젝트 관리지수로 응용해 본 것이다[2]. 표에서 보면 1, 2등급은 현상 유지를 하면서 프로그램 개발을 진행해도 좋으나, 3등급 이하는 사용자 요구사항이 충분히 반영되지 못한 상태에서 진행되므로 개발 후 사용자 이용 시 고객의 불만과 동시에 엄청난 유지보수 비용이 발생할 것으로 판단된다. 따라서 이때는 개발을 계속 진행하기보다는 개발 시 관련되는 문서를 토대로 다시 한번 사용자 요구사항이 충실히 반영되었는지를 점검해야 할 것이다.

<표 8> 평가결과에 따른 소프트웨어 프로젝트 관리방법

평점(%)	등급	결 과	관 리 방 법
100~90	1등급	충족도가 매우 좋음	관리 간소화나 개발비용 절감방법 고려
89~80	2등급	충족도가 높음	이상적인 상태이므로 유지해감
79~70	3등급	일부항목 수정요망	충족도에 불충분한 사항은 개선을 요함
69~60	4등급	충족하지 않음	오류가 많이 발생. 관리개선을 요함
59이하	5등급	전면적인 개선요망	긴급대책 강구

이와 같이 개발 초기 단계인 분석 단계에서 평가를 통하여 나타난 문제점을 신속히 피이드백(feedback)시켜 줌으로써 조기에 오류(error)를 시정할 수 있고 결과적으로는 유지보수비를 절감할 수 있다고 본다. 결함(error) 수정 비용과 관련하여 「1:10:100」 원칙에 의하면 분석(초기)단계에서 결함을 발견하여 수정하는 비용을 1로 보면 출하단계에서는 10, 사용자 단계에서는 100배의 비용이 든다는 원칙이다[11]. 따라서 초기단계의 품질평가 및 관리의 중요성을 아무리 강조해도 지나치지 않다. 또한 시정조치를 요구할 때에는 일방적인 지시나 무조건적인 개선을 요구하기보다는 가능한 한 해결방안이나 기법 등을 제안함으로써 소프트웨어의 품질향상을 기하는 것이 바람직하다.

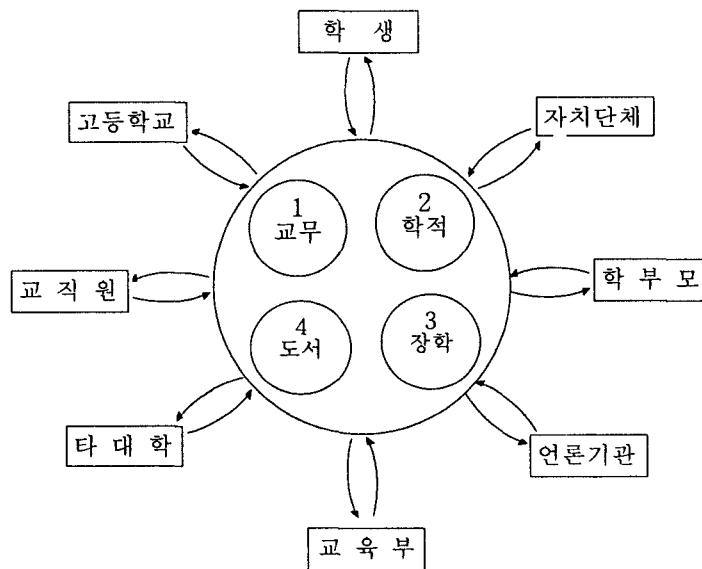
3. 적용사례

3.1 개요

개발된 기법을 S 대학의 주요 학사업무중 교무, 학적, 장학 및 도서 4종에 대해 적용해 보았다. 각 업무의 개략적인 내용을 보면

- 교무업무 : 교원인사관리(인사 기본, 연구실적, 교수출장), 교육과정관리, 수업관리 등
- 학적업무 : 학생학적현황(재적, 재학, 휴학, 제적 등), 성적관리, 증명서 발급 등
- 장학업무 : 신입생 장학생관리, 교내외 장학금 현황, 등급별 장학생 명단 관리 등
- 도서업무 : 수서관리, 도서정리, 대출관리 등이다.

이러한 학사업무의 개략적인 내용을 (그림 2)과 같이 체계 배경도(context diagram)로 나타내어 보았으며 학생, 교직원, 학부모, 교육부, 고등학교, 타 대학 등과 같은 조직(사람)과 관련을 가지고 업무는 이루어지고 있다[21].



(그림 2) 학사업무 체계 배경도

분석단계의 품질평가 대상으로는 자료흐름도(DFD : Data Flow Diagram), 자료사전(DD : Data Dictionary), 자료기술서(Mini-specification), 개발일정 및 공정표, 조직운영 및 인력 투입 계획표, 분석범위 파악표, 요구사항 및 문제분석표, 입출력 자료 정의표, 운영요건 조사서, 사용자 등급 구분표 및 데이터 용량 산출서 등이 있다.

3.2 내부특성 값 산출

먼저 내부특성 값 산출을 위해서는 개발된 <표 6> 평가 점검표 양식에 따라 평가항목에 따라 품질측정을 해야 한다. 이때 품질측정 및 평가는 현장 실무 경험이 있고, 전산 개발 및 시스템 분석·설계 능력이 있는 15년 이상인 전문가가 실시했다. <표 9>는 교무업무와 관련된 내부특성 중 완전성에 대해서 품질을 측정하고 평가한 내용이다.

<표 9> 평가 점검표

내부특성 : 01 완전성(completeness)		해당□에 √표시를 한다					
		우수	양호	보통	미약	불량	해당 없음
No.	평가 항목	10	8	6	4	2	
01	· 사용자의 모든 요구사항이 충분히 분석되었는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
02	· 사용자의 요구와 관련된 기존 시스템에 대한 충분한 조사가 이루어졌는가?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
03	· 시스템의 기능이 충분히 설명되었는가?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
04	· 시스템 외부와의 연관성이 완전히 정의되었는가?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
소 계		10	16	6			
총 점		32					

평가 내용을 살펴보면 4개 평가 항목 중 보통(6점) 1개, 양호(8점) 2개 및 우수(10점) 1개 항목으로써 완전성 값의 백분율은 식 (2.1)에 따라 다음과 같이 산출하면

$$A_{\text{완전성}} = \{(6+8+10+8)/10 \times 4\} \times 100 = 80.0 \% \text{ 이다.}$$

위와 같은 요령으로 4종 업무에 대하여 내부특성 값을 정리하면 <표 10>과 같다.

<표 10> 분석단계 업무별 평가 결과

코드	내부특성	교무	학적	장학	도서
01	완전성	80.0	84.2	83.2	85.3
02	추적가능성	83.2	85.5	85.5	71.3
03	일관성	78.3	80.5	76.5	70.0
04	자기기술성	73.5	76.6	70.8	72.5
05	무모순성	83.0	82.6	84.3	80.8
:	:	:	:	:	:
25	적시성	80.6	83.5	87.6	83.4
26	적량성(기억성)	86.7	85.4	83.7	86.4
28	선택성	85.6	85.7	75.8	76.8
30	안전성	85.7	85.1	73.6	75.4
35	확장성	75.7	76.8	73.8	70.0

3.3 품질부특성, 주특성 및 최종 소프트웨어 값 산출

품질부특성 값은 식 (2.2)에 따라 <표 10>의 평가 결과 값을 이용하여 산출할 수 있으며 예로 학적업무의 품질부특성중 성숙성 값을 계산해 보면,

$$B_{\text{성숙성}} = (84.2 \times 0.2) + (85.5 \times 0.2) + (80.5 \times 0.2) + (76.6 \times 0.2) + (77.5 \times 0.2) = 80.86 \text{ 이다.}$$

여기서 가중치 부여는 완전성, 추적가능성, 일관성, 자기기술성 및 모듈성 5개 내부특성 모두 동일하게 강한관계에 있으므로 0.2를 부여하였다. 같은 요령으로 장애허용성 76.96 및 회복성

77.76 값을 얻을 수 있으며, 이를 정리하면 <표 11>과 같다.

<표 11> 분석단계 학적업무 품질부특성 값

품질부특성	코드	내부특성	평가치	가중치	결과치	산출과정
성숙성	01	완전성	84.2	0.2	80.86	84.2*0.2+85.5*0.2+80.5*0.2+76.6*0.2+77.5*0.2
	02	추적가능성	85.5	0.2		
	03	일관성	80.5	0.2		
	04	자기기술성	76.6	0.2		
	14	모듈성	77.5	0.2		
장애허용성	01	완전성	84.2	0.1	76.96	84.2*0.1+85.5*0.1+80.5*0.2+73.8*0.3+72.5*0.3
	02	추적가능성	85.5	0.1		
	03	일관성	80.5	0.2		
	12	견고성	73.8	0.3		
	13	무결성	72.5	0.3		
회복성	01	완전성	84.2	0.08	77.76	84.2*0.08+85.5*0.08+80.5*0.15+76.6*0.15+74.6*0.15+75.9*0.12+72.5*0.12+77.5*0.15
	02	추적가능성	85.5	0.08		
	03	일관성	80.5	0.15		
	04	자기기술성	76.6	0.15		
	11	엑세스감사성	74.6	0.15		
	12	견고성	73.8	0.12		
	13	무결성	72.5	0.12		
	14	모듈성	77.5	0.15		

여기서 가중치 부여는 ISO/IEC 9126에서 제시하고 있는 내부특성과 외부특성인 품질부특성, 품질주특성과의 관련정도 <표 5>에 따라 상호 비교하여 적용했음을 밝혀 둔다[2]. 보다 더 합리적이고 객관적인 품질평가를 위해서는 적용되는 시스템별, 품질특성별 가중치 부여가 달라져야 한다고 본다. 이를 위해서는 품질은 사용자(고객)로부터 나오기 때문에 사용자가 만족할 수 있는 가중치 부여 방법에 대해서 추가 연구를 계속할 예정이다.

품질주특성 값은 식 (2.3)에 따라 산출할 수 있으며, 예로 학적업무 신뢰성 값은

$$C_{(신뢰성)} = (80.86 + 76.96 + 77.76) / 3 = 78.53 \text{ 이며,}$$

같은 방법으로 기능성 80.38, 사용성 82.19, 보수성 78.45 및 이식성 79.64 이다. 최종적인 소프트웨어 결과 값은 식 (2.4)에 따라 기능성을 포함한 5개의 품질주특성 평균값으로 산출하며 예로 학적업무 결과 값은

$$D_{(학적)} = (80.38 + 78.53 + 82.19 + 78.45 + 79.64) / 5 = 79.84 \text{ 가 된다.}$$

<표 12> 업무별 품질주특성 결과 값

품질주특성	교무	학적	장학	도서
기능성	77.25	80.38	79.14	77.03
신뢰성	75.21	78.53	74.59	73.48
사용성	80.02	82.19	77.13	77.57
보수성	74.42	78.45	73.97	73.11
이식성	76.29	79.64	80.24	79.90
평균	76.64	79.84	77.01	76.22

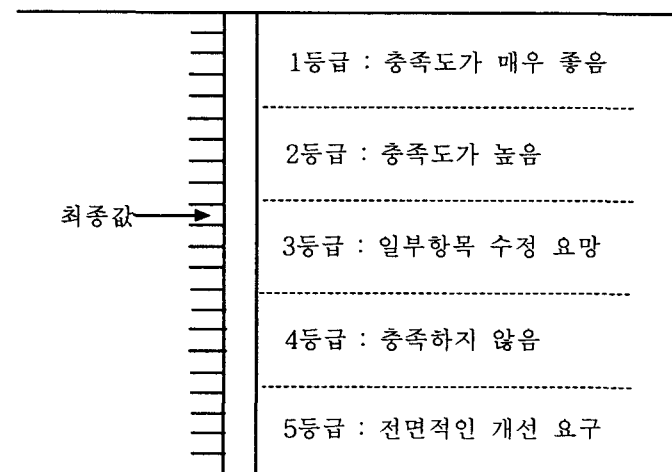
같은 요령으로 업무별 품질주특성 값과 소프트웨어 최종 값(평균)을 집계하면 <표 12>과 같이 정리된다.

여기서 효율성 값이 없는 것은 <표 5>에서 보듯이 설계단계에서는 효율성과 내부특성간에는 관련성을 갖고 있지 않기 때문이다.

3.4 평가결과 분석 및 프로젝트 관리방법

최종 소프트웨어 평가결과 값을 <표 12>에서 보면 교무업무 76.64, 학적 79.84 장학 77.01 및 도서 76.22 점으로 <표 8>의 평가 결과에 따른 소프트웨어 프로젝트 관리방법에 의하면 4종 업무 모두 3등급으로 사용자 만족을 위해 품질개선(QI : Quality Improvement) 활동을 하지 않으면 안 된다[17]. 현재 S 대학의 학사업무 전산화 사용 실태를 보아도 사용자들이 많은 불편을 느끼고 있는 실정이다. 따라서 시스템 분석단계의 산출물을 중심으로 사용자들이 요구한 항목들이 충실히 반영되어 있는지를 하나하나 확인을 하면서 반영되지 못한 요소를 중심으로 유지보수를 조기에 실시해야 한다.

(그림 3)은 소프트웨어 최종 값을 <표 8>의 평가결과에 따른 소프트웨어 프로젝트 관리방법 이용하여 등급을 가시적으로 도표로 나타낸 것이다[19]. 이러한 가시적인 표현 방법으로는 품질관리 활동에서 많이 사용되는 QC 7가지 도구/신 QC 7가지 도구 및 방사형 도표(radar chart) 등이 있다. 이렇게 가시적으로 나타냄으로써 프로그램 개발자, 유지보수자 및 프로젝트 관리자들이 쉽게 품질 결함 원인을 빨리 발견할 수 있다. 이렇게 발견된 결함을 조기에 수정함으로 유지보수비를 절약할 수 있고, 또한 개발과정에서의 소프트웨어 품질을 확보하는데 도움이 된다.



(그림 3) 품질 측정치 및 결과 등급

4. 결론 및 향후 연구과제

본 연구를 통하여 소프트웨어 개발주기 상에서 분석단계에서의 산출물에 대한 소프트웨어 품질측정 및 평가기법을 제시하였다. 이를 위해서 국제표준화 규격인 ISO/IEC 9126에서 언급되고 있는 소프트웨어 품질특성을 중심으로 평가항목을 선정하고, 그에 따른 평가 점검표를 개발하여 내부특성 값을 도출하였다. 내부특성 값을 기초로 하여 외부특성 값인 품질부특성 값과 품질주특성 값을 산출해 내고 최종적으로 소프트웨어 전체를 평가하는 내용이다. 개발된 기법을 실 업무에 적용하여 나름대로 검증도 시도해 보았다.

본 연구의 주 내용을 요약하고 결론을 내리면 다음과 같다.

첫째, 소프트웨어 품질을 대상으로 품질평가를 정량적으로 측정하고 평가를 시도해 보았다. 수치로서 나타낼 수 있어야만 관리할 수 있다는 산업공학적인 관점에서 연구의 특성을 찾자 했다.

둘째, 소프트웨어 프로젝트 관리방법을 제시해 보았다. 소프트웨어 개발시 개발자(프로그래머) 위주 관리에서 탈피하여 소프트웨어 평가자에 의한 관리 기초를 마련하고자 했다.

셋째, 적용 사례를 통하여 S대학의 학사업무 프로그램은 사용자(고객) 만족을 위해서 유지보수가 필요하다고 판단된다. 교무, 학적, 장학 및 도서업무 4종 모두 3등급 이하로 사용자 충족도에 불충분한 사항으로 품질개선이 요구된다고 볼 수 있다.

또한 본 연구의 내용이 응용 소프트웨어 중심과 적용대상이 대학의 학사업무로 한정했기 때문에 이를 일반화하는 데는 다소 무리가 있을 것으로 생각되나 본 논문에서 제시한 분야에 대해서는 적용가능하며, 다른 분야의 소프트웨어 품질에 대해서는 본 평가 모델의 기법은 적용 가능할 것으로 판단되나, 평가항목 및 가중치 적용은 달라져야 한다고 본다. 추가 연구가 계속 이루어지면 여러 분야에 적용 가능한 평가모델이 탄생할 것이다. 따라서 향후 연구 과제로는 소프트웨어 적용 분야별 사용자(고객)가 만족할 수 있는 품질특성별 가중치 부여방법과 이러한 품질측정과 평가를 할 수 있는 자동화 도구 개발이 추가적으로 연구되어야 할 것으로 본다.

참고문헌

- [1] 시스템공학연구소, “소프트웨어 품질평가 기술개발(I),” 정보통신부, 1997.
- [2] 장영숙, 권영식, “분석단계에서의 소프트웨어 품질 평가에 관한 연구,” 한국공업경영학회 1999년도 춘계학술대회 발표 논문집, 삼척대학교, 강원, pp. 293-300, 1999.
- [3] 한국전산원, “정보시스템 품질관리 감리지침에 관한 연구,” 1998.
- [4] 한국표준협회, 기업체질 강화를 위한 TQC 추진 체크리스트, 1994.
- [5] 菅野 孝男, 實務者のための ソフトウェア 外注管理, コンピュータ エージ社, 1996.
- [6] 森口繁一; ソフトウェア 品質管理 ガイドブック, 日本規格協會, 1990.
- [7] Boehm, B.W., Software Engineering Economics, Prentice Hall, 1981.
- [8] Crosby P.B., Quality is Free, A Mentor Book, 1979.
- [9] Evans, M. & J. Marciniak, Software Quality Assurance and Management, John Wiley & Sons, 1987.
- [10] Feigenbaum A.V., Total Quality Control, 3rd ed., McGraw-Hill, 1991.