

프로세스 파라미터를 이용한 소프트웨어 품질평가 시스템의 설계 및 구현

이 준 길[†] · 권 기 태^{††}

요 약

오늘날 소프트웨어의 복잡성과 대형화로 유지보수의 비용이 증대하고 있다. 이런 이유로 소프트웨어의 품질을 향상시키는 것이 중요한 이슈가 되고 있다. 이에 따라 소프트웨어 공학에서는 소프트웨어 품질을 평가하는 방법으로 프로세스의 최종 산출물인 소프트웨어의 소스를 분석하는 방법과 소프트웨어의 생명주기에 따라 그 단계별 산출물을 평가하는 연구가 계속되고 있고, 또한 소프트웨어를 생산하는 전 과정을 일련의 프로세스로 보고 각각의 프로세스를 평가하여 소프트웨어 품질을 향상시키는 연구도 진행되고 있다. 본 연구에서는 SPICE 모델에 따라서 소프트웨어를 지원, 생산, 공급하는데 관련이 있는 프로세스 범주를 5가지로 분류하고 각 범주에 속한 기본활동의 파라미터인 작업산출물과 ISO/IEC9126 모델의 소프트웨어 품질특성과의 연관성을 찾아 소프트웨어 품질을 평가하고자 한다. 즉 소프트웨어를 생산하는 프로세스에서 투입 및 산출되는 인출력들을 평가하여 프로세스 수행과정을 개선 및 변경하는 것이 소프트웨어 품질에 어떤 영향을 미치는지를 보여주는 소프트웨어 제품의 품질을 평가하는 시스템을 설계 및 구현한다.

Design and Implementation of Software Quality Evaluation System Using Process Parameters

Joon-Kil Lee[†] · Ki-Tae Kwon^{††}

ABSTRACT

The increasing complexity and mass production of software make the expenses of maintenance higher in these days. That's why it becomes the issue to improve the quality of software. Along with these trends, studies are under way of enhancing software by evaluating the production step-by-step, depending on the life cycle of software. In addition, a research is also being undertaken that considers the complete phase for software production as a kind of process and advances each process to improve the quality of software. Our primary research interest is designing and implementing the system for evaluating the quality of software product with process parameters, which will indicate influences of changing or enhancing process performance through assessing input and output in the process of software production for the quality of software.

1. 서 론

오늘날 소프트웨어의 대형화, 복잡화에 따른 유지보

수비의 증가로 고품질의 소프트웨어가 요구되고 이에 따라 소프트웨어의 품질을 측정하고, 품질을 향상시키는 연구가 중요한 문제로 대두되었다. 이는 소프트웨어의 라이프사이클에서 유지보수가 차지하는 비용이 전체비용의 80%를 육박하게 됨으로써 소프트웨어의 품질을 향상시키는 것이 소프트웨어 공급자, 획득자,

※ 이 논문은 2000년도 두뇌한국21사업에 의하여 지원되었음

† 정 외 원, 김릉대학교 전기계산소 근무

†† 홍산의원, 김릉대학교 컴퓨터과학과 교수

논문접수 : 1999년 12월 21일, 심사완료 : 2000년 10월 12일

평가자 모두에게 중요한 문제로 부각되었다[1-5].

일반적으로 품질(Quality)이란 ISO8402에 의하면 명시되거나 암시된 욕구를 충족시킬 수 있는 재화나 용역의 총체적인 특징이나 특성이라고 정의하고 있고, 소프트웨어의 품질이란 주어진 욕구를 만족시킬 수 있는 능력을 가지고 있는 소프트웨어 제품의 특징과 특성의 총체라고 정의할 수 있다. 이런 소프트웨어 품질은 소프트웨어 제품의 여러 속성의 정도를 나타내는 것으로써, 사용자의 이용편의는 물론, 사용자 요구의 만족 정도를 표현할 수 있으며, 운영상의 결함이나 오류가 있음을 나타내는 척도이다[1, 6].

소프트웨어 품질평가는 소프트웨어 개발종료 후, 개발과정에서 생성된 생산물과 최종산출물인 소프트웨어를 평가함으로써 생명주기 전 과정에서 발생한 오류를 검출할 수 있고, 이를 바탕으로 소프트웨어의 신뢰성을 측정하는 평가방법이다[7].

소프트웨어 척도는 크게 3가지 범주로 구분할 수 있는데 제품 척도, 프로세스 척도, 프로젝트 척도이다. 제품 척도는 제품의 크기, 복잡도, 설계특징, 성능, 품질수준과 같은 특성에 관한 것이고 프로세스 척도는 소프트웨어의 개발 및 관리 프로세스를 개선하는데 사용되는 것이다. 프로젝트 척도는 프로젝트 특성 및 실행을 기술하는 것이다. 소프트웨어 개발 요원수, 소프트웨어 생명주기, 비용, 일정, 인력배치 등이 프로젝트 척도의 예가 될 수 있다. 소프트웨어 척도는 프로젝트 척도보다는 프로세스 및 제품 척도와 밀접한 관련이 있지만, 개발인원의 수, 기술적인 능력, 일정, 규모, 조직구조와 같은 프로젝트 파라미터도 분명히 소프트웨어 제품의 품질에 영향을 미친다[8].

이 연구에서는 프로세스 모델인 SPICE 모델과 소프트웨어 품질 모델을 나타내는 ISO/IEC9126시리즈를 중심으로 프로세스 파라미터들의 변경 또는 개선사항이 소프트웨어 품질에 어떠한 영향을 미치는가를 알아보고 이를 근거로 소프트웨어 품질평가를 하고자 한다.

이 연구의 구성 및 전개는 제1장 서론에 이어서, 제2장 관련 연구에서는 프로세스 모델과 소프트웨어 품질평가 모델의 연구를 살펴보고, 제3장 프로세스 파라미터와 소프트웨어 품질특성간의 관계를 보여주는 제안된 모델에 대한 연구를 알아보고, 제4장 소프트웨어 품질평가 시스템에서는 제안된 모델의 내부구성 및 구현을 설명하겠다. 그리고 결론으로 제5장 결론 및 향후 연구과제의 순서로 구성되어 있다.

2. 관련 연구

2.1 기존의 프로세스 관리 모델

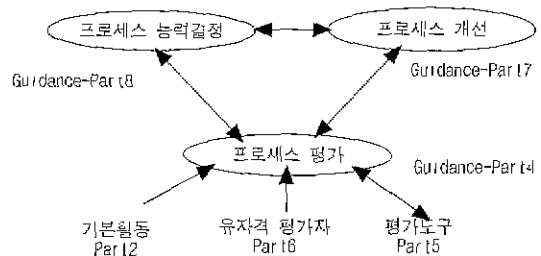
2.1.1 SPICE 모델

(1) 개념

SPICE(Software Process Improvement Capability dEtermination)는 소프트웨어 프로세스 평가에 대한 프레임워크를 제공한다. 이는 계획, 관리, 감독, 통제와 관련된 조직과 소프트웨어를 획득, 공급, 운영, 진화 및 지원을 개선하는 조직이 프로세스 개선을 위해 자체적인 프로세스의 상태를 이해하고, 특정한 요구사항에 지체의 프로세스가 적합한가를 결정하며, 특정한 계약에 따라 또 다른 조직의 프로세스의 적합성을 결정하는 목적으로 사용한다. SPICE는 <표 1>과 같이 구성된 9가지의 제품 또는 구성요소로 구성되어 있고, SPICE의 구성요소와 프로세스 평가, 개선, 능력결정간의 관계를 보면 (그림 1)과 같다[8, 9].

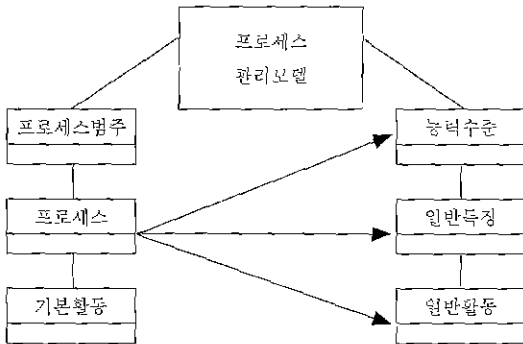
<표 1> SPICE의 구성요소 내용

구성요소	내 용
Part1	건체적인 구성을 기술하고 있으며 이의 선택 및 사용에 대한 지침을 제공
Part2	사경도구의 구축, 선정 및 사용자지침을 설명
Part3	평가 수행에 필요한 프레임워크 프로세스 능력의 등급, 점수기준을 정의
Part4	뒤 수준의 소프트웨어 프로세스 평가의 수행에 대한 지침을 제공
Part5	평가수행을 지원하는 도구를 만드는데 필요한 프레임워크 요소들 정의
Part6	프로세스 평가 수행에 필요한 평가자의 교육경리, 훈련 및 경험에 대하여 기술
Part7	프로세스 개선을 목적으로 한 지침을 기술
Part8	프로세스 능력결정을 목적으로 한 지침을 기술
Part9	용어 정의



(그림 1) SPICE의 구성요소간의 관계

이 평가 모델은 (그림 2)와 같이 프로세스 중심의 기본활동과 일반활동으로 구성된 2차원 모델이다. 일반활동은 일반특징과 능력수준으로 구성되어 있고, 평가결과는 평가대상 프로세스에 대한 프로세스 능력수준 등급을 포함한다[8, 10].



(그림 2) SPICE에서 프로세스와 능력수준의 2차원 구조

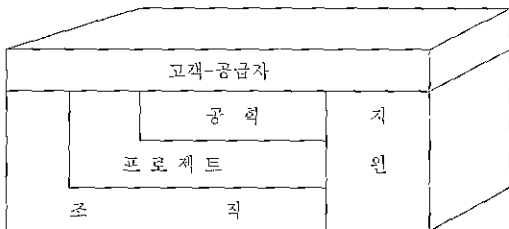
(2) 프로세스 관리 모델

① 개요

이 모델은 (그림 2)에서 보는 것과 같이 일반활동과 기본활동이라는 2차원 구조를 가지고 있다. 일반활동은 어떠한 프로세스에도 적용할 수 있는 프로세스를 관리하고 프로세스 능력을 향상시키기 위해 필요한 활동이고, 기본활동은 특정한 프로세스의 필수 활동이다 [10, 11].

② SPICE모델의 구성요소

SPICE 모델은 고객-공급자 프로세스, 공학 프로세스, 프로젝트 프로세스, 지원 프로세스, 조직 프로세스라는 5가지 범주로 구분된다. 프로세스 범주간의 관계는 (그림 3)과 같고, 그 범주의 내용은 <표 2>와 같다 [8, 9, 11, 12]



(그림 3) 프로세스 범주간의 관계

<표 2> 프로세스 범주의 내용

프로세스범주	내용
고객-공급자	고객에게 직접적인 영향을 미치고, 운영 및 사용을 지원하는 프로세스
공학	시스템 및 소프트웨어 제품의 구현, 유지보수 등과 관련된 프로세스
프로젝트	프로젝트의 구성, 지원관리, 상호협조에 관련된 프로세스
지원	프로젝트의 다른 프로세스의 수행을 가능하게 하고 지원하는 프로세스
조직	사업목적을 설정하고, 목적달성을 위해 자원을 개발하는 것과 관련된 프로세스

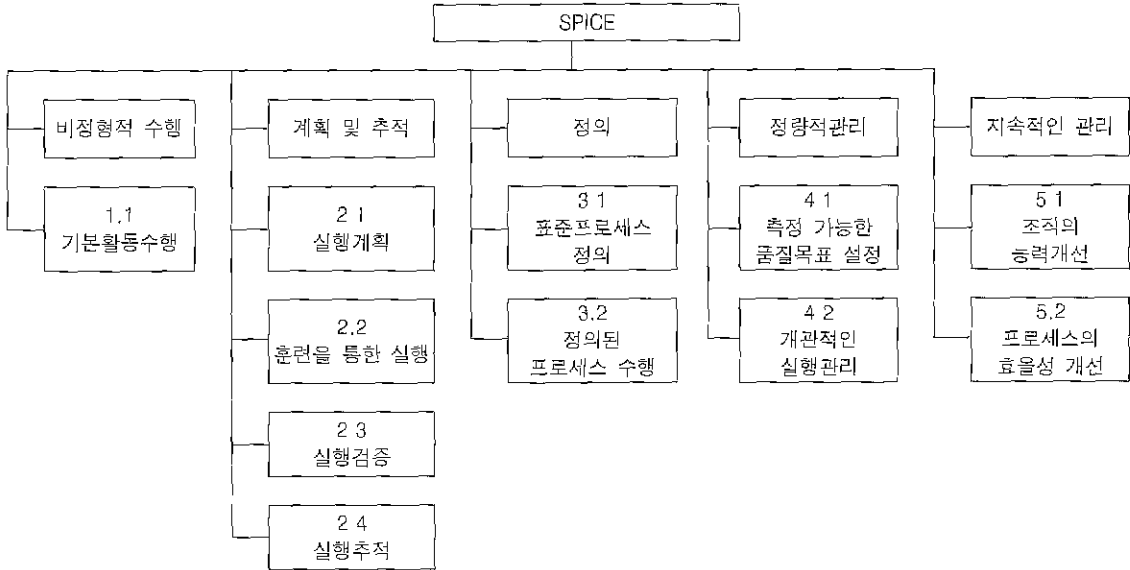
SPICE의 각 프로세스는 각각의 목적을 달성하기 위해 필요한 기본활동으로 구성되어 있다. 이 기본활동은 특정 프로세스를 수행하기 위한 소프트웨어 공학적인 또는 관리적인 활동이다.

SPICE 모델을 구현 또는 체계화하는 유형별로 분류하면 능력수준, 공통특징, 일반활동으로 구분할 수 있다. 이들은 각각 다른 추상화 수준에서 프로세스 수행 능력을 개선하거나 프로세스를 관리하는데 필요한 활동을 다루고 있다. 능력수준은 공통특징의 집합체이다. SPICE 모델은 프로세스 능력수준을 6단계로 구분한다 [12].

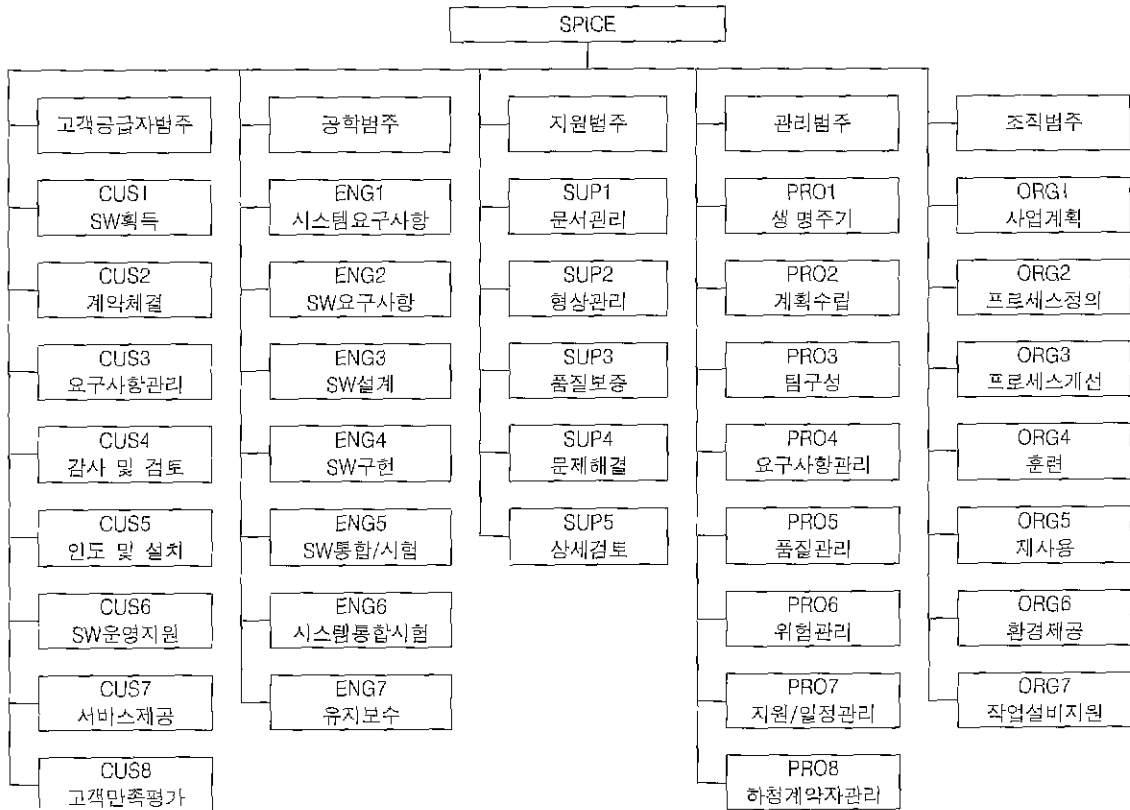
③ 기본활동 및 일반활동

평가를 수행하는 동안 능력수준과 공통특징에 의해 분류되는 일반활동은 프로세스 능력결정에 사용된다. (그림 4)에 정의한 모든 일반활동은 SPICE 모델내의 모든 프로세스 적용될 수 있는 것이다. SPICE의 각 범주별 프로세스 및 기본활동은 (그림 5)와 같다[11, 13].

그의 관련 연구 모델로서는 정보통신 제품의 개발 및 지원 능력을 평가하는 모델로서 Bell Canada, Northern Telecom, Bell-Northern Research의 공동 프로젝트 결과물인 Trillium, 미국 Carnegie Mellon대학 소프트웨어 공학연구소(SEI: Software Engineering Institute)의 성숙도 프레임워크를 적용하여 현재 소프트웨어 산업계에서 사용하는 모델의 근간으로 개발되고 있는 CMM (Capability Maturity Model), 유럽소프트웨어 업계에 널리 적용되고 있는 BOOTSTRAP과 ESPRIT, ISO 9000 시리즈, Software Technology Diagnostic, AMI, Healthcheck 등 많은 모델들이 있다[8, 14-16]



(그림 4) 수준별 일반활동



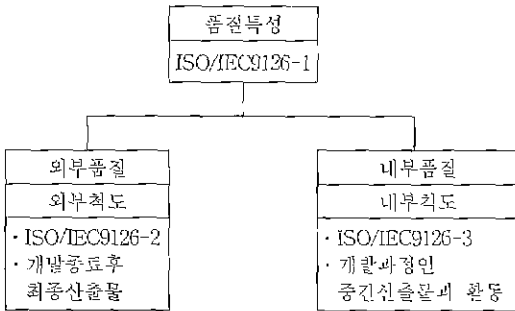
(그림 5) 범주별 프로세스 및 기본활동

2.2 기존의 소프트웨어 품질평가 모델

소프트웨어 품질을 평가하기 위해서는 우선적으로 소프트웨어 제품에 요구되는 품질특성과 평가기준에 대한 명확한 정의가 필요하다. 소프트웨어의 품질은 하드웨어와 달리 보는 사람마다 다르게 해석하는 경우가 있기 때문이다. 이와 같이 소프트웨어 품질은 여러 관점에서 파악되므로 평가를 하기 위해서는 요구되는 품질특성을 명확히 정의하고, 이를 평가하기 위한 기준을 사전에 설정해야 한다. 이러한 평가기준을 제시한 여러 연구들을 고찰하기로 한다. 소프트웨어 품질은 특성에 따라 여러 요인으로 나누어서 생각할 수 있다[8, 17].

2.2.1 ISO/IEC9126의 품질 모델

소프트웨어의 품질특성은 컴퓨터 프로그램 및 펌웨어 자료를 포함하여 모든 종류의 소프트웨어에 적용할 수 있는 것으로서 ISO/IEC9126에 기초를 두었다. ISO/IEC9126은 소프트웨어의 품질특성을 기능성, 신뢰성, 사용성, 효율성, 유지보수가능성, 이식성의 6가지로 구분하고 있다. 품질특성, 부특성 등에 따른 ISO/IEC9126



(그림 6) ISO/IEC9126 소프트웨어 품질특성과 척도

품질 모델은 <표 3>과 같다. 소프트웨어의 품질특성과 그 하위특성의 정의를 고찰해 보면 다음과 같다[1, 5, 9, 18].

(1) ISO/IEC9126의 소프트웨어 품질특성 및 부특성 (하위특성)

① 기능성

- 적절성: 규정된 업무를 수행하기 위한 일련의 기능준제 및 적합성과 관련된 특성
- 정확성: 올바른 또는 합의된 결과나 효과의 조향과 관련된 특성
- 상호운용성: 규정된 시스템과 상호작용하는 능력과 관련된 속성
- 준수성: 소프트웨어가 법에서 요구하는 표준이나 권래, 또는 규정에 따르는가에 대한 속성
- 보안성: 고의적인 우연이든 간에 프로그램과 데이터를 허가 없이 접근하는 것을 방지하는 능력과 관련된 특성

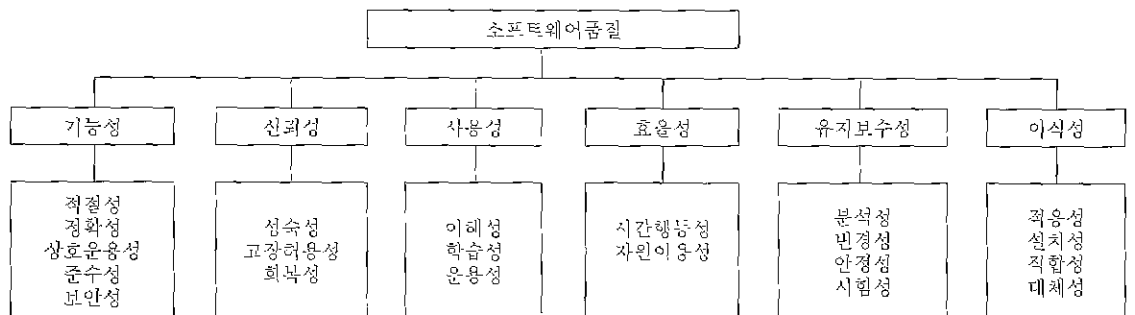
② 신뢰성

- 성능성: 소프트웨어 결함에 따른 고장빈도와 관련된 속성
- 고장허용: 소프트웨어 오류나 규정된 인터페이스 위반의 경우 규정된 성능수준을 유지하는 능력과 관련된 속성
- 회복성: 고장의 경우 원래 성능수준으로 회복하고 직접적으로 영향을 받는 데이터를 원래 상태로 되돌려 놓는 능력과 관련된 속성.

③ 사용성

- 이해성: 논리적 개념과 그의 적용을 이해하기 위한 사용자의 노력과 관련된 속성

<표 3> ISO/IEC9126 품질 모델



- 학습성 : 사용자가 배우는데 관련된 속성
- 운용성 : 운용, 운용재어를 위하여 필요한 사용자 노력과 관련된 속성

④ 효율성

- 시간행동 : 응답, 처리시간, 기능 수행에서 처리율과 관련된 속성
- 자원행동 : 기능 수행에서 사용된 자원의 양과 사용시간과 관련된 속성

⑤ 유지보수성

- 분석성 : 부적합하거나 고장 원인 분석이나 수정된 부분의 식별을 위하여 필요한 노력과 관련된 속성
- 변경성 : 수정이나 결함제거 또는 환경변화를 위하여 필요한 노력과 관련된 속성
- 안정성 : 수정에 따른 예기치 않은 결과로부터 위험과 관련된 속성
- 시험성 : 수정된 소프트웨어를 확인하기 위하여 필요한 노력과 관련된 속성

⑥ 이식성

- 적응성 : 규정된 환경과 다른 환경에 적응 가능성과 관련된 속성
- 설치성 : 규정된 환경에 소프트웨어를 설치하기에 필요한 노력과 관련된 속성
- 적합성 : 소프트웨어 이식성과 관련된 표준이나 관례에 맞게 하는 속성
- 대체성 : 다른 특정 소프트웨어를 대체 사용할 수 있는 가능성과 노력과 관련된 속성[8,19].

그의 관련 연구 모델로서는 품질을 평가하기 위해서 소프트웨어의 품질요인을 제품정의에 기초를 두고 제품운영, 제품변경, 제품개정의 3가지로 구분하는 McCall의 품질 모델, 품질개념을 사용자측면의 용어를 사용하여 설명한 Boehm의 품질 모델, 11가지의 품질 목표를 설정하여 소프트웨어의 생명과 관계되는 목표, 소프트웨어의 수명을 연장시키는 목표, 응용분야 및 다른 시스템과의 연결성 그리고 사용자들 위한 사용성에 품질목표를 둔 Perry의 품질 모델, 응용성, 유지보수성, 적응성 3가지 범주화로 구분하는 Spardat 모델, 조직이 조직의 특성에 적합한 척도를 이용해 방법론을 사용하는 IEEE1061 소프트웨어 품질척도 방법론 등이 있다[20, 21, 22].

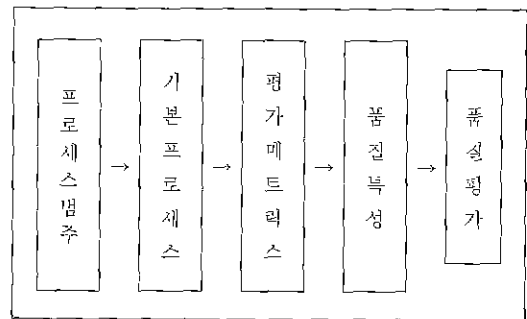
3. 제안된 모델

3.1 시스템의 기본구성

소프트웨어 품질평가를 위해서는 소프트웨어 품질요인을 분석할 수 있는 분석 시스템이 요구된다 따라서 본 연구에서는 소프트웨어 품질에 영향을 미치는 요인에 대한 정량적인 품질관리를 위하여 프로세스의 작업 산출물을 이용하여 품질요인들을 측정, 평가하고 분류 및 저장하는 시스템을 구현하고자 한다.

이 논문에서 제안하는 모델의 기본구성은 (그림 7)과 같다. 이 연구의 목적은 프로세스의 기본활동과 프로세스의 최종산출물인 소프트웨어 품질과의 연관성을 찾아서 프로세스의 기본활동을 개선하여 품질 높은 소프트웨어를 생산하고자 하는데 목적이 있다. 이 모델의 연구방법은 SPICE 모델의 프로세스 5가지 범주를 기준으로 프로세스 기본활동들의 작업산출물들을 평가하고, 이 작업산출물들과 ISO/IEC9126 모델의 소프트웨어 품질특성간의 연관성에 따라 중간산출물에 대한 소프트웨어 품질특성을 평가하고자 한다.

이 모델의 기본구성에 따른 기능적 역할은 프로세스 범주에서 고객-공급자, 공학, 지원, 관리, 조직 등 5가지로 분류하고, 각 범주를 역할별로 기본프로세스로 분류한다 또 프로세스 활동을 수행할 경우 입력된 작업산출물과 수행 시 생산되는 작업산출물을 경리한다 이 경리된 작업산출물을 평가 매트릭스를 이용하여 평가하고 수치화하여 소프트웨어 품질특성의 품질을 결정한다.



(그림 7) 제안된 모델의 기본구성

3.1.1 매트릭스 평가 기준값

품질 평가매트릭스는 기본활동들을 구성하고 있는 작업산출물특성의 정확성 정도에 따라 평가단계를 <표

4>같이 분류하고, 이에 따른 응답수준별 평가단계를 점수화하였다[23]

<표 4> 정확성 정도에 따른 평가단계 및 평가점수

응답 수준	적합성정도	점수
매우 적합	F	3
어느 정도 적합	L	2
약간 적합	P	1
적합치 않음	N	0

3.1.2 품질평가결과의 평가수준

품질평가가 끝나면 평가결과에 대한 의미를 부여하기 위하여 측정된 평점결과로 ISO/IEC9126에서 설정한 5단계 등급수준을 사용하여 소프트웨어품질의 평가수준을 결정한다. 본 논문에서는 평가도구를 이용하여 측정된 결과에 대해 <표 5>와 같이 판정하기로 하였다[5, 24].

<표 5> 산출물 평가결과의 평정수준

점수	등급 수준
95점이상	매우 우수
85점이상~95점미만	우수
75점이상~85점미만	양호
65점이상~75점미만	미약
65점미만	불량

3.2 프로세스 평가절차

3.2.1 작업산출물의 평가방법

작업산출물은 한 조직의 활동들의 실행결과로서 한

조직의 프로세스나 활동을 실현하는데 있어서 입력물들로 이용된다. 작업산출물의 평가는 작업산출물특성들을 각각의 평가정도에 따라 이루어지며, 이 작업산출물특성들은 프로세스 또는 활동이 생산한 작업산출물을 평가하는데 도움을 주는 속성에 대한 지침을 제공한다.

작업산출물표에서 작업산출물 식별자(WP id#)는 프로세스 관리테이블에서 작업산출물을 참조하기 위하여 사용된다. 작업산출물은 작업산출물의 특성에 관련된 전형적인 이름으로 부여하며, 활동이나 프로세스를 산출하는 작업산출물로서의 식별자이다. 각각의 작업산출물의 평가방법은 각각의 작업산출물특성에 따라 적합성 정도를 <표 4>와 같이 4가지로 분류하고, 이 평가수준별에 따라 0점부터 3점까지 점수를 부여하고, 작업산출물별로 점수화한다. <표 6>은 작업산출물의 평가가 각각의 작업산출물특성들의 내용으로 결정되어지는 것을 보이고 있다. 작업산출물특성은 평가자가 기본활동의 존재유무 평가 또는 정확성 평가를 결정할 때 이렇게 판단하는가 하는 기준을 제시한다[24].

3.2.2 프로세스 범주의 평가방법

<표 7>의 관계표의 목적은 평가자나 평가시스템이 작업산출물과 기본활동 내에 정의된 특성들과의 관련성을 이해하고 작업산출물의 평가에 따라서 기본활동들의 평가를 구하는데 있다. 하나의 기본활동의 점수는 그 기본활동내의 작업산출물들의 합의 평균값으로 구해진다. 작업산출물은 기본활동의 입력이나 출력에

<표 6> 작업산출물특성 평가표

WP id#	작업산출물	작업산출물 특성	평가레벨				평가치
1	소프트웨어 개발 방법론	소프트웨어를 개발하는데 사용하는 접근 / 방법의 식별 소프트웨어를 개발하는데 사용되는 생명주기모델의 식별 프로세스, 관용, 그리고 통제의 높은 수준 기술을 제공	N	P	L	F	F
2	소프트웨어개발 생명주기 모델	각각의 생명 주기 단계에서 수행된 활동의 높은 수준 기술 생명 주기 단계의 순서 중요한 생명 주기 단계 의존의 식별 각각의 생명 주기 단계에서 요구된 임준도의 식별 중요한 주 결성점의 식별 모델에 대한 성질 제어집의 식별	N	P	L	F	F
109	인사기록	이름, 주소, 배우자 유무, 출생일 학위, 급여, 수상 경력 교육훈련 경력	N	P	L	F	L

WP id# Work product indicator number

관련이 있고 작업산출물특성들과 연관성을 갖고 있다. 기본활동의 결과는 소프트웨어 품질특성과 관련하여 프로세스 5가지 범주 중 취약성을 갖는 범주 또는 개선이 요구되는 기본활동을 확인할 수 있다.

<표 7> 프로세스별 기본활동과 작업산출물의 관계

기본활동	작업산출물 ID	평가치
CUS.1.1	44 신출물 요구 평가	L
	52 요구사항 성숙	F
	88 내부적, 외부적 고객 요구사항	L
CUS.1.2	44. 신출물 요구 평가	L
	52. 요구사항 성숙	F
⋮		
ORG.7.4	17. 프로젝트 계획	L
	52 요구사항 성숙	F
	104. 개발 환경	L
ORG.7.5	17 프로젝트 계획	L
	52 요구사항 성숙	F
	104 개발 환경	L

3.2.3 품질특성의 평가방법

작업산출물과 소프트웨어 품질부특성과의 관계는 <표 8>과 같고, <표 7>에서는 프로세스의 기본활동과 작업산출물과의 관계를 보여주고 있다. 각 기본활동들과 품질특성들과의 관계에 따라 <표 7>에서 얻어진

값을 이용하여 품질특성에 따른 각각의 특성값을 구한다. 각 프로세스는 품질특성별로 기본활동들의 합을 구하고 최종적으로 프로젝트 구성요소별 5가지 범주에서 소프트웨어의 품질특성별 평가값에 따라서 프로젝트의 최종산출물인 소프트웨어 품질특성의 평가값을 구한다.

<표 8> 작업산출물과 소프트웨어 품질부특성과의 관계

WP id#	작업산출물	부특성
1	소프트웨어 개발 방법론	적절성
2	소프트웨어개발 생명주기 모델	직접성
3	프로세스 기술	상호운용성
4	작업절차, 활동	완수성
	⋮	
107	설치 기록	설치성
108	시스템 구성요소	자원행동
109	인사 기록	학습성

기본활동과 소프트웨어 품질특성과의 관계를 보여주는 <표 9>는 <표 8>에서 작업산출물과 부특성의 관계와 <표 7> 기본활동과 작업산출물의 관계에서 기본활동들 속에 나타난 부특성의 빈도로서 밀집한 정도를 나타내는 표이다. 이 표는 연구의 성격에 따라서 관계성의 정도에 의한 가중치를 부여하는 근거로 사용될 수 있으나 본 연구에서는 가중치 부여는 고려하지 않았다.

<표 9> 프로세스의 기본활동과 소프트웨어 품질특성과의 관계

품질특성		프로세스범주		기능성				신뢰성		사용성		효율성		유지보수성		이식성										
				적절성	경의성	상호운용성	준수성	보안성	성숙성	고장허용성	회복성	이해성	학습성	운용성	시간행동	자원행동	분석성	변경성	인정성	시행성	적응성	설치성	적합성	대체성		
고객공급자	CUS.1	CUS1.1	◎									○														
		CUS1.2	◎																							
		CUS1.3		○																			○			
		CUS1.4		○																			○			
		CUS1.5	○			○		○								○							○			
조직	ORG.7	ORG7.1	◎				○																			
		ORG7.2	◎				○																			
		ORG7.3																								
		ORG7.4	◎				○																			
		ORG7.5	◎				○																			

● 관련성이 매우 클 ○ 관련성이 높긴 ○ 관련성이 적음

3.3 프로세스 평가결과

3.3.1 작업산출물의 평가결과

임의의 프로젝트에 대한 샘플 데이터들 <표 6>의 작업산출물특성 평가표에 의거하여 평가한 예를 보면 <표 10>와 같다. 이 결과는 각각의 작업산출물에 대한 평가수준을 <표 4>를 기준으로 점수화한 것이다.

<표 10> 작업산출물 평가표

WP id#	작업산출물 형태	평가치
1	소프트웨어 개발 방법론	3
2	소프트웨어개발 생명주기 모델	3
3	프로세스 기술	3
4		
	⋮	
108	시스템 구성요소	3
109	인사 기록	2

3.3.2 프로세스 범주의 평가결과

프로세스 범주에 대한 평가는 각각의 범주를 구성하고 있는 기본활동들을 평가함으로써 이루어진다. <표 11>에서 평가점수는 <표 10>에서의 작업산출물에서 얻어진 점수이고, 이들의 합이 기본활동과 프로세스 범주의 수행결과가 된다. <표 11>에서 환산점수는 <표 5>의 평정수준으로 평가하기 위해 기본활동에 대한 점수 3을 100점 기준으로 환산한 점수이다 다음 식 (1), 식 (2)는 기본활동 평균점수와 이에 대한 환산점수를 구하는 식이다.

$$\text{기본활동 평균점수} : V_oBP = \frac{S_oWPVal}{C_oWP} \quad (1)$$

<표 11> 프로세스별 기본활동 및 작업산출물과의 관계

기본활동	작업산출물 ID	평가점수	평균점수	환산점수
CUS.1.1	41. 산출물 요구 평가	2	(2+3+2)/3 = 2.3	76.7
	52. 요구사항 상술	3		
	83. 내부적, 외부적 고객 요구사항	2		
CUS.1.2	44. 산출물 요구 평가	2	(2+3)/2 = 2.5	83.3
	52. 요구사항 상술	3		
	⋮			
ORG.7.4	17. 프로젝트 계획	2	(3+3+2)/3 = 2.3	76.7
	52. 요구사항 상술	3		
	104. 개발 환경	2		
ORG.7.5	17. 프로젝트 계획	2	(2+3+2)/3 = 2.3	76.7
	52. 요구사항 상술	3		
	104. 개발 환경	2		

$$\text{평균점수에 대한 환산점수} : ConVal = \frac{V_oBP * 100}{BV_oBP} \quad (2)$$

여기서, V_oBP : 각 기본활동의 평가값

S_oWPVal : 해당 작업산출물의 평가값의 합

C_oWP : 해당 작업산출물의 수

$ConVal$: 환산점수

BV_oBP : 기본활동 기준값

기본활동에 있어서 평가할 작업산출물의 많고 적음의 문제로 평가점수의 절대값이 상당한 거차를 갖는 문제는 각각의 기본활동에 대한 평균값을 적용함으로써 각 작업산출물이 기본활동에 미치는 영향을 균일하게 하였다. 이와 같은 방법이 소프트웨어 품질부특성과의 관계에도 적용되었다.

3.3.3 품질특성의 평가결과

소프트웨어 품질특성의 평가는 <표 7>과 <표 8>의 관계에서 프로세스 기본활동별로 소프트웨어 품질특성을 평가할 수 있다. <표 12>은 기본활동별 예의 일부분을 보여주고, <표 13>에서는 프로세스 5가지 범주별로 평가된 소프트웨어 품질특성을 보여주고 있다. 또한, <표 13>에서 평가결과를 보면 효율성과 유지보수성의 품질특성이 다른 품질특성보다 상대적으로 취약한 결과를 보이고 있고, <표 5>의 평정수준으로 볼 때 전체적으로 “양호”와 “미약”정도의 평가결과를 보이고 있어 프로세스의 개선이 요구된다.

3.3.4 품질평가 결과의 표현방법

품질평가 지원을 위해 각 품질측정값을 평가한 결과

〈표 12〉 프로세스의 기본활동과 소프트웨어 품질특성과의 관계

품질특성		기능성					신뢰성			사용성			효율성		유지보수성			이식성					
		속설성	정확성	상호운용성	준수성	보안성	성숙성	고장허용성	회복성	이해성	학습성	운용성	시간행동	자원행동	분석성	변경성	인식성	시험성	적응성	설치성	지향성	대체성	
고객공급자	CUS1	CUS1.1	25									2											
		CUS1.2	25																				
		CUS1.3		3																	2		
		CUS1.4	3	3																	2		
		CUS1.5	2			2									3						2		
	CUS2	CUS2.1				3																	
		⋮																					
		CUS2.4					3								3						2		
⋮																							
조직	ORG7	ORG7.1	25			2																	
		ORG7.2	25			2				2													
		ORG7.3									15												
		ORG7.4	25			2																	
		ORG7.5	25			2																	

〈표 13〉 소프트웨어 품질특성의 평가값

품질특성		기능성					신뢰성			사용성			효율성		유지보수성			이식성				
		적절성	정확성	상호운용성	준수성	보안성	성숙성	고장허용성	회복성	이해성	학습성	운용성	시간행동	자원행동	분석성	변경성	안정성	시험성	적응성	설치성	지향성	대체성
고객공급자		2.83	2.5	2.2	2.82		2			2.5	3	2.33	1.83	1.5	2.23	1	3	2	2	1.89	2	3
공회		2.89		2.88		2	2.29		2	2.33		2		2.83	1.89	2.83	3	2	2.45	2	3	3
프로젝트		2.72	2.37	2.18	2.88	2	2	3		2	3	2.1	2.38	1.95	1.8	2	2		2	2	3	
시원		2.72	2.29	2.67			2	3		2		2.25	3	3	2.75	2.17	3	2	2	1.25	3	3
조직		2.69	2.21	2.67	2.43	2	2	3	1.67	2	3	2.5	2	1.71	2	1.46	2		2	1	3	3
평균 (환산점수)		2.74	2.31	2.55	2.77	2	2.03	3	1.8	2.09	3	2.26	2.21	2.03	2.07	1.93	2.19	2	2.22	1.78	2.94	3
		2.45(81.67)					2.2(73.33)			2.38(79.33)			2.22(74)		2.11(70.33)			2.32(77.33)				

를 사용자가 쉽게 파악할 수 있도록 다음 그래프를 이용하여 가시화 하였다[17, 19].

(1) 기본활동별 비교 그래프

5가지 프로세서 범주를 구성하고 있는 기본활동간의 득점을 시각적으로 비교해 볼 수 있도록 구성한 그래프

(2) 범주별 비교 그래프

품질평가를 실시하는 프로젝트를 구성하는 범주별간에 득점수준을 비교할 수 있도록 차트나 방사형으로 표현한 그래프

(3) 내부품질부특성간의 득점별 비교 그래프

21개 내부특성에 대해 시각적으로 비교해 볼 수 있도록 품질부특성들간에 득점수준을 그래프화 한 것으로 하나의 품질특성에 대한 관련 품질부특성 21항목 전체에 대한 비교를 위한 막대 그래프

(4) 품질특성별 비교 그래프

품질특성간 6항목에 대해 득점수준을 비교할 수 있도록 방사형 차트, 파이 차트 등을 이용하여 구성한 그래프

(5) 품질평가 비교 그래프

반복된 평가결과에 대하여 품질 수준이 어떻게 변화하고 있는가를 점검하고 문제점을 파악 대처하도록 쉬운선 그래프를 이용해 구성한 비교 그래프

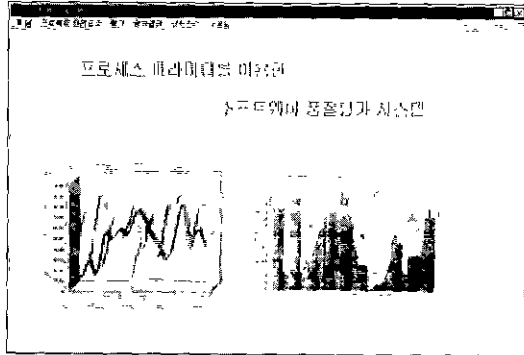
4. 소프트웨어 품질평가 시스템

4.1 소프트웨어 품질평가 시스템 구현환경

소프트웨어 품질평가 시스템의 구현환경은 서버는 SUN Ultra10 SunOs5.5.1시스템과 오리를 73버전의 데이터베이스 환경에, 클라이언트는 Windows98에 개발툴 Power Builder 5.0을 사용하여 구현하였다.

4.2 소프트웨어 품질평가 시스템 기능

(그림 8)은 구현한 소프트웨어 품질평가 시스템의 메인화면이다. 이 시스템에서 제공되는 기능과 각 기본 메뉴에 대한 설명은 다음과 같다



(그림 8) 소프트웨어 품질평가 시스템

4.2.1 제공되는 기능

- 사용자 인터페이스 기능 : GUI를 사용하여 메뉴방식 사용
- 품질평가 지침 제공 기능 : 작업단계별 사용자 지침 제공
- 품질평가요소의 입출력 : 관리요소, 측정값, 코드 입력
- 품질평가요소의 평가처리 평가처리
- 품질평가요소의 평가결과 확인 : 평가결과를 품질 특성별 출력, 그래픽 인터페이스 제공
- 품질평가도구의 도움말 기능 : 각 기능별 의미와 사용방법에 관한 설명 및 메뉴

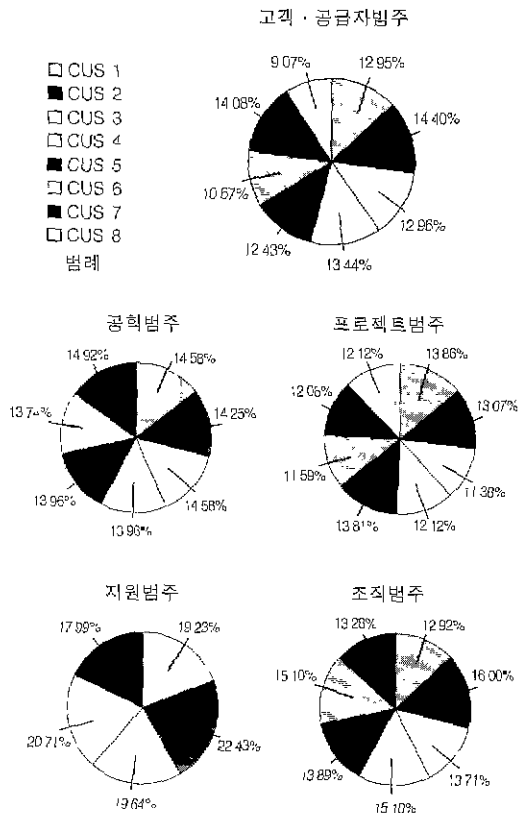
4.2.2 기본 메뉴에 대한 설명

구현된 소프트웨어 품질평가 시스템은 기본 메뉴와 기존 메뉴에 속한 부 메뉴들로 구성되어 있다. 그 내용을 살펴보면, 기본 메뉴는 개념, 프로젝트 관리요소, 평가, 평가결과, 코드관리, 도움말 등으로 이루어져 있고, 부 메뉴는 기본 메뉴에 따라 품질평가 기본항목 입력 및 평가, 평가결과를 가시화하고, 또한 이런 일련의 과정에 대한 개념 및 도움말로 이루어져 있다.

4.3 품질평가 결과의 가시화

4.3.1 기본활동별 비교 그래프

(그림 9)는 각각의 범주 그래프에서 한 범주 그래프 내의 각 기본활동들의 비율은 한 범주를 이루고 있는 구성 비율이다.

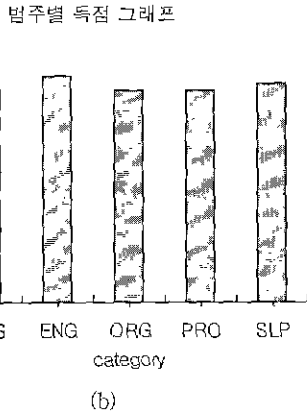
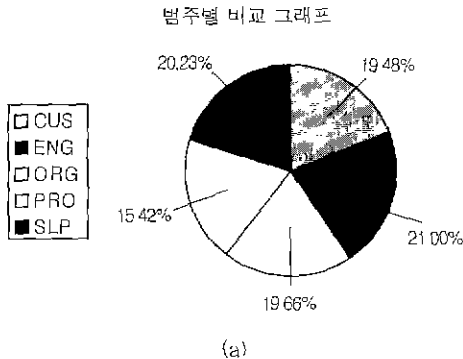


(그림 9) 범주별 기본활동별 비교 그래프

4.3.2 범주별 비교 그래프

(그림 10)은 그래프(a)에서는 상대적으로 높은 특징

수준을 보이고 있으나, 그래프(b)를 보면 각 범주별로 획득한 점수를 알 수 있다. <표 5>의 기준으로 볼 때 범주별 평가값은 우수하다고 볼 수 없으며, 특히 고객-공급자, 프로젝트, 조직 범주는 더욱 개선이 필요하다는 것을 알 수 있다. <표 14>은 범주별 평가값을 나타내고 있다.



(그림 10) 범주별 비교 그래프

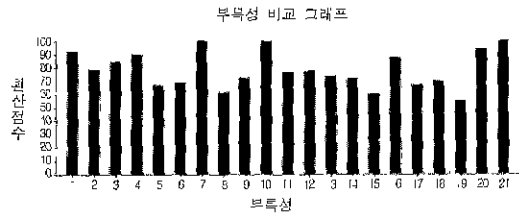
<표 14> 범주별 평가값(환산점수)

평가순	범주	고객-공급자	공학	프로젝트	지원	조직
I		78.13	84.24	78.75	81.13	78.86

4.3.3 내부품질부특성 간의 득점별 비교 그래프

(그림 11)은 소프트웨어 품질부특성에 대해 직접적으로 평가값이 나타나 취약한 성질에 대한 직관적인 파악과 <표 8>에서 관계된 작업산출물의 확인으로 미흡한 작업산출물의 특성들을 보완할 수 있다. 이 결과

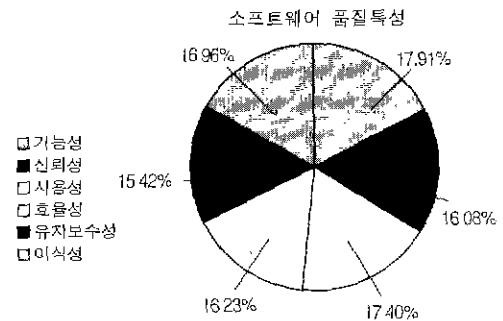
로 볼 때 회복성, 변경성, 설치성 등이 미흡하여 이들 부특성과 관계 있는 작업산출물을 개선하여 평가를 반복수행 한다.



(그림 11) 내부품질부특성 간의 비교 그래프

4.3.4 품질특성별 비교 그래프

(그림 12)는 소프트웨어 품질특성 6가지는 각 성질에 대한 특성별 만족도를 나타낸 것으로 서로간의 성질에 영향을 주지 않고 단지 상대적 비교가 가능하고, 이 그래프는 취약한 성질에 대한 파악이 가능하므로 품질향상에 노력할 품질특성부분을 결정하기 위해 이용이 가능하다. 이 그래프에서는 전체적으로 균형된 비율을 보이고 있는 것처럼 보이나, <표 13>에서 알 수 있는 것처럼 품질평가결과의 평가값은 만족할 만한 평가값이 아니다.



(그림 12) 품질특성별 비교 그래프

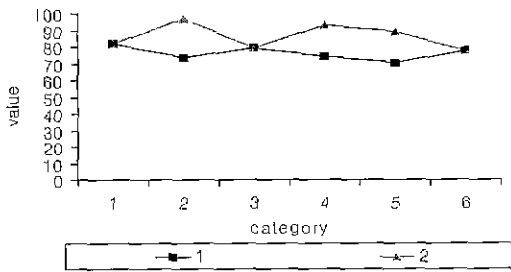
4.3.5 품질평가 비교 그래프

(그림 13)은 반복 수행한 평가결과의 비교로 향상된 품질평가를 확인할 수 있다 (그림 13)은 <표 15>에서 보는 것 같이 1차 평가에서 취약한 품질특성인 신뢰성, 효율성 및 유지보수성을 보완하여 2차 평가를 실시한 결과를 보여주고 있다. 이는 만족할 만한 평가수준을 얻을 때까지 3차, 4차 계속해서 평가수행을 거듭할 수 있다.

〈표 15〉 1,2차 소프트웨어 품질특성의 평가값

		(관선점수)					
품질특성 평가순	기능성	신뢰성	사용성	효율성	유지 보수성	이식성	
1차	81.67	73.33	79.33	74	70.33	77.33	
2차	81.67	96.67	79.33	92.67	89	77.33	

프로세스 품질 변화 그래프



(그림 13) 품질평가 비교 그래프

5. 결론 및 향후 연구 과제

본 연구에서는 프로젝트 파라미터와 소프트웨어 품질특성간의 관계성을 이용하여 프로젝트 파라미터인 작업산출물을 평가하여 소프트웨어 품질을 평가하고 향상시키는 평가시스템을 구현하였다. 이 소프트웨어 평가시스템은 작업산출물에 대한 정성적 평가를 정량적으로 계측하여 입력하고 그 입력된 값을 이용하여 소프트웨어를 생산하는 관련된 범주별 프로세스와 현 단계에서의 소프트웨어 품질을 측정하였다. 그리고 측정된 수치를 그래프화 하여 평가의 이해를 가시화 하였다.

품질평가 시스템은 현재의 품질수준을 평가하여 확인하는 것보다는 소프트웨어 품질의 향상에 목적이 있으므로, 평가결과를 근거로 현 단계에서 미흡한 부분의 프로세스 과정을 파악, 개선하는 방법으로 사용한다. 그러므로 이 품질평가시스템은 프로젝트 진행과정에 따른 평가와 기존의 소프트웨어 단계별 평가방법을 수행하면서 두 평가 결과값을 비교하면서 병행하는 것이 효과적이다.

향후, 이 논문의 효용성을 높이기 위하여 프로세스 진행과정 중에 프로젝트 파라미터를 이용하여 중간생산물을 평가하는 방법과 기존의 최종산출물을 이용한 품질평가결과를 비교하고, 계속적으로 자료축적과 이에 따른 프로세스 개선을 통해 관련된 평가 매트릭스의 객관성과 타당성에 대한 지속적인 노력이 필요하겠

으며, 또 평가결과의 관리로 평가결과에 대한 적절한 평정수준을 마련해야 하겠다.

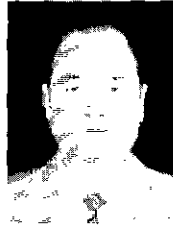
참고 문헌

- [1] Roger S Pressman, 'Software Engineering A Practitioner's Approach', 3rd ed McGraw-Hall Inc, pp 549-558, 1992
- [2] Chidamber, S. R. and Kemerer, C F., "Towards a Metrics suite for Object Oriented Design." OOPSLA '91, pp.197-211, Oct. 1991.
- [3] Conte, S D., Dunsmore, H. E and Shen, V. Y., "Software Engineering Metrics and Models," Benjamin/Cummings Publishing Company, Inc., 1985. 1.
- [4] Dunke, R., Neuman, K. and stoeffler, K., "The Metrics Based Compiler-A Concurrent Requirement," ACM SIGPLAN Notices, Vol. 27, No.12, pp.29-38, Dec. 1992.
- [5] 양해술, 이하용, '설계단계에서의 품질평가 툴킷의 설계 및 구현', 정보과학회 논문지, 제3권 제3호, 1997. 6.
- [6] 정호원, 양해술, 'ISO9000시리즈와 소프트웨어 품질시스템(상)', 하이테크정보, 1993
- [7] 이하용외, '소프트웨어 복잡성 측정 시스템의 설계 및 구현', 한국정보 처리학회 논문지, 제2권 제3호 pp.315. 1995.
- [8] 경기원의, '소프트웨어 프로세스와 품질', 홍릉과학출판사, 1997.
- [9] ISO/IEC Software Process Assessment-Part1 Concepts and Introductory Guide Working Draft V1.00.
- [10] Mark C. Paulk, 'A Perspective on the Issues Facing SPICE', Software Engineering Institute Carnegie Mellon Univ pp.415-424, 1995.
- [11] ISO/IEC Software Process Assessment-Part2. A model for process management Working Draft V1 00
- [12] Michael D. Konral, Mark C. Paulk, "An Overview of SPICE's Model for Process Management", Software Engineering Institute Pittsburgh, PA. 1995.
- [13] 안유환외, '소프트웨어 품질평가기술 개발(1), 1차년도 연구개발결과 보고서', 정보통신부, 1997.
- [14] Mark C Paulk, "The Capability Maturity Model For Software, Version 1.1", CMU/SEI-93-TR-24, ESC-

TR-93-177.

- [15] Trillium Release 3 Model for Telecom Product Development and Support Process Capability, Corporate Standard 133.20, Feb 1995.
- [16] 한재형편저, '실용프로젝트 관리론', 도서출판 니브, 1998.
- [17] 박철수, '회계관리 소프트웨어 패키지의 품질평가. 계층적분석과경의 적용'. 한국과학기술원 석사논문, 1995.
- [18] ISO/IEC 9126, "Information technology Production Software Evaluation Quality Characteristics and guidelines for their use," First edition Dec. 15. 1991.
- [19] 이경환저, '최신 소프트웨어 공학', 청문각, 1998
- [20] Schweiggert, F., Software Quality Softwaretest 5, 1994.
- [21] Perry, W. E., "Effective Methods of EDP Quality Assurance," In V N. Reinhold Handbook of Software Quality Assurance, 1987
- [22] ISO/IEC 9126 Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for their use, 1992.
- [23] ISO/IEC Software Process Assessment-Part4 Guide to conducting assessments Working Draft V1.00
- [24] ISO/IEC Software Process Assessment-Part5

Construction, selection and use of assessment instruments and tools Working Draft V1.00



이 준 길

e-mail : brother@kangnung.ac.kr
 1987년 서울시립대학교 전산통계학과 졸업(이학사)
 2000년 강릉대학교 산업대학원 컴퓨터과학과(이학석사)
 1989년 내무부 교통지도국 임용

현재 강릉대학교 전지계산소 근무
 관심분야 : 프로젝트 관리, 소프트웨어 품질평가 등



권 기 태

E-mail : ktkwon@kangnung.ac.kr
 1986년 서울대학교 계산통계학과 (이학사)
 1988년 서울대학교 계산통계학과 (이학석사)
 1993년 서울대학교 계산통계학과 (이학박사)

1995년~1996년 미국 Univ of Southern California, Dept of Computer Science, Post-Doc.
 1990년~현재 강릉대학교 컴퓨터과학과 부교수
 관심분야 : 소프트웨어공학, 데이터베이스, 인터넷 응용