

소프트웨어 품질향상을 위한 품질평가 모델에 관한 연구

조재규⁰ 이승종

국방대학교

dr-cjk@hanmail.net ljc@kndu.ac.kr

The Study of Quality Evaluation Model for Software Quality Improvement

Cho JaeKyoo Lee SungJong

Korea National Defense University

요 약

정보화 사회로의 급속한 전환으로 인해 고품질 소프트웨어에 대한 관심과 요구가 증대되었다. 따라서 양질의 소프트웨어를 확보하여 시스템의 안정성과 신뢰성을 높이기 위해서는 개발활동의 전 단계에서 품질보증 활동 및 평가가 이루어져야 한다. 개발 초기에는 프로젝트의 특성과 품질요구사항을 철저히 파악하여 품질목표를 설정하고 개발단계에서는 품질목표의 충족여부를 면밀히 점검하여야 한다.

본 논문에서는 소프트웨어 품질 평가의 기준이 되는 ISO 9126을 기반으로 소프트웨어 개발 전 단계에 대한 품질특성의 상관관계와 가중치를 설정하고, 정량화된 품질평가 모델을 제안하여 소프트웨어 개발의 성공에 실질적으로 지원할 수 있는 평가기준 정립 및 표준화를 위한 기초로 활용할 수 있다.

1. 서 론

정보화 사회로의 전환이 급속하게 이루어지고 있고, 국가기관 및 민간기업, 개인의 본격적인 정보화 추진으로 소프트웨어에 대한 실생활의 의존도가 크게 증가하고 있다. 이에 따라 고품질 소프트웨어에 대한 요구와 관심이 증대되었고, 소프트웨어의 개발, 운용, 사용 및 유지보수에 따른 위험이 점점 더 커지고 있다. 그리하여, 소프트웨어에 수반되는 위험을 사전에 식별하고 평가하여 소프트웨어 사용에 신뢰성과 안전성을 도모하는 소프트웨어 품질보증과 품질평가가 소프트웨어 개발과 함께 중요한 활동이 되고 있다.

기존의 소프트웨어 개발은 과거의 축적된 소프트웨어 평가경험 및 지식의 공유, 또한 평가기준과 절차에 대한 표준화 노력이 미흡하여 관련 소프트웨어 개발간 중복된 노력과 비효율적인 작업이 존재하였다. 또한, 품질보증 및 품질평가에 대한 모호한 기준으로 평가 시행시 지적사항에 대한 객관적인 기준제시를 통한 평가결과와 신뢰성이 제고되어야 하는데 제한적인 평가로 인하여 객관성과 충분성이 부족하여 품질평가의 신뢰성 확보에 한계가 있었다.

양질의 소프트웨어를 확보하여 시스템의 안정성과 신뢰성을 높이고 유지보수 노력을 최소화하여 경제성을 높이기 위해서는 개발활동의 전 단계에서 품질보증 활동 및 평가가 이루어져야 한다. 개발 초기에는 프로젝트의 특성과 품질요구사항을 철저히 파악하여 품질목표를 설정하고 개발단계에서는 품질목표의 충족여부를 면밀히 점검하여야 한다. 왜냐하면 소프트웨어는 개발이 완료된 상태에서의 시험평가만으로는 그 신뢰성과 안정성을 보장할 수 없고, 일단 개발이 완료된 소프트웨어를 개선하기 위하여 수정 혹은 변경하는데 개발에 투입된 이상의 시간과 비용이 소요되기 때문이다.

본 논문에서는 소프트웨어 품질 평가의 기준이 되는 ISO 9126을 기반으로 소프트웨어 개발 전 단계에 대한 품질특성의 상관관계와 가중치를 설정하고, 정량화된 품질평가 모델을 제안하고자 한다.

본 연구의 목적을 달성하기 위해 소프트웨어 품질과 품질 특성을 ISO/IEC 9126을 통하여 조사하고, 소프트웨어 각 개발단계에서 품질특성간 가중치를 분석하였다.

2. 소프트웨어 품질평가 일반 모델

소프트웨어 품질을 평가하기 위해서는 우선적으로 소프트웨어 제품에 요구되는 품질특성과 평가기준에 대한 명확한 기준이 필요하다. 소프트웨어 품질은 하드웨어와 달리 보는 사람의 관점에

따라 다르게 해석되는 경우가 많기 때문이다. 소프트웨어 품질특성의 정의 및 평가척도를 위한 품질모델에 관한 연구는 Boehm, McCall 및 기타 여러 연구를 통해 다양하게 제시되고 개선되어 왔다. 그러나, 현재 국제적으로 사용의 효과성을 인정받고 있는 품질평가 모델로는 ISO 9126을 들 수 있다.

ISO/IEC 9126은 사용자 관점에서 본 소프트웨어의 품질인자에 대한 국제표준으로 소프트웨어 품질을 객관적이고 계량적으로 평가할 수 있는 기본적 틀을 제공해 주기 위해 제정되었다.[1]

ISO/IEC 9126은 소프트웨어 평가를 위한 품질특성과 사용을 위한 지침으로 사용자 관점에서 품질에 관한 6개 외부 품질특성과 21가지 품질 부특성, 그리고 직접 외부특성인 품질특성과 품질 부특성을 추출하기 곤란한 경우가 있음을 고려하여 개발자 관점에서 파악할 수 있는 40개 항목의 내부특성을 제시하였다.[2]

ISO/IEC 9126의 품질특성은 소위 "FRUMPE" 모델이라 칭할 수 있는데 '이'는 품질특성의 첫 문자만을 엮은 것이다. ISO/IEC 9126의 품질특성 및 하부특성을 정리하면 [표 2-1]와 같다.[3]

품질특성	품질 부 특성
기능성	합목적성, 정확성, 상호운용성, 표준적합성, 보안성
신뢰성	성숙성, 장애허용성, 회복성
사용성	이해성, 습득성, 운용성
효율성	시간효율성, 자원효율성
보수성	해석성, 변경성, 안정성, 시험성
이식성	환경적응성, 설치성, 규격적합성, 치환성

[표 2-1] ISO/IEC 9126 소프트웨어 품질특성과 부특성

품질 내부특성은 개발자 관점에서의 소프트웨어 특성이며, 품질평가 결과를 산출하여 개발과정에 순환(feedback)하여 품질의 향상을 가져올 수 있으며, 각 항목별 세분화하여 정량적 수치를 부여함으로써 객관적인 품질평가가 가능해진다.

3. 소프트웨어 품질관리 및 품질평가

3-1 소프트웨어 품질관리

소프트웨어 품질관리는 프로젝트 수행시 요구된 개발 및 보증 활동과 이와 관련된 사항을 사전에 계획하고 조직화시킨 품질관리 체계에 따라 이루어져야 한다. 이에 따라 소프트웨어 품질관리 체계는 품질목표를 정의하고 활동방안을 모색하는 품질관리

계획, 각 개발단계별로 품질을 평가하여 문제점을 분석, 이를 시정하는 품질평가 및 개발된 제품의 변경을 처리하고 제품의 형상을 무결하게 유지하는 형상관리로 구성된다.[4]

품질관리 계획은 품질목표를 정의하고, 이를 실현하기 위해 개발 중 적용할 품질활동의 시행계획을 수립하여 공지하는 단계이다. 이 계획을 통하여 조직간 책임소재를 명확히 하고 유기적인 협조체제를 유지하게 된다.

품질평가는 개발내용이 요구사항을 충족하는지를 입증하고 개발형식이 제반 표준 및 지침에 부합하는지를 판정하기 위하여 품질점검 항목을 근거로 시행된다.

형상관리는 시스템의 구성요소를 관리가 가능한 형상개체로 분할하여 식별하고, 공식적인 변경절차에 따라 형상개체의 변경을 체계적으로 관리 하는 것이다.

3-2 소프트웨어 품질평가

소프트웨어 품질평가는 고품질의 소프트웨어를 생산하기 위한 하나의 방법이다. 이러한 소프트웨어 제품의 품질평가가 가장 중요한 요소는 평가에 있어서 객관성이 보장되어야 한다는 것과 정량적으로 측정하여 관리할 수 있어야 한다는 것이다. 따라서, 소프트웨어의 품질을 측정하고 평가하기 위해서는 소프트웨어 생명주기 단계별로 품질평가 체계를 확립하고 이러한 체계에 적용할 수 있는 평가 점검표의 개발이 필수적이다.

소프트웨어 품질평가 절차는 요구 정의 단계, 평가 준비 단계, 평가 단계의 세 가지로 구분하고 있다. [5]

품질요구 정의단계에는 품질 특성 및 이용 가능한 하부 특성들을 사용하여 품질 요구사항을 규정하는데, 이는 개발 이전에 반드시 정의되어야 하는 것이다. 소프트웨어 제품이 구성요소로 분해될 때, 전체 제품을 대상으로 선정된 품질 요구사항이 각 구성요소별로는 다를 수 있음을 인식하여야 한다.

평가준비 단계에는 품질 요구사항을 측정할 수 있는 매트릭스를 준비하여야 한다. 이때 소프트웨어 제품의 성질뿐만 아니라 환경과의 상호작용에 대한 매트릭스도 함께 준비하여야 한다. 매트릭스를 사용하여 측정된 값이 어느 등급에 속하는지에 대한 기준이 설정되어야 하며, 최종적인 판정기준도 사전에 정의되어야 한다.

마지막, 평가단계에는 실제로 측정하고 등급을 부여하며 수용 또는 기각 등의 판정을 내리게 된다. 측정이라 함은 선정된 매트릭스를 소프트웨어 제품에 적용하는 것이며, 판정이란 개별적으로 평가된 품질 특성들을 총체적인 관점에서 심사하여 품질수용 여부를 결정하는 것이다.

4. 제안된 평가기준 모델

4-1 연구모형의 설정

본 논문의 목적은 소프트웨어 개발 전 단계에 걸쳐 개발단계별 품질평가 기준을 도출하는데 있다. 따라서 본 논문에서는 ISO/IEC 9126에서 정의한 6개의 품질특성과 21개의 부특성, 40개의 내부특성을 이용하여 품질특성간 가중치를 파악하고 소프트웨어 개발 전 단계의 평가기준 모델을 제시하고자 한다.

본 연구의 진행을 위해 설정한 연구모형은 [그림 4-1]과 같다.

4-2 소프트웨어 개발 단계별 가중치 설정

4-2-1 품질 특성간 상관관계

ISO/IEC 9126에서 제시된 소프트웨어 품질속성의 정형화된 품질특성은 다분히 정성적인 의미를 지닌다. 따라서 소프트웨어 품질 측정은 품질특성이 포함하고 있는 정량적인 성질을 갖는 속성을 사용하여 간접적으로 이루어진다.[1]

품질 부특성과 품질 내부특성은 N : N 의 함수 관계를 갖는다. 즉, 하나의 부특성을 표현하는데 여러 개의 내부특성이 사용되며, 하나의 내부특성은 서로 다른 부특성에 영향을 미친다.

품질평가의 기준이 되는 품질 속성 사이에 존재하는 함수 관계로 소프트웨어 제품이 갖는 속성들의 집합인 품질특성 사이에도

관계성이 존재함을 확인할 수 있다.



개발단계별 주요 품질특성 가중치

[그림 4-1] 본 연구의 연구모형

4-2-2 개발단계별 품질 특성 매트릭스

품질 특성마다 측정하려는 측면이 다르기 때문에 각 단계에서 행하는 작업에 따라 강조되는 품질특성도 달라지게 된다.

[표 4-1]은 개발단계별 품질특성 매트릭스를 나타낸 것이다. 개발단계에 따른 품질특성이 강조되는 정도에 따라 아주 중요, 중요, 보통으로 구분할 수 있으며, 소프트웨어의 유형에 따라 선택적 보통이 포함될 수도 있다.

품질 특성 / 부 특성	기능성		신뢰성		사용성		효율성		보수성		이식성									
	합목적성	정확성	표준화합성	보안성	장애허용성	회복성	이해성	습득성	운용성	시간효율성	자원효율성	해석성	변경성	안정성	시험성	환경적응성	실치성	규격적합성	치환성	
계획 단계	●	○	○	◇	◇	◇	◇	○	◇	◇	◇	◇	◇	◇	◇	◇	○	○	○	○
요구 분석 단계	●	◇	◇	○	○	○	◇	○	○	○	◇	◇	◇	◇	○	○	○	○	○	○
설계 단계	○	○	○	◇	○	○	◇	○	○	○	○	○	○	○	○	○	○	○	○	○
구현 단계	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
시험 단계	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
전이 단계	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

(● : 아주 중요, ○ : 중요, ○ : 보통, ◇ : 선택적 보통)

[표 4-1] 개발단계별 품질특성 매트릭스

계획단계에서 중요한 품질특성은 기능성, 사용성 및 이식성이다. 이 단계에서는 소프트웨어 개발 대상인 업무체계의 통합과 사용자에 대한 정확한 이해가 중요 관리요소이다. 이는 품질 부특성 중 합목적성, 이해성, 환경적응성이 중요한 요인으로 활용될 수 있음을 나타낸다.

요구분석 단계에서 중요한 품질특성은 기능성과 사용성, 이식성이다. 이 단계에서는 계획단계를 보다 상세화하고 문서화하는 단계로 업무기능이 얼마나 적절하고, 구체적으로 문서화되었는지를 판정하는 것이 중요한 요소이다. 품질 부특성 중에서는 합목적성, 정확성, 장애허용성, 이해성 등이 중요한 요인으로 활용된다.

설계단계에서 중요한 품질특성은 기능성, 효율성, 보수성이다. 이 단계에서는 시스템 기능을 구조화하고 상세화하는 것과 주어진 자원 조건하에서 최대의 효율발휘가 중요관리요소이다. 품질 부특성 중에서는 합목적성, 장애허용성, 시간효율성, 자원효율성, 해석성, 변경성, 시험성이 중요한 요인으로 활용된다.

구현단계에서 중요한 품질특성은 신뢰성, 효율성, 보수성이다. 이 단계에서는 요구기능의 설계내용에 충분한 이행, 사용자에 영향을 미치는 각종 처리시간 및 자원관리, 유지보수에 대한 준비가 중요관리요소이다. 품질 부특성 중에는 정확성, 성숙성, 장애허용성, 회복성, 이해성, 운용성, 시간효율성, 자원효율성, 시험

성이 중요한 요인으로 활용된다.

시험단계에서 중요한 품질특성은 기능성, 신뢰성, 효율성이다. 이 단계에서는 사용자에게 의해서 요구된 기능이 얼마나 정확히 동작하는지, 소프트웨어 운영간 우발상황에 대한 대처는 적절한지, 명세된 기능이 준비된 시간과 자원을 사용하는지가 중요관리요소이다. 품질 부특성은 합목적성, 정확성, 성숙성, 장애허용성, 회복성, 이해성, 시간효율성, 자원효율성, 시험성이 중요한 요인으로 활용된다.

마지막으로 전이단계에서는 사용성과 이식성이 중요한 품질특성이다. 이 단계에서는 새로운 소프트웨어에 대한 사용자의 교육과 이해와 새로운 환경에서의 소프트웨어의 적용이 중요관리요소이다. 품질 부특성은 이해성, 운용성, 안정성, 환경적응성, 규격적합성이 중요한 요인으로 활용된다.

4-3 품질평가

소프트웨어 품질평가 대상은 각 단계에서 생산되는 개발규격서, 계획서, 보고서 등의 제반 문서 및 프로그램이다. 품질평가 대상에 대해 품질평가 점검표를 작성하여 평가한다. 평가 점검표의 각 평가항목은 주관성을 최소화 하기 위해 가능한 한 범위를 축소하여 명확하게 답변할 수 있는 내용이어야 하고, 판정을 위한 지침 또는 기준을 사전에 명시해야 한다.[6]

평가점검표에 의한 평가단위별 등급은 5단계 품질평가 등급표 [표 4-2]를 통해 평가수준을 적용할 수 있다.

평가등급	평가점수	평가수준
우수	5	적합
양호	4	적합
보통	3	보통
미약	2	부적합
불량	1	부적합

[표 4-2] 품질평가 등급표

각 단계별 평가항목에 대한 평가점수는 다음과 같이 산정할 수 있다.

먼저 품질 내부특성이 획득한 평점 C_j 는 다음 식을 이용해서 계산된다. 여기서 평가등급은 5단계 평가로 가정하며, N 은 내부특성별 평가항목 수를 나타내고 N_i 은 평가항목별 평가점수를 나타낸다.

$$C_j = \frac{\sum N_i}{N}$$

품질 부특성 F_k 의 평점은 해당 내부특성 C_j ($j=1, \dots, m$)에 대한 가중평균으로 다음 식을 이용해서 계산한다. 여기서 W_j 는 내부특성 C_j 의 가중치이다.

$$F_k = \frac{\sum_{j=1}^m W_j \times C_j}{\sum_{j=1}^m W_j}$$

품질 특성 S_x 의 평점은 해당 품질 부특성 F_k ($k=1, \dots, n$)에 대한 가중평균으로 다음 식을 이용해서 계산한다. 여기서 W_k 는 품질 부특성 F_k 의 가중치이다.

$$S_x = \frac{\sum_{k=1}^n W_k \times F_k}{\sum_{k=1}^n W_k}$$

현 개발단계에서의 품질목표 달성수준 G 는 다음 식을 이용해서 계산한다. 여기서 S_x 는 각 단계별 해당되는 품질특성의 평점이며, N 은 품질특성의 수를 말한다.

$$G = \frac{\sum S_x}{N}$$

품질목표 달성수준 G 는 5단계 품질합격 기준표[표 4-3]에 적용하여 다음 단계로 진행할 것인지, 이전 단계로 피드백 할 것인지를 판단한다.

평가등급	평가점수	평가수준
우수	4.50 이상	적합
양호	4.00 ~ 4.49	적합
보통	3.50 ~ 3.99	적합
미약	3.00 ~ 3.49	부적합
불량	3.00 미만	부적합

[표 4-3] 품질합격 기준표

5. 결론 및 향후 연구과제

소프트웨어의 품질에 대한 관심이 높아지고, 대규모 소프트웨어 개발에서 품질저하의 심각성이 대두됨으로써 품질보증 활동 및 품질평가를 통한 고품질 소프트웨어 개발의 중요성이 높아지고 있다. 따라서 품질평가에 있어서는 기존의 주관적인 평가방식을 버리고 객관적인 방법을 도입하여 실시해야 하며, 소프트웨어 개발초기부터 소프트웨어 품질특성을 고려하여 개발하고 각 단계별 품질관리를 철저히 하는 것이 중요하다.

본 연구에서는 소프트웨어 개발 각 단계별로 체계적이고 표준화된 품질평가를 실시하여, 발견된 오류에 대한 피드백을 실시하고 단계별 생산되는 산출물의 품질을 높이기 위해 품질평가의 기준모형을 제시하였다.

품질평가 기준모형의 객관성을 높이기 위해 국제적으로 인정받고 있는 ISO/IEC 9126 에 근거한 품질특성을 바탕으로 각 품질부특성, 품질 내부특성에 관한 가중치를 제시하였다. 이러한 연구결과를 적용해본 결과, 소프트웨어 전체개발소요 시간을 단축시킬 뿐만 아니라 각 단계별 오류발생 가능성을 최소화 할 수 있다. 또한 품질특성에 관한 존재하는 관계성 확인에 그치고 있는 기존의 소프트웨어 품질에 관한 연구에 발전된 기준이 될 것으로 본다.

앞으로 계속 연구되어야 할 부분은 제시된 품질특성 평가기준모형을 산출물에 적용하여 그 효용성을 판단하고, 지속적으로 보완해야할 필요성과 품질특성간 가중치 정량화에 대한 지속적인 연구와 분석이 필요하다. 또한 소프트웨어 개발단계 뿐만 아니라 소프트웨어 유형에 대한 평가기준 모형을 정립하고 개발단계 및 유형이 모두 고려된 평가기준 모형을 개발해야 할 필요가 있다.

참고문헌

- [1] 정기원, 윤창섭, 김태현, 소프트웨어 프로세스와 품질, 홍릉과학출판사, 1997.
- [2] ISO/IEC, International Standard Information Technology, - Software Evaluation, ISO/IEC 9126, 1991.
- [3] 선우중성 외 3인, 정보시스템 품질관리 감리지침에 관한 연구, 한국전산원, 1998.
- [4] 박남규, 소프트웨어 프로젝트 수행 방법론, 한국 실리론, 1996.
- [5] 양해술, 정호원, ISO 9000 시리즈와 소프트웨어 품질시스템, 하이테크정보, 1993.
- [6] 이광진 외 3인, 소프트웨어 품질평가체계에 관한 연구, 한전품질보증단, 1988.