소프트웨어 품질관리를 위한 품질평가 기술

권오탁(경북실업전문대학 경영정보과)

제 1 장 서 론

소프트웨어 비용의 상승과 그 중요성이 증대함에도 불구하고 소프트웨어 그 자체가 사용자에게 신뢰성을 제공하지 못할 뿐 아니라, 그것을 개발하는 전산 담당자에게도 비용, 개발과정, 기술적인 면에서의 성능 등의 제약조건을 충족시키지 못하여 실패한 다면 불행한 일이다.

소프트웨어 제품의 품질이 보장되지 못하는 것은 제품 생산시에 소프트웨어의 품질 특성이 충분하게 파악, 이해되지 못함으로 인한 관리부재에 그 원인이 있다.

따라서 본 연구는 소프트웨어 제품이 지녀야 할 품질특성 및 평가척도에 관한 기존의 연구이론을 정리하여 보고, 평가하기 위한 일반화 된 품질척도 측정내용을 연구하고자 한다.

연구의 내용은 먼저 소프트웨어의 의미와 분류 그리고 소프트웨어 품질에 대하여 정리하였으며 소프트웨어 품질 평가절차와 품질평가를 위한 기존의 품질특성을 조사 연구하여 품질특성별 품질척도 내용을 제시하고자 한다.

제 2 장 소프트웨어 품질

1. 소프트웨어 정의

Pressman에 의하면 소프트웨어는 원하는 기능을 수행하는 컴퓨터 프로그램(working program)과 프로그램이 적절하게 다룰 수 있도록 하는 자료구조(data structure) 그리고 프로그램이 설계, 이용, 개발, 추진보수하는데 필요한 문서 체계(document)라고 정의하였다.

한편 Milis H.D는 소프트웨어를 <그림 2-1>과 같은 7가지 형태의 집합이라고 했다.

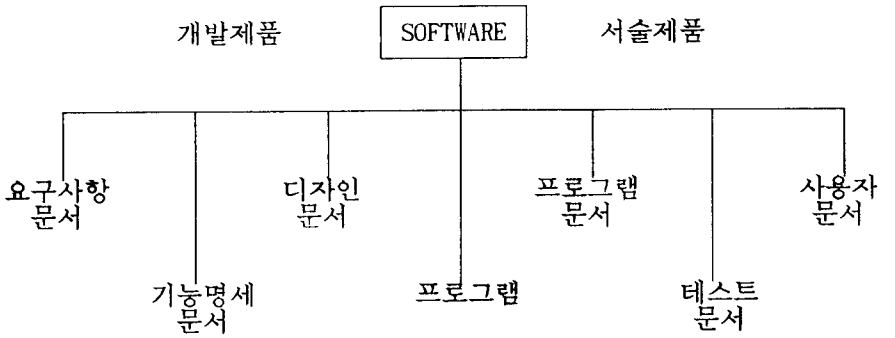
이와같이 소프트웨어는 단순한 컴퓨터 프로그램 또는 코드만이 아닌 요구정보 분석에서부터 최종 제품의 생산 및 유지보수 과정까지 모든 단계에서 생성되는 모든 것의 총체이다.

소프트웨어를 자세히 이해하기 위해서는 소프트웨어 종류의 내용이 무엇인지를 알면 도움이 된다.

소프트웨어를 소프트웨어 특성에 따라 분류를 할 수 있는데 이는 매우 다양하다. 최근의 우리나라의 소프트웨어 분류체계 중 한국소프트웨어산업협회에서는 시스템 소프트웨어, 시스템 개발지원 소프트웨어, 범용응용 소프트웨어, 특정 업무용 소프트웨어와 같이 크게 4가지로 분류하였다. 시스템 소프트웨어에는 운영체제, 데이타 통신 및

네트워크, 데이타베이스 관리시스템, 컴퓨터 언어, 인간-기계 인터페이스, 시스템 운용·관리, 유틸리티로 구분하였다.

<그림 2-1> 소프트웨어 정의



시스템 개발지원 소프트웨어는 CASE도구, 소프트웨어 재사용으로 구분하였으며, 범용응용 소프트웨어는 사무자동화, 인공지능, 그래픽 및 화상처리, 설계자동화, 유통자동화, 공장자동화, 음성 및 음악처리, 경영정보, 기술계산 및 통계, 교육용, 오락용, 지식기반 등으로 분류하였다.

이외의 특정업무를 위한 소프트웨어는 특정업무용 소프트웨어로 분류하였는데 그 내용은 [표 2-1]과 같다.

2. 소프트웨어 품질

소프트웨어 품질을 정의하기란 매우 어렵다. 미국전기전자공학회(IEEE)에 의하면 '소프트웨어가 지닌 바람직한 속성의 정도'라고 정의하며, 미국 국방성에서는 '요구되는 기능을 발휘할 수 있는 소프트웨어 특성의 정도'라고 정의하고 있다. 품질관리 학자인 J. M. Juran은 '소프트웨어 상품의 사용적정도(Fitness for Use)'라 정의한다.

품질의 의미는 구체적인 의미는 시대와 더불어 변하고 있으며 품질의 의미는 크게 몇가지로 분류된다. 과정품질(Process Quality)은 제품을 개발하기 위해 필요한 모든 활동(기획, 설계, 실현)의 모든 과정에서 최상의 기술수준으로 개발되었는가를 나타내는 품질이며, 최종제품품질(Product Quality)은 완성된 소프트웨어가 운용될 환경에 올려져 최종 시스템이 완성되었을 때 소비자가 요구하는 바에 얼마나 부합되는지를 나타내는 품질이다. 그리고 제조품질(Quality of Conformance)은 적합품질이라고도 하며, 생산과정에서 제조된 제품이 설계품질에 어느 정도 적합한가를 나타내는 품질 등을 의미한다. 서비스 품질(Quality of Service)은 소비자가 제품을 올바르게 사용할 수 있도록 사용방법을 전달해 주는 질, 그리고 제품 사용상 문제가 생겼을 때 아프터서비스의 질 등을 의미한다.

소프트웨어 품질은 관점에 따라 다양하다. 관점으로는 개발자, 사용자(구매자), 평가자 및 관리자 측면으로 구분되며, 각 관점에 따른 품질평가 내용은 다음과 같다.

개발자(Developer) 관점의 품질평가는 코드의 양과 같이 단독적으로 평가가 가능한 내부적(internal) 품질속성, 신뢰성과 같이 시스템 환경이 고려되어야 만 평가할 수 있는 외부적(external) 품질속성, 시스템을 실행하지 않고도 측정가능한 정적(static) 품질속성, 시스템을 실행하여야만 측정가능한 동적(dynamic) 품질속성이 있다.

〈표 2-1〉 소프트웨어 분류표

대분류	중 분 류	소 분 류
시 스 템 소프트웨어	운영체제	단일유저용, 다수유저용, 실시간처리용, 기타
	데이타통신 및 네트워크	네트워크운영체제, 통신, 프로토콜, 통신, 에뮬레이터, 네트워크 화일시스템, 기타
	데이타베이스 관리시스템	단일유저용, 다수유저용, 분산처리용, 엔지니어링용, 멀티미디어지원, 기타
	컴퓨터 언어	고급언어컴파일러, 인터프리터, 어셈블러, 기타
	인간-기계 인터페이스	윈도우시스템, 간이언어(저작언어)처리기, 기타
	시스템 운용·관리	프로그램 관리, 시스템 감시, 시스템 보안처리, 자동 오퍼레이션, 기타
	유틸리티	에디션, 화일시스템, 디스크 정보관리, 데이타 압축 및 해체, 기타
	기타	한글 및 한자 처리, 바이러스 처리, 기타
시스템 개발지원 소프트웨어	CASE도구	요구분석지원, 설계지원, 프로그램 생성기, 시험지원, 기타
	소프트웨어 재사용	소프트웨어 부품관리, 프로그램 생성기, 시험지원, 기타
범 용 응 용 소프트웨어	사무자동화	문서 편집기, 표 계산기, 전자우편, 전자출판, 그림 및 도형 편집기, 일정관리, 프리젠테이션 지원 저작도구, 기타
	인공기능	인공지능언어, 자연어처리(기계번역, 자연어검색, 음성인식, 음성합성), 전문가 시스템, 영상처리(문서, 영상), 기타
	그래픽 및 화상처리	지도관리, 의료진단, 기상·탐사, 검사장치, 디자인(광고, 도안), 애니메이션도구, 안내지원 저작도구, 기타
	설계자동화	설계자동화, 3-D시뮬레이션, 도면관리, 구조해석, 기타
	유통자동화	매장관리, 재고관리, 판매관리, 기타
	공장자동화	CIM, 생산관리, 기타
	음성 및 음악처리	음성발생처리기, 악보작성처리기, 음악실행기, 기타
	경영정보	관리, 회계, 인사·급여, 영업관리, 의사결정지원, EDI관련, 기타
	기술계산 및 통계	해석 및 예측처리, 통계처리, CAE, 기타
	교육용	교육용 소프트웨어 저작도구, 코스웨어, 학사관리, 기타
	오락용	게임제작도구, 게임
	지식기반	취미, 전자사전, 그림 부품집, 특수목적문서집, 기타
특정업무용 소프트웨어	건설업, 제조업, 운수업, 도소매업, 음식점업, 물품임대업, 조사서비스업, 금융업, 부동산업, 숙박업, 의료업, 사설교육업, 기타	

사용자(user)관점의 품질평가는 소프트웨어의 내부적 품질속성이나 개발방법보다는 소프트웨어의 성능 및 소프트웨어를 사용함으로써 나타나는 효과를 고려하므로 소프트웨어의 외부적, 동적인 면을 평가하고 중간제품보다는 최종 제품 품질평가에 관심을 둔다.

평가자(evaluator) 관점의 품질평가는 전문평가기관의 전문가로써 개발자나 사용자 측의 요구(또는 양측의 일치된 요구)에 의해 품질평가를 수행한다. 품질비용이 발생되고 법적인 문제도 다루어야 한다.

판정자(assessor)관점의 품질평가는 개발기관이나 발주기관의 개발과 관련된 최고 책임자로써 개발기관의 판정자와 발주기관 판정자가 있다. 개발기관 판정자는 사업성이나 개발비용 등을 고려하여 개발된 제품의 시장출하 또는 발주기관으로의 공수결정 및 시기 등을 결정하는 책임자이며, 발주기관 판정자는 제품인수에 앞서 제품의 전반적인 품질을 판정하는 인수결정자이다.

3. 품질특성

종래의 소프트웨어 품질을 만드는 측면에서는 품질척도가 평가되어 왔다. 그러나 소프트웨어 제품이 널리 유통됨에 따라 품질에 대해서도 많은 측면에서 평가되도록 되었다. 또한 품질요구도 복잡하게 되어왔으며, 소프트웨어 품질을 사용하는 측의 입장과 개발하는 측의 입장 양방향에서 대각적으로 평가하려고 하는 방법론이 개발되었으며 소프트웨어를 모델화함과 함께 품질척도가 마련되었다.

3.1 Boehem의 품질특성

Boehem의 품질특성 모델은 소프트웨어 품질을 휴대성, 유용성, 보수성으로 크게 구분하였고, 유용성은 다시 신뢰성, 효율성, 조작성으로, 보수성은 테스트 용이성, 이해성, 갱신성으로 세분하였다.

3.2 McCall의 품질특성

McCall은 소프트웨어의 품질요인을 운영, 변경, 개선의 변경관련에는 이식성, 재사용성, 상호운용성 그리고 개선관련 요인은 보수성, 융통성, 테스트 용이성 등으로 구분하였다.

3.3 ISO/IEC 9126의 품질특성

가) 기능성(Functionality)

기능집합과 사양화 된 내용을 실현하는 특성의 집합으로써 명시적 또는 암시적 요구를 만족하는 속성 집합

① 적합성(Suitability) : 사양화 된 태스크에 대한 일련된 기능존재나 적절한 기능을 지닌 소프트웨어 속성

② 정확성(Corectness) : 올바른 규격, 일치된 결과 그리고 정확한 결과를 나타내는 소프트웨어 속성

③ 상호호환성 (Interoperability) : 사양화 된 소프트웨어가 다른 시스템과 상호운용능력을 지닌 소프트웨어 속성

④ 유연성(Compliance) : 법률이나 동일한 형태의 규칙, 규격, 협정에 대한 유연성을 지닌 소프트웨어 속성

⑤ 보안성(Security) : 우발적이거나 고장에 의한 것이 아닌 부당한 액세스로 부터 프로그램이나 데이타를 보호하는 소프트웨어 속성이 아닌 부당한 액세스를 방지하는 속성

나) 신뢰성(Reliability)

명시된 조건하에서 명시된 기간, 소프트웨어의 실행 레벨을 유지하기 위한 능력을 만족하는 속성의 집합

① 성숙성(Maturity) : 소프트웨어에 잔재하는 장해에 의한 고장을 대처하는 기능의 소프트웨어 속성

② 오류허용성(Error Tolerance) : 소프트웨어 장애 또는 사양화 된 인터페이스를 유지하지 않은 경우에 사양화 된 실행수준으로 회복하기 위한 능력과 시간을 가지는 소프트웨어 속성

③ 회복성(Recoverability) : 고장시에 소프트웨어의 실행수준을 재확인하여 직접 영향을 받는 데이타를 회복하기 위한 능력과 시간을 가지는 소프트웨어 속성

다) 사용성(Usability)

명시적 또는 암시적으로 사용자가 이용하기 위해 필요한 노력으로 각각의 사용결과에 대한 평가를 나타내는 속성의 집합

① 습득성(Learnability) : 소프트웨어의 운용관리, 입출력 방법 등을 습득하기 위한 사용자의 노력을 나타내는 소프트웨어 속성

② 운용성(Operability) : 소프트웨어의 운용과 운용관리를 하기 위한 사용자 노력 정도를 나타내는 소프트웨어 속성

③ 이해성(Understandability) : 소프트웨어의 논리적인 개념을 사용자의 이해정도를 나타내는 소프트웨어 속성

라) 효율성(Efficiency)

명시적인 조건하에서 소프트웨어의 실행레벨과 사용된 자원 양자 간의 관계를 나타내는 소프트웨어 속성의 집합

① 실행효율성(Time behavior) : 소프트웨어 기능을 수행할 때 응답시간과 처리시간을 나타내는 속성

② 자원효율성(Resource behavior) : 소프트웨어의 기능을 실행할 때 사용되는 자원의 양과 사용시간을 나타내는 소프트웨어 속성

마) 유지보수성(Maintainability)

사양화 된 개정을 처리하기 위해 필요로 하는 노력을 나타내는 속성의 집합

① 해석성(Analyzability) : 고장의 원인 분석이나 결합분석 및 개정 가능한 부분의 정도와 사용시간을 나타내는 소프트웨어 속성

② 변경성(Changeability) : 개정을 위한 장애제거나 환경변경에 필요한 노력을 나타내는 소프트웨어 속성

③ 안정성(Stability) : 개정에 의해 예기되는 효과의 포함성을 나타내는 속성

④ 시험성(Testability) : 소프트웨어의 타당성 검증에 필요한 노력을 나타내는 속성

바) 이식성(Portability)

임의의 환경에서 다른 환경으로 소프트웨어를 이식하기 위한 속성의 집합

① 환경적응성(Adaptability) : 이식목적에 적합한 작업이나 수단 이외의 것은 이용하지 않고 사양화 된 다른 환경으로 소프트웨어를 이식할 수 있는 소프트웨어 속성

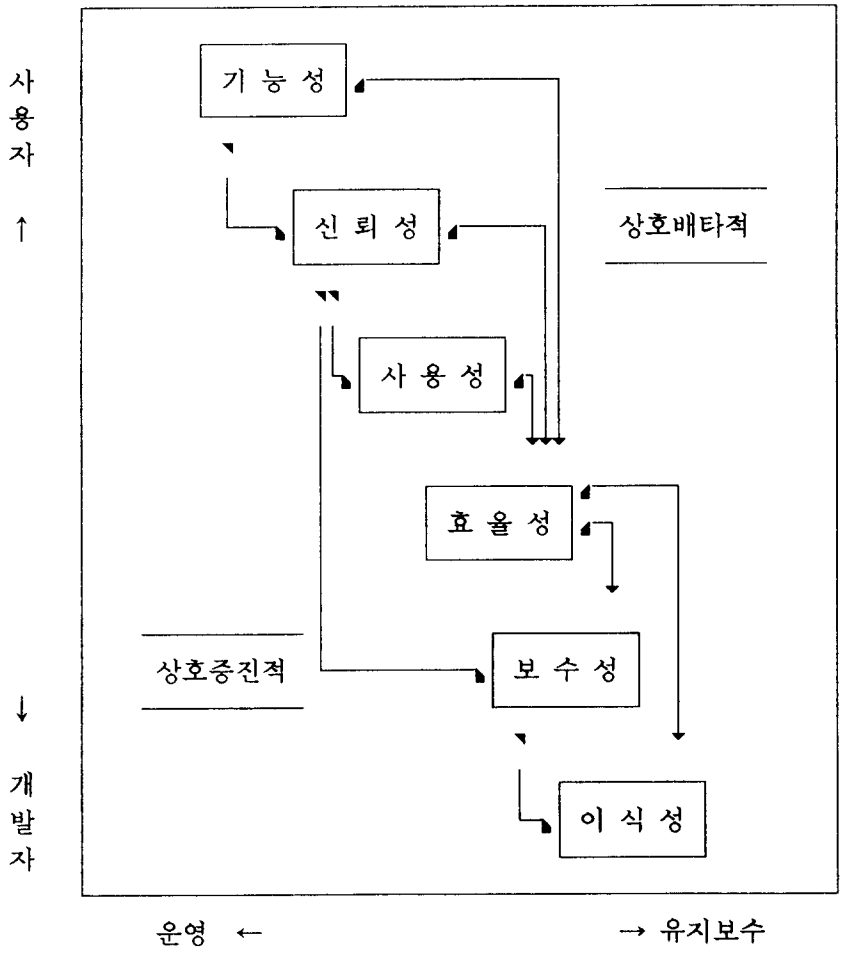
② 이식작업성(Installability) : 소프트웨어를 사양화 한 환경에 도입하기 위해 필

요한 노력을 나타내는 속성

③ 일치성(Conformance) : 규격이나 규정에 소프트웨어를 적합시킬 수 있는 속성

④ 치환성(Replaceability) : 사양화 된 다른 소프트웨어와 치환되어 그 환경에서 사용하기 위한 기회나 노력을 나타내는 속성

<그림 2-2> 품질특성의 상호관련



한가지 중요한 사실은 소프트웨어 품질특성 중 사용자가 [그림 2-2]에서와 같이 어느 특성을 더 중요시하는가를 결정해야 하는 것이지, 모든 특성을 고루 갖춘 소프트웨어를 제작할 수는 없다는 것이다. 효율성은 기능성, 신뢰성, 사용성, 보수성, 이식성과 상호배타적인 성질이며, 기능성은 신뢰성과, 신뢰성은 사용성과 보수성, 보수성은 이식성과는 상호증진적인 성질이 있다.

일반 사용자라면 먼저 기능성이 강조되고 그 다음 신뢰성, 사용성이 필요할 것이며, 전문 소프트웨어 하우스에서는 이식성과 유지보수성이 중요시 된다. 업무 소프트웨어라면 사용성과 유지보수성을 우선으로 할 것이다.

제 3 장 소프트웨어 품질평가

1 품질평가 정의 및 대상

소프트웨어 품질평가는 개발중인 소프트웨어가 사용자의 요구사항을 만족하는가를 평가하기 위해 객관적이고 정량적인 방법을 이용하여 소프트웨어의 품질을 측정하는 활동이다. 따라서 고품질의 소프트웨어를 확보하기 위해서는 개발전에 사용자의 요구사항에 품질요구사항이 정확히 명시되어야 하며 개발과 병행하여 체계적인 절차에 따라 평가수행되어야 한다.

품질평가의 대상으로 소프트웨어 개발 단계에서 작성되는 산출물 및 개발과정과 개발지원기술 등이 있다. 개발지원기술은 제품의 품질을 예상하기 위하여, 분석이나 설계기법 및 CASE 도구가 적절히 선택되고 사용되는지를 평가한다. 자원은 제품의 품질 및 프로젝트 비용을 예상하기 위하여 인적자원이 적절히 선택되고 배치되었는지를 평가한다. 중간산출물은 중간 개발단계에서 만들어지는 산출물을 평가하는 것으로, 다음 단계로의 진행을 결정하는 중요한 요소가 된다. 최종산출물은 최종승인의 대상이 되는 개발 제품을 평가하거나, 상호경쟁이 되는 제품들을 비교하기 위해 평가한다.

2 품질평가 절차

품질평가를 위한 절차에 대해 ISO/IEC 9126에서는 [그림 3-1]과 같이 제시하고 있다.

2.1 요구단계

품질요구 정의(Quality Requirement Definition)는 요구자의 요구를 만족하기 위해서 소프트웨어에 요구되는 조건을 분석하고 사양화하는 과정이다. 품질평가에서는 일반적으로 기능요구가 기초하여 품질요구를 품질특성에 적용시킨다. 품질요구 정의 단계에서는 사용자의 요구에 적합한 소프트웨어를 개발하기 위해 단순히 기능요구 뿐만 아니라, 어떤 형태의 품질특성이 얼마나 중요한가를 명확히 정의하여야 한다.

2.2 준비단계

가) 메트릭스 선택(Metrix Selection)

메트릭스는 계측대상의 소프트웨어와 공정에 따라서 다르다. 각 응용분야에 따라 각기 다른 메트릭스가 추출될 수 있으므로 구체적이고 명시적인 메트릭스 선택의 원칙을 제시하기는 어려우나 일반적으로 사용하고 있는 성능, 신뢰성 계측을 위한 메트릭스 등 국한된 메트릭스 뿐만아니라 특수한 메트릭스를 개발함으로써 특수한 목적을 지니는 소프트웨어도 평가할 수 있도록 메트릭스를 개발하거나 대상복구에 따라서 메트릭스를 선택적으로 사용하는 것이 가능하다.

나) 평정수준설정(Rating Level Definition)

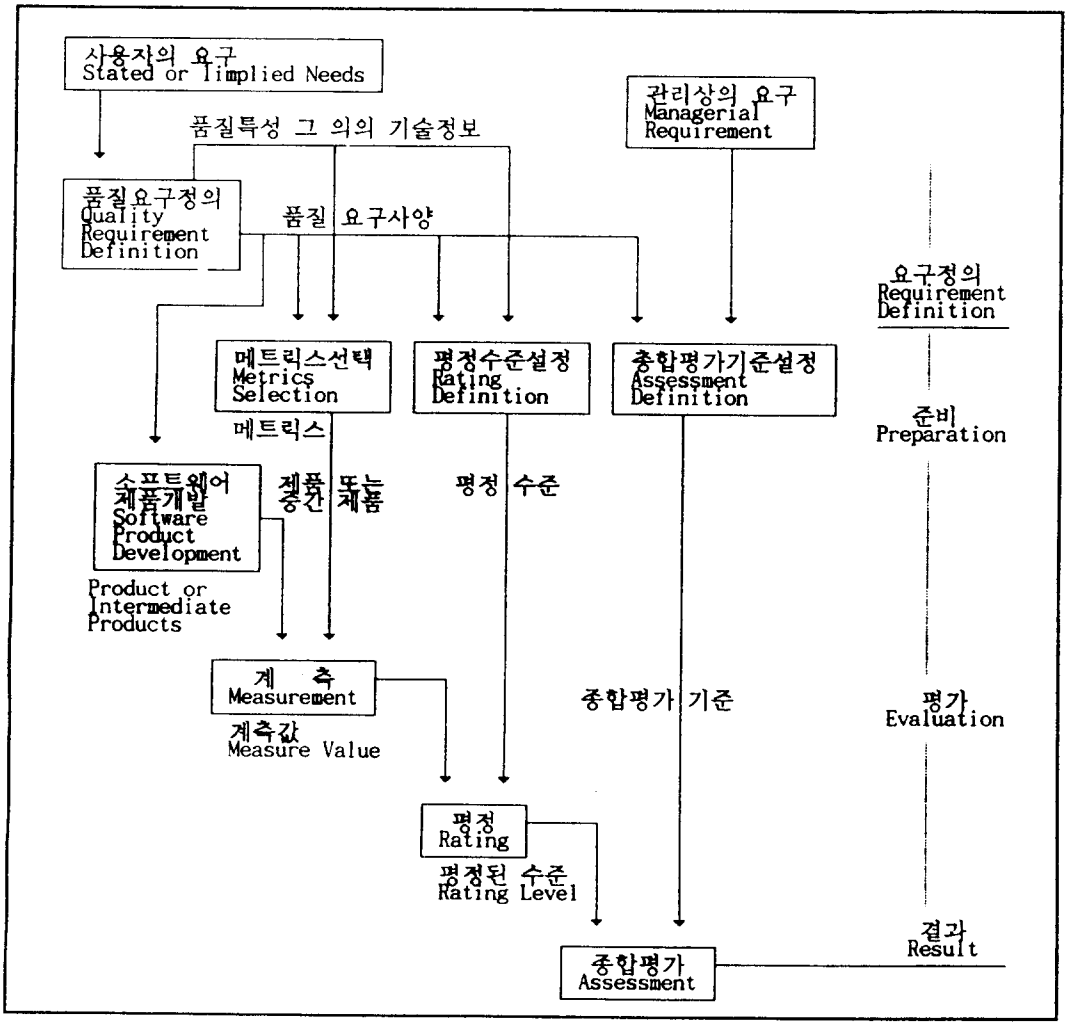
메트릭을 소프트웨어 속성에 적용하는 과정은 측정단계에서 일어나게 되는데 측정이 정상적으로 수행되었다면 모든 측정의 결과로 측정값이 산출된다. 계측한 결과는 단지 수치를 나타내는 것이므로 그것이 적절한 값인지 또는 부적절한 값인지를 알 수 없다. 따라서 대상으로 하는 시스템에 대하여 측정값의 적절한 정도를 우수/적당/부적

당 등과 같이 몇 개의 단계로 분류하여야 한다. 이와 같은 것을 소프트웨어 품질평정 수준(평정레벨)이라 한다.

다) 총합평가 기준설정(Assessment Criteria Definition)

품질특성, 품질척도마다 평정수준을 결정하고, 이 결과를 기초로하여 전체적으로 소프트웨어의 품질이 어느 레벨에 해당되는지를 결정할 수 있도록 총합평가 기준을 설정하는 것이다. 총합평가기준은 소프트웨어 품질에 우수/적당/부적당한지를 평가하기 위해 기준 을 설정하는 단계이다.

<그림 3-1> ISO/IEC 9126 품질평가 절차



2.3 평가단계

가) 품질계측(Quality Measurement)

평가대상 소프트웨어를 실제로 계측하여 척도로 나타내는 단계를 품질계측 단계라 한다.

이는 소프트웨어의 품질특성이 요구되는 품질을 갖추고 있는지 없는지를 평가하기 위해 메트릭스 값을 실제로 측정하여 정량적으로 평가하는 단계이다.

나) 평정(Rating)

메트릭을 적용하여 얻은 측정값과 평정수준을 비교하여 소프트웨어의 품질특성이 어느 수준에 속하는지를 결정하는 단계이다. 평정수준에 정의된 목표측정값과 측정결과 값이 상대적 및 절대적 비교, 허용범위 내에 포함되어야 하는 결과값의 갯수, 평균값, 중위수, 편차 및 분산 등의 통계치를 이용하여 평가한다. 계측값이 주관적인 입장을 완전히 배제할 수 없으므로 평정 가이드라인을 제공할 필요가 있다.

다) 총합평가(Assessment)

평가대상이 되는 소프트웨어 제품의 총합적인 평가하기 위해 계측된 모든 품질특성의 결과값과 가중치를 고려한 계산식에 의해서 산출된 결과값을 총합평가 기준에 적용하여 판정하는 단계를 총합평가 단계라 한다.

2.4 평가보고 단계

평가단계를 완료한 후 이제까지의 평가수행 내역을 정리하여 평가보고서로 기록, 보관한다. 평가보고서는 개발기관 품질보증조직이 내부적, 비공식적인 평가를 수회반복하여 수행함으로써 자체 개발제품에 대한 품질의 달성정도를 점검하는 수단으로 이용할 수 있고, 개발 요구자의 요구사항 및 품질만족도를 확인하는 등 개발자와 요구자 사이의 의사교환수단으로 활용할 수 있다.

제 4 장 품질특성별 평가항목

2장에서의 내용과 같이 대표적인 품질특성은 Boehem의 품질특성, McCall의 품질특성, ISO/IEC 9126의 품질특성이다.

이중 국제적으로 인정받고 있는 가장 최근의 품질특성 모델인 ISO/IEC 9126의 품질특성 모델을 이용하여 품질척도를 측정하기 위한 일반적인 평가항목 및 평가내용을 연구하였다.

소프트웨어 품질관리를 위한 표준화의 일환으로 소프트웨어 품질특성을 ISO/IEC 9000에서 정의하였다. 그후 ISO/IEC 9126에서 품질척도까지 정리하여 공표하였다. 그러나 객관적인 평가척도를 사용할 수 있는 평가내용이 정립되지 않고 있는 상태이다. 물론 대상 소프트웨어의 성격에 따라 평가방법이 매우 다양하지만 소프트웨어 특성에 따라 선택해서 사용할 수 있는 내용이 필요하다고 생각되서 다음과 같이 정리해 보았다.

1. 기능성 평가항목

품 질 특 성	기능성(Functionality)	
기능집합과 사양화 된 내용을 실현하는 특성의 집합으로써 명시적 또는 암시적 요구를 만족하는 속성 집합		
품 질 척 도	평 가 항 목	평 가 내 용
적합성 (Suitability)	사양화 된 태스크에 대한 일련된 기능존재나 적절한 기능을 지닌 소프트웨어 속성	
	요구충족성	. 시스템이 사용자의 요구사항을 만족하는 정도
	실행효율성	. 시스템이 최소 실행시간으로 의도한 기능을 수행하는 정도
	기억장치효율성	. 시스템이 기억용량을 최소로 소비하면서 의도한 기능을 수행하는 정도
정확성 (Corectness)	올바른 규격, 일치된 결과 그리고 정확한 결과를 나타내는 소프트웨어 속성	
	출력의 정확성	. 입력범위 및 내용 . 출력범위 및 내용 . 상수확대 . 계산과정 및 값의 정확성 . 범위 밖의 값의 처리내용
상호호환성 (Interoperability)	사양화 된 소프트웨어가 다른 시스템과 상호운용 능력을 지닌 소프트웨어 속성	
	범용성	. 모듈이 현재 사용되는 부분이외에도 널리 유용하게 쓸 수 있는 정도
유연성 (Compliance)	법률이나 동일한 형태의 규칙, 규격, 협정에 대한 유연성을 지닌 소프트웨어 속성	
	기법 및 도구	. 정의된 기법 및 도구를 규칙에 따라 사용하였는지의 여부
	결함사항	. 발견된 결함사항의 처리방법 준수여부
	표준활동준수	. 표준으로 지정된 것을 충실히 따르고 있는 정도
보안성 (Security)	우발적이거나 고장에 의한 것이 아닌 부당한 액세스로 부터 프로그램이나 데이타를 보호하는 소프트웨어 속성이 아닌 부당한 액세스를 방지하는 속성	
	사용통제성	. 시스템이 자료처리 과정과 자료사용을 통제하는 정도
	추적가능성	. 개발과정이나 운용되는 시스템에서 요구사항이 제대로 구현되어 있음을 확인하는 정도
	사용감사성	. 자료처리과정, 자료출력된 결과의 상황을 쉽사리 추적조사할 수 있는 정도
	시험용이성	. 시스템이 구조와 정확성을 시험하는데 소요되는 노력의 정도

2. 신뢰성 평가항목

품 질 특 성	신뢰성(Reliability)	
명시된 조건하에서 명시된 기간, 소프트웨어의 실행 레벨을 유지하기 위한 능력을 만족하는 속성의 집합		
품 질 척 도	평 가 항 목	평 가 내 용
성숙성 (Maturity)	소프트웨어에 잔재하는 장해에 의한 고장을 대처하는 기능의 소프트웨어 속성	
	결함의 수	. 설계, 프로그래밍 단계에서의 결함수 . 유지보수단계에서의 결함수 . 작업유형* . 운용기간
	오류계수	. 시험기간동안 발생되는 고장수
	오류파종	. 파종오류와 총오류수
	입력영역	. 어떤 입력시에 대해 S/W가 고장나지 않고 작동하는 확률
오류허용성 (Error Tolerance)	소프트웨어 장애 또는 사양화 된 인터페이스를 유지하지 않은 경우에 사양화 된 실행수준으로 회복하기 위한 능력과 시간을 가지는 소프트웨어 속성	
	입력자료 허용성	. 입력값의 범위 점검, 입력항목 및 입력자료 점검, 오류통보문의 제시 여부
	온라인시스템의 허용성	. 입력오류 즉시 처리 여부 . 처리하기 전 입력오류 수정기회 여부
	처리허용성	. 제어변수 점검 . 계산절차의 허용범위 점검 . 계산값의 타당성 여부 . 오류수정 및 처리방법 여부
회복성 (Recoverability)	고장시에 소프트웨어의 실행수준을 재확인하여 직접 영향을 받는 데이타를 회복하기 위한 능력과 시간을 가지는 소프트웨어 속성	
	실패간 평균시간 (MTBF)	. 평균실패시간 . 평균보수시간
	고장사이의 시간	. 고장사이의 시간간격
	회복의 방법	. 회복방법의 절차 및 이해성
	추적가능성	. 모듈 추적성 . 논리추구성 . 입력, 출력, 상수 및 데이타 추적성

3. 사용성 평가항목

품 질 특 성	사용성(Usability)	
명시적 또는 암시적으로 사용자가 이용하기 위해 필요한 노력으로 각각의 사용결과에 대한 평가를 나타내는 속성의 집합		
품 질 척 도	평 가 항 목	평 가 내 용
습득성 (Learnability)	소프트웨어의 운용관리, 입출력 방법 등을 습득하기 위한 사용자의 노력을 나타내는 소프트웨어 속성	
	온라인 도움기능	. 온라인 도움말 구조의 명확함 . 온라인 도움말 내용의 충실함 . 온라인 도움말 접근의 명확함
	온라인 튜토리얼 기능	. 제공방식의 수준 . 온라인 튜토리얼 내용의 충실함 . 온라인 튜토리얼 접근의 용이함
	학습성취도	. 특정기능을 처리하는데 걸리는 시간 . 주어진 시간동안의 작업처리율 . 주어진 시간동안의 오류율
운용성 (Operability)	소프트웨어의 운용과 운용관리를 하기 위한 사용자 노력 정도를 나타내는 소프트웨어 속성	
	시스템 작동방법의 일관성	. 색상, 기능키 등의 사용 . 입력방법 . 화면출력 내용 . 입력에 대한 시스템 반응
	오류방지 및 복구 기능	. 오류방지 유형 . 오류복구의 편리함 . 오류메세지의 유용함
	융통적인 시스템 제어기능	. 화면접근의 용이함 . 인터럽트처리기능 . 명령수행의 용이함 . 입력방법의 용이함 . 사용자 인터페이스 특성변경
	시스템 운용정보 제공기능	. 상태정보의 명확함 . 인스트럭션 제공형식의 명확함 . 실행에 대한 시스템 반응
	시스템 설치의 용이 및 구동의 신속함	. 시스템 설치의 단순함 . 시스템 초기화의 편리함 . 시스템 구동시간
이해성 (Understandability)	소프트웨어의 논리적인 개념을 사용자의 이해정도를 나타내는 소프트웨어 속성	
	인터페이스 제공 방식	. 예상되는 사용자 수준에의 적합함 . 제공 대화방식별 평가항목
	사용자 입력 인터페이스	. 입력의 용이함 . 입력화면의 명확함
	사용자 출력 인터페이스	. 정보의 시각적 명확함 . 출력화면 구성의 적절함 . 정보표현의 친숙함

4. 효율성 평가항목

품 질 특 성	효율성(Efficiency)	
명시적인 조건하에서 소프트웨어의 실행레벨과 사용된 자원 양자 간의 관계를 나타내는 소프트웨어 속성의 집합		
품 질 척 도	평 가 항 목	평 가 내 용
실행효율성 (Time behavior)	소프트웨어 기능을 수행할 때 응답시간과 처리시간을 나타내는 속성	
	시스템 환경명세	. 환경명세가 정리되어 있는가
	성능의 목표	. 사용자가 원하는 성능의 목표를 충족시키는가
	응답시간	. 유사한 시스템에서 모은 결과와 일관성이 있는가 . 좋은 경우, 나쁠 경우, 보통의 경우에 대하여 응답시간을 측정 . 예측된 응답시간은 성능목표 범위 내의 값인가
	처리속도	. 처리결과를 얻기 위해 처리속도의 성질
	처리능력	. 단위시간에 처리할 수 있는 작업량의 성질
자원효율성 (Resource behavior)	소프트웨어의 기능을 실행할 때 사용되는 자원의 양과 사용시간을 나타내는 소프트웨어 속성	
	H/W량	. 사용된 H/W 자원이 계획대로의 투입여부 . 계획대로 사용기간과 일치하는가
	S/W량	. 사용된 S/W관련자원이 계획대로 사용 및 투입여부 . 사용기간과 일치하는가
	인적자원	. 기능을 수행하기 위한 인원배치 및 사용기간이 계획과 일치하는가

5. 유지보수성 평가항목

품 질 특 성	유지보수성(Maintainability)	
사양화 된 개정을 처리하기 위해 필요로 하는 노력을 나타내는 속성의 집합		
품 질 척 도	평 가 항 목	평 가 내 용
해석성 (Analyzability)	고장의 원인 분석이나 결함분석 및 개정 가능한 부분의 정도와 사용시간을 나타내는 소프트웨어 속성	
	표현성	. 입출력과 처리논리 및 결과의 표현여부
	단순성	. 기능을 쉽게 이해할 수 있는 정도
	주석기술의 일관함	. 주석표기 방법 . 주석기술 형식
변경성 (Changeability)	개정을 위한 장애제거나 환경변경에 필요한 노력을 나타내는 소프트웨어 속성	
	프로그램 구조의 단순함	. 기능의 중복성 배제 . 모듈의 입출구 . 전역변수 사용 . 프로그램의 크기
	프로그램의 구조화 정도	. 구조적 언어사용 . 제어흐름의 복잡함 . 코드의 복잡함
안정성 (Stability)	개정에 의해 예기되는 효과의 포함성을 나타내는 속성	
	모듈간의 결합도	. 하나의 모듈에서 호출되는 모듈의 수 . 해당기능을 수정하기 위한 모듈의 역할 기능이 분명한가
	모듈간의 중립도	. 모듈간의 관련성 여부 . 모듈의 기능은 몇 종류 . 원래의 모듈내용대로 환원되는가
	모듈의 크기	. 해당기능을 수행하기에 모듈의 크기가 적절한가(프로그램 라인수)
시험성 (Testability)	소프트웨어의 타당성 검증에 필요한 노력을 나타내는 속성	
	모듈의 주석량	. 각 모듈을 이해하고 보수하기 위한 주석이 충분한가
	주석의 유용함	. 모듈의 목적 및 기능의 설명의 명확함 . 조건, 반복문 등에 대한 주석 . 변수에 대한 주석 . H/W 종속적 코드에 대한 주석
	사용된 언어의 표현력	. 이해하기 쉬운 언어 사용 . 변수명의 적절함 . 코드의 논리적 블럭화 . 인덴테이션의 적절함 . 줄당 라인수

6. 이식성 평가항목

품 질 특 성	이식성(Portability)	
임의의 환경에서 다른 환경으로 소프트웨어를 이식하기 위한 속성의 집합		
품 질 척 도	평 가 항 목	평 가 내 용
환경적응성 (Adaptability)	이식목적에 적합한 작업이나 수단 이외의 것은 이용하지 않고 사양화 된 다른 환경으로 소프트웨어를 이식할 수 있는 소프트웨어 속성	
	확장성	. 시스템에 부가적인 기능이나 자료의 저장을 위해 또는 계산능력 증진이 쉽게 이루어지는 정도
	의사교환성	. 시스템이 사용자에게 유용한 출력과 도움말, 오류 통보문이 이해하기 쉽도록 구성된 정도
이식작업성 (Installability)	소프트웨어를 사양화 한 환경에 도입하기 위해 필요한 노력을 나타내는 속성	
	인수시험 (시험적합성)	. 사용자 요구사항을 충분히 시험할 수 있는가
	설치활동의 산출물	. 설치보고서 . 변환보고서 . 최종시험결과의 준비여부
	운용성	. 시스템의 적재, 개시, 실행, 종결하기가 용이한 정도
	교육적합성	. 시스템 및 그 사용법을 설명하기에 노력이 소요되는 정도
일치성 (Conformance)	규격이나 규정에 소프트웨어를 적합시킬 수 있는 속성	
	통신 공통성	. 시스템이 공통적으로 또는 표준으로 사용되는 통신규약이나 모듈을 사용하기 위한 정도
	자료 공통성	. 시스템이 표준이나 공통된 자료형식, 유형, 표현구조를 사용하는 정도
치환성 (Replaceability)	사양화 된 다른 소프트웨어와 치환되어 그 환경에서 사용하기 위한 기회나 노력을 나타내는 속성	
	범용성	. 모듈이 현재 사용되는 부분이외에도 널리 유용하게 쓸 수 있는 정도
	재사용	. 구현된 모듈에 다른 시스템에 적용될 수 있는 정도

제 5 장 결론

본 연구는 소프트웨어의 품질관리를 위해서 객관적인 측정을 위한 품질척도 측정기술에 대하여 연구하였다. 소프트웨어 품질척도 측정을 하기 위하여 먼저 소프트웨어와 소프트웨어 품질 그리고 소프트웨어 품질평가 절차 및 품질척도 측정내용을 정리하였다.

품질척도의 기준이 되는 품질특성은 ISO/IEC 9126의 품질 평가 특성을 근거로 하였다. 이는 기존의 다양한 품질특성요인들을 정리하여, 소프트웨어 품질관리 표준화를 하기 위한 틀이라고 할 수 있다.

본 연구는 ISO(국제표준기구)에서 제시한 품질평가특성을 근거로 하여 품질척도 및 측정내용을 정리하여 보았다.

소프트웨어 품질관리를 위한 품질척도의 표준화 및 측정내용은 과학적이고 객관적인 평가의 근거가 되고 소프트웨어 개발자와 사용자 그리고 유지보수자에게 소프트웨어비용을 절감하고 생산성을 증가시키기 위한 기초적인 자료로 이용된다.

본 연구의 내용은 실질적으로 검증이 되지 않은 상태이며 이는 앞으로의 연구과제라 생각된다.

앞으로 평가내용을 보완하여 평가 모듈의 양식을 일반화하고 개량화할 수 있는 측정방법을 연구하여 사용이 용이하고 합리적인 기법을 만들기 위해서 연구해야 된다.

- 참 고 문 헌 -

1. ISO/IEC 9126, "Software product Evalution - Quality characteristics and Guidlines for thieruse", 1991.
2. ISO/IEC JTCI/SC7/WG6, "Guide for Measurement, Rating and Assessment 1 : Developer's Guide", 1992.
3. ISO/IEC JTCI/SC7/WG6, "Guide for Measurement, Rating and Assessment 2 : Buyer's Guide", 1992.
4. ISO/IEC JTCI/SC7/WG6, "Evaluator's Guide", 1992.
5. 나관상 외2인, "프로그램 품질평가 기술에 관한 연구", 품질보증 제6권 제2호, 1993. 8.
6. 송재형, "소프트웨어 품질보증 계획과 적용", 정보과학회지 제8권 제4호, 1990. 8.
7. 양해술, 정호원, "소프트웨어 공학과 품질관리 방법론의 동향", 정보처리 제1권 제1호, 1994. 3.
8. 이경환, "소프트웨어 품질과리 기술의 현황과 전망", 정보과학회지 제8권 제4호, 1990. 8.
9. 이주헌, "소프트웨어 공학론", 법영사, 1993.