

소프트웨어 신뢰도 품질측정 방안에 관한 연구

임광현*, 류광선**, 손호선***, 류근호****

The Study of quality measurement plan for software reliability.

Kwang-Hyeon Lim*, Kwang-Sun Ryu**, Ho-Sun Shon***, Keun-Ho Ryu****

요 약

소프트웨어 품질 평가를 위한 국제 표준인 ISO/IEC 9126에서는 소프트웨어의 품질을 이루는 품질 특성과 부 특성들을 정의하고 있다. 또한 이 표준에서는 신뢰성의 부특성인 성숙성, 오류허용성, 복구성, 준수성에 따라 품질 평가 항목이 매트릭스로 제시되어져 있다. 그러나 제시되어 있는 이 신뢰성 평가 항목은 수리적인 어려움과 데이터 수집에 따르는 문제점으로 인해 적용하지 못하고 있는 실정이다. 이러한 문제점의 해결을 위해 본 논문에서는 품질 속성별로 구체적인 테스트 기법을 개발하여 품질을 측정 할 수 있는 방안을 소개한다. 즉, 가장 일반적으로 사용되는 개발 방법인 생명주기별 테스트 활동과 기법을 정리하여 지침으로 활용할 수 있도록 구체적인 가이드라인을 제시한다. 구체적인 방법은 손쉽게 활용할 수 있는 체크리스트 등을 이용하여 해당 소프트웨어의 특성상 가장 중요한 품질특성 순으로 가중치를 부여하고 세부 평가항목에 대해 활용 가능한 테스트 기법과 체크포인트를 적용함으로써 품질결과를 도출한다. 이러한 방법은 소프트웨어의 새로운 신뢰도 품질측정 방안으로써 ISO/IEC 9126의 신뢰성 평가항목이 내포하는 문제점을 개선하였다.

▶ Keywords : 소프트웨어 신뢰도, ISO/IEC 9126, 품질특성, 매트릭

Abstract

International standard for software quality evaluation, ISO/IEC 9126 defines the quality characteristics and sub-characteristics consisting of quality of the software. Also, in this standard,

• 제1저자 : 임광현 • 교신저자 : 류근호

• 투고일 : 2012. 08. 31, 심사일 : 2012. 10. 10, 게재확정일 : 2012. 10. 23.

* 충북대학교 컴퓨터과학과(Dept. of Computer Science, Chungbuk National University, Korea)

** 충북대학교 컴퓨터과학과(Dept. of Computer Science, Chungbuk National University, Korea)

*** 충북대학교 데이터베이스/바이오인포메틱스 연구실 (Database/Bioinformation Laboratory, Chungbuk National University)

**** 충북대학교 데이터베이스/바이오인포메틱스 연구실 (Database/Bioinformation Laboratory, Chungbuk National University)

※ 이 연구는 충북대학교 연구년 중에 수행하였음

the quality assessment items are presented in metrics based on maturity, fault tolerance, recoverability and compliance which are sub-characteristics of reliability. However, this reliability assessment items which are presented are not applied now due to the mathematical difficulties and the problems caused by collection of data. In order to resolve these problems, this study introduces measures to develop a specific test technique by quality attributes and to measure the quality. In other words, the detailed guidelines are presented to be utilized as a standard by summarizing test activities and techniques by each development life the most commonly used. The SW quality results are derived by assigning weights to software in order of the most important quality characteristics of respective software using checklist that can be easily utilized and applying test technique that can be utilized for detailed evaluation items and checklist. This improves problems that include reliability assessment item of ISO/IEC 9126 as the measures to measure new reliability quality of software.

▶ Keywords : Software Reliability, ISO/IEC9126, Quality Characteristics, Metrics

I. 서 론

소프트웨어의 품질특성 중 신뢰도 보장을 위해서 효율적인 품질측정을 위한 테스트 방안이 요구되고 있다. 이는 소프트웨어 유형별 주요 품질특성을 알고, 그 품질 특성을 확인할 수 있는 구체적인 테스트 방법을 안다면, 무엇을 어떻게 테스트해야 하는지 파악하고 쉽게 접근할 수 있다.

먼저, 다양한 종류, 다양한 도메인의 소프트웨어들의 분류가 나오고 각각 어떤 품질 속성이 최우선적으로 보장되어야 하는가를 알아야 한다. 둘째, 특정 품질속성을 검증하기 위해 어떤 테스트 활동을 수행할 것인가에 대한 구체적인 전략 기법, 테스트 활동 지침에 대한 정보가 필요하다. 즉, '소프트웨어-품질속성' 혹은 '품질속성-검증기법'에 대한 정보를 알 수 있다면 특정 소프트웨어에 대한 품질을 확인하고자 할 때 해당 소프트웨어에 있어서 가장 중요한 품질 속성과 그 속성들을 검증하기 위한 테스트 방법들을 도출하여 이를 기반으로 구체적인 테스트 활동을 수행할 수 있게 된다[1].

본 논문에서는 위에서 언급한 '품질속성-검증기법'에 대해 중점적으로 연구하였다. 특히 소프트웨어의 품질을 결정하는 여러 가지 특성 중에서도 신뢰성 품질특성만을 고려하여 연구하였다. 본 연구를 위해 소프트웨어 품질 평가를 위한 국제 표준인 ISO/IEC 9126을 근간으로 한 구체적인 품질검증 방법들을 참고하였다. 하지만 이들을 자세히 살펴보면 실제 수행에 있어서 시간과 인력은 많이 들고 중요도는 낮은 활동들로 구성되어 있어서 자원이 부족하거나 시간이 부족한 경우

테일러링이 필요한 부분이 많다. 더욱이 실제 적용 시 정작 중요한 구체적인 실행 가이드가 약하다는 것이다[2].

이러한 문제점들의 개선을 위해 본 연구에서는 두 가지를 제시하였다. 먼저 소프트웨어의 품질특성 중 신뢰성 특성의 품질측정 모듈의 개선된 측정 방법을 제시한다. 두 번째로는 소프트웨어 신뢰성 분야의 품질특성의 중요도와 테스트 활동 실행 가능성 등을 고려하여, 소프트웨어 개발 현장에서 개발자나 전문적인 테스터가 실제적으로 활용할 수 있는 가이드라인을 제공한다[3]. 이는 손쉽게 활용할 수 있는 체크리스트 등을 이용하여 해당 소프트웨어의 특성상 가장 중요한 품질특성 순으로 가중치를 부여하고 세부 평가항목에 대해 활용 가능한 테스트 기법과 체크포인트를 제시하며, 평가 결과를 입력하면 최종 품질 수준을 계산하여 도출할 수 있다.

실제로 특정 소프트웨어에 대해 테스트를 하고자 할 때, 가장 중점적으로 확인해야 할 품질특성이 무엇인가를 파악한다면, 본 논문에서 제안하는 가이드라인을 활용하여 테스트 활동에 적용할 수 있을 것이다. 또한 제시한 체크리스트는 '현실성'과 '효율성'에 중점을 두어 우선순위에 따라 테일러링이 가능하므로 개발방법론 상에서 품질측정을 위한 별도의 과정을 포함시켜두면 개발 과정에서 소프트웨어의 신뢰도를 상시 보장할 수가 있게 된다. 이러한 방법은 ISO/IEC 9126에서 제시되어 있는 신뢰성 평가 항목이 수리적인 복잡함과 데이터 수집에 따르는 문제점으로 인해 실제 적용 시 발생하는 문제점을 개선할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 ISO/IEC 9126의 소프트웨어 품질특성과 테스트 방법론을 알아보고 여러 가지 품질특성 지침들을 살펴봄으로써 문제점

과 개선방향에 대해 논의한다. 3장에서는 2장에서 도출된 문제점을 해결하기 위한 개선된 모듈을 제시하여 그것의 상세내용과 적용방법을 자세히 살펴 볼 것이다. 4장에서는 3장에서 정의된 모듈을 바탕으로 실험 평가를 통해 우수성을 검증한다. 마지막으로 5장에서는 더 나은 활용방안을 살펴보고 향후 연구 과제를 제시하고 글을 맺는다.

II. 관련 연구

1. ISO/IEC 9126

ISO/IEC 9126은 소프트웨어 품질 특성과 척도에 관한 지침을 제공하고 있으며 모델의 구성은 다음과 같다.

표 1. ISO/IEC 9126 모델 구성
Table 1. ISO/IEC 9126 Model

Part	구성
Part 1	Quality Model
Part 2	External Metrics
Part 3	Internal Metrics
Part 4	Quality in use Metrics

그 중 ISO/IEC9126의 소프트웨어 품질 속성을 표 2와 같이 여섯 가지 특성으로 분류할 수 있으며, 이 품질 특성은 다시 부특성들로 세분하여 이에 따른 세부 매트릭(평가항목)을 제시하고 있다[4].

표 2. ISO/IEC 9126 소프트웨어 품질 모델
Table 2. ISO/IEC 9126 Software Quality Model

품질특성	부특성
기능성	적합성, 정확성, 상호운용성, 보안성, 준수성
신뢰성	성숙성, 결함허용성, 복구성, 준수성
사용성	이해성, 학습성, 운용성, 선호도, 준수성
효율성	시간반응성, 자원효율성, 준수성
유지보수성	분석성, 변경성, 안전성, 시험성, 준수성
이식성	적용성, 설치성, 공존성, 대체성, 준수성

표 1에서 보듯이 Part 1의 품질 모델과 관련하여 외부 품질, 내부 품질이라는 개념을 포함하고 있는데, 외부 품질

(External Quality)이란 제품이 규정된 조건에서 사용될 때 요구사항을 만족시키는 정도를 말하며, 내부 품질(Internal Quality)이란 제품이 규정된 조건에서 요구사항을 만족하기 위해 제품의 설계와 실제 코드가 어떤 품질을 갖는가 하는 것을 말한다.

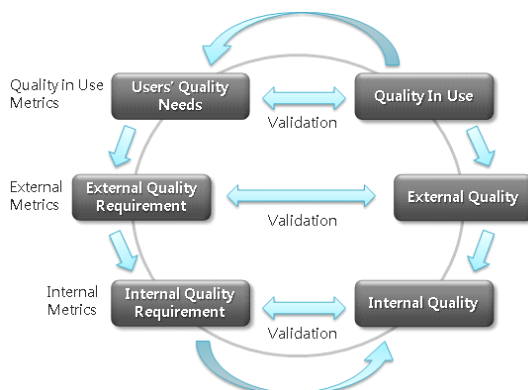


그림 1. ISO/IEC 9126 소프트웨어 품질모델의 구조
Fig. 1. Structure of ISO/IEC 9126 Software Quality Model

품질 모델 내의 여러 요소들을 그림 1에서와 같이 소프트웨어의 생명주기를 놓고 볼 때 내외부적 품질과 사용 중 품질 등이 어떤 의미를 갖는가 알 수 있다. 설계 및 코딩의 검토 단계에서는 내부 품질을 측정하고, 대상 소프트웨어가 완성된 후에는 외부 품질 평가를 통해 품질 요구사항이 만족되는가를 확인한다. 사용 중 품질이란 제품이 특정 환경에서 사용될 때 사용자의 작업 효율성, 생산성, 안정성, 만족도 등 사용자의 요구를 충족시키는 정도를 말한다. 사용 중 품질에 대한 평가는 고객에

게 인도된 후, 향후 수정 작업이나 유사 프로젝트를 위하여 실제 현장에서 사용자에게 의해 평가되는 것이다[3][5].

2. ISO/IEC 9126기반 품질 측정 방법 연구

최근 들어 ISO/IEC9126을 기반으로 하여 소프트웨어 품질측정을 위한 구체적인 방법들이 많이 개발되었다. 정보통신산업진흥원(NIPA)의 '소프트웨어 기술성 평가기준 해설서'나, 한국정보통신기술협회(TTA)의 '웹기반 소프트웨어 품질 평가 지침'에 구체적인 품질 측정 방법들이 제시되어 있다. 보통 ISO/IEC9126 및 ISO/IEC12119 기반의 심사기준인 평가모듈을 근간으로 하여 국내에 맞게 바꾸었다[5][6].

위 NIPA와 TTA 평가 모델 둘 다 ISO/IEC 9126의 품질모델을 근간하고 하고 있으며 본 논문에서는 이 두 가지에 대해 각 품질측정 방법을 비교하여 개선점을 도출하고자 한

다. 즉, 이 두 가지 평가모듈을 분석하여 특정 소프트웨어마다 필요한 요소 들을 중심으로 효율적으로 수행할 수 있는 품질 특성별 테스트방안을 제시하고자 한다. 이를 통해 테스트 프로세서가 약한 조직이나 소규모의 프로젝트에서도 쉽게 활용할 수 있는 실제적이고 효율적인 테스트 가이드라인을 제시하고자 한다.

2.1 NIPA의 소프트웨어 품질측정 방법

NIPA는 '소프트웨어 기술성 평가기준 해설서'에 소프트웨어 품질 측정 방법을 정의하고 있으며 문서의 구성은 다음과 같다[7][8].

* 품질속성 (예 : 기능성, 신뢰성 등)

1. 부특성이름 (예 : 기능성 아래 상호운용성, 보안성 등)

1.1 항목 정의

1.1 측정항목 및 측정 방법

1.1.1 데이터 항목

- 데이터 항목 (예 : 업무 서식 수, 서식작성 기능 수 등)
- 평가대상 (예 : 관련문서, 프로그램 등)
- 측정유형 (예 : Number, Y/N/NA 등)
 - 측정요소 및 측정방법

1.1.1 측정요소 및 측정방법

- **세부평가항목** (예 : 업무형태 적합성)별 계산식, 설명, 결과값 유형, 예
 - 계산식 (예시)

- 결함 회피율 : $1 - \min(\text{발견된 결함 수} / \text{단위 운용시간})$
- 다운 회피율 : $1 - \frac{\text{소프트웨어의 재시동이 필요한 결함의 수}}{\text{발견된 결함 수}}$

1.1 적용절차

그림 2. 소프트웨어 기술성 평가기준 해설서
Fig. 2. Document of Software Technological ratings

그림 2에서 1. '부특성 이름'이 문서의 기본단위로서, '세부평가항목'을 기본단위로 하는 TTA의 평가 모델에 비해 분량이 적고 간략하다는 특징이 있다. 또한 모든 품질 특성에 대해 일괄적으로 적용하지 않고 평가요소별로 가중치를 부여하고 있

다[5,7]. 표 3은 그림 2의 문서를 바탕으로 ISO/IEC9126-3에서 제시하는 가중치를 적용한 예를 보여주고 있으며 전체 품질특성에서 신뢰성 특성 부분을 보여준다.

표 3. 신뢰도 품질특성의 중요도
Table 3. Importance of Reliability Quality Characteristics

특성	부특성	가중치
신뢰성	문제해결성	L
	결함발생율	L
	다운고장 회피성	L
	오조작 방지성	L
	데이터 회복성	H
	복구성	H

* 가중치(1~9) - Low : 1~3, Medium : 4~6, High : 7~9

모든 평가 요소를 평가하기 어려운 경우를 위해 반드시 반영해야할 필수 항목과 선택항목을 구분지어 표시하고 있다. 이는 품질특성을 테일러링하여 특정 소프트웨어 개발이나 프로젝트에 맞추어 적용가능하다.

표 4. 품질특성 및 부특성의 필수 선택 여부
Table 4. Required or optional Quality Characteristics

특성	필수항목	선택항목
신뢰성	<ul style="list-style-type: none"> ▪ 결함발생율 ▪ 결함회피율 ▪ 다운/고장 회피성 ▪ 오조작 방지성 	<ul style="list-style-type: none"> ▪ 데이터 회복성 - 사무용 S/W, - 기업용 S/W

2.2 TTA의 소프트웨어 품질평가 지침

TTA에서는 '웹 기반 소프트웨어 품질평가 지침'을 제공하고 있다. '품질특성 - 부특성 - 세부평가항목'으로 연결되며 평가항목 단위(매트릭(metrics))로 제공해준다. 또한 평가항목 단위로 상세한 해설을 제공하는데, 이를 '품질평가 모듈'이라고 하며 본 문서의 구성은 다음과 같다[9][10].

* 품질속성(예 : 기능성, 신뢰성 등)
 * 부특성 이름 (예 : 기능성 아래 상호운용성, 보안성 등)

1. 평가항목 (매트릭)

1.1 개요

1.1.1 매트릭의 개념

1.1.1.1 특정 목적

1.1.1.1 매트릭의 범주 (예 : 기능성 - 적합성 - 기능구현 완전성)

1.1.1.1 용어설명 - '해당사항 없음'이 많음

1.1 매트릭의 적용

1.1.1 적용대상 및 필요자원

1.1.1.1 기법 - '해당사항 없음'이 많음

1.1.1.1 적용 시 고려사항

1.1 참조문서 (예 : ISO/IEC 9126-2 8.1.1 등)

1.1 매트릭

1.1.1.1 측정 항목

- 측정 항목 (예 : 업무 서식 수, 서식작성 가능 수 등)
- 평가 대상 (예 : 관련문서, 프로그램 등)
- 측정 유형 (예 : Number, Y/N/NA 등)

1.1.1.1 측정 방법

1.1.1.1 계산식 (예시)

$$-B = \frac{d}{n} (SuccessTC_i / TotalTC_i)$$

1.1 적용절차

그림 3. 품질평가 모듈의 품질검사표 구성
 Fig. 3. Inspection Table of Quality Evaluation Module

그림 3에서 보는 바와 같이 NIPA의 문서는 품질의 '부특성' 단위인 반면 TTA는 그 아래 있는 부특성별 '평가항목(매트릭)'단위이기 때문에 훨씬 상세하다. 그러나 TTA의 문서는 문서를 구성하는 항목이 지나치게 많은 감이 있다. 항목이 많은 만큼 내용이 각기 다르거나 혹은 중복성이 없거나 중요한 정보를 제공하지 않는 한 필요 없는 항목이 많게 느껴져 모델을 활용하는데 되레 역효과를 줄 수가 있다. 예를 들어 그림 3의 상단부분에 있는 매트릭의 범주 항목은 문서 단위와 동일한 '품질특성 - 부특성 - 세부평가항목'을 그대로 반복하므로 불필요한 정보이다. 바로 아래의 '용어설명'도 해당 내용이 없는 경우가 많고 문서의 구성요소로서 항목정의만 되어 있다.

TTA의 품질평가 지침에서 신뢰성 특성 만 정리하면 표 5

와 같다. 기법 항목을 보면 '없음' 또는 '문서검토'라고 표기되어 있는데 이것만 가지고는 어떤 산출물을 말하는지 알기 어렵다.

표 5. TTA 품질 평가 지침 신뢰성 특성 요약
 Table 5. Reliability Characteristics of TTA Quality Evaluation Module

특성	부특성	평가항목(매트릭)명	기법
신뢰성	성숙성	문제해결이력정보제공	문서검토
		문제 해결을	없음
		결함 회피율	없음
	오류허용성	다운 회피율	없음
		고장 회피율	없음
		오조작 회피율	체크리스트
	회복성	데이터 회복정보 제공	문서검토
		데이터 회복을	없음
		복구 가능을	없음
	준수성	복구 효과율	없음
		신뢰성 표준준수 정보 제공	문서검토
		신뢰성 표준 준수율	없음

2.3 그 외 소프트웨어 품질평가 방법론

소프트웨어 품질과 관련된 평가 방법으로는 위에서 제시한 NIPA, TTA 외에도 한국표준협회나 한국정보통신기술협회에서 제정한 내용을 들 수 있다.

표 6. 소프트웨어 품질관련 KS 및 KICS 표준
 Table 6. KS and KICS Standard for Software Quality

표준번호	제목
KS X 2216-1	정보기술 - 소프트웨어 제품 품질 제1부 : 품질 모델 (2001.04)
KS X 2223-2	정보기술 - 소프트웨어 제품 평가 제2부 : 계획과 유지보수 (2001.04)
.....

(a) KS 표준

표준번호	제목
KICS.IS-12207	소프트웨어 생명주기 공정 표준 - ISO/IEC 12207 (1996.12)
KICS.IS-8402	소프트웨어 품질의 측정, 평점 및 심사를 위한 기술지원서 (1994.12)
.....
KICS.IS-9126	소프트웨어 패키지의 품질요구사항과 시험에 관한 기술지원서 (1996.05)

(b) KICS 표준

위 표 6에서는 소프트웨어 품질과 관련된 국내 표준에 대한 내용이다. 대체적으로 ISO/IEC 9126을 기반으로 하여 국내 표준을 제정하였다. 이 표준 내에 품질을 측정하는 품질 평가방법들이 상세히 제시되어 있다. 이외에도 국외의 품질특정방법 체계를 간단히 살펴보면 표 7과 같다.

표 7. 소프트웨어 품질관련 KS 및 KICS 표준
Table 7. KS and KICS Standard for Software Quality

국가명	품질측정 방법
미국	민간인증 기관에게 위임 (VeriTest, NSTL, NTS/XXCAL)
독일	GGS(독일 소프트웨어 산업협회) ISO/IEC12119 기반 인증
덴마크	DELTA 사, ISO/IEC 9126 기반

3. 소프트웨어 품질측정 모듈 개선 방향

앞 절의 관련연구에서는 소프트웨어의 품질특성을 정의하는 표준인 ISO/IEC 9126과 이를 기본으로 하여 NIPA와 TTA에서 제시하는 구체적인 품질평가 방법과 품질평가 모델을 각각 살펴보았다.

본 논문에서는 위 세 가지 모델을 참조하고 각 특징의 장점을 취하여 꼭 필요한 정보들만 담은 간략하고 실제적인 품질 측정 모듈을 제안한다. 또한 실제 적용을 위해 개발단계와 인수단계에서 사용할 수 있는 평가 체크리스트를 제공하여 활용성을 높이고자 한다.

관련 연구 내용을 바탕으로 개선 방향에 대해 정리하면 다음과 같다.

첫째, NIPA에서 볼 수 있는 것처럼 품질특성 및 부특성에 대하여 가중치를 부여하고 필수/선택 여부를 정할 수 있게 한다. 이는 소프트웨어의 분류에 따라 중요한 품질 특성이 다르고, 프로젝트의 가용 자원을 따라 테일러링을 해야 하기 때문이다.

둘째, TTA 평가 모듈의 품질검사표와 같이 표로 압축된 형태를 취하여 활용성을 높인다. 그러기 위해서는 개념이나 용어, 측정 방법의 복잡한 수식 등 필요 없는 요소들은 제거하고 평가항목에 대한 주요 체크포인트, 계산법 등이 간략한 형태로 정리되어야 한다.

셋째, 테스트 활동을 수행함에 있어서 실제적으로 유용한 정보들을 제공해야 한다. TTA의 '기법' 항목은 실제적으로 테스트를 수행함에 있어서 꼭 필요한 정보이기 하나 그 내용이 매우 빈약하다. 따라서 테스트 기법에 대한 정보를 많이 보강해야 하고, 특히 문서 검토가 필요한 경우 어떤 문서인지 알아

야 한다. 따라서 소프트웨어 개발 프로젝트에서 일반적으로 사용되는 테스트 기법들과 중점 체크포인트를 일목요연하게 정리하여 참고할 수 있게 하여 산출물 리스트를 제공하고 각 평가항목을 점검할 수 있는 구체적인 문서를 명시해야 한다.

마지막으로, 평가 틀 형태로 통합하며 이를 위해 필요한 항목과 함수, 수식을 제공해야 한다. 평가를 위해서는 절대적인 결과 (0~1 사이의 숫자, Yes/No 등) 외에도 목표품질을 지정하고 이에 대비하여 품질이 합격선인지 아닌지를 판단할 수 있어야 한다.

위 내용을 종합하여 볼 때, 본 논문에서는 '효율성'과 '현실성'을 목표로 한다. 즉, 소규모의 프로젝트나 테스트 인력이 적은 경우 또는 문서작업 등의 이중적인 부담을 최소화하고자 할 때, 유용하게 사용할 수 있는 테스트 수행 가이드라인과 이를 구체화 하기위한 도구인 테스트 체크리스트와 품질활동 체크리스트를 제공하는 것이 목적이다.

III. 개선된 신뢰성 품질특성 세부내용 및 적용방안

1. 신뢰성 품질 측정 모듈의 구성

소프트웨어의 품질특성을 테스트하기 위해서는 사업수행에 필요한 모든 품질특성에 대해 테스트 방안을 제시하여야 하나 본 논문에서는 소프트웨어의 신뢰도를 결정짓는 품질 특성 중에 신뢰성 측면만 고려하여 품질측정 모듈을 제안하고자 한다. 즉, 본 논문에서 제안하는 품질 측정 모듈은 '신뢰성 품질 측정 모듈 상세 정의'와 이를 가장 단순하고 직관적인 형태로 요약한 체크리스트 형태의 테스트 체크리스트와 품질활동 체크리스트로 구성된다.

본 논문에서 제안하는 신뢰성 부문 품질 측정 모듈의 구조는 그림 4와 같다. 다음 절에는 이 신뢰성 부문 품질 측정 모듈을 상세히 설명한다.

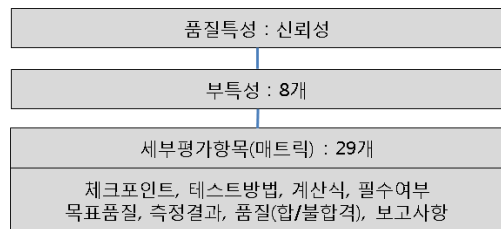


그림 4. 신뢰성 품질측정 모듈의 구조
Fig. 4. Reliability Quality Evaluation Module

2. 신뢰성 품질 측정 모듈 상세 설명

이 장에서는 ISO/IEC 9126품질 모델을 바탕으로 하여 소프트웨어 제품의 신뢰성을 측정할 수 있는 품질 평가 모듈을 상세히 설명한다. 관련연구에서 살펴본 NIPA와 TTA의 문서로부터 개선점을 도출하여 본 논문의 목적에 맞게 수정하였다. 그림 4에서와 같이 신뢰성 품질특성에서 부특성을 8개 도출하고 세부평가항목(매트릭)을 29개 도출하여 새로운 평가 도구로 활용 할 수 있도록 재구성하였다. 표 6은 개선된 품질 평가 모듈 신뢰성 특성 요약을 보여준다.

표 8. 개선된 품질 평가 모듈 신뢰성 특성 요약
Table 8. Improved Reliability Characteristics of Quality Evaluation Module

특성	부특성	평가항목(매트릭)명	필수
신뢰성	성숙성	문제해결이력정보제공	
		문제해결을	필수
		사용환경시험여부	
		사용환경시험의적절성	
	오류허용성	결함회피율	필수
		다운회피율	필수
		고장회피율	필수
	회복성	오조작회피율	
		데이터회복정보제공	
		데이터회복을	필수
	준수성	복구기능을	
		신뢰성수준정보제공	필수
	문제진단/해결성	신뢰성수준준수를	필수
		진단기능정보제공	필수
		진단기능지원율	필수
		진단구분을	필수
	결함발생율	결함발생빈도	필수
		문제해결정보제공	필수
		문제해결구현율	필수
	오조작방지성	예비지식정보제공	
가능이해도		필수	
인터페이스이해도		필수	
입출력데이터이해도		필수	
내용일관성			
복구성	사용자안내성		
	오류복구용이성	필수	
	메시지이해용이성	필수	
	운영절차조정정보제공	필수	
	진행상태 파악가능성	필수	

표 8의 재구성된 개선된 품질 평가 모듈 신뢰성 특성 요약에 따라 품질특성 모듈을 살펴보면 '품질특성 - 부특성 - 평가항목(매트릭)명 - 필수구분'의 구성으로 되어 있음을 알수 있

다. 각각의 특성명과 부특성 명은 그것의 정의를 포함하고 있고 정의 외에도 지침이나 간단한 설명을 붙여서 부특성의 범위를 명확하게 한정하고 있다. 평가항목(매트릭)명은 다시 개념, 테스트방법, 계산식, 필수여부 등 세부적인 항목으로 나누었다. 그리고 마지막으로 필수 여부를 부여하여 테일러링이나 프로젝트의 규모에 맞는 평가활용이 가능하도록 정의하였다. 아래 그림 5는 이 논문에서 제안한 개선된 신뢰성 품질 특성 품질평가 모듈의 구조를 일반화 하였다.

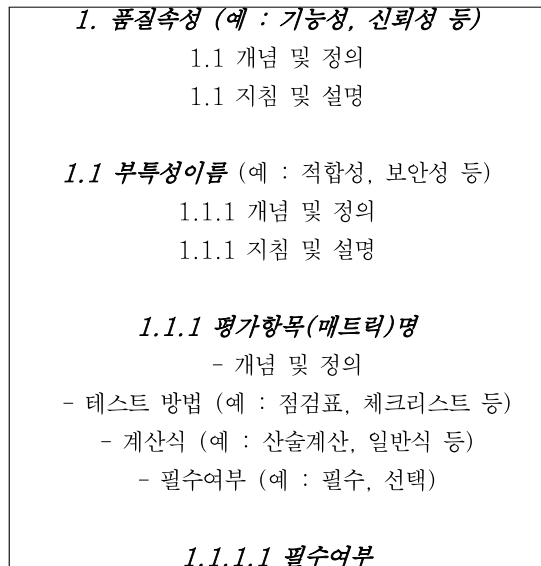


그림 5. 개선된 신뢰성 품질특성 품질평가 모듈의 구조
Fig. 5. Improved Quality Evaluation Module for Reliability Quality Characteristics

위의 일반화된 구조에 맞게 29개의 신뢰성 평가항목을 모두 정의 하였으며 그 중의 일부를 아래 그림 6의 예시와 같이 정의하여 사용할 수가 있다. 예시로는 '신뢰성 - 결함발생율 - 결함발생빈도 - 필수'의 조합을 가지고 측정모듈을 사용할 경우의 측정 상세 예시를 보여주고 있다.

1. 신뢰성

1.1 소프트웨어가 특정 조건에서 사용될 때 규정된 성능수준을 유지하거나 사용자로 하여금 오류를 방지할 수 있도록 하는 소프트웨어 제품의 능력을 의미한다.

1.1 결함발생율

1.1.1 동일한 시간 간격 동안의 결함수를 기반으로 과거의 결함율을 이용하여 현재의 결함율을 구함으로써 앞으로의 결함발생을 보다 정확히 예측하고자 하기 위해 사용되는 개념

1.1.1 적용된 유명한 모델로는 Schneidewind 모델이 있다.
(중략)

1.1.1 결함발생 빈도

- 개념 및 정의 : 과거에 비해 얼마나 결함이 자주 발생하는 가를 나타낸다.
- 테스트 방법 : 과거 결함율 vs 현재 결함율
- 계산식 : ((결함율 계산식 결과) - (현재 결함율 계산식 결과)) / (현재 결함율 계산식 결과) * 100
- 필수여부 : 필수

1.1.1.1 필수

그림 6. 개선된 신뢰성 측정 모듈의 적용 예시 ("신뢰성" > "결함발생율" 의 예시)

Fig. 6. Example for Improved Quality Measurement Module

IV. 적용 및 평가

1. 신뢰성 품질 측정 모듈 적용

소프트웨어 유형에 따른 품질 특성을 선별하고 이 품질 특성을 검증할 수 있는 테스트 전략을 짜기 위해서는 '소프트웨어 분류 - 주요 품질 속성 - 테스트 기법'간의 연관성이 분석되어야 한다. 즉, 개발하거나 테스트하고자 하는 소프트웨어가 어떤 유형에 속하는지 파악하고 해당 유형의 소프트웨어들에게 중요한 품질 특성이 무엇인가를 알고, 이를 측정할 수 있는 테스트 방법을 알아야 한다.

이를 위해 소프트웨어 분류 체계별 품질 특성 매트릭이 개발되고 본 논문이 제안한 것처럼 품질 특성별 테스트 방안을 제시하는 품질 평가 모듈을 활용하면 효과적으로 테스트 활동을 수행할 수 있다.

본 논문의 특징인 '효율성'을 최대화하여 평가항목을 선별할 수 있도록 하였으므로 프로젝트가 목표로 하는 소프트웨어의 성격에 따라, 시간적, 인적, 환경적 테스트 자원에 따라 최적의 테스트 과정을 테일러링 할 수 있다.

특정 소프트웨어가 속하는 유형 그룹에서 어떤 품질 특성이 중요한지 파악하고 (예를 들어 그림 6의 경우 소스코드의 '신뢰성'과 그 중에서도 '결함발생율'이 중요한 품질 요소로 판단되었다면) 표 9의 신뢰성 품질 측정 모듈을 활용하여 테스트 방법을 알 수 있다. 표 9에서 제시된 각 항목들의 설명이라든가 평가 항목들은, 테스트 방법 등은 관련 연구에서 살펴본 NIPA 나 TTA 의 자료에서 인용하거나 신뢰도를 높이기 위해 본 논문에서 새롭게 추가하여 구성하였다. 이는 물론 장량적으로 검증이 가능한 식 들이며 프로젝트의 성격, 테스트

표 9. '신뢰성' - '결함발생율' 품질 측정 모듈 예시
Table 9. Quality Measurement Module of 'Reliability' - 'Error Detection Rate'

부특성	가중치	평가항목 (메트릭)명	필수	개념(체크포인트)	테스트 방법	계산식	목표 품질	측정 결과	품질	보고사항
신뢰성	H	결함발생 빈도	0	과거에 비해 얼마나 또는 어떤 경우에 결함이 자주 발생하는가? 등	과거 결함율 vs 현재 결함율	((결함율 계산식 결과) - (현재 결함율 계산식 결과)) / (현재 결함율 계산식 결과) * 100	1	x	x	빈도 측정 불가능
		문제해결 정보제공	0	발생한 결함을 조치하기 위한 기본 정보를 제공해주는가? 등	제품설명서 검토 사용자지침서 검토	발생하는 오류의 증상 및 문제해결에 관한 정보 제공	1	x	x	누락되거나 잘못 구현된 정보 항목
		문제해결 구현율	0	발생하는 문제해결 정보에 따라 발생한 문제를 해결할 수 있도록 구현되어 있는가? 등	문제해결 정보 항목 vs 제품기능	각 항목별 테스트케이스 성공을 합 / 평가할 문제해결 항목 수	1	x	x	명확히 구분되지 않는 기능 항목

의 성격 등에 따라 위 표 9는 일부 테일러링하여 사용가능하다. 물론 이를 바탕으로 만들어진 그림 7의 각종 체크리스트들도 신뢰도 평가측정 내용에 따라 테일러링이 가능하다.

표 9에서 보이는 것처럼 실제 프로그램 개발상황에서 품질 측정 모듈을 활용한 테스트 전략을 수립하고 수행하는 절차는 다음과 같다.

- 1단계 : 개발하고자 하는 소프트웨어의 유형을 파악한다.
- 2단계 : 해당 유형의 소프트웨어에 중요한 품질 특성 / 부 특성, 세부평가항목을 파악한다.
- 3단계 : 프로젝트와 소프트웨어의 성격, 시간적, 인적 자원, 고객의 품질 요구사항을 토대로 최종 검토하여 개발 단계별 테스트 전략을 수립한다.
- 4단계 : 제안된 품질 측정 모델과 테스트 기법 그리고 다음 절에서 소개할 품질측정모듈의 응용인 체크리스트 등을 활용하여 품질 측정 결과를 얻는다. (필요 시 이 과정을 반복할 수 있다.)
- 5단계 : 품질측정 결과를 보고하고 이를 기초로 하여 개선 활동을 수행한다.

2. 신뢰성 품질 측정 모듈 응용

본 논문에서 소개 하는 개선된 품질측정 모듈을 응용한 형태로 품질을 검증 할 수 있는 여러 가지 체크리스트를 제작할 수도 있다. 이 응용 체크리스트는 설계 및 구현단계에서 사용 가능한 '화면구성 일관성 체크리스트', 개발 단계에서 사용할 수 있는 '소스코드 체크리스트', 그리고 테스트 단계에서의 '단위 테스트 체크리스트' 마지막으로 인수단계에서 QA에 의해 작성

되는 '품질보증 담당자 심사 체크리스트'와 '워크스루체크리스트' 등으로 품질측정 모듈과 범위가 확장 가능하다.

결과유형	결과 내용	
적합 Y	테스트 결과가 정상인 경우	
부적합 N	테스트 결과 기밀에 영향을 미치고 있는 경우	
해당없음 N/A	테스트 항목이 해당사항없는 경우	
분류	테스트 케이스	결과
검역	1) 검색 상행로 크기가 및 위치에 일관성이 있는가?	Y
	2) 검색상에 사용한 폰트 사이즈 및 컬러에 규격이 있는가?	Y
	3) 검색기능 및 결과의 정확이 보장되는가?	Y
개시안	1) 검색문의 크기가 및 스타일에 일관성이 있는가?	Y
	2) 관련 버튼의 그룹핑이 잘 되어 있는가? (제시된 이통버튼과 글쓰기관련 버튼)	Y
	3) 볼 수 있는 페이지 수는 적절하게 규정되었는가?	Y
	4) 글수정, 삭제입력 오류시, 페이지 시작점의 크기 및 형태에 일관성이 있는가?	N
이동메시지	1) 입력폼의 앞쪽의 구성요소가 동일한 모양으로 동일한 위치에 있는가? (라디오버튼, 체크박스, 스크롤바 등 일괄 사항 등)	Y

(a) 화면구성일관성 체크리스트

결과유형	결과 내용	
적합 Y	소스코드가 개발표준을 준수한 경우	
부적합 N	소스코드가 개발표준을 위반한 경우	
해당없음 N/A	항목이 해당사항없는 경우	
분류	항목	결과
개발물	1) 프로토타입에서 지정한 표준 개발법을 사용한다	Y
	2) 소스파일은 지정된 디렉토리에 위치한다.	Y
	3) 파일 확장자는 파일명뒤에 적당하다	Y
	4) 파일 확장자는 파일명뒤에 적당하다	Y
	5) 한 파일이 2000바이트를 넘지 않도록한다.	Y
인터페이스	1) 자바소스파일은 하나의 public class 또는 interface로 구성된다	Y

(b) 소스코드체크리스트

결과유형	결과 내용		
응용할 A	데이터의 불일치 또는 해당 기능을 정상적으로 진행할 수 없는 상황		
공용할 B	일부 미미한 결함을 포함하고 있으나 시스템의 다른 기능을 미연하여 처리가 가능한 결함		
단순결함 C	해당 기능을 사용할 수 없는 결함이나 사용자들의 혼동, 불편을 유발시킬 수 있는 결함 (User Friendly)		
권고사항 D	결함을 처리되지 않으나, 사용자들의 편의를 위해 개선되는 것이 바람직하다고 생각되는 사항		
결과유형	결과 내용		
적합 Y	테스트 결과가 정상인 경우		
부적합 N	테스트 결과 기밀에 영향을 미치고 있는 경우		
해당없음 N/A	테스트 항목이 해당사항없는 경우		
개시	테스트 케이스	결과유형	결과
테스트	1) 입력란 값과 다른 데이터가 저장되는 등 동작가능이 정상적으로 수행된다	A	
	2) Primary Key 2이 중복되지 않는다 (이때 동일한 데이터가 또 저장되는 경우)	A	
	3) Primary Key 값에 NULL값이나 공백(Space)이 저장되지 않는다	A	
	4) NULL / NOT NULL / SPACE / BLANK 값이 명확히 구분된다	B	
	5) From - To 범위의식의 데이터(특히, 날짜)의 연속 관계를 확인한다	B	
6) 입력 필드와 then True and False를 적절히 사용한다	D		

(c) 단위테스트 체크리스트

심사단계	개발단계	심사결과	체크대상 건수 : 22 건, 부적합 건수 : 0 건	
체크리스트 ID	구분	대상	체크포인트	결과 (Y/N)
1	프로세스	프로그램개발	프로그램 개발시 개발표준을 준수하였는가?	Y
2	프로세스	프로그램검역	테스트는 테스트 계획서에 기술된 환경에서 수행되었는가?	Y
3	프로세스	프로그램 (데이터) 불일치	테스트 결과는 테스트계획서에 의거 수행되었는가?	Y
4	프로세스	프로그램에서	단위테스트는 지정된 단위테스트체크리스트를 활용하였는가?	Y
5	프로세스	단위테스트	단위테스트 수행 결과에 따라 오류 및 개선사항을 조치하였는가?	Y
6	프로세스	통합테스트	통합테스트는 통합테스트 시나리오에 따라 수행되었는가?	Y

(d) 품질보증담당자 심사 체크리스트

심사단계	개발단계	심사결과	체크대상 건수 : 18건, 부적합 건수 : 0 건	
체크리스트 ID	구분	대상	체크포인트	결과 (Y/N)
1	프로세스	프로그램개발	프로그램 개발시 개발표준을 준수하였는가?	Y
2	프로세스	단위테스트	단위테스트는 지정된 단위테스트체크리스트를 활용하였는가?	Y
3	프로세스	프로그램 (데이터) 불일치	단위테스트 수행 결과에 따라 오류 및 개선사항을 조치하였는가?	Y
4	산출물	프로그램목록	동시동행 및 사용자관리 기능을 수행하는 프로그램이 모두 기재되었는가?	Y
5	사출물	프로그램목록	가능할 경우 프로그램이 가능한 형태로 기재되었는가?	Y

(e) 워크스루 체크리스트

그림 7. 신뢰성 품질 측정 모듈 응용 예시

Fig. 7. Example for Adapted Quality Measurement Module

3. 적용 및 평가

제안한 개선된 신뢰성 측정 모듈을 실제 프로젝트에 적용하는 데는 다소 어려움이 있었다. 우선, 프로젝트의 기간이 보통 3~5개월로 그 결과를 수집하고 도출하는 데는 상당한 시간이 소요되었다. 또한 제안 모듈을 이용하여 테스트 하는 경우와 다른 측정도구를 이용하여 테스트 하는 경우를 하나의 프로젝트에서 각각 적용해 보아야 하므로 똑같은 조건으로 적용테스트를 하는데 많은 제약이 따랐다. 따라서 비교적 적은 투입 MM, 난이도 중하 규모의 LOC, 적당한 수준의 FP를

갖는 업무용 단위 프로그램의 개발을 진행하는 조건에서 테스트가 진행되었다. 본 모듈은 실제 프로젝트에 적용하여 검증해 보았고 이 때, 개선된 모듈을 적용한 경우와 적용을 하지 않은 경우 각각 신뢰도가 어떻게 다르게 나타나는가를 정량적으로 테스트하고 결과를 도출해보았다. 실제 적용하기 위한 프로젝트의 객관적 조건은 표 10과 같다.

표 10. S 프로젝트 적용 환경
Table 10. S Project Environment for Adaptation

속성	주요내용	비고
발주	국내 서울소재 관공서	-
프로젝트	행정 업무 웹 어플리케이션 개발	난이도 : 중하
MM	2MM (5~6년차 인력)	
LOC	5,000 이상	
FP	150 FP	
방법론	정보공학, Waterfall 모델	
적용단계	구현, 단위테스트	
비교도구	관련연구에서 소개한 NIPA와 TTA의 문서	

위 조건대로의 프로젝트를 수행하면서 실제 개발방법론을 그대로 적용하였고 개선된 신뢰도 특성 측정을 적용하기 위해 개발방법론의 해당 적용단계(구현, 단위테스트)에서만 개선된 신뢰도 측정 체크리스트를 한 번 더 적용을 하는 방식으로 진행하였다. 비교 대상이 되는 측정도구로는 관련연구에서 소개한 NIPA와 TTA의 문서를 이용하였고 이의 비교를 통해 본 논문에서 제안한 모듈이 표 11에서 보이듯이 더 좋은 결과가 도출됨을 알 수 있었다.

표 11. 개선된 신뢰도 평가 모듈 적용 결과
Table 11. Adaptation Result of Improved Reliability Evaluation Module

부특성	NIPA	TTA	제안
성숙성	90.6	97.69	99.42
오류허용성	97.7	91.04	97.17
회복성	96.19	96.4	95.35
준수성	99.93	95.1	99.03
문제진단/해결성	98.4	92.64	93.05
결함발생율	91.2	98.17	92.18
오조작방지성	90.82	93	98.66
복구성	91.8	91.53	98.41
평균	94.58	94.45	96.66

위 표 11에서 나타나는 숫자는 부특성의 하위 구성요소인 평가항목(매트릭)을 0과 1사이의 값으로 환산하여 모두 산출한 후 이를 부특성 별로 평균을 구하고 이를 전체적으로 다시 평균을 내어서 전체 값을 도출하였다. NIPA의 문서는 전체 소프트웨어 개발과정을 평가한 경우에는 제안한 도구보다 좋은 결과가 도출되는 반면, 신뢰도 측정부문에서는 제안한 모듈이 NIPA나 TTA의 문서보다 더 나은 결과가 도출됨을 알 수 있었다. 물론 신뢰도 평가의 개선을 위해 NIPA나 TTA의 문서를 개선하기는 하였지만 여러 가지 계산식이나 매트릭을 좀 더 적절하게 재배치 한 것이 주된 개선 요인인 것으로 나타났다. 그러나 일부의 부특성에서의 시험 결과가 제안한 결과보다 높게 나타난 경우도 있는데 이는 표 10에서 소개한 웹 어플리케이션의 개발 진행 과정상의 문제로 귀결된다. 즉, 소개한 프로젝트는 MM 이나 LOC 면에서 규모가 작은 관계로 QAO나 PMO의 충실한 활동에 다소 제약이 있었고, 제한적인 PM 의 활동에만 의존하여 진행한 상황이므로 최적화된 데이터들로 입력이 되지 않았기 때문이다. 그럼에도 불구하고 수치상 별 차이를 보이지 않기 때문에 관리의 충실함이 보장된 상황이라면 모든 수치에서 우월한 결과를 보일 것으로 예상된다.

V. 결론 및 향후 연구방향

본 논문에서 제안하는 신뢰도 품질 측정 모듈은 ISO/IEC 9126의 품질 모델을 기반으로 하고 있다. NIPA와 TTA의 도구 역시 이 모델에 기반을 하고 있지만, 제안한 도구가 더 우수함을 입증하기 위해 관련연구에서 제시한 두 모듈을 실제 프로젝트에서 동일하게 적용하여 정량적으로 상호 비교하였다.

이 논문에서는 가장 일반적으로 사용되는 개발 생명주기별 테스트 활동과 기법을 정리하여 지침으로 활용할 수 있도록 구체적인 가이드라인을 제시하였다. 이는 손쉽게 활용할 수 있는 체크리스트 등을 이용하여 해당 소프트웨어의 특성상 가장 중요한 품질특성 순으로 가중치를 부여하고 세부 평가항목에 대해 활용 가능한 테스트 기법과 체크포인트를 적용함으로써 계산결과를 도출할 수 있도록 하였다. 이는 소프트웨어의 새로운 신뢰도 품질측정 방안으로써 ISO/IEC 9126의 신뢰성 평가항목이 내포하는 문제점을 개선하였다.

개선된 신뢰도 측정 모듈은 그림 6이나 표 9에서 제시된 내용만으로도 신뢰성을 확보하는데 문제가 없지만 이를 응용한 응용 체크리스트를 활용하면 사용하기에도 편하고 결과 도출도 쉬워서 정량적인 결과가 훨씬 더 객관적으로 평가받을 수 있다. 이는 프로젝트나 소스코드의 신뢰성을 기존의 방식

이나 도구들에 비해 높은 수준으로 손쉽게 확보할 수 있는 환경을 제공한다. 또한 특정 개발 방법론이나 모델에 종속되지 않고 테일러링이나 필수/선택 옵션을 통하여 프로그램 개발과 프로젝트에서 범용적으로 사용할 수 있음을 보여준다.

추가 연구되어야 할 부분으로는 실제 적용 시 제시하는 항목들이 많게 느껴 질 수도 있으며, 범위가 너무 많다. 경우에 따라서는 개념이 비슷하여 통합하거나 생략 가능한 항목들이 있을 수도 있다. 또한 계산을 위해서 대부분 평가할 항목의 수를 소프트웨어 전체에 걸쳐 카운트하고 있는데 이는 현실적으로 쉽지 않다. 따라서 실제 프로젝트에 적용하기 위해서는 좀 더 품질모델 자체에 대한 조정과 현실화가 필요하다. 그리고 제안되어 있는 테스트 방법이나 계산식은 향후 더 정밀하고 정확성을 높일 수 있도록 차츰 개선해야 할 부분이다.

참고문헌

- [1] Myungmuk Kang, Taewan Gu, Jongmoon Baik, "Software Fault Detection and Removal Effort-based Reliability Estimation Model", KIISE : Software and Application 37th 7ed, 2010.07.
- [2] Hye-Jung Jung, "A Study on the Method of Software Reliability Quality Testing", Dept of Information Statistics, Pyeong-Taek University, 2006.06.
- [3] Won-Il Kwon, "Trends Survey of International standards of Software Quality", ISO/IEC 14598 Software Wngineering, TTA, 2003.01.
- [4] ISO/IEC 9126 : Information Technology - Software product evaluation - Quality Characteristics and Guidelines for Their Use.
- [5] ISO/IEC 12119 : Information Technology - Software Packages - Quality Requirement and Testing.
- [6] Ki-Chang Kim, Yong-Hwan Kim, Ju-Hwan Shin, Ki-Jun Han, "A Case Study on Application for Software Reliability Model to Improve Reliability of the Weapon System", KIISE : Software and Application 38th 8ed, 2011.08.
- [7] NIPA, "Software Technique Evaluation Standards (MKE 2010-53ed), 2010.05.
- [8] The laws and regulations and formulas,

<http://www.nipa.kr>.

- [9] TTAS, "Web-based Software Quality Evaluation Guideline", TTAS.KO-11.0059.
- [10] TTA, <http://www.tta.or.kr/index.jsp>.
- [11] G. Albeanu, A. Averian, I. Duda, "WEB SOFTWARE RELIABILITY ENGINEERING", Spiru Haret University, Bucharest, Romania, R&RATA # 4, (Vol.2) 2009, December.
- [12] Jongmoon Baik, "The Defense Case for Software Reliability Evaluation Process", KAIST, 2010.04.

저자 소개



임 광 현

2004 : 한양대학교 컴퓨터공학부 석사
 현 재 : 충북대학교 컴퓨터과학화
 박사과정
 관심분야 : 멀티미디어 데이터베이스,
 데이터마이닝
 소프트웨어 신뢰성 분석
 Email : khlim888@naver.com



류 광 선

2012 : 충북대학교 컴퓨터과학과 석사
 현 재 : 충북대학교 컴퓨터과학과
 박사과정
 관심분야 : 메디컬인포메틱스,
 데이터마이닝
 Email : ksryu@dblab.chungbuk.ac.kr



손 호 선

2010 : 충북대학교 컴퓨터과학과
 이학 박사
 현 재 : 충북대학교 데이터베이스/
 바이오인포메틱스 Post-Doc
 관심분야 : 메디컬인포메틱스,
 데이터마이닝
 Email : shon0621@dblab.chungbuk.ac.kr



류 근 호

1976 : 숭실대학교 전산학과 이학사

1980 : 연세대학교 공학대학원
전산전공 공학석사

1988 : 연세대학교 대학원
전산전공 공학박사

1976 ~ 1986 :

육군군수 지원사 전산실(ROTC 장교),
한국전자통신연구원(연구원),
한국방송통신대 전산학과(조교수) 근무

1989 ~ 1991 :

Univ. of Arizona Research Staff
(TemplS 연구원, Temporal DB)

1986 ~ 현 재 :

충북대학교 전기전자 컴퓨터공학부 교수

관심분야 : 시간 데이터베이스,
시공간 데이터베이스,
Temporal GIS,
지식기반 정보검색 시스템,
유비쿼터스 컴퓨팅 및 스트
림데이터처리, 데이터 마이닝,
데이터베이스, 보안,
바이오 인포메틱스

Email :

khryu@dblab.chungbuk.ac.krr