

# Study for Level of Portability to Quality Improvement of Embedded Software

Eun Ser Lee<sup>†</sup>

## ABSTRACT

As the importance of software has been recognized, the impact of embedded software quality on product quality has become increasingly important. Therefore, in this study, reliability is extracted according to the embedded characteristics and it is described that the reliability evaluation of the embedded software is made easier.

**Keywords :** Software Quality, Level of Reliability, Evaluation of Portability, Estimation of Software Quality

# 임베디드 소프트웨어 품질 향상을 위한 이식성 품질 수준 평가에 관한 연구

이 은 서<sup>†</sup>

## 요 약

소프트웨어의 중요성이 인식되면서 임베디드 소프트웨어의 품질이 제품 품질에 미치는 영향이 증가함에 따라 소프트웨어의 신뢰성 확보가 매우 중요한 문제로 떠올랐다. 따라서 본 연구에서는 임베디드 특성에 맞춰 신뢰성을 추출하여 임베디드 소프트웨어의 신뢰성 평가를 보다 쉽게 할 수 있도록 기술하였다.

**키워드 :** 소프트웨어 품질, 신뢰성 수준, 이식성 평가, 소프트웨어 품질 측정

## 1. 서 론

스마트폰 시장이 폭증하면서 소프트웨어의 중요성이 인식되고 제품의 고부가가치를 높이기 위한 임베디드 소프트웨어(SW)의 수요도 함께 증가하였다[1]. 또한 임베디드 소프트웨어의 품질이 제품 품질에 미치는 영향이 증가하였고, 이에 따라 임베디드 소프트웨어 신뢰성을 높이기 위한 노력을 더욱 강화해야 하며 임베디드 소프트웨어 신뢰성 확보가 매우 중요한 문제로 떠오르고 있다[2]. 소프트웨어의 신뢰성 평가는 많이 공용화 되었으나, 임베디드 소프트웨어 특성에 맞춘 신뢰성 관별을 하기에는 많은 결함이 따른다. 따라서 본 연구에서는 임베디드 특성에 근거하여 신뢰성을 추출하여 임베디드 소프트웨어의 신뢰성 평가를 보다 쉽게 할 수 있도록 기술하였다. 신뢰성 평가 항목으로는 이식성을 선정하여 이식성 품질 수준을 제시하고 검증하였다.

## 2. 관련 연구

### 2.1 결함 제거 효율 (DRE : Defect Removal efficiency)

소프트웨어 공학에서 결함을 제거하고 이를 평가하기 위하여 많은 방법들이 존재한다. 본 연구는 여러 방법 중에서 결함 제거 효율을 이용하여 검증을 할 것이다.

결함 제거 효율은 다음과 같이 정의된다.

결함 제거 효율 = 개발 단계가 끝나기 전의 결함 수 / 개발 단계가 끝나기 전의 결함 수 + 해당 개발 단계가 종료된 이후의 결함 수

결함 제거 효율은 1에 가까울수록 결함을 잘 발견해 제거가 된 것이다[3].

### 2.2 임베디드 소프트웨어의 특성

임베디드 시스템은 여러 전자 장비에 내장된 형태로서 정해진 기능을 수행하는 형태를 말한다. 따라서 이러한 기능을 수행하는 소프트웨어를 임베디드 소프트웨어라고 한다[4].

임베디드 소프트웨어의 특성은 다음과 같이 정의할 수 있다[6].

※ 이 논문은 안동대학교 기본연구지원사업에 의하여 연구되었음.

† 중신회원 : 안동대학교 컴퓨터공학과 교수

Manuscript Received : July 3, 2018

Accepted : July 25, 2018

\* Corresponding Author : Eun Ser Lee(eslee@anu.ac.kr)

- 실시간성 : 소프트웨어가 사용되는 목적에 따라서 시간의 제약을 받아서 수행되는 것이다.
- 신뢰성: 예상하지 못한 상황에 따른 오동작이 발생할 경우 빠른 대처가 요구되는 경우이다.
- 최적화 기술 지원: 임베디드 소프트웨어가 사용되는 환경에서 최적의 성능을 지원하기 위하여 기술 지원이 이루어지는 경우이다.
- 특정 시스템 전용: 임베디드 소프트웨어가 수행되는 목적이 일반적인 사용이 아닌 특정 목적을 갖는 경우이다.
- 네트워크/멀티미디어 처리기능 지원: 임베디드 소프트웨어가 네트워크에 연결되어서 멀티미디어 자료를 기반으로 수행되는 경우이다.
- 열악한 개발환경: 다양한 개발환경을 대비하기 위하여 여러 환경을 고려한 솔루션과 도구가 지원되는 경우이다.

2.3 임베디드 시스템의 경향

임베디드 시스템은 많은 변화가 진행되었다. 시스템의 특성상 하드웨어와 소프트웨어의 결합이 필수적인 요소가 되었으며 특정 목적을 수행하기 위한 형태로 발전되었다. 이러한 변화가 진행되는 또 다른 원인은 소프트웨어에서 다루는 자료가 급격히 변화되었기 때문이다. 또한 시스템의 규모가 작아지고 빠른 수행을 위하여 소프트웨어와 하드웨어의 크기가 작아지고 있다. 이러한 상황을 지원하기 위하여 고성능의 프로세스가 개발되어서 여러 기기에 사용되고 있다[14, 15].

3. 본 론

3.1 이식성 수준 평가를 위한 항목 분석

임베디드 소프트웨어의 특성은 실시간성, 신뢰성, 최적화 기술항목, 특정 시스템 적용, 네트워크/멀티미디어 처리기능 지원, 열악한 개발환경 등이 존재한다. 본 연구에서는 임베디드 소프트웨어의 품질 척도를 제시하기 위하여 신뢰성 항목을 선정하였다. 신뢰성은 여러 가지 기준으로 평가되어진다. 여러 항목 중에서 이식성을 기준으로 신뢰성 수준 평가기준을 제시하였다. 이식성이란 소프트웨어를 수정을 거의 하지 않고 다른 시스템 환경으로 이동할 수 있는 특성을 의미한다 [7-10, 13].

이식성에 대한 항목을 추출하기 위하여 5단계의 척도를 기준으로 하였다. 각 단계는 연속적인 수준이다. 연속적인 수준의 의미는 하위수준을 달성해야 상위수준을 달성할 수 있다. 상위수준은 달성하는데 하위 수준을 달성하지 못한다면 상위 수준을 달성하지 못하게 된다[11, 12].

Table 1은 이식성을 인터페이스, 플랫폼, 요구사항추출, 추적성 관점에서 5단계 수준으로 제시하였다. 내용은 다음과 같다.

인터페이스 관점에서의 이식성 수준은 다음과 같다. 수준1은 소프트웨어 로직을 수정해야 하는 경우이다. 수준2는 인터페이스의 파라미터 타입과 개수를 수정하는 경우이다. 수준3은 함수의 이름이 일치하지 않아서 변경을 요구하는 경우이다. 수준4는 기존의 인터페이스를 변경하지 않고 집합코드

Table 1. Level of Portability

Items \ Level	1	2	3	4	5
Interface(I)	Change of SW logic	Change of parameter type and number	Change of function name	Using of glue code	Not change
Platform(P)	Change of data base and system hierarchy	Change of function architecture	Using of middleware	Only recompile	Not change
Requirements elicitation	Just only requirements elicitation	Elicitation of priority	Elicitation of commonality and variability	Elicitation of function for the reusability	Using of object method
Traceability	Just only function elicitation	Elicitation of relationship among the functions	Check the cause of defect	Define of defect solution	Automation of defect solution

(glue code)를 사용하여 연동할 수 있는 경우이다. 수준5는 기존의 인터페이스 변경 없이 그대로 사용할 수 있는 경우이다.

플랫폼 관점에서의 이식성 수준은 다음과 같다. 수준1은 데이터베이스와 시스템 계층구조의 변경이 필요한 경우이다. 수준2는 함수의 연동을 위하여 클래스의 구조가 필요한 경우이다. 수준3은 이질적인 자료의 연동을 위하여 미들웨어를 사용하는 경우이다. 수준4는 다른 변경 없이 하드웨어의 구성 등이 다른 경우 컴파일을 다시 수행하는 것만으로도 사용이 가능한 경우이다. 수준5는 기존의 플랫폼을 변경 없이 그대로 사용할 수 있는 경우이다.

요구사항 추출 관점에서의 이식성 수준은 다음과 같다. 수준1은 이식성을 고려하지 않고 일반적인 요구사항추출 과정을 수행하는 경우이다. 수준2는 요구사항의 우선순위를 추출하는 경우이다. 우선순위는 요구사항들의 연동 시에 우선시 되는 요구사항을 추출하여 수행할 요구사항을 결정하는데 활용된다. 수준3은 요구사항들이 연동되기 위하여 여러 구조에 활용될 수 있는 요구사항과 특정 구조에 수행되는 요구사항을 분류할 필요가 있다. 이와 같은 공용성과 가변성의 추출은 이식성에서 올바른 요구사항이 기능화되는지를 측정하는 척도가 된다. 이식성이 있더라도 올바른 기능이 수행되지 않는다면 전체적인 품질이 떨어지게 된다. 수준2의 경우에도 이와 같은 이식성의 품질을 위하여 요구되는 사항이 된다. 수준4는 재사용성을 위하여 함수를 추출하는 경우이다. 재사용 함수를 위하여 요구사항을 추출하여 특정 기능을 수행하는 함수를 활용하여 수정 없이 사용하는 경우이다. 수준5는 완전한 재사용성을 높이기 위하여 객체지향 기법을 활용하여 요구사항을 추출하는 경우이다.

추적성 관점에서의 이식성 수준은 다음과 같다. 수준1은 단순히 기능을 추출하는 경우이다. 여기에서 기능 추출은 재사용을 고려하지 않는 경우이다. 이 경우는 “요구사항 추출의 수준4”와는 다른 경우이다. 수준2는 기능들간의 연관 관계를 추출하여 연동 가능 여부를 확인할 수 있는 경우이다. 연관 관계는 서로 주고받는 자료를 정의할 수 있게 된다. 수준3은 결함 발생 시에 원인을 확인하고 발견할 수 있는 경우이다. 이 경우는 결함이 이벤트 특성을 가지고 산발적으로 발생하는 경우이며 체계적인 오류 측정 방법이 적용된 경우된 경우가 아니다. 수준4는 결함을 해결하기 위한 해결책을 정의하고 있는 경우이다. 따라서 결함의 종류와 해결책이 정의된 경우

이다. 수준5는 결함을 확인하고 해결하는 과정에서 자동화 시스템이 구축되어서 문제를 해결할 수 있는 경우이다.

3.2 이식성 수준 평가를 위한 항목 분석

임베디드 소프트웨어의 특성은 실시간성, 신뢰성, 최적화 기술항목, 특정 시스템 적용, 네트워크/멀티미디어 처리기능 지원, 열악한 개발환경 등이 존재한다. 본 연구에서는 임베디드 소프트웨어의 품질 척도를 제시하기 위하여 신뢰성 항목을 선정하였다. 신뢰성은 여러 가지 기준으로 평가되어진다. 여러 항목 중에서 이식성을 기준으로 신뢰성 수준 평가기준을 제시하였다. 이식성이란 소프트웨어를 수정을 거의 하지 않고 다른 시스템 환경으로 이동할 수 있는 특성을 의미한다 [7-10, 13]. 3.1절 Table 1에서 제시한 표를 활용하여 이식성의 수준을 정량적으로 평가하고자 한다.

기본이론은 각 항목당 5개의 수준을 기준으로 가중치를 부여하여 산출하는 방식이다. 수식은 Equation (1)과 같다.

$$pq(i, j, k, r) = (\sum_{i=1}^5 I'(i) + \sum_{j=1}^5 P'(j) + \sum_{k=1}^5 R'(k) + \sum_{r=1}^5 T'(r)) / 4 \quad (1)$$

- I : 인터페이스 항목의 부여 값
- P : 플랫폼 항목의 부여 값
- R : 요구사항 추출 항목의 부여 값
- T : 추적성 항목의 부여 값
- i : 인터페이스 수준별 가중치
- j : 플랫폼 수준별 가중치
- k : 요구사항 추출 수준별 가중치
- r : 추적성 수준별 가중치
- I' : 인터페이스 항목과 수준별 가중치 연산 값
- P' : 플랫폼 항목과 수준별 가중치 연산 값
- R' : 요구사항 추출 항목과 수준별 가중치 연산 값
- T' : 추적성 항목과 수준별 가중치 연산 값
- pq : 정량적인 이식성 품질 수치

수식에서는 인터페이스, 플랫폼, 요구사항 추출, 추적성에 대하여 5단계 중에서 선택하여 이식성 품질을 측정하기 위한 것이다. 수식에서는 인터페이스, 플랫폼, 요구사항 추출, 추적성 항목의 5단계에 해당되는 I, P, R, T를 수준별 가중치에 곱하게 된다. 가중치는 개발 과제의 특성에 따라서 비중을 할당할 수 있다. 가중치는 기본적으로 1단계 0.1, 2단계 0.25, 3단계 0.5, 4단계 0.75, 5단계 1로 균등 설정하였다. 가중치 설정은 임베디드 소프트웨어의 특성에 맞게 사용자 정의가 되어야 한다. 본 연구에서는 가중치 설정은 향후 연구 과제로 하고자 한다. 본 논문에서는 가중치를 일반형으로 균등하게 할당하여 진행하고자 한다. 따라서 I', P', R', T'이 산출되다.

4. 적용 및 사례

4.1 의사결정 지원 도구 개발

3장의 이론을 적용하기 위하여 의사결정 지원 도구를 개발하였다. 의사결정 지원 도구는 개발 후에 임베디드 소프트웨어

형태를 가지며 휴대폰이나 태블릿에 탑재하여 사용하게 된다.

의사결정 지원 도구는 의사결정과정에서 발생하는 문제점들을 효율적으로 해결해 줄 수 있으며 기존의 의사결정 지원 도구들의 문제점을 고려하여 만든 의사결정 지원 도구이다. 운영체제 'Windows 7 Enterprise' 이용하였으며, 프로세서는 AMD사의 'FX(tm)-8300 Eight-Core Processor'이다. 설계된 애플리케이션 사용자 인터페이스는 운영체제 '안드로이드 OS 4.4.2'로 구동된다. 구현된 개발 형태는 다음과 같다.

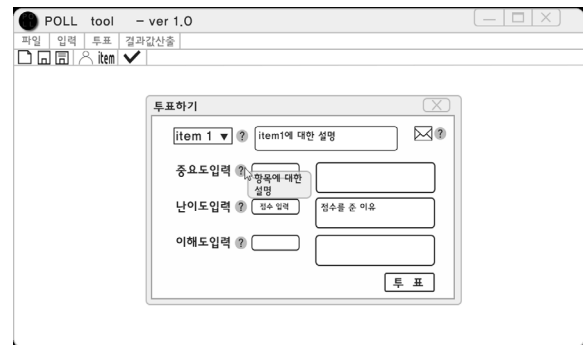


Fig. 1 Contents of Development

4.2 검증

의사결정 지원 도구 개발 시에 품질을 확인하기 위하여 개발 단계를 두 가지 방법으로 적용하였다. 방법1은 결함 관리를 생명주기에 적용하지 않으면서 개발을 한다. 방법2는 결함 관리를 위한 형상관리와 품질보증 활동을 수행하였다. 두 방법 모두 결과를 확인하기 위하여 개발 단계별 결함이 개수를 측정하였다.

두 방법을 3장의 기준으로 평가를 수행하였으며 이식성 품질이 낮게 나온 방법이 높게 나온 방법보다 결함 제거 효율이 떨어졌다. 따라서 3장의 방법에 대하여 검증을 하고자 한다.

이식성 품질을 측정하기 위하여 방법1과 2를 3장의 이론에 근거하여 산출하였다.

Table 2. Measurement

2	4	ε	5	↑	level
		5	↑		Interface
			5	↑	Platform
		5	↑		Requirements
		5		↑	Traceability

방법1과 2의 측정에 의하여 이식성 품질을 산출하면 다음과 같다.

방법1의 이식성 품질 수치 : (0.25 + 0.1 + 0.25 + 0.1) / 4 = 0.175

방법2의 이식성 품질 수치 : (0.5 + 0.25 + 0.5 + 0.5) / 4 = 0.4375

이식성 품질 수치는 1이 최대치며 1에 가까울수록 이식성에 대한 품질이 높음을 의미한다.

산출된 이식성 품질 수치를 검증하기 위하여 2장의 결함 제거 효율 방법을 활용하였다. 방법1과 2의 결함 제거 효율의 산출 내용은 다음과 같다.

방법1의 결함 개수는 다음과 같다.

Table 3. Numbers of Defect for Method 1

	E <sub>i</sub>	E <sub>i+1</sub>
Requirement Phase	13	7
Design Phase	10	4

요구사항 분석 단계 :  $13/13+7 = 0.65$

설계단계 :  $10/10+4 = 0.71$

방법2의 결함 개수는 다음과 같다.

Table 4. Numbers of Defect for Method 2

	E <sub>i</sub>	E <sub>i+1</sub>
Requirement Phase	4	1
Design Phase	6	1

요구사항 분석 단계 :  $4/4+1 = 0.8$

설계단계 :  $6/6+1 = 0.857$

방법1과 2의 요구사항 분석 단계의 결함 제거 효율은 65%, 80%로 방법2가 우수하게 산출되었다.

방법1과 2의 설계 단계의 결함 제거 효율은 71%, 85.7%로 방법2가 우수하게 산출되었다.

이와 같은 항목은 여러 항목 중에서 품질 척도를 추출하기 위하여 신뢰성 항목을 선정하였다.

### 5. 결 론

본 연구에서의 임베디드 소프트웨어에서 이식성 품질을 산정하기 위하여 인터페이스, 플랫폼, 요구사항 추출, 추적성에 대하여 5단계 수준을 제시하였다. 제시한 내용을 검증하기 위하여 결함 제거 효율 방법을 활용하였다. 향후 과제로는 임베디드 소프트웨어의 형태를 분류하고 형태에 따른 수준별 가중치를 정의하여 이식성 품질 측정의 정확도를 높이고자 한다.

### References

[1] Choi Jong Chang, Study for enhanced of embedded SW, Korean Electronics Technology Institute, 2013.  
 [2] Seo Jang Hoon, Development of the Reliability Evaluation Model and the Analysis Tool for Embedded Softwares, IE interfaces, 2008.

[3] Norman E. Fenton, A Critique of Software Defect Prediction Models, IEEE Computer Society, 1999.  
 [4] Park Tae Joon, Domestic standard of embedded SW, TTA, 2008.  
 [5] Hye-Jung Jung, A Study on the Standard of Software Reliability Quality Testing, THE KIPS SPRING CONFERENCE 2006, 2006  
 [6] Jian-Jia Chen, Embedded software reliability for unreliable hardware, International Conference on Embedded Software, 2014  
 [7] Ki-Du Kim, A Study on Reliability Evaluation for Embedded Software, The Journal of the Institute of Internet, Broadcasting and Communication, 2009.  
 [8] ISO/IEC 9126, Information Technology -Software Quality Characteristics and Metrics - Part 1, 2, 3.  
 [9] ISO/IEC 14598 Software Engineering Evaluation, 2001  
 [10] ISO/IEC 25000 "SQuaRE(Software product Quality Requirements and Evaluation)."  
 [11] ISO/IEC 982.2-1988 "IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Softwar."  
 [12] ISO/IEC 12119 "Software Packages - Quality requirement and Testing."  
 [13] Park chang soon, Embedded SW in life, U-BOOK, 2007.  
 [14] Hermann Kopetz, "Real Time Systems-Design Principles for Distributed embedded Applications," Kluwer Academic Publisher, 1997.  
 [15] Wayne Wolf, Computers as 'Components,' Morgan Kaufmann Publisher, 2001.



### 이 은 서

<https://orcid.org/0000-0002-7637-3036>

e-mail : eslee@anu.ac.kr

2001년~현 재 ISO/IEC 15504 국제

선임 심사원

2004년 중앙대학교 컴퓨터공학과(박사)

2008년~현 재 안동대학교 컴퓨터공학과

교수

관심분야 : CBD, Formal method, Quality model, SPI(Defect Analysis)