

논문 2009-3-28

임베디드 소프트웨어 신뢰성 평가에 관한 연구*

A Study on Reliability Evaluation for Embedded Software

김기두*, 김영철**, 김장한**

Ki-Du Kim, R. Young-Chul Kim, Jang-Han Kim

요 약 오늘날 산업의 융·복합화로 임베디드 소프트웨어의 비중이 높아져 가고 있다. 그 결과 임베디드 소프트웨어 개발이 필요했고 많은 업체 및 제품이 활용되었다. 임베디드 소프트웨어는 무기와 의료장비 등 사람의 생명에 큰 영향을 미치는 곳에 사용되기 때문에 신뢰성이 중요하다. 일반적인 소프트웨어 신뢰성 평가에 대해서는 품질관련 표준인 ISO/IEC 9126, ISO/IEC 12119 등에서 언급하고 있다. 하지만 임베디드 소프트웨어의 특성을 고려하지 않고 있기 때문에 신뢰성 평가에 적용하기에는 미흡하다. 이를 해결하기 위해 우리는 ISO/IEC 9126, 12119, 14598에서 소프트웨어 신뢰성 평가항목들을 추출하고, 추출된 항목들을 임베디드 소프트웨어의 특성을 고려한 체크리스트를 제안한다.

Abstract Nowadays there are gradually the importance issues of embedded software for productivity as the fusion and complexity of industry. Especially, people are recognizing the importance of reliability within the embedded software systems(e.g. weapons and medical equipments, etc.). It makes currently mention of software reliability on ISO/IEC 9126, 12119, etc.. However, it is insufficient for the embedded software area in spite of being mentioned the generic software. To do this, we suggest our embedded software reliability assessment through extracting the embedded software measurement items from reliability assessment items of ISO/IEC 9126, 12119 series.

Key Words : 신뢰성(Reliability), 임베디드 소프트웨어(Embedded Software), 평가리스트(Ch

I. 서 론

2000년도 이래로 많은 산업분야들의 융·복합화가 활발히 진행되고 있다. 이에 전체 소프트웨어가 제품생산에서 높은 비중을 차지하게 되었다. 그 결과 임베디드에 대한 관심이 높아졌고 임베디드 소프트웨어 개발자에 대한 요구가 증가하게 되었다. 그래서 임베디드의 장비 제공 업체들이 많이 생겨났다. 그러나 일반적인 임베디드 제품은 보통 확장성이 고려되어 있지만 안정성 및 신뢰성이 보장되지 않는다. 또한 제공되는 소프트웨어의 품질은 비교적 낮은 수준이다.

임베디드 시스템은 사람의 생명과 직결되는 무기, 항공, 자동차 등의 산업분야에서는 내부의 각 기관들의 제어를 하드웨어 내부에 내재되는 임베디드 소프트웨어가 많은 역할을 하고 있다. 이에 소프트웨어의 신뢰성 또한 소프트웨어를 평가하는 중요한 항목으로 인식되었다. 소프트웨어 신뢰성은 명세된 조건에서 사용될 때 성능수준을 유지할 수 있는 능력을 말한다[2]. 임베디드 소프트웨어의 경우 시스템 제어 및 사용자의 안전보장을 위해 일반 소프트웨어보다 더욱 신뢰성 평가가 매우 중요하다. 소프트웨어의 경우 신뢰성 측면으로 다양한 평가모델 개발을 통해 신뢰성 평가를 시도하지만, 임베디드 소프트웨어에 대한 신뢰성 평가에 대한 연구노력은 미비하였다. 또한 일반적으로 신뢰성평가 자체가 미흡하여 산업현장에서 소프트웨어 개발 시 문제가 생길 수 있어 신뢰성 평

*정회원, 한국정보통신기술협회 연구원

**정회원, 홍익대학교 컴퓨터정보통신 교수
접수일자 2009.05.15, 수정완료.2009.06.10

가가 적용될 필요성이 있다.

대부분의 개발사들은 임베디드 소프트웨어에 대한 신뢰성 평가를 임베디드의 특성을 고려하지 않은 일반 소프트웨어 신뢰성 평가방법을 적용하고 있어 임베디드 소프트웨어에 적용하기에는 부족하다. 즉, 현재는 대부분의 소프트웨어 개발사들이 일반적인 소프트웨어의 신뢰성 평가를 사용하고 있기 때문에 임베디드 소프트웨어의 특징인 실시간 처리, 작은 메모리 사이즈, 즉각적인 처리 등의 특성들이 고려되지 않고 있다[9]. 또한, 임베디드 분야의 전문가가 아니라면 신뢰성 평가항목을 적용하기 어려운 문제점을 갖고 있다.

본 연구에서는 임베디드 소프트웨어의 특성을 분석하고 임베디드 소프트웨어의 신뢰성 평가를 위해 ISO/IEC 9126. 12119, 14598에서 제안하는 소프트웨어 신뢰성 항목들을 임베디드 소프트웨어 신뢰성 측면으로 추출한 체크리스트를 제안하고, 적용방법에 대한 교육을 통해 일반 품질 담당자들도 용이하게 평가하는 방법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 기존의 소프트웨어 신뢰성과 임베디드 소프트웨어에 대하여 설명한다. 3장에서는 제안한 신뢰성 평가 방법에 대해 언급한다. 4장에서는 제안한 임베디드 소프트웨어 체크리스트에 대해 언급한다. 5장에서는 적용사례로서 GS인증을 신청한 임베디드 소프트웨어 10개 제품을 대상으로 수행해 본다. 마지막으로 6장에서는 결론 및 향후 연구를 언급한다.

II. 관련 연구

1. 소프트웨어 신뢰성(Software Reliability)

소프트웨어 신뢰성에 대한 정의는 여러 연구소 및 관

련 업체에서 다양하게 언급하고 있다. 그중에서 국제표준인 ISO/IEC 9126에서는 품질 항목에 신뢰성을 포함하고 있으며 소프트웨어 품질 평가를 위한 특성 및 척도를 제안하고 있다.

가. ISO/IEC 9126

ISO/IEC 9126 시리즈는 소프트웨어 품질을 정량적으로 측정하기 위한 주품질 특성 및 부품질 특성으로 정의하고 이들 사이의 관계를 정의하고 있다. 세부적으로 개발자, 구매자, 인증기관에서 정량적으로 품질평가에서 사용될 수 있는 서로 다른 품질 척도들을 정의하고 있으며 아래와 같이 구성된다.

- ISO/IEC 9126-1(소프트웨어 품질 특성 및 척도) : 주품질 특성 및 부품질 특성
- ISO/IEC 9126-2(소프트웨어 품질 특성 및 척도) : 외부척도
- ISO/IEC 9126-3(소프트웨어 품질 특성 및 척도) : 내부척도

ISO 9126-1은 소프트웨어의 품질평가를 위한 주품질 특성과 부품질 특성을 정의하고 있다. 품질특성 사이에는 중복성을 배제하고, 각 주품질 특성은 다시 세분화되어 여러 개의 부품질 특성으로 정의된다. ISO 9126-1에서 정의한 품질특성은 개발자, 구매자, 인증심사원이 공동으로 사용하는 특성이 된다[2].

ISO/IEC 9126에서 주품질과 주품질을 세분화한 부품질들 사이의 구조를 살펴보면 아래와 같이 정의하고 있다.

- 기능성(Functionality) : 명시적 또는 묵시적 필요를 만족하는 소프트웨어 기능의 존재



그림 1. ISO/IEC 9126 구조[2]

Fig. 1. Structure of ISO/IEC 9126

- 신뢰성(Reliability) : 성능수준을 유지하는 능력
- 사용성(Usability) : 사용의 편리성 및 선호도
- 효율성(Efficiency) : 성능과 사용된 자원의 양 사이의 관계
- 유지보수성(Maintainability) : 수정에 필요한 능력
- 이식성(Portability) : 소프트웨어가 다른 환경으로 이전되는 능력

ISO/IEC 9126에서는 신뢰성에 대해 “명시된 조건에서 사용될 때 성능수준을 유지할 수 있는 소프트웨어 제품의 능력(The capability of the software product to maintain a specified level of performance when used under specified conditions)” 이라고 정의하고 있다[2]. 그림 1은 ISO/IEC 9126에 대한 구조를 도식화 한 것이다. 소프트웨어 신뢰성에 대해 아래와 같은 항목들이 있다.

- 성숙성(Maturity) : 소프트웨어 내의 결함으로 인한 고장을 피해가는 소프트웨어 제품의 능력
- 오류허용성(Fault tolerance) : 명세된 인터페이스의 위반 혹은 소프트웨어 결함이 발생했을 때 명세된 성능수준을 유지할 수 있는 소프트웨어 제품의 능력
- 회복성(Recoverability) : 고장발생시 명세된 성능수준을 재유지 시키고 직접적으로 영향 받은 데이터를 복구하는 소프트웨어 제품의 능력
- 준수성(Reliability compliance) : 신뢰성과 관련된 표준, 협약, 규정을 준수하는 소프트웨어 제품의 능력

나. IEEE std 982.2-1988

IEEE std 989.2는 신뢰성 있는 소프트웨어 프로세스 측정을 위한 개념과 동기를 제공하고 있다. 또한, 프로젝트에서 적용하기 위해 IEEE std 982.2-1988에서는 소프트웨어 라이프사이클을 통해 제품과 프로세스에 적용, 지속적인 자기 평가와 신뢰성 향상을 위한 수단, 신뢰성 있는 소프트웨어 개발의 최적화 방법, 비용과 시간의 압박을 예상하여 초기 개발단계에서 시작, 운영과 유지보수 동안의 실제 사용 환경에서 소프트웨어 신뢰성을 최대화 하는 방법, 신뢰성 관리를 위한 수단 개발 등에 대한 정보들을 제공한다[7].

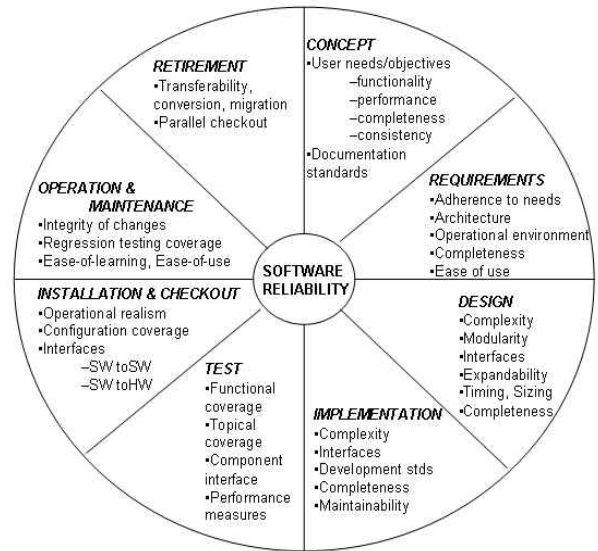


그림 2. 신뢰성에 영향을 주는 품질 요소[7]
Fig. 2. Elements that affect the quality of Reliability

그림 2는 다양한 라이프사이클에서 적용할 수 있는 소프트웨어 신뢰성 항목들을 보여준다. 또한 소프트웨어 신뢰성 관리프로그램은 개발비용과 시간에 영향을 받지 않도록 균형 있는 구조를 갖고, 개발 중간마다 품질 수준을 식별하는 것은 사용자가 요구하는 최종 품질수준을 달성하는데 도움이 된다[6].

2. 임베디드 소프트웨어

임베디드 소프트웨어는 특정한 목적을 수행하기 위해 개발된 시스템의 한 구성요소이다. 예를 들면 자동차의 ABS(Anti-lock breaking system)에 사용되는 마이크로 프로세스 및 하드웨어와 ABS를 제어를 위해 사용되는 프로그램이 임베디드 소프트웨어다[1,6].

임베디드 소프트웨어는 일반 소프트웨어와 다른 특징이 몇 가지 존재한다. 임베디드 소프트웨어는 일반 소프트웨어와 달리 특정 하드웨어 스펙과 개발 목적에 따라 개발되기 때문에 개발이 제한적이다. 즉, 하드웨어가 개발되면 그 구조에 적합한 소프트웨어를 설계하는 방법으로 개발된다. 이는 하나의 시스템에 다양한 임베디드 소프트웨어가 적용될 수 있음을 나타낸다.

임베디드 소프트웨어는 일반 소프트웨어와 달리 적용 분야들의 특성으로 실시간 처리를 요구한다. 상황처리를 위한 시스템들의 경우 예를 들어, 엘리베이터의 자동문의 경우 센서를 통해 들어온 정보를 실시간으로 처리해야하며 잘못될 경우 사용자의 신체 및 생명에 영향을 줄

수 있다. 따라서 외부상황에 즉각 반응해야하는 실시간 처리를 제한시간 안에 처리해야 것이 중요하다.

임베디드 소프트웨어가 세탁기, 냉장고, 게임기 등 일 반생활에 많이 활용되기 시작하였다. 이들은 모두 대량 으로 양산해야하는 판매구조를 갖고 있다. 즉, 모두 임베 디드 시스템으로 판단하여 사용자가 평가를 하게 된다. 최종 사용자는 하드웨어의 스펙 혹은 임베디드 소프트웨 어에 좌우되지 않는다.

다른 특성으로는 이동성(Mobility)이 있다. 임베디드 시스템의 크기가 작아지고 휴대를 목적으로 개발되는 제 품이 많이 나오면서 장시간 이동 및 운영을 위한 저전력 소모의 소프트웨어 개발에도 많은 관심을 갖게 되었다.

III. 제안한 임베디드 소프트웨어 신뢰성 평가 방법

기존의 소프트웨어 신뢰성 평가는 Markov Process Model, Poisson Process Model, Bayesian Model 등 소프 트웨어 신뢰성 성장 모델을 이용한 방법들이었다. 하지 만, 소프트웨어 신뢰성 성장모델의 경우 평가 방법 및 적 용이 일반 품질 사용자들에게 어렵기 때문에 성장모델을 사용하지 않는다. 본 연구에서는 한국정보통신기술협회 에서 시험을 수행하면서 쌓은 시험 데이터를 기반으로 ISO/IEC 표준들에서 언급하지 않은 임베디드 소프트웨 어의 특성을 고려한 신뢰성 평가 항목을 확장하여 체크 리스트를 제안하고 이를 사용하여 일반 품질 담당자들이 사용할 수 있는 측정 방법을 제안하고자 한다[5].

1. 임베디드 소프트웨어 신뢰도 평가척도

본 연구에서 제안하고 있는 임베디드 소프트웨어 신 뢰성 평가 항목은 기존의 ISO/IEC 9126, 12119, 14598에 서 제시하고 있는 소프트웨어 신뢰성 항목을 기반으로 IEEE std 982.2-1988을 적용하였다. 이를 통해 소프트웨 어 개발 라이프사이클에서 실제 적용될 수 있는 임베디 드 신뢰성 항목들을 추출하였다[2,3,4].

즉, ISO/IEC 9126, 12119, 14598에서 소프트웨어 품질 과 관련하여 정의된 요소들 중 소프트웨어의 신뢰성을 평가할 수 있는 대분류 항목들을 추출하였다. 추출된 항 목들 중 성숙성, 결함허용성, 회복성, 기능성을 중심으로 세부항목들을 추출하였다. 추가적으로 일반적 소프트웨

어 신뢰성을 나타내는 기준은 IEEE std 982.2-1988의 TEST, REQUIREMENT 항목들을 고려하여 시간반응 성, 자원효율성 추출하여 제정의 하였다. 각 대항 목에 대한 정의 내용은 아래와 같다.

- 성숙성(Maturity) : 고장을 피하기는 소프트웨어의 능력[2,3,9]
- 결함허용성(Fault Tolerance) : 결함이 발생했을 때 명세된 성능수준을 유지하는 능력[2][3][9]
- 회복성(Recoverability) : 고장 발생 시 성능수준을 재유지하고 직접적으로 영향받은 데이터를 복구하는 제품 능력[2,3,9]
- 시간반응성(Time Behavior) : 명시된 조건에서 적절 한 반응시간, 처리시간, 처리율을 제공하는 능력 [2,3,9]
- 자원효율성(Resource Utilization) : 명시된 조건에서 적절한 자원을 사용하는 능력[2,7]
- 기능성(Functionality) : 규정된 기능들의 기능특성들 의 구현수준[2,7]

2. 임베디드 소프트웨어 신뢰성 평가 척도

앞 장에서 언급한 ISO/IEC 표준들이 언급한 일반 소 프트웨어 신뢰성 평가 항목에서 임베디드 소프트웨어 시 험 시 고려해야 할 사항들을 기반으로 체크리스트를 표 1과 같이 제안하였다.

현재의 적용방법은 일반적인 임베디드 소프트웨어 적 용을 목적으로 연구된 방법이기 때문에 분류항목 별로 가중치 설정을 적용하지 않는 상태 이다.

표 1. 임베디드 소프트웨어 신뢰성 평가 척도
Table 1. Reliability rating scale in Embedded SW

대분류	세부항목
성숙성 (Maturity)	문제해결이력정보제공(PRI)
	문제해결력(PRR)
	결함회피율(FOR)
	결함발생평균시간(FAT)
결함허용성 (Fault Tolerance)	다운회피율(DAR)
	고장회피율(FAR)
	오조작 회피율(IOA)
회복성 (Recoverability)	데이터회복정보제공(DIP)
	데이터회복율(DRR)
	이용가능율(UPR)
	평균복구시간(ART)
	복구가능율(RR)
	복구효과율(RTR)

시간반응성 (Time Behavior)	시간반응 정보제공(TBI)
	평균 반응시간(AAT)
	평균 처리율(APR)
	평균 처리시간(APT)
자원효율성 (Resource Utilization)	I/O자원 사용률(IOU)
	메모리 사용률(MMU)
	데이터 전송률(DTR)
	전력 사용율(EPU)
기능성 (Functionality)	적합성(SUT)
	정확성(AUC)
	상호운용성(ITB)
	보안성(SEC)

특정 임베디드 소프트웨어 분야에 적용하기 위해서는 해당 소프트웨어 특징을 반영하는 가중치가 반영되어야 할 것으로 판단된다.

3. 임베디드 소프트웨어 신뢰성 평가 방법

임베디드 소프트웨어 신뢰성 평가 시 우선 시험 대상 제품을 분석해야한다.

표 2. 결함허용성 테스트 케이스 도출 예
Table 2. Example of Availability defect Test case

대분류	세부항목	테스트 케이스
결함허용성 (Fault Tolerance)	오조작 회피율 (IOA)	프로그램에서 올바른 입력 구분 제시 여부
		입력 필드에 입력된 내용을 확인하는 기능 여부
		도움말에 제시된 정보에 의한 오조작 유도 여부
		사용자가 정의된 작업 순서를 따르지 않을 시 유용한 옵션과 옵션에 대한 정보 제공 여부
		메뉴 및 기능 버튼의 간격에 의한 사용자의 버튼 오조작 유발 여부 등
회복성 (Recoverability)	데이터 회복 정보 제공 (DIP)	사용자 설명서 등 데이터 회복에 대한 정보 제공 여부
	데이터 회복율 (DRR)	데이터 회복 후 일부 회복되지 않는 데이터가 존재 여부
		데이터 회복 후 해당 데이터 정상 사용 여부
		데이터 회복 후 손상된 데이터 존재 여부
		데이터 회복 시 불필요한 어플리케이션 종료 여부
		데이터 회복 후 다른 어플리케이션 및 시스템에 영향을 주는지 여부
		데이터 회복 후 불필요한 시스템 재부팅 여부
		데이터 회복 이력에 대한 정보를 제공 여부 등

이를 기반으로 임베디드 소프트웨어 신뢰성 평가를 위한 평가 항목별 테스트 케이스 및 시나리오를 도출한다. 표 2는 임베디드 소프트웨어를 대상으로 평가항목별 테스트 케이스를 도출한 예이다.

테스트 케이스 수행 결과를 임베디드 소프트웨어 신뢰성 평가를 위한 평가 체크리스트에 적용한다. 체크리스트를 적용한 결과를 토대로 최종적으로 임베디드 소프트웨어에 대한 신뢰성을 점수로 평가 한다. 그림 3은 임베디드 SW 신뢰성 평가를 위한 절차를 도식화 한 것이다.

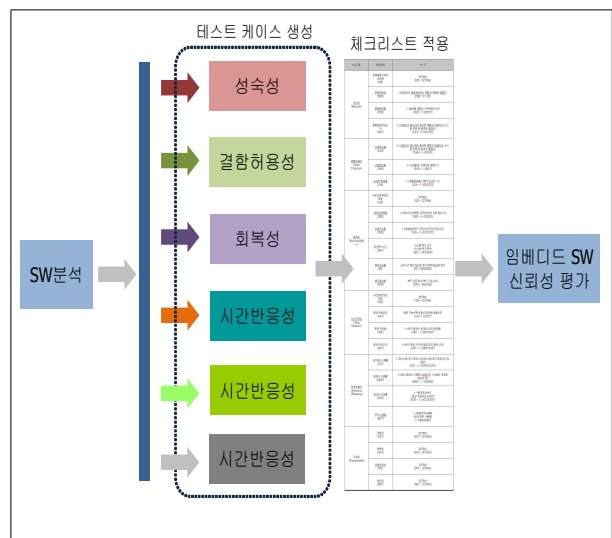


그림 3. 임베디드 SW 신뢰성 평가 절차
Fig. 3. Reliability Evaluation procedures in Embedded SW

IV. 임베디드 소프트웨어 신뢰성 체크리스트

이번 장에는 앞에서 언급한 임베디드 소프트웨어 신뢰성 평가항목을 기준으로 실제 품질 담당자가 측정해야 하는 내용을 체크리스트로 <표 3>과 같이 제안하였다.

소프트웨어 품질 담당자는 임베디드 소프트웨어 신뢰성 체크리스트를 통해 신뢰성 평가를 용이하게 수행할 수 있다. 또한, 동일한 체크리스트를 사용하여 임베디드 소프트웨어들 간의 동일한 신뢰성을 유지할 수 있다.

표 3. 임베디드 소프트웨어 신뢰성 체크리스트
Table 3. Reliability checklist in Embedded SW

대분류	세부항목	수식	점검사항	측정값
결함허용성 (Fault Tolerance)	오조작 회피율 (IOA)	1-(결함발생율/사용자 오조작 수) (IOA = 1-(FR/UIO))	시험 중 발견된 결함 수(F)	
			시험 시간(운영시간)	
			사용자 오조작 발생 개수	
회복성 (Recoverability)	데이터 회복정보 제공 (DIP)	X/Y/NA (DIP = X/Y/NA)	데이터 회복정보 제공여부	
	데이터 회복율 (DRR)	1-(데이터가 회복된 경우/데이터 오류 발생 수) (DRR = 1-(DR/DF))	데이터 회복이 가능한 부분의 개수	
			데이터 오류가 발생한 결함 수	

* X : 해당 항목의 내용을 제공하지 않는 경우 = 0
Y : 해당 항목의 내용을 제공하는 경우 = 1
NA : 해당 항목과 관련이 없는 경우 = None

V. 임베디드 소프트웨어 평가

본 연구의 적용 대상은 '06년도에 GS인증을 신청한 임베디드 소프트웨어 10개 제품을 대상으로 적용하였다. 신뢰성 평가는 업무를 담당하지 3개월 이내의 시험원과 2년 이상된 시험원을 대상으로 평가를 실시하였다. 평가 대상 중 5개 제품은 GS인증을 획득하였고, 5개 제품은 GS인증에 누락된 제품을 대상이었다. 교육용 임베디드 소프트웨어 신뢰성 평가항목에 적용한 결과 공정제어, 감시 장비 등 10개 제품의 평가 결과 평균은 그림 4와 같다.

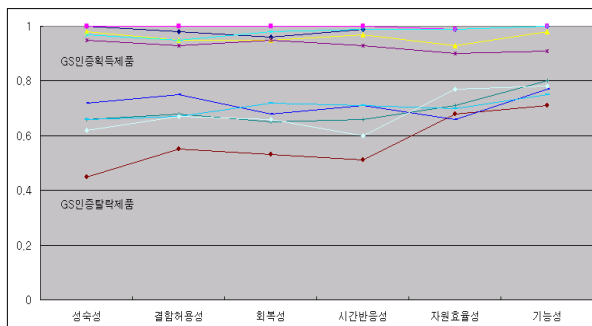


그림 4. 임베디드 소프트웨어 신뢰성 평가 결과
Fig. 4. Reliability Evaluation procedures result in Embedded SW

대상제품을 측정한 결과 GS인증 기준을 통과한 5개 제품의 경우 측정한 결과가 신뢰성 평가 점수가 평균 0.80이상의 점수를 받았다. 하지만, GS인증을 획득하지 못한 5개 제품의 경우 모두 평균 0.70 미만의 점수를 획득하였다. 즉, 실제 교육용 임베디드 소프트웨어 제품을 대상으로 평가하여 품질 개선 시 신뢰성이 차이가 있음을 확인 할 수 있었다.

3개월 이내의 시험원이 유사 소프트웨어의 신뢰성 평가 시간이 평균 5일이 소요되었으며, 체크리스트를 적용하지 않고 수행했을 때 7일 소요된 것에 비해 2일이 줄어들었다. 2년 이상된 시험원의 경우 소요시간이 4일이 소요되었으며, 체크리스트 적용하지 않고 수행했을 때 5일이 소요된 것에 비해 1일이 줄어들었다.

VI. 결론

소프트웨어 기술의 급격한 발전으로 오늘날 모든 분야에서 소프트웨어가 적용되고 있다. 특히 시스템 제어 등의 목적으로 사용되는 임베디드 소프트웨어의 경우 갈수록 복잡해지고 다양한 기능이 요구됨에 따라 사람의 생명과도 연결되는 임베디드 소프트웨어 신뢰성에 대한 관심 또한 높아지게 되었다.

본 연구에서는 임베디드 소프트웨어 신뢰성을 평가하기 위해 기존 ISO/IEC 표준에서 제공하는 소프트웨어 신뢰성을 평가방법에서 부족한 부분을 IEEE 982.8-1988과 실제 임베디드 소프트웨어가 갖고 있는 특성을 고려하여 신뢰성 평가항목을 추출하였다. 이를 통해 임베디드 소프트웨어 신뢰성 체크리스트를 사용한 신뢰성 평가방법을 제안하였다. 제안한 방법으로 임베디드 소프트웨어들의 동일한 신뢰성이 확보 및 평가 시간이 절약 될 것으로 예상된다.

향후 임베디드 신뢰성 측정을 위한 항목을 세부적으로 조정하여, 다양한 임베디드 시스템과 평가자를 대상으로 적용할 예정이다. 또한 소프트웨어 신뢰성 수준을 구분하는 신뢰성 성숙도 모델(RMM)을 개발하려고 한다.

참고 문헌

[1] David E. Simon, "An Embedded Software Primer", Addison Wesley 1999

- [2] ISO/IEC 9126 "Information Technology- Software Quality Characteristics and Metrics - Part 1,2,3
- [3] 한국정보통신기술협회, "Embedded S/W의 품질평가 모델 개발연구", 2002.
- [6] 박인수 " 국제 표준의 소프트웨어 품질평가기술과 임베디드 소프트웨어" 임베디드 소프트웨어 테스트 세미나. 2005
- [4] ISO/IEC 14598 "Software Engineering Evaluation, 2001
- [5] ISO/IEC 25000 "SQuaRE(Software product Quality Requirements and Evaluation)"
- [7] ISO/IEC 982.2-1988 "IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software"
- [9] ISO/IEC 12119 "Software Packages - Quality requirement and Testing"

※ 본 연구는 2007학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음
 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터육성지원사업의 연구결과로 수행되었음
 (C1090-0903-0004).

저자 소개

김 기 두(정회원)

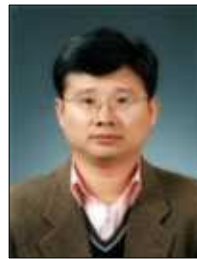


e-mail : kdkim@tta.or.kr

- 2003년 홍익대학교 컴퓨터정보통신 (학사)
- 2005년 홍익대학교 일반대학원 소프트웨어공학전공(석사)
- 2005년 ~ 현재 일반대학원 박사과정
- 2005년 ~ 현재 한국정보통신기술협회 SW시험인증연구소 연구원

<주관심분야 : 테스트 성숙도 모델, 테스트 프로세스, 임베디드 소프트웨어 테스트, 소프트웨어 신뢰성 테스트>

김 영 철(정회원)



e-mail : bob@selab.hongik.ac.kr

- 2000년 Illinois Institute of Technology(공학박사)
- 2000년 ~ 2001 LG 산전 중앙연구소 Embedded system 부장
- 2001년 ~ 현재 홍익대학교 컴퓨터정보통신과 부교수

<주관심분야 : 테스트 성숙도 모델, 임베디드 S/W 개발 방법론 및 도구 개발, 모델 기반 테스트, CBD, BPM, 사용자 행위 분석 방법론>

김 장 한(정회원)



e-mail : jkiml@hongik.ac.kr

- 1977년 서울대학교 전자공학과(학사)
- 1979년 서울대학교 전자공학과(석사)
- 1993년 홍익대학교 전자공학과(박사)
- 1979 ~ 1994년 홍익공업전문대학 부교수
- 1994년 ~ 현재 홍익대학교 컴퓨터정보통신 정교수

<주관심분야 : 영상신호처리, 통신이론, 디지털통신>