원전 MMIS 소프트웨어 개발을 위한 시험 프레임웍 개발

이종복° 서상문 서용석 장귀숙 금종용 구인수
한국원자력연구소 계측제어·인간공학연구부
{jblee1°, smsuh, yssuh, gsjang, jykeum, iskoo}@kaeri.re.kr

Development of Test Framework to develop MMIS Software for Nuclear Power Plants

Jongbok Lee° Sungmoon Suh, Yongsuk Suh, Gwisook Jang, Jongyong Keum, In-soo Koo
Korea Atomic Energy Research Institute

요 약

소프트웨어 시험은 소프트웨어 제품의 고 품질을 보장하기 위한 중요한 요소들 중의 하나이고, 특히 신뢰도가 원자력 발전소의 안전에 직결되는 디지털 기반의 원전 계측제어계통 소프트웨어는 고품질과 고신뢰도를 제공하여야 한다. 그러므로 원자력발전소에 사용되는 소프트웨어는 안전성과 신뢰성을 제공하기 위해 체계적인 시험을 통하여 설계의 정당성을 확인하고, 요건명세서나 설계사양서에 나타난 계통 및 구성요소의 기능과 요건들이 만족하게 실행됨을 확인하여야 한다. 규제기관에서도 소프트웨어의 안정성, 기능의 완전한 수행, 소프트웨어 자체가 계통의 기능을 저하 시키는지와 계통에게 예정되지 않은 기능을 수행 하도록 영향을 주는지의 확인 등을 소프트웨어 시험을 통해 확인하도록 요구하고 있다. 이와 같이 원자력발전소에 사용되는 소프트웨어의 시험을 위해서는 보다 엄격하고 명확한 시험 프레임웍을 개발하고 적용하는 것이 필요하다.

본 논문에서는 소프트웨어 시험과 관련된 인허가 규제요건을 분석하고, 이에 따라 현재 설계를 진행중인 SMART MMIS 소프트웨어 시험에 적용될 소프트웨어 개발생명주기 시험활동, 시험 조직, 시험문서, 소프트웨어 등급별 시험방법 등 시험 프레임웍을 제시한다.

1. 서 론

소프트웨어 시험은 소프트웨어 제품의 고 품질을 보장하기 위한 중요한 요소들 중의 하나이다. 특히 신뢰도가 원자력 발전소의 안전에 직결되는 디지털 기반의 원전 계측제어계통 소프트웨어는 고품질과 고신뢰도를 제공하여야 한다. 그러므로 원자력발전소에 사용되는 소프트웨어는 안전성과 신뢰성을 제공하기 위해 체계적인 시험을 통하여 설계의 정당성을 확인하고, 요건명세서나 설계사양서에 나타난 계통 및 구성요소의 기능과 요건들이 만족하게 실행됨을 확인하여야 한다. 원자력 규제기관에서는 소프트웨어의 안정성, 기능의 완전한 수행, 소프트웨어 자체가 계통의 기능을 저하 시키는지와 계통에게 예정되지 않은 기능을 수행 하도록 영향을 주는지의 확인 등을 소프트웨어 시험을 통해 확인하도록 요구하고 있다. 이에 따른 시험방법으로 수계산, 비교 가능한 증명된 컴퓨터 프로그램을 사용한 계산, 기술문헌상의 경험(실험)자료나 정보사용, 입력 매개변수의 요구범위 확인, 시험논리 분기점의 확인 등을 예로 들고 있다. 이와 같이 원자력발전소에 사용되는 소프트웨어의 시험을 위해서는 보다 엄격하고 명확한 시험 프레임웍을 개발하고 적용하는 것이 필요하다. 이에 따라 본 논문에서는 현재 설계를 진행중인 SMART(System-integrated Modular Advanced ReacTor) MMIS(Man-Machine Interface System) 소프트웨어 시험에 대하여 RG 1.170과 RG 1.171에서 기술하고 있는 내용과 KEPIC QAP 원자력 품질보증, ENB 6370, 그리고 SMART MMIS 확인 검증 절차를 바탕으로 하여 소프트웨어 시험의 정당성을 보장하는 소프트웨어 시험 프레임웍을 제시한다.

2. 시험관련 인허가 규제요건

전력산업기술기준(KEPIC) QAP 원자력 품질보증과 ENB 6370에서는 소프트웨어 시험을 통하여 설계문서에 기술된 기능 및 운전 범위에 대해서 요구되는 성능을 확인하도록 하며, RG 1.170과 1.171에서는 안전계통에 관련된 소프트웨어 시험에 관한 문서와 단위시험에 관한 내용을 기술하고 있다. RG 1.170 지침서는 IEEE Std 829의 내용을 인준 하지만, 몇 가지 예외 규정을 마련하고 있다. RG 1.171 또한 예외 사항을 포함하여 IEEE Std 1008을 인준하고 있다. 이에 따라 소프트웨어 개발수명 주기활동에 따른 SMART MMIS 소프트웨어 시험들은 계통별 단위시험, 계통별 모듈통합시험, 계통별 계통시험, MMIS 통합시험, 계통별 현장인수시험, MMIS 시운전시험으로 구성된다.

3. 소프트웨어 개발생명주기 시험활동

SMART MMIS 소프트웨어 시험과 관련되는 대표적인 문서는 시험계획서, 시험설계명세서, 시험절차서, 시험요

약보고서 등이며 SMART MMIS 소프트웨어 개발 수명주기에서 이루어지는 시험은 계통별 단위시험, 계통별 모듈통합시험, 계통별 계통시험, MMIS 통합시험, 계통별 현장인수시험, 시운전시험으로 구성된다. 따라서 각각의 시험마다 위에 기술된 시험 관련문서를 작성한다. 그림 1은 SDLC(Software Development Life Cycle)에 따른 시험과정을 설명하고 있다.

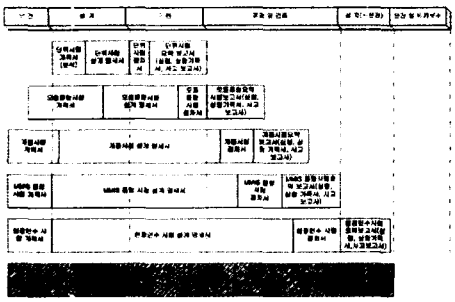

그림 1 소프트웨어 개발 수명주기별 시험활동

SMART MMIS 소프트웨어의 등급은 안전관련 기능의 중요도에 따라 아래와 같이 세 등급으로 분류한다.

- 안전-필수(SC, Safety-Critical) 소프트웨어
- 안전-관련(SR, Safety-Related) 소프트웨어
- 비안전(NS, Non-Safety) 소프트웨어

위와 같은 소프트웨어 등급분류의 목적은 소프트웨어 등급별로 소프트웨어 시험을 차등적으로 적용하기 위한 것이다.

3.1 소프트웨어 시험 수행주기

SMART MMIS 소프트웨어 개발과정에서 수행하는 계통별 단위시험, 계통별 모듈통합시험, 계통별 계통시험, MMIS 통합시험, 계통별 현장인수시험, MMIS 시운전시험에 공통적으로 적용하는 시험수행활동주기는 그림 2과 같다.

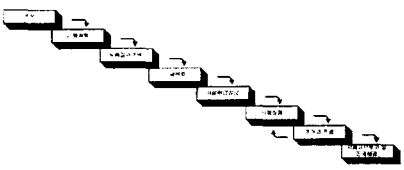

그림 2 시험 수행주기

3.2 시험조직

프로그래머가 자신이 개발한 소프트웨어를 시험하는 것을 디버깅(debugging)으로 정의하며 공식적인 시험이라 하지 않는다. 공식적인 시험의 의미는 시험하고자 하는 소프트웨어를 프로그래밍하지 않은 사람이 공식적인 시험절차에 의해 수행하는 시험을 의미한다. SMART MMIS 소프트웨어 시험의 경우 안전-필수 소프트웨어는 공식적인 시험을 수행하더라도 개발조직과 소속, 재정적으로 독립된 "독립검토 팀" 조직에서 시험 또는 검토를 받아야 하며, 안전-관련 및 비안전 소프트웨어는 개발조직의 공식적인 시험조직에 의해서 수행된다.

SMART MMIS 시험관련 조직의 구성은 시험 관리자, 시험 분석가, 시험 기술자로 구분한다. 또한 특이한 사항이 발생하면 외부의 구성원을 포함할 수 있다. 여기서의 특이한 사항이란 시험결과의 분석, 교육 등과 같이 시험조직 내에서만 해결할 수 없는 상황을 의미한다. 시험조직의 구성인원은 소프트웨어 개발책임자에 의해서 시험과 관련되는 계통의 특성과 업무량에 따라서 정해진다.

3.3 시험수행절차

SMART MMIS 소프트웨어 시험수행절차는 그림 3와 같다. 시험조직을 구성하고 계통교육과 시험관련 교육을 실시한 후, 시험할 계통에 대한 요건명세서, 설계사양서 등의 관련문서를 이해한다. 시험계획서와 시험설계명세서, 시험절차서를 작성하고, 시험수행을 완료 후 시험요약보고서를 작성한다.

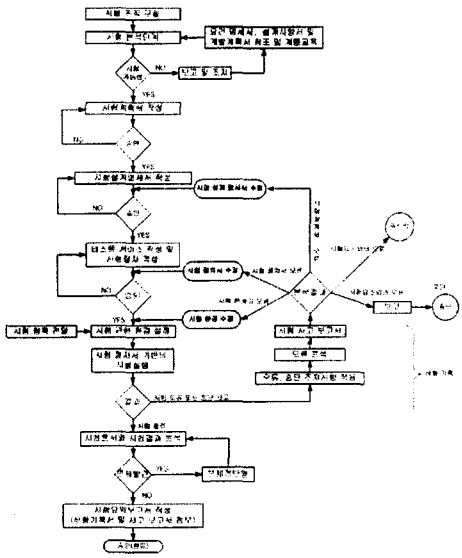

그림 3 시험 활동

3.4 소프트웨어 시험 문서

SMART MMIS 소프트웨어 시험관련 문서는 계통별로 작성하며, 시험설계명세서는 시험계획서에 기술된 모든 시험대상을 포함하도록 하며, 시험절차서는 시험설계명세서의 시험항목을 기준으로 작성된다. 시험상황기록서는 각각의 시험절차서 마다 작성되며, 사고보고서는 사고발생시 작성된다. 시험완료 후에는 시험수행과 결과를 평가하는 시험요약보고서를 작성한다. 그림 4은 소프트웨어 시험문서의 관계를 나타낸 것이다. 한 계통에 안전-

필수 소프트웨어, 안전-관련 소프트웨어, 비안전 소프트웨어로 구분되어 있을 경우 시험과련 문서 내에서 안전-필수 소프트웨어, 안전-관련 소프트웨어, 비안전 소프트웨어로 구분하여 작성한다.

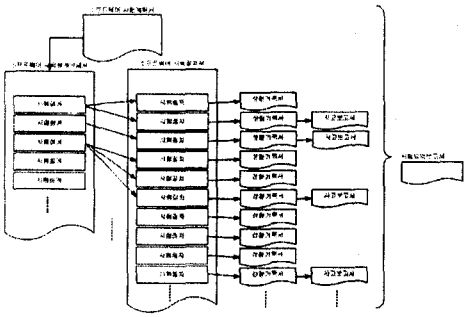

그림 4 시험관련 문서

3.5 소프트웨어 등급별 시험방법

SMART MMIS 소프트웨어 시험은 소프트웨어 등급별로 시험대상과 시험방법은 표 1과 같이 다르게 적용한다.

표 1 소프트웨어 등급별 시험단계
규제기관 권고사항(△)

소프트웨어 등급 / 시험단계	안전-필수	안전-관련	비안전
단위시험	인터페이스 시험, 크기 및 타이밍 시험, 제어 및 데이터 흐름 시험, 구조시험, 기능시험 회귀시험,	인터페이스 시험, 크기 및 타이밍 시험(△), 제어 및 데이터 흐름 시험(△), 구조시험, 기능시험 회귀시험,	크기 및 타이밍 시험(△), 제어 및 데이터 흐름 시험(△), 구조시험(△), 기능시험 회귀시험,
모듈통합 시험	인터페이스 시험, 크기 및 타이밍 시험, 구조시험, 기능시험, 통계시험(△), 응력시험, 회귀시험, 검증시험	인터페이스 시험, 크기 및 타이밍 시험(△), 구조시험, 기능시험, 응력시험(△), 회귀시험, 검증시험	인터페이스 시험, 크기 및 타이밍 시험(△), 구조시험(△), 기능시험, 응력시험(△), 회귀시험(△), 검증시험(△)
계통시험	크기 및 타이밍 시험, 구조시험, 기능시험, 통계시험(△), 응력시험, 회귀시험, 검증시험	크기 및 타이밍 시험(△), 구조시험, 기능시험, 응력시험(△), 회귀시험, 검증시험	크기 및 타이밍 시험(△), 구조시험(△), 기능시험, 응력시험(△), 회귀시험(△), 검증시험(△)
MMIS 통합시험	기능시험, 응력시험, 회귀시험, 검증시험	기능시험, 응력시험, 회귀시험, 검증시험	기능시험, 응력시험, 회귀시험, 검증시험
현장인수 시험	기능시험, 응력시험, 회귀시험, 검증시험	기능시험, 응력시험, 회귀시험, 검증시험	기능시험, 응력시험, 회귀시험, 검증시험
시운전시험	SMART 플랜트 시운전 계획에 귀속됨		

표 1은 소프트웨어 안전성과 신뢰성을 확보하기위해 규제기관에서 요구한 최소한의 시험방법을 시험단계별로 적용한 것으로 SMART MMIS 소프트웨어는 이를 기준으로 소프트웨어의 안전성과 신뢰성을 확보한다.

표 1에서 알 수 있듯이 SMART MMIS 소프트웨어는 시험단계와 소프트웨어 등급에 따라서 시험방법을 달리하고 있으며, 특히 안전-필수 소프트웨어의 경우는 모든 시험방법을 적용한다.

4. 결 론

본 논문에서는 소프트웨어 시험과 관련된 인허가 규제요건을 분석하고, SMART MMIS의 고품질 및 고신뢰도 소프트웨어를 개발하기 위하여 RG 1.170과 RG 1.171에서 제시하고 있는 내용과 KEPIC QAP 원자력 품질보증, ENB 6370, 그리고 SMART MMIS 확인 검증 절차를 바탕으로 하여 소프트웨어 시험의 정당성을 보장하는 소프트웨어 시험 프레임웍을 제시하였다. 향후에는 고신뢰도와 안전성이 요구되는 소프트웨어의 시험 및 평가 방법론에 대한 많은 연구가 수행되어야 할 것이다.

참고문헌

[1] KEPIC QAP-1, "원자력 품질보증계획 일반기준", 대한전기협회, 2000.
[2] KEPIC ENB 6370, "안전계통 디지털 컴퓨터", 대한전기협회, 2000.
[3] 이장수 , "원전계측제어고신뢰도소프트웨어 확인/검증기술현황", Journal of Korean Nuclear Society, 1994
[4] Regulatory Guide 1.152, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants.", Rev.01, 1996.
[5] Regulatory Guide 1.170, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.", 1997.
[6] Regulatory Guide 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.", 1997.
[7] IEEE Std 829, "IEEE Standard for Software Test Documentation", 1998.
[8] IEEE Std 1008, "IEEE Standard for Software Unit Testing", 1987.
[9] IEEE Std 1012, "IEEE Standard for Software Verification and Validation Plans", 1986.