Proceedings of the 14th
KACC, October 1999

원전 디지탈 제어 계통 소프트웨어의 확인 및 검증 방법론 고찰

A Study for Methodology of Software Verification and Validation of Digital I&C System on Nuclear Power Plant

°이 병 윤*, 김 동 욱**, 신 창 훈, 문 홍 주, 윤 명 현, 이 성 우

* 전력연구원 원자력연구실(Tel : 81-042-865-5664; Fax : 81-042-865-5504 ; E-mail: bylee@kepri.re.kr)
** 전력연구원 원자력연구실(Tel : 81-042-865-5667; Fax : 81-042-865-5504 ; E-mail: dwkim@kepri.re.kr)

Abstract : In this paper, the methods and procedures of Software verification and validation are considered. For the development of the digital I&C system in the nuclear power plant, the safety and reliability are the most important properties to be required. The software verifications and validations are the activities to guarantee the correctness and completeness of the process and results of software development under the software quality assurance plan.

Keywords: S/W Verification & Validation, S/W Engineering, Digital Control System, Nuclear Power Plant, I&C(Instrumentation and Control)

1. 서론

최근 선진국의 신규 원자력 발전소 및 국내 차세대 원전 개발 과정에서 원자력발전소의 계측제어 계통에 대한 성능향상과 운전원의 인적 실수를 줄여서 발전소 이용률을 획기적으로 향상시키기 위한 방안으로 기존의 아날로그 제어 방식에서 첨단의 디지털 제어 방식으로 전환하고, 주제어실 설계개념도 기존의 U자형 제어반 개념에서 워크스테이션을 이용한 콤팩트 제어실 개념으로 집중 개선하고 있다. 뿐만 아니라 기존 가동원전의 아날로그 계측제어 설비의 노후화에 따른 오동작 가능성의 증대와 기기 단종으로 인한 예비품 공급의 어려움이 예상되어 성능, 유지보수성 등 여러 가지 측면에서 비교 우위에 있는 디지털 제어 시스템으로의 전환이 점차 추진되고 있는 추세이다.

원자력 발전소의 그 특성상 신뢰성과 안전성은 원전 설비에서 우선 고려 대상이다. 제어시스템 개발시 디지털 제어 시스템의 공통모드 고장(Common Mode Failure)에 대비하기 위하여 심층방어(Defense In Depth)와 다양성(Diversity)을 주요 설계개념으로 요구하고 있으며[4] 시스템의 개발결과가 엄격히 목표에 부합되게 하기 위해 개발과정의 확인(Verification) 및 개발 결과물의 엄격한 검증(Validation)을 요구하고 있다.

이러한 과정에서 신기술 적용에 따른 원전의 안전성 및 신뢰성 확보를 위해 디지털 H/W와 S/W의 품질 보증에 따른 확인 및 검증(Verification and Validation)은 매우 중요한 현안이 되고 있다. 또한, 원전 디지털 제어 시스템의 확인 및 검증 절차와 방법론에 대해서 선진국에서 수행한 몇 가지 사례가 있으나 원자력 규제기관(KINS)이 인정하는 체계적인 절차와 방법론이 현재까지는 없다. 본 논문에서는 소프트웨어 확인 및 검증의 기본 원칙과 기본 수행 방법을 여러문헌과 관련 법규 및 표준을 통해 살펴보고 원전 디지탈 제어 시스템 의 소프트웨어 개발을 위하여 필요한 확인 및 검증 방법을 고찰해 보고자 한다.

2. 원자력 분야의 소프트웨어 확인 및 검증

1. 원자력 분야의 소프트웨어 확인 및 검증 소개

1960년대와 1970년대를 거치면서 디지털 기술은 항공산업, 국방산업, 통신산업 그리고 에너지 산업등에서 점차 중요한 역할을 수행하게 되었다. 디지털 시스템이 중요하고 복잡한 응용분야로 점차 영역을 넓혀가게 되자 개발자와 사용자들은 시스템의 예측가능성 및 신뢰성을 확보해야만 했다. 이것은 개발할 시스템이 무엇을 해야 하는가부터 개발된 시스템이 목적대로 동작하는가까지의 관련된 모든 사항들을 포함하고 있다. 이러한 배경으로 소프트웨어 공학(Software Engineering)이라는 기존의 전통적인 엔지니어링의 체계적인 접근방법과 유사한 영역이 출연하게 되었는데, 소프트웨어 확인 및 검증(S/W Verification and Validation)은 소프트웨어 공학(S/W Engineering)의 소프트웨어 품질관리(S/W Quality Control)의 한 분야로서 안전성(Safety)과 동작성공(Mission Critical)이 매우 중요한 군사용 소프트웨어의 품질을 향상시키기 위한 방법으로 그 개념이 처음으로 고안되었다.

원자력분야에서는 1990년대에 들어서야 디지털 계측제어 시스템으로 전환이 본격화되었고 원자력 규제기관에서도 디지털 설비의 높은 효용성을 인정하면서 공공의 안전에 미치는 영향에 대해서 조심스럽게 고려하게 되었다. 현재, 소프트웨어 확인 및 검증을 위한 많은 방법 및 도구가 있으며 주요 디지털 시스템의 품질을 통제하기 위한 업계 표준도 상당부분 진척되고 있다.

소프트웨어의 확인 및 검증은 개발과정과 개발결과물을 동시에 고려해야한다. 철저한 시험을 통해서 최종 결과물의 품질을 점검할 수 있지만 현실적으로 개발자나 사용자가 모든 경우를 고려하고 모든 경우에 대해서 시험해 볼 수 없기 때문에 최종 결과물의 시험을 통한 품질검사만으로는 충분하지 못하다. 더군다나, 현재까지는 안전성이 매우 중요한 응용시스템(Safety Critical Application System)의 소프트웨어의 신뢰성을 측정할 수 있는 객관적인 방법이 없다. 따라서, 소프트웨어에 대해 보다나은 신뢰도를 확보하기 위해서는 개발 과정에 대한 철저한 확인 및 검증을 통해서 신중한 설계, 개발 단계별 빈도 높은 확인, 철저하고 숙련된 개발 프로세스를 따라야 한다[3].

2. 소프트웨어 확인 및 검증의 정의

소프트웨어 확인 및 검증의 정의는 아래(IEEE Standard 610.12-1990)와 같다.

2.1 확인(Verification)

각 개발 단계(Software Life Cycle)에서 발생한 결과물이 시작될 때 부과된 조건을 만족하는가를 결정하기 위해 시스템이나 구성요소를 평가하는 프로세스이다.

2.2 검증(Validation)

최종 개발단계에서 전체 시스템이나 구성요소에 대해 명시된 요구사항들을 만족하는가를 평가하는 프로세스이다.

3. 소프트웨어 확인 검증 수행의 원칙

소프트웨어 확인 및 검증은 개발 대상의 특성, 개발 조직의 특성 등에 따라서 여러 가지로 접근할 수 있지만 기본적으로 확인 및 검증을 위해서 아래의 기본 원칙을 준수해야 한다[1].

3.1 소프트웨어 생명 주기모델(Software Life Cycle Model)을 통한 확인 및 검증 계획 수립

소프트웨어의 확인 및 검증은 소프트웨어 개발 생명주기(Software Life Cycle)를 통해서 이루어져야 한다. 소프트웨어 개발 생명주기 모델은 그림 1과 같은 폭포수모델(Waterfall Model), 단계별 구현 모델(Phased Implementation Model), 나선형 모델(Spiral Model)등의 여러 가지가 있으며 가장 적합한 모델을 선택하여 개발계획과 그에 따른 확인 및 검증 계획을 수립한다.

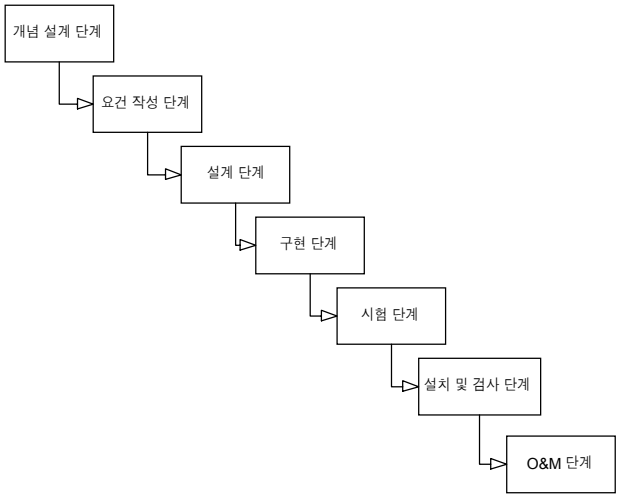

그림 1. 폭포수 모델(Waterfall Model)

3.2 계획에 따른 소프트웨어 확인 및 검증

소프트웨어의 확인 및 검증활동은 반드시 선행 계획(Advance Planning)을 따라야 한다. 계획 작성을 위한 첫 번째 단계는 시스템과 소프트웨어에 필요한 완성도 레벨을 결정하기 위한 전체 플랜트의 안전성과 동작에 대한 중요도를 확인하는 것이다. 그런 다음 요구되는 완성도를 제공하기 위해 필요한 일련의 확인 및 검증 활동과 기술적인 접근방법이 포함된 체계적인 계획을 수립하여 확인 및 검증 활동을 수행한다.

3.3 개발과정과 개발 결과물을 대상으로 하는 소프트웨어의 확인 및 검증활동

소프트웨어의 확인 및 검증은 개발과정과 개발결과물을 동시에 고려해야 한다. 개발과정은 개발과정의 감시, 감독 혹은 조사 등을 통하여 계획에 따라 수행하고 시험을 통하여 개발 결과물을 검증한다.

3.4 소프트웨어 개발의 각 단계 결과물들의 추적성 확보

소프트웨어 확인 및 검증 활동은 소프트웨어 요건을 추적 가능하게 해야한다. 확인 및 검증활동의 정의 자체에서 나타나듯이 요건(Requirement)에서 요구하는 사항을 만족하는지 확인해야하므로 각 설계사양(Design Specification)이나, 결과물(Results)들에 대한 해당 상위 요건의 추적이 가능해야한다. 특정 결과물의 모순이나 오류가 발생했을 때 상위결과물들의 건전성도 함께 파악할 수 있다.

3.5 고신뢰성 소프트웨어의 개발 투명성

안전성과 동작에 매우 중요한 S/W는 개발자와는 독립적으로 평가되고 시험될 수 있어야 한다. 즉, 확인 및 검증 수행자가 개발자의 도움없이 요건, 설계, 코드, 그리고 시험을 이해할 수 있어야 한다.

3.6 소프트웨어 확인 및 검증 수행자의 독립성

확인 및 검증 수행자는 개발자와는 독립적으로 개발자의 가정, 결론에 대해 비판할 수 있어야 한다. 독립성은 개인적 독립성(Individual Independence)과 조직적 독립성(Organizational Independence)이 있는데 개인적 독립성은 개발자와 같은 조직에 속해 있지만 업무적으로 개발 업무와 분리되어 있는 것이고 조직적 독립성은 보다 엄격한 것으로서 개발자와 확인 및 검증 수행자가 서로 다른 조직에 속해 있는 것을 말한다.

3.7 소프트웨어 확인 및 검증 수행의 효율성

확인 및 검증 활동은 고가치성 고위험성이 있는 분야에 더 집중되어야한다. 대부분의 시스템은 복잡하거나 성능의 불확실성이 있거나 혹은 고장 시 중대사고를 유발하는 몇 개의 기능을 제외하고는 일반적인 기능으로 구성되어 있다. 확인 및 검증 자원을 효율적으로 이용하여 목적을 달성하기 위해서는 일반적인 기능 외에 특정한 기능을 하는 부분에 집중적인 확인 및 검증 작업이 이루어지게 하는 것이 좋다

3.8 소프트웨어 확인 및 검증 수행의 문서화(Documentation)

확인 및 검증 작업을 통해서 밝혀진 시스템의 모순점들은 문서화되어야 한다. 모든 기술적인 문제점들의 발생과 해결이 문서화됨으로서 개발된 시스템에서 해결된 기술적 문제점들이 명시화되게 된다. 또한 문서화를 통하여 개발과정에서 흔히 발생되는 문제의 경향을 파악할 수 있고 향후 개발 과정의 품질을 향상시킬 수 있는 근거자료로서의 역할을 할 수 있다.

4. 원전 계측 제어 시스템의 소프트웨어 분류

원자력 발전소에서 디지털 제어 시스템의 응용범위는 매우 넓기 때문에 소프트웨어의 설계, 확인 및 검증 등의 방법을 정립하여 적절하게 적용하기 위해서는 소프트웨어의 특성에 따라 적절히 분류하여 대상의 범위를 적절히 줄이는 것이 필요하다[2]. 소프트웨어의 기능 및 안전 영향도가 해당 시스템의 위험성을 적절히 반영하고

있는지 분석하고, 그것에 근거한 다단계적 분류 기준을 마련하여 신뢰성을 유지하면서 비용 면에서도 효과적인 개발 및 관리를 하는 것이 바람직하다.

5. 확인 및 검증 수행 방법론

원자력 발전소의 계측제어 시스템의 소프트웨어 확인 및 검증을 위한 방법, 테크닉 그리고 소프트웨어 도구는 매우 광범위하고 그 종류는 크게 검토 및 검사(Review and Inspection), 분석(Analysis) 그리고 시험(Testing)의 3가지로 나뉠 수 있다.

5.1 검토 및 검사(Review and Inspection)

검토 및 검사(Review and Inspection)는 독립 검토(Independent Review), 독립 참관(Independent Witnessing) 및 검사(Inspection)로 나뉠 수 있다.

독립 검토(Independent Review)는 개발과정의 감시(Process Audit)와 기술적인 검토(Technical Review)의 두가지관점에서 이루어질 수 있다. 개발과정의 감시는 개발 과정에서 절차상의 오류가 발생하지 않도록 감시하는 것이며 기술 검토는 개발 대상에 초점을 둔 검토 방법이다. 기술검토는 제안되고 행해지는 기술적인 해결책이 전문가의 입장에서 충분히 요건과 사양(Requirement and Specification)을 만족시키는지 검토되어야 한다.

조사(Inspection)는 검토(Review)의 일반적이고 넓은 범위에 대한 기술적 접근보다 세분화되고 심도있는 기술적 접근을 의미하고 수행자로 하여금 보다 깊은 전문성과 과제에 대한 깊은 지식을 요구한다.

독립 참관(Independent Witnessing)은 확인 및 검증 수행자가 특정 기술이나 과제의 특성을 이해하지 못하는 경우 해당 확인 및 검증 활동의 수행에 참관하는 것으로 대체하는 것을 말한다.

5.2 분석(Analysis)

분석 방법은 소프트웨어의 구성요소가 어떠한 방법으로 표현되는지에 대해 매우 종속적이다. 설계문서를 분석하는 경우 자연어로 씌어진 문서보다 체계적인 형태를 갖춘 표, 데이터베이스, 그라프 등으로 표현된 문서가 다양하고 폭넓게 그리고 정확하게 분석될 수 있다. 소프트웨어의 요건을 기술하는 방법을 정하는 것은 분석에 있어서 매우 중요하다. 대부분의 경우 처음에 사용한 기술방법이나 분석도구가 향후 개발 단계에도 계속해서 영향을 미치기 때문이다. 요건(Requirement)을 엄격한 형태로 기술하는 것 자체가 요건분석의 중요한 부분 중의 하나이다.

요건(Requirement) 기술 방법은 자연어 기술(Natural Language Text), 구조적 방법(Structured Methods), 정형방법(Formal Methods) 그리고 준 정형방법(Semi formal Methods)등이 있다.

자연어 기술방법(Natural Language Text)은 현재까지 가장 널리 쓰이는 형태로 각 요건들을 자연어의 형태로 기술하는 방법이다. 하지만, 시스템의 요건이 복잡하고 기능의 중요성이 커질 수록 정확성과 완성도면에서 부적합하다.

구조적 기술 방법(Structured Methods)은 자연어 가술 방법의 불명확성을 극복하기 위해 70년대 이후로 개발된 방법이다. 구조적 기술 방법은 상의 하달식 접근방법(Top-down Approach)을 주로 채택하고 있으며 우선 넓은 의미의 사용자 요구사항을 먼저 포함하고 자세한 사항에 대해서 점차 세분화해서 작성한다. 표현방법은 그라프형태로 나타내며 기능, 데이터의 흐름, 데이터의 저장 그리고 기본적인 수준의 프로시져(Procedure, Function)의 정의를 하게 된다. 최근에는 객체지향 설계기법을 바탕으로 한 요건 기술방법이 활성화되고 있으며 UML(Unified Modeling Language)과 같은 것이 있다.

정형적 방법(Formal Methods)은 시스템의 성질을 수학을 기반으로 하는 기술방법을 말한다. 따라서, 그 특성상 추상적(abstract)이고 명백한 의미를 가진다. Z나 VDM과 같은 정형언어(Formal Language)는 정확성, 자동화된 분석법 등 다양한 장점이 있지만 적용을 위해서는 수학적 전문성, 숙련성이 요구되는 단점이 있다. 이에 비하여 준 정형언어는 자연어 기술 방법에 비해 정형언어가 가지는 정확성, 명백성 등의 장점을 상당부분 가지면서 활용을 위한 사전요건으로 수학적 전문성이 많이 요구되지 않는다.

5.3 시험(Testing)

시험은 기능 시험(Functional Testing), 구조 시험(Structural Testing) 그리고 기타 시험의 크게 3가지의 영역으로 나뉜다.

기능 시험(Functional Testing)은 요건(Requirement)에서 기술된 사실대로 주어진 입력에서 정확한 출력만을 확인하는 것으로 내부적으로 그 구현이 어떻게 되어 있는지는 확인하지 않는다. 하지만, 구조적 시험은 내부적으로 어떠한 구조로서 구현되었는가를 확인하는 것으로 기능시험에서는 시험할 수 없는 부분을 보충 시험하게 된다.

기능시험은 상세한 내부구현 보다는 요건 단계에서 정해지는 기능에 대해 시험하는 것이기 때문에 시스템 레벨에서의 검증 시험 사례(Validation Test Cases)는 시스템 요건(System Requirement)으로부터 작성되어 질 수 있고 그 하위의 구성요소에 대한 기능 시험(Verification Test)은 해당요소의 기능이 세분화되고 정의되는 대로 구현 전에 시험 사례(Verification Test Cases)를 작성할 수 있다. 모든 작성된 시험 사례(Test Cases)들은 요건(Requirement)과 추적 가능하게끔 하여 모든 요건들이 시험되었는가를 확인 할 수 있어야 한다.

구조시험의 시험 사례(Test Cases)를 작성함에 있어서 프로그램의 모든 가능한 경로를 한번 이상 포함해서 소프트웨어의 구조적 건전성을 확인할 수 있도록 해야 하는 것이 좋으나 현실적으로 비경제적일 수 있다.

3. 원자력 발전소 디지털 제어 시스템 개발을 위한 확인 및 검증 계획 작성

지금까지 원자력 발전소의 디지털 제어 시스템을 위한 확인 및 검증에 일반적인 기술 현황과 방법론 등에 대해서 기술하였다. 확인 및 검증은 사전 계획 및 절차에 따라 수행하는 것이 필수적이므로 그 계획 작성이 매우 중요하다. 소프트웨어의 확인 및 검증 계획은 전체 개발 계획과 밀접한 관계가 있으며 개발 계획작성의 일부라고 볼 수 있고 계획작성 시 전체적인 목표와 조직에 부합하도록 하는 것이 중요하다. 다음과 같은 순서에 의해 확인 및 검증 계획을 수립한다

1. 개발 계획

소프트웨어 확인 및 검증 활동 계획의 기초는 개발계획이다. 개발계획 수립 시 개발 대상과 개발 조직에 적합한 개발 모델(Software Life Cycle Model)을 선정해서 작성한다. 또한, 개발하고자하는 소프트웨어에 대해 적절한 등급을 부과하여 요구되는 확인 및 검증 활동을 확인할 수 있도록한다.

2. 품질 보증 계획

소프트웨어 확인 및 검증 활동은 소프트웨어 품질 관리 활동의 한 영역이라고 할 수 있다. 따라서, 품질 목표에 도달하기 위한 절차와 계획을 담은 품질 보증 계획은 소프트웨어 확인 및 검증 계획 이전에 선행작성 되어야 할 문서이며 확인 및 검증 활동 영역을 전체 품질 목표 달성을 위한 수단으로서의 기능과 조직을 적절히 기술하고 있어야 한다. 품질 목표를 설정하고 품질 보증 활동을 함에 있어 필수적인 것은 품질의 측정 및 평가와 그 결과를 품질 목표 달성을 위해 반영하는 것이다. 이를 위해서는 품질 측정 척도(Metrics)와 측정방법(Measure)이 필요하지만 현재까지 이의 표준적인 방법이 마련되지 않고 있으며 현재 표준화 작업이 상당히 진행 중이다.

원자력 안전 기술원(KINS)에서 미국의 NUREG-0800의 Safety Regulatory Plan Chap 7.0을 기반으로 작성된 "경수로형 원전 안전 심사 지침 제7장 계측 및 제어"의 안전 심사 기준에 정의된 평가기준을 소프트웨어의 품질 평가 기준의 일부로 적용한다. 이 지침은 원자력 계측제어 계통 설비의 인허가를 위해서 디지털 시스템이 반드시 확보해야할 품질을 기술하고 있으며 이는 원전 계측 제어 시스템이 갖추어야 할 품질의 최소 요구사항이라고 할 수 있다[5].

3. 개발 조직, 책임 및 권한

원전 디지털 제어 시스템 개발을 위한 조직은 크게 확인 및 검증 조직을 포함한 품질 보증조직과 기기의 개발과 구현을 담당하는 개발 조직으로 구성한다. 확인 및 검증 활동의 기본 원칙 중의 하나인 조직적 독립성을 적용한다. 안전성품목 등급의 경우 개인적 독립성보다 조직적 독립성을 적용함으로서 확인 및 검증 활동의 독립성을 명백히 한다.

확인 및 검증 수행 조직은 확인 및 검증활동 수행 책임자와 확인 및 검증 활동 수행을 위한 전문가로 구성한다. 책임자는 확인 및 검증 활동 관리, 수행 인력, 조직 관리의 책임과 권한이 있다. 확인 및 검증활동 수행자는 소프트웨어의 확인 및 검증 활동을 수행하는데 필요한 전문지식과 경험을 갖춘 인력을 운영하며 확인 및 검증 활동 업무 할당은 확인 및 검증 계획에 맞추어서 책임자가 할당하고 업무를 관리한다.

4. 확인 및 검증 활동 관리

확인 및 검증 활동의 기록 및 보고에 대한 체계적인 기준과 보고서 작성 방법, 조치방법, 적용 코드 및 표준, 문제 발생보고, 문제 해결, 및 확인 및 검증 계획 변경 방법 등 활동 중에 발생하는 여러 가지 사건에 대해 체계적으로 조치할 수 있는 절차를 수립한다.

5. 각 단계별 상세 확인 및 검증 계획 수립

각 단계별 V&V 상세 계획은 전체 시스템 V&V 계획에 준하여 작성하게 되는데 각 단계 수행 직전까지 완성시키면 된다. 각 단계별 V&V 상세 계획은 각 단계별 V&V 대상 확인/명시, 각 V&V 대상별 V&V Task 확인/명시, 각 Task가 전체 V&V 목적 달성 기여하는 점, 각 V&V Task를 수행하는데 필요한 세부 방법/절차, 각 V&V Task를 수행하는데 필요한 입력자료(Source, 형태) 및 결과물 명시, 각 단계별 V&V 수행 일정(Schedule), 각 V&V Task를 수행하는데 필요한 자원을 분류하여 명시(인력, 장비, S/W 등), 각 V&V Task 수행에 대한 위험도와 가정들을 명시, 그리고 각 V&V Task에 수행 책임 명시한다.

4. 결론

지금까지 원자력발전소의 디지털 계측 제어계통의 소프트웨어 개발을 위한 확인 및 검증 활동에 살펴보았다. 확인 및 검증 활동은 개발 과정과 결과물을 모두 고려하는 것으로서 정해진 확인 및 검증의 사전 계획에 따라 수행하게 된다. 대상 소프트웨어의 안전성 및 신뢰성에 의해 요구되는 품질에 따라 확인 및 검증을 위한 방법의 정형성, 조직의 독립성, 수행 절차 및 방법이 결정되므로 개발 대상 소프트웨어를 적절한 기준으로 분류하여 확인 및 검증 계획을 수립하고 수행한다.

확인 및 검증활동은 원자력 분야에만 국한되는 것이 아니라 소프트웨어 개발과 관련된 모든 분야에 해당되며 원자력 분야와 같이 공공의 안전에 큰 영향을 줄 수 있는 분야에서는 확인 및 검증 활동의 중요성이 강조된다. 현재까지 원자력 분야를 위한 표준화되고 체계적인 소프트웨어의 확인 및 검증 방법이 없는 실정이며 확인 및 검증을 위한 몇가지 원칙과 일반적이고 광범위한 내용을 담은 산업표준이 있다. 하지만, 최근 소프트웨어 개발을 관리하고 지원하는 소프트웨어 공학의 발전과 이에 따른 CASE(Computer Aided Software Engineering) 도구의 폭넓은 보급으로 확인 및 검증 활동이 용이해지고 있다.

참고문헌

[1] S. Levy Incorporated, *Handbook for Verification and Validation of Digital Systems*, EPRI, Dec., 1994.

[2] N. Storey, *Safety-Critical Computer Systems*, Addison-Wesley, 1996

[3] R. S. Pressman, *Software Engineering*, McGraw-Hill, 1992.

[4] N. G. Leveson, Safeware, Addison Wesley, 1995.

[5] 한국원자로안전기술원, *경수로형 원전 안전 심사 지침 제7.0절, 1998. 5.*