'99 추계학술발표회 논문집
한국원자력학회

원전 소프트웨어 개발의 소프트웨어 공학 기법 적용
Appliance of Software Engineering in Development of Nuclear Power Plant

백영우, 김희철, 윤 청
충남대학교
대전시 유성구 궁동 220

김복렬
한국원자력안전기술원
대전시 유성구 구성동 373-1

요 약

원자력 산업계에서 컴퓨터 기술의 적용은 다른 응용 분야와 마찬가지로 요구되는 시대적 변화라 할 수 있다. 그러나 현재까지 원자력 분야에 컴퓨터 적용이 용이하지 않았던 주된 요인은 이러한 기술의 적용이 원자력 분야의 안전성에 미치는 잠재적인 영향 때문이라고 해도 과언이 아니다. 그리고 가동중인 원자력 발전소의 노후화된 계측제어 설비를 교체하는 디지털 개선사업과 차세대 원자력 발전의 디지털 계측제어 시스템에 대한 안전성, 신뢰성을 얻고 품질보증을 평가할 수 있는 지침의 설정 및 규제 기술을 개발하는 것이 시급한 과제가 되고 있다. 따라서 소프트웨어 개발과 운영에 있어 기술적인 한계를 극복하고 품질을 보증할 수 있는 방안들이 마련되어 일관된 틀과 방향이 제시되어야 한다. 본 논문에서는 기존의 폭포수 모델과 원전 소프트웨어 개발에 필요한 소프트웨어 생명주기의 차이점을 살펴보고 원자력발전소의 계측제어 시스템 개발에 필요한 일관된 틀을 제시하고자 한다.

Abstract

Application of computer technology in nuclear power plant is also a necessary transformation as in other industry fields. But until now, application of software technology was not wide-spread because of its potential effect to safety in nuclear field. It is an urgent theme to develop evaluation guide and regulation techniques to guarantee safety, reliability and quality assurance. To meet these changes, techniques for development and operation should be enhanced to ensure the quality of software systems. In this study, we show the difference between waterfall model and software life-cycle needed in development of nuclear power plant and propose the consistent framework needed in development of instrumentation and control system of nuclear power plant.

1. 서론

시스템 제어가 소프트웨어로 이루어지는 경우 하드웨어를 사용하는 것보다 오류를 발견하기 어려

워 사람의 안전과 관련된 시스템의 경우 소프트웨어를 사용하는 것에 많은 저항이 있어 왔다. 안전성이 중요시되는(Safety critical) 원자력 발전소의 계측제어 시스템에 필요한 소프트웨어의 경우에도 품질을 보증하고 안전성 및 신뢰성을 확보하는 문제가 중요한 쟁점으로 떠오르고 있다. 이에 컴퓨터 기술을 적용하여 디지털 계측제어 시스템을 개발하는데 있어 품질을 보증할 수 있는 표준 개발 체계를 확립하는 것이 요구되고 있다. 이를 위해 국제 표준에 맞는 문서화된 품질 시스템의 개발은 필수적이다.

현재 소프트웨어 개발의 표준을 제시하는 국제표준기구(ISO : International Standard Organization), IEEE의 표준 요구사항을 반영하여 원전 소프트웨어 개발에 요구되는 프로세스 및 품질을 보증 할 수 있는 기본 방향을 제시하고 있다.

본 논문에서는 고전적 라이프 사이클 패러다임인 폭포수 모델과 원전 소프트웨어 개발의 소프트웨어 생명주기와의 차이점을 제시하며 일관된 틀을 제시하고자 한다.

2. 고전적 라이프 사이클 패러다임(classic life-cycle paradigm)

고전적 라이프 사이클 패러다임은 폭포수 모델(Waterfall Model)이라고도 하며 다른 공학에서도 많이 사용되고 있는 전형적인 기법이다. 폭포수 모델은 요구사항 분석, 설계, 구현(프로그래밍), 시험 및 유지보수 순서로 시스템의 개발이 이루어진다. 소프트웨어 일부분인 경우, 시스템 차원에서 분석이 이루어져 기능이 일부가 소프트웨어에 할당된다. 이 경우 전체 시스템에 대한 연구가 우선된 후에야 소프트웨어와 다른 시스템 요소들과의 인터페이스가 정의된다.

폭포수 모델은 소프트웨어 개발을 단계적이고 체계적이며 순차적인 접근 방법을 사용하여 정의하고 있으며, 가장 오래되고 널리 사용되는 패러다임이다. 이 모델은 개념 정립에서 구현까지 하향식(top-down) 접근방법을 사용하여 높은 추상화(abstraction) 단계에서 시작하여 낮은 추상화 단계로 옮겨하고 있다. 폭포수 모델의 개발체계는 아래 그림과 같다.

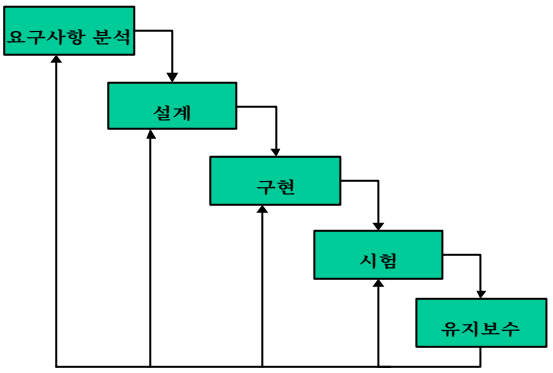

그림 1 폭포수 모델

3. 원전 소프트웨어에 적용되는 소프트웨어 생명주기

소프트웨어 공학에 대한 기본 개념 및 계측제어 시스템 개발의 기본 원리를 안전성이 중요시되는 원자력 발전 디지털 계측제어 시스템 개발에 적용하여 높은 품질의 시스템을 개발할 수 있는 틀을 확립해야 하고 소프트웨어 품질을 보증하기 위해 소프트웨어 공정과정인 분석, 설계, 구현, 시험 및 유지보수 단계에 적용될 수 있는 기법과 도구들이 확립되어야 하고, 각 공정 과정의 임무, 입력물,

산출물, 사용도구 등도 정의되어야 한다.

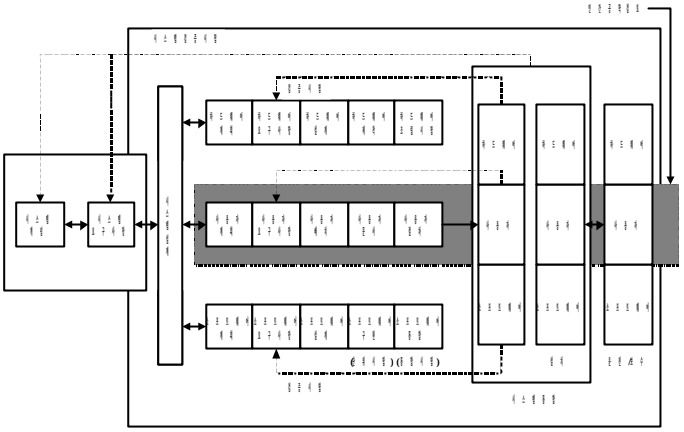

그림 2 시스템 생명주기[3]

소프트웨어의 품질 보증은 소프트웨어가 달성하여야 하는 품질을 이루도록 도와주는 보호 활동으로 품질과 마찬가지로 산출물에 대한 품질 보증과 공정에 대한 품질보증 두 가지가 존재한다.

소프트웨어는 시스템의 한 구성요소이므로 시스템 전반의 이해와 요구사항 및 설계가 이루어진 후에야 기존에 하드웨어 수행하던 작업을 소프트웨어로 대체할 수 있게된다.

3.1 폭포수 모델과의 차이점

시스템의 작동오류가 위험한 결과를 초래할 수 있는 안전성(Safety)이 강조되는 시스템의 경우에는 고전적인 라이프 사이클 패러다임인 폭포수 모델의 개념에다 소프트웨어 생명주기 전 범위에 걸쳐 적용이 되는 안전성 분석(Safety Analysis), 확인 및 검증 작업(V&V : Validation and Verification), 형상관리(CM : Configuration Management) 작업을 추가하여 보다 높은 품질의 소프트웨어를 얻을 수 있다.

원전 소프트웨어 개발에 쓰이는 소프트웨어 생명주기는 아래의 그림과 같다.

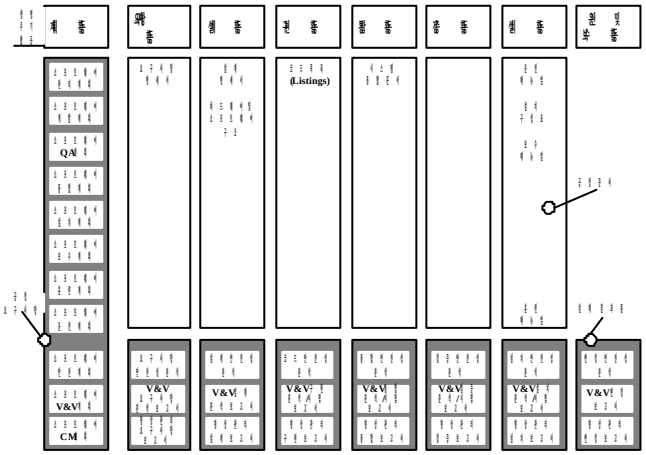

그림 3 소프트웨어 생명주기[3]

원전 소프트웨어 개발에 쓰이는 소프트웨어 생명주기는 기존의 폭포수 모델의 개발주기와는 크게 다르지는 않지만 품질보증활동을 보다 중요시하고 있는 구조를 보이고 있다. 원전 소프트웨어 개발에 쓰이는 소프트웨어 생명주기는 각 공정단계마다 안전성을 분석하는 활동, 소프트웨어가 사용자의 요구사항을 만족시키는가의 여부를 밝히는 활동인 검증(Validation), 개발주기상에서 산출물이 바로 이전 단계의 산출물과 일치하는가를 결정하는 확인(Verification)활동, 소프트웨어 개발 및 유지보수 과정에서 발생하는 각종 결과물(문서, 프로그램, 하드웨어)들에 대한 계획, 개발, 운용 등을 종합하여 시스템의 형상을 만들고, 이에 대한 변경을 체계적으로 관리, 제어하기 위한 활동인 형상관리 활동을 추가한 구조를 보이고 있다.

3.2 소프트웨어 위험분석

소프트웨어 생명주기 전 범위에 걸쳐 적용이 되는 안전성 분석의 한 방법인 소프트웨어 위험분석을 나타내는 그림은 아래와 같다.

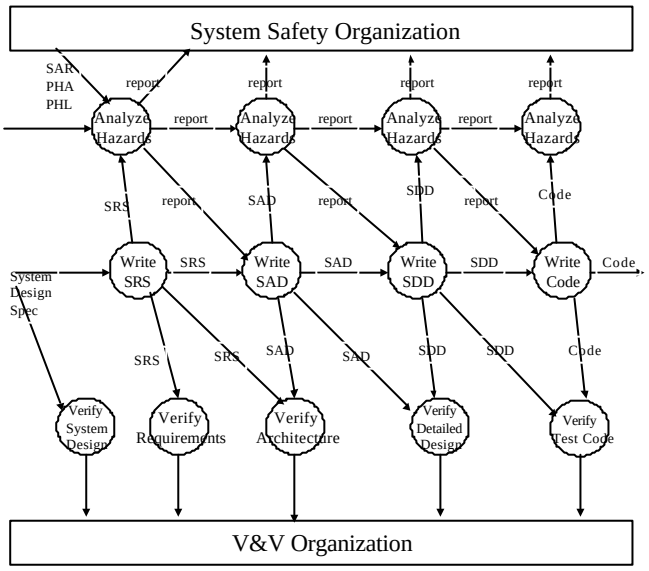

그림 4 소프트웨어 위험분석[6]

소프트웨어 위험분석이란 소프트웨어 위험 그리고 소프트웨어와 시스템의 인터페이스에 관련된 위험을 제거 혹은 제어하기 위해 사용된다. 이러한 소프트웨어 위험분석은 직접적으로 시스템 위험분석과 관련이 되어 있다. 이는 입력으로 시스템 위험분석을 받아들이기 때문이다. 소프트웨어가 시스템의 일부분으로 작용하기 때문에 소프트웨어 위험분석이 중요시되고 있다. 원전과 같이 안전성이 중요시되는 시스템에서는 소프트웨어 생명주기 전 범위에 걸쳐 위험분석과 확인 및 검증이라는 활동을 하게 된다.

각 공정단계의 산출물이 위험분석 활동과 확인 및 검증 활동의 입력물로 들어간다. 하나의 예로 분석단계의 산출물인 요구사항 명세서(SRS : Software Requirements Specification)은 위험분석 활동의 입력물로 들어가며, 또한 확인 및 검증 활동의 입력물로 들어간다. 분석단계에 있어서의 확인 및 검증활동은 요구사항 명세서에 기술되어 있는 요구사항을 평가하고 시스템 요구사항과의 관계를 살펴보며 인터페이스에 대한 분석을 하게 된다. 위와 같은 안전이 중요시되는 시스템에서는 일

관된 생명주기의 틀을 확립하고 이에 따른 품질보증활동을 수행해야만 한다.

4. 결론

안전성이 중요시되는 시스템은 대부분 구성요소들 간에 상호 밀접하게 연결되어 있고 매우 복잡한 상호작용을 갖는다. 따라서 어느 한 부분에서의 오류가 전체 시스템의 작동에 심각한 영향을 미치게 된다. 더구나 소프트웨어 내에서 발생하는 작은 오류도 매우 큰 파급효과를 가질 수 있다는 점에서 안전성을 확보하는 문제를 더욱 어렵게 만들고 있다.

본 논문에서는 원전 소프트웨어 개발에 요구되는 소프트웨어 생명주기 및 소프트웨어 위험분석 과정에 대해 기술하였다.

앞으로 원전 시스템뿐만 아니라 안전성이 중요시되는 시스템의 일부인 소프트웨어를 개발하는데 있어 품질보증활동에 대한 연구 및 위험분석 방법에 대한 많은 연구가 진행되어야 할 것이다.

참고문헌

[1] 시스템공학연구소, "한국형 정보시스템 개발방법론"

[2] 윤청, "성공적인 소프트웨어 개발 방법론", 생능 출판사, 1996

[3] Laura M. Ippolito & Dolores R. Wallace, "A Framework for the Development and Assurance of High Integrity Software", December, 1994

[4] Laura M. Ippolito & Dolores R. Wallace, "A Study on Hazard Analysis in High Integrity Software Standards and Guidelines, January 1995

[5] J.D.Lawrence, " Software Reliability and Safety in Nuclear Reactor Protection Systems", 1993

[6] J.D.Lawrence, "Software Safety Hazard Analysis" February 1996

[7] 정안나, 염근혁, "객체 지향 소프트웨어 개발을 위한 품질 보증 시스템 프레임워크" 정보과학회 논문지(B) 제 25권 12호 p.1769 - 1778, 1998