|  |
| --- |
| ㈜ 에이블맥스 |
| 위험분석서 초안(BNP&SE) |
| 2017 SW공학기술 현장적용 지원사업 |

|  |
| --- |
| 링크투 주식회사  2017-7-24 |

목차

[I. 위험분석의 소개 2](#_Toc488177769)

[II. SW신뢰성과 안전성 4](#_Toc488177770)

[1. SW 신뢰성과 안전성의 개요 4](#_Toc488177771)

[2. SW 신뢰성에 관련된 국제표준 4](#_Toc488177772)

[2.1. IEC 61508 4](#_Toc488177776)

[2.2. IEC 61508과 각 산업별 파생 규격 5](#_Toc488177777)

[III. 위험분석 프로세스 7](#_Toc488177778)

[1. 위험분석절차 7](#_Toc488177779)

[2. FMEA 기법 8](#_Toc488177780)

[IV. BNP&SE 분석 결과 9](#_Toc488177781)

# 위험분석의 소개

위험분석은 해당 시스템이 어느 정도의 위험을 내포하고 있는지 계량적으로 분석하고, 분석된 결과를 가지고 더욱 신뢰성 있고 안전한 시스템이 될 수 있도록 관리하고자 하는 목적을 갖는다. 신뢰성과 안전성이 바로 위험분석을 통해 달성하고자 하는 목적이며, 위험분석도 정해진 절차에 따라 프로세스를 체계적으로 수행하여 절차에서 누락됨이 없이 수행되어야 할 필요가 있다.

위험분석을 위해 먼저 살펴보아야 할 지식체계로서 SW신뢰성과 안전성이라는 개념부터 이해할 필요가 있으며, 국제표준에 입각하여 객관적으로 증명된 규칙에 따라 위험이 분석되고 계량적으로 증명되어 심각한 문제가 발생하는 사태를 미연에 방지할 수 있다.

안전성의 정의는 국제 안전규격을 위한 가이드인 ISO/IECGUIDE51에 기술되어 있다. 이것에 의하면, 안전성은 ‘수용 할 수 없는 위험성이 없는 것(freedom fromunaccept able risk)’이라고 표현되어 있다. 직역하면, ‘수용 할 수 없는 위험으로부터의 해방’이라는 정의이다. 달리 말하면, 사람 또는 재산에 대한 재해의 위험성이 허용 가능한 수준으로 억제되어 있는 상태라 할 수 있다.

이에 따라 시스템에서는 안전에 대한 개념을 1차적 안전(Primary safety), 기능 안전성(Functional safety), 간접 안전성(Indirect safety) 으로 분리 할 수 있다. 1차적 안전성은 화재, 감전과 같은 하드웨어에 의한 직접적 사고로부터의 안전성으로 정의한 것이고, 기능 안전성은 리스크 평가 측정결과에 따라서 설계 과정을 통해 위험이 제거되는 장비의 안전성을 말하며, 간접 안전성은 데이터베이스 정보에러와 같이 잘못된 정보제공으로 일어날 수 있는 위험원으로부터의 안전성을 정의한다.

소프트웨어의 안전성은 과거 소프트웨어 중심의 프로세스 안전성 관점에서 시스템과 소프트웨어를 동시에 바라보는 개념으로 변화하고 있다. 이 같은 변화는 소프트웨어를 전체 시스템의 일부로 인식하여 전체 시스템 레벨에서 위험성 분석을 선행하고, 시스템 위험분석 결과로 도출된 시스템 안전 요구사항을 바탕으로 소프트웨어 안전 요구사항을 도출해야 한다는 것을 의미한다. 소프트웨어 안전성을 보증한다는 의미는 소프트웨어가 가지고 있는 여러가지 속성 중에서 특히 안전기능(Safety Function)이 올바르게 선정되었는지를 확인 할 수 있어야 하고, 최종 개발 결과물이 수행하는 안전기능이 정상적으로 작동하는가를 확인하는 것이다.

예를 들어 “원자로를 안전하게 보호 할 수 있고, 위험성을 알릴 수 있는 SW를 개발하자”라는 안전요구사항을 기반으로 프로젝트를 시작한다고 하면 먼저 사업계획서를 만들고, 요구사항 명세서를 정리하고, 상세 기술 명세서를 만들고, 소스코드를 만드는 등 여러 엔지니어들의 손과 머리를 거치게 된다. 하지만 작업이 끝난 후엔 처음에 의도했던 바와 전혀 다른 결과물이 나올 수도 있게 된다. 그렇기 때문에 처음에 의도한 안전 요구사항과 최종 결과물의 일치성을 확보하고, 특히 안전기능(Safety Function)이 원래 의도한대로 개발되었는가를 반드시 검사해야 한다.

안전 소프트웨어 개발이라는 목표달성이라는 측면에서 볼 때 가장 중요한 요소는 세가지로 구분된다.

첫째, SW 개발 안전수명주기에 대한 이해이다. SW 개발 안전수명주기는 개발조직내의 모든 개발자가 숙지하고 각 단계별 결과물들을 사전에 계획된 바에 따라 생성해 나가야 하는 것인데 수명주기는 경우에 따라 국제표준에서 제시하는 가이드를 일부 테일러링하여 개발 프로젝트에 최적화 시킬 수 있다.

둘째, 시스템 및 소프트웨어 위험분석기법에 대한 이해가 필요하다. 안전한 소프트웨어 개발은 시스템 위험분석을 통하여 도출된 시스템 안전 기능요구사항을 근간으로 개발이 이루어지기 때문이다. 시스템 레벨에서 도출된 안전 기능요구사항 중에서 소프트웨어가 수행하여야 하는 요구사항들이 소프트웨어 관점에서 명세화되고 이를 바탕으로 소프트웨어가 구현되어야 한다.

셋째, 소프트웨어 안전 기능이 의도한 바와 같이 구현이 되었는지를 확인하는 올바른 검증시험 방법을 적용하는 것이 중요하다. 소프트웨어의 안전성을 검증하는 시험은 디바이스레벨에서 이루어지며, 검증의 목표는 시스템 안전 요구사항 및 소프트웨어 안전 요구사항이 원래 의도와 같이 구현되었는지를 검증하는 것이기 때문이다.

# SW신뢰성과 안전성

## SW 신뢰성과 안전성의 개요

SW 신뢰성을 “SW가 고장이나 결함에서 자유로울 수 있는 정도”라고 정의할 수 있다. 그리고 SW 안전성은 “응용 프로그램에서 원래 사양에 정의된 서비스를 오류 없이 얼마나 정확하게 제공하느냐”에 관한 것이다. 이와 같이 신뢰성과 안전성을 구분하기는 모호하다. 본 글에서는 SW신뢰성과 안전성을 통칭하여 SW신뢰성이라고 하겠다. SW 신뢰성에 대해서는 1990 년대에 미국에서 그리고 2000년대에는 유럽에서 다양 한 국제 규격과 논문, 서적들이 활발하게 발간되었다. 최근에 SW신뢰성이 중요해진 이유는 시스템의 복잡도가 높아지고, 시스템의 가동중단이 사회에 심각한 영향을 미치기 때문이다. 여기서 시스템이라고 하는 것은 하드웨어와 소프트웨어를 모두 포함하는 개념이며, SW신뢰성을 살펴보기 위해, 하드웨어의 신뢰성도 고려해야 할 필요가 있다. 신뢰성은 발생되는 고장 또는 결함이 잔존하는 확률과 관계가 있으며, 하드웨어는 제품의 소모나 외부의 요인에 의한 고장으로 신뢰성이 저하된다. 하지만 소프트웨어는 코드 내에 잘못된 알고리즘의 구현에 기인한다.

## SW 신뢰성에 관련된 국제표준

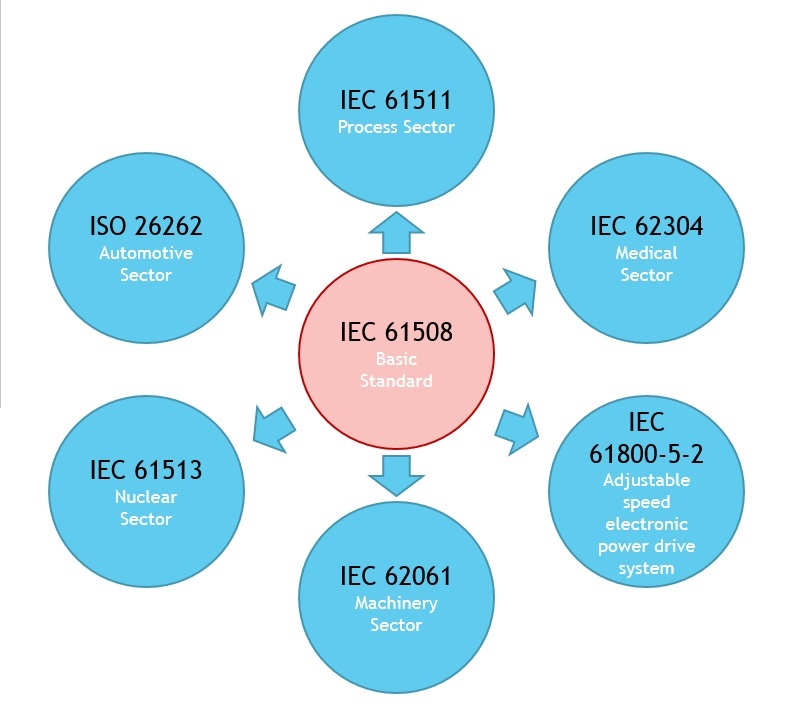
SW 신뢰성을 “특정 한 환경에서 특정한 기간 동안 소프트웨어가 오류 없이 작동할 확률(ANSI/IEEE STD-729-1991)”이라고 한다면 일반적으로 소프트웨어의 신뢰성을 향상시키는 방법은 소프트웨어의 오류를 줄이는 것으로, 소프트웨어의 Verification & Validation, 시험 등의 공학적 기법을 통해 소프트웨어의 오류를 검출하고 제거하는 것이라 고 할 수 있다. 아래에서는 실제로 높은 신뢰도를 요구하고 있는 산업군들을 대상으로 공표되어 있는 국제 규격들을 살펴 보고, 이들 규격에서는 SW 신뢰성을 향상시키기 위해 이러한 관점이 어떻게 반영되어 있는지 살펴보기로 한다.



### IEC 61508

IEC 61508은 SW분야의 기능안전규격에서의 기능안전규격의 가장 기본적인 규격이라 할 수 있다. IEC61508 Electrical/Electronic/Programmable Electronic Safety-Related Systems(E/E/PES)에서는 기능안전성에 대하여 센서, 논리연산자, 액추에이터로 구성된 전기, 전자, 프로그램 가능한 전자장치를 이용하여 안전관련 시스템을 작동시키는 안전기능으로 정의하고 있으며, SIS(Safety Instrumented System)를 고려하여 해당 안전시스템의 기능안전에 대한 타당성을 검증하였다. SIS는 센서와 단일 또는 다중의 제어기로 구성되어 개인, 장비, 환경에 대한 보호 시스템을 의미하며, SIS의 기능안전성은 안전기능성과 안전 무결성으로 구성 되어 있다. 이러한 안전 무결성은 SIL(Safety Integrity Level)을 활용하여 정량적으로 평가한다.

### IEC 61508과 각 산업별 파생 규격



파생규격 중 가장 대표적인 표준규격이 자동차 분야에 적용되는 ISO26262이다.

1. ISO26262

ISO26262는 IEC61508을 바탕으로 자동차분야의 적용을 위해 2011년 11월에 발표된 표준이다. 자동차 ECU의 오작동으로 인한 사고 및 인명손실을 최소화하는 것이 목적이다. ISO26262는명세, 설계, 구현, 통합, 검증, 인증에 이르는 개발 전 단계에서 최신 개발 방법 및 테스트 방법을 적용하도록 하고 있으며 기능안전규격을 준수하기 위한 각 단계에서의 요구사항을 정의한다.

세부적으로는 시스템 레벨, 하드웨어 레벨, 소프트웨어 레벨에서 각 요구사항들이 정의되어 있다. 이는 제품개념단계에서부터 폐기까지 전수명주기에 걸쳐서 전자장치의 고장으로 인한 자동차의 안전성을 저해 할 수 있는 위험을 체계적으로 분석하고 그 위험에 대처하는 수단이 효과적임을 보장해야 하는 것으로서 시스템공학, 하드웨어, 소프트웨어공학, 신뢰성공학 안전성분석 및 프로세스능력이 모두 요구된다.

1. IEC62304

의료기기 기능 안전성 확보를 위해 국제표준이 제정되어 각 나라에서 사용하고 있다. IEC60601 에서는 의료기기에 대한 안전 요구사항을 도출해 안전성을 분석하고 있으며, IEC62304에서는 의료기기의 개발 및 유지보수의 개발 생명주기를 정의하고 각 단계별로 수행해야 하는 활동과 지표 및 산출물을 정의하고 있다. ISO14971은 의료기기 개발 시 필요한 위험분석 방법을 규정하고 있으며 소프트웨어 개발 단계에 이를 적용하고 있다.

어떤 종류의 소프트웨어도 안전성을 100% 보장할 방법은 없다. 의료기기 소프트웨어의 안전성을 안전성을 향상시키는 주요 원칙 세 가지는 위험관리, 품질관리, 소프트웨어 엔지니어링이다. 안전한 의료기기 소프트웨어를 개발하고 유지 보수하려면 적절한 소프트웨어 엔지니어링 방법 및 기술에 대한 전체적인 체제로서 품질관리시스템의 일부로 위험관리를 확립해야 한다. 이러한 세 가지 개념을 조합하면 의료기기 제조업체가 잘 구성되고 일관성 있게 반복 정밀도가 높은 의사결정 프로세스를 사용해 의료기기 소프트웨어의 안전성을 향상시킬 수 있다.

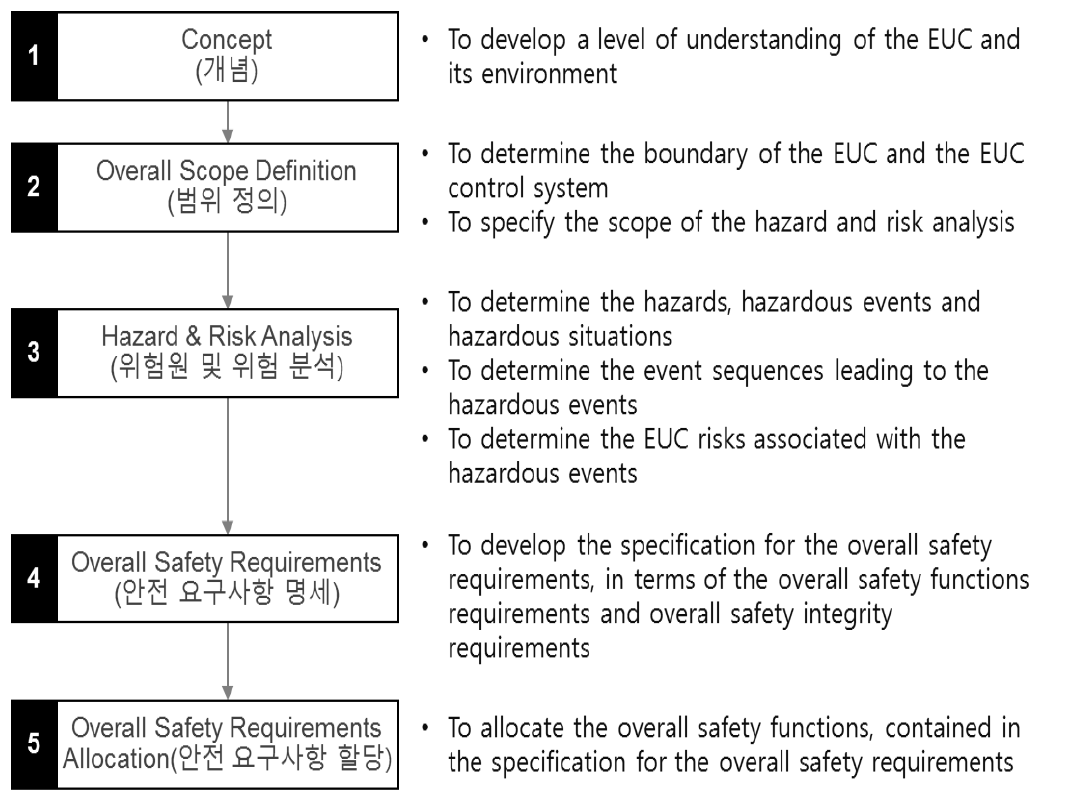
# 위험분석 프로세스

## 위험분석절차

위험성은 위험원의 발생빈도와 심각도의 조합으로서 발생빈도가 빈번하거나 발생빈도가 빈번하지 않더라도 사고가 발생하면 치명적인 결과를 초래하는 위험원은 위험성이 크다고 정의한다. 사고관련 위험원의 위험성을 허용 할 수 있는 수준으로 제어하고자 하는 안전성 관리는 시스템수명주기의 요구사항분석 단계부터 안전계획을 수립하여 도출된 안전요구사항을 바탕으로 안전한 시스템이 되도록 설계, 구현 및 시험을 수행하여 위험원을 지속적으로 관리하고 확인 및 검증을 통해 시스템의 안전성을 확보한다.

안전성 확보를 위한 위험분석절차는 대상제어시스템에서 위험원을 중심으로 결과, 손실, 위험성을 파악하는 위험분석단계와 해당 위험에 대한 원인분석을 통해 위험제거 및 감소를 위한 안전기능을 도출하여 SIL을 할당하고 이를 시스템 하위 서브시스템, HW, SW에 할당하는 단계로 정의 할 수 있다.

IEC61508에서 제시하는 전체 안전수명주기에서 위험분석단계로 아래 그림과 같은 개념정의(Concept), 범위정의(Overall Scope Definition), 위험원 및 위험분석(Hazard & Risk Analysis), 안전 요구사항 명세(Overall Safety Requirements), 안전 요구사항 할당(Overall Safety Requirements Allocation)으로 구성된다.



위험성은 위험원의 발생빈도와 심각도의 조합으로서 발생빈도가 빈번하거나 발생빈도가 빈번하지 않더라도 발생시 치명적인 결과를 초래하는 위험원은 위험성이 크다. 사고관련 위험원의 위험성을 허용 할 수 있는 수준으로 제어하고자하는 안전성 관리는 시스템 수명주기의 위험분석 단계부터 안전계획을 수립하여 도출된 안전 요구사항을 바탕으로 안전한 시스템이 되도록 설계, 구현 및 시험을 수행하여 위험원을 지속적으로 관리하고 확인 및 검증을 통해 시스템의 안전성을 확보한다.

## FMEA 기법

FMEA는 What If 분석을 좀더 체계화한 것이다. 즉, “만약 무슨 일이 벌어진다면 어떻게 될까(what happens if)?" 라는 질문을 염두에 두어 하나의 재료, 부품, 장비 등이 고장 났을 경우 그것이 전체 제품이나 사용자, 혹은 제품기능에 어떠한 영향을 미치는가, 생각의 범위를 점차 넓혀 가면서 상위 수준으로 분석하여 가는 것이다.

그러므로 이 기법은 전형적인 귀납적 분석방법이며 상향적(bottom-up), 정성적인 위험성 분석기법의 대표라고 할 수 있다. 특히 결함과, 다음 상위 수준의 기능적 제품에 미치는 영향과 메커니즘을 연구하는데 적합하다.

IEC812는 이 기법의 목적을 다음과 같이 밝히고 있다.

• 원인이 무엇이든 제품 기능적 구조의 다양한 수준에서, 각각의 규정된 품목의 고장모드가 초래할 수 있는 사건의 영향과 연쇄를 평가한다.

• 각 고장모드가 제품의 정상적 기능이나 성능, 또는 관련된 과정의 신뢰도나 안전성에 미치는 중요성이나 치명도를 결정한다.

• 판명된 고장모드를 검출성 (detectability), 진단성 (diagnosability), 시험성 (testability), 교체성 (replaceability), 보상 및 운용성, 기타 적절한 특성에 따라 분류한다.

• 자료의 가용여부에 따라 중요도와 고장확률을 추정한다.

시기적으로, 이 분석은 제품 구상, 계획, 정의 단계에서 시행될 수는 있으나 제품의 구성과 기능에 관계된 구체적이고도 많은 자료를 필요로 하기 때문에 그 효과가 한정적이어서, 주로 제품 설계 단계와 개발 단계에서 이루어지는 것이 일반적이다.

# BNP&SE 분석 결과

출처

우경일, 우준석, “SW 신뢰성 향상을 위한 국제 규격 현황”, 정보과학회지, 2012.2

JiantaoPan, Software Reliability, CMU, 1999