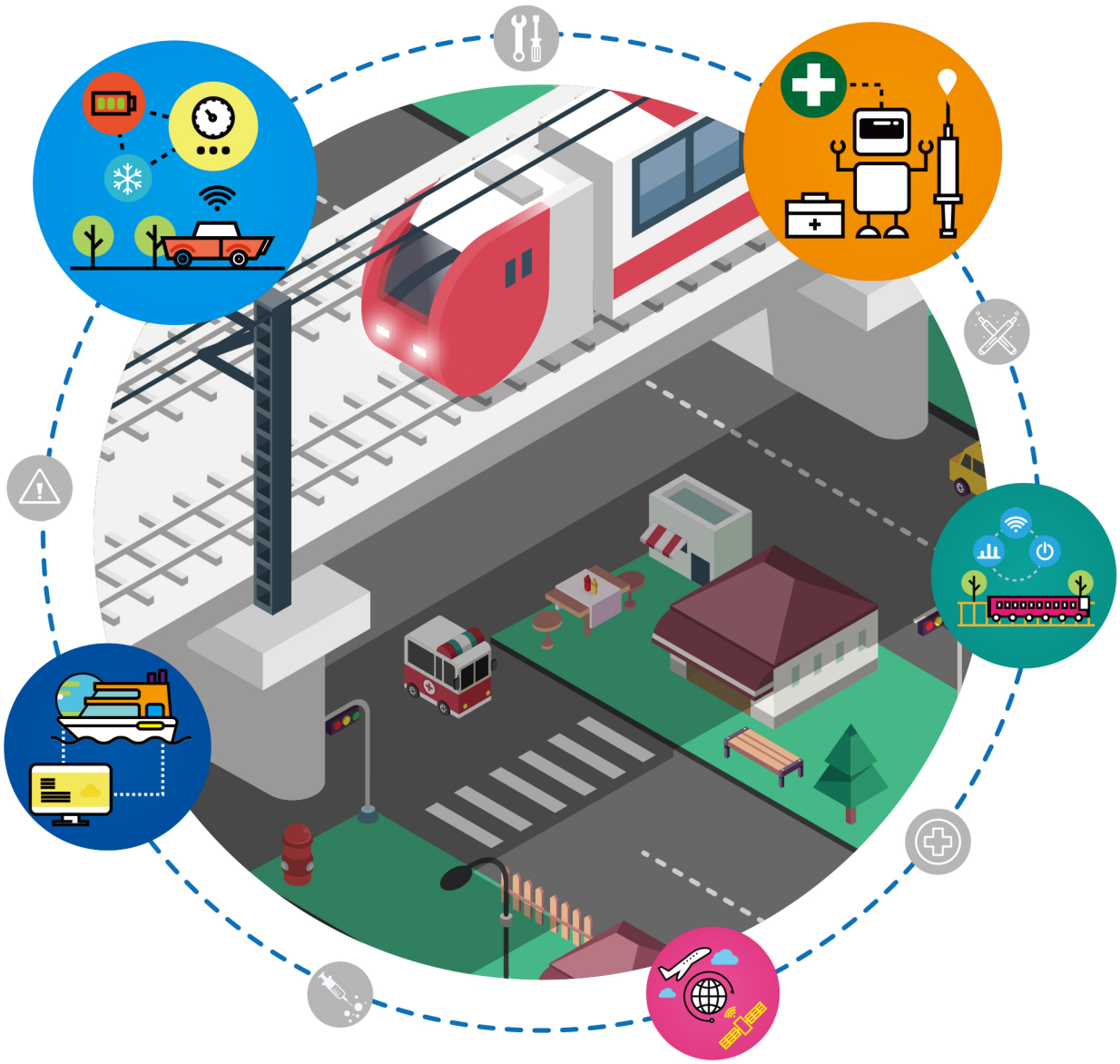
SW안전가이드

공통 분야

SOFTWARE SAFETY



과학기술정보통신부
Ministry of Science and ICT

nipa 정보통신산업진흥원
National IT Industry Promotion Agency

SW안전가이드를 발간하며

소프트웨어가 사회 각 분야를 주도하는 소프트웨어 중심사회가 도래하면서 하드웨어 위주의 시스템 하부 구성품으로만 여겨졌던 소프트웨어가 최근에는 전체 제품·서비스의 가치를 결정하고 있으며, 특히나 국민 안전을 책임지는 핵심요소로 부각되고 있습니다.

특히 철도, 자동차, 항공, 의료 등 국가 기반시설 및 주요 산업분야에서 소프트웨어의 비중이 나날이 증가하고 있는 만큼, 안전과 관련한 소프트웨어의 결함 발생 시 대형 인명사고 및 대규모 경제적.사회적 비용을 유발할 수 있기 때문에 소프트웨어의 안전 확보는 매우 중요하다고 할 수 있습니다. 이와 더불어 선진국이나 개발도상국의 경우 자국의 안전을 위해 주요 산업분야에서 소프트웨어의 안전이 확보된 제품을 사용하도록 의무화하고 있는 추세이며, 이를 위해 별도의 법이나 규정으로 국제안전표준 및 기술기준을 준수하도록 하고 있습니다.

하지만 국내 소프트웨어 기업들이 이러한 국제안전표준을 현장실무에서 쉽게 이해하고 적용하는데 많은 어려움이 있어, 각 산업 군 별 안전 표준에서 요구하는 안전 목표 수준 달성에 대한 구체적인 수행활동, 현장에서의 실무 적용을 위한 풍부한 사례를 포함한 SW안전가이드를 발간하게 되었습니다.

이번에 발간한 SW안전가이드를 활용함으로써 인력이나 기술 수준이 영세한 각 산업 군 별 소프트웨어 개발관련 기업들이 안전성 높은 소프트웨어 시스템을 개발하기 위해서 무엇이 필요하고 어떻게 수행해야 하는지를 이해할 수 있을 것입니다.

‘아는 만큼 보인다.’는 말처럼, 본 가이드를 통해 안전한 소프트웨어 개발을 위한 주요 활동, 절차, 완료 기준 및 산출물 양식 등을 이해하는 것이 해당 기업체의 소프트웨어 경쟁력 강화에 큰 도움이 될 것이라고 확신합니다.

아무쪼록 본 SW안전가이드가 소프트웨어의 안전 확보를 위한 경쟁력 향상과 소프트웨어 기업의 실질적인 안전품질을 높이는 데 도움이 되길 기대하며, 이번 SW안전가이드가 발간이 되기까지 지필과 많은 도움을 주신 학교 및 연구기관, 컨설팅 기업 등 관계자 분들께 다시 한 번 감사드립니다.

정보통신산업진흥원
원 장　윤 종 록

SW안전가이드 발간을 축하하며

인공지능(AI), 사물인터넷(IoT), 빅데이터, 클라우드와 같은 4차 산업혁명 기술들은 개인·기업·정부 전반에 광범위하게 사용되어 인간의 삶의 질을 향상시키고 기업과 정부의 경쟁력을 지속적으로 제고할 수 있다는 기대와 함께 기술에 대한 연구개발과 산업적 활용이 광범위하게 확산되고 있습니다.

앞으로 대부분의 산업이 고도의 정밀도를 요구하는 첨단산업으로 변화하게 될 것이고, 소프트웨어는 4차 산업혁명의 가치를 실현하는 기술적 중심이 될 것입니다. 다시 말하면, 소프트웨어 품질이 최종 제품의 고부가가치 실현을 위한 핵심을 담당하게 될 것입니다.

또한, 자동차, 항공기, 선박 및 의료장비 등 다양한 산업 제품에 복잡한 소프트웨어 활용이 증가하면서 소프트웨어 결함이 발생할 확률이나 사람의 생명과 직결되는 대형 사고의 위험성도 동시에 높아지는 점을 감안하여 제품 개발 시 철저한 대비가 필요합니다.

안전하고 신뢰성이 높은 소프트웨어의 개발은 시대적인 요구가 되었습니다. 이러한 요구는 국내 소프트웨어 기업에 또 다른 형태의 시장진입 장벽으로 작용할 수도 있습니다. 이에 과학기술정보통신부는 소프트웨어의 안전성과 신뢰성을 확보할 수 있는 적절한 방법과 절차를 담은 'SW안전가이드'를 마련하였습니다.

본 가이드는 일차적으로 산업 공통, 철도, 의료, 자동차의 네 부문으로 작성되었으며, 향후 항공, 제조, 로봇 등 전 산업 분야로 확장될 예정입니다.

본 가이드가 소프트웨어 기능 안전성 및 신뢰성 분야 종사자들에게 널리 활용되어 4차 산업혁명 시대에 대비하고 국내 소프트웨어 융합 산업의 역량 제고에 기여할 수 있을 것으로 기대합니다.

SW안전가이드의 발간을 진심으로 축하드립니다.

과학기술정보통신부
소프트웨어정책관 노 경 원

공통 분야 SW안전가이드를 집필하며

기능안전성(Functional Safety)는 시스템과 장비에 요구되는 전체 안전성(Safety) 중에서 전자 장비와 관련 소프트웨어에 대한 안전성을 의미한다. 공공장소, 생산현장, 사무실뿐만 아니라 가정에서 사람의 오작동 혹은 여러 가지 이유로 인해 전자장비의 결함이 발생하여 사용자에게 위험을 초래하고, 결과적으로 인명과 환경을 해치는 크고 작은 사고로 이어지기도 한다.

최근 시스템의 복잡도가 증가하면서 장비의 안전성에 대한 사회적 기대가 더 높아지고 있으며, 특히 소프트웨어의 비중이 높아짐에 따라 소프트웨어 기능안전성에 대한 관심과 요구가 많아지는 추세이다.

안전성 관련 소프트웨어 제품 및 소프트웨어를 포함하는 시스템을 개발하는 기업은 표준에서 제시하는 기준에 부합하는 제품의 기능안전성을 확보하기 위해 표준의 해석하고 적절한 방법을 도입해야 한다. 구체적 방안이 제시되지 않은 상황에서 소프트웨어 기능안전성 확보를 위한 활동을 수행하는 것은 기업에 어려움을 주게 되는 것은 자명하다. 특히, 이러한 상황의 변화와 적응에 미숙한 다수의 국내 소프트웨어 기업에는 또 다른 형태의 시장진입 장벽으로 작용할 수 있다.

본 공통 분야 SW안전가이드는 소프트웨어 기능안전성 확보를 위해 소프트웨어 기능안전성의 목표수준 설정 및 기능안전성 확보를 위한 활동과 현장 적용에 대한 사례를 포함하였고, 소프트웨어 기능안전성을 도입하고자하는 다양한 산업분야의 프로젝트에 공통적으로 활용이 가능하다. 기능안전성 공통 표준인 IEC61508의 Part 1, 3을 기준으로 소프트웨어 분야에 맞게 SIL 2 Recommend 수준에 맞춰 개발되어 실무에 활용할 수 있는 유용한 참조물이 되도록 하였으며, 분야별 상세한 내용은 함께 고민하면서 마련한 자동차 분야, 철도분야, 의료분야 SW안전가이드를 함께 참고하길 바란다.

우리나라 소프트웨어 업계에 종사하는 모든 분들이 소프트웨어 기능안전성의 중요성과 의미에 대해 충분히 이해하고, 소프트웨어 기능안전성 생명주기를 프로젝트에 적절하게 도입하여 소프트웨어 기능안전성이 확보된 제품을 개발할 수 있는 역량을 키워나갈 수 있기를 간절히 희망한다.

한국해양대학교 산학협력단, ㈜나이스컨설팅 컨소시엄
책임자: 이 서 정 교수

목 차

표 목차

그림 목차

Ⅰ. 서론

제 1 장. 배경

현대 사회에서 자동차, 철도, 항공우주, 원자력, 국방 등의 다양한 분야에서 대부분의 장치들이 소프트웨어를 내장하고, 제어용 소프트웨어 시스템이 탑재됨에 따라 소프트웨어의 안전성에 대한 중요도가 높아지고 있다. 소프트웨어 중심 사회가 되어가면서 대부분의 국가기반시설 및 대단위 산업분야에서 소프트웨어를 이용한 제어가 이루어지고, 금융, 자동차, 항공, 전력, 국방, 의료, 교육 등 대부분 분야에서 소프트웨어 의존도가 높아짐에 따라 이의 오류로 인한 사고의 피해 범위와 규모가 확대되고 있다.

다양한 산업 분야에 소프트웨어가 사용되면서 소프트웨어에 의한 사고의 위협도 높아지기 때문에 소프트웨어 오동작에 의한 안전성 위협이 큰 이슈로 떠오르게 되었다. 경미하거나 사소한 소프트웨어 오류가 사람의 생명을 앗아가거나 막대한 경제적 피해를 초래한다. 안전의 관점에서 심각한 경제적 물질적 피해를 초래하거나 인간의 생명에 막대한 위협이 되는 경우에는 보다 거시적이고, 체계적인 대응이 필요하다. 소프트웨어의 안전을 확보하는 일은 소프트웨어에 기능적 오류나 고장을 방지하려는 예방과 오류나 고장 발생을 사전에 감지하여 적절한 대응을 하고, 오류 상황이 발생했을 경우에는 그 피해를 최소화하는 노력이 복합적으로 필요하다. 모든 산업 분야를 막론하고 소프트웨어 기능안전 관련 사고가 발생하지 않도록 개발과정에 안전한 상태를 확보할 수 있는 조치를 취하는 것이 중요하다.

소프트웨어의 사고는 사용자의 오조작에 의해서 발생할 수도 있지만 근본적으로 개발 초기 과정에서의 안전성에 대한 고려와 검증이 제대로 이루어지지 않아 발생하게 된다. 소프트웨어에 의한 사고를 예방하기 위해서는 소프트웨어 안전성에 대한 면밀한 고려가 소프트웨어 개발 단계에서 매우 중요하게 다루어져야한다. 소프트웨어 시스템은 외부의 잘못된 입력으로부터 시스템을 안전하게 지켜주고, 재난 발생을 막기 위한 통제를 수행할 수 있는 가에 대한 고려에서 출발한다. 때문에

소프트웨어 안전성을 만족하기 위해서는 제품개발의 수명주기에 기반한 적절한 안전성 확보 활동의 도입이 필수적이고, 이를 제품개발의 전단계에 적용하는 의지와 노력이 필요하다. 여타의 소프트웨어 품질활동과 마찬가지로 안전성 확보활동의 결과는 산출물 형태의 문서로 기록되어야하고 안전성 확보가 필요하다고 파악된 기능항목은 테스트 단계에서 반드시 검증되어야한다. 기능안전성과 관련된 사고를 줄이기위해 시스템과 장비를 생산하는 단계에서 안전과 관련된 기능에 대한 위험을 평가하고 무결성을 확보하는 방안이 세계적으로 다양한 산업분야에서 진행되어 왔다. 전기, 전자, 또는 프로그램이 가능한 전자 (E/E/PE: Electric, Electronic or Programmable Elecronic) 시스템 및 제품에 대한 포괄적 기능안전성 표준 IEC61508 을 시작으로, 철도, 항공, 에너지, 의료 등 주요 산업분야에서 분야 특성을 고려한 기능안전성 표준을 정의해왔다. 하지만 이러한 기존 표준 및 지침들은 표준문서의 특성상 기능안전성을 확보에 대한 목표와 확인 기준만을 명세하고 있으며, 목표와 기준을 달성하기 위한 도구 등 구체적 방안에 대한 내용은 다루고 있지 않다. 특히 소프트웨어 기능안전성에 대해서는 더욱 부족한 상황이다.

즉, 산업분야를 막론하고 소프트웨어 기능안전성 도입하기 위해서는 기본적으로 안전성에 대한 이해와 표준에서 제시하는 목적을 달성하기 위한 적절한 활동이 수반되어야한다. 본 『SW 안전성 공통 개발 가이드』는 산업분야 전반에 걸쳐 소프트웨어 기능안전성을 확보하기 위해 필요한 기본적인 개념과 SW 안전 수명주기 및 산출물 그리고 작성방법을 상세하게 제공한다.

제 2 장. 가이드 목적 및 적용 범위

1. 목적

본 『SW 안전성 공통 개발 가이드』는 소프트웨어 기능안전성 확보를 위해 소프트웨어 기능안전성의 목표수준 설정 및 기능안전성 확보를 위한 활동과 현장 적용에 대한 사례를 포함한 지침서로 소프트웨어 기능안전성을 도입하고자하는 모든 산업분야 종사자들에게 안전성에 대한 기본적인 이해를 높이고, IEC61508 에서 요구하는 SW 안전성 요건을 확보할 수 있도록 형식을 갖춘 산출물을 포함한 SW 안전 수명주기를 제공하는 것에 그 목적이 있다.

2. 적용 범위

본 『SW 안전성 공통 개발 가이드』는 소프트웨어 기능안전성 확보가 필요한 전 산업분야의 소프트웨어 프로젝트에 적용가능하다.

- 소프트웨어 개발 수명주기를 이미 정의되어 있는 프로젝트의 경우, 본 가이드의 SW 수명주기를 참고하여 기존의 수명주기를 조정하여 수행할 수 있다.
- 소프트웨어 개발 수명주기 단계별 산출물이 이미 확보되어 있는 프로젝트의 경우, 본 가이드의 산출물을 참고하여 기존의 내용을 조정하여 적용할 수 있다.
- 본 가이드는 전 산업분야에 공통적으로 참고할 수 있는 SW 안전성 개발 가이드를 지향하고 있으며, 특정 산업분야에 도입하는 경우, 분야 특성을 고려하여 생명주기 및 산출물을 조정할 수 있다.
- 기능안전성 공통 표준인 IEC61508 의 Part 1, 3 을 기준으로 소프트웨어 분야에 맞게 SIL 2 Recommand 수준에 맞춰 개발되어 실무에 활용할 수

있는 유용한 참조물이 되도록 하였으며, 분야별 개발 가이드인 철도, 자동차, 의료기기 가이드와의 연계하여 활용할 수 있도록 구성하였다.

- 본 가이드에서 제시하는 생명주기와 산출물은 SW 안전성 확보를 위해 기본적으로 필요한 활동과 산출물로써 프로젝트의 규모, 특성 및 운용환경 등 다양한 특성에 따라 적절하게 수정하여 적용할 수 있다.

※ IEC61508 은 최신본은 2010 년 개정되었으나, 국내 표준(KS C IEC61508)은 Part 1 은 2010 년발행 판 기준인데 나머지 Part 2 ~ 7 은 1998 년 기준이므로 참조 불가능하여 국제 규격인 2010 년 IEC61508 영문판을 참조했다.

제 3 장. 문서의 구성

본 문서는 다음과 같이 구성되어 있다.

파트 I 은 서론으로 본『SW 안전성 공통 개발 가이드』의 배경, 가이드 목적 및 적용 범위, 문서의 구성 및 용어 정의를 포함한다.

파트 II 는 SW 안전성 이해를 위해 안전성/위험성의 개념, 안전 관련 국제 표준과 안전 수명주기별 안전성 관리에 대해 자세한 설명이 제공된다.

파트 III 는 위험원 식별 방안 및 분석 절차 그리고 그에 따른 실제 위험분석 사례를 제시하고 실무에서 자주 사용하는 위험분석 기법에 대해 설명한다.

파트 IV 에서는 SW 안전 수명주기에 대한 기본적인 사항을 1 장에서 설명하고, SW 개발 수명주기를 2 장에 설명한다. SW 계획(SP)단계, SW 요구분석(SR)단계, SW 설계(SD)단계, SW 구현(SC)단계, SW 통합(SI)단계 등으로 구성된다. 3 장 안전관련 SW 개발 기법에서는 SW 안전 수명주기 각 단계에 필요한 기법 및 세부 항목에 대해 구체적으로 설명한다.

파트 V 에서는 SW 운영 시 주요 활동과 필요한 산출물 작성법에 대해 설명한다.

파트 VI 부록에서는 SW 안전 수명주기 각 단계에 기본적으로 필요한 산출물의 종류, 양식 및 각 산출물 양식의 작성에 대한 상세한 설명을 제공한다. 실무에서 자주 요구되는 안전 대책 기술서와 본문(파트 IV)에서 나온 IEC61508 Technique/Measure 의 목록을 수록하였다.

제 4 장. 용어 정의

Safety 는 확률적 개념이므로 본 가이드에서는 '안전'이라는 표현보다는 '안전성'으로 정의하여 기술하며, 반대 개념인 Risk 또한 안전성의 반대 개념이므로 '위험'이라는 표현보다는 '위험성'으로 표현하는 것이 적절하다. 다음은 IEC61508 에서 정의하고 있는 기능안전성관련 전문 용어를 설명한다.

1. Harm(피해)
   사람들의 건강에 대한 물리적 부상 또는 환경/자산에 대한 물리적 피해
2. Hazard(위험)
   피해의 잠재적 요인
3. Hazardous situation(위험한 상황)
   하나 이상의 위험에 노출된 사람, 자산, 환경의 상황
4. Hazardous event(위험한 사건)
   피해를 발생시키는 사건
5. Harmful event(유해한 사건)
   피해를 발생시키는 위험한 상황 또는 위험한 사건의 발생
6. Risk(위험성)
   피해발생의 확률과 피해의 심각도의 합
7. Tolerable risk(허용 가능한 위험성)
   사회의 현재 가치에 기반한 주어진 상황을 반영하는 위험
8. Residual risk(잔여 위험성)
   보호 조치를 취하고 남은 위험성
9. EUC risk(EUC 위험성)
   EUC 또는 EUC 제어 시스템의 상호작용으로부터 발생하는 위험성
10. Target risk(목표 위험성)
    전기/전자/프로그램 가능한 전자 안전 관련 시스템과 기타 위험성 감소 조치와 함께 EUC 위험을 고려한 특정 위험에 도달하는 것을 의도하는 위험성
11. Safety(안전)
    수용 불가능한 위험으로부터 벗어난 상태

12. Functional safety(기능 안전)

E/E/PE 안전 관련 시스템과 기타 위험성 감소 조치의 올바른 기능에 의한 EUC 와 EUC 제어 시스템과 관련된 전체 안전의 일부

13. Safe state(안전 상태)

EUC 가 안전에 도달한 상태

14. Reasonably foreseeable misuse(타당한 예비 오용)

제품, 절차 또는 서비스의 사용이 공급자에 의도대로 사용되지 않는 경우, 이는 쉽게 예측 가능한 사람의 행동으로부터 생길 수 있음.

15. Equipment under control, EUC(제어 되는 장비)

생산, 프로세스, 운송, 의료 또는 기타 활동에 사용되는 장비, 기계, 기관, 플랜트를 말함 EUC 제어시스템과 EUC 는 별개임.

16. Environment(환경)

특정 조건하에 있는 응용이나 모든 안전성 생명주기 단계에서 기능 안전에 도달할 수 있도록 하는 모든 관련 요소 예를 들어, 물리적 환경, 사용 환경, 법적 환경, 유지 보수 환경 포함가능

17. Functional unit(기능 유닛)

특정 목적을 이룰 수 있게 하는 하드웨어 또는 소프트웨어(혹은 둘 모두를 포함) IEV 191-01-01 에서는 기능 유닛 대신 item(아이템)을 사용, item 에는 사람이 포함될 수 있음

18. Application(응용)

E/E/PE 시스템보다 EUC 와 관련된 작업

19. Software(소프트웨어)

데이터 처리 시스템 운용과 관련된 프로그램, 프로시저, 데이터, 규칙, 그리고 관련 문서로 구성되는 지적 생산물

20. System software(시스템 소프트웨어)

PE 시스템의 소프트웨어, 이것은 프로그램 가능한 장치 스스로 서비스를 생산하고 기능함, 응용 소프트웨어와 대비되는 개념

21. Configuration data(응용소프트웨어, 응용데이터, 설정데이터)

기능을 구체화 하는 PE 소프트웨어, 프로그램 가능한 장치 스스로 제공하는 서비스나 장치 스스로의 기능보다는 EUC 와 관련된 작업을 수행함

22. Pre-existing software(이미 존재하는 소프트웨어)

이미 존재하고 있으며, 현재 프로젝트나 안전 관련 시스템과 특별히 관계가 있지 않은 소프트웨어 요소

23. Data(데이터)

컴퓨터의 통신, 해석, 프로세싱을 위한 의미로 표현된 정보

24. Software on-line support tool(소프트웨어 온라인 지원 도구)

소프트에어 실행 시간 동안 안전 관련 시스템에 직접 영향을 줄 수 있는 소프트웨어 도구

25. Software off-line support tool(소프트웨어 오프라인 지원 도구)

소프트웨어 개발 생명주기의 한 단계를 지원하는 소프트웨어 도구, 실행시간에 직접 영향을 주지 못함. 소프트웨어 오프라인 지원도구는 T1, T2, T3 로 구분

-T1:(데이터를 포함한)실행 코드에 직/간접적으로 기여하는 산출물을 생산하지 않음

-T2:설계 또는 실행 코드에 대해 테스트 및 검증을 지원

-T3:(데이터를 포함한)실행 코드에 직/간접적으로 기여하는 산출물을 생산함

26. Programmable electronic, PE(프로그램 가능한 전자의 (어떤 것))

하드웨어, 소프트웨어, 입력 또는 출력 유닛으로 구성될 수 있는 컴퓨터 기술에 기반한 (어떤 것).

27. Electrical/electronical/programmable electronic, E/E/PE(전기/전자/프로그램 가능한 전자의 (어떤 것))

전기, 전자 또는 프로그램 가능한 전자 기술에 기반한 (어떤 것)

28. Limited variability language(제한된 다양성 언어)

상업적 및 산업적으로 프로그래밍 가능한 전자 제어장비를 위한 소프트웨어 프로그래밍 언어, 응용에 제한적임

29. Application specific integrated circuit, ASIC(주문형 반도체)

특정 기능을 목적으로 생산되고 고안된 집적 회로, 제품 생산자에 의해 기능이 정의됨(wikipedia)

30. PE system(PE 시스템)

하나 이상의 PE 장치로 이루어진 제어, 보호, 모니터링을 위한 시스템, 여기서 PE 장치에는 전원 장치, 센서, 입력 장치, 데이터 고속도로, 통신 패스, 액츄에이터, 기타 출력 장치 등을 포함함

31. E/E/PE system(E/E/PE 시스템)

하나 이상의 E/E/PE 장치로 이루어진 제어, 보호, 모니터링을 위한 시스템. 여기서 PE 장치에는 전원 장치, 센서, 입력 장치, 데이터 고속도로, 통신 패스, 액츄에이터, 기타 출력 장치 등을 포함함

32. EUC control system(EUC 제어 시스템)

프로세스 또는 오퍼레이터로부터 입력 받은 신호에 대한 응답을 하는 시스템. 또한, 원하는 방식으로 EUC 가 출력신호를 생성하는 시스템

33. Architecture(구조)

시스템에서 하드웨어 및 소프트웨어의 특정한 설정

34. Software module(소프트웨어 모듈)

프로시저 및 자료선언의 구성을 구조화 함. 또한, 다른 구성과 상호작용 할 수 있도록 구조화 함

35. Channel(채널)

기초 안전 기능을 독립적으로 구현하는 요소 혹은 요소들의 그룹

36. Diversity(다양성)

요구되는 기능을 수행하는 다른 방법 다양성은 물리적 방법 또는 디자인 접근방법을 다르게 함으로써 달성될 수 있음

37. Safety-related system(안전 관련 시스템)

아래의 특성을 갖는 시스템

- EUC 의 안전한 상태를 달성 또는 유지하는데 필요한 안전 기능을 구현함
- 시스템 자체 또는 E/E/PE 안전 관련 시스템이나 다른 위험성 감소 조치를 포함하여 요구되는 안전 기능에 대해 필요한 안전 무결성을 달성하려는 것

38. Other risk reduction measure(기타 위험성 감소 조치)

위험성을 감소시키거나 완화시키기 위한 조치. 이것은 E/E/PE 안전 관련 시스템과 분리되고 구분됨

39. Low complexity E/E/PE safety-related system(낮은 복잡도의 E/E/PE 안전 관련 시스템)

아래와 같은 특성을 갖는 E/E/PE 시스템

- 각각 개별적인 컴포넌트의 실패 모드가 잘 정의됨
- 결함 조건에 맞는 시스템의 응답행동이 완전히 결정될 수 있음

40. Subsystem(서브시스템)

안전 관련 시스템의 최상위 구조 설계 결과물

41. Element(요소)

하나 이상의 기초 안전 기능을 수행하는 단일 컴포넌트 또는 컴포넌트의 그룹으로 구성된 서브시스템

42. Redundancy(여분)

요구되는 기능을 수행하기 위해 또는 정보를 표현하기 위해 하나 이상의 방법을 갖는 존재

43. Safety function(안전 기능)

특정 위험한 사건에 대하여 E/E/PE 시스템 또는 기타 위험성 감소 조치에 의해 구현된 기능

44. Overall safety function(전체 안전 기능)

특정 위험한 사건에 대하여 EUC 를 위한 안전한 상태를 달성 또는 유지하기 위한 방법

45. Element safety function(기초 안전 기능)

요소에 의해 구현되는 안전 기능의 일부

46. Safety integrity(안전 무결성)

E/E/PE 안전 관련 시스템이 정해진 시간 안에 정해진 모든 조건에 맞는 특정 안전 기능을 만족스럽게 수행할 확률

47. Software safety integrity(소프트웨어 안전 무결성)

(소프트웨어 속성에 맞는)장애의 위험한 모드에서 시스템적 장애와 관련한 안전 관련 시스템의 안전 무결성

48. Systematic safety integrity(시스템적 안전 무결성)

장애의 위험한 모드에서 시스템적 장애와 관련한 안전 관련 시스템의 안전 무결성

49. Hardware safety integrity(하드웨어 안전 무결성)

장애의 위험한 모드에서 임의의 하드웨어 장애와 관련한 안전 관련 시스템의 안전 무결성

50. Safety integrity level, SIL(안전 무결성 등급)

안전 무결성 값의 범위에 따라 구분된 등급. 예를 들어, 4 단계로 나누었을 때 4 단계가 가장 높고 1 단계가 가장 낮음

51. Systematic capability(시스템적 능력)

요소의 시스템적 안전 무결성이 특정 SIL 의 요구사항을 만족함을 보여줄 수 있는 기준

52. Software safety integrity level(소프트웨어 안전 무결성 등급)

안전 관련 시스템의 서브시스템으로 구성된 소프트웨어 요소의 시스템적 능력

53. E/E/PE system safety requirements specification(E/E/PE 시스템 안전 요구사항 명세)

안전 기능 및 그와 관련된 안전 무결성 등급을 위한 요구사항을 담고 있는 명세

54. E/E/PE system safety functions specification(E/E/PE 시스템 안전 기능 명세)

안전 관련 시스템에 의해 수행되어야 하는 안전 기능에 대한 요구사항을 담고 있는 명세

55. E/E/PE system safety integrity requirements specification(E/E/PE 시스템 안전 무결성 요구사항 명세)

안전 관련 시스템에 의해 수행되어야 하는 안전 기능의 안전 무결성 요구사항을 담고 있는 명세

56. E/E/PE system design requirements specification(E/E/PE 시스템 설계 요구사항 명세)

서브시스템과 요소의 관점에서 E/E/PE 안전 관련 시스템을 위한 설계 요구사항을 담고 있는 명세

57. Safety-related software(안전 관련 소프트웨어)

안전 관련 시스템에서 안전 기능을 구현하는데 사용되는 소프트웨어

58. Mode of operation(동작 모드)

안전 기능을 동작하는 방법

- 낮은 요청 모드: EUC 가 특정 안전 상태에 들어가기 위해서 안전 기능이 즉시 수행되어야만 하는 경우, 연간 요청 주기가 1 개 미만인 경우
- 높은 요청 모드: EUC 가 특정 안전 상태에 들어가기 위해서 안전 기능이 즉시 수행되어야만 하는 경우, 연간 요청 주기가 1 개 이상인 경우
- 계속적 모드: 보통의 동작에서 안전 기능이 EUC 가 안전한 상태를 유지할 수 있게 하는 경우

59. Target failure measure(목표 장애 기준)

안전 무결성 요구사항에 대해 위험한 모드의 장애의 목표 확률

- 요청이 왔을 때 안전 기능에 대한 위험 장애의 평균 비율(낮은 요청 모드)
- 위험 장애 [h-1]의 평균 주기(높은 요청 모드 또는 계속적 모드)

60. Necessary risk reduction(필요 위험성 감소)

위험성의 허용 가능한 범위를 초과하지 않도록 하기 위하여 E/E/PE 안전 관련 시스템 및 기타 위험성 감소 조치에 의해 달성되는 위험성 감소

61. Fault(결함)

요청된 기능을 수행할 때 기능 유닛의 능력이 감소하거나, 손실이 발생하는 비정상적 조건

62. Fault avoidance(결함 회피)

안전성 생명주기의 어떤 단계 중에도 결함을 피할 수 있도록 하는 기술 및 프로시저의 사용

63. Fault tolerance(결함 허용)

결함 또는 오류가 존재할 때 기능 유닛이 요청된 기능을 계속 수행할 수 있는 능력

64. Failure(장애)

요청된 방법 이외의 방법으로 기능 유닛 기능 또는 동작을 정상적인 제공이 멈추는 경우

65. Random hardware failure(임의의 하드웨어 장애)

임의의 시간에 발생하는 하나 또는 그 이상의 하드웨어 오염 메커니즘을 발생시키는 장애

66. Systematic failure(시스템적 장애)

설계 또는 생산 프로세스의 수정이나 동작 프로시저, 문서 또는 기타 관련 요인에 의해 제거될 수 있는 장애

67. Dangerous failure(위험한 장애)

A) EUC 가 위험한 상태 또는 잠재적으로 위험한 상태로 들어가도록 요청 모드일 때 안전 기능이 실행되는 것을 막거나 계속적 모드에서 안전 기능이 실패하게 함

B) 필요 시 안전 기능이 제대로 동작하는 확률을 감소시킴

68. Safe failure(안전한 장애)

A) EUC 가 안전한 상태에 들어가거나 안전한 상태를 유지하기 위해 안전 기능의 거짓 동작이 발생함

B)거짓 동작의 발생 확률이 증가함

69. Dependent failure(의존적 장애)

의존적 장애의 확률은 개별 사건에 대한 확률의 곱으로 나타낼 수 없음

※두 사건 A 와 B 가 의존적일 때,$P(A \text{ and } B) > P(A) \times P(B)$.

70. Common cause failure(일반적으로 발생하는 장애)

멀티 채널 시스템에서 둘 또는 그 이상의 분리된 채널의 장애가 동시발생 하는 경우, 시스템 장애 발생

71. Error(오류)

이론적으로 올바른 값이나 조건과 계산, 관찰 또는 측정된 값이나 조건과의 차이. 이론 값과 실측 값의 차이

72. Soft-error(소프트 오류)

(물리 회로 자체에는 영향을 주지 않는) 데이터 컨텐츠의 잘못된 변화

73. No part failure(무 역할 장애)

안전 기능을 구현하는데 아무런 역할을 하지 않는 컴포넌트의 장애

※안전 장애 비율 산정방식을 사용하지 않음

74. No effect failure(무 영향 장애)

안전 기능을 구현하는데 역할을 하지만 안전 기능에 직접적으로 영향을 미치지 않는 요소의 장애

※ 안전 장애 비율 산정방식을 사용하지 않음

75. Safe failure fraction(안전 장애 비율)

(Σλ_(Savg)+Σλ_(Ddavg))/(Σλ_(Savg)+Σλ_(Davg))

Σλ_(Savg):average of the rate of safe failure

Σλ_(Dd avg): average of the rate of detected dangerous failure

Σλ_(D avg): average of the rate of dangerous failure

76. Probability of dangerous failure on demand, PFD(즉발적 위험한 장애에 대한 확률)

EUC 또는 EUC 제어 시스템에서 요청 발생 시 E/E/PE 안전 관련 시스템이 특정 안전 기능을 실행할 때의 안전 비가용성

77. Average probability of dangerous failure on demand, PFDavg(즉발적 위험한 장애에 대한 확률의 평균)

EUC 또는 EUC 제어 시스템에서 요청 발생 시 E/E/PE 안전 관련 시스템이 특정 안전 기능을 실행할 때의 평균 비가용성

78. Average frequency of a dangerous failure per hour, PFH(시간당 위험한 장애의 평균 발생 주기)

주어진 시간 동안 E/E/PE 안전 관련 시스템이 특정 안전 기능을 수행할 때 발생하는 위험한 장애의 평균 발생 주기

79. Process safety time(안전 처리 시간)

EUC 또는 EUC 제어 시스템에서 장애가 발생한 시점부터 발생한 장애가 완전히 해결된 시점 사이의 시간

여기서 장애는 위험한 사건을 일으킬 수 있는 잠재적 장애이며, 장애를 해결하는 것은 위험한 사건이 발생하지 않도록 방지하는 것

80. Mean time to restoration, MTTR(복원 평균 시간)
복원 상태에 도달하기까지 예상 시간

81. Mean repair time, MRT(평균 복구 시간)
전체 복구 예상 시간

82. Safety lifecycle(안전 생명 주기)
안전 관련 시스템을 구현하기 위하여 프로젝트의 컨셉을 잡는 것부터 시스템이 더 이상 사용되지 않을 때 까지 포함되는 모든 필요한 활동들을 말함

83. Software lifecycle(소프트웨어 생명주기)
소프트웨어의 개발이 시작될 때부터 영구적으로 폐기될 때까지 발생하는 모든 활동

84. Configuration management(형상 관리)
포넌트에 대한 변화를 제어하고 지속성 및 추적성을 유지하는 것에 대한 규칙

제 5 장. 참고 문헌

본 가이드는 IEC61508 기준으로 작성되었지만, 표준만으로는 안전성에 대한 개념이해 및 실무 적용에 한계가 있어, 연관된 국제 표준 및 규격을 추가하고 국내외 국가, 연구기관, 인증 기관, 컨설팅사 등의 공표된 문헌을 참조하였다.

1. 표준 및 규격

- IEC 61508-1:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements
- IEC 61508-2:2010, Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems

- IEC 61508-3:2010, Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 3: Software requirements
- IEC 61508-4:2010 Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 4: Definitions and abbreviations
- IEC 61508-5:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity levels
- IEC 61508-6:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
- IEC 61508-7:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures
- KS C IEC61508 - 1:2010 전기/전자/프로그램 가능 전자식 안전관련 시스템의 기능 안전성 - 제 1 부: 일반 요구사항
- 철도 소프트웨어 안전 기준 및 체계 구축 연구보고서, 2008
- DOD Joint-SW-Systems-Safety-Engineering-Handbook
- IEEE730 A guide to writing successful SQA plans
- IEEE829:2008 SW test documentation
- IEEE1016:2009 Software design description
- IEEE1012:2004 Standard for Software Verification and Validation
- IEEE1228:1994 IEEE Standard for Software Safety Plans
- IEEE1540:2001 Software Engineering Risk Management: Measurement-Based Life Cycle Risk Management
- ISO12207:2004 Systems and software engineering - Software life cycle processes

2. 해외 문헌

- JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING HANDBOOK, DOD, USA
- NUREG CR-6430 UCRL-ID-122514 SW Safety Hazard Analysis, NRC, USA
- NUREG IA-0145 RELAP5 Assessment Against - Revision 1, NRC, USA
- Example risk assessment for a motor vehicle mechanical repair workshop, HSE, UK
- Marine risk assessment, HSE, UK
- Risk assessment programme Overview, NHS, UK
- Healthcare risk assessment made easy, NHS, UK
- Health & Safety Risk Assessment Template, ETE, UK
- OCCUPATIONAL HEALTH & SAFETY TEMPLATE, Ohs, UK
- CASS TOES FOR THE OVERALL SAFETY LIFECYCLE ASSESSMENT (IEC 61508-1: 2010)
- CASS TOES FOR FUNCTIONAL SAFETY MANAGEMENT ASSESSMENT (IEC 61508-1: 2010)
- CASS TOES FOR THE E/E/PE SYSTEM SAFETY LIFECYCLE ASSESSMENT (IEC 61508-2: 2010)
- CASS TOES FOR ELEMENT AND SUBSYSTEM SIL-CAPABILITY ASSESSMENT (IEC 61508-2: 2010)
- CASS TOES FOR SOFTWARE ASSESSMENT (IEC 61508-3: 2010) – ABRIDGED
- IEC 61508 Functional Safety Assessment - One Series Safety Transmitter, EXIDA
- Safety Risk HAZID Workshop, DNV
- RISK ASSESSMENT APPLICATIONS FOR THE MARINE AND OFFSHORE OIL AND GAS INDUSTRIES, ABS
- Specification of Safety Requirements (general section), SIEMENS
- HAZID for RoPax, SAFEDOR

- An overview of functional safety standards and easing certification, Exida / Texas Instruments

3. 국내 문헌

- 이중연료엔진의 연료가스공급시스템에 대한 안전무결도 기반 안전계장시스템 설계, 대한조선학회
- 열차제어시스템 위험원 분석을 위한 HAZOP-KR 에 대한 연구, 한국철도학회
- SIL4 인증문서 한글 표준양식(템플릿) 적용사례 연구, 한국철도학회
- 소프트웨어 안전성 평가를 위한 소프트웨어 고장 유형과 영향 분석에 관한 연구, 멀티미디어학회
- 소프트웨어 개발 프로세스에서의 안전성 분석 및 관리 활동의 적용방안, 중소기업융합학회
- 안전 필수 시스템을 위한 안전성 분석기법, 중소기업융합학회
- 특집 ISO 26262 에서 요구하는 안전 활동 관리 (Safety Activity Management) 방안 연구, 전자공학회지
- 리스크의 개념에 대한 고찰, 한국안전학회
- 공정안전향상을 위한 Safety Integrity Level 의 적용 방향, 한국안전학회
- FMEA 를 활용한 플랜트공사 위험성평가 방안, 한국안전학회
- 항공기 인증을 위한 시스템 안전성 평가, 한국항공진흥협회
- FMEA 를 활용한 중점안전관리 항목 도출방안, 한국건설관리학회
- 위험성 평가 - 일반 가이드, 안전보건공단
- 사업장 위험성평가(Risk assessment) 매뉴얼, 안전보건공단
- 『위험성평가』실시규정(절차서), 안전보건공단
- 방호계층분석(LOPA)기법에 관한 기술지침, 안전보건공단
- 위험관리 계획서 및 보고서 작성 가이드, 안전보건공단
- 2016 위험성평가 – 평가담당자 교육, 안전보건공단

- 의료기기 소프트웨어 허가·심사 가이드라인, 식품의약품안전처
- 의료기기 위험관리 가이드라인(Guidelines for Risk Management of Medical devices), 식품의약품안전청
- 임상시험 전자 자료 처리 및 관리를 위한 가이드라인, 식품의약품안전청
- 임상시험 전자 자료 처리 및 관리를 위한 가이드라인 해설서, 식품의약품안전청
- 2016 철도안전전문가 기본과정, 교통안전공단
- 2016 철도안전전문가 전문과정, 교통안전공단
- 2007 위험도 분석 및 안전성 평가 (II. 분석), 교통안전공단
- 위험성 결정(Risk Evaluation), 한국산업보건협회
- 차량전장용 운영체제 검증 사례를 통한 소프트웨어 안전성 검증 기법 소개, 정보통신산업진흥원
- 소프트웨어 안전성 확보 체계에 관한 연구- 시험, 평가, 인증을 중심으로, SPRI
- IT 융합 산업의 H/W 및 S/W 의 안전표준화 기술 동향, 한국방송통신전파진흥원
- 원자력발전소 I&C 시스템을 위한 안전성분석 기법, 건국대학교
- 위험사회에서 국민의 안전보호의무를 지는 보장국가의 역할- 현행 안전법제에 관한 고찰을 겸하며 -, 서울대학교
- 가동중인 석유화학공장에서의 LOPA 를 통한 위험성평가 사례, SK 이노베이션
- 기능안전 적용을 위한 소프트웨어 개발 가이드라인, KTL
- 자동차 전기/전자 개발에 안전 엔지니어링의 도입: 문제점과 해결책, VECTOR

II. SW 안전성 개요

제 1 장. 안전성 개념

1. 안전성(Safety)의 이해

안전성의 정의는 국제안전규격을 위한 가이드인 ISO/IEC GUIDE51 에 기술되어 있다. 이것에 의하면, 안전성은 '수용할 수 없는 위험성이 없는 것(freedom from unacceptable risk)'이라고 표현되어 있다. 직역하면, '수용할 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 1 차적 안전(Primary safety), 기능 안전성(Functional safety), 간접 안전성(Indirect safety)으로 분리할 수 있다. 1 차적 안전성은 화재, 감전과 같은 하드웨어에 의한 직접적 사고로부터의 안전성으로 정의한 것이고, 기능 안전성은 리스크 평가 측정결과에 따라서 설계과정을 통해 위험이 제거되는 장비의 안전성을 말하며, 간접 안전성은 데이터베이스 정보 에러와 같이 잘못된 정보 제공으로 일어날 수 있는 위험원으로부터의 안전성을 정의한다.

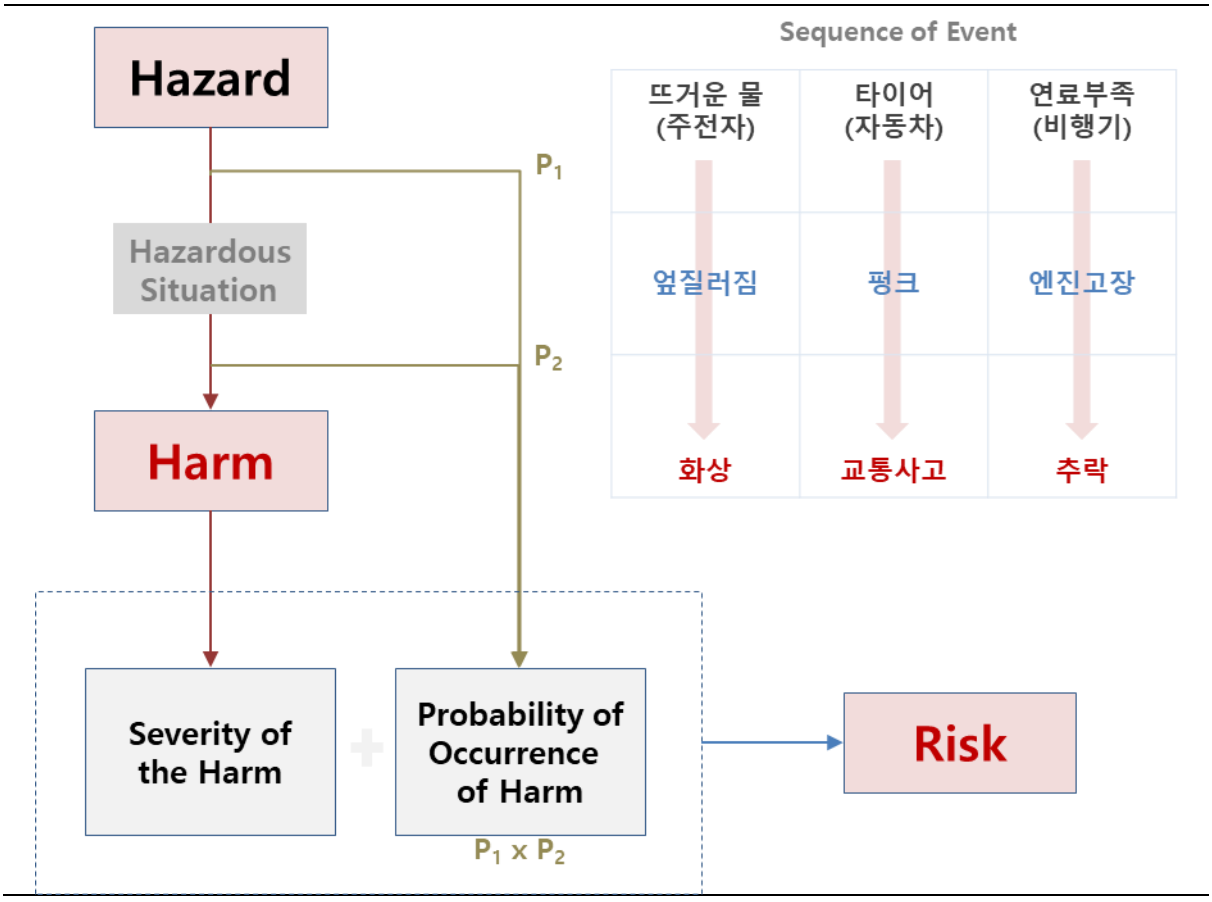
2. 위험성(Risk)에 대한 이해

일상에서 "Risk 가 있다"라고 할 때, Risk 는 '위험성, 일이 잘 안 풀릴 우려, 손실의 가능성' 등을 의미할 수 있다. 산업분야에서도 적용 형태에 따라 다양하게 쓰이는데 공학, 교육, 식품(농업/어업) 관련 분야에서 Risk 는 위험, 사망, 부상, 질병 등의 의미로 사용되고 금융에서는 비용손실, 신뢰 상실, 모험, 도박 등을 의미하며, 정보통신기술에서는 정보유출, 가동중단 등의 의미를 나타내는 등 다양하게 쓰이고 있다. 산업안전보건 또는 환경분야의 경우 개념을 별도로 정의하여 'Risk'를 '위험성', '위해성'으로 사용하고, 위험성은 'Risk', 유해(Harm, Accident)를 일으키는 위험원은 'Hazard'로 구분하여 사용한다. Hazard 는 확률 개념을 포함하지 않고, 확률 개념인

Risk 의 근본 원인을 Hazard 로 표현한다. 위험을 위험성 평가는 'Risk Assessment'를 각각 의미한다.

그림 II.1.1 은 유해(Harm), 위험원(Hazard), 위험성(Risk)의 관계를 표현한 것으로 예를 들어, 뜨거운 물이 엎질러져 화상을 입는 경우에 뜨거운 물은 위험원이 되고, 물이 엎질러지는 상황은 위험한 상황, 그로 인해 화상을 입는 것을 유해로 예시하면 이러한 일련의 과정이 일어날 수 있는 가능성 및 결과의 심각성을 위험이라 표현하였다. 뜨거운 물이 엎질러져 화상을 입거나, 타이어 펑크로 인해 교통사고가 발생하거나, 연료부족으로 인한 엔진 고장으로 비행기가 추락하는 경우를 생각해 보면 개념에 대한 이해를 좀더 명확히 할 수 있다.

그림 II.1.1 Hazard, Harm, Risk 이해도



정리하면 위험성(Risk)은 '유해 위험원(Hazard)이 유해(Harm)인 상해 또는 사고로 이어질 수 있는 가능성(빈도, Probablity)과 심각성(강도, Severity)을 조합한 것'으로 정의할 수 있다. 위험성 평가는 '위험원(Hazard)를 찾아내어 사고 발생 확률과 사고 크기를 분석하여 그때 발생하는 영향을 정량화하여 대책을 세우는 과정' 이라 할 수 있다. 이때 위험원(Hazard)은 정성적 기법으로 찾아내고, 위험성(Risk)의 대상인 사고발생확률과 사고 크기는 정량적 기법으로 찾아낸다.

안전성(Safety)은 확률적인 개념으로 절대적 안전이란 존재하지 않고 상대적인 안전 정도로 판단한다. 위험성(Risk)의 반대 개념으로 생각하여 보면 이해가 쉬운데, 매우 안전하다는 것은 위험하지 않다는 것이다. 안전성 평가는 확률 또는 정도의 개념으로 위험성에 대한 정량화하여 위험성을 제거 또는 감소하기 위한 기능이나 대책을 마련하는 위험성 평가 과정을 포함하며 여기서 나온 기능 및 대책이 안전 기능 및 안전 대책이 된다. 그림 Ⅱ.1.2 는 두 개념의 관계를 보여준다.

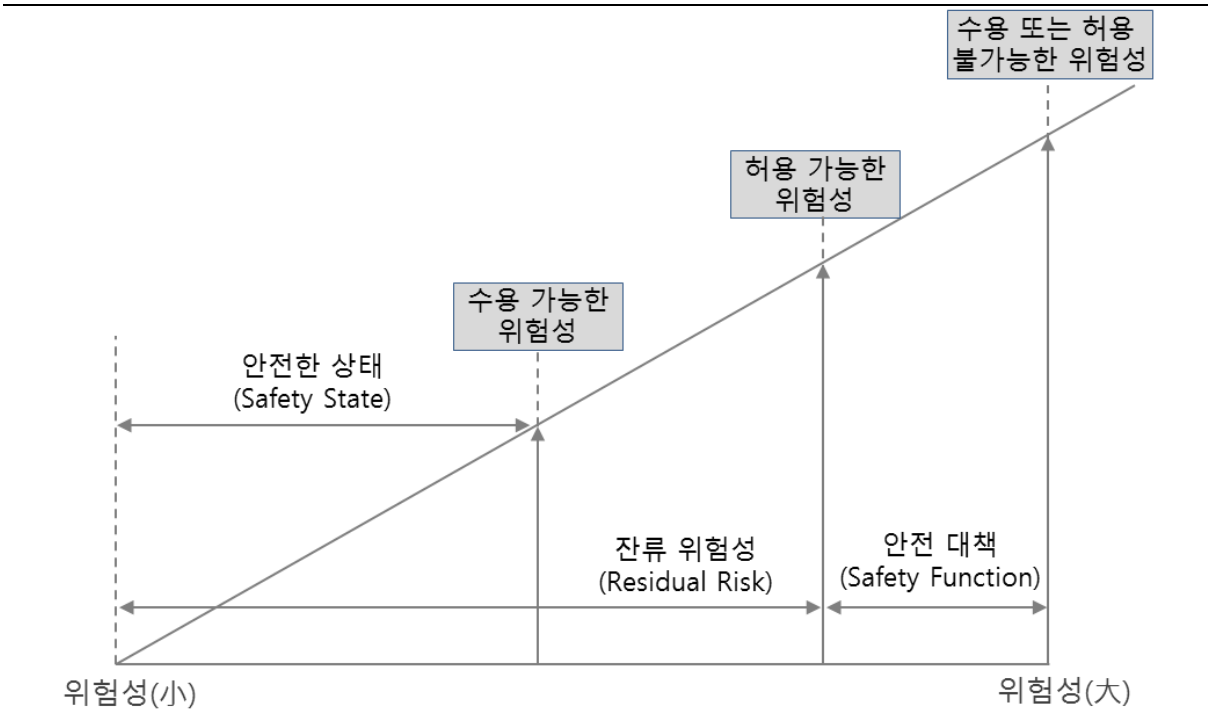
그림 Ⅱ.1.2 위험성(Risk) 과 안전성(Safety)의 관계

이와 같이 안전성은 위험성을 이용하여 정의하고 정량화하여 평가할 수 있고, 안전성이란 위험성의 크기가 모두 수용 가능하고 허용 가능한 것만으로 되어 있는 상태(Safty State)라고 할 수 있다. 확률적 개념인 안전성의 수준을 안전무결성(Safety Integrity level)로 표현할 수 있으며 자세한 설명은 이후 본문에 상세한 설명을 참조한다.

3. 수용 또는 허용 가능한 위험(Risk)

안전하다고 해도 반드시 약간의 위험성이 남아 있기 마련이기 때문에 절대적으로 안전하다고 하는 절대 안전을 주장하고 있는 것은 아니므로 반드시 어떤 크기의 위험성이 남아 있고 항상 사고는 일어날 수 있다는 것을 표현하고 있는 것이다. 남아 있는 위험성, 즉 수용할 수 있거나 허용할 수 있는 위험성은 일반적으로 잔류 위험성(residual risk)이라고 일컬어진다. 여기에서 무엇을 가지고 안전하다고 할 것인가 할 때에 두 개의 다른 단어가 나온다. 하나는 「수용 가능한 위험성」(Acceptable Risk)이고, 다른 하나는 「허용 가능한 위험성」(Tolerable Risk)이다.

그림 II.1.3 수용 또는 허용 가능한 위험성과 안전

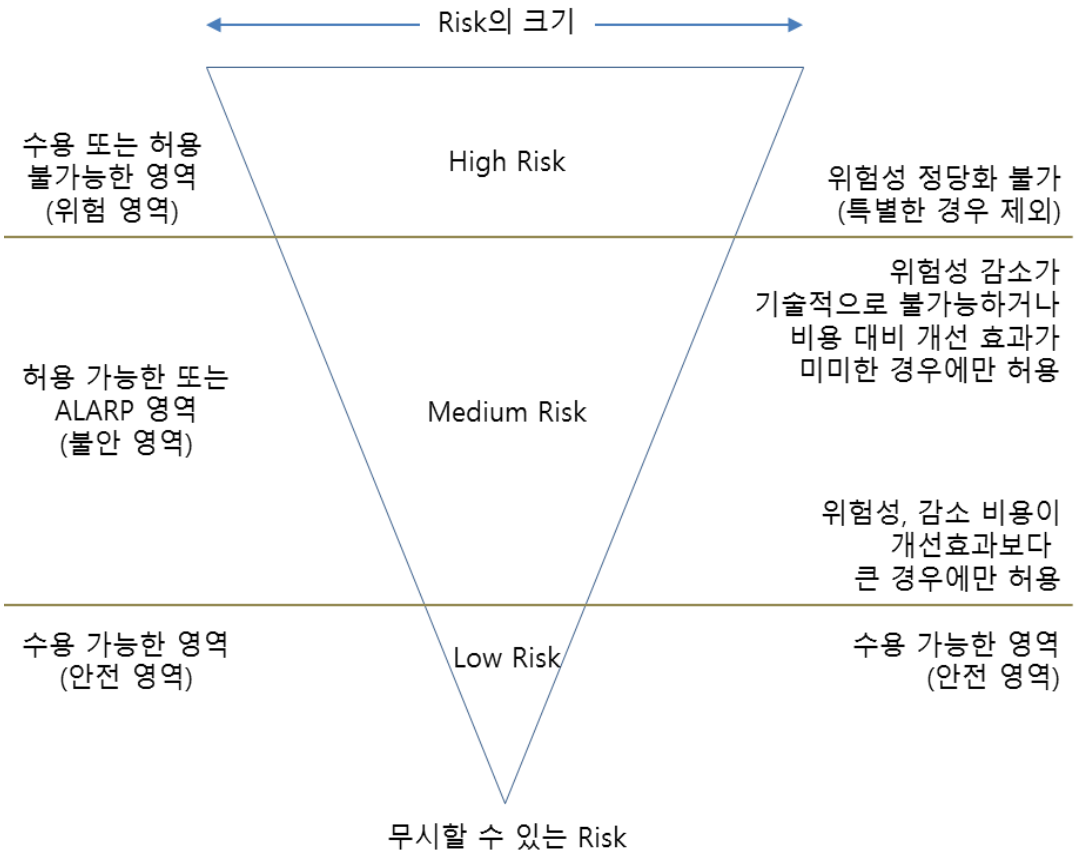


(참조 : 위험성 결정(Risk Evaluation), 한국산업보건협회)

제품 또는 시스템을 설계하는 초기 단계에서는 여러 가지 큰 위험성(risk)이 존재하고 안전상 불안한 상황이다 다. 그래서 각각의 위험성에 대하여 설계, 제작 및 운영 단계에서 각종 안전 기능 또는 안전 대책을 마련하여 위험성의 크기를 줄여야 한다. 누가 생각하더라도 이 정도 크기의 위험성만 존재한다면 문제가 되지 않는 상태로 수용 가능한 위험성이라고 할 수 있다. 한 마디로 위험성이 매우 적거나 적게 되었기 때문에 문제가 되지 않은 위험성 영역이 바로 진정으로 안전한 상태(Safety State)라고 말할 수 있다. 그림 Ⅱ.1.3 과 그림 Ⅱ.1.4 를 참조한다.

예를 들면, 운석이 지구 위에 떨어져 이것에 부딪혀 사망하거나 부상을 입는다고 하는 위험성은 명백히 존재한다. 그러나 이와 같은 것이 발생할 가능성은 매우 낮으므로 적은 위험성이다. 그러나 이와 같은 위험성은 매우 심각하지만 빈도가 낮기 때문에 누구도 방어를 생각하거나 지하로 숨어 나오지 않거나 하지는 않는다.

그림 Ⅱ.1.4 위험성의 크기별 영역과 수용 가능성



(참조 : 위험성 결정(Risk Evaluation), 한국산업보건협회)

위험성은 누구라도 수용하는 위험성, 즉 수용 가능한 위험성이고, 운석에 부딪혀 부상을 입거나 사망을 한다고 하는 그런 위험성에 대해서는 안전하다는(수용 가능하다는) 생각을 한다. 공장의 기계 장치나 시스템의 경우에도 큰 사고가 일어나지 않도록 하는 대책이 확실하게 수립되어 있어, 만약 사고가 발생하더라도 일시 정지 등 경미한 사고만 일어난다면 낮은 위험성으로 되어 있으면, 수용 가능한 위험성 즉 안전하다고 말할 수 있다.

그러나 현실적으로는 안전대책을 마련하는데는 제약사항이 많다. 비용이 너무 많이 들어 비현실적인 경우가 많고 이와 같은 경우, 그 기계 장치 또는 시스템에서 받는 편익 등을 고려하여 어쩔 수 없다고 보아 수용하게 되는 수준의 위험성이 허용 가능한 위험성이다. 바꾸어 말하면 위험성이 합리적으로 실행 가능한 수준까지 감소되어 있는(ALARP, As Low As Reasonably Practicable) 위험성 영역이라고 할 수 있다.

예를 들어 자동차를 생각해 보자. 우리나라에서도 매년 5,000 명 이상의 사람들이 교통사고로 사망하고 있으니 이 정도로 위험한 기계는 없을 것이다. 그럼에도 불구하고 자동차는 지구상에서 널리 이용되고 있다. 모두가 받아들이고 있는 것이다. 이것은 누구라도 수용하는 적은 위험성 이라고는 도저히 생각되지 않는다. 그러나 많은 사람들은 자동차의 편리성 때문에 정비를 하거나 안전운전에 유의하는 방법을 통하여 안전성을 확보한 후에 자동차의 위험을 수용하면서 각오하고 운전을 하고 있는 것이다. 이 점에서는 자동차의 위험성은 허용 가능한 위험성이라고 인식되고 있다. 물론 위험성의 크기를 생각하여 자동차의 운전을 하지 않거나 타지 않는 사람이 있는 것도 사실이다. 그러한 사람들의 경우, 자동차는 허용 가능한 위험성보다 큰 위험성이므로 안전하지 않다고 판단하고 있는 것이다. 그러므로 무엇을 가지고 허용한 위험성으로 할 것인지는 사람, 상황, 대상 기계 설비 등에 따라 다르다.

위험성의 크기로 말하면 허용 가능한 위험성이 수용 가능한 위험성보다 큰 것이 된다. 본래라면 모든 위험성에 대하여 수용 가능한 위험성으로 감소시키고 나서

안전하다고 말해야 하지만 현실적으로는 허용 가능한 위험성의 수준으로 안전하다고 말하게 되고, 거기에는 무시할 수 없는 잔류 위험성이 항상 존재하게 된다.

4. 기능안전성과 안전한 시스템

기능안전성은 시스템이나 장비에 의해 좌우되는 전체적인 안전성의 일부분으로 안전 기능을 수행하는 시스템이나 장치는 조건 입력에 대해 사전에 정의한 대로 의도된 반응이 올바르게 수행되어야 한다.

전기 모터의 권선에 열 센서를 달아 과열되기 전에 동력을 차단하기 위한 과온 방지 장치는 기능안전성의 한 예로 볼 수 있다. 그러나 고온을 견디기 위해 특수화된 절연체를 입히는 것은 기능 안전성에 포함되지 않는다. 안전의 예시이고 동일한 위험원에 대해 방어 또는 보호할 수 있는 위험 방지 대책에는 해당하지만 기능안전성이라고 말하지는 않는다. 기능안전성은 시스템 전체적인 측면과 상호 작용하는 환경을 고려하지 않고서는 결정될 수 없다. 안전 기능은 그 입력에 응답하여 시스템 또는 올바르게 작동하는 장치로 구성되며 유해 사건을 방지하거나 유해 사건의 결과를 감소, 방지하기 위하여 제어 기기 및 장치를 활성화하여 잠재적으로 발생 가능한 위험한 상황을 사전에 검출하는 활성화된 기능이다.

그림 II.1.5 기능안전성과 안전관련 시스템(SRS)의 관계

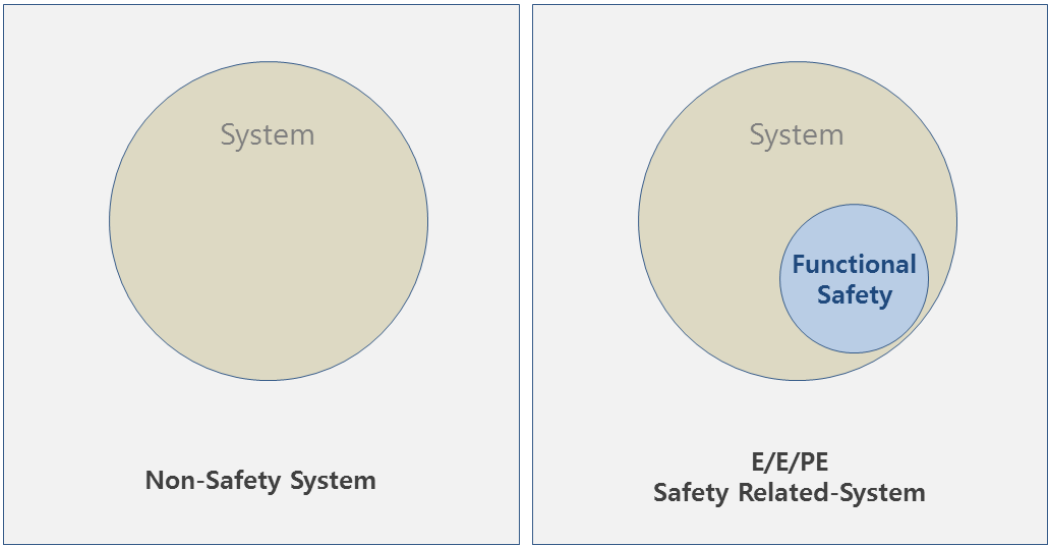


그림 II.1.5 는 기능안전성과 안전관련 시스템의 관계를 보여준다. Non-Safety System 에서는 별도의 Functional Safety 를 요구하지 않지만 Safety Related System 에서는 안전기능이 반드시 필요하다.

일반적으로 안전 설비 및 사전 정의된 환경에서 작동하도록 만들어진 안전 기능인 제어 시스템의 주요 위험원은 분석자 또는 개발자에 의한 위험원 분석을 통해 사전에 식별되어야 한다. 이 분석을 통해서 주요 위험원에 대한 적절한 보호를 위해 기능안전성이 꼭 필요한 것인지 결정하여 설계 시에 적절한 방법을 고려하여 구현할 수 있도록 해야 한다. 기능안전성은 위험원을 다루는 하나의 방법일 뿐이고 설계를 통해 고유의 안전성을 확보하는 것과 같이 위험원을 제거하거나 감소시키는 방법이 가장 중요한 것이다.

'안전관련(Safety Related)'이라는 용어는 특정한 기능을 수행하거나 위험성을 허용 가능한 수준으로 유지되도록 보증하는 안전 기능들을 수행하는 시스템을 설명하는데 사용한다.

안전기능 요구사항은 위험원 분석을 통해 도출되고, 안전무결성 요구사항은 위험성 평가를 통해 도출된다. 안전 무결성(Safety Integrity)은 주어진 모든 조건하에 있는 안전관련 시스템이 주어진 시간 내에 요구되는 안전기능을 만족스럽게 수행할 수

있는 확률로 정의된다. 안전무결성수준이 높을수록 해당 장비 또는 시스템의 고장 발생 가능성은 낮아진다. 이와 같이 어떠한 기술로 구현된 시스템이 안전기능을 수행하면 안전관련시스템(SRS, Safety Related System)이라고 한다. 안전관련시스템은 장치 또는 시스템을 제어하는 기존 시스템과 분리하여 작동하거나, 그 안에 포함되어 일부로서 작동하거나, 기존 제어시스템 그 자체가 안전관련시스템의 역할을 하기도 한다. 안전무결성 요구사항 수준이 높을수록 안전관련 시스템이 더욱 엄격하게 적용해야 한다. 표 II.1.1 은 기능안전성 관련 시스템의 사례를 보여준다.

표 II.1.1 기능 안전에 의존하는 안전관련시스템 예시

- 위해한 화학 장치 공장에서 응급 가동 중단 시스템
- 크레인 안전 하중 지시계(Clane safe load indicator)
- 철도 신호 시스템
- 기계 가드 연동 및 비상정지 시스템
- 속도 제한을 위한 보호용 가변 속도 모터 드라이브
- 의료용 방사선 기계 노출량 제어 및 연동을 실시하는 시스템
- 동적 포지셔닝(선박 입출항시 선박 움직임 제어)
- 선박 항해시 안전 수심확인을 위한 수심측정기(echo sounder)
- 항공 비행 표면 제어의 플라이 바이 와이어 작동
- 자동차 표시등, 브레이크 잠김 방지 및 엔진 관리 시스템
- 네트워크 활성화 장치 설비의 원격 모니터링, 작동 또는 프로그래밍
- 치명적 사고 대비를 위한 정보기반 의사결정 지원 도구
- 원자력발전소 제어 시스템

이러한 시스템은 보통 복잡하고, 모든 고장 모드를 완전하게 사전에 예측하거나 발생 가능한 모든 경우를 사전에 테스트하는 것을 현실적으로 불가능하다. 테스트는 항상 필수적으로 수행해야 하지만, 안전을 위한 충분한 성능을 예측 및 시험하는 것은 매우 어려운 일이다. 기능 안전성 달성을 하기 위해서는 위험한 고장을 방지하거나, 고장이 발생할 때 그것을 제어하는 방식으로 시스템을 설계해야 한다.

5. 소프트웨어의 안전성

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

예를 들어 "원자로를 안전하게 보호할 수 있고, 위험성을 알릴 수 있는 SW 를 개발하자" 라는 안전 요구사항을 기반으로 프로젝트를 시작한다고 하면 먼저 사업계획서를 만들고, 요구사항 명세서를 정리하고, 상세기술 명세서를 만들고, 소스코드를 만드는 등 여러 엔지니어들의 손과 머리를 거치게 된다. 하지만 작업이 끝난 후엔 처음에 의도했던 바와 전혀 다른 결과물이 나올 수 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 목표 달성이라는 측면에서 볼 때 가장 중요한 요소는 세가지 정도 구분된다.

첫째, SW 개발 안전 수명주기에 대한 이해이다. SW 개발 안전 수명주기는 개발 조직 내의 모든 개발자가 숙지하고 각 단계 별 결과물들을 사전에 계획된 바에 따라 생성해 나가야 하는 것인데 수명주기는 경우에 따라 국제표준에서 제시하는 가이드를 일부 테일러링하여 개발 프로젝트에 최적화시킬 수 있다.

둘째, 시스템 및 소프트웨어 위험분석 기법에 대한 이해가 필요하다. 안전한 소프트웨어 개발은 시스템 위험분석을 통하여 도출된 시스템 안전기능 요구사항을 근간으로 개발이 이루어 지기 때문이다. 시스템 레벨에서 도출된 안전기능 요구사항 중에서 소프트웨어가 수행하여야 하는 요구사항들이 소프트웨어 관점에서 명세화되고 이를 바탕으로 소프트웨어가 구현되어야 한다.

셋째, 소프트웨어 안전기능이 의도한 바와 같이 구현이 되었는지를 확인하는 올바른 검증 시험 방법을 적용하는 것이 중요하다. 소프트웨어의 안전성을 검증하는 시험은 디바이스 레벨에서 이루어지며, 검증의 목표는 시스템 안전 요구사항 및 소프트웨어 안전 요구사항이 원래 의도와 같이 구현되었는지를 검증하는 것이기 때문이다. IEC61508 에서는 각 수명주기별 수행해야 할 Level 별 Technique/Measure 를 제시하고 있다.

안전 기능으로 다뤄지는 항목들은 대개 표 II.1.2 의 범주를 벗어나지 않으며, 아래 항목에 있는 부분들은 최상위 안전등급으로 다뤄져야 한다.

표 II.1.2 Safety Critical 소프트웨어 유형

유형	소프트웨어의 안전기능
Fault Detection	Fault Detection, Safety Recovery, Correcting Logic 등
Interrupt Processing	Interrupt Processing 기능, Interrupt Function 을 해제하거나 구동 기능
Autonomous Control	Safety Critical HW 를 독자적으로 직접 제어 기능
SW Controled Movement	위험 발생 가능한 HW 부품을 제어하는 신호 생성 기능, Safety Critical 행위의 초기화하는 기능
Safety-Critical Display	Safety Critlcal HW 시스템의 상태 Display 기능
Critical Data Computation	Safety Critical 데이터를 계산하는 기능

소프트웨어는 하드웨어와 달리 제품이 마모 파손되거나 허용 오차가 증가와 같은 방식으로 오류 발생하지 않으며 일반적으로 구현 오류 (코드 오류, 설계 요구 사항을

잘못 해석) 혹은 외부 소프트웨어에 오류에 대하여 발생한다. 따라서 소프트웨어와 관련된 위험을 평가하는 것은 다소 복잡하다. 정확하게 소프트웨어 오류 발생을 예측할 수 없기 때문에 추가적으로 위험 분류의 방법이 필요하다. 소프트웨어의 동작에 영향을 주거나 일반 요인을 이용하여 위험 분류 방법은 표 II.1.3 과 같다

표 II.1.3 안전관련 소프트웨어의 제어 분류

구분	분류	설명
1	자동	· 사고나 위험의 발생을 배제되고 안전 감지 및 개입이 없는 잠재적으로 안전한 하드웨어 시스템, 서브 시스템, 구성 모듈에 자율 제어 권한이 있는 소프트웨어 기능
2	반자동	· 사고나 위험을 완화 하거나 제어하는 독립적인 안전 메커니즘에 의해 결정된 안전 탐지 및 감지 개입시간을 허용하는 잠재적으로 안전한 하드웨어 시스템, 서브 시스템, 구성 모듈에 제어 권한이 있는 소프트웨어 기능 · 사고나 위험에 완화 또는 제어를 위한 소정의 작업을 실행하는 즉시 운용자 개입이 필요로 하는 안전 중요한 정보표시 · 소프트웨어 예외, 실패, 오류, 지연 허용하거나, 사고 발생을 방지하기 위해 운용자 개입 정보제공
3	복수의 고장 허용력	· 명령기능 수행 완료를 위해 안전 중요 하드웨어 시스템, 서브 시스템, 구성 모듈에 명령을 발생시키는 소프트웨어 기능 · 시스템 감시 및 기능적 반응은 각각 정의된 독립적인 결함 조건에 대한 중복, 고장 허용 메커니즘 포함
4	영향성 있음	· 운용자에 의해 의사 결정을 하는 데 사용되는 안전 관련 특성 정보를 생성하지만 사고를 방지하기 위해 작업자의 조치가 필요 없음
5	안전에 대한 영향성 없음	· 안전에 상당한 하드웨어 시스템, 서브 시스템, 구성 요소에 명령이나 통제 권한을 갖고 있지 않으며 안전에 중요한 정보를 제공하지 않는 소프트웨어 기능 · 안전에 유해 하거나 시간에 민감한 데이터 나 제어 개체의 상호 작용을 필요로 정보를 제공하지 않음 · 전송 또는 안전에 유해 하거나 시간에 민감한 데이터 통신을

		확인하지 않음

다음은 현장에서 많이 발생하는 소프트웨어 안전성을 위배하는 오류들이다.

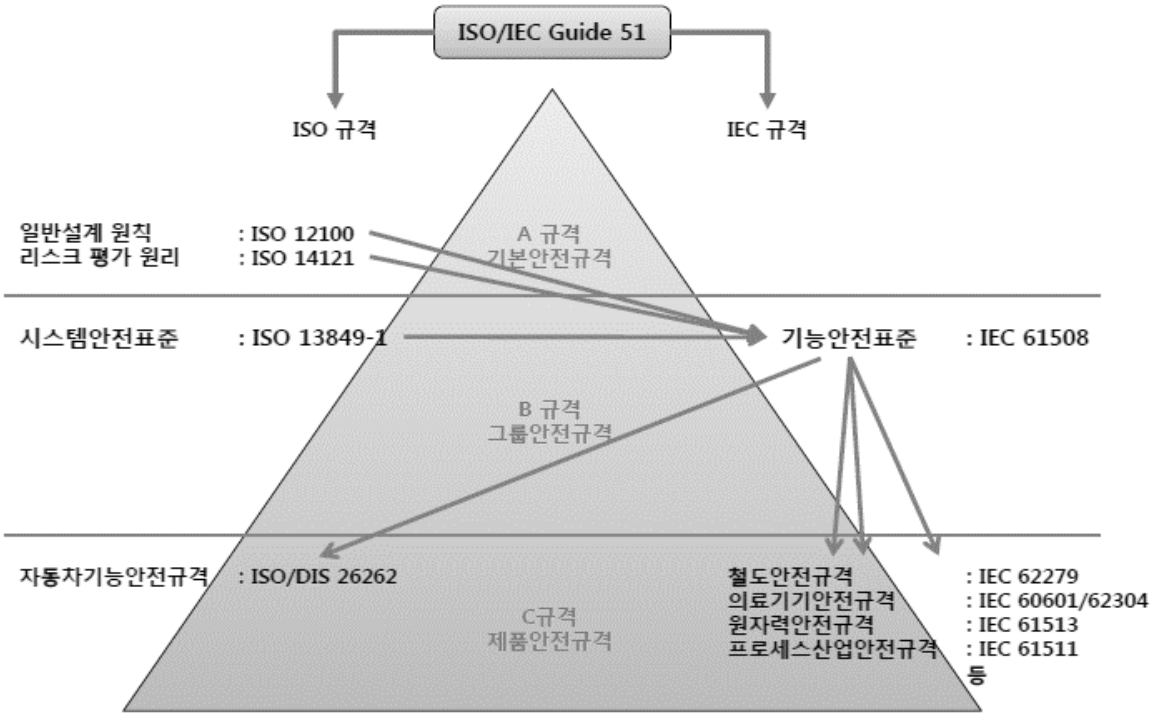
- 시스템, 하드웨어 또는 소프트웨어의 올바른 기능과 요구사항 명세의 불일치, 소프트웨어와 시스템 사이의 인터페이스 불일치
- 안전 요구사항 명세에서 누락(예를 들면, 서로 다른 모드들로 운영되는 동안에 관련된 모든 안전 기능이 나타나지 않는 고장)
- 전체 시스템 관점에서 요구사항 완전성 부족하면 도출된 요구사항을 모두 구현했더라도 전체 시스템 관점에서는 불안전한 상태임
- 요구사항이 시스템 안전을 위해 필요한 특정 행위를 미반영, 요구사항 이외에 소프트웨어가 의도하지 않는 행위를 수행
- 하드웨어 우발 고장 메커니즘을 소프트웨어에 미반영
- 공통 소프트웨어 고장에 의한 영향
- 인적 오류(소프트웨어 조작자의 실수)
- 환경적인 영향(예를 들면, 전자기, 온도, 기계적인 현상들)
- 공급 시스템 전압 불안정(예를 들면, 공급 손실, 전압 감소, 공급의 재연결)

제 2 장. 안전 국제표준

1. ISO/IEC Guide 51

ISO/IEC Guide51 은 제품 규격에 안전에 관한 규정을 도입하기 위한 기본적인 가이드라인으로 가이드라인의 A 규격은 광범위한 제품, 프로세스 및 서비스에 대해서 적용하는 일반적인 안전 측면에 관한 기본 개념과 원칙, 요구사항을 포함하고 있고 B 규격은 몇 개 또는 한 무리의 유사한 제품, 프로세스 및 서비스에 적용할 수 있는 안전 측면을 포함하는 규격으로 IEC61508 규격 등이 이에 해당한다. C 규격은 특정 분야의 제품, 프로세스 또는 서비스의 안전 측면을 포함하는 규격으로 IEC62278, 60601, 61511 등이 이에 해당한다. 여기에서 하위 규격은 상위규격에서 산업분야별로 파생되었거나, 상위 규격에 근거하여 규격이 제정되고 있다. 그림 II.2.1 과 같다.

그림 II.2.1 전기전자 기능안전 규격군



2. IEC 61508

1998 년 IEC 에서는 전기, 전자, 프로그램 가능한 전자시스템의 기능안전(Functional safety of electrical/ electronic/ programmable electronic safety-related systems) 표준으로 IEC61508 을 발표하였으며, 최근 국내외에서는 안전시스템의 복잡화로 인하여 의도된 기능 수행에 대한 확신을 마련하기 위하여 안전시스템의 관리 방법론으로 IEC61508 또는 그와 관련한 표준에 대해 주목하고 있다. 모든 종류의 산업에 적용 가능한 기본적인 기능 안전 표준이 될 의도로 작성되었다.

IEC61508 은 안전생명주기, 하드웨어, 소프트웨어 등 세 가지에 대한 안전성 구현 방법 및 검증 방법을 제시하고 있는데, 안전관련 시스템은 IEC61508 에서 정의된 안전수명주기에 따라 위험분석 및 평가, 안전무결성수준(SIL: Safety Integrity Level)을 설정하고, 하드웨어와 소프트웨어를 목표된 수준(SIL 수준)에 충족하도록 구현하며, 설치, 운영, 유지보수, 변경, 폐기까지 관리해야 한다.

IEC 61508 의 구성은 표 II.2.1 와 같다.

표 II.2.1 IEC61508 의 구성

구분	구성
Part 0	기능안전성과 IEC61508
Part 1	일반 요구사항
Part 2	전자/전자/프로그램 가능한 전자장치 안전 관련 시스템의 요구사항
Part 3	소프트웨어 요구사항
Part 4	정의와 약어
Part 5	안전무결성 수준 결정 방법의 예
Part 6	IEC61508 의 Part2 와 Part3 의 적용 지침
Part 7	기법과 수단의 개요

IEC61508 에서 안전 기능 도출 과정은 6 단계로 이루어져 있으며, 시스템 개념 정의에서부터 출발하여 위험 분석을 통한 안전 요구사항을 도출하고, 최종적으로 안전 기능 요구사항을 도출하게 된다.

IEC61508 에서는 안전수명주기를 정의하고 이에 따른 활동, 절차, 기술을 정의하고 있는데 이들은 위험 검증과 무결성 수준(integrity level)을 만족하도록 설계하는데 필요한 것으로 위해요인(Hazard)분석을 통해 위험감소 대상을 식별하고 ISO9001 과 같이 조직, 프로세스, 인적자격요소 등이 식별된 위험을 감소시키기 위해 어떤 활동을 해야 하는지 이를 다루고 있다. 즉 이 표준에서는 안전 수명주기의 사용으로 시스템의 모든 단계에 관한 시스템적인 상태에 적용되는 안전을 보증하는 것을 뒷받침하며 시스템적인 에러에 관해 가능성을 감소하려는 목적을 가진다. 안전 수명주기 동안 적용되는 시스템 안전 활동은 위험원을 증명, 리스크를 분석, 위험원을 감소 또는 소거하기 위한 설계를 사용하는 것을 지침을 제공하고 있는 것이다. IEC61508 은 전체 개발 단계에서 수행해야할 각 단계별 안전 활동에 대한 지침을 제공하고, 정량적·정성적 리스크 평가(Risk Assessment)법을 제시하고 있다. 그림 II.2.2 은 안전기능 요구사항 도출과정을 보여준다.

그림 II.2.2 안전기능 요구사항 도출과정

IEC61508 에는 기능안전이 구현된 하드웨어에 대한 요구사항이 정의되어 있다. 안전 요구사항에 따라 하드웨어를 설계하고 구현하는 것과 이것들에 대한 계획, 검증, 구조적 제한, 결함방지능력, 시험, 수정시 영향분석을 정의하고 있다. 또한 하드웨어는 정량적인 신뢰성 예측(고장율)을 통해 안전무결성수준을 검증하게 되는데, 다음과 같이 신뢰성 예측기술과 안전무결성수준 검증 기술을 다루고 있다.

- 안전 요구사항 정의, 설계, 검증(validation), 확증(verification), 구조상 제약사항, 결함방지능력(fault tolerance), 시험, 수정활동과 같은 수명주기활동 정의
- 설정된 안전무결성수준 목표치에 대비하여 정량적인 신뢰성 분석을 통한 평가(예측)의 필요성을 설명하고 신뢰성 예측
- 시스템적인 하드웨어 고장에 대비하기 위한 기술과 절차
- SIL 에 대한 구조상 제약사항(architectural constraint) 정의

IEC61508 에는 소프트웨어를 설계하는데 요구되는 활동 및 설계기술의 요구사항도 정의되어 있다. 안전 시스템의 구성 중 소프트웨어 경우 실제 고장률을 측정한다는 불가능하므로 주로 시스템이 결합되는 하드웨어 및 시스템의 전체 고장률 또는 허용 가능한 범위의 고장률을 결정하여 목표하는 SIL(안전무결성)을 결정한 다음, IEC61508 에서 제시하는 단계별 요구사항을 따르도록 하고 있다. 즉 SW 에 대해서는 SIL 에 대한 달성정도를 증명하지 않고, 단계별 기술 요구사항에 따라 수행해야할 활동에 대한 증거를 확인함으로써 목표하는 SIL 을 달성되었다고 가정한다. 이는 소프트웨어의 경우 수명주기 접근 방법에 따라 필수적으로 수행해야 할 활동들에 의하여 결함이 각 수명주기마다 추가되는 가능성을 낮추거나 방지 가능하다는 것을 전제로 하고 있다.

- 소프트웨어라는 특성상, 시스템적인 고장에 대해서 다루고 있으며 정량적인 신뢰성 예측은 포함되지 않는다.
- 소프트웨어 설계기술들에 대해 각 SIL 마다 적용가능성과 수행해야할 활동에 대해서 표로 제공하고 있다.

IEC61508 은 목표안전 달성을 위하여 먼저, 어떤 안전기능을 추가할 것인가를 결정(안전기능 요구사항)하고 그 다음으로 정의한 안전기능의 달성 가능한 정도(안전기능의 성능 정도)를 안전무결성 수준(SIL: Safety Integrity Level)으로 결정하는 절차를 제시한다. 즉, 시스템 전체 안전기능 요구사항은 안전무결성 요구사항(SIL)과 함께 구성되어 각 하위 시스템의 요구사항으로 할당함으로써, 하위 시스템의 구조 또는 각 시스템의 기능들을 구현하는 기술 및 측정법을 결정하게 되는 것이다. 안전 기능요구사항과 안전무결성 요구사항을 시스템 설계측면에서 그 역할의 차이를 구별한다면, 안전 기능 요구사항은 시스템을 구성하는 서브시스템(Sub-system) 또는 컴퍼넌트의 생성에 영향을 미치고, 안전무결성 요구사항은 구조에 영향을 미치는 것이다. 안전무결성 요구사항에 의해 시스템의 구조가 결정이 되고, 시스템의 구조를 어떻게 정의하느냐에 따라서 SIL 즉, 고장확률 수준이 결정된다고 할 수 있다.

안전공학(Safety Engineering) 측면에서 리스크(Risk) 제로상태는 불가능 하므로 허용가능한 수준의 위험까지 리스크를 줄이도록 하는 리스크 평가 및 관리가 오히려 더 중요. 이 점을 고려하여 International Electronic Commitment(이하 IEC)는 완전무결한 시스템은 불가능함을 수용하고 허용 가능한 범위의 리스크에 대해 정의하였다. 즉 IEC61508 의 기능안전 요구사항은 시스템의 위험한 사건의 원인이 될 수 있는 위험원에 대응 하도록 예측 가능한 모든 위험을 제거하고 남은 허용 가능한 위험수준을 안전무결성 요구사항으로 정의하여 시스템의 SIL 을 정의하도록 한 것이다.

또 IEC61508 은 다른 안전 표준들과 달리 단순한 안전 요구사항만을 제시하는 것이 아니라 위험 감소를 위한 정량적인 측정법을 이용하여 구현된 안전 기능이 달성되었음을 평가 및 검증단계와 결합되어 있다. 평가 및 검증은 정성적인 평가뿐만 아니라 고장률, 즉 확률적 고장률을 이용한 안전무결성수준(Safety Integrity Level, SIL) 에 대한 정의를 통해 정량적인 접근을 제공한다.

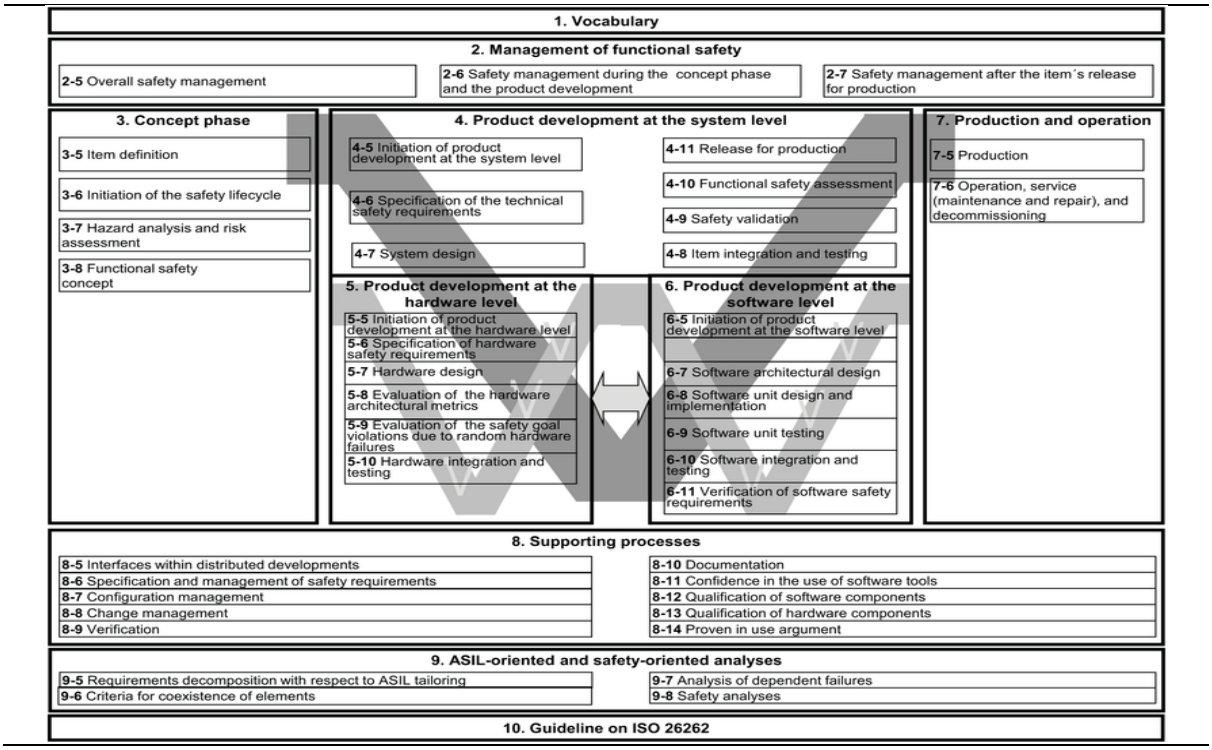
3. 분야별 안전관련 표준 및 가이드

가. 자동차 분야 안전관련 주요 표준 및 가이드

1) ISO26262 표준

ISO26262 는 IEC61508 을 바탕으로 자동차 분야의 적용을 위해 2011 년 11 월에 발표된 표준이다. 자동차 ECU 의 오작동으로 인한 사고 및 인명손실을 최소화 하는 것이 목적이다. ISO26262 는 명세, 설계, 구현, 통합, 검증, 인증에 이르는 개발 전 단계에서 최신 개발 방법 및 테스트 방법을 적용하도록 하고 있으며 기능안전 규격을 준수하기 위한 각 단계에서의 요구사항을 정의한다. 세부적으로는 시스템 레벨, 하드웨어 레벨, 소프트웨어 레벨에서 각 요구사항들이 정의되어 있다. 이는 제품 개념 단계에서부터 폐기까지 전 수명주기에 걸쳐서 전자장치의 고장으로 인한 자동차의 안전성을 저해할 수 있는 위험을 체계적으로 분석하고 그 위험에 대처하는 수단이 효과적임을 보장해야 하는 것으로서 시스템공학, 하드웨어, 소프트웨어 공학, 신뢰성 공학 안전성 분석 및 프로세스 능력이 모두 요구된다.

그림 II.2.3 ISO26262 의 구성



ISO26262 의 세부 구성 내용은 다음과 같다.

Part1 Vocabulary 는 표준에서 사용되는 용어, 정의, 그리고 약어를 설명하고 있으며, 총 142 개의 용어 및 정의와 53 개의 약 어로 구성되어 있다.

Part2 Management of functional safety 는 기능안전 관리를 위한 요구사항을 정의한 파트로 기능안전에 관련된 개발활동을 계획, 조정, 그리고 추적하는 요건에 대해 기술하고 있다.

Part 3 Concept phase 는 아이템 정의를 기반으로 위험원 분석 및 리스크 평가를 통해 ASIL 수준을 판정하며, 안전 목표와 안전 메커니즘을 정의 하는 파트이다. ASIL 은 대상 시스템이 달성하고자 하는 기능 안전성의 수준을 나타내는 것으로 최저 등급인 ASIL A 부터 최고 등급인 ASIL D 총 4 개의 등급으로 구성되어 있다. ASIL 등급이 높다는 것은 해당 시스템이 사고가 날 경우 그 피해가 심각할 수 있다는 것을 의미한다.

Part4 Product development at the system level 은 제품 개발 단계 중 시스템 수준에서의 개발을 명시한다. 시스템 수준의 개발은 기본적으로 V 모델을 따른다.

Part5 Product development at the hardware level 은 하드웨어 수준의 제품 개발을 기술하며, 시스템 설계 명세를 기반으로 하여 시스템의 하드웨어 개발이 이루어진다. 이 또한 시스템 내부에서 다시 V 모델을 따르며 개발, 통합, 검증에 대한 요구사항을 포함한다.

Part6 Product development at the software level 은 소프트웨어 수준의 제품개발 또한 V 모델을 따르며 개발, 통합 검증에 대한 요구사항을 정의하고 있다.

Part7 Production and operation 은 제품의 생산, 운영, 서비스, 그리고 폐기를 위한 요구사항을 포함한다.

Part8 Supporting processes 는 안전 요구사항의 명세 및 경영, 형상관리, 변경 관리, 검증, 문서화, 소프트웨어 도구 사용에 대한 신뢰, 사용증명 논거/주장 등에 대한 요구 사항을 정의하고 있다.

Part9 Automotive Safety Integrity Level (ASIL) – oriented and safety-oriented analysis 는 ASIL 과 안전에 기반한 분석을 위한 요구사항을 기술하고 있다.

Part10 Guideline on IOS26262 는 주요 개념, 안전 케이스, ASIL 분해 등 ISO26262 의 이해에 도움이 되는 정보를 기술하고 있다.

ISO TC22/SC32/WG8 은 ISO 26262 제정 후, 3 년간의 산업계 경험 및 의견을 반영하고자, 2nd edition 으로 개정하는 결정을 하였고, 2015 년 1 월 독일 베를린에서 회의를 시작하여 아래와 같은 일정으로 개정 작업을 진행 중이다.

ISO 26262 2nd edition 에서는 기존의 3.5 톤 이하 승용차에 대해서만 적용되던 범위를 상용차(Truck & Bus) 및 오토바이까지 확대한다. 이에 따라 Part1 에 PTO, Trailer, Body Builder 등 상용차 전용 용어를 추가하고, Part3 에는 트럭 및 버스의 상황에 맞는 S, E, C 에 대한 Table 을 추가하고 파생 차종(variance) 다양화에 따른 H&R 을 기술하였다. 또한 Part 11 에 반도체 설계 분야에 대한 내용이 새롭게 추가되었으며 칩 성능과 패키지 형태, 동작 시간, 온도 환경에 따라 반도체가 고장 날 확률이 어느 정도인지 계산하는 고장율 예측에 대한 내용이 포함된다. 2nd edition 의 주요 개정 내용을 요약하면 아래와 같다.

표 II.2.2 ISO 26262 2nd edition 주요 개정 내용

Part	Title	주요 개정 내용
Part 1	용어 정의	fault injection 에 대한 정의 추가 Truck & Bus 를 포함한 차량 범위 확대 및 상용차 전문 용어 반영 Failure Time Interval, Fault Tolerant Time Interval 에 대한 정의 추가 System, reasonable/unreasonable risk, safe state 에 대한 정의 추가
Part 2	기능안전 관리프로세스	functional safety 와 cyber security 간 연계되는 프로세스 측면에 대한 요구사항 정의 FSC, TSC 에 대한 Confirmation review 추가 PIU, Safety case 에 대한 CR 제거 ASIL A 에 경우 safety plan 에 대한 confirmation review 수행

Part	Title	주요 개정 내용
		ASIL B 에 대한 safety assessment 수행 Confirmation measure 에 대한 independence level 조정 safety case 에 대한 confirmation review 를 수행함(단, completeness 에 대한 check 는 하지 않음)
Part 3	Concept 설계	트럭 및 버스 상황에 맞는 S, E, C, table 추가 파생차종(Variance) 다양화에 따른 H&R 유의사항 명기 (Bus vehicle, configuration, operation 상황 고려 및 모든 파생 차종 조건을 고려하여 위험도 분석) Fault handling time 에 대한 새로운 정의 추가
Part 4	System 개발	트럭 및 버스의 파생차종 다양화에 따른 대표 검증(V&V) 명기 Cybersecurity 관련 part4 의 normative requirements 해결 방안 (현재 part2 의 Annex F 에 관련 informative 정보가 정의되어 있음) safety measure 를 safety mechanism 이라는 용어로 변경 diagnostic test interval 은 diagnostic test strategy 로 변경 verification plans 와 관련된 사항은 part 2 clause 6 로 정리됨. verification report 와 관련된 사항은 part 8 에 정의됨 (consistency 를 유지하기 위해) 따라서 이를 reference 하는 것이 적절할 것을 판단됨" 관련 요구사항 삭제 DFA(Dependent Failure Analysis)는 공통적인 부분이기 때문에 일관성 유지를 위해 part 9 에 통합"
Part 5	HW 개발	scaling factor 에 대한 요구사항 제거 PMHF 목표값 불충족시 정량적인 대안 방법 또는 PMHF, Cut-set 과 동등한 방법으로서 RRA 적용 item complexity(n 개 제어기로 구현되는 경우)에 따른 PMHF 목표값 증가하는 개념 추가 하드웨어 개발시, FSC, TSC 수준에서 할당되어 결정된 목표값 적용 FSC 가 존재하지 않을 시(예 tier 1 모듈설계) table 6 목표값의 10% 계산 Safety fault 에 대한 재 정의 Diagnosis Coverage 에 대한 “proper rationale” 개념 정의 Hardware safety requirements 의 allocation 수준 확립 HW safety mechanism 의 effectiveness 분석 수행 기준 수립

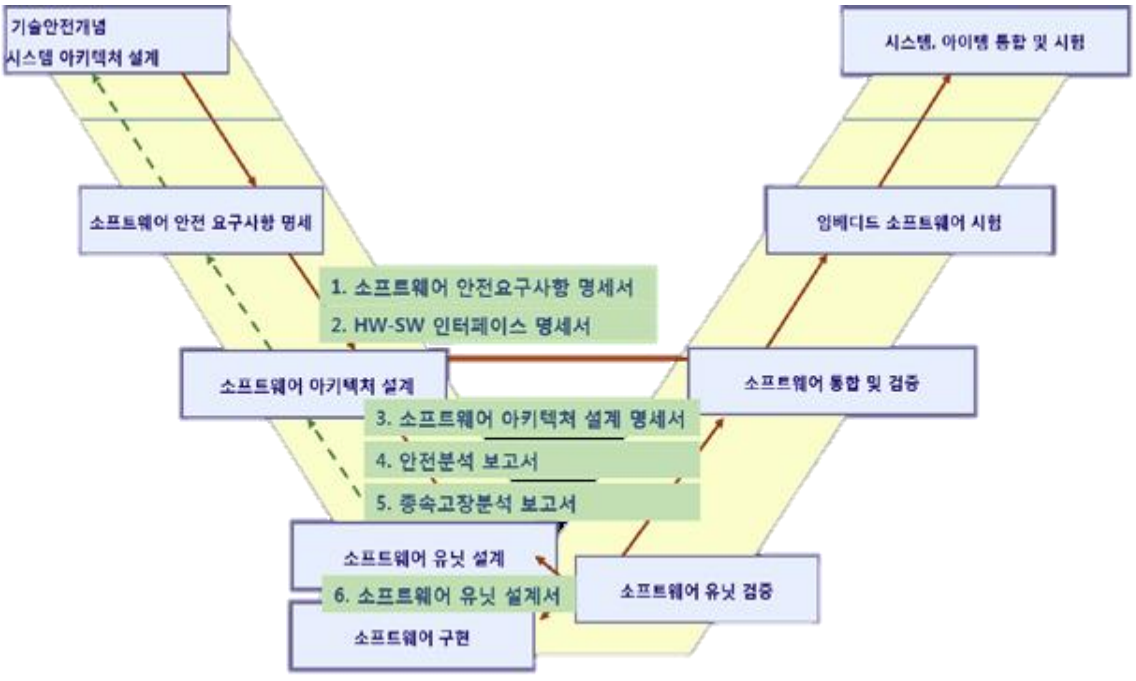
Part	Title	주요 개정 내용
		HW part failure rate 산출 관련 “artificial reduction” 논의 Non-single system 에 대한 PMHF 목표값 설정 기준 수립 FMD, FMC 기반 RF, MPF-L 고장율 산출 기준 수립 Plausible DPF evaluation 방법론 재정의 Failure split methods 를 failure mode distribution 으로 변경 “Sufficient low probability of occurrence”를 입증하는 방법에, dedicated measure 에 추가로 고장율 기반 방법을 추가로 정의 수행
Part 6	SW 개발	Annex E guidance 추가 Annex B Model based development 을 V 모델에 기반하여 normative, informative 요구사항을 상세화 SW planning 활동에 대한 단순화
Part 7	제품생산, 운용, 수리 등	Truck & Bus 의 bodybuilding, re-building, multi-stage build 및 supplier(T1, T2)의 ISO 9001, ISO TS 16949 에 대한 내용을 정의할 것인가에 대한 논의
Part 8	지원 프로세스	TI scale 변경에 의한 TCL 및 qualification method 변경 Off-road PTO example 을 16 장에 추가 5. Interfaces with distributed developments 의 범위를 전체로 확장 분산개발계획시 사전준비사항에서 DIA 를 삭제하고 RFQ 는 유지 Assessment 시 제 3 자에 대한 지명권을 OEM 과 부품사 모두 가능하도록 개정
Part 9	ASIL 엔지니어링	Coexistence criteria, Decomposition, Dependent failure 설명 추가 dependent failure analysis 시 ASIL 별 권고되는 방법 정의₩
Part 10	기능안전 가이드라인	FTTI, PMHF, Fault tolerant 내용 일부 변경
Part 11	차량용 반도체	ISO 26262 2nd edition 에 추가되는 part 로써, title 은 "Application of ISO 26262 to Semiconductor"임. DPAS 19451-1 의 내용 및 ISO 26262-10 Annex A, new topic 포함 MEMS, Fault Injection test at IC level 포함

2) 자동차 전자제어장치분야 SW 신뢰 안전성 가이드 소개

가이드에서는 ISO 26262-6 에 정의된 소프트웨어 수명주기와 각 프로세스 단계별 요구사항을 정리하여 전체 프로세스에 대한 설명을 다룬다. 프로세스 단계는 다음과 같다.

- 소프트웨어 안전요구사항 명세 단계
- 소프트웨어 아키텍처 설계 단계
- 소프트웨어 단위설계 및 구현 단계
- 소프트웨어 단위시험 단계
- 소프트웨어 통합 및 시험 단계
- 소프트웨어 안전 요구사항 검증 단계

그림 II.2.4 ISO26262 기반 안전 가이드 적용 개발 프로세스



각 프로세스 단계에서는 아래와 같은 내용이 상세히 설명되어 있다.

- 해당 프로세스의 목표
- 해당 프로세스를 수행하기 위해 필요한 입력물

- 해당 프로세스의 활동 요약
- 해당 프로세스의 활동 요구사항
- 해당 프로세스의 결과로서 생성되는 산출물
- 참고표준

그림 II.2.5 작성된 가이드 요구사항 명세 세부 내용 예시

3.1 소프트웨어 안전 요구사항 명세

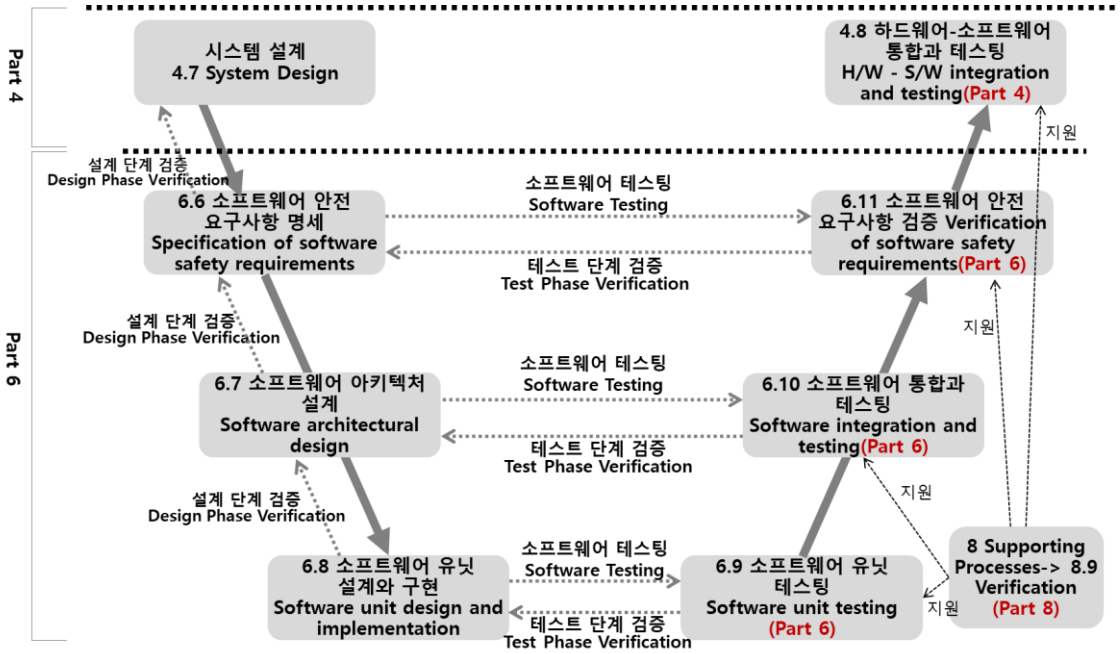
Attribute	Description
목표 (Objectives)	• 소프트웨어 안전 요구사항을 명세 • 하드웨어와 소프트웨어간의 인터페이스 요구사항 상세화 • 소프트웨어 안전 요구사항과 하드웨어와 소프트웨어간의 인터페이스 요구사항이 기술안전 개념과 시스템 설계 명세서와 일치함을 검증
입력물 (Prerequisite)	• 기술안전 개념 • 시스템 설계 명세서 • 하드웨어-소프트웨어 인터페이스 명세서 • 안전 계획(갱신) • 소프트웨어 검증 계획
활동(Activities)	하드웨어의 제약들과 이런 제약들이 소프트웨어에 미치는 영향을 고려한 소프트웨어 안전 요구사항 명세서 작성
활동 요구사항 (Requirement of Activities)	
(1) 소프트웨어 안전 요구사항 명세서 작성 1) 소프트웨어 안전 요구사항은 소프트웨어 기반의 기능 고장으로 인해 소프트웨어에 할당된 기술안전 요구사항의 위반을 야기할 수 있는 모든 소프트웨어 기반의 기능을 다루어야 한다. 2) 소프트웨어 안전 요구사항 명세는 기술안전 개념과 시스템 설계서로부터 생성되어야 하며 다음 사항을 고려해야 한다. • ISO 26262-8, 6 절에 따른 안전 요구사항의 명세 및 관리 • 명세된 시스템과 하드웨어 설정 • 하드웨어-스프트웨어 인터페이스 명세서(HSI) • 하드웨어 설계 명세서와 관련된 요구사항 • 시간제약 • 외부인터페이스 • 소프트웨어에 영향을 미치는 차량, 시스템 또는 하드웨어의 작동모드 3) ASIL 분해가 소프트웨어 안전 요구사항에 적용되면 ISO 26262-9:2012, 5 절을 준수해야 한다. 4) 하드웨어와 소프트웨어간 인터페이스 명세서는 하드웨어의 올바른 제어 및 사용이 가능한 수준으로 자세하게 설명되어 있어야 하고 각각의 하드웨어와 소프트웨어 사이의 안전과 관련된 종속 관계가 기술되어야 한다. 5) 안전 요구사항에 추가로 임베디드 소프트웨어에서 수행되는 그 외의 안전 요구사항을 위한 다른 기능들이 존재한다면 이런 기능들을 명시하거나 그 명세에 대한 참조가 있어야 한다.	

(2) 소프트웨어 안전 요구사항 검증(Verification) 1) 소프트웨어 안전 요구사항, 하드웨어와 소프트웨어간 인터페이스의 갱신된 명세서에 대한 검증은 ISO 26262-8:2012, 9 절에 따라 계획되어야 한다. 2) 갱신된 하드웨어와 소프트웨어간 인터페이스 명세서는 시스템, 하드웨어, 소프트웨어 개발 담당자가 공동으로 검증해야 한다. 3) 소프트웨어 안전 요구사항과 갱신된 하드웨어와 소프트웨어간 인터페이스 요구사항은 다음 사항을 보여주기 위해 ISO 26262-8, 6 절과 9 절에 따라 검증해야 한다. • 기술안전 요구사항과의 부합 및 일치 • 시스템 설계와의 부합 • 하드웨어-소프트웨어 인터페이스와의 일관성	
산출물 (Work product)	• 소프트웨어 안전 요구사항 명세서 • 하드웨어-소프트웨어 명세서(갱신) • 소프트웨어 검증계획(갱신) • 소프트웨어 검증 보고서
참고 표준 (Reference)	ISO 26262-6, 6 절 소프트웨어 안전 요구사항 명세 (Specification of software safety requirements)

Table 19: 소프트웨어 안전 요구사항 명세

본 가이드는 ISO 26262 의 소프트웨어 테스트 메소드 내용을 주로 다루며, 하드웨어는 본 가이드의 범위를 벗어난다. 하드웨어와 소프트웨어 통합 관련 테스트 메소드도 다루지만 역시 소프트웨어 테스트 메소드 내용이 중심이다.

그림 II.2.6 ISO 26262 Part 6 소프트웨어 개발 모델



ISO 26262 Part 8.9 Verification(검증)의 Verification(검증) 가이드 범위는 ISO/IEC/IEEE 29119 를 기반으로 소프트웨어 테스트로 한정하고 테스트 산출물 템플릿의 구조와 템플릿 구성 목차를 주로 설명한다.

나. 철도 분야 안전관련 주요 표준 및 가이드

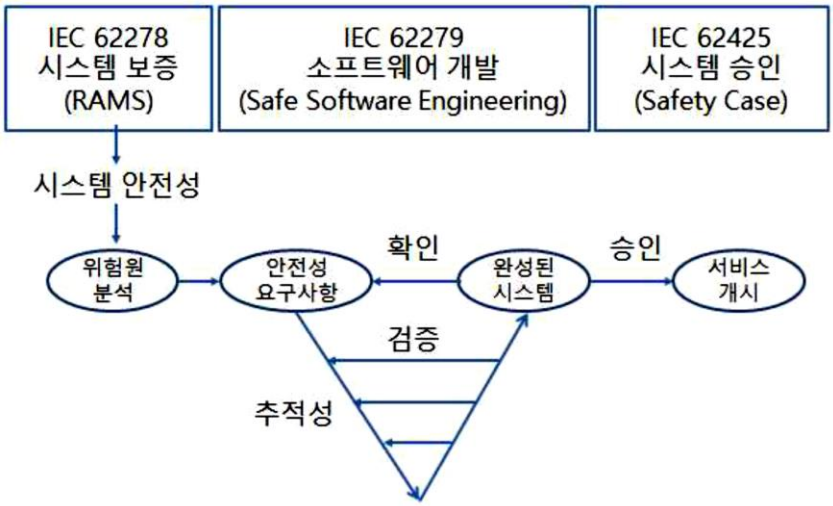
1) 철도분야 국제 안전관련 표준

IEC 62278(또는 EN 50126, 이하 표준)은 Railway application-specification of demonstration of Reliability, Availability, Maintainability and Safety(RAMS)에 관한 기술적 내용을 다루며 철도 시스템 RAMS 관리 원칙을 제공하는 국제 표준이다. 표준은 개념설계부터 폐기까지 생명주기 모든 단계에서 안전성과 신뢰성 등을 확보하기 위한 각 단계별 과정과 절차에 대해 다룬다. 본 표준은 철도 차량 뿐만

아니라 전력설비 등 철도 시스템 전반을 대상으로 하여 다음과 같은 기본적인 개념을 다룬다.

철도 신호와 관련된 제어, 명령 및 보호(control, command and protection) 시스템의 안전성을 높이기 위해서 국제 표준이 제정되었다. 철도 선진국인 유럽에서 주도하여 유럽 표준인 CENELEC EN 50126, EN 50128, EN 50129 를 먼저 만들었다. 그 후에, 이들 유럽 표준은 국제 표준 IEC 62278, IEC 62279, IEC 62425 로 각각 제정되었다. 그림은 국제 표준들 간의 관련성이다.

그림 II.2.7 철도 관련 국제 표준의 관계



철도 시스템과 장치사양 및 장치의 개발/운용의 각 프로세스에 대해 안전 요구사항을 규정하는 EN 계열의 안전성 표준들과 철도 시스템의 안전성을 확보하기 위한 RAM 관리 및 평가에 대한 요구사항들이 IEC 표준화 되었다. 국내/외 철도 산업에서 활용되는 표준은 아래표와 같다.

표 II.2.3 철도분야 안전표준 주요 특징 및 설명

표준명	표준 명칭	주요내용

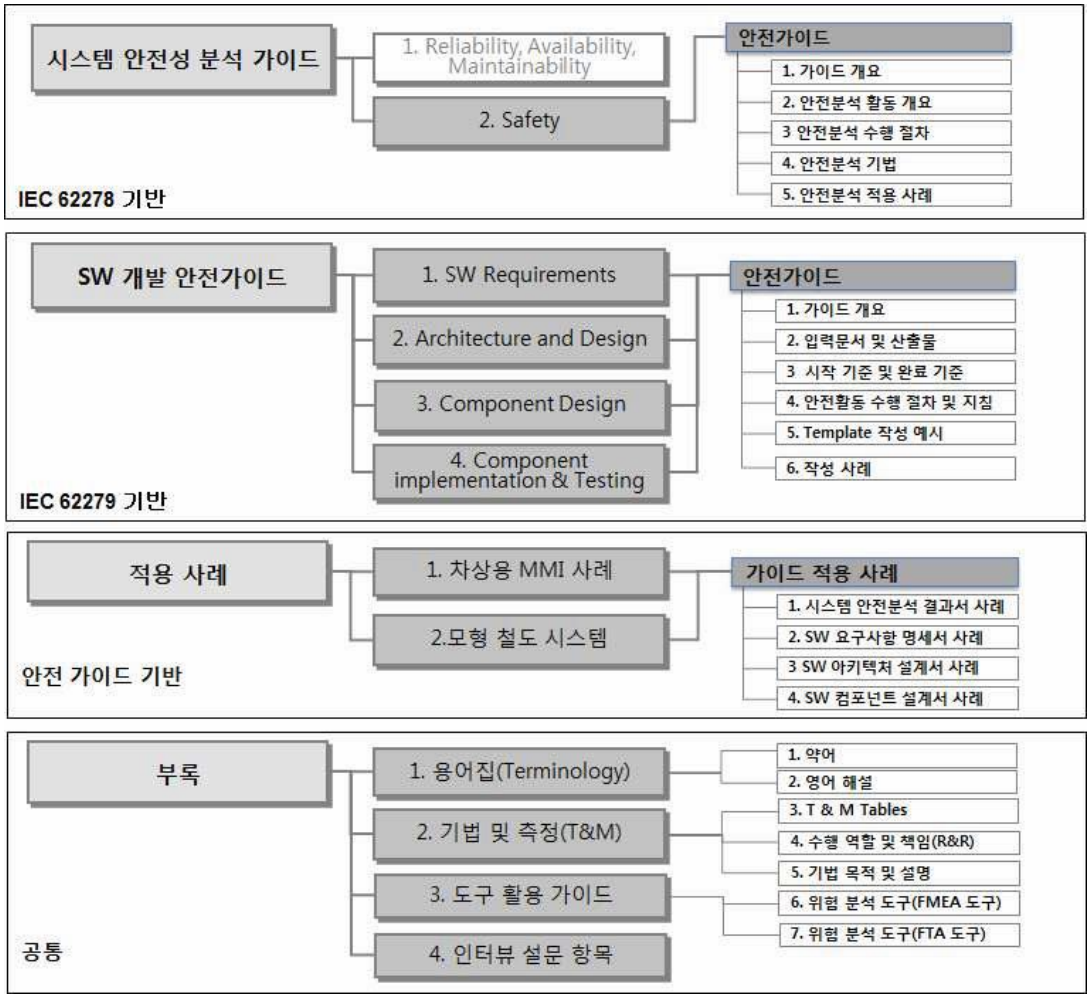
IEC 62278	Railway applications - Specification and demonstration of reliability, availability, maintainability and safety(RAMS)	RAMS 규격으로, 철도기관과 철도 관련 사업을 위해 신뢰성, 가용성, 유지보수성 및 안전관리를 지속적으로 수행하기 위한 전체 생명주기 14 단계에 대해서 세부적인 RAMS 활동에 대해 정의하고 있다. 따라서 RAMS 요구사항을 개발하고 이행하기 위한 기준을 제공한다.
IEC 62279	Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems	철도 시스템에 대한 소프트웨어 규격으로 철도 분야의 안전관련 소프트웨어의 개발, 시험, 검증 및 유지보수와 준수해야 할 일련의 요구사항을 제공한다.
IEC 62280	Railway applications - Communication, signalling and processing systems - Safety related communication in transmission system	통신 규격으로 전송 시스템의 안전 관련 통신에 대해 고려해야 할 요구사항을 제공하고, 안전 관련 전자 시스템이 다른 장소간의 정보를 전송할 경우, 전송 시스템은 안전 관련 시스템의 필수 부분이 되고 통신이 안전하다는 것이 증명 되어야 한다.
IEC 62425	Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signaling.	철도 신호분야에서 안전 관련 전자 시스템의 승인을 위한 요구사항을 정의한 규격이다. 신호용 안전 관련 전자시스템은 하드웨어와 소프트웨어 측면이 모두 고려되어야 하는데, 이 규격은 안전관련 하드웨어와 전체 시스템에 대한 요구사항을 제공한다.

2) 철도분야 SW 신뢰 안전성 가이드 소개

철도 안전 가이드는 철도관련 시스템을 설계하고 개발을 담당하는 조직 및 이해관계자들이 철도 분야 국제 표준을 실무 수준에서 쉽게 이해하고 적용 가능하도록 시스템 개발 생명주기 각 단계 별 안전 활동의 구체적인 수행 절차와 지침, 실사례 등을 포함한 상세 가이드를 개발하고 제공함으로써 신규 철도 시스템의 도입 및 구축, 기존 철도 시스템의 개선 및 변경 시 철도 시스템의 안전성을 효과적으로 적용하고 향상시킬 수 있도록 지원하는데 그 목적이 있다. 또한 가이드를 적용하고 활용함으로써 철도 분야 소프트웨어 시스템 개발에 종사하는 개발기관의

기능 안전성 확보를 위한 소프트웨어 공학 경쟁력을 제고 하는데도 그 목적이 있다. 이러한 목적을 달성하기 위해서 기존에 작성된 가이드를 개선하고자 한다. 참고로, 기존 가이드의 구성은 아래 그림과 같다.

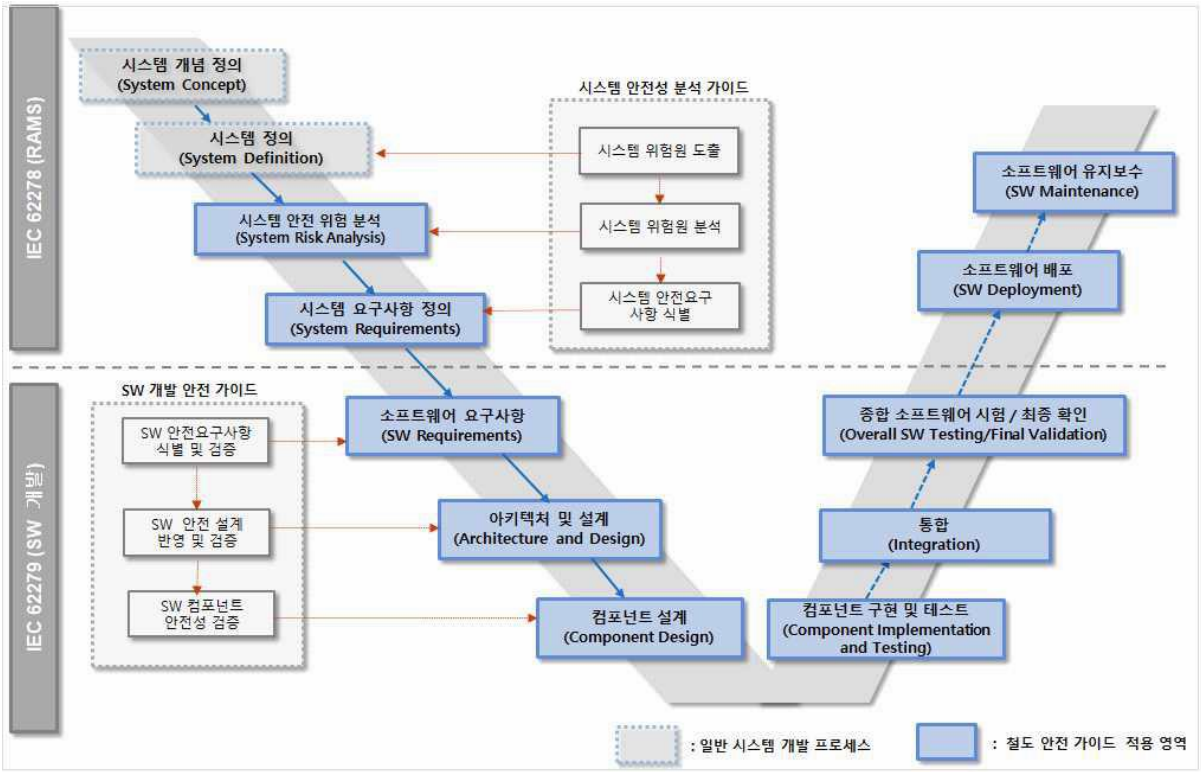
그림 II.2.8 철도 안전 가이드 구성



2017 년도에 개선된 내용은 크게 두 가지이다. 첫째, IEC 62278 기반으로 실무에서 널리 사용되는 안전성 분석 및 위험도 평가 방법인 PHA, SHA, SSHA, IHA, O&SHA, ETA, FTA, FMEA, FHA, HAZOP, FRACAS 을 상세하게 안내하고 해설한다. 이전 가이드에서는 일부 기법만 다루었으나, 본 과제에서는 실무에서 사용되는 안전성 분석 및 위험도 평가와 관련된 거의 모든 기법을 다룬다. 둘째, IEC 62279 에 명시된

소프트웨어 전 단계를 지원하도록 기존 가이드를 확장한다. 이전 가이드에서는 시간 제약으로 인해서 요구사항, 아키텍처 및 설계, 컴포넌트 설계, 콤포넌트 구현 및 테스팅 단계만을 다루었다. 즉, V 모델로 얘기하면, 위에서 아래로 내려가는 왼쪽 영역을 다루었다. 본 과제에서는 통합단계, 종합 테스트 단계, 배포 단계, 유지보수 단계를 추가함으로서 IEC 62279 소프트웨어 개발 전 단계를 지원한다.

그림 II.2.9 철도 안전 가이드 적용 범위



시스템의 잠재 위험원을 식별하고, 식별된 위험원을 분석해 이를 제거하거나 일정수준 이하로 관리하는 안전 요구사항을 도출하는 일련의 수행 절차와 기법 등을 포함하는 시스템 안전성 확보 활동은 시스템 안전성 분석 가이드를 적용해 안전

목표를 달성할 수 있다. 시스템 위험원과 안전 요구사항으로부터 소프트웨어 안전 요구사항을 식별하고 이를 소프트웨어 아키텍처 설계 및 컴포넌트 설계에 반영, 통합, 검증하는 수행절차와 기법 등을 포함하는 소프트웨어 안전성 확보 활동은 소프트웨어 개발 안전가이드를 적용함으로써 안전 목표를 달성할 수 있게 된다.

가이드는 시스템 및 소프트웨어 개발 가이드를 적용함에 있어 안전성 활동에 필요한 선행 산출물, 활동 수행에 따른 결과물, 세부 수행 지침, 분석 기법, 안전 무결성 등급에 따른 점검 방법 등을 다음의 단계 별 안전가이드 표를 활용함으로써 시스템 안전성 활동의 전반적인 내용을 용이하게 이해할 수 있게 작성되어 있다.

다. 의료기기 분야 안전관련 주요 표준 및 가이드

1) 의료기기 분야 국제 안전관련 표준

의료기기 소프트웨어와 관련된 주요 국제표준은 IEC 62304:2005 Medical software – Software life cycle processes, ISO 13485:2016 Medical devices –Quality management systems – Requirements for regulatory purposes 등이 있으며, 이러한 의료기기 소프트웨어 국제표준을 개발하고 있는 국제표준 전문위원회에는 IEC TC 62 SCA 와 ISO TC 215 JWG 이 있다. 아직까지 의료기기 소프트웨어를 위하여 개발된 표준들은 많지 않으나 최근 의료기기 소프트웨어 사용환경과 적용되는 기술들의 변화로 관련 표준들의 개정이 활발히 이루어지고 있다.

국내 의료기기는 의료기기법을 기반으로 하여 시행령과 시행규칙이 수립되어 있으며, 실제 의료기기에 대한 구체적인 규제 요구사항들은 규제기관인 식품의약품안전처에서 고시 형태(총 23 종)로 개발하여 운영하고 있다. 국내 의료기기는 사용목적과 인체에 미치는 잠재적 위험성 등의 차이에 따라 의료기기 등급을 4 개로 분류하고 있다. 의료기기 소프트웨어 관련 요구사항들은 아래와 같이 정리할 수 있다.

- 의료기기 허가 시 기술문서 작성에서 의료기기 소프트웨어 관련 요구사항
- 의료기기 소프트웨어 적합성 확인보고서 관련 요구사항

- 의료기기 소프트웨어 성능개선 허용 관련 요구사항
- 의료기기 소프트웨어 변경 요구사항 (경미한 변경 포함)

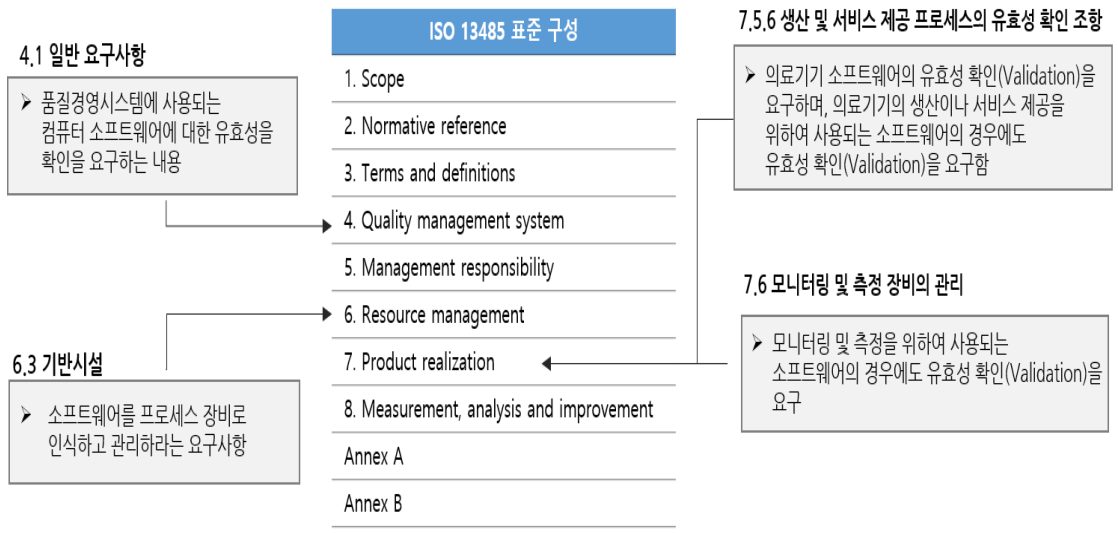
의료기기 시장의 높은 점유율을 차지하고 있는 주요국가의 의료기기 소프트웨어 규정 현황은 다음과 같다.

표 II.2.4 국외 주요국 의료기기 소프트웨어 관련 규제 현황

국가	의료기기 등급 분류	의료기기 소프트웨어 관련 규정
미국	3 개	Guidance for the Content of Premarket Submission for Software Contained in Medical Devices
유럽	4 개	Harmonized Standard (EN ISO 13485:2012, EN 60601-1-4:1996 등)
중국	3 개	YY/T 0664-2008 *Medical device software – Software life cycle processes*
일본	4 개	JIS T 2304:2017 *Medical device software – Software life cycle processes*
브라질	4 개	INMETRO Ordinance 54/2016 REQUIREMENT OF COMPLIANCE ASSESSMENT FOR EQUIPMENT UNDER REGIME OF HEALTH SURVEILLANCE

의료기기를 개발하고 생산하는 조직은 규정에 따라 반드시 품질관리시스템을 수립하고 유지해야 한다. 이에 적용되는 표준으로 ISO 13485:2016 Medical devices — Quality management systems — Requirements for regulatory purposes 표준이 있다. ISO 13485 표준은 2016 년에 개정되었으며, 아래 그림을 통하여 13485:2016 구성 및 의료기기 소프트웨어 관련 주요 요구사항을 살펴볼 수 있다.

그림 II.2.10 ISO 13485:2016 구성 및 소프트웨어 관련 주요 요구사항



의료기기 소프트웨어가 그 자체로서 의료기기가 될 수 있고 의료기기의 일부가 될 수 있기 때문에 반드시 위험관리를 필수적으로 수행하여야 한다. ISO 14971:2012 Medical devices — Application of risk management to medical devices 표준은 의료기기의 위험관리를 위해 개발된 표준으로 의료기기 소프트웨어 개발 시 반드시 ISO 14971 표준을 적용해야 한다.

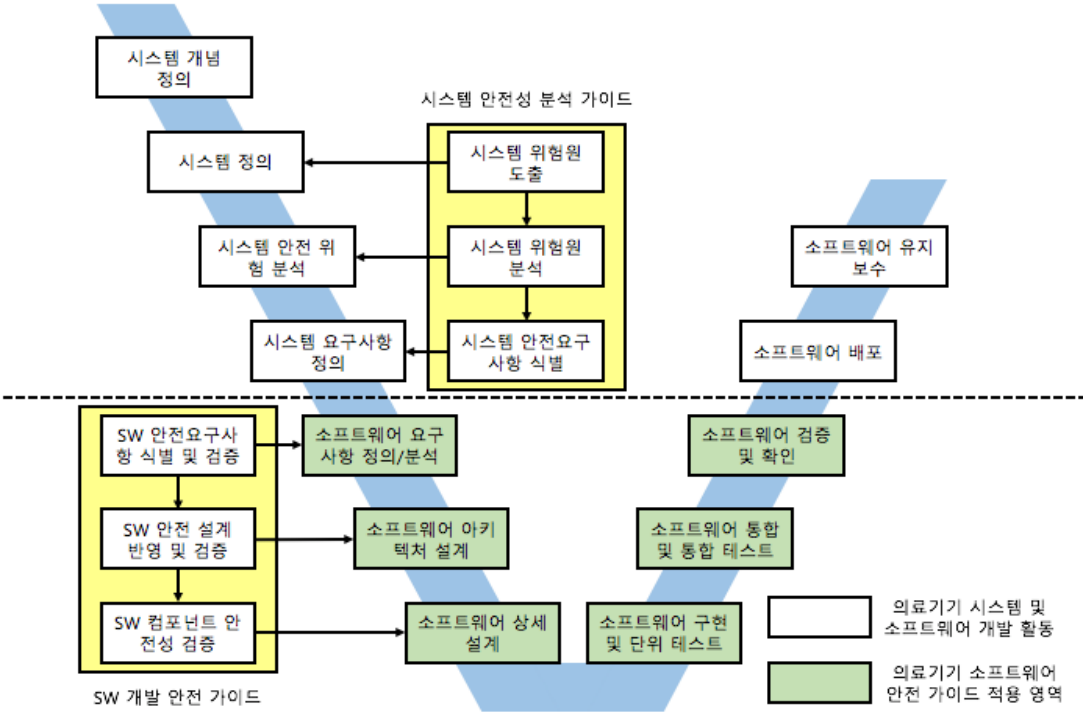
2) 의료기기 분야 SW 신뢰 안전성 가이드 소개

본 가이드는 의료기기 소프트웨어 표준을 기반으로 개발 프로세스를 포함하여 품질관리, 위험관리 등 의료기기 소프트웨어 개발 전체 단계에 대한 프로세스를 제시하였다. 또 의료기기 소프트웨어 관련 국내외 표준 현황 및 국내외 주요국 규제 등을 분석하여 관련 실무자가 의료기기 소프트웨어 분야의 동향을 파악하는데 도움을 주고자 하였다. 특히 의료기기 분야에 대한 국내 IT 기업의 대응 현황 분석 및 시범 적용 과정에서 관련 기업들의 고충과 애로 사항 조사를 통하여 본 가이드의 실용성을 확장시키고, 향후 가이드 고도화에 대한 니즈와 방향을 제시하고자 하였다.

소프트웨어를 포함한 의료기기의 안전성 및 유효성을 확고히 하기 위하여 소프트웨어의 의도와 해당 소프트웨어가 예기치 못한 위험 없이 의도된 사항들을 수행하는 것이 필요하다. 『의료기기 분야 소프트웨어 안전성 및 유효성 확보를 위한 가이드』에서는 IEC 62304:2006 Medical software – Software life cycle processes 와

ISO 14971:2012 Risk management for medical devices 을 기반으로 의료기기 소프트웨어의 안전 설계 및 유지보수에 필요한 소프트웨어 생명주기 프로세스를 제시한다. 다음은 의료기기 소프트웨어 개발 시 적용 가능한 가이드의 범위를 나타낸 것이다.

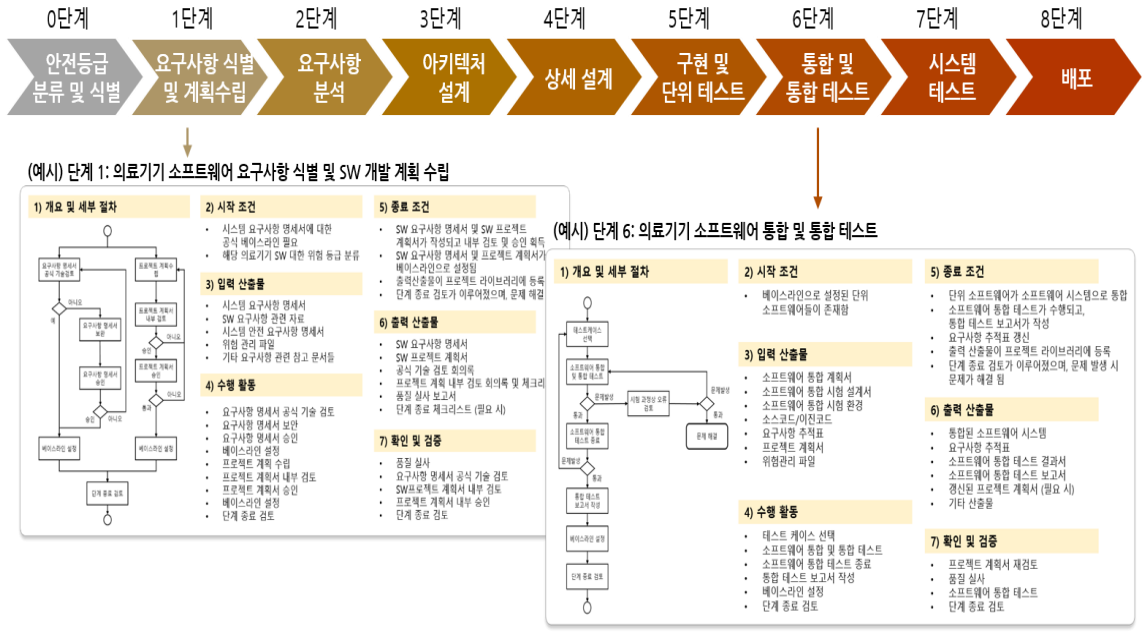
그림 II.2.11 의료기기 소프트웨어에 대한 안전가이드 적용 범위



본 가이드는 의료기기 분야 관련 표준 및 규제·규정·인허가 동향 등에 대한 전반적인 내용을 수록하여 현황에 대한 실무자들의 이해를 돕고자 하였고, 개발 시 참고할 수 있도록 의료기기 규제에 적용되는 의료기기 소프트웨어 분야의 국제 표준을 기반으로 개발 및 위험관리, 유지보수 등 프로세스와 수행 활동, 입출력 산출물을 제시하였다.

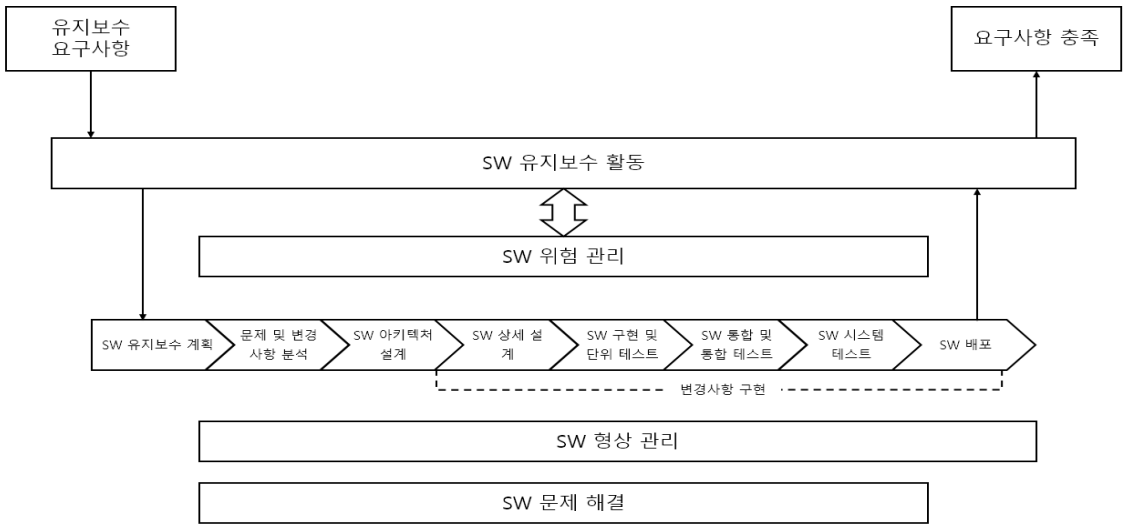
의료기기 소프트웨어 개발 생명주기 단계에 따라 세부 절차, 시작 조건, 입력 산출물, 수행 활동, 종료 조건, 출력 산출물, 확인 및 검증에 대한 구체적인 내용을 제시하고 있다. 의료기기 소프트웨어 개발 단계는 총 9 단계로 요구사항 식별 및 계획수립, 요구사항 분석, 설계, 테스트, 배포 등의 순서로 구성되어있다.

그림 II.2.12 의료기기 소프트웨어 개발 생명주기 단계 별 수행 활동 및 입력 산출물



개발된 의료기기 소프트웨어에 대하여 보고된 문제 발생 상황 또는 사용자 및 관련자의 변경 요청에 대응하기 위한 유지보수 활동을 정의한다. 유지보수 프로세스는 소프트웨어 개발 생명주기를 모두 완료하여 시스템 테스트를 마친 소프트웨어에 적용되며, 주요 활동으로는 소프트웨어 작동 시 발생한 결함의 복구, 신규 기능의 추가 등이 있다. 유지보수란 소프트웨어 시스템에 대한 추가적인 개발 활동을 하는 것이므로 활동의 흐름은 개발 프로세스에 준하여 실시한다.

그림 II.2.13 의료기기 소프트웨어 유지보수 프로세스



라. 그밖의 안전관련 분야

항공에는 DO178C, 원자력에는 IEC61513 등의 표준이 있으나, 조선분야의 경우 아직 안전성 관련 국제 표준이 존재하지 않고, 선주나 선급의 요구에 따라 기능이 만들어지는 실정이다. 따라서 본 가이드는 안전이 중요하지만 기능 안전성 관련 표준이 존재하지 않는 조선분야를 예를 들어 기술하고 있다.

제 3 장. 안전성 관리

1. IEC 61508 의 안전성 관리

안전성 관리를 위한 국제 규격 IEC61508 은 수명주기 단계별 요구사항을 제시하고 있다. 수명주기 별 요구사항은 규격을 적용하는 시스템의 환경 및 범위 별 차별화되지 않고 모두 적용할 수 있도록 보편화되어 있으므로 요구사항의 만족을 입증하기 위한 문서화 및 입증자료 작성에 대한 구체적인 요구사항을 요구하지 않는다. 국제 표준을 적용한 국내외 인증기관들이 제공하는 입증자료인 위험분석 기법이나 위험원 목록의 양식이 다양한 이유도 이러한 요구사항의 보편성 때문이다.

안전성 관리는 안전관련 시스템 또는 소프트웨어 안전수명주기의 하나 이상의 단계에 대해 책임이 있는 사람들이 수행해야 할 기능안전성 관리 책임을 규정하는 것으로, 안전관련 활동 또는 안전수명주기에 대한 책임자가 해야 할 일은 다음과 같다.

- 시스템 및 수명주기 단계별 안전관련 활동 수행자의 역할 할당/조정
- 이들 단계와 타 조직/시스템 간 인터페이스
- 기능안전성 달성 정책, 전략, 평가 수단, 소통 수단
- 감지되는 모든 위험한 사건 분석, 권고사항의 반복 발생 가능성 최소화
- 기능안전성 평가 실행/조정(의사소통, 계획 및 문서화, 판단, 권고 등)
- 기능안전성이 이 표준의 목적 및 요구사항에 따라 달성 및 입증되고 있는지를 확인

또한 다음으로부터 발생하는 것들을 포함하여 안전관련 시스템에 관하여 즉시 대응 및 권고사항의 만족스러운 해결을 확보할 수 있는 절차를 개발해야 한다.

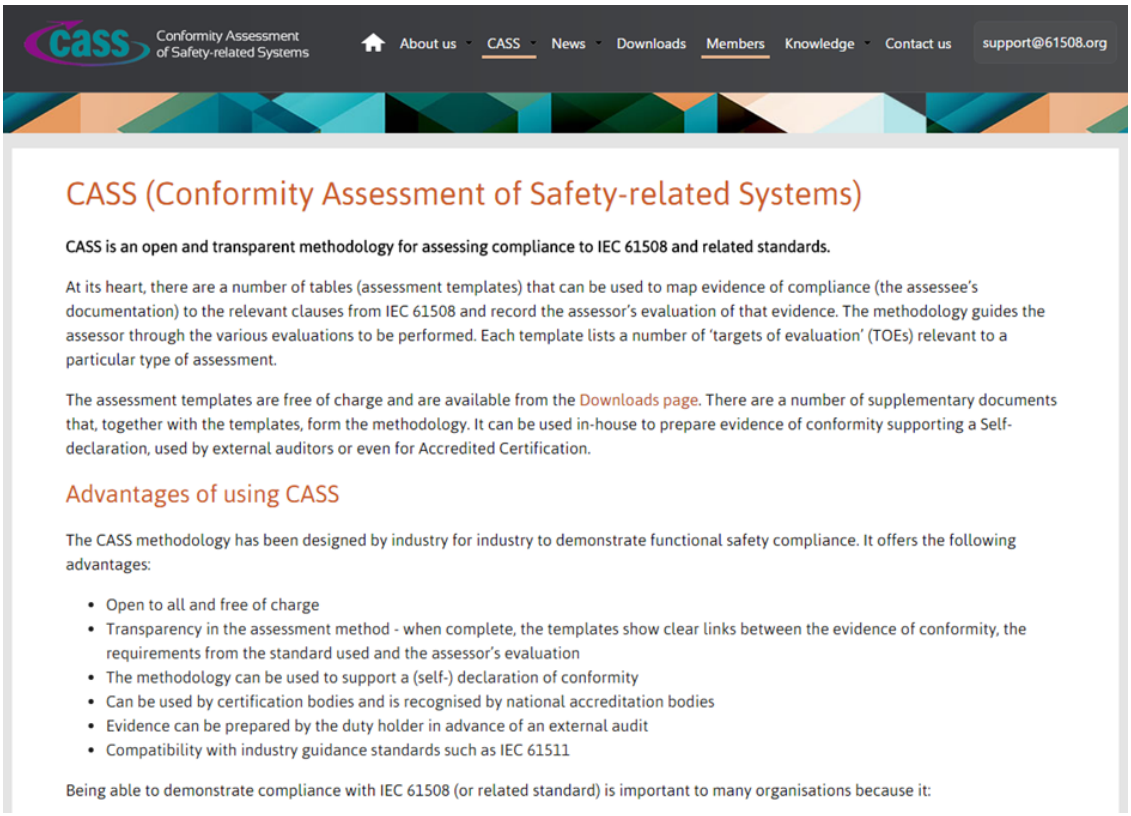
- 위험원 및 위험성 분석, 기능안전성 평가

- 각 수명주기의 단계별 산출물 검증 활동
- 전체 안전 검증 계획 및 검증 활동
- 위험관련 사건의 보고 및 분석
- 기능안전성 심사 빈도, 심사자들의 독립성 수준, 필요한 문서화와 후속 활동
- 위험원과 위험 사건, 안전 기능 및 안전관련 시스템에 대해 정확한 정보를 유지하기 위한 절차를 개발

안전성 관리를 위해 사고의 원인이 되는 위험원(Hazard)의 크기를 상대적으로 정량화한 위험성(Risk)가 허용할 수 있는 수준으로 제어된 상태를 의미하므로 안전성 관리를 위해서는 반드시 사고에 대한 정의가 먼저 수행되어야 한다. 이를 위해 대상 시스템의 정의와 안전성 관리 범위를 선정한다. 이어 위험원 식별, 위험성 계산, 위험성 결정을 수행하고 해당 허용 또는 수용 가능 여부에 따라 적정한 안전 수준을 확보하기 위한 안전기능을 추가하는 과정을 되풀이하여 가능한 안전 기능을 할당하고 이를 요구사항에 반영하여 개발 수명주기에 따라 관리한다.

영국과 같이 안전분야의 선진국에서는 안전관련 시스템의 도입, 수정, 유지보수의 수명주기 단계별로 안전 활동을 수행하고 그 결과물을 국가가 지정한 기관에서 심사하도록 정부차원에서 강제하고 있다. 이러한 공인 기관의 심사를 위해서는 안전성 관련 요구사항 준수여부에 대한 객관적이고 효율적인 평가를 위해 공통의 양식이 필요하게 되었으며, 이러한 요구에 의한 안전성 관리 및 문서화 지침은 CASS(Conformity Assessment of Safety-related Systems)에 공유되어 많은 사업의 안전성 관리 문서화에 활용되고 있다. 그림 Ⅱ.3.1 은 CASS 자료공개 웹사이트를 보여준다.

그림 II.3.1 CASS 자료 공개 사이트(http://www.61508.org)

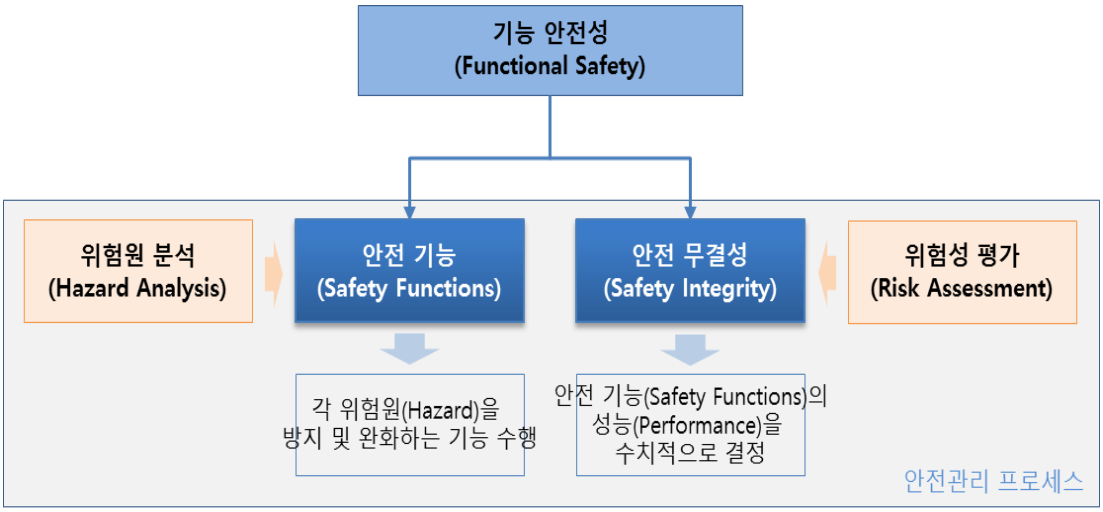


최종 사용자가 제시한 안전 요구사항의 만족을 입증하기 위해서는 수많은 입증자료가 요구된다. 이러한 입증 자료의 작성과정에서는 준비된 입증 자료의 최종 사용자 인정 여부가 실무에서 가장 어려움으로 나타나고 있다. 이러한 어려움은 초기 안전계획서의 완성도에 반비례하므로 객관적이고 실용적인 안전 계획의 수립을 위해서는 본 가이드에서 제공한 안전 수명주기와 기법, 각종 참조 산출물을 활용하여 개발 과정의 혼란을 최소화 할 수 있도록 한다.

2. 안전 기능과 안전 무결성

안전기능 요구사항은 위험원 분석을 통해 도출되고, 안전무결성 요구사항은 위험성 평가를 통해 도출된다. 안전 무결성(Safety Integrity)은 주어진 모든 조건하에 있는 안전관련 시스템이 주어진 시간 내에 요구되는 안전기능을 만족스럽게 수행할 수 있는 확률로 정의되고 안전무결성수준이 높을수록 해당 장비 또는 시스템의 고장 발생 가능성은 낮아진다. 그림 II.3.2 는 안전 기능과 안전무결성의 관계를 보여준다.

그림 II.3.2 안전 기능과 안전무결성



안전무결성수준(SIL)은 전기전자프로그래머블제어기의 국제규격인 IEC61508 에 명시되어 있다. 안전무결성수준이 널리 사용되는 이유는 적용 대상이 명확하게 정의되지 않은 경우에 시스템의 안전성을 평가하여 시스템의 안전성을 상호 비교할 수 있는 방법이기 때문이다.

예를 들어 전자 연동 장치의 종합 안전성은 사고를 열차 충돌과 탈선으로 정의하는 경우에 전자연동장치의 출력에 의해 제어되는 선로 전환기나 신호기와 같은

장치와의 인터페이스 구조 및 전자연동장치 운영시나리오에 따라 결정된다. 따라서 연동장치가 설치될 역의 규모, 인터페이스 장치의 안전수준, 운용인력의 안전문화 등이 고려되지 않으면 열차충돌 및 탈선에 대한 안전성 확보를 평가할 수 없다. 이러한 경우 전자연동장치의 안전성을 평가하기 위해서 전자연동장치의 위험측 고장(Dangerous Failure)의 발생빈도를 평가하여 등급을 부여한 것이 안전무결성수준이다.

위험측 고장률은 위험원이라는 특별한 요건에 대한 발생빈도를 예측하고 입증하여 평가한 정량적 수치이다. 정의된 사고와 관련된 위험원을 도출하고 분석하는 단계는 위험원 관리와 같이 수행되며 위험원들의 발생빈도는 FTA(Fault Tree Analysis)와 ETA(Event Tree Analysis) 등의 기법에 의해 정량화되어 목표와 비교된다.

최종 사용자의 시스템 안전성 요구사항은 정량적 수치로 주어지기도 하지만 정성적이고 모호하게 제시되기도 한다. 이러한 경우 안전이 확보되어야 할 사고의 정의부터 위험성의 제어를 위해 필요한 안전 대책의 적용 범위도 명확히 분석되어야 한다. 시스템 개발 과정에서 공급범위를 벗어나는 안전 대책에 의해서만 위험원이 안전하게 제어되는 경우를 예측하여야 한다.

안전 요구사항의 할당은 시스템 안전성과 종합 안전성에 따라 상이하다. 시스템적 안전성만을 관리하는 경우에는 안전무결성수준과 같이 시스템에 요구되는 위험측 고장률의 목표 만족을 위한 하부 구성요소들의 위험측 고장률 목표를 설계단계에서 위험원 도출 및 분석을 통해 산출하여 배분해야 한다. 종합적 안전성의 경우 시스템적 안전성을 포함하여 설치, 테스트, 운용, 유지보수와 같이 인적요소가 가입되는 사항에 대한 안전 요구사항을 도출하여 해당 수명주기에 할당해야 한다.

3. 수명주기별 안전성 관리

가. 수명주기 개요

안전관련 시스템에 의해 수행되는 안전기능에 대해 필요한 안전무결성 수준을 달성하기 위한 모든 활동을 체계적인 방법으로 다루기 위하여, IEC 61508 에서는 기술 프레임워크로서 전체 안전수명주기를 제시하고자 한다. 전체 안전수명주기는 이 표준 준수를 위한 객관적 참조물로 제시되고, 각 단계의 목적과 요구사항에 충족된다면, 다른 SW 개발 안전수명주기를 이용할 수 있다.

소프트웨어 개발을 위한 안전수명주기는 IEC 61508 에 따라 전체 시스템 개발계획에 안전계획이 결정되고 명시되어야 한다. 표준에서 요구하는 사항을 만족시키는 안전수명주기 모델은 본 가이드에서 참조 가능하도록 상세하게 제시하였으며, 프로젝트나 조직의 특정한 구체적 요구에 맞추어 적절하게 조정하여 사용할 수 있다. 다음은 IEC 61508 에서 제시하는 수명주기 관련 요구사항은 다음과 같다.

- 품질과 안전 보증 절차는 안전수명주기의 각 활동들과 통합되어야 한다.
- 소프트웨어 안전수명주기의 각 단계는 적용범위를 갖는 기초 활동과 각 단계를 위해 명시된 입력과 그 결과물인 산출물로 구분되어야 한다.
- SW 수명주기 단계에 관한 자세한 정보는 ISO/IEC 12207 을 참조한다.
- IEC 61508 - 1 의 안전수명주기 단계별 산출물들이 포함되어야 한다.
- 산출물은 E/E/PE 안전관련 시스템의 개발마다 때로는 몇 개가 병합되기도 하고 세분화되어 나누어진 별개의 문서로 제공될 수도 있다.
- 핵심적인 요구사항은 모든 안전수명주기 단계들의 산출물은 그 의도된 목적에 적합해야 한다는 것이다. 간단한 개발에서는 일부 안전수명주기 단계들이 병합될 수 있다.
- 소프트웨어 안전수명주기가 IEC61508 에서 요구하는 사항을 만족시킨다면, V 모델 단계들의 정도, 수, 작업규모를 안전무결성 및 프로젝트의 복잡성을 고려하여 일부 절차 및 활동은 조정할 수 있다.

- 표준에서 제시하는 수명주기 단계들의 전체 리스트는 대체로 규모가 크고 새로 개발된 시스템에 적합하다. 예를 들면 소규모 시스템에서는 소프트웨어 시스템 설계 및 아키텍처 설계 단계를 병합하는 것이 적합할 수도 있다.
- 표준의 모든 목적과 요구사항을 만족한다면, 다르게 소프트웨어 프로젝트를 관리할 수 있다(즉, 다른 소프트웨어 안전수명주기 모델 이용 가능).
- 각 안전수명주기 단계에서 SW 안전성 보증을 위해 적절한 기법과 수단이 이용되어야 한다.
- 소프트웨어 안전수명주기에서의 활동 결과들은 문서화되어야 한다.

나. 수명주기 개발 참조 관련 규격 및 문헌

- IEC 61508-1:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements
- IEC 61508-2:2010, Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
- IEC 61508-3:2010, Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 3: Software requirements
- IEC 61508-4:2010 Functional safety of electrical/electronic/programmable electronic safetyrelated systems – Part 4: Definitions and abbreviations
- IEC 61508-5:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity levels
- IEC 61508-6:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
- IEC 61508-7:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures

- KS C IEC61508 - 1:2010 전기/전자/프로그램 가능 전자식 안전관련 시스템의 기능 안전성 - 제 1 부: 일반 요구사항
- KS C IEC61508 - 3:2010 전기/전자/프로그램 가능한 전자장치 안전관련 시스템의 기능안전성 - 제 3 부 : 소프트웨어 요구사항
- 철도 소프트웨어 안전 기준 및 체계 구축 연구보고서, 2008
- NUREG CR-6430 UCRL-ID-122514 SW Safety Hazard Analysis, NRC, USA
- NUREG IA-0145 RELAP5 Assessment Against - Revision 1, NRC, USA
- IEEE730 A guide to writing successful SQA plans
- IEEE829:2008 SW test documentation
- IEEE1016:2009 Software design description
- IEEE1012:2004 Standard for Software Verification and Validation
- IEEE1228:1994 IEEE Standard for Software Safety Plans
- IEEE1540:2001 Software Engineering Risk Management: Measurement-Based Life Cycle Risk Management
- ISO12207:2004 Systems and software engineering - Software life cycle processes
- SIL4 인증문서 한글 표준양식(템플릿) 적용사례 연구, 한국철도학회
- 소프트웨어 개발 프로세스에서의 안전성 분석 및 관리 활동의 적용방안, 중소기업융합학회
- 기능안전 적용을 위한 소프트웨어 개발 가이드라인, KTL

다. 안전 수명주기 구성

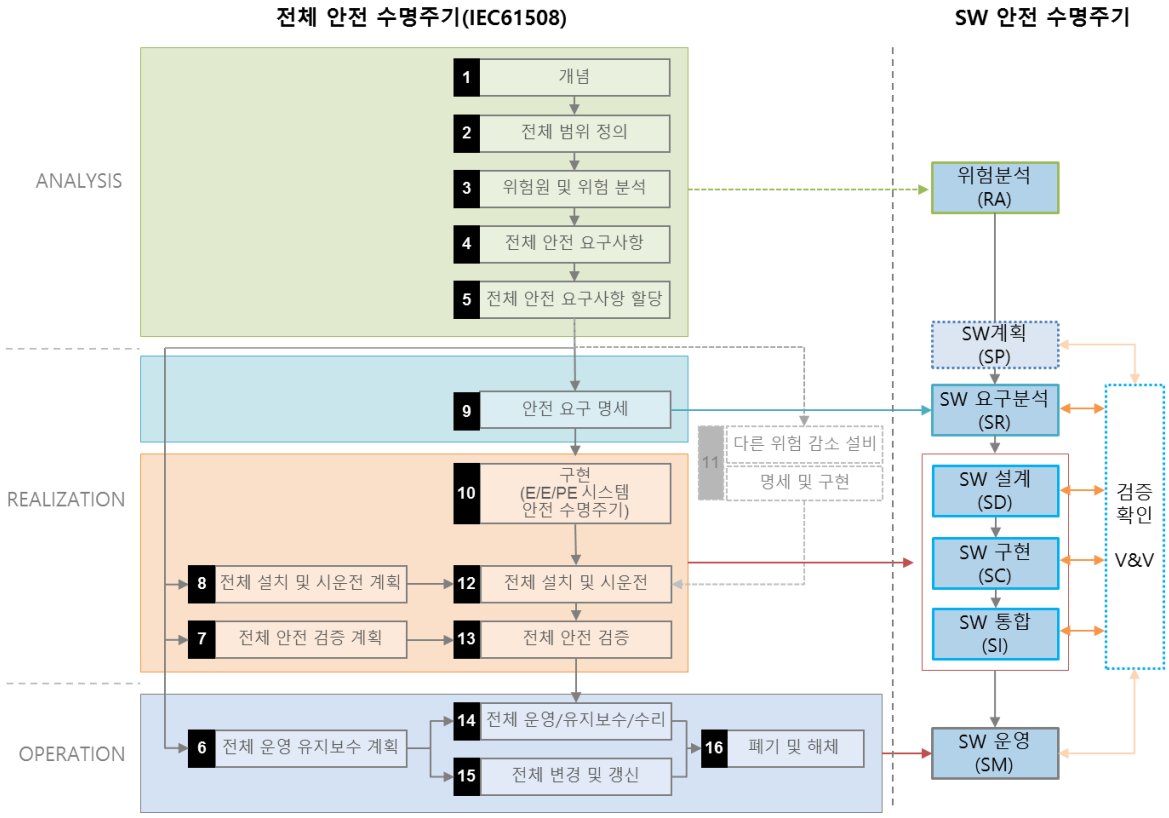
기능 안전 표준인 IEC61508 과 SW 수명주기 표준인 ISO12207 을 참고하여 안전 수명 주기를 구성하였다. SW 안전 수명주기는 크게 세 부문으로 구성되고 각 부문에는 아래와 같이 세분화된 단계로 구성될 수 있다.

- Analysis 부문
  - 위험분석 단계
- Realization 부문

  - SW 계획 단계
  - SW 요구분석
  - SW 설계 단계
  - SW 구현 단계
  - SW 통합 단계
  - SW 확인 검증 단계
- Operation 부문
  - SW 운영 단계

Realization 부분의 SW 검증 및 확인 단계는 Realization 부문의 단계 시작 전에 계획되어 전 단계에 걸쳐 활동이 진행된다. 본 가이드에서는 계획은 위험분석 단계에 수행하는 것으로 배치하였고, 일반적인 프로젝트에서는 프로젝트 착수 단계에서 사전에 품질관리 계획 수립과 함께 고려되어야 한다.

그림 II.3.3 전체 안전 수명주기(IEC61508)



본 가이드에서는 전체 안전 수명주기를 세 부문으로 나누어 설명한다. II. SW 안전성 개요에서는 안전성 개념, 표준, 안전성 관리 절차를 설명하고, III. 위험분석 부문에서는 위험분석 절차, 세부 내용, 사례, 기법 등을 제시하고, IV. SW 개발 부문에서 SW 요구분석, SW 설계, SW 개발, SW 통합, 검증 및 확인의 단계에서 해야할 활동, 기법, 산출에에 대해서 상세히 제시하고, V. SW 운영 단계에서는 운영, 유지보수, 폐기 등의 활동에 대한 설명을 제시하고 있다. 그림 II.3.3 은 IEC61508 의 안전수명주기와 본 가이드의 SW 안전수명주기의 관계도를 보여준다.

III. 위험분석(Analysis)

제 1 장. 위험 분석 개요

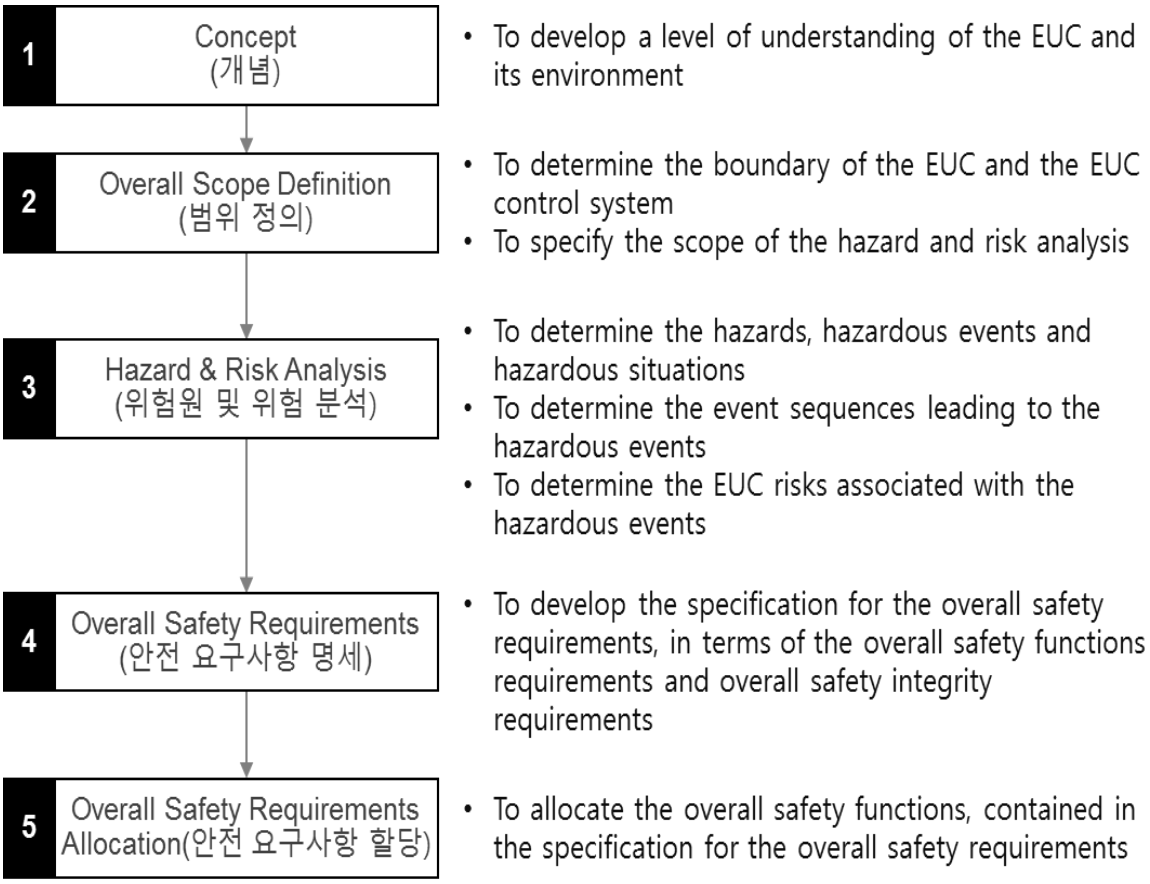
1. 위험 분석 절차

위험성은 위험원의 발생빈도와 심각도의 조합으로서 발생빈도가 빈번하거나 발생빈도가 빈번하지 않더라도 사고가 발생하면 치명적인 결과를 초래하는 위험원은 위험성이 크다고 정의한다. 사고관련 위험원의 위험성을 허용할 수 있는 수준으로 제어하고자 하는 안전성 관리는 시스템 수명주기의 요구사항 분석 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.

안전성 확보를 위한 위험분석 절차는 대상 제어시스템에서 위험원을 중심으로 결과, 손실, 위험성을 파악하는 위험분석 단계와 해당 위험에 대한 원인 분석을 통해 위험 제거 및 감소를 위한 안전 기능을 도출하여 SIL 을 할당하고 이를 시스템 하위 서브시스템, HW, SW 에 할당하는 단계로 정의할 수 있다.

IEC61508 에서 제시하는 전체 안전 수명주기에서 위험분석 단계로 그림 Ⅲ.1.1 과 같은 개념정의(Concept), 범위 정의(Overall Scope Definition), 위험원 및 위험 분석(Hazard & Risk Analysis), 안전 요구사항 명세(Overall Safety Requirements), 안전 요구사항 할당(Overall Safety Requirements Allocation) 으로 구성 된다.

그림 III.1.1 IEC61508 기준 위험분석 절차



위험성은 위험원의 발생빈도와 심각도의 조합으로서 발생빈도가 빈번하거나 발생빈도가 빈번하지 않더라도 발생 시 치명적인 결과를 초래하는 위험원은 위험성이 크다. 사고관련 위험원의 위험성을 허용할 수 있는 수준으로 제어하고자 하는 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. 표 III.1.1 은 안전성분석 절차를 보여준다.

표 III.1.1 위험 분석 절차 상세 설명

구분	수행할 안전 요구사항

1. 개념		EUC(Equipment Under Control) 이해 필요한 제어 기능 및 물리적 환경 정의 유해 사건 원인 결정, 위험원 정보, 현재 안전규정(법/제도) 등 파악
2. 범위 정의		EUC 와 제어시스템의 경계 결정 위험원과 리스크 분석 적용 범위(프로세스, 환경) 결정 고려할 외부 사건, 연관된 장비/시스템, 고려할 사건 유발 유형 결정
3. 위험원 및 위험 분석	위험원 식별	위험원, 위해 사건 식별 (위험원 제거/감소 도 고려) 위험한 상황(결함, 예상 가능한 오용, 악의, 비 허가) 결정
	위험성 계산	결정된 위해 사건으로 이어지는 사건 순서를 결정 명시된 상황에서 위해 사건 발생 가능성 평가 결정된 위해 사건과 연관된 잠재 결과 확인(심각도) - 정량적 또는 정성적인 위험원 및 리스크 분석 기법 적용 위험한 사건과 관련된 위험 결정(평가/추정)
4. 안전 요구사항 명세		기능 안전성 달성(위험 제거/감소) 전체 요구사항 명세 개발 안전 기능(Safety Function) 요구사항 명세 안전 무결성(SIL, safety Integrity Level) 요구사항 명세 - 목표 안전 무결성 요구사항 – 허용 리스크에 부합 - 필요한 리스크 감소 : 허용 가능한 리스크 달성 - 허용 가능한 사건 : 허용 가능한 리스크 충족
5. 안전 요구사항 할당		안전 기능 → 안전관련 시스템, 위험 감소 수단에 할당 - 안전 기능이 허용 가능한 리스크 달성하도록 할당 지속 - 달성이 어려우면 명세를 변경하면서 지속적으로 할당 반복 - 전체 안전 기능이 할당되고, 목표 고장 기준이 안전 기능에 할당 안전 무결성 수준(SIL) → 각 안전 기능에 할당 - 확률을 조합하여 적절한 기법 이용, 공통 원인 고장 가능성 고려 - 목표 고장 기준 : 저요구/고요구/연속적 작동 모드

2. 위험 분석에 대한 접근

가. 시스템 안전 분석의 일부로서 SW 위험 분석

SW 는 시스템의 일부분으로 위험 분석은 전체 시스템의 설계 맥락에서 수행되어야 한다. SW 그 자체만으로는 안전성 문제를 발생시키지 않고, SW 가 시스템의 일부분으로 구성될 때 비로소 문제가 발생한다. 따라서 SW 안전성은 HW 와 결합되는 지점부터 위험분석이 시작되어야한다. 또한 관련된 HW, 주변 환경, 그리고 사람 등을 고려해야 한다. 때문에 SW 위험 분석을 수행하는 분석가는 시스템 안전 기능의 수행과 시스템 제어 및 감시 기능의 수행에 있어서 SW 역할을 이해해야 하고, 해당 시스템 안전성 달성에 있어서 SW 가 시스템에 미치는 영향을 잘 파악해야 한다.

시스템 설계에 있어 안전 특성들은 기본적으로 품질, 다양성, 그리고 심층방어의 세 가지 설계 원칙으로 결정된다. 이 원칙들이 여러 계층의 설계에 적용될 수 있으며, 어디에 어떻게 적용할 것인지를 결정하는 것이 설계 공정의 중요한 활동이고, 세 가지 원칙은 다양한 형태의 공정제어 장치에 적절하게 적용 가능하다.

물리적인 관점에서 심층방어 개념을 예를 들어 살펴보면, 핵분열 생성물의 누출을 통제 하기 위해 세 계층의 심층방어 설계가 마련되어 있다. 각 계층은 어느 정도 심각한 수준의 방사선으로부터 일반 대중의 피폭을 방지할 수 있어야 한다. 첫 번째 계층은 핵연료봉이 그 피복재로 싸여져 있다. 두 번 째 계층은 핵연료봉에서 세어나온 핵분열 생성물이 E/E/PES 시스템냉각재장치(RCS)에 의해 격리된다. 세 번째 계층은 E/E/PES 시스템 건물이 E/E/PES 시스템 냉각재 장치를 에워싸고 있다. 이들 각 계층은 기본적으로 서로 다른 설계이며, 각 계층마다 다양성을 부여하고 있다.

계측제어 관점에서 살펴본 심층방어 개념으로는 각 기능이 여러 개의 독립적인 시스템들에 의하여 작동될 수 있다. 예를 들면, 제어봉은 제어 장치, E/E/PES 시스템보호장치, 원자로정지불능과도사건(ATWS) 완화 장치, 공학적안전설비 작동

장치(ESFAS)에 의해 자동 또는 수동으로 작동될 수 있다. 또한 컴퓨터 기반 제어 및 보호장치들을 포함한 신형 E/E/PES 시스템 설계에서는 적어도 두 개의 자동시스템이 각 안전 기능을 개시할 수 있어야 한다. 그리고 운전원이 각 안전 기능을 시작, 가동, 종료할 수 있도록 충분한 정보와 수동 제어기능을 마련하여야 한다.

나. SW 설계의 일부로서 SW 위험 분석

위험 분석의 궁극적인 목표는 부적합사항을 찾아서 시정하고, 필요한 안전조치를 위한 정보를 제공하는데 있다. SW 위험원 분석에서도 적절한 조치가 취해지지 않는다면, 분석의 의미가 없어지게 되므로, 적어도 다음 유형의 조치들이 상황에 맞게 적절하게 취해져야 한다.

- 시스템 설계는 SW 에 의해 영향을 받거나, SW 로 적절하게 처리되지 않는 확인된 위험원들을 제거하려는 목적으로, 그 위험원을 허용 가능한 수준까지 줄이거나 확인된 위험원들이 심층방어 설계로써 제거될 수 있도록 시스템 구조를 변경할 수 있다.
- SW 설계는 확인된 위험원들을 없애거나, 그것들을 허용 가능한 수준까지 줄이려는 목적으로 변경될 수 있다.
- SW 품질은 허용 가능한 수준까지 특정 위험원의 발생 가능성을 줄여서 충분한 정도까지 나아질 수 있다.
- 응용 시스템은 만약 그 시스템이 너무 위험하다면 폐기될 수도 있다.

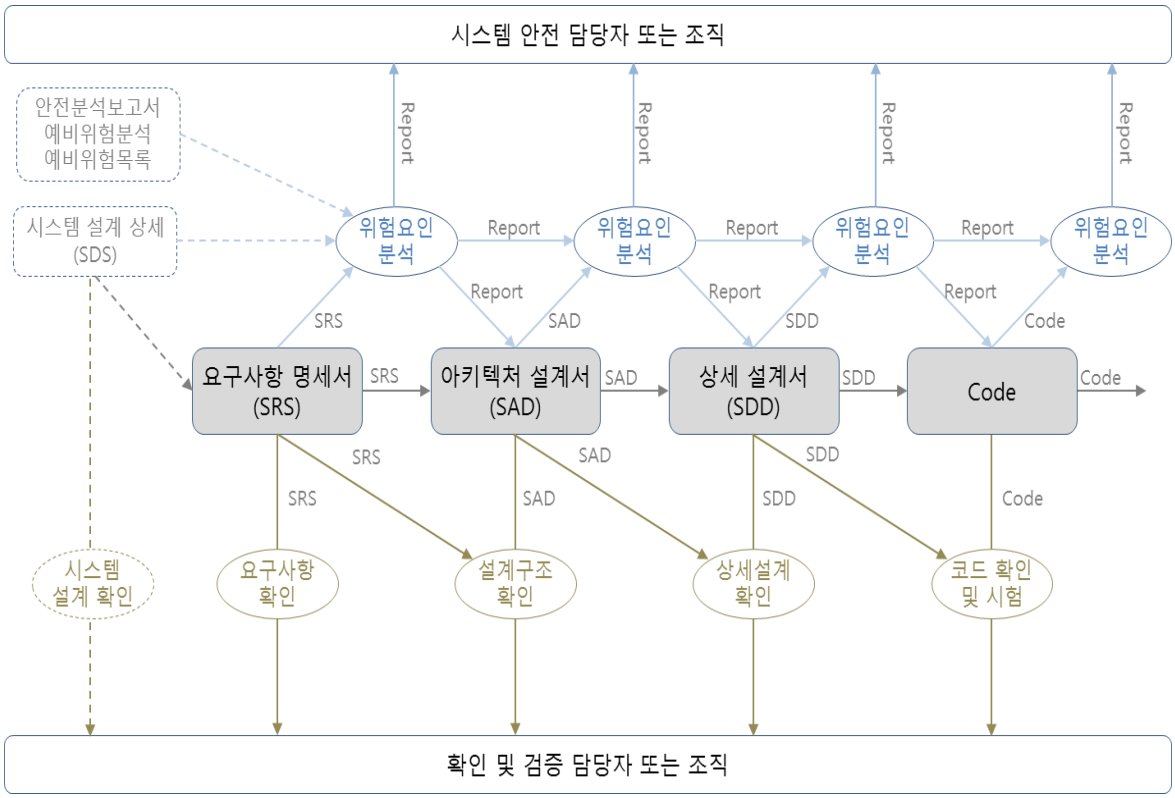
다. SW 위험 분석에 관한 일반적인 접근방식

SW 위험 분석 활동은 SW 수명주기에 걸쳐 잘 정의되어야 한다. 일반적으로 위험요인 분석은 시스템의 설계 분석 및 안전분석관련 정보를 제공 받으면서 시작되며, 그 설계 분석은 안전한 작동 영역의 허용치(안전무결성수준)를 결정하게 된다. 이 설계분석에서 SW 위험원 분석을 위한 출발점이 될 많은 다양한 정보들을

제공하게 된다. 여기에는 기술적 개발활동(요건, 구조, 설계, 코드), 확인 및 검증 활동, 위험원 분석 활동 등이 포함된다.

각 선행단계에서는 그림 Ⅲ.1.2 와 같은 하나 이상의 문서가 작성되어야 한다. 이 문서들이 다음 단계의 위험요인 분석을 수행하는데 필요하게 되며, 단계별 검증 및 확인의 대상이 된다.

그림 Ⅲ.1.2 SW 수명주기에 따른 SW 위험원 분석공정



(참조: JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING HANDBOOK, DOD)

위에서 제시한 절차가 일반적으로 요구되지만, 실무에서는 사업에 따라 부분적으로 커스터마이징이 필요하다. SW 는 개발이 진행되면서 반복적인 위험 분석 활동이 필요하므로 절차를 반드시 준수해야 하는 것은 아니다. 예를 들면 SW 요구사항에 대한 위험 분석이 이루어지기 전에 예비 위험원 분석이 필요한데, 그러한 분석 또는 다른 형태의 요구사항 분석 결과가 시스템 설계변경을 초래하여 예비위험원분석을 반복 수행해야 할 수도 있다.

제 2 장. 위험 분석 절차 상세

1. 단계 : 개념

이 단계에서는 개발하거나 운영해야 할 대상 제어 시스템(EUC)과 그 환경(물리적, 법적 등)을 충분히 이해하여 이후 위험 분석 등 안전수명주기 활동이 만족스럽게 실시될 수 있도록 하는 것이다. 이를 위해 다음의 내용에 대한 이해 및 분석을 위한 기초정보 확보가 필요하다.

- 대상 제어 시스템(EUC)에 필요한 아래와 같은 제어 기능 및 물리적 환경에 대해 완벽히 이해
- 위험원, 위험한 상황 및 유해한 사건을 발생시킬 수 있는 원인을 결정
- 결정된 위험원에 대한 정보(예: 기간, 강도, 독성, 노출한도, 기계적인 힘, 폭발 조건, 반응성, 인화성 등)
- 안전규정관련 국내 및 해외 정보
- 다른 장비 또는 시스템과의 연동으로 인한 위험원, 위험한 상황 및 유해한 사건을 다른 EUC 와 함께 고려

2. 단계 : 범위 정의

범위 정의에서는 대상 제어시스템(EUC)과 이를 제어하는 시스템의 경계를 결정하고, 위험원 및 위험 분석의 적용 범위를 명시하는 것이다 (예: 프로세스 위험원, 환경적 위험원 등). 이를 위해 다음과 같은 정보 및 활동이 필요하다. 그림 Ⅲ.2.1 는 범위 정의의 예시이다.

- EUC 와 EUC 제어 시스템의 경계는 관련 위험원 및 유험한 사건과 연관된 모든 장비 및 시스템(해당되는 경우 인간을 포함)을 포함

- ECU 및 ECU 제어 시스템을 포함하여, 위험원 및 리스크 분석의 범위에 포함될 물리적 장비
- 위험원 및 리스크 분석에서 고려해야 할 외부적 사건
- 위험원 및 위험한 사건과 연관된 장비 및 시스템
- 고려가 필요한 사건 유발 유형(예: 위험한 사건을 유발할 수 있는 부품 고장, 절차상의 결함, 인적 오류, 종속적 고장 메커니즘 등)
- 얻은 결과와 정보를 문서화

그림 III.2.1 범위 정의 예시

- One Series Safety Transmitter 는 시스템 프로세스의 온도 또는 압력을 감지하고 안전하지 않은 상태가 발생하기 전에 해당 시스템을 모니터링하거나 종료하기위한 출력을 제공하는 2 선 송신기입니다.
- 4-20mA 출력은 안전 PLC 에서 사용하기위한 프로세스의 아날로그 표시를 제공하며, 솔리드 스테이트 세이프티 릴레이 출력은 프로그래밍 된 작동 모드 및 제한을 기반으로 최종 요소를 직접 제어하거나 셧다운하고, 스위치 상태 출력은 솔리드 스테이트 릴레이 출력의 기능 및 상태를 반영하는 개별 출력입니다.
- 현재 작동 중(IAW) 출력은 자체 진단을 기반으로 한 개별 출력이며 송신기 상태를 나타내고, 장애가 발생하면 결과를 오류 안전 상태로 전환합니다. One Series 안전 트랜스미터의 4 개 결과는 모두 안전에 중요한 결과물로 이용할 수 있으며 Trip-to-Trip (DTT) 모드로 작동합니다.
- One Series 안전 트랜스미터는 하드웨어 결함 허용 오차가 0 인 IEC 61508 에 따라 유형 B1 장치로 분류됩니다.

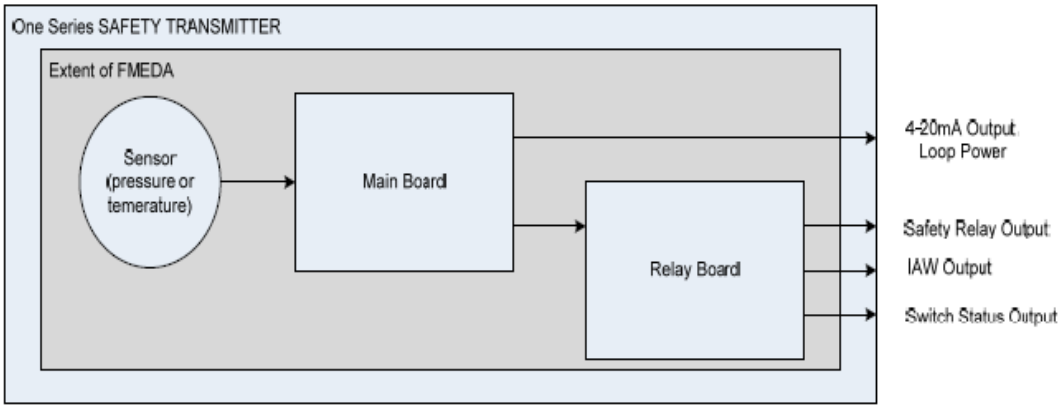


Figure 1: One Series Safety Transmitter

(참조: IEC61508 Function Safety Assessment Report, EXIDA : IEC61508 에는 구체적인 예제가 제공되지 않아 타 인증기관(Exida)의 결과를 예시로 사용함.)

3. 단계 : 위험원 및 위험 분석

이 단계에서는 결함 상황 및 예상 가능한 오용을 포함하여 모든 합리적으로 예측 가능한 상황에 대해, EUC 와 EUC 제어 시스템과 관련된 위험원과 위험한 사건 및 상황을 결정하고, 이에 이르게 하는 사건 순서를 결정하며, 결정된 위험한 사건과 연관된 EUC 리스크를 결정한다.

운영자들이나 제작사들은 과거의 경험으로 축적된 사고 리스트나 위험한 사건 발생 리스트를 이용하여 시스템 위험원 분석을 한다. 분석의 목적은 시스템 개념정의를 통하여 위험상황을 막기 위하여 시스템에 도입되어야 할 수단과 시스템에 내재한 위험상황에 대하여 아이디어를 모으기 위한 것으로 잠재적으로 위험한 사건 사고를 이끌 수 있는 요인을 정의하는 것이 목적이다.

위험원 분석은 기술적으로 광범위하고, 중립적으로 다루어져야 하고 시스템 레벨의 위험원에 대해서는 Top-Down 방식으로 분석한다. 예비 위험원 분석 (PHA : Preliminary Hazard Analysis)이 요구된다.

예비 위험원 분석(PHA, Primary Hazard Analysis)은 개발 범위에 포함해야 할 위험원의 선정을 목적으로 개념설계 단계에서 수행된다. 개발되는 시스템 응용분야 전문가, 소프트웨어 응용분야 전문가 등이 함께 참여하여 수행한다.

예비 위험원 분석의 결과인 예비 위험원 목록은 인적요소가 포함되어 개발된 시스템 관련 위험원으로서 도출된 위험원의 위험성을 평가하여 안전성 확보의 유무를 판단하게 된다. 따라서 예비 위험원 분석에서 고려되지 않는 위험원은 시스템 안전성에 고려되지 않게 되므로 위험원의 누락을 방지하기 위한 건전성의 확보가 필요하다. 이러한 건전성의 확보를 위한 다양한 방법 중 기존 보유한 위험원 목록을 대상으로 하는 경우와 검증된 체크리스트를 이용하는 방법이 대표적이다.

산업 별로 사고의 발생 기록을 토대로 위험원을 제시하거나 새로운 개념의 시스템이 개발되고 복잡도가 증가함에 따라 응용환경에 적합한 예비위험원분석을 위한 체크리스트를 제공하기도 한다. 그림 Ⅲ.2.2 는 체크리스트의 예시이다.

예비 위험원 분석은 시스템 수명주기의 초기 단계에서 수행되므로 구체적인 시스템 고장률 등은 고려하지 않는다. 따라서 예비 위험원 분석에서 도출된 위험원의 위험성 평가는 예비 위험원 분석에 참여하는 전문 인력의 경험에 의존하게 되며, 정량적인 데이터를 근거로 한 위험성의 평가는 수명주기 중 설계 이후 시행되는 위험원 도출 및 분석을 통해 가능해 진다.

그림 III.2.2 예비 위험원 분석 체크리스트 예시

Hazard	✓	Hazard	✓
Mechanical/Kinetic		**Chemicals/Substances**	
Contact with moving plant/parts (cutting, shearing, entanglement, etc)		Inhalation of dusts, gases, fumes, vapours and mists	
Contact with sharp objects/edges		Ingestion of chemicals/substances	
Contact with moving vehicles/mobile plant		Absorption of chemicals/substances through skin	
Struck by projectiles or ejected items (including struck in eye by object)		**Radiation**	
Mechanical damage to services, PPE or other items		Exposure to ionising radiation source (industrial radiography, non-destructive testing)	
Gravitational			
Fall from height		Exposure to non-ionising radiation source (laser, welding flash, infrared, radiofrequency)	
Falling object from height			
Slips and trips		**Biological**	
Access/work beneath a suspended load/unstable object		Exposure to algal, bacterial, fungal, viral or parasitic agents (skin contact, ingestion, inhalation)	
Thermal and/or Explosive			
Fire/Explosion		Animal, insect and spider bites/stings Sharps/needle-stick exposure	
Ignition of gas/dust in a hazardous area			
Contact with hot/cold objects/parts		**Manual Handling/Postural**	
Excessively hot/cold environments (including heat stress)		Handling heavy, unstable or awkward objects/loads Repetitious movements	
Electrical			
Contact with live electrical parts (overhead power line, etc)		Maintaining static or awkward postures	
Exposure to high fault currents (within switchboards, battery banks)		Tool use that requires excessive force	
Mechanical damage to power leads/fixed electrical wiring		**Psychological/Mental, Social, Medical**	
Ingress of water into electrical components		Working for excessive time periods and/or while fatigued	
Noise and/or Vibration		Exposure to workplace bullying, harassment, violence	
Exposure to increased noise (levels that may cause hearing damage)		**Work Environment**	
Contact with vibrating plant/vehicles/tools/objects		Inadequate lighting	
Environmental		Wet/slippery/uneven/unstable work surface	
Air/ground/water contamination (including spills, uncontrolled release, etc)		Weather conditions (including flooding, lightning, wind)	
Release of harmful solid, liquid or gas during transport on/off site		Working alone	
Incorrect waste disposal		Unfavourable atmospheric conditions (dusty)	
Import of unauthorised soils, materials, plants or machinery		Restricted access or working space	
Pressurised		**Other:**	
Release of stored gas, liquid, solid under pressure		Aviation:	
Release of spring/tension energy			

(참조 : OCCUPATIONAL HEALTH & SAFETY TEMPLATE, Ohs)

예비 위험원 분석에서 도출된 대표 위험원의 위험성을 정량적으로 평가하기 위해 구체적인 시스템 사양을 바탕으로 위험성을 평가하기 위한 절차가 위험원 도출 및 분석이다. 위험원 도출 및 분석은 인적 요소를 포함한 전체 위험원의 위험성 평가를 위해 분석의 범위를 시스템, 인터페이스, 운영시나리오로 나누어 수행한다.

시스템 위험원 도출 및 분석은 순수하게 시스템의 고장 중 정의된 사고의 원이 되는 요인을 분석하며, 인터페이스 위험원 도출 및 분석은 기존 시스템의 인터페이스

부분을 대상으로 개발 시스템의 입력 및 최종 출력에서 나타난 고장 중 정의된 사고의 원인이 되는 요소를 분석한다. 마지막으로 운용 과정에서 발생될 수 있는 위험원을 도출하고 분석한다. 따라서 시스템 위험원 도출 및 분석의 결과는 인터페이스 위험원 도출 및 분석의 입력 데이터로 사용되며, 대부분이 HW 고장을 고려하게 된다. 운영시나리오 위험원 도출 및 분석은 시스템 SW 와 매우 밀접하게 관계를 갖는다.

위험원 누락의 최소화를 위해서는 체계적인 분석이 요구되며, 이러한 체계적 분석을 위한 대표적 방법론에는 FMEA 와 HAZOP Study 가 있다. FMEA 와 HAZOP Study 는 모두 고장발생 기준에 대한 결과와 전체 시스템에 미치는 영향을 분석하고 경우에 따라 위험성을 평가하는 방법론이다. 다만 사용되는 고장 발생 기준이 FMEA 의 경우 고장모드(Failure Mode)를 사용하고, HAZOP Study 의 경우 지시어(Guide Word)를 사용하는 것이 차이점이다. 고장모드는 경험에 의해 추정할 수 있는 고장의 상태를 정의하고, 각각의 고장모드를 기준으로 분석 대상의 고장 영향을 평가하는 방법이며, 지시어는 모든 제어 출력의 고장형태를 분류하여 분석하는 방법이다.

위험원목록(Hazard Log)은 안전성활동의 입증자료 문서화 단계로써 시스템, 인터페이스, 운영시나리오로 나누어 수행된 위험원 도출 및 분석의 결과들을 종합하는 단계이다. 위험원목록의 작성을 통해 시스템, 인터페이스, 운영시나리오별 위험원 중 공통된 사항의 중복을 처리하고 인적오류를 포함한 전체 시스템의 위험원별 위험성을 평가하여 안전 확보여부를 판단한다. 따라서 안전의 확보를 목적으로 위험원별 위험성완화를 위해 사용된 안전대책들이 설계의 경우 해당 도면, 교육/훈련의 경우 교육/훈련 매뉴얼 및 수료증, 운영규정의 경우 해당규정에 대한 문서번호를 첨부해야 한다.

시스템의 도입이 완료된 후에도 위험원목록은 지속적으로 관리되어야 한다. 해당 시스템이 개량되면 개량된 사항이 기존에 확보된 안전성에 영향을 미치므로 위험원목록에서 개량과 관련된 부분을 다시 분석하고 평가하여 안전성의 유지여부를 평가해야 한다. 또한 신규시스템의 도입 시에도 신규시스템과 인터페이스 되는 기존 시스템들의 위험원목록을 입수하여 신규시스템으로 인해 발생되는 안전관련 영향이

평가되고 변경사항이 발생하면 위험원목록의 해당부분이 갱신되어야 한다. 시스템별로 위험원목록이 구축되면 신규사업의 예비타당성 조사 시 기존 시스템의 안전에 미치는 영향을 용이하게 평가할 수 있다.

결과분석은 모든 가능한 시나리오 및 위험원과 관련된 위험을 평가하기 위하여 위험원으로 인한 결과를 규명하는 것이 목적이다. 일반적으로 정량적으로 수행되어야 하는데 이는 예비 위험원 분석과는 반대로 낮은 레벨의 위험원으로부터 높은 레벨의 위험원으로 분석을 수행한다. 즉 상향식 방식으로 생각할 수 있는 최악의 경우에 대하여 결과를 도출하여야 한다. 위험원으로 인한 사고의 결과는 심각도 항목으로 정량적으로 할당하여야 한다.

위험성은 위험원에 의해 잠재적으로 촉발되는 사고의 발생가능성과 심각도를 추정하여 평가한다. 일반적으로 결과 분석에는 위험원이 발생할 수 있는 상황 및 환경을 고려하기 위하여 시스템 및 운영 환경에 대한 전문적인 지식이 필요하다. 예를 들어 기술적인 방호벽 등 위험원이 사고로 진전되지 않도록 하기 위한 다양한 위험성 저감방안에 대한 세부적인 지식이 필요하다.

결과에 대한 심각도를 도출하기 위하여 최악의 경우의 시나리오 방법을 적용할 수 있다. 어떤 위험원은 사고상황이 다양하고, 사고의 심각도 또한 다양하므로, 제어시스템의 안전성 분석을 간략히 하기 위하여 근사화하는 작업이 필요한데 최악의 경우의 시나리오란 발생 가능한 최악의 결과를 특정 위험원과 관련시키는 것이다.

위험원 및 위험 분석를 위험원 식별과 위험성 계산으로 나누어 좀더 상세한 내용은 표 III.2.1 부터 표 III.2.4 를 참조한다.

표 III.2.1 위험원 및 위험분석 절차 정리

위험원 식별	
1. 예비위험원목록 (PHL)	응용시스템에 대한 예비위험원목록(PHL)을 작성한다. 이것은 모든 확인된 위험원을 수록하고, 일반적으로 안전성 분석보고서와 가상개시사건 목록을 참고한다.
2. 예비위험원분석 (PHA)	SW 에 의해 영향을 받는 응용시스템과 하부시스템들에 대한 예비위험원분석(PHA)을 작성한다. 이 분석에서 예비위험원목록(PHL)에 수록된 각 위험원을 평가하고, 그리고 각 위험원에 미칠 수 있는 SW 의 영향을 기술한다.

3. 위험원 조사 및 평가	요구되는 위험원 조사와 평가를 응용시스템과 그 하부 시스템의 수준에서 수행한다. 이것은 위험원들에 미치는 SW 의 영향 평가를 포함하여야 한다. 각 위험원에 관련된 SW 영향요소는 다음과 같다. ① SW 가 E/E/PES 시스템 안전 장치를 위협할 수 있다. 즉 SW 가 정확하게 동작하지 않으면 위험한 상황을 만들 수 있고, 그 상황은 다른 시스템에 의해서 재거 또는 완화되어야 한다. 예를들어 SW-기반 제어 장치가 고장나면 E/E/PES 시스템을 불안전한 작동으로 시스템 이상 현상을 일으킬 수 있다. ② SW 는 어떤 위험원이 사건(incident)으로 진전되는 것을 막을 수 있다. 때문에 SW 가 정확하게 동작하지 않으면, 그 위험원이 사건으로 진행할 가능성이 있다. 예를 들어, E/E/PES 시스템 정지 장치의 SW 가 고장나면 비상시 E/E/PES 시스템 이상로 아주 심각한 사건으로 진행할 수 있다. ③ SW 는 그 시스템을 위험한 상태에서 위험하지 않는 상태로 가져갈 수 있다. 위험한 상태는 SW 가 아닌 응용시스템의 특정 부분에서 생길 수 있다. 비상노심냉각장치를 제어하는 SW 가 그러한 사례이며, 이 장치는 다른 냉각 장치를 사용할 수 없을 때 E/E/PES 시스템을 고온 정지에서 저온 정지로 전환하기 위해 잔열을 제거한다. ④ SW 는 사고 결과를 완화하는데 사용될 수 있다. 예를 들면, 격납 건물 격리 장치를 제어하는 SW 는 일반 대중에게 피해를 줄 수 있는 방사성 방출을 격납건물 안으로 격리시킬 수 있다.
위험성 계산	
4. 심각도 및 발생가능성 산정	확인된 각 위험원에 대해서 발생에 따른 심각도 수준과 발생 가능성을 정한다. 표 III.2.2 과 표 III.2.3 는 이를 위한 기준으로 사용될 수 있다. IEC 61226 과 MIL-STD- 882 를 기반으로 작성되었다.
5. 위험성 추정	위의 단계 4 에서 인용한 도표를 이용하여 표 II.2.4 를 작성한다. 표 III.2.3 는 각 위험원에 대해 리스크 추정치를 도출하는데 사용될 수 있다. 표 III.2.4 전체 리스크 척도를 얻기 위해 표 III.2.2 의 위험원 심각도와 표 III.2.3 의 위험원 발생확률을 조합한 것이다. 따라서 치명적인 심각도와 비교적 빈번한 발생 확률을 갖는 사건들은 높은 리스크를 갖는 것으로 판단된다.
6. 위험성 수준 파악	예비위험원목록(PHL), 예비위험원분석(PHA) 또는 다른 위험원분석에서 확인된 각 위험원에 대해서는 위의 단계 5 에서 작성된 도표를 이용하여 리스크 수준을 파악한다.

표 III.2.2 위험원 심각도 (IEC 61126 참조)

내 용	범 주	정 의
파국적 (catastrophic)	A	사망, 시스템 상실, 또는 심각한 환경 파손
치명적 (critical)	B	심각한 재해, 심각한 직업병, 상당한 시스템 또는 환경 파손
한계적 (marginal)	C	사소한 재해, 사소한 직업병, 사소한 시스템 또는 환경 파손
무시가능(negligible)	-	사소한 재해나 직업병보다 더 낮음, 사소한 시스템 이나 환경 파손보다 더 낮음

표 III.2.3 위험원 발생확률(Mi-Std-882C 참조)

내 용	수준	확률 추정치
자주 발생 (frequent)	A	자주 발생할 수 있음.
빈번히 발생 (probable)	B	수명 기간에 몇 차례 발생함.
가끔 발생 (occasional)	C	수명 기간에 한, 두 차례 발생할 수 있음.
거의 발생치 않음 (remote)	D	수명 기간에 발생하지는 않으나, 배제할 수는 없음.
발생 가능성 없음 (improbable)	E	확실히 발생하지 않으나, 일어나지 않을 것으로 가정함.

표 III.2.4 위험성 수준 결정을 위한 매트릭스

구분	위험원 범주			
빈 도	파국적 (catastrophic)	치명적 (critical)	한계적 (marginal)	무시 가능 (negligible)
자주 발생 (frequent)	높음	높음	높음	중간

빈번히 발생 (probable)	높음	높음	중간	낮음
가끔 발생 (occasional)	높음	높음	중간	낮음
거의 발생치 않음 (remote)	높음	중간	낮음	낮음
발생 가능성 없음 (improbable)	중간	낮음	낮음	낮음

4. 단계 : 안전 요구사항 명세

이 단계에서는 E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비에 대해 필요한 기능안전성을 달성하기 위해 안전기능 요구사항 및 안전무결성 요구사항의 측면에서 전체 안전 요구사항에 대한 명세를 개발한다.

위험원 및 위험 분석으로부터 얻은 위험한 사건에 기초하여 필요한 모든 전체 안전기능 요구사항에 대한 명세를 개발한다. 보안 위협이 확인된 경우 보안 요구사항을 규정하기 위해 취약점 분석을 수행한다.

각각의 전체 안전기능을 위해 목표 안전무결성 요구사항을 허용 리스크에 부합되도록 결정하고 각각의 요구사항은 정량적 또는 정성적 방법으로 결정하며, 이는 전체 안전무결성 요구사항의 명세를 구성되며, 전체 안전무결성 요구사항은 다음중 하나의 관점에서 규정한다.

- 허용 가능 리스크를 달성하는 데 필요한 리스크 감소
- 허용 가능 리스크를 충족하기 위한 허용 가능 위험한 사건

EUC 리스크를 평가할 때 단일 EUC 제어 시스템 기능의 위험측 고장의 평균 빈도가 시간당 10^{-5} 이하일 경우 그 EUC 제어 시스템은 요구사항에 따르는 안전관련 제어 시스템으로 간주한다.

EUC 제어 시스템의 고장으로 하나 이상의 E/E/PE 안전관련 시스템 및/또는 기타 리스크 감소 설비가 필요한 경우, 그리고 EUC 제어 시스템을 안전관련 시스템으로 지정하지 않는 경우, 다음의 요구사항을 적용한다.

- 위험측 고장률은 다음에서 얻은 데이터로써 확인
  - 유사한 응용에서의 EUC 제어 시스템의 실제 운영 경험
  - 인정되는 절차에 의해 수행된 신뢰도 분석
  - 일반 장비의 신뢰성있는 산업 데이터베이스
- 위험측 고장률은 시간당 10^{-5} 보다 높아야 함
- 합리적으로 예측 가능한 모든 위험측 고장 모드는 전체 안전 요구사항에 대한 명세의 개발을 고려
- E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비로부터 독립적

위 요구사항을 충족할 수 없는 경우, EUC 제어 시스템을 안전관련시스템으로 지정해야 한다. EUC 제어 시스템 기능의 안전무결성 수준에 따라서 EUC 제어 시스템에 대해 요구되는 위험측 고장률에 의해 결정된다. 그러한 경우, 할당된 안전무결성 수준에 해당되는 요구사항을 EUC 제어 시스템에 적용한다

5. 단계 : 안전 요구사항 할당

지정된 E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비에 전체 안전 요구사항(전체 안전기능 요구사항과 안전무결성 요구사항 모두)에 대한 명세를 포함하여 안전기능을 할당하고, 각 안전 기능에 목표고장 기준과 안전무결성 수준을 할당한다.

필요한 기능안전성을 달성하기 위해 사용될 지정된 안전관련 시스템을 명시한다. 허용가능 리스크 감소는 E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비에 의해 충족될 수 있다.

각 안전기능과 안전무결성 요구사항을 하나 이상의 지정된 E/E/PE 안전관련 시스템 또는 기타 리스크 감소 설비에 할당하여 안전기능을 위한 허용 가능 리스크가 달성될 수 있도록 한다. 이런 할당은 반복적이며, 허용 가능 리스크가 달성되기 어렵다고 판단되면 EUC 제어 시스템, 지정된 E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비를 위한 명세를 변경하고, 할당을 반복한다.

안전무결성 요구사항 할당은 확률을 조합하는 적절한 기법을 이용하여 수행한다. 할당이 충분히 진행되면 E/E/PE 안전관련 시스템(들)에 할당된 각 안전기능에 대한 안전무결성 요구사항을 안전무결성 수준으로 명시한다. 표 Ⅲ.2.5 을 참고한다.

- 저요구 작동모드(low demand mode of operation)에 대한 안전 기능의 요구 시 위험측 고장평균 확률(PFDavg, Probability of Failure on Demand)
- 고요구 작동모드 (high demand mode of operation)에 대한 안전 기능의 위험측 고장 평균 빈도(PFH, Probability of Failure per Hour)
- 연속적인 작동모드(continuous mode of operation)에 대한 안전 기능의 위험측 고장 평균빈도(PFH).

위와같이 세가지 유형의 작동 모드를 구분하는 이유는 빈번한 호출이 이루어지지 않는 경우에는 확률적으로 정확한 측정이 어렵기 때문에 이를 보완하기 위한 별도의 측정 변수 및 기준이 필요하기 때문이다.

표 III.2.5 안전무결성 수준(IEC61508 기준)

:: 안전무결성 수준: 저요구 작동모드에서 운영되는 안전기능에 대한 목표고장 기준

안전무결성 수준 (SIL)	안전기능 요구에 대한 위험측 고장평균 확률 (PFDavg)
4	$\geq 10^{-5}$에서 $< 10^{-4}$
3	$\geq 10^{-4}$에서 $< 10^{-3}$
2	$\geq 10^{-3}$에서 $< 10^{-2}$
1	$\geq 10^{-2}$에서 $< 10^{-1}$

:: 안전무결성 수준: 고요구 작동모드 또는 연속적인 작동모드에서 운영되는 안전기능에 대한 목표고장 기준

안전무결성 수준 (SIL)	안전기능[h^{-1}] 의 위험측 고장평균 빈도 (PFH)
4	$\geq 10^{-9}$에서 $< 10^{-8}$
3	$\geq 10^{-8}$에서 $< 10^{-7}$
2	$\geq 10^{-7}$에서 $< 10^{-6}$
1	$\geq 10^{-6}$에서 $< 10^{-5}$

서로 다른 안전무결성 수준을 가지는 안전기능들을 구현하는 E/E/PE 안전관련 시스템의 경우 특정 안전기능들 간의 구현 독립성이 충분하다는 것을 보여줄 수 없다면, 이렇게 구현 독립성이 불충분한 안전관련 하드웨어와 소프트웨어의 각 부분(Part)들은 가장 높은 안전무결성 수준을 가지는 안전기능에 속하는 것으로 취급한다. 그러므로 가장 높은 관련 안전무결성 수준에 적용 가능한 요구사항들을 모든 부분(Part)에 적용한다.

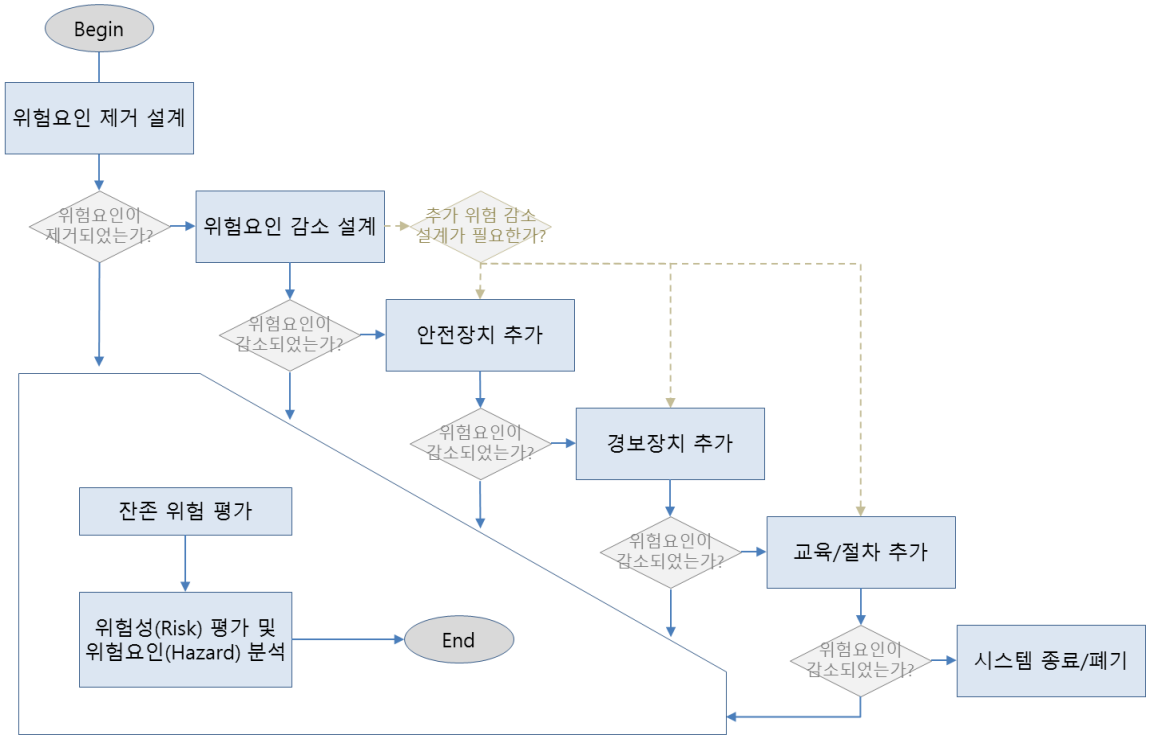
가. 안전 기능 할당

시스템의 위험원에 대하여 위험성이 정해진 후 해당 위험성을 수용할 수 없을 경우에 어떻게 위험원의 발생빈도를 허용 가능한 발생빈도로 낮출 것인지, 제어시스템의 위험성을 허용 가능한 목표치에 도달시키기 위한 안전 기능은 무엇인가에 대한 문제를 해결한다. 이러한 문제를 해결하기 위해서는 다음 사항을 만족 해야한다.

- 모든 위험원과 관련된 발생원인을 세부적으로 도출한다.
- 발생 원인이 위험원으로 진전되지 않도록 하는 안전기능을 도출한다.
- 발생 원인을 나눌 경우 가능한 한 최소한으로 나누어야 한다. (Minimal Cut Set)

각 안전기능은 개발된 관련 안전무결성 요구사항을 하나 이상의 지정된 E/E/PE 안전관련 시스템 및/또는 기타 리스크 감소 설비에 할당하여 안전기능을 위한 허용 가능 리스크가 달성될 수 있도록 한다. 이런 할당은 반복적이며, 허용 가능 리스크가 달성되기 어렵다고 판단되면 위험요인을 감소하기 위한 기능을 추가로 개발하여 EUC 제어 시스템, 지정된 E/E/PE 안전관련 시스템 및 기타 리스크 감소 설비를 위한 명세를 변경하고, 할당을 반복한다. 명세된 안전기능을 모두 할당하고 목표고장 기준으로 각 안전기능을 정의한다. 그림 Ⅲ.2.3 는 위험원 제거를 위한 안전기능 할당의 절차를 보여준다.

그림 Ⅲ.2.3 위험원 제거을 위한 안전 기능 할당 절차



(참조 : JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING HANDBOOK, DOD)

제어시스템의 안전 기능을 도출하여 해당 위험성을 낮추려고 할 경우에는 가능한 한 최악의 경우를 고려한다. 그러나 제어시스템과 같이 복잡한 시스템의 안전기능의 정도를 설정함에 있어서 다음과 같은 어려움이 있다.

- 시스템의 갖가지 기능들이 고장날 경우 그 고장 정도는 매우 다양하다.
- 시스템의 고장을 방지할 안전 기능은 일반적으로 몇 가지 위험원에만 영향을 미친다.
- 위와 같은 특성을 가지는 안전기능에 단순히 안전 무결성 수준을 설정하여 평준화하는 것은 주관적이며, 모호한 작업일 수 있다.

이러한 SIL 할당의 어려움을 보완하기 위하여 다음과 같은 절차를 적용한다.

- 조치를 취해야 하는 위험원에 대하여 최대로 수용 가능한 발생 빈도 또는 허용 가능한 위험률을 적용한다.
- 어느 고장측 고장이 어떤 기능에 영향을 미쳐 잠재적으로 위험원이 발생하는지 확인한다. 이는 어느 고장측 고장이 다른 고장측 고장과 And 게이트로 연결되어 있는지 확인하는 것이다.
- 잠재적인 위험원 중 가장 낮은 값을 선택한다.
- 해당 안전 기능을 안전무결성 수준으로 전환한다.

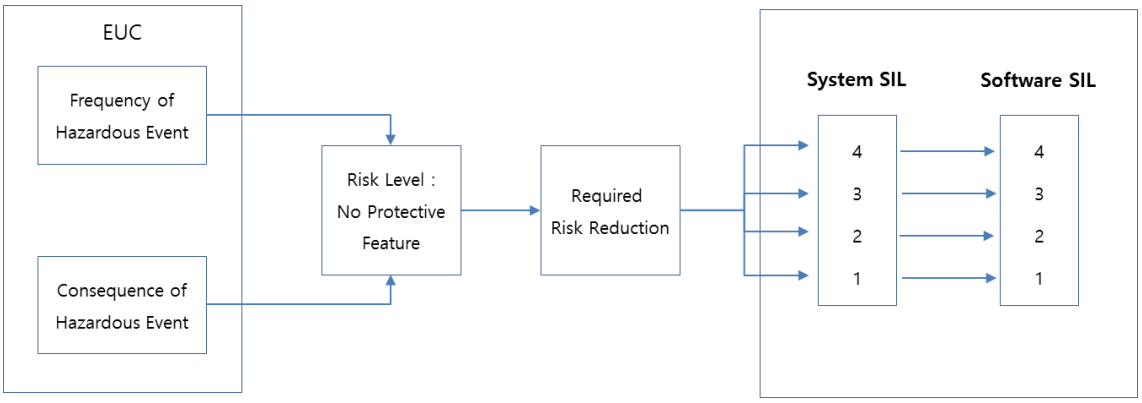
제어시스템을 운영하는 동안 다양한 상황이 발생한다. 매우 다양한 위험원이 발생할 수 있으나 그 발생 빈도를 평가한다는 것은 매우 어려운 작업이다. 예를 들어 인적 요인과 관련된 위험원이 그런 경우이다. 따라서 각각의 위험원을 보완하는 안전기능들이 필요한지, 필요하지 않은지를 결정하는 문제부터, 안전기능에 필요한 레벨은 어느 정도인지를 결정하기 위해서 주어진 운영환경의 특징 및 통계값을 살펴보고, SIL 에 기반을 두어 좀더 자세히 위험성 분석을 수행해야 한다.

나. 안전 무결성 수준(SIL) 결정

IEC61508 에서는 안전 무결성수준(SIL)을 주어진 조건하에 있는 안전관련 시스템이 주어진 시간 내에 요구되는 안전 기능을 만족스럽게 수행할 수 있는 확률로 정의하며, 크게 네가지 등급으로 분류하고 있으며, SIL 4 가 가장 높은 수준이고 SIL 1 이 낮은 수준이다. 안전 무결성은 안전 기능을 수행하는 안전관련 시스템의 성능과 관계되어 있다.

SW 안전 무결성 수준은 모 시스템의 안전 무결성 수준을 하위시스템으로 할당하는 과정을 통해서 결정된다. 즉, 시스템의 안전 무결성 수준을 SW 를 하나의 구성요소로 갖고 있거나 또는 SW 가 유일한 구성요소일 수 있는 하위시스템으로 할당하므로 하위 SW 시스템에 대한 안전 무결성 수준은 상위의 시스템 안전 무결성 수준과 같은 수준으로 배정한다. 그림 III.2.4 에서 할당 과정을 보여준다.

그림 III.2.4 안전 무결성 할당 과정



이와 같은 조건은 SW 안전 무결성 수준이 다음과 같은 사항들을 고려하여 감소될 때까지 그대로 유지된다.

위험 완화 기능을 수행하는 하위 시스템에서 단일 또는 복수 고장이 발생하여도 위험한 사건(hazardous Event)이 감소되도록 안전 기능을 복수 이상 제공하는 구조적 설계를 시스템에 반영한다. 하위 시스템이 도출된 위험한 사건 또는 감소

기능에서 어떤 역할을 수행하는지 파악한다. 하위 시스템들의 역할과 그 연계를 파악할 수 있도록 충분히 상세하게 시스템의 구조적 특징이 정의되어야 한다.

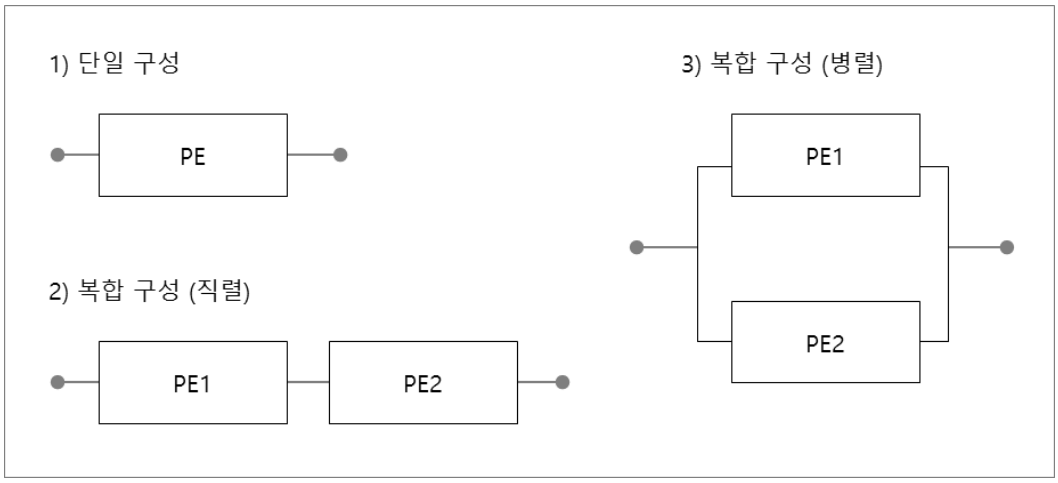
SW 의 무결성 수준을 결정하기 위해 필요한 내용은 다음과 같다.

- 시스템 안전 무결성 수준
- 위험원 목록과 각 위험원에 대한 다음과 같은 정보
- 위험원을 초래한 수 있는 개시 사건들
- 각 개시 사건에 대한 발생 예상 빈도 또는 확률
- 각 하위시스템의 역할을 결정하고 완화 기능을 파악하기 위해 충분히 상세한 시스템 구조의 정의

하위 시스템에 안전 무결성 수준을 할당하는 방법은 아래와 같다.

- 대상 시스템을 구성하는 모든 하위 시스템들을 규명한다.
- 어떤 하위시스템의 고장이 단일 또는 복합적으로(다른 하위시스템의 상태와 조합해서) 시스템의 위험원이 되는지 결정한다. 복합 구성에서 만약 한 하위 시스템의 고장이 독자적으로 시스템의 위험원이 된다면 그 하위 시스템의 안전 무결성 수준은 시스템의 안전 무결성 수준과 동일하게 할당된다. 만약 그 하위시스템의 고장이 다른 하위시스템들의 상태와 조합해서 시스템의 위험원이 된다면 그 하위시스템의 무결성 수준은 다음에 정의된 평가결과에 따라 감소될 수 있다. 다음에 언급된 평가는 강제 사항은 아니나, 하위시스템의 무결성 수준을 감소시키기 위해 수행한다.
- 어떤 하위시스템의 고장이 독립적으로 또는 다른 하위시스템의 상태와 조합해서 시스템의 완화 기능의 수행 여부에 영향을 줄 수 있는지 결정한다. 그림 III.2.5 은 시스템 사이의 관계를 개념적으로 보여준다.

- **그림 III.2.5 시스템 사이의 관계**



하나의 하위 시스템의 고장이 독자적으로 시스템의 완화 기능을 수행하지 못하게 한다면 그 하위시스템의 무결성 수준은 시스템의 무결성 수준과 동일하게 할당된다. 만약 그 하위시스템의 고장이 다른 하위시스템들의 상태와 조합해서 시스템의 완화 기능을 수행하지 못하게 한다면 그 하위시스템의 무결성 수준은 다음에 정의된 평가결과에 따라 감소될 수 있다. 다음에 언급된 평가는 강제사항은 아니나, 하위 시스템의 무결성 수준을 감소시키기 위해 수행한다.

- 고장이 시스템의 위험원이 되지 않거나, 시스템 사건의 완화 기능과 관계가 없는 SW 가 탑재된 하위시스템들이 있는지를 결정한다. 그러한 SW 는 가장 낮은 무결성 수준을 할당한다. SW 고장이 위험원을 초래할 수 없도록 하기 위하여 결함 격리(fault isolation)가 필요하다. 시스템의 설계 담당자와 안전(또는 품질보증) 담당자는 결함이 적절하게 격리되도록 하기 위하여 시스템의 구조가 적합한지 조사해야 한다. 만약 결함 격리가 고장처리방법으로 가능하다면 그 방법은 시스템과 동일한 SW 무결성 수준이 할당된다.

위의 네 가지 절차는 SW 만으로 이루어진 모든 하위시스템들의 무결성 수준이 결정될 때까지 또는 SW 를 하나의 구성요소로 하는 모든 하위시스템들의 무결성 수준이 설계 담당자와 안전 담당자에게 적절하다고 인정될 때까지 반복적으로 적용한다.

다. SW 안전 무결성 수준 결정(참고)

SW 의 안전 무결성 수준은 SW 가 탑재되는 시스템의 안전 무결성 수준을 하위 시스템으로 할당하는 과정을 통해 결정하는 경우도 있다.

안전을 중요한 목적으로 개발 또는 사용되는 SW 의 SIL 은 일반적으로 수행되어야 할 기능의 안전 중요도에 따라 5 개 등급, 또는 표 Ⅲ.2.6 과 같이 1~4 등급으로 분류할 수도 있다. SW 의 안전 무결성 수준이 높을수록 SW 의 안전중요도는 높아지며, 안전중요도가 높다는 것은 SW 의 고장이 시스템에 미치는 위험성이 높다는 것을 의미한다.

표 Ⅲ.2.6 SW 안전 무결성 수준 체계 예시

SW 안전 무결성 수준(SIL)	안전 중요도
4	매우 높음
3	높음
2	중간
1	낮음

SW 의 안전 무결성 수준을 결정하기 위해서는 다음과 같은 조건이 만족되어야 한다.

- 시스템의 안전 무결성 수준이 결정되어 있어야 한다.
- 하위 시스템들의 역할과 그 연계를 규정할 수 있도록 충분히 상세한 시스템의 구조적 특성이 정의되어 있어야 한다.
- 위험원 목록과 각 위험원에 대하여 위험원을 초래할 수 있는 개시 사건들 및 각 개시 사건에 대한 위험성이 결정되어 있어야 한다.

SW 는 제어 특성에 따라 표 Ⅲ.2.7 과 같이 6 개의 제어 범주, ①, ②, ③, ④, ⑤, ⑥로 구분할 수도 있다.

표 III.2.7 SW 제어 구분(예시)

SW 제어 범주	SW 의 제어 특성
①	시스템의 기능을 독자적으로 제어하는 SW. 해당 SW 의 오작동으로 인한 위험원 발생을 완화하기 위한 다른 독립 적인 하위 시스템이 없어서 해당 SW 의 고장이 직접적으로 시스템의 위험원을 발생시킨다.
②	시스템의 기능을 제어하는 SW. 해당 SW 의 오작동으로 인한 위험원 발생을 완화하기 위한 다른 독립적인 하위 시스템이 있다.
③	시스템의 위험원을 탐지하고, 위험원을 완화하기 위한 사용자의 조치를 요구하는 기능을 수행하는 SW. 해당 SW 의 오작동은 시스템의 위험원을 발생시킨다.
④	시스템의 기능을 제어하는 SW. 해당 SW 의 오작동으로 인한 위험원을 방지할 수 있는 다중의 독립된 하위 시스템이 존재한다.
⑤	시스템의 위험원을 탐지하고, 위험원을 완화하기 위한 사용자의 조치를 요구하는 기능을 수행하는 SW. 그러나 해당 SW 외에도 다중의 독립적인 상태정보를 제공하는 독립적인 하위 시스템이 있다.
⑥	시스템의 기능을 제어하지 않으며, 사용자 조치를 위한 정보를 제공하지도 않는 SW

표 III.2.8 은 SW 의 안전무결성 수준 결정을 위한 SW 위험원 심각성 매트릭스로 SW 의 안전무결성 수준은 SW 의 제어특성과 시스템의 안전무결성 수준의 조합으로 결정된다. 세로축은 SW 의 제어범주를 나타내며, 가로축은 시스템의 안전무결성 수준을 나타낸다. 예를 들어, 시스템의 안전무결성 수준이 4 이고 SW 의 제어범주가 ④이면, 해당 SW 의 안전무결성 수준은 최소한 2 이상으로 개발되어야 한다.

표 III.2.8 SW 위험원 심각성 매트릭스 예시

안전중요도 / 제어 범주	소프트웨어 안전 무결성 수준 (SIL)			
	매우높음	높음	중간	낮음
①	4	3	2	1
② & ③	3	2	1	0
④ & ⑤	2	1	0	0
⑥	1	0	0	0

제 3 장. 위험 분석 사례

1. 단계 : 개념

최근 부유식 원유생산저장하역설비(FPSO) 또는 초대형액화천연가스수송선(Large LNG carrier)이 많이 건조됨에 따라 천연가스와 디젤유를 연료로 사용하는 이중연료엔진(Dual Fuel engine)의 사용이 증가하였고, 앞으로는 국제해사기구(IMO)의 환경규제 강화로 인하여 친환경적인 이중연료엔진의 사용이 일반화될 전망이다. 이중연료엔진은 천연가스를 주 연료로 사용하며, 점화를 위한 파일럿유(Pilot oil)가 소량 사용된다. 이 때 천연가스는 이중연료엔진이 설치된 기관실로 연료가스공급시스템(Fuel Gas Supply system, FGS system)을 통해 일정한 압력, 온도로 공급되어야 한다. 만약 천연가스가 기관실 내에서 누출되면, 천연가스의 높은 폭발성으로 인해 플랜트 또는 선박은 인적, 물적, 환경적으로 큰 피해를 입을 수 있다.

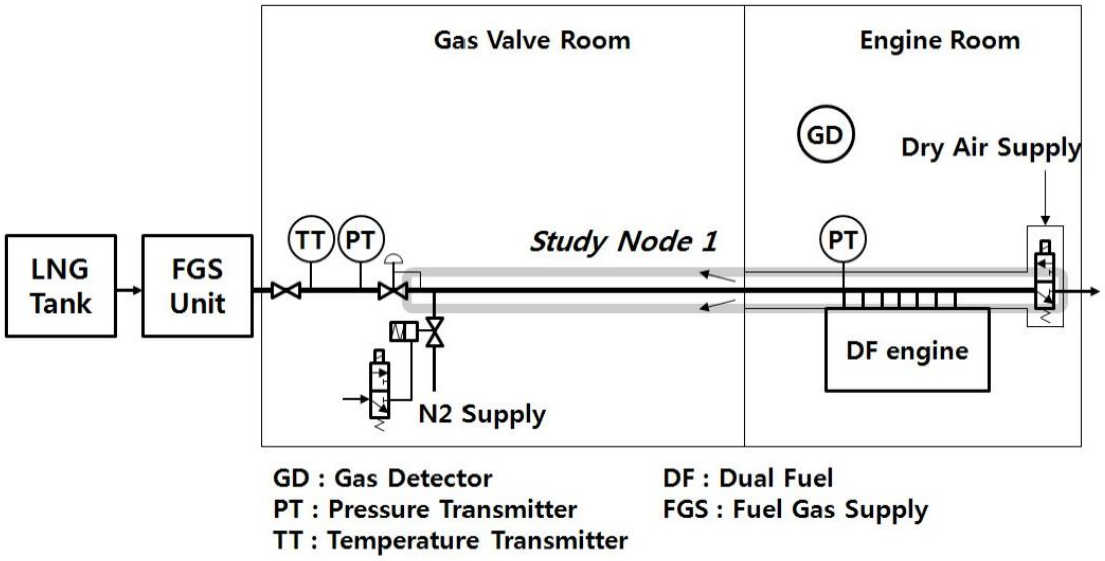
기관실 내 가스배관의 경우 이중배관으로 규제되고, 압축공기를 이용해 이중배관과 가스배관 사이의 공간을 항상 불어내고 있어 가스 누출로 인한 화재 및 폭발 가능성이 비교적 낮다고 생각할 수 있다. 하지만 이중배관이 항상 정상 상태를 유지하는지 확인할 수 없으며, 천연가스는 이중배관만 통과하는 것이 아니라 엔진 자체 배관, 필터, 계장류 등 많은 장치를 거쳐 다수의 엔진 실린더로 공급되기 때문에 가스 누출로 인한 사고에 대해 안심할 수 없다. 따라서 연료가스공급시스템의 안전시스템은 안전무결성에 기반하여 설계하고 검토한다.

본 사례는 2012 년 대한조선학회에 발표된 "이중연료엔진의 연료가스공급 시스템에 대한 안전무결성 기반 안전계장시스템 설계" 논문에 제시된 사례를 본 가이드의 위험분석 단계(IEC61508 기준)에 맞게 편집한 내용이다.

2. 단계 : 범위 정의

시스템을 여러 개의 분석대상으로 나눠 실시한다. 일반적으로 분석대상은 분리기와 같은 장비 자체 또는 장비와 그 장비 사이를 연결하는 배관을 의미한다. 시스템이 단순할 경우전체시스템이 하나의 분석대상이 될 수도 있다. 그림 Ⅲ.3.1 은 대상시스템 구성도이다.

그림 Ⅲ.3.1 Fuel Gas Supply system 구성도



분석 대상은 차단밸브(2 개) 및 관련 안전장치를 제외시킨 일반적인 연료가스공급시스템 및 이중연료 엔진의 간략한 P&ID(Piping and Instrument Diagram)이다. 연료가스공급 유닛(FGS Unit)에서 공급된 일정 온도, 압력의 천연가스가 압력제어밸브에 의해 이중연료엔진 작동에 적당한 압력으로 감압된다. 안전을 위해 엔진실의 가스배관은 이중배관으로 되어 있으며, 압축 공기를 주입하여 가스 누출 시 가스가 항상 가스밸브실(Gas Valve Room)로 배출한다. 엔진실에는 가스 누출 감지기가 설치되어 가스 누출을 감지한다. 그리고 가스 공급 중단 시 가스 공급 배관은 질소에 의해 제거되어 항상 안전한 상태를 유지한다. 이중연료엔진과

관련된 연료가스공급시스템은 하나만 설치되는 것이 아니라 일반적으로 4 세트 이상 설치된다. 본 분석에서는 4 세트의 시스템이 설치되었다고 가정하고, 연료가스공급시스템의 운용모드(Operation Mode)는 '정상적인 가스 공급(Normal Operation)'만 고려하였다.

3. 단계 : 위험원 및 위험 분석

본 사례에서는 HAZOP(Hazard and Operability, 상세한 내용은 위험분석기법-HAZOP 참조)을 이용하여 위험 분석을 수행한다. 대부분의 정성적 위험원 식별 작업과 마찬가지로 HAZOP 역시 난상토론(Brainstorming)을 원활히 하기 위해서는 안내 단어(Guideword)가 필요하다. 안내 단어는 일반적으로 공정의 비정상적인 상태(deviation)를 나타낸다. 실제 안내 단어는 아주 많이 있으며 해당 시스템에 적합한 안내 단어를 HAZOP 회의 전에 여러 참가자들과 함께 선택한다. 표 III.3.1 은 안내 단어의 예시를 보여준다.

HAZOP 회의는 설계, 시운전, 유지보수, 위험도 등 다양한 경험을 가진 엔지니어들로 구성된 팀에 의해 수행되어야 한다. HAZOP 결과의 품질은 HAZOP 에 참여한 엔지니어들의 실력에 가장 큰 영향을 받기 때문에 HAZOP 진행자는 회의 주최 시 이 부분을 가장 중요시해야 한다. 모든 위험원을 식별하기 위해서는 유지보수, 퍼지(Purge), 정비, 시동 등 모든 운용모드에 대해 HAZOP 이 수행한다. 본 사례 는 운영모드에 대한 사례이다.

HAZOP 의 절차를 간단히 소개하면 아래와 같다.

① 안내단어(Guide-word): 먼저 GUide-word 예시에서 안내 단어 하나를 선택한다.
② 원인(Possible cause): 안내단어의 원인이 되는 요소를 먼저 식별한다.
③ 결과(Consequence): 원인으로 인한 어떤 결과가 발생하지를 식별한다.

④ 안전장치(Safeguards): 원인의 발생빈도 또는 결과의 심각도에 영향을 미치는 모든 안전 기능 또는 장치를 식별한다.

표 III.3.1 안내 단어 (Guide-word) 예시

No.	Guide words		가능 위험원
1	높은 유속	High flow	지나친 압력
2	낮은 유속	Low flow	압력 제어 밸브의 고장
3	유속 없음	No flow	가짜 밸브 닫음
4	역류	Reverse flow	
5	고수준	High level	수준 조절 밸브의 고장
6	저수준	Low level	용기 누출
7	수준 없음	No level	
8	고압	High pressure	압력 제어 밸브의 고장
9	저압	Low pressure	
10	고온	High temperature	온도 조절 밸브의 고장
11	저온	Low temperature	열 교환 실패
12	2 상태 흐름	2-phase flow	가스 시스템에 액체 있음
13	불순물	Impurities	기체, 액체, 고체
14	기구	Instruments	제어 충분성, 많은 도구, 바른 위치
15	시운전	Commissioning	장비 없음, 잘못된 절차
16	유지보수	Maintenance	잘못된 절차
17	기타 작업	Other operation	시동, 정지, 비활성, 폭기 등
18	화재 또는 폭발	Fire and explosion	누출, 점화원인, 정전기
19	단순화	Simplicity	과량 장비, 밸브

회의 결과 식별된 원인, 결과, 안전 장치는 표 III.3.2 와 표 III.3.3 을 참조한다.

표 III.3.2 HAZOP Worksheet

안내어	원인	결과	안전장치
빠른 유속 (High flow)	압력조절밸브 고장	파이프 또는 장치 고장, 가스밸브실, 엔진실 가스누출 화재나 폭발로 인한 치명상.	수동 밸브, 이중벽 파이프.
느린 유속 (Low flow)	압력조절밸브 고장	가스공급 실패, DF engine 부하 바로 감소	D.O supply system, 압력송신기.

	가스배출밸브가 잘못 열리거나 새고 있음	가스공급실패, DF engine 부하감소, 가스밸브실, 엔진실 외부에 화재 또는 폭발	D.O supply system, 압력송신기.
유속 없음 (No flow)	압력조절밸브가 닫혀있음	가스공급실패, MDO mode 로 변화.	D.O supply system, 이중 DF engine.
불순물 (Impurities)	파이프에 불순물이 있음	압력조절밸브 고장, 압력조절 실패, 밸브가 닫힌 상태에서 가스 누출, DF engine 가스 투입 밸브 손상, Sol. v/v 손상 또는 폭발	
화재 또는 폭발 (Fire and explosion)	가스밸브실의 파이프에 가스 누출	화재 또는 폭발	공간 격리, 가스밸브실 환풍기, 이중벽 파이프.
	엔진실의 파이프에 가스 누출	화재 또는 폭발	공간격리, 가스밸브실 환풍기, 이중벽 파이프.

표 III.3.3 위험도, 빈도지수, 결과지수 기준

HAZID(Hazard Identification, HAZID) 또는 HAZOP(Hazard and Operability)과 같은 정성적 위험도 평가에서는 위험원에 대한 빠른 위험도 수준을 결정해야 하며, 그 방법 중 가장 효율적인 방법이 위험도 매트릭스이다. 위험도 매트릭스는 정성적 위험도 평가에 쓰이는 가장 일반적이며 합리적인 도구이다.

위험도(Risk Matrix)

Conse-Quence severity \ Event Frequency		1	2	3	4	5
		Once per Over 1,000 years	Once per 1,000 ~ 100 years	Once per 100 ~ 10 years	Once per 10 ~ 1 year	More than once per 1 year
Catastrophic	5	H	H	H	H	H
Critical	4	M	H	H	H	H
Major	3	M	M	H	H	H
Minor	2	L	L	M	M	H
Negligible	1	L	L	L	L	M

- H(High) 위험도: 허용 불가능한 위험도이며, 추가적인 정량적 분석을 통해 정확한 위험도 분석이 필요하다.
- M(Medium) 위험도: 허용 가능하지만, 비용-편익분석을 통해 위험도를 합리적으로 실행 가능한 수준까지 낮춰야 한다.

• L(Low) 위험도: 충분히 낮은 수준의 위험도이므로 무시할 수 있다.

이 위험도 매트릭스는 HAZOP 에 참석한 모든 참석자의 동의를 받아야 사용할 수 있고, 만약 불합리한 부분이 있으면 즉시 수정되어야 한다. 위험도 매트릭스는 정량적 개인위험도 허용기준(사망확률 10^{-4}/year)에 어느 정도 부합하지만, 정성적인 기준이기 때문에 완벽히 일치하지는 않는다. 아래에서 위험도 매트릭스의 빈도지수(Frequency Index)와 결과지수(Consequence Index)를 구체적으로 제시하고 있다.

빈도 지수(Frequency Index)

Index	Description	Frequency of event occurrence
5	Frequent	More than once per 1 year
4	Probable	Once per 1~10 year
3	Occational	Once per 10~100 years
2	Remote	Once per 100~1,000 years
1	Improbable	Once per over 1,000 years

결과 지수(Consequence Index)

Index	Consequence	Description	
		Effect on human safety	Effect on a offshore plant
1	Negligible	Disturbance or fatigue	No effect, Minor material damage
2	Minor	Medical treatment more than 12 hours	Minor production influence
3	Major	Permanent disability or prolonged hospital treatment	Production interrupted for weeks
4	Critical	One fatality	Production interrupted for months
5	Catastrophic	Several fatalities	Total loss

앞서 수행한 HAZOP 절차에 이어 추가로 아래와 같은 절차를 거쳐 위험 발생 빈도, 결과지수, 위험의 크기를 결정하고, 추가적인 안전 조치(기능)을 도출하는 작업을 수행하였다.

⑤ 빈도지수(Frequency Index, FI), 결과지수(Consequence Index ,CI), 위험도지수(Risk Index, RI) : 안전장치를 고려하여 결과의 발생빈도, 결과를 결정하고, 위험도 매트릭스를 참고하여 위험도지수를 결정한다.

⑥ 추가안전조치(Actions required) : 위험도 지수가 High 또는 Medium 이라면 추가적인 안전장치 또는 조치를 식별한다.

⑦ 안전조치 책임자(Actions allocated to) : 추가안전조치(Actions required)의 항목을 수행할 부서 또는 기관을 선정한다.

앞서 작성한 기록지에 추가로 도출된 내용은 표 III.3.4 와 같다. 총 7 개의 위험원이 식별되었다. 그 중 3 개는 High, 2 개는 Medium, 2 개는 Low 수준의 위험도를 가지고 있다.

표 III.3.4 HAZOP Worksheet

안내어	원인	결과	안전장치	빈도 결과 위험	추가안전조치 (안전요구사항)	안전 책임자
빠른 유속 (High flow)	압력조절밸브 고장	파이프 또는 장치 고장, 가스밸브실, 엔진실 가스누출 화재나 폭발로 인한 치명상.	수동 밸브, 이중벽 파이프.	3 4 H	가스검출기 설치 가스차단시스템 설치	설계팀
느린 유속 (Low flow)	압력조절밸브 고장	가스공급 실패, DF engine 부하 바로 감소	D.O supply system, 압력송신기.	4 1 L		
	가스배출밸브가 잘못 열리거나 새고 있음	가스공급실패, DF engine 부하감소, 가스밸브실, 엔진실 외부에 화재 또는 폭발	D.O supply system, 압력송신기.	2 3 M	가스배출시스템에 가스감시장치 설치	설계팀

유속 없음 (No flow)	압력조절밸브가 닫혀있음	가스공급실패, MDO mode 로 변화.	D.O supply system, 이중 DF engine.	2 1 L		
불순물 (Impurities)	파이프에 불순물이 있음	압력조절밸브 고장, 압력조절 실패, 밸브가 닫힌 상태에서 가스 누출, DF engine 가스 투입 밸브 손상, Sol. v/v 손상 또는 폭발		4 2 M	필터 설치	설계팀
화재 또는 폭발 (Fire and explosion)	가스밸브실의 파이프에 가스 누출	화재 또는 폭발	공간 격리, 가스밸브실 환풍기, 이중벽 파이프.	2 4 H	가스검출기 설치, 가스차단 시스템 설치	설계팀
	엔진실의 파이프에 가스 누출	화재 또는 폭발	공간격리, 가스밸브실 환풍기, 이중벽 파이프.	1 5 H	가스검출기 설치, 가스차단 시스템 설치	설계팀

4. 단계 : 안전 요구사항 명세

가. 안전 기능 요구사항 명세

안전기능 요구사항 명세를 위해 앞서 위험원 및 위험 분석 단계에서 도출된 겨로가에서 Low 수준의 위험원은 더 이상 고려할 필요가 없으며, High 와 Medium 수준의 위험도를 가진 위험원에 대한 안전 기능 요구사항 명세는 아래와 같다.

① 가스차단시스템 설치 - High Risk

가스공급배관에서 가스 누출 시 누출된 가스를 감지하고 가스 공급을 차단하기 위한 안전시스템이다. 관련 원인에 의한 위험도를 허용 가능한 수준까지 낮추기 위해 방호계층분석을 이용한 분석이 필요하다.

② 필터 설치 - Medium Risk

가스 공급 시 이물질을 차단시켜주는 필터를 비용/편익을 고려해서 설치한다.

이물질 유입으로 인한 압력제어밸브 손상, 이중연료엔진의 가스주입 밸브(Gas injection valve) 손상 등이 발생할 수 있기 때문에 필터 설치는 합리적인 것으로 판단된다.

③ 가스배출시스템에 가스감시장치 설치-Medium Risk
가스배출 용 솔레노이드 밸브(Solenoid valve)에서 소량의 가스가 지속적으로 누출되는 것을 방지하기 위해 가스배출시스템에 가스감시장치(가스검출기)를 비용/편익을 고려해 설치한다. 하지만 현재 주어진 자료만으로는 위험도를 정확히 결정할 수 없다. 그래서 전체 시스템 도면 및 장치 배치 상태를 검토 후 가스 배출로 인한 위험도를 다시 평가한다. 만약 위험도가 낮다(Low)면 가스감시장치를 설치할 필요가 없으며 위험도가 높다(High)면 가스감시장치의 설치를 고려한다.

안전 기능으로 'High' 위험도로 식별된 위험원을 허용 가능한 수준까지 위험도를 낮추기 위해 가스차단기능이 식별되었다. 식별된 원인은 가스공급배관 또는 가스밸브실 내의 배관에서의 천연가스 누출이다. 가스공급배관에서 누출된 가스는 이중배관을 따라 가스밸브실로 이동하여 외부로 빠져 나가며, 가스밸브실 내의 배관에서 누출된 가스도 역시 가스밸브실을 거쳐 외부로 빠져나간다. 즉, 어떤 원인이든 누출된 모든 가스는 가스밸브실을 지나 외부로 빠져 나간다.

그래서 가스차단기능의 정의는 '**가스밸브실 내의 천연가스 농도를 항상 감시하며, 일정 농도 이상이 되면 차단밸브를 작동시켜 가스 누출을 차단하는 기능**'이다. 가스차단 기능은 전체적 안전기능이며, 제어대상 장비는 가스밸브실이다.

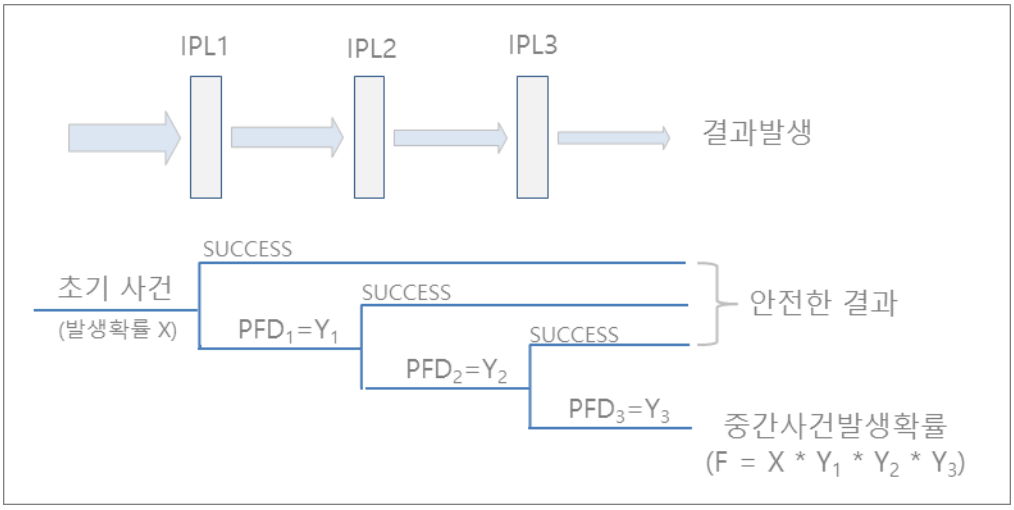
나. 안전 무결성 요구사항 명세

사례는 IEC61508 에서 제시하는 LOPA(IEC61508-5, Annex F)에 따라 아래와 같은 절차에 맞추어 실시하였다.

(1) 영향사건설명(Impact event description)과 심각도 수준(Severity level)
(2) 초기사건발생빈도(Initiating event frequency)
(3) 방호계측 식별 및 PFD 입력
(4) 중간사건 발생 빈도
(5) 안전무결성 수준과 안전무결성 요구사항

방호계층분석에서 사용하는 용어는 IEC61508 에서는 용어 정의가 미흡하여 한국산업안전보건공단의 방호계층분석(LOPA)기법에 관한 기술지침 을 인용하였다.

- "방호계층분석(Layer of protection analysis, LOPA)"이란 원하지 않는 사고의 빈도나 강도를 감소시키는 독립방호계층의 효과성을 평가하는 방법 및 절차를 말한다.
- "독립방호계층(Independent protection layer, IPL)"이란 초기사고나 사고 시나리오와 관련한 다른 어떤 방호계층의 작동과는 관계없이 원하지 않는 결과로 전개되는 것으로부터 사고를 방호할 수 있는 장치나 시스템 또는 동작을 말한다. 독립적이라는 것은 방호계층의 성능은 초기사고의 영향을 받지 않고 다른 방호계층의 고장으로 인한 영향을 받지 않는다는 것을 말한다.



- "방호계층(Protection Layer)"이란 시나리오가 원하지 않는 방향으로 진행 하지 못하도록 방지할 수 있는 장치, 시스템, 행위를 말한다.
- "안전계장기능(Safety Instrumented Function, SIF)"이란 한계를 벗어나는 (비정상적인) 조건을 감지하거나, 공정을 인간의 개입 없이 기능적으로 안전한

상태로 유도하거나 경보에 대하여 훈련받은 운전원을 대응하도록 하는 특정한 안전무결수준(SIL)을 가진 감지장치, 논리해결장치, 최종요소의 조합을 말한다.

- "기본공정제어시스템(Basic Process Control System, BPCS)"이란 공정이나 운전원으로 부터 나온 입력신호에 대응하는 시스템으로서 출력 신호를 발생시켜 공정이 원하는 형태로 운전되도록 하는 것을 말한다. 기본공정제어 시스템은 센서, 논리연산기, 공정제어기 및 최종제어요소로 구성되며 공정을 정상 생산범위 내에서 운전되도록 제어한다.
- "초기사건"란 원하지 않는 결과로 유도하는 시나리오를 개시시키는 사건을 말한다.
- "시나리오"란 원하지 않는 결과를 가져오는 사건이나 사건의 연속을 말한다.
- "작동요구시 고장확률(Probability of failure on demand, PFD)"이란 시스템 이 특정한 기능을 작동하도록 요구받았을 때 실패할 확률을 말한다.

1) 영향사건설명(Impact event description)과 심각도 수준(Severity level)

안전 무결성 요구사항 명세를 위해 목표 안전 무결성 수준을 사례에서 다루는 분야인 선박 해양 분야를 기준으로 설정했다. HSE(The Health and Safety Executive)의 "Reducing risk, protecting people, 2001(R2P2)"에 나타난 개인 위험도 허용 기준에서 직원에 대한 최대 허용 가능한 위험도는 10^{-3}/year 수준이다. 본 사례에서는 노르웨이에 의해 국제해사기구(IMO)에 제출된 표 Ⅲ.3.5 의 선박 위험도에 대해 10 배 강화된 기준을 적용했다.

표 Ⅲ.3.5 Suggested criteria for target risk for new ships from IMO MSC 72/16

Risk exposed persons	Individual Risk
Target individual risk for worker (e.g. crew members)	10^{-4}/year
Target individual risk for public (e.g. passengers and public ashore)	10^{-5}/year

컨테이너 선박에 이중연료엔진 및 연료가스공급시스템이 설치된다면, 그 장치 또는 시스템을 포함한 컨테이너 선박의 전체 개인위험도는 1×10^{-4}/year 를 넘어서는 안된다.

위험도 매트릭스의 각 결과수준(Consequence level)에 맞는 목표완화사건발생빈도(Target mitigated event likelihood, TMEL)를 결정해야 한다. 정량적인 목표완화사건발생빈도 기준은 발주처들에 따라 다양하다. 본 사례의 영향사건 설명은 앞 단계에서 도출된 HAZOP Sheet 의 결과와 그 내용이 같다. 그러므로 초기사건(Initiating event)에 의한 영향사건설명은 '화재/폭발로 인한 인명 및 장비 손상'이 될 수 있다. 심각도 수준은 결과지수와 일치하기 때문에 HAZOP 기록지와 같이 '4'가 된다. 표 III.3.6 을 참조한다.

표 III.3.6 Target mitigated event likelihood for LOPA

Severity level	Description		Target mitigated event likelihood
	Effect on human safety	Effect on a offshore plant or a ship	
1	Self treatment, Disturbance or fatigue	No effect, Minor material damage	$2\text{-}10^{-2}$/year
2	Medical treatment more than 12 hairs	Minor production influence, Minor propulsion intervention	$2\text{-}10^{-3}$/year
3	Permanent disability or prolonged hospital treatment	Production interrupted for weeks, Propulsion failure for weeks	$2\text{-}10^{-4}$/year
4	One fatality	Production interrupted for months, Propulsion failure for months	$2\text{-}10^{-5}$/year
5	Several fatalities	Total loss	$1\text{-}10^{-5}$/year

2) 초기사건발생빈도(Initiating event frequency)

초기사건 발생빈도는 장비 고장률에 대한 일반 데이터, 테스트 간격, 기존 기록 등을 통해 계산한다. 초기사건 발생빈도 낮을 경우는 데이터에 대한 충분한 통계적 근거가 있는 경우에만 사용해야 한다.

초기사건의 경우 HAZOP 에서 식별된 내용과 거의 같지만, 근본원인(Root cause)이 포함하지 못하는 경우도 있다. 이 경우 정확한 분석을 위해 위험상황을 잘 이해하고 근본원인을 반드시 식별해야 한다. 초기사건은 HAZOP 기록지의 원인와 그 내용이 같다.

- 압력제어밸브 고장 : 압력제어밸브가 오작동하고 비정상적인 고압의 가스가 엔진으로 공급되면 가스공급배관에서 가스가 누출될 수 있다.
  아래 표의 'BPCS instrument roop fail'의 값을 참고하여 1 세트의 시스템에 대해 $1{\cdot}10^{-1}$ 로 발생빈도를 결정하였다. 연료가스공급시스템이 4 세트이기 때문에 $4{\cdot}10^{-1}$ 이다. 표 III.3.7 를 참조한다.
- 가스공급배관에서의 가스누출 : 압력제어밸브가 정상 작동 중에도 가스공급배관에서 여러 원인으로 인해 가스가 누출될 수 있다.
  아래 표의 'Piping leak(10% section)-100m'와 'Gasket/packing blowout'를 참고하여 1 세트의 시스템에 대해 $1{\cdot}10^{-2}$ 로 발생빈도를 결정하였다.
  연료가스공급시스템이 4 세트이기 때문에 $4{\cdot}10^{-2}$ 이다.

표 III.3.7 Typical frequency values assigned to Initiating events adapted from CCPS (CCPS, 2001)

Initiating event	Frecuency range from literature (per year)	Example of a value chosen by a company
Pressure vessel residual failure	10^{-5} to 10^{-7}	$1\text{-}10^{-6}$
Piping leak (10% section)-100m	10^{-3} to 10^{-4}	$1\text{-}10^{-3}$
Atmospheric tank failure	10^{-3} to 10^{-5}	$1\text{-}10^{-3}$
Gasket / packing blowout	10^{-2} to 10^{-6}	$1\text{-}10^{-2}$
Lightning strike	10^{-3} to 10^{-4}	$1\text{-}10^{-3}$
Safety valve opens spuriously	10^{-2} to 10^{-4}	$1\text{-}10^{-2}$

Pump seal failure	10^{-1} to 10^{-2}	$1\text{-}10^{-1}$
BPCS instrument loop failure	1 to 10^{-2}	$1\text{-}10^{-1}$
Regulator failure	1 to 10^{-1}	$1\text{-}10^{-1}$

3) 방호계측 식별 및 PFD 입력

각 방호계층은 다른 계층과 독립적으로 기능하는 장비 또는 관리 제어 그룹으로 구성된다. 방호계층은 HAZOP Sheet 의 안전장치와 그 내용이 같다. 각 방호계층을 LOPA 기준으로 간단히 설명하면 'General process design'에는 이중배관 또는 용기가 해당되며, 'BPCS(Basic Process Control System)'에는 제어 시스템이 해당되며, 'Alarms'에는 운전자의 안전조치가 해당된다. 'Additional mitigation'에는 기계적 장치, 구조적 장치, 안전 절차가 해당되며, 'IPL(Independent Protection Layer) additional mitigation'에는 감압밸브, 수로(Dike), 일제개방밸브(Deluge system), 이산화탄소 배출시스템(CO2 release system)가 해당된다.

- 방호계층(압력제어밸브 고장) :
  - 알람 또는 기타 방호계층(Alarms, etc.) : 가스공급라인에 압력이 상승하면 압력감지기가 그 압력을 감지하여 알람을 울린다. 그 알람을 확인한 제어실의 엔지니어가 수동밸브를 잠근다. 여기에 대한 PFD 은 표 III.3.8 의 'Operator alarm with sufficient time available to respond'와 아주 유사하기 때문에 $1\cdot10^{-1}$ 로 정하였다.
  - 추가완화(Additional Mitigation) : 가스 누출이 계속 될경우의 점화확률은 보수적으로 판단하여 $1\cdot10^{-1}$ 로 결정했다.
- 방호계층(가스공급배관에서의 가스누출) :
  - 추가완화 : 가스공급배관에서 가스 누출 시 가스누출 감시장치 없이는 엔지니어가 가스 누출을 인지하기는 쉽지않다. 그래서 이 초기 원인에 대한 방호계층은 점화확률 밖에 없으며, 점화확률을 보수적으로 판단하여 $1\cdot10^{-1}$ 로 정하였다.

표 III.3.8 PFDs for PLs (CCPS, 2001)

Independent Protection Layer	PFD
BPCS, if not associated with the initiating event being considered	$1\text{-}10^{-1}$
Operator alarm with sufficient time available to respond	$1\text{-}10^{-1}$
Relief valve	$1\text{-}10^{-2}$
Rupture disc	$1\text{-}10^{-2}$
Flame / detonation arrestors	$1\text{-}10^{-2}$
Dike / bund	$1\text{-}10^{-2}$
Underground drainage system	$1\text{-}10^{-2}$
Open vent (no valve)	$1\text{-}10^{-2}$
Fireproofing	$1\text{-}10^{-2}$
Blast-wall / bunker	$1\text{-}10^{-3}$
Identical redundant equipment	$1\text{-}10^{-1}$ (max credit)
Diverse redundant equipment	$1\text{-}10^{-1}$ to $1\text{-}10^{-2}$
Other events	Use experience of personnel

4) 중간사건 발생 빈도(Intermediate Event Likelihood)

중간사건발생빈도는 방호계층 적용시의 발생빈도로 초기사건발생빈도와 방호계층의 PFD 을 모두 곱하면 각 초기사건에 대해 4·10-3 로 간단히 구할 수 있다. 차단기능이 각 초기사건에 대한 안전계장기능이기 때문에 각 초기사건에 대한 중간사건발생빈도를 모두 더하면 총중간사간발생빈도(Total intemediate event likelihood)는 8·10-3/year 가 된다.

각 초기사건의 심각도 수준이 '4'이기 때문에 목표완화사건발생빈도는 $2\cdot10^{-5}$/year 가 된다.

5) 안전무결성 수준과 안전무결성 요구사항

추가 안전기능의 필요성을 판단하기 위해 총중간사건발생빈도와 목표완화사건발생빈도를 서로 비교해야 한다. 만약 총중간사건발생빈도가 목표완화사건발생빈도 보다 작다면 위험도가 충분히 낮은 수준이기 때문에 추가적인 안전기능은 필요 없고, 높다면 추가적인 안전기능이 필요하다. 표 III.3.9 의 총중간사건발생빈도 8·10-

^{3}/year 은 목표완화사건발생빈도 $2{\cdot}10^{-5}$/year 보다 크기 때문에 추가적인 안전기능이 필요하다.

추가적인 안전기능으로 가스차단기능을 고려하기 전에, 현재 해당 시스템 및 관련 장치에 존재하는 방호계층을 통해 초기사건발생빈도, 심각도 수준, 중간사건발생빈도를 낮출 수 있는지 먼저 평가했지만, 해당 방호계층은 없는 것으로 식별되었다.

안전무결성 수준은 목표완화사건발생빈도 $2{\cdot}10^{-5}$ 을 총중간사건발생빈도 $8{\cdot}10^{-3}$ 로 나누면 $2.5{\cdot}10^{-3}$ 으로 간단히 계산할 수 있다. 이 값은 가스차단기능의 PFD 가 되며 안전무결성 수준 결정시 사용된다.

IEC61508 의 낮은 수준의 안전무결성 요구는 PFD 수준에 따라 4 단계로 구분된다. 가스차단기능의 안전무결성은 가스차단기능 PFD $2.5{\cdot}10^{-3}$ 가 포함되는 범위인 안전무결성 2 로 결정할 수 있다. 실제 설계될 차단시스템의 PFD 는 여기서 결정된 가스차단기능 PFD $2.5{\cdot}10^{-3}$ 와 안전무결성 2 를 모두 만족시켜야 한다.

표 III.3.9 LOPA worksheet

영향사건 설명	심각도 수준	초기 사건	초기사건발생빈도	방호 계층					중간사건발생빈도	SIF PFD	완화사건발생빈도	목표완화사건발생빈도 (TMEL)	비고
				일반 공정 설계	BPCS	알람, 기타	추가완화 (접근제한)	독립방호, 추가완화 대책(다이크,압력방출)					
엔진룸 화재/폭발로 인명 피해, 장비 손상	4	압력조절배르 고장	4.0E-01	1	1	1.0E-01 알람/운전원	1.0E-01	1	4.0E-03	2.5E-03 정지기능	1.0E-05	2.0E-05	
		가스공급관 누수	4.0E-02	1	1	1	1.0E-01	1	4.0E-03		1.0E-05		
								총계	총 중간사건발생빈도	SIF PFD	완화사건발생빈도	TMEL	SIL 요구사항

		8.0E-03	2.50E-03	2.00E-05	2.00E-05	2

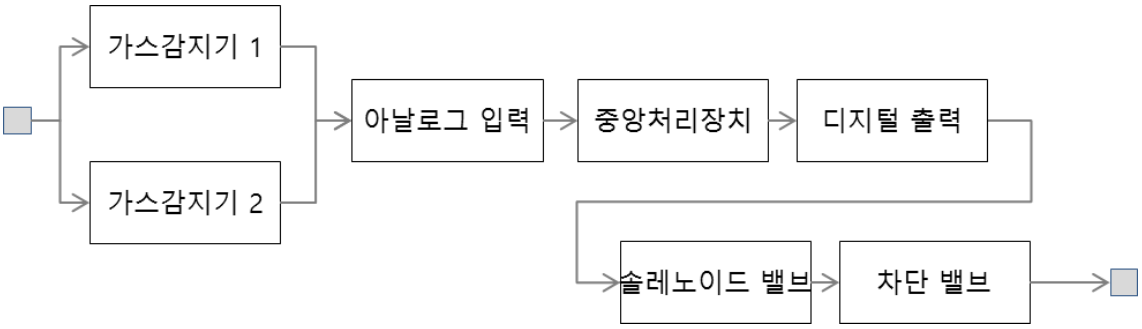
5. 단계 : 안전 요구사항 할당

안전제어시스템의 작동은 낮은 요구모드와 지속/높은 요구모드로 나눌 수 있고, 플랜트 또는 선박에 사용되는 대부분의 안전계장시스템은 낮은 요구모드이다. 낮은 요구모드로 작동하는 안전계장시스템은 제어대상장비가 정상 상태일 때 작동하지 않다가, 제어대상장비에 비정상적인 문제가 발생할 경우 작동한다. 차단시스템도 낮은 요구모드이기 때문에 PFD 에 따라 안전무결성 수준이 결정된다.

차단시스템의 주 기능은 가스밸브실 내부로 누출된 가스를 감지하여 일정 수준이상의 농도가 되면 차단밸브를 닫는 것이다. 가스감지기는 1oo2(1 out of 2) 보팅으로 작동하여 2 개의 센서중 하나 이상 작동하면 가스감지 기능은 성공이다. 솔레노이드밸브는 항상 열린(Normal open) 상태이며, 전기 차단 또는 공압 차단 시 자동으로 차단밸브(Shutdown valve)가 작동(Fail close)하여 안전한 상태를 유지한다.

그림 Ⅲ.3.2 의 신뢰도블록선도(Reliability Block Diagram, RBD)은 차단시스템 작동 성공을 나타낸다. 가스감지기(1oo2, 2 채널 중 1 아키텍처. 두 채널 중 하나가 안전 기능을 수행할 수 있는 경우)는 2 개 중 하나 이상 작동하고, 논리연산자의 구성 부품인 아날로그입력, 중앙처리장치, 디지털출력이 모두 작동하고, 차단밸브의 구성 부품인 솔레노이드밸브, 차단밸브가 모두 작동하면 차단시스템의 기능은 성공한다.

그림 III.3.2.1 Reliability Block Diagram(RBD) for shutdown System



위 신뢰도블록선도에서는 보팅로직을 사용하는 부품은 가스 감지기 뿐이며, 다른 부품은 보팅로직을 사용하지 않는다. 가스감지기의 경우 공통원인고장를 고려하여 PFD 를 계산해야 한다. 아래와 같이 각 부품의 PFD 를 계산할 수 있다.

- $PFD_{GD(CCF)(1oo2)} = 0.06 \cdot 0.6 \cdot 10^{-6} \cdot 8760/2 = 1.58 \cdot 10^{-4}$
- $PFD_{GD(independent)(1oo2)} = [0.6 \cdot 10^{-6} \cdot 8760]2/3 = 9.21 \cdot 10^{-6}$
- $P_{TIF-GD(1oo2)} = 0.06 \cdot 1.0 \cdot 10^{-3} = 6.0 \cdot 10^{-5}$
- $PFD_{AI(independent)} = 0.16 \cdot 10^{-6} \cdot 1460/2 = 1.17 \cdot 10^{-4}$
- $PFD_{CPU(independent)} = 0.48 \cdot 10^{-6} \cdot 1460/2 = 3.50 \cdot 10^{-4}$
- $PFD_{DO(independent)} = 0.16 \cdot 10^{-6} \cdot 1460/2 = 1.17 \cdot 10^{-4}$
- $PFD_{XV(independent)} = 2.1 \cdot 10^{-6} \cdot 1460/2 = 1.5 \cdot 10^{-3}$
- $PFD_{Pilot(independent)} = 0.8 \cdot 10^{-6} \cdot 1460/2 = 5.84 \cdot 10^{-4}$

표 III.3.10 PFD for shutdown system

장치/시스템		PFD (CCF)	PFD (Independent Failure)	Ptif	PFD (CSU)
가스감지기 (1oo2)		$1.58 \cdot 10^{-4}$	$9.21 \cdot 10^{-6}$	$6.0 \cdot 10^{-5}$	$2.27 \cdot 10^{-4}$
제어로직 처리	아날로그 입력	0	$1.17 \cdot 10^{-4}$	N/A	$6.34 \cdot 10^{-4}$
	중앙처리장치	0	$3.50 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	

	디지털 출력	0	$1.17 \cdot 10^{-4}$	N/A	
처리 기기	차단 밸브	0	$1.53 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$2.22 \cdot 10^{-3}$
	솔레노이드 밸브	0	$5.84 \cdot 10^{-4}$	N/A	
총 계		$1.58 \cdot 10^{-4}$	$2.68 \cdot 10^{-3}$	$2.10 \cdot 10^{-4}$	$3.08 \cdot 10^{-3}$

위에 계산된 각 부품의 PFD 를 모두 더하면 차단시스템의 PFD 는 $3.05 \cdot 10^{-3}$ 로 쉽게 구할 수 있다. 차단시스템의 PFD 결과값은 안전무결성 2 수준을 만족시키지만, 방호계층 분석의 안전무결성 수준 $2.5 \cdot 10^{-3}$ 는 만족시키지 못한다.

안전무결성 수준을 만족하기 위하여 안전계장시스템 고장원인 중 가장 큰 비중을 차지하는 차단밸브를 1oo2 보팅으로 변경하고, PFD 를 재계산하여 $2.5 \cdot 10^{-3}$ 를 만족하는지 확인하였다.

그림 III.3.2.2 Reliability Block Diagram(RBD) for revised shutdown System

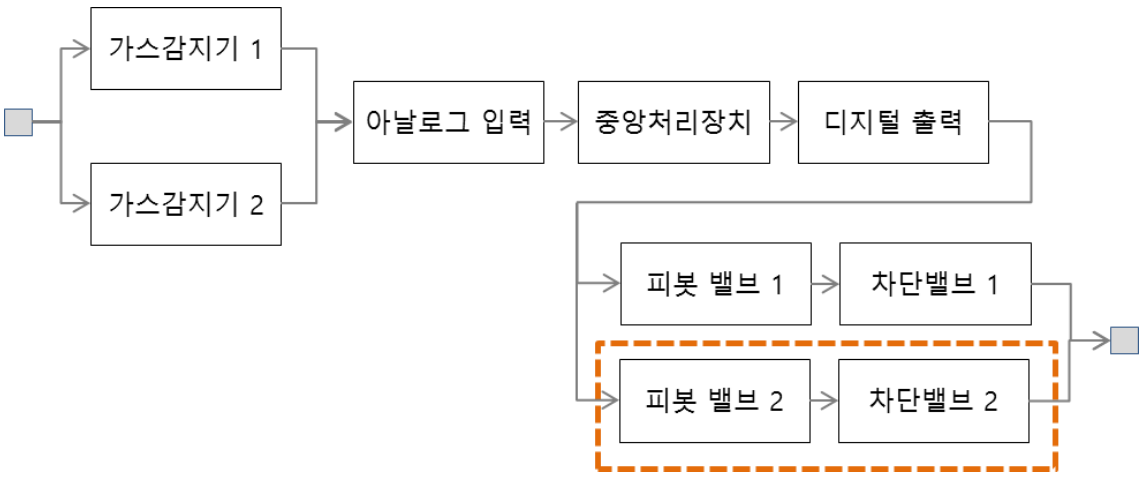


그림 III.3.2 의 신뢰도블록선도는 차단시스템의 작동 성공을 나타낸다. 가스감지기(1oo2)는 2 개 중 하나 이상 작동하고, 논리연산자의 구성 부품인

아날로그입력, 중앙처리장치, 디지털출력이 모두 작동하고, 차단밸브(피봇밸브+차단밸브, 1oo2) 2 개 중 하나 이상 작동하면 차단기능은 성공한다.

- $PFD_{XV(CCF)(1oo2)} = 0.03 \cdot 2.1 \cdot 10^{-6} \cdot 1460/2 = 4.60 \cdot 10^{-5}$
- $PFD_{XV(independent)(1oo2)} = (2.1 \cdot 10^{-6} \cdot 1460)2/3 = 3.13 \cdot 10^{-6}$
- $P_{TIF-XV(1oo2)} = 0.03 \cdot 1 \cdot 10^{-4} = 3.0 \cdot 10^{-6}$
- $PFD_{Pivot(CCF)(1oo2)} = 0.1 \cdot 0.8 \cdot 10^{-6} \cdot 1460/2 = 5.84 \cdot 10^{-5}$
- $PFD_{Pivot(independent)(1oo2)} = (0.8 \cdot 10^{-6} \cdot 1460)2/3 = 4.55 \cdot 10^{-7}$

표 III.3.11 PDF for revised shutdown System

장치/시스템		PFD (CCF)	PFD (independent failure)	Ptif	PFD (CSU)
가스감지기 (1oo2)		$1.58 \cdot 10^{-4}$	$9.21 \cdot 10^{-6}$	$6.0 \cdot 10^{-5}$	$2.27 \cdot 10^{-4}$
제어로직 처리	아날로그 입력	-	$1.17 \cdot 10^{-4}$	-	$6.34 \cdot 10^{-4}$
	중앙처리장치	-	$3.50 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	
	디지털 출력	-	$1.17 \cdot 10^{-4}$	-	
처리 기기	차단 밸브	$4.60 \cdot 10^{-5}$	$3.13 \cdot 10^{-6}$	$3.0 \cdot 10^{-6}$	$1.11 \cdot 10^{-4}$
	솔레노이드 밸브	$5.84 \cdot 10^{-5}$	$4.55 \cdot 10^{-7}$	-	
총 계		$2.62 \cdot 10^{-4}$	$5.97 \cdot 10^{-4}$	$1.13 \cdot 10^{-4}$	$9.72 \cdot 10^{-4}$

수정된 차단시스템의 PFD 결과값은 $9.72 \cdot 10^{-4}$ 으로 방호계층분석의 가스차단기능 PFD $2.5 \cdot 10^{-3}$ 보다 충분히 작기 때문에 수정된 차단시스템은 허용 가능한 위험도 수준을 만족하는 안전계장시스템이라고 할 수 있다.

안전무결성 결정을 위해 사용된 방호계층분석 및 기타 다른 방법들은 안전무결성을 빠르게 결정하기 위한 방법이기 때문에, 많은 불확실성을 내포하고 있다. 항상 입력값의 정확도와 보수적인 판단이 충돌하지만 발주처, 설계, 위험도 평가 관련 사람들이 모여 안전무결성 수준을 결정하는 것이 가장 바람직하다. 또한 결정된 안전무결성 수준은 추후 정량적 위험도 평가를 통해 정확히 검증되어야 한다.

제 4 장. 위험 분석 기법

1. HAZOP (Hazard and Operability)

가. 개요

HAZOP 의 기본 전제는, 제품이 원래 설계된 대로 작동하는 한 근본적인 위험성이나 운용상의 문제는 없다는 것이다. 그러나, 만일 무엇인가의 이유로 인해 이상요인이 발생하게 되면, 제품이 그 충격을 감당할 수 있느냐 없느냐에 따라 사고가 발생한다는 것이다.

이 기법은 영국의 임페리얼 화학공업주식회사 (Imperial Chemical Industries Ltd., ICI) 에서 개발되어 미국 화공기술자협회 화학공정안전센터 (American Institute of Chemical Engineers Center for Chemical Process Safety) 에서 정리된 기법으로서, 여러 전문분야의 구성원들로 이루어진 분석 팀이 조직적으로 난상토론 (brainstorming) 을 하는 과정에서 시스템의 위험원들과 운용상의 문제점을 구명하는 것이다.

방법론과 분석내용으로 보아서는 앞에서 살펴 본 What If 분석과 크게 다르지 않지만, 분석을 연구 절점 (study nodes) 이라고 하는 특정부분에 집중시키며, 지침어 (guide words) 라고 하는 독특한 분석지침을 도입하기 때문에 좀 더 조직적이고 체계적으로 분석을 할 수 있다는 점에 차이가 있다.

HAZOP 의 분석 목적은 다음과 같이 정리할 수 있다.

- 의도된 설계기능이나 조건 등을 포함하여, 제품이나 시스템의 상세한 설명을 제공한다.
- 시스템이나 제품의 모든 부분을 체계적으로 검토함으로써, 설계의도로부터의 일탈이 어떻게 발생하는가를 파악한다.
- 이러한 일탈들이 위험이나 사용상의 문제들을 초래할 수 있는지 결정한다.

HAZOP 에서는 기본적으로, 제품의 기능특성이 제품의 상태변수를 통해 표현되어야 한다. 제품의 상태변수 (process parameters) 란 제품의 기능이나 운용상태를 나타내기 위한 물리적 변수들을 말하는데, 표 Ⅲ.4.1 에서 보는 바와 같이 온도, 습도, 압력, 유속, 반응속도 등이 그 예이다.

표 Ⅲ.4.1 제품의 상태변수 예시

흐름	시간	빈도	혼합
압력	구성	점도	추가
온도	산성도 (pH)	전압	분리
수준	속도	정보	반응

나. 분석절차 및 내용

HAZOP 의 분석 순서는 다음과 같다.

1) 목적 및 분석범위의 설정

분석목적과 범위를 결정한다. 지나치게 많은 분석 대상은 많은 노력을 필요로 하기 때문에, 보통은 위험성이 높다고 판단되는 부분을 선정하여 분석을 집중하는 것이 보통이다.

2) 분석 팀의 구성

분석 팀은 학제간 연구가 가능하여야 한다. 이것은 최초 ICI 사에서 규정한 HAZOP 의 정의에서도 분명히 규정하고 있는 사항이다. 그러므로 분석 팀은 전문가들이 참여하여야 하고, 아울러 관리가능하여야 하기 때문에 대체로 5~7 인 정도로 유지하는 것이 중요하지만, 작은 제품의 경우에는 경험많은 전문가 2~3 인으로도 가능하다.

분석팀의 구성원이 분석에 전념할 수 있도록 하기 위해서는 기록이나 보고서 작성을 위한 인원을 추가할 수 있다.

3) 예비조사

예비작업은 일반적으로 다음과 같이 진행된다.

- 필요한 자료의 획득
- 획득된 자료를 적절한 형태로 전환하거나 분석하기 위한 절차의 수립
- 회의일정의 계획

HAZOP 을 실시하기 위해서는 사전에 대상 제품에 대하여 충분히 숙지하고 있어야 한다. 또한 제품변수를 파악하고 그 결과를 예측하기 위해서는 제품설계도, 사용절차, 공정 흐름도 등 상세한 공정설명과 관계자료를 확보하고 있어야 한다. 이 때 확보되고 검토되어야 하는 자료들은 다음과 같다.

- P&IDs
- 제품설계도
- 제품흐름도
- 선형 다이어그램
- 제품설명
- 사용설명서/절차
- 제품재료정보 및 규격
- 필요시 논리 다이어그램 추가

4) 난상토론의 실시

팀 구성원들이 함께 모여 난상토론을 한다. 난상토론 (brainstorming) 의 주된 장점은 팀 구성원들의 다양한 전문지식과 의견교환을 통하여 창조력과 상상력을 자극하여 다양한 아이디어를 만들어낸다는 점이다.

이 때, 분석 및 검토대상은 보통 연구 절점 (study node) 이라고 부르는 제품변수들의 일탈이 조사될 제품 구조 및 기능상의 위치를 말한다. 일탈 (deviation)

이란 지침어를 체계적으로 적용시킴으로써 의도된 기능으로부터 벗어나는 상황을 가리킨다.

안내 단어 (guide word) 란 HAZOP 만이 가지고 있는 독특한 기법으로, 난상토론을 통해 쉽게 일탈을 발견할 수 있도록, 의도된 기능을 정성화 또는 정량화하는 데 사용되는 간단한 용어들을 말한다. 각 안내 단어는 공정변수들과 적절히 결합될 수 있으며 연구절점, 제품의 일부, 사용단계 등 어떤 대상점에 대해서도 활용될 수 있다. 예를 들어 표 Ⅲ.4.2 과 같은 결합이 가능하다는 뜻이다.

표 Ⅲ.4.2 안내 단어와 일탈

지침어	운용상의 일탈	
없음 (None)	흐름 없음	No Flow
	역류	Reverse Flow
	무반응	No Reaction
과다 (More)	과다 흐름	Increased Flow
	과다 압력	Increased Pressure
	과다 온도	Increased Temperature
	과다 반응률	Increased Reaction Rate
과소 (Less)	과소 흐름	Reduced Flow
	과소 압력	Reduced Pressure
	과소 온도	Reduced Temperature
	과소 반응률	Reduced Reaction Rate
부분적 (Part of)	재료 유입률의 변경	Change of ratio of materials presents
대등한 (As well as)	대체 재료	Different Materials presents
기타 (Other)	비정상 공정조건	Different Plant Conditions from normal operation

공정변수들의 종류와 성격에 따라 많은 안내 단어들이 만들어질 수도 있고, 그렇지 않을 수도 있다. 예를 들어 온도의 경우에는 적정온도 이상 혹은 이하의 두 가지만

있을 뿐이다. 반면 어떤 변수들에 대해서는 이 안내 단어들이 적절하지 않을 수도 있으므로 약간의 수정을 가하여 활용하기도 한다.

5) 분석결과의 기록

난상토론을 통하여 얻어지는 결과는 다음과 같다.

- 발견된 위험원 또는 제품 사용상의 문제
- 안전성 향상을 위해 수행되어야 하는 설계 및 사용절차의 수정사항
- 미진한 분석결과를 보완하기 위하여 수행되어야 하는 심층분석

이 결과는 표 Ⅲ.4.3 과 같은 양식으로 문서화한다.

표 Ⅲ.4.3 분석결과 기록 양식의 예

HAZOP

분석팀 :　　도면번호 :

날 짜 :　　수정번호 :

품목번호	일탈	원인	결과	안전장치	조치

다. 분석사례

표 Ⅲ.4.4 는 What If 분석에서 보았던 DAP (Diammonium phosphate) 반응기에 대한 분석 사례이다. 이미 밝힌 바와 같이 이 공정에서는 인산 (phosphoric acid) 용액과 암모니아 용액이 제어밸브에 의해 통제되는 파이프를 통해 교반기에 투입되며, 이 두 용액은 서로 반응하여 DAP 를 형성하게 된다. 이 때 발생할 수 있는 규정치로부터의 일탈과 그 영향을 평가하는 것이다.

표 III.4.4 HAZOP 분석사례 예시

HAZOP

분석팀 : HAZOP Team #3 도면번호 : 70-0BP-57100

날 짜 : 12 / 27/ 99 수정번호 : # 2

품목번호	일탈	원인	결과	안전장치	조치
1.0 용기 - 암모니아 용액 저장 탱크. 상온 상압에서 암모니아 저장					
1.1	과다 수준	암모니아 저장 탱크내에 적절한 공간이 없는데 투입구에서 암모니아 투입 암모니아 저장 탱크 수준표시기가 과소표시	암모니아의 잠재적 대기방출	저장탱크에 수준표시기 대기로 난 암모니아 저장 탱크의 릴리프밸브	투입전 저장 탱크에 적절한 공간이 있는지 확인하는 암모니아 투임 절차 검토 릴리프밸브 방출기를 세정기로 보내는 방안을 검토 암모니아 저장 탱크에 과다수준 경보기를 독립적으로 추가설치하는 방안 검토
2.0 라인 - DAP 에 암모니아 공급.					
2.1	과다 흐름	암모니아 공급라인의 밸브가 고장으로 열림 흐름표시기가 고장으로 과소표시 운용자가	반응하지 않은 암모니아가 DAP 저장탱크로 운반되어, 결국에는 작업장에 유출됨	밸브의 정기보전 암모니아 감지기 및 경보	반응기에 암모니아 과다유입에 대하여 추가적인 경보/제품 차단 방안 검토 밸브가 적절한지 정기검사와 보전을 확인 적절한 환기가 이루어지고 있는지 확인 밀폐된 DAP 저장 탱크의

		암모니아 유출속도로 과다책정			사용 검토

라. 기타사항

제품은 운용하는 도중에도 여러 가지 이유로 인해 크고 작은 수정이 가해지게 된다. 이러한 제품의 변경은 제품의 기능변화는 물론 동시에 기능장애나 사고의 가능성이 도입될 수 있으므로, 이 기법은 제품을 수정하거나 보정을 하는 경우 권장할 만한 도구로 알려져 널리 활용되고 있다.

HAZOP 분석은 공정흐름 다이어그램 (Process Flow Diagram, PFD), 파이프 및 기기 다이어그램 (Pipe and Instrument Diagram, P&ID) 이 결정된 단계에서 실행되어야 한다. 그러므로 HAZOP 은 설계단계나 운용단계 중 어느 때에도 수행될 수 있지만, 가장 좋은 시기는 제품 설계가 상당히 완성되었을 때이다. 새로운 제품에 대해서는 설계가 거의 완성된 시점에서 실시하는 것이 좋고, 기존공정에 대해서는 재설계가 계획되는 단계에서 실시하는 것이 좋다.

HAZOP 분석을 지원하는 도구는 다음과 같다.

- CAHAZOP (NUS Corp., San Diego, California)
- HAZOP-PC (Primatech, Inc., Columbus, Ohio)
- HAZOPtimizer (A.D.Little, Cambridge, Massachusetts)
- HAZSEC (Technica, Inc., Columbus, Ohio)
- HAZTEK (Westinghouse Electric Corp., Pittusburgh, Pennsylvania)
- LEADER (JBF Associates, Inc., Knoxville, Tennessee)
- SAFEPLAN (Du Pont, Westlake Village, California)

2. FMEA (Failure Modes and Effects Analysis)

가. 개요

FMEA 는 What If 분석을 좀 더 체계화한 것이다. 즉, 만약 무슨 일이 벌어진다면 어떻게 될까 (what happens if) ?" 라는 질문을 염두에 두어 하나의 재료, 부품, 장비 등이 고장났을 경우 그것이 전체 제품이나 사용자, 혹은 제품기능에 어떠한 영향을 미치는가, 생각의 범위를 점차 넓혀가면서 상위수준으로 분석하여 가는 것이다.

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

IEC 812 는 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, 정의단계에서 시행될 수는 있으나 제품의 구성과 기능에 관계된 구체적이고도 많은 자료를 필요로 하기 때문에 그 효과가 한정적이어서, 주로 제품 설계단계와 개발단계에서 이루어지는 것이 일반적이다.

나. 분석절차 및 내용

FMEA 분석절차는 대체로 아래와 같은 과정을 따른다.

1. 제품과, 그 기능적 최소 운용요구조건의 정의
2. 기능적 신뢰도 블록 다이어그램, 기타 다이어그램 또는 수학적 모형의 개발과 설명
3. 분석 수행상의 기본원칙과 그에 상응하는 문서양식의 설정
4. 고장모드, 그들의 원인과 영향, 상대적인 중요성, 그리고 그 연쇄들의 구명
5. 고장검출과 격리규정 및 방법의 구명
6. 특히 바람직하지 않은 사건에 대한 설계 및 운용규정의 구명
7. 사건 치명도 (event criticality) 의 결정 *
8. 고장확률의 평가 *
9. 필요한 경우, 고려되어야 하는 특정조합의 다중고장에 대한 탐색
10. 권장사항

구체적인 분석사항은 다음과 같다.

1) 분석과제 정의 및 분석준비

가장 먼저 이루어져야 할 것은, FMEA 에 포함되어야 할 구체적인 항목과, 그것들이 분석되어야 할 조건들을 규정하는 일로서, 적절한 분석수준과 분석의 경계 조건들을 정의하는 것을 말한다.

다음의 절차를 수행한다.

1. 제품을 효율적으로 다룰 수 있게 어셈블리로 분할한다.
2. 제품과 어셈블리의 기능 다이어그램, 구성도, 도면 등을 검토하여 그들의 연관관계를 결정한다.
3. 분석되어야 할 각 어셈블리에 대하여 완벽한 구성부품목록을 준비한다.

2) 분석의 실시

분석 수행시 가장 중요한 것은, 치명적인 상호작용과 숨겨져 있는 제품 설계상의 상호작용을 도출하기 위하여 분석자가 부품의 구조와 기능에 관하여 충분한 지식을 가지고 있어야 한다는 점이다.

FMEA 는 결함수 분석 (Fault Tree Analysis) 과 같이 제품 구성요소간의 상세한 기능적 연관관계나 종속성에 관한 정보를 제공하지는 않으므로 이 부족한 상세사항은 분석팀의 경험과 지식으로 보충되어야 한다. 분야 전문가를 포함한 여러 분야의 전문가들의 협력에 의하여 학제간 연구 (multidisciplinary study) 로 이루어져야 한다.

그러므로, 최소한 분석 팀에는 다음의 전문가가 반드시 참여하여야 한다.

- 제품의 설계와 운용을 잘 아는 제품 공학자나 사용 전문가
- 전기적 제어설계, 논리, 사용장비 등을 잘 아는 제어 전문가

직접 수행되는 구체적인 분석내용은 다음과 같다.

- 제품에 영향을 미치는 운용상의 또는 환경적인 스트레스를 설정한다.
- 학적 도면이나 기능 다이어그램의 분석으로부터 구성요소에 영향을 미칠 수 있는 중요한 고장 메커니즘을 결정한다.
- 모든 구성부품의 고장모드를 판명한다.
- 운용, 스트레스, 인적반응조치, 사건들의 조합에 있어서 고장이나 손상의 가능성을 증가시키는 특별한 기간이 있는지, 구성부품에 영향을 미치는 각 조건들을 나열한다.
- 위험성 범주를 평가한다.
- 위험성을 제거하거나 최소화하기 위한 예방대책 또는 사후대책을 나열한다.

작업의 결과는 표 Ⅲ.4.4 과 같은 형태로 문서화한다.

표 III.4.5 FMEA 분석결과 문서화 예시

고장모드 및 영향분석									
날 짜 :					페이지 : / 쪽				
제품명 :					분석자 :				
항목 번호	장비특성기능	고장모드	예상원인	고장의 영향		검출법	시정조치	치명도	비 고
				하부 시스템	제품				

- 항목번호는 도면이나 블록 다이어그램에서 식별할 수 있는 식별기호를 나타낸다.
- 장비특성에는 장비유형, 운용모드, 기타 고장모드와 효과에 영향을 미칠 수 있는 기능 특성, 예를 들어 고온, 고압, 부식 등을 기입한다.
- 고장모드에는 장비특성과 관련된 부품의 모든 고장모드를 나열한다. 이 때 각각의 고장은 제품 내의 다른 고장들과 관계없이 독립적으로 발생한다고 가정한다.
- 고장원인을 나열하는 이유는, 고장 시나리오를 함축적으로 서술함으로써 고장의 성질을 명확히 하기 위한 것이다.

고장영향에는 고장위치에 미치는 부품고장의 직접적인 효과와, 다른 장비나 전체 제품에 영향을 줄 것으로 예상되는 결과를 모두 기입한다. 특히 제품의 사고를 예방하기 위해서는, 제품 설계내에 존재하는 안전방호장치가 정상적으로 작동하지 않는다고 가정하고, 그 결과 예견될 수 있는 최악의 상황을 염두에 두고 영향을 평가한다.

고장검출방법은 고장모드를 어떻게 검출할 것인가 하는 방법을 말한다.

시정조치에는 해당 고장모드와 관련된 효과의 발생 가능성을 감소시킬 수 있는 모든 사후조치를 나열한다. 만약 제품의 자동제어기능이 별다른 손실없이 고장의 영향을 흡수해 버릴 수 있다면 이 사실도 기입되어야 한다.

FMEA 에서는, 피해규모의 경중에 차이없이 제품 기능 상실을 초래하는 고장들의 위중함이 모두 같다고 가정되지만, 실제로 고장이 제품에 미치는 영향은 서로 다르기 때문에 집중적으로 관리할 필요성이 대두되기 때문에, 이를 위해 제품 기능에 미치는 치명성을 언급하는 공란을 추가하거나 비고란을 추가하여 이에 대한 사항을 기록한다. 표 Ⅲ.4.4 의 치명도는 바로 이러한 목적을 위해 이용된다.

3) 분석결과의 정리 및 심층 분석

이상의 분석이 끝나면 해당 구성요소의 고장이 각 제품 수준에 미치는 대략적인 영향을 파악할 수 있다. 그러나 좀 더 정확하고 비교가 가능한 척도를 얻기 위하여 고장확률을 계산할 수 있다.

구체적인 내용은 다음과 같다.

- 각 구성부품의 고장발생확률을 기입한다.
- 하부 어셈블리, 어셈블리, 제품의 고장확률을 계산한다.
- 구성부품의 치명도를 계산하는 등 분석을 계속하고, 고장이 임무수행에 가져올 수 있는 영향을 분석한다.

다. 분석사례

표 Ⅲ.4.5 는 가정용 커피분쇄기에 대한 FMEA 사례이다.

표 Ⅲ.4.6 FMEA 분석사례

고장모드 및 영향분석 제품: 가정용 전동

커피분쇄기

부품 명칭/ 번호	기능	고장모드 및 원인	고장영향		고장 확률 (10-6)	사후 또는 권장조치
			다음 상위품목	최종품목/제품		
뚜껑	커피가 쏟아지는 것을 막는다. 회전칼날에 의해 다치지 않도록 사용자가 컵속에 손가락을 집어넣지 못하게 한다.	플라스틱 파편들과 부품들이 분리된다. 단단한 표면에 부서지기 쉬운 플라스틱이 낙하한다. 실제 사용시 밟히거나 지나치게 과다한 힘의 영향을 받는다.	없음	없음	1	깨지지 않는 플라스틱을 선택
스위치 작동레버암	분쇄기를 작동시키려면 스위치 작동레버암의 개방단(free end) 을 금접구멍까지 내리눌러 그 상태를 유지하여야 한다.	사용자의 거친 취급, 밟힘, 또는 낙하에 의한 파손이 발생한다.	뚜껑을 약하게 할 우려가 있음. 또한 만약 작동레버암이 뚜껑 부위에서 파손되는 경우에는 파손됨	제품을 불안전하게 만들 소지가 있다.	100	재디자인. 스위치를 뚜껑밑에 위치시킴으로써 암을 제거
플라스틱 케이스	다른 조립부품들을 유지하는	충격이나 충돌에 의하여 파손될 수	-	결과적으로 날카로운 단들과	0.5	내충격 플라스틱 사용

	주요구성부품. 이동하는 물체나 전기적 부품과의 접촉으로부터 제품을 보호한다.	있다.		위험점들을 만들 수 있다.		
진동 완충기 (2)	케이스 속에 장착된 고무 패드. 플라스틱 케이스로부터 금속 모터 프레임을 격리시킴으로써 진동과 소음이 저감된다.	고무의 기능열화. 제 위치에 고정되어 있지 않기 때문에 기능이 상실될 수 있다.	파손되기 쉬운 플라스틱의 피로	파손되기 쉬운 플라스틱에 대한 피로	0.01	접착제 사용
순간 스위치	모터에 전력을 공급하기 위한 회로를 구성한다.	개방고장 : 누름 버튼이 파손된다. 외부 연결이 분리된다. 내부 전기경로가 파손된다. 폐쇄고장 : 내부의 용접 접촉부위. 내부 스프링 파손된다.	모터의 작동실패 모터가 상시 작동	모터의 작동실패 뚜껑이 벗겨진 채 작동	35	고신뢰도 부품 사용 고신뢰도 부품 사용 직렬구조인 두 스위치

		성능열화 : 내부 스프링이 취약성에 기인한 이상 작동이 발생한다.	간헐 작동	간헐 작동		고신뢰도 부품 사용

제품을 구성하는 부품들의 명칭을 나열하고 기능을 서술한 후, 그 고장이 발생하는 원인과 고장모드를 기입하며, 마지막에는 대책을 제시한다. 그리고 필요하다면 표에서 보는 바와 같이 발생확률란을 추가한다.

라. 기타사항

FMEA 는 체계적으로 인과관계를 구명함으로써, 치명적일 수 있는 고장모드, 특히 전파될 수 있는 단일결함의 최초 징후를 제공한다는 점이 가장 큰 장점이며, 또한 비전문가도 이해하기 쉽고, 다른 복잡한 기법들, 예를 들어 FTA 보다 시간이 적게 걸린다는 장점도 무시할 수 없다.

FMEA 의 분석결과는 제품의 운용수명을 증가시키기 위하여 부품과 설계의 어느 부분이 개선되어야 하는가를 결정하는 데 아주 긴요하게 활용된다. IEC 812 는 아래와 같이 이 기법의 장점을 정리하고 있다.

1. 개별적으로 발생하였을 때 수용할 수 없거나 중요한 영향을 미치는 고장을 구명하고, 기대되거나 요구되는 운용에 심각한 영향을 미칠 수 있는 고장모드를 결정한다.
2. 다음과 같은 사항에 대해 필요성을 결정한다.
- 중복성
- 고장발생시 고장나도 안전 (fail-safe) 한 결과의 발생확률을 증가시키는 설계구조
- 추가적인 감률 (derating) 이나 설계 단순화
3. 대안적인 재료, 부품, 장치, 구성요소들에 대한 선택 필요성을 결정한다.
4. 심각한 고장결과를 구명()하고, 설계 심사나 수정의 필요성을 결정한다.
5. 제품의 이례적인 운용조건의 확률을 평가하는 데 필요한 논리 모형을 제공한다.
6. 안전상의 위험성 및 책임영역, 또는 법규상의 비준수사항을 노출시킨다.
7. 시험 프로그램이 잠재적인 고장모드를 검출하는지 확인한다.
8. 마모고장을 예상하고 회피하는 가동주기를 설정한다.
9. 품질, 검사, 제조공정관리를 집중시킬 주요 영역을 찾아낸다.
10. 설계결함의 조기 구명에 의해 값비싼 수정을 회피한다.
11. 자료기록과 감시의 필요성을 결정한다.
12. 예방보전이나 사후보전점을 선택하거나, 문제해결 방안, 내장 시험장비, 적절한 시험점 등의 개발에 관한 정보를 제공한다.
13. 예를 들어 성능시험이나 신뢰도 시험과 같은 시험기준, 시험계획, 진단절차의 결정을 촉진하거나 지원한다.
14. 종종 변수편차와 관련된 고장모드에 대해 요구되는 최악상황분석 (worst case analysis) 이 필요한 회로를 구명한다.
15. 결함격리순서의 설계를 지원하고, 운용모드의 대안과 구조 재편계획을 지원한다.
16. 다음과 같은 정보교환을 촉진한다.
- 일반기사와 전문기사
- 장비 제조자와 공급자
- 제품 사용자와 설계자, 또는 제조자
17. 분석자의 지식과 연구분석 대상장비의 거동에 대한 이해를 향상시킨다.
18. 제품 설비의 연구분석에 대한 체계적이고 정밀한 접근방법을 제공한다.

주요한 단점으로는 다음과 같은 것들이 있다.

- 부품수가 많아지거나, 고장의 영향이 크면 많은 시간과 노력이 소요된다.

- 원인과 결과 사이에 직접적인 연관관계가 없으면 복잡하고 방대하여 감당할 수 없게 되는 경우도 있다.
- 제품 내 상이한 부품들에 발생하는 결함들간의 다중 종속성이나 복잡한 상호작용을 고려하기에도 곤란하다.
- 제품에 미치는 영향을 정량화하기 위하여 균일한 근거를 제공할 수 있을 만큼 상세하지 못하다.
- 통상 인간과오나 환경적인 영향 등 공통모드고장 (common mode failure) 또는 공통원인고장 (common cause failure) 의 원인을 고려할 수 없다.

이런 점과 관련하여, 특히 제품 사용시의 위험성에 미치는 결과를 분석하기 위하여, 아예 FMEA 를 인간의 과오나 불안전행동의 분석에 활용하는 것을 오용 모드 및 영향분석 (Misuse Mode & Effects Analysis, MMEA) 또는 인적과실 모드 및 영향분석 (Human Error Mode & Effects Analysis, HEMEA) 이라고 한다.

표 Ⅲ.4.5 는 자동차용 어린이 보호장치에 대한 MMEA 의 예로서, 발생율과 재해강도가 0 부터 10 까지의 평점으로 평가되는 한편, 위험 우선수 (Risk Priority Number ; RPN) 라는 척도는 개별적인 발생율 평점과 강도 평점의 곱으로 얻어진다.

표 Ⅲ.4.7 MMEA 사례

오용모드 및 영향분석

제 품 명 : 자동차용 어린이 보호장치　　번호 : CA82-35E

공급자 성명 : 김이박 용지 : -3257

부품/기능	잠재적 오용모드	오용의 영향	오용의 원인	최초 평가			취해진 조치	재평가			요구되는 후속조치
				발생	강도	RPN		발생	강도	RPN	
1) 잠금기구	잠금장치가 작동되지	좌석외 형이 세워진 자세에	잠금기구의 자동 작동	7	8	56	스프링이 장착된 잠금 기구	0	0	0	없음

	않는다	서 눕혀진 자세로 넘어질 수 있다	부재								
2) 자동차 안전띠의 조임쇠와 틀의 접촉 부분	조임쇠가 틀위에 놓여질 수 있다.	충돌시 조임쇠가 파괴될 수 있다.	틀이 좌석 바닥과 등받이의 교차부분에 너무 가깝다.	7	9	63	교차선으로부터 거리가 15 mm 가 되도록 틀의 재설계가 요구된다.	0	0	0	없음

발생 가능성 강도 (안전에 대한 영향) 날짜 :
오용 없음 = 0 안전에 영향 없음 = 0
오용이 있을 것 같지 않음 = 1 주목할 만한 영향 없음 = 1
비교적 오용이 거의 없음 = 2 - 3 중요하지 않은 고장 = 2 - 3
가끔 오용 = 4 - 6 보통의 고장 = 4 - 6
반복되는 오용 = 7 - 8 심각한 고장 = 7 - 8
거의 불가피한 오용 = 9 - 10 매우 심각한 고장 = 9 - 10 서명 :

결과적으로 이 기법은 부품의 수가 증가함에 따라 많은 노력이 든다는 단점은 있지만, 특별한 교육을 받지 않고도 제품의 위험성을 분석할 수 있다는 점에서 널리 활용되고 있다.

한편, 형식상으로는 대체로 이미 앞에서 보아 왔던 위험성 분석기법들, 즉 PHA, FHA, SHA 등과 비슷하기 때문에, 하부 시스템 위험성 분석 (Subsystem Hazard Analysis; SSHA) 이나 시스템 위험성 분석 (System Hazard Analysis; SHA) 에도 많이 이용된다.

치명도 분석 (Criticality Analysis, CA) 과 고장모드, 영향, 및 치명도 분석 (Failure Mode, Effect, and Criticality Analysis, FMECA) 를 합성으로 이용되기도 한다. 이는 고장 간격이나 확률을 추정하는 방법으로 항공분야에서 보전주기 (maintenance interval) 와 요구사항을 결정하는 데 널리 이용되고 있다.

분석결과는, 제품의 구체적인 설계안이 확정되고 관련자료가 수정될 때마다 최신정보를 활용하여 제품 특성에 맞게 지속적으로 수정되어야 한다.

분석경험을 지원하기 위하여 현재까지 개발되어 소개된 소프트웨어로는 CARA (Technica, Inc., Columbus, Ohio), FMEA-PC (Primatech, Inc., Columbus, Ohio), HAZOPtimizer (A.D.Little, Cambridge, Massachusetts), SAFEPLAN (Du Pont, Westlake Village, California) 등이 있다.

3. FTA (Fault Tree Analysis)

가. 개요

FTA 는 결함수 분석, 결함수목분석, 고장수목분석 등 여러 가지 용어로 번역되어 사용된다. 고장 (failure) 이란 해당 부품이나 제품이 다시 정상적으로 작동하기 전에 수리를 요하는 기능장애를 말하고, 결함 (fault) 이란 일단 기능장애를 야기시킨 조건이 교정되면 저절로 치유될 수 있는 기능장애를 말한다. 이 차이를 고려하면 제품의 안전성을 향상시키기 위해 조사하고 분석해야 할 대상은 고장뿐만 아니라 결함도 망라하여야 하므로, 고장수목분석 이 아니라 결함수분석 또는 결함수목분석 이라고 부르는 것이 바람직하다.

원래 이 기법은 미국의 미닛트 맨 (Minute Man) 미사일 발사 시스템을 개발하던 도중 1962 년 벨 (Bell) 전화 연구소의 왓트슨 (Watson) 이라는 사람에 의해 제안되었는데, 초기의 개발 목적은 지금도 충족되고 있다.

- 정상 사건(top event)을 초래하는 원인이나 원인들 조합의 구명
- 특정한 제품 신뢰성 척도가 서술된 요구사항을 충족시키는가 여부의 결정
- 제품의 독립성과 고장의 비관련성에 대하여, 다른 분석들에서 이루어진 가정들이 위배되지 않는가에 대한 결정
- 특정 신뢰도 척도에 가장 심각한 영향을 미치는 요인들과 그 척도를 개선하기 위하여 필요한 변경들의 결정
- 공통적 사상이나 공통원인고장의 구명

결함수란 ETA 의 나무모양의 구조를 말하는데, 다른 점이 있다면 ETA 의 나무는 촉발사상으로부터 시작되어 제품 상태를 나타내는 결과로 발전하여 가는 귀납적 구조였지만, FTA 의 나무는 정상 사상 (top event) 이라고 부르는 바람직하지 않은 사상을 시작으로 그 발생원인이나 거기에 기여하는 조건들이나 요인들을 찾아 시간적 흐름을 거슬러 분석해 가는 연역적 구조라는 점이다. 또한, 정성적인 분석과 정량적인 분석이 모두 가능하고, 제품 구성수준측면에서 보면 하향성 분석방법 (top-

down approach) 이며, 수학적 논리는 부울 대수 (Boolean Algebra) 에 의해 지원되고 있다.

나. 분석절차 및 내용

FTA 의 분석절차는 분석 목적이나 분석 수준에 따라 다르지만 일반적으로 다음과 같은 순서에 따라 진행된다. 즉, 분석범위의 정의 및 분석수준의 결정, 대상 제품의 특성파악, 정상사상의 설정, 결함수의 구성, 결함수의 정성적 분석, 결함수의 정량적 분석, 분석결과의 평가 및 보고라는 단계를 거치는 것이다.

1) 분석범위의 정의 및 분석수준의 결정

분석되어야 할 제품, 분석목적과 범위, 제품 운용상의 초기조건들과 현재의 조건, 그리고 기본적인 가정사항들을 정의한다. 이 때 기본적인 가정들이란 모든 사용조건 하에서의 제품 성능뿐만 아니라, 예상되는 운용조건들과 보전조건들과 관련된 모든 가정사항들을 포함한다.

2) 대상제품의 특성 파악

결함수 분석이 성공적으로 이루어지려면 분석자가 제품을 상세히 숙지하고 있어야 한다. 이 때 필요한 지식이란 생산공정의 구성, 기능, 작동 및 작업방법이나 동작 등 현장정보에 의한 제품의 안전보건상의 문제점을 파악하는 데 관계되는 지식들을 말한다. 또한 여기에는 부품, 구성품들의 고장률 자료와 신뢰도 구조를 입수하여, 신뢰도 블록 다이어그램 분석 (Reliability Block Diagram Analysis) 을 행하는 것도 포함되며, 인간과오의 요인, 형태, 발생확률 등의 자료도 확보되어야 한다.

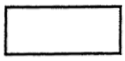
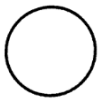
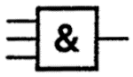
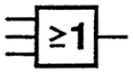
3) 정상사상의 정의

작업자의 과오나 기계설비로 인한 경과를 모형화하고 대책을 설정할 문제점들에 대해 중요도나 우선순위를 결정하여 분석할 대상이 되는 사항을 정상사상으로 선정한다. 이 때 분석대상이 되는 정상사상은 분석목적과 수준, 범위에 맞게 설정되어야 한다.

4) 결함수의 구성

정해진 기호를 이용하여 재해사고의 발생과 관련된 요인들간의 논리적인 관계를 나무모양으로 구성한다. 사용기호에는 표 Ⅲ.4.6 과 같이 여러 가지가 있으나, 분석자의 편의에 따라 첨삭이나 수정이 가능하다.

표 Ⅲ.4.8 FTA 사용 기호

기호	기능	설명
	사상 설명 블록	사상 명칭이나 설명, 사상 코드 그리고 (필요시) 발생 확률. 이 기호 안에 포함되어야 한다
	기본 사상	세분화될 수 없는 사상
	AND 게이트	모든 입력 사상들이 동시에 발생할 때에만 사상이 발생한다
	OR 게이트	하나든 조합이든 어떤 입력사상이라도 발생하면 사상이 발생한다
	전입	결함 수목 내의 다른 곳에서 정의되는 사상

수목을 구성하는 방법으로는, 우선 분석하려고 하는 제품 전체의 고장이라든가 결함과 같은 바람직하지 않은 사상을 정상사상으로 채택하여 최상단에 직사각형을 그리고 그 안에 내용을 기입한다. 이것이 결함수 최정상의 출력사상, 다시 말해 제 1 수준의 출력사항이라고 할 수 있다.

다음에는 정상사상의 하단에 그 재해의 직접원인이 되는 기계 등의 불안전 상태나 작업자의 과오인 결함사상들, 다시 말해 논리 게이트의 입력사상들을 나열한다. 이것은 제 2 수준에 해당한다. 그리고 나서, 입력사상들과 출력사상과의 관계를 고려하여 제 1 수준의 정상사상과 제 2 수준의 기초사상들과를 논리 게이트로 연결한다.

다음 단계에서는 반복적으로 제 2 수준의 결함사상들을 각각 하나의 중간사상으로 하여, 그것들의 직접원인이 되는 결함사상들을 각각 제 3 수준에 쓰고, 제 2 수준의 사상들과 관계를 고려하여 다시 제 2 수준의 사상들과 제 3 수준들과의 사이를 논리 게이트로 연결한다.

이와 같은 과정을 반복해서 위에서부터 아래로, 차례대로 써 나가 최종적으로 모든 나뭇가지의 끝이 모두 다음 중 어느 하나에 해당되면 분지(, branching) 작업을 종료한다.

- 통상사상
- 기본사상
- 미개발 (생략) 사상
- 전이기호

5) 결함수목의 정성적 분석

결함수의 정성적 분석이란 정량적인 변수들을 이용하지 않고 제품의 구조적 특성이나 각 기본사상들이 정상사상의 발생에 미치는 상대적 중요도 등을 평가하는 분석을 말한다.

이 분석은 다음과 같이 3 단계로 나뉘어진다.

- 결함수의 타당성 조사
- 결함수의 축약
- 절단집합과 경로집합

이 중 가장 중요한 것은 최소절단집합 (minimal cut sets) 의 도출이다. 최소절단집합이란, 정상사상, 즉 원하지 않는 재해사고가 발생하기 위해 동시에 발생하여야 하는 최소한의 기본사상들의집합을 말하는데, 이것은 이후 정량적인 분석을 진행해 나가는 데에도 매우 중요하다.

6) 결함수목의 정량적 분석

사상 발생확률의 평가

결함수에 대한 정량적 분석의 최대 장점 중의 하나로서, 각 기본사상들의 발생확률만 알고 있다면 몇 가지 가정사항들을 추가함으로써 중간사상들이나 정상사상의 발생확률을 계산할 수 있다.

중요도 지수

중요도란 어떤 기본사상의 발생이 정상사상의 발생에 어느 정도의 영향을 미치는가를 정량적으로 나타낸 것으로서, 재해예방책 선정의 우선 순위를 제시한다. 여기에는 구조중요도, 확률중요도, 치명중요도 등 여러 가지가 있으며, 이 척도들을 이용하면 재해사고의 예방을 위하여 어느 사상부터, 혹은 어떤 부품부터 개선하여야 할 것인가를 결정할 수 있다.

평균고장률과 평균수리시간

만약 제품이 수리가능한 제품이라면, 하위수준의 기본사상이나 중간사상들의 평균고장률 λ와 평균수리시간 τ로부터 정상사상으로 인한 제품의 평균고장률이나 평균수리시간도 쉽게 구할 수 있다.

기타 신뢰도척도의 추정

이 외에도 해당 제품의 여러 가지 신뢰도 특성이나 척도들을 추정할 수 있다. 이런 경우에 매우 유용한 것이 최소절단집합이다.

7) 분석결과의 평가 및 보고

이상의 과정을 거쳐 정성적 분석 또는 정량적 분석이 종료되면, 최종적인 보고서를 준비한다. 보고서에 포함되어야 하는 기본적인 사항들은 다음과 같이 정리될 수 있으며, 분석목적과 상황에 따라 약간씩 차이가 있을 수 있다.

- 목적과 범위
- 제품 설명
  - 설계 설명

  - 제품 운용
  - 상세한 제품 경계정의
- 가정사항
  - 제품 설계 가정
  - 운용, 보전, 시험 및 검사의 가정
  - 신뢰도 및 가용도 모형의 가정
- 제품 결함의 정의 및 기준
- 결함수 분석
  - 분석내용
  - 자료
  - 사용기호
- 결과 및 결론

이 이외에 결과 보고서에 추가적으로 포함될 수 있는 자료는 아래와 같다.

- 제품 블록/회로 다이어그램
- 사용된 신뢰도 및 보전도 자료의 요약
- 컴퓨터 입력 양식의 결함수 표현

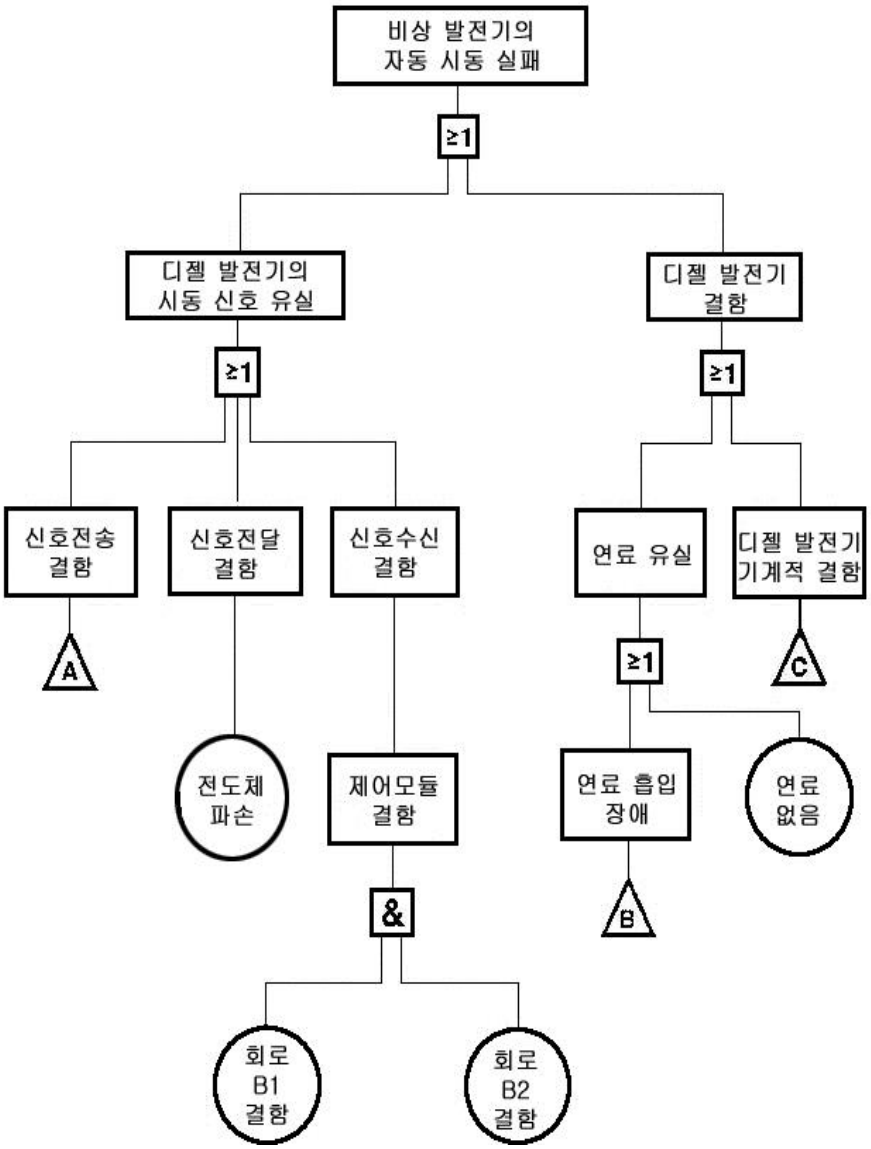
끝으로 이렇게 분석보고까지 끝나면 효과적인 개선안을 검토한다. 최소절단집합은 재해를 예방하기 위한 노력이 어느 곳에 집중되어야 하는가를 나타내기는 하지만 실제적인 비용, 시간, 기술적 이유 등 여러 가지 면에서 제약이 있을 수 있으므로, 여러 가지 중요도 지수와 경제성, 보수성 등을 고려하고, 또 절충연구 (trade-off study) 의 결과와도 비교하여 대책을 수립한다.

다. 분석 사례

그림 Ⅲ.3.3 은 비상발전기의 자동시동이 실패한 경우를 분석하기 위하여 구성된 결함수목의 예이다. 분석은 이와 같은 결함수목을 근거로 중복사상의 제거,

최소절단집합의 획득, 각 사상들의 발생확률 도출 등의 순서로 진행되지만, 계산상의 과정은 상당히 복잡하여 지원도구가 필수적이다.

그림 III.3.3 FTA 분석결과 예시



라. 기타사항

결함수목분석은 여러 가지 하부 시스템으로 구성된 복잡한 제품을 분석하는 데 적합하다. 또한 부품만이 아니라 인간과오, 소프트웨어 과오, 환경 스트레스에 의한 고장 등 다중고장의 해석이 가능하다. 그러나, 논리적인 분석에 있어서 부울 대수나 최소절단집합, 또는 중요도지수를 이용하므로 일반인이 이해하기 어려운 부분들이 있으며, 시간적 연쇄를 취급하거나, 공통원인고장 등을 고려한 역동적인 분석은 곤란하다.

4. ETA (Event Tree Analysis)

가. 개요

만약 제품을 사용하는 중에 발생할 수 있는 여러 가지 상황들을 다 그려볼 수만 있다면 재해사고예방은 좀 더 효과적일 것이다. 바로 이런 목적을 위해 개발된 것이 사상수목분석이다. 즉, 이 기법은 의사결정수목 (Decision Tree) 의 원리를 이용, 재해사고의 발생과정을 재해요인들의 연쇄로 파악하여, 재해발생의 초기사상 혹은 촉발사상 (initiating event) 으로부터 재해사고까지의 연쇄적 전개를 나뭇가지 형태로 표현하는 귀납적인 제품 안전성 분석기법이다.

더욱이 각 재해발생요인들의 발생확률을 알고 있다면, 정성적인 분석기법인 동시에 정량적인 분석기법의 장점도 활용할 수 있다.

나. 분석절차 및 내용

어떤 사고에든 여러 가지 재해발생요인들이 연관되어 있다. 이 요인들을 도표 상단에 왼쪽에서부터 오른쪽으로 차례대로 나열한다. 이 때 가장 왼쪽의 요인은 제품에 고장이나 사고가 발생하게 되는 부정적인 사상, 다시 말해 사고의 촉발사상을 기입하는 것이 보통이고, 오른쪽 끝은 제품 구성요소의 상태조합에 의한 결과상황들이 나열되는 것이니까, 그 중간의 재해요인들은 가급적 시간경과에 따라 재해사고가 전파되거나 혹은 확산되는 데 관계되는 요인들을 나열하도록 한다.

재해촉발사상이 결정되었으면 그 점에서 다음 요소의 발생사상에 따라 가지를 나눈다. 이 때 성공사상, 다시 말해 제품 구성요소가 정상적으로 작동하는 경우를 맨 윗가지에, 정상적으로 작동하지 못하는 고장상태를 맨 아래 가지에 할당한다. 필요하다면 다양한 고장양식에 따라 그 중간에 여러 개의 가지를 더 만들 수 있다. 그 다음 단계에서는 벋어진 가지의 끝 점에서, 또 다시 다음 재해발생요소의 성공,

실패에 따라 가지를 나누어 간다. 이렇게 하여 결과상황까지 벋어 나가면, 제품에 발생할 수 있는 모든 상황들이 오른쪽 가지 끝에 나열되게 되며, 각 결과상황들은 상호 배반적이라는 통계적 성질을 유지하게 된다.

그림 Ⅲ.3.4 는 이렇게 하여 만들어진 기본적인 사상수목을 보여주고 있다. 초기사상 즉 촉발사상에 해당하는 것은 파이프 파단이며, 재해요인들이 5 개이니까 최종적인 결과사상은 $2^{(5-1)}=2^4=16$ 가지가 나열된다. 일반적으로 재해요인들이 n 개라면 결과사상들의 가지수는 $2^{(n-1)}$가지이다.

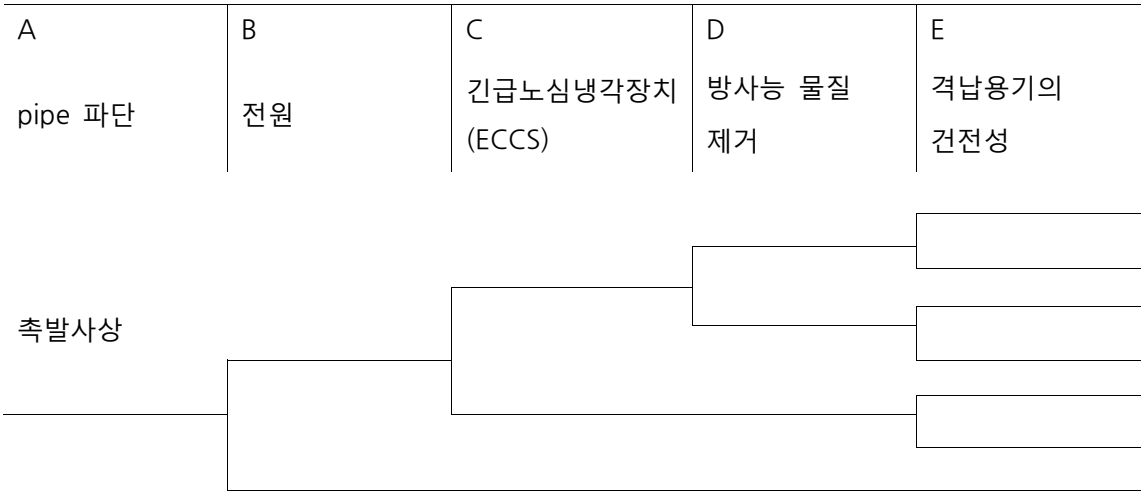
그림 Ⅲ.3.4 ETA 사상수목 예(원자력 분야)

그러나 경우에 따라서는 이렇게 많은 가지들을 각각 모두 고려할 필요가 없을 수 있다. 왜냐하면 어떤 요인의 성공 혹은 고장여부는 이후 다른 요인들의 성패에 관계없이 제품의 상태를 결정해버리기 때문이다.

예를 들어 위의 그림에서 전원의 공급이 이루어지지 않으면 그 이후의 긴급노심냉각장치 (ECCS), 방사능제거, 격납용기의 건전성 등에 관한 분석은 하나마나이다. 그러므로 이런 경우에는 전원공급실패 이후에는 가지를 분지하지 않고

그냥 하나의 가지로 남도록 가지를 잘라 버린다. 이 작업을 전지 작업이라고 하며, 그 결과는 그림 Ⅲ.3.5 와 같이 나타나게 된다.

그림 Ⅲ.3.5 ETA 사상수목 예(2)

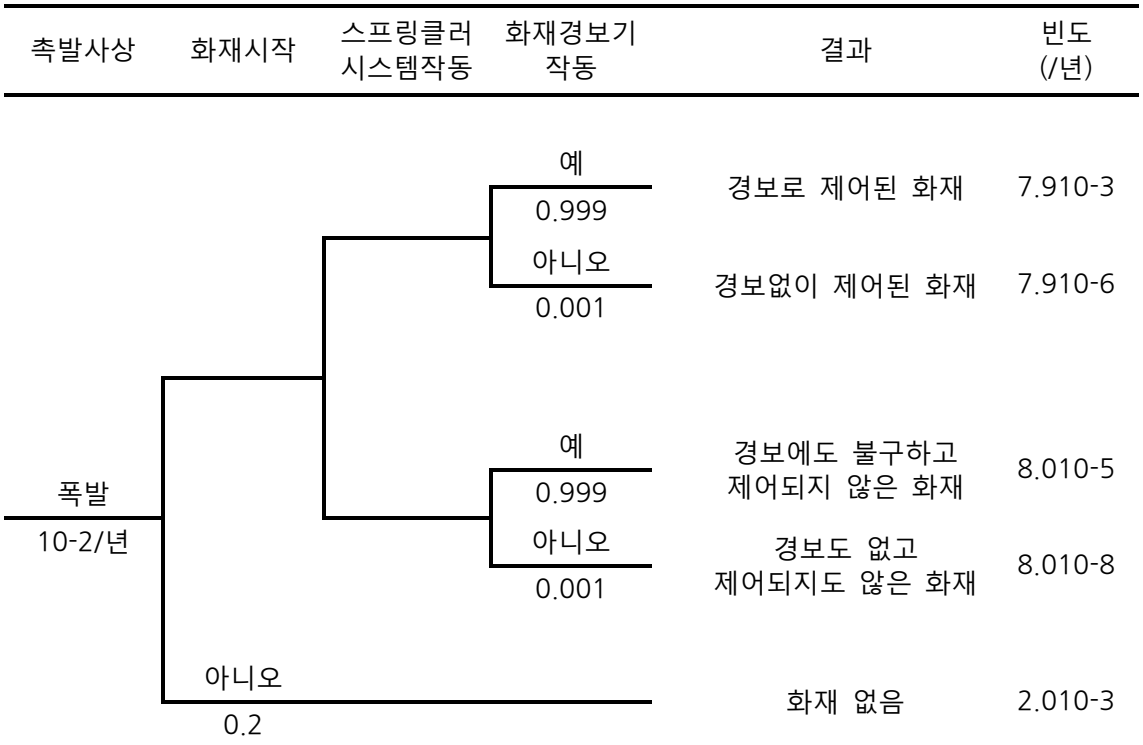


마지막으로, 재해요인들의 정상적 작동확률 (신뢰도) 이나 고장발생확률 (불신뢰도)을 알 수 있다면, 각각의 요소에 그 수치들을 기입하고, 그 확률들을 곱하면 결과적으로 발생될 수 있는 상황의 발생확률을 얻게 된다. 이 때 모든 상황에 대한 확률들의 최종합이 1 이 되어야 하는 것은 물론이다.

다. 분석사례

사례는 가스압력을 자동제어하는 제품에 있어서 고압가스폭발에 의한 화재사고를 사상수목분석한 사례이다. 최초의 촉발사상은 가스 폭발로부터 시작하며, 발화로부터 스프링클러와 화재경보 시스템의 작동여부에 따라 어떤 결과가 발생할 수 있는지, 그리고 그 확률은 대체로 얼마인지 추정할 수 있음을 그림 Ⅲ.3.6 에서 보여준다.

그림 Ⅲ.3.6 사상수목 분석사례



라. 기타사항

이렇게 하면 어떤 과정을 거쳐 사고가 발생하는지 상호 배반적인 상황의 전개를 눈으로 확인할 수 있어서 종래에는 보아 넘기기 쉬웠던 재해의 확대 요인을 쉽게 검출할 수 있다. 그러므로 사상수목분석결과는 이후에 설명할 결함수목분석 (Fault Tree Analysis) 을 수행하기 위한 기초자료를 제공한다는 점에서 매우 중요하다. 즉, 이 분석기법은 FTA 를 이용하여 더 상세히 분석되어야 할 사상들을 구명하는 데 효과가 높다.

한편, ETA 는 분석의 목적에 따라 어떤 요인이라도 촉발사상으로 삼을 수 있다. 똑같은 자동제어 제품이라 할지라도 분석자의 시점에 따라 재해연쇄 (accident sequence) 를 어떻게 이해하느냐, 즉 관련요소들의 나열순서가 어떻게 되느냐에 따라 또 다른 사상수목을 구성할 수 있기 때문에 제품 분석의 융통성이 매우 높은 기법이다.

이 때 주의하여야 할 것은 각 고장요인들의 고장발생이 서로 독립이라고 가정하고 있다는 것이며, 만약 각 고장요인들의 고장발생이 서로 독립이 아니라면 엄격한 의미에서 곱셈계산에 의한 확률추정은 잘못된 것이다.

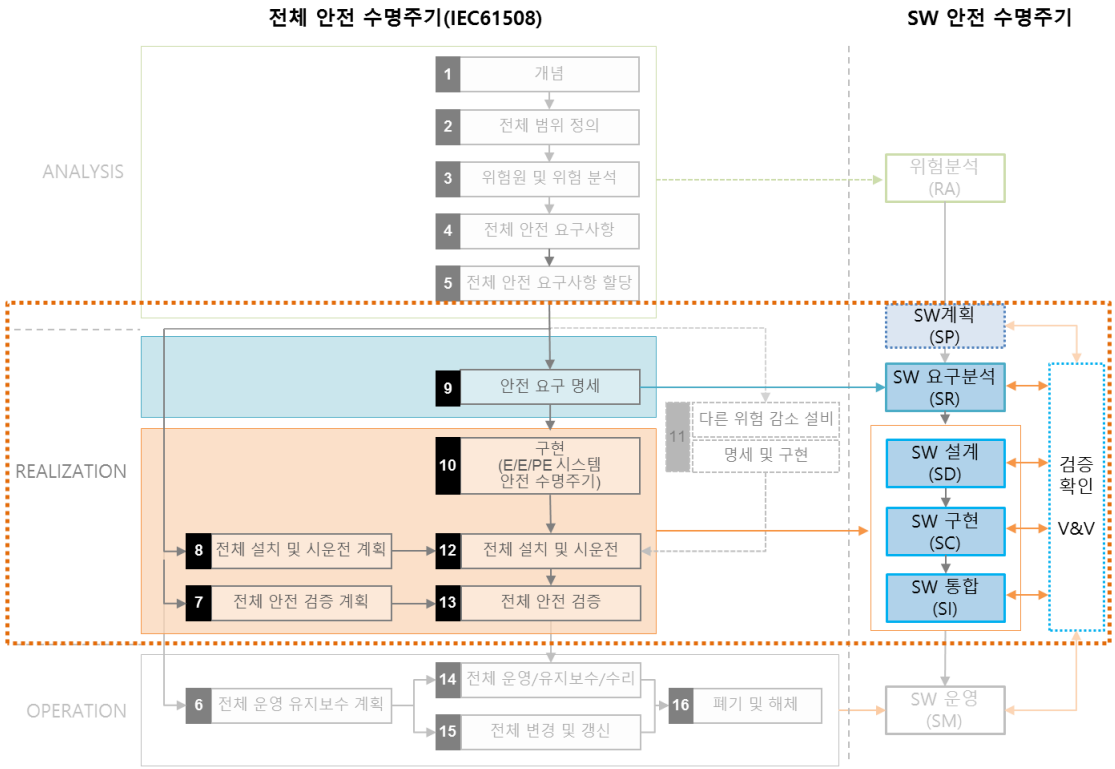
IV. SW 개발(Realization)

제 1 장 개요

1. 범위 및 목적

앞장에서 수명주기의 Analysis 부문의 위험분석 단계에 대해서 설명하였고, 본 장에서는 Realization 부문의 SW 계획, SW 요구분석, SW 설계, SW 구현, SW 통합, SW 확인 검증 등의 단계에 대해 설명한다. 그림 IV.1.1 은 IEC61508 의 전체 안전 수명주기와 본 가이드의 SW 안전 수명주기의 연결을 보여준다. 본 가이드의 SW 안전 수명주기는 ScarFS(Software to be Careful about Functional Safety)로 정한다.

그림 IV.1.1 SW 개발 수명주기



2. SW 개발 수명주기(Realization) 세부 항목

모든 단계의 공통 항목으로 다음을 포함한다.

- 단계 흐름도: 단계 내의 활동과 입출력물을 순서도로 도식화한 것으로서 반드시 순서의 의미를 갖지 않는 활동도 기술된다.
- 목적: 단계의 고유한 목적을 기술한다.
- 책임과 권한: 개발 단계 책임자 또는 수행자들의 책 임과 권한을 기술한다. 책임과 역할의 구분을 명료하게 하기 위하여 편의상 품질 담당과 안전 담당을 분리하여 기술하였으나, 실무에서는 품질담당자가 안전담당자가 수행하는 안전 확인 및 검증 업무를 병행하여 수행하는 경우가 많다. 하지만, 위험 분석 단계에서 시스템 및 SW 의 위험을 분석하는 것은 안전 전문가의 영역이다.
- 시작 및 종료 기준: 단계의 시작과 종료 기준을 설정한다.
- 활동: 단계에서 수행되어야 할 활동으로서, 프로세스를 구조적 분해한 것이다. 활동은 크게 개발활동, 확인 검증 활동, 안전 관련 활동으로 구분한다. 활동은 경우에 따라 세부 활동으로 세분화 되기도 한다. 특히 안전활동의 경우는 활동에 대한 기본 설명 후 "IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)"을 제공하여 각 개발 단계별로 SIL Level 별 점검해야 할 내용을 제공하였다.
- 기타: 단계와 관련된 고려사항들로서, 사용 양식, 적용 기법 등으로 구성되어 있다.

기법 및 사용 양식은 고유 번호를 부여하여 식별한다. 단계 명은 단계의 영문명을 참고한 2 자리 이내의 영문 식별자를, 이어서 그 유형에 따라 기법은 T: Technique, 문서는 D: Documentation, V: Validation & Verification 으로 구분자가 부여되고, 이어서 단계별 일련번호를 부여한다.

본 방법론은 안전과 관련된(safety-related) SW 를 개발할 때 활용할 수 있다.

- 절차서는 방법론을 구성하는 각 단계와 각 단계에 포함된 활동을 보여준다. 각 단계는 개발, 확인 검증 및 안전과 관련된 활동으로 구성되어 있으며, 활동들은 주어 진 입력을 받아들여 출력을 생성하기 위한 과정이다.

- 안전관련 SW 개발 기법은 활동 내역으로 설명하기에는 보다 기술적인 내용을 포함하고 있는 사항들을 모아둔 것으로 본 장의 내용이다. 이러한 기법들은 기술이 발전되면서 지속적으로 확대해 나갈 수 있다.
- 산출물은 절차서에서 정의된 입출력물에 대해 목차 및 그 구성·내용을 설명하고 있다. 이 양식을 활용함으로써 사용자는 보다 쉽게 방법론에서 원하는 입출력물을 활용할 수 있다. 부록 Appendix1 에 세부적인 내용이 기술되어 있다.

3. SIL 수준에 맞는 검증기법 선정 및 활용법

안전활동에 포함된 "기능 안전 검증 기법(Technique/Measures)"은 각각의 SIL Level 별로 단계별 검증해야 할 것을 IEC 61508-6 Annex E 에 나와있는 내용을 예시로 설명하면 표 IV.1.1 과 같다. 여기서는 SIL 2 를 기준으로 "Software safety requirements specification" 부분에 대한 점검 목록 및 활용방법을 설명한다.

표 IV.1.1 점검 대상 목록 : Software safety requirements specification

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	준 정형 기법	IEC61508-3 B.7	R	R	HR	HR
1b	정형 기법		---	R	R	HR
2	시스템 안전 요구 사항과 소프트웨어 안전 요구 사항 간의 전방 추적 성		R	R	HR	HR
3	안전 요구 사항과 인식 된 안전 요구 사항 간의 역 추적 성		R	R	HR	HR
4	위의 적절한 기술 / 조치를 지원하는 컴퓨터 지원 사양 도구		R	R	HR	HR

SIL 1 에서 4 에 대한 Technique/Measure 적용은 다음을 참고한다.

- HR : Highly Recommended, 이 기법 및 수단은 안전무결성을 위해 반드시 권고되는 기법임을 나타낸다. 이 기법이나 수단이 이용되지 않는다면 안전 계획에서 그 합리적 근거가 상술 되어야 하고 평가자의 동의를 얻어야 한다.
- R : Recommended, 이 기법 및 수단은 안전무결성을 위해 HR 보다 낮은 권고 수준을 갖는다.
- **---** : 이 기법 및 수단은 안전무결성을 위해 권고되는 사항이 없음을 의미한다.
- NR : Not Recommended, 이 기법 및 수단은 안전무결성을 위해 절대 권고되지 않음을 의미한다. 이 기법이나 수단을 이용한다면 그것을 이용하는 합리적 근거를 안전 계획중에 상술되어야 하고 평가자의 동의를 얻어야 한다.

다음의 표 IV.1.2 는 SIL2 기준에 맞는 Technique/Measure 적용 여부를 점검한 예시이다. '이 적용에 대한 해석' 컬럼의 내용에 점검자가 확인한 내용을 기술한다.

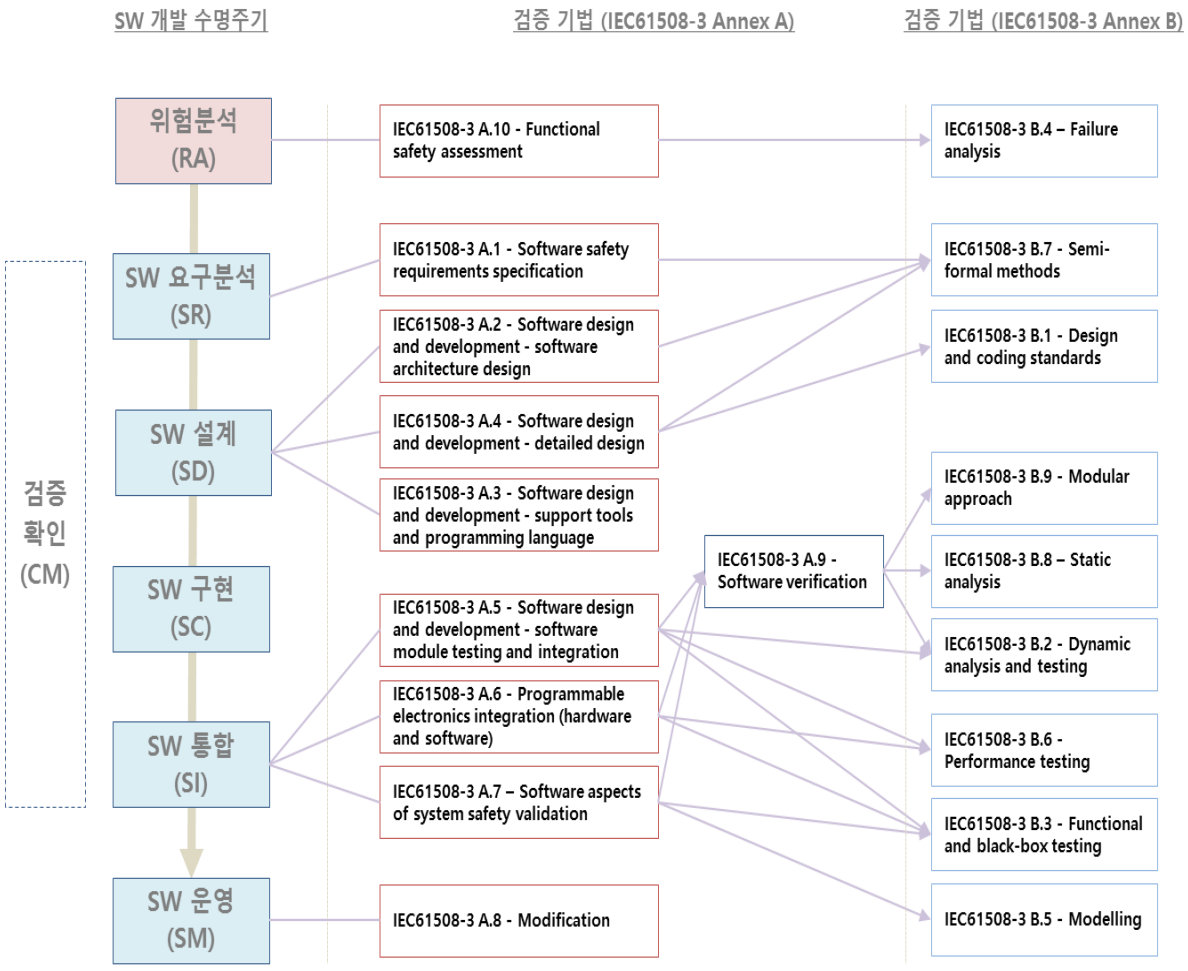
표 IV.1.2 목록 활용 방법 : Software safety requirements specification

Technique/Measure		SIL 2	이 적용에 대한 해석
1a	준 정형 기법	R	원인 - 효과 다이어그램, 시퀀스 다이어그램, 기능 블록. 일반적으로 PLC 응용 소프트웨어 요구 사양에 사용됨
1b	정형 기법	R	사용되지 않음
2	시스템 안전 요구 사항과 소프트웨어 안전 요구 사항 간의 전방 추적 성	R	완전성 검사 : 모든 시스템 안전 요구 사항이 소프트웨어 안전 요구 사항에 의해 다루어 지는지 검토함
3	안전 요구 사항과 인식 된 안전 요구 사항 간의 역 추적 성	R	복잡성과 기능 최소화 : 시스템 안전 요구 사항을 해결하기 위해 모든 소프트웨어 안전 요구 사항이 실제로 필요하다는 것을 검토함
4	위의 적절한 기술 / 조치를 지원하는 컴퓨터 지원 사양 도구	R	PLC 제조업체가 제공하는 개발 도구 사용
비고 : 소프트웨어 안전 요구사항은 자연어로 기술되어 있음			

표 IV.1.2 의 SIL 2 수준은 모두 R, 권고 수준이므로 해당 Technique/Measure 필수로 적용할 필요는 없다. 하지만 SIL 3 수준은 HR 에 해당하는 Technique/Measure 은 반드시 적용해야 한다.

적용하는 Technique/Measure 은 반드시 하나의 단계에서만 사용하는 것이 아닌 여러 단계에서 사용하는 경우도 있다. 각 단계별 Technique/Measure 간의 연결은 그림 IV.1.2 와 같다.

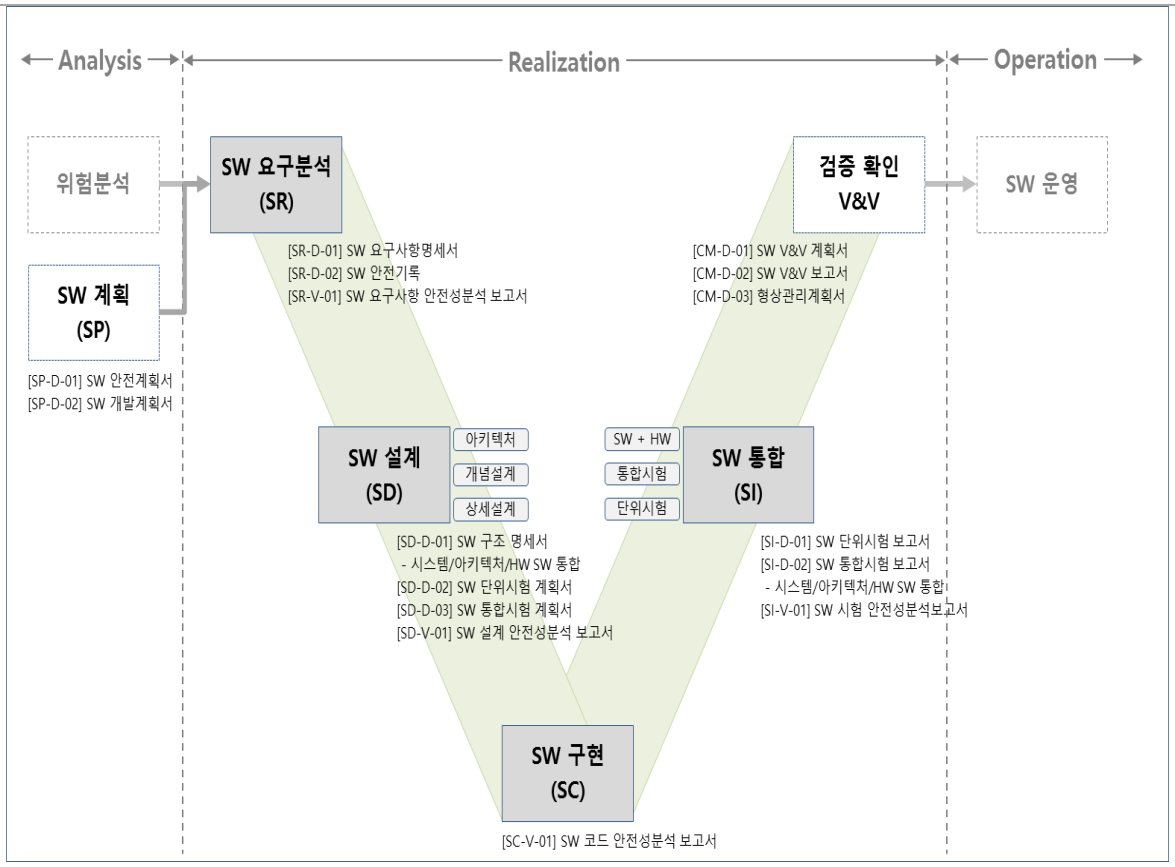
그림 IV.1.2 수명주기별 Technique/Measure 매칭



제 2 장 SW 개발 (Realization) 수명주기 상세

본 가이드의 SW 안전 수명주기를 V&V 다이어그램 형태로 표현하면 그림 IV.2.1 과 같다.

그림 IV.2.1 신뢰 안전 SW 개발 단계



본 개발 수명주기는 시스템 개발에 관한 활동은 사전에 이루어졌으며, 시스템 중 SW 에 할당된 요구사항이 확정되어 있다고 가정한다. 따라서 본 수명주기에 포함하는 부분은 SW 계획수립에서 SW 검증 확인 단계이다. 각 단계에는 개발활동, 검증 및 확인 활동과 안전 활동으로 구성된다. 표 IV.2.1 과 같다.

표 IV.2.1 기능안전 방법론 구성 상세

구분	세부활동			산출물		기법
	개발 활동	확인검증 활동(품질보증)	기능안전 활동(안전관리)	입력물	출력물	
SW계획 (SP)	1.1.1 SW 개발계획 수립 1.1.2 SW 개발환경계획 수립 1.1.3 SW 개발표준 수립	1.2.1 SW 검증계획 수립 1.2.2 SW 개발계획 평가 1.2.3 확인검증 작성	1.3.1 SW 안전 분석 1.3.2 SW 안전계획 수립	• 개발 계획 • RFP, 고객/시스템 요청사항 • 위험/안전 기록 • EUC 환경 및 관련 정보 • 시스템 요구사항 명세서	[SP-D-01] SW 개발계획서 [SP-D-02] SW 안전계획서 [CM-V-01] SW 확인검증 계획서 [CM-V-02] SW 확인검증 보고서 [CM-V-03] 형상관리계획서	
SW요구분석 (SR)	2.1.1 요구사항명세 2.1.2 추적성 유지	2.2.1 SW요구사항 평가 2.2.2 요구사항테스트명세서 작성 2.2.3 SW요구사항확인검증 작성	2.3.1 SW 요구사항 안전평가 수행 2.3.2 SW 안전기록 작성	• 시스템 요구사항 명세서 • 시스템 안전성 요구사항 명세서 • 시스템 구조 기술서 • SW 개발계획서 • SW 확인검증계획서 • SW안전계획서	[SR-D-01] SW 요구사항명세서 [SR-D-02] SW 안전기록 [SR-V-01] SW 요구사항 안전성분석 보고서	[SR-T-01] SW 요구사항 명세 기법 [SR-T-02] SW 요구사항 위험분석 기법
SW설계 (SD)	3.1.1 SW 구조 설계 3.1.2 SW 설계명세서 정의 3.1.3 SW 통합테스트 명세서 작성 3.1.4 SW 모듈설계 명세서 작성 3.1.5 SW 모듈테스트명세서 작성	3.2.1 SW 설계 평가 3.2.2 통합테스트명세서 검증 3.2.3 SW 모듈테스트명세서 검증 3.2.4 확인검증 작성	3.3.1 SW설계 안전평가 수행 3.3.2 SW 모듈안전평가 수행 3.3.3 SW 안전기록 작성	• SW 요구사항 명세서 • 시스템 안전성 요구사항 명세서 • 시스템 구조 기술서 • SW개발계획서 • SW 확인검증계획서 • SW안전계획서	[SD-D-01] SW 설계 명세서 - 아키텍처/시스템/모듈 [SD-D-02] SW 단위시험 계획서 [SD-D-03] SW 통합시험 계획서 - 시스템/아키텍처/HW SW 통합 [SD-D-04] SW 코딩 메뉴얼 [SD-V-01] SW 설계 안전성분석 보고서	[SD-T-01] SW 설계 기법 [SD-T-02] SW 구조 위험분석 기법 [SD-T-03] SW 모듈 위험분석 기법
SW구현 (SC)	4.1.1 SW 모듈 구현	4.2.1 SW 구현 평가 4.2.2 확인검증 작성	4.3.1 SW 구현 안전평가 수행 4.3.2 SW 안전기록 작성	• SW 모듈설계 명세서 • 시스템 안전성 요구사항 명세서 • SW개발계획서	[SC-V-01] SW 코드 안전성분석 보고서	[SC-T-01] SW구현 안전성평가 기법
SW통합 (SI)	5.1.1 SW 모듈테스팅 5.1.2 SW 통합테스트 개발 5.1.3 SW 통합테스트 수행 5.1.4 SW/HW 테스트 수행	5.2.1 추적가능성 분석 5.2.2 모듈테스트 결과 확인 5.2.3 통합테스트 결과 검증 5.2.4 SW/HW 테스트 검증 5.2.5 확인검증 작성	5.3.1 SW 통합 안전평가 수행 5.3.2 SW 안전기록 작성	• SW 모듈 • SW통합계획서 • 시스템 안전성 요구사항 명세서 • SW 개발 계획서	[SI-D-01] SW 단위시험 보고서 [SI-D-02] SW 통합시험 보고서 - 시스템/아키텍처/HW SW 통합 [SI-V-01] SW 시험 안전성분석보고서	[SI-T-01] SW 단위시험 절차 [SI-T-02] SW 통합시험 절차
SW운영 (SM)	6.1.1 SW 운영, 유지보수 및 수리 6.1.2 변경 및 갱신 처리 6.1.3 폐기 및 해체 처리	6.2.1 SW 운영 평가 6.2.2 확인검증 작성	6.3.1 SW 운영 안전평가 수행 6.3.2 SW 안전기록 작성	• SW 변경/폐기 요청 • SW 오류	[SR-D-01] SW 유지보수계획서 [SR-V-01] SW 운영 안전성분석보고서	

1. SW 계획(SP) 단계

Software Planning Phase

<table>
<tr><td>목적</td><td>프로젝트 시작을 위한 SW 개발 계획 및 검증 계획을 수립한다.</td></tr>
<tr><td>시작 기준</td><td>프로젝트 계획이 수립됨
고객의 요청사항이 접수됨
개발할 안전관련시스템 대상 관련 정보가 수집됨</td></tr>
<tr><td>활동</td><td>• 프로젝트 계획
• RFP, 고객/시스템 요청사항
• 위험/안전 기록
• EUC 환경 및 관련 정보
• 시스템 요구사항 명세서

<table>
<tr><th>개발 활동</th><th>확인 검증 활동</th><th>안전 활동</th></tr>
<tr><td>1.1.1 SW 개발계획 수립
1.1.2 SW 개발환경계획 수립
1.1.3 SW 개발표준 수립</td><td>1.2.1 SW 검증계획 수립
1.2.2 SW 개발계획 평가
1.2.3 확인검증 작성</td><td>1.3.1 SW 안전 분석
1.3.2 SW 안전계획 수립</td></tr>
</table>

[SP-D-01] SW 개발계획서
[SP-D-02] SW 안전계획서
[CM-V-01] SW 확인검증 계획서
[CM-V-02] SW 확인검증 보고서
[CM-V-03] 형상관리 계획서</td></tr>
<tr><td>확인</td><td>대상 시스템에 대한 이해로부터 범위가 정해지고 관련 위험 분석 결과를 반영함
프로젝트 관리자에 의해 각종 계획들이 승인됨
계획은 문서화되어 단계별 일관성이 유지 방안이 마련됨</td></tr>
</table>

종료기준	SW 위험 분석 및 개발계획이 작성된다. SW 품질보증계획, 형상관리계획, 검증계획, 통합계획, 확인계획이 수립된다.

가. 목 적

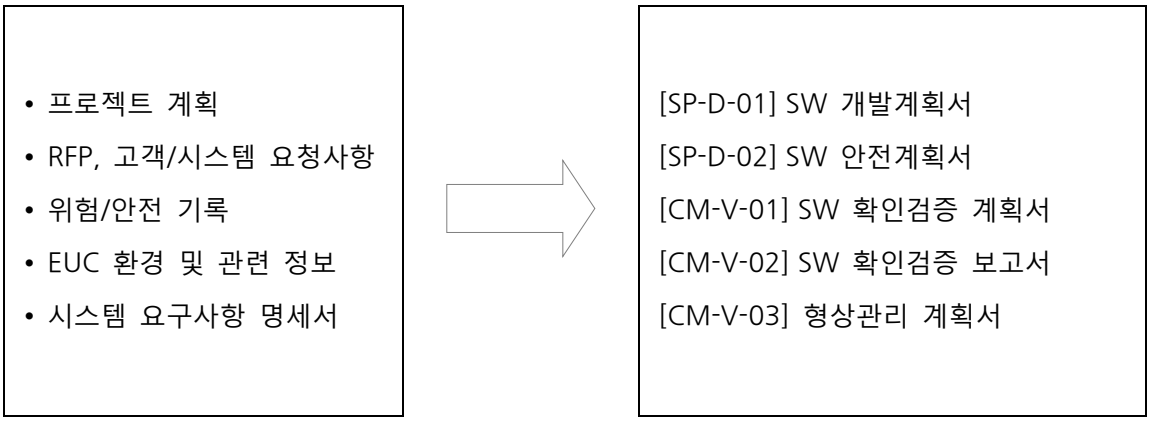
프로젝트 시작을 위한 SW 개발 계획 및 검증 계획을 수립한다.

나. 책임과 권한

활동 \ 역할		계획수립 담당자	품질담당자	안전담당자	PM
개발활동	1.1.1 SW 개발계획 수립	R			A
	1.1.2 SW 개발환경계획 수립	R			A
	1.1.3 SW 개발표준 수립	R			A
검증활동	1.2.1 SW 검증계획 수립	S	R	S	A
	1.2.2 SW 개발계획 평가		R		A
	1.2.3 확인 검증 보고서 작성		R	S	A
안전활동	1.3.1 SW 안전 분석	S	S	R	
	1.3.2 SW 안전계획 수립	S	S	R	A

(Responsible:담당, Approve:승인, Support:지원)

다. 입력물 및 출력물



라. 개발 활동

1.1.1 SW 개발 계획 수립

SW 에 할당된 요구사항을 구현하기 위한 SW 개발계획서를 작성한다. SW 개발계획서는 시스템의 전체 수명주기에 참여하는 사람들이 이해 가능 하도록 표현한다.

개발계획서에는 개발 조직, 작업계획, 통제 계획 등의 기본적인 내용을 포함하고, 개발 절차, 자원 계획 등이 포함된다.

1.1.2 SW 환경 계획 수립

SW 개발 환경에 대한 계획을 수립한다. 개발 환경에는 사용언어 및 컴파일러, SW 테스트 환경 등이 포함된다. SW 환경은 SW 전체 수명주기에 걸쳐 요구되는 SW 안전 무결성 수준에 적합하게 수립한다.

SW 개발환경은 다음을 고려하여 선정한다.

- SW 개발환경은 최종 SW 에 미치는 잠재적인 위험을 최소화하기 위해 선정한다.
- 자격 있는 도구 또는 도구들의 조합, SW 개발환경은 어느 한 부분에서 도입된 오류를 다른 곳에서 찾아낼 수 있는 확신의 수준을 달성하기 위해 선정된다.
- 잠재적인 SW 개발 환경과 관련된 오류를 최소화하기 위해 SW 안전 무결성 수준을 고려한 SW 검증 활동 또는 SW 개발 표준이 정의되어야 한다.
- 가능하다면 검증자 및 확인자의 니즈를 고려하여, 자동화 테스트 도구와 통합 개발 도구를 사용한다.
- SW 안전 무결성 수준에서 요구되는 정도까지, 선정된 프로그래밍 언어는 다음 중 하나의 조건을 만족하는 번역기 또는 컴파일러를 제공해야 한다
  - 인정된 국내 국제 표준 규격에 대해 "확인 인증"
  - 목적 적합성을 상세히 설명한 평가 보고서
  - 변환 오류 검출을 제공하는 중복 서명 통제 기반의 프로세스.
- 선정된 언어는 다음의 요구사항을 반드시 만족해야 한다.
  - 선정된 언어는 프로그래밍 오류 식별을 활성화하는 특성을 포함해야 한다.
  - 선정된 언어는 설계 방법에 맞는 특성을 지원해야 한다.
  - 위의 내용이 만족되지 않을 경우 대안으로 선택된 언어가 목적에 적합하다는 정당성을 설명한 내용을 SW 개발계획서에 기록한다.
- SW 테스트 환경은 통합 단계의 결과물을 테스트하는데 사용될 방법, 도구, 절차 및 HW 를 정의한다. 테스팅은 타겟 컴퓨터, 타겟 컴퓨터 에뮬레이터 또는 호스트 컴퓨터 시뮬레이터를 사용하여 수행될 수도 있다. 이 때 에뮬레이터 또는 시뮬레이터와 타겟 시스템과의 차이점을 고려해야 한다.

1.1.3 SW 개발 표준 수립

SW 개발 표준은 SW 개발을 위한 규칙과 제약사항을 정의한다. SW 개발표준에는 SW 요구사항 표준, SW 설계 표준, SW 코드 표준 등이 포함된다. SW 개발 표준은 검증함 수 없거나 안전 요구사항에 적합하지 않은 결과물을 생성하는 것을 방지할 수 있어야 한다.

- 코딩 관련 표준 규격은 훌륭한 프로그래밍 프랙티스를 명시하며, 불안전한 언어 특성을 사용 금지하며, 소스 코드 문서화 절차를 설명하여야 한다. 최소한 각 SW 모듈은 소스 코드 내에 작성자, 형상 이력, 간단한 설명을 포함하여야 한다. 이러한

정보에 대한 표준 양식이 사용되는 것이 좋으며, 모든 모듈에 대하여 동일하게 적용되도록 한다.

마. 확인 검증 활동

1.2.1 SW 검증 계획 수립

검증 활동이 적절히 안내되고. 특정 설계 또는 기타 검증 니즈가 적절히 제공되도록 SW 확인 검증 계획서를 생성한다. 개발 과정에서 여러 개의 작은 문서로 분할 또는 추가될 수 있다.

SW 확인 검증 계획서에는 다음을 포함한다.

- 검증 전략 및 기법의 선정
- SW 테스트 장비의 선정 및 활용
- 검증 활동의 선정 및 문서화
- 검증 결과의 평가
- 신뢰성 요구사항의 평가
- 테스트 프로세스에 참여한 사람들의 역함 및 책임
- 테스트 커버리지 정도

1.2.2 SW 개발 계획 평가

SW 계획, SW 환경 및 SW 개발 표준이 SW 안전 무결성 수준에 적합한지 다음 항목을 사용하여 검증한다.

- 선정된 방법이 본 방법론의 절차를 준수하는지 여부
- SW 개발 방법이 일관성 있게 적용되는지 여부 각 단계에서 만들어지는 결과물이 추적될 수 있는지 여부
- SW 계획 단계의 결과물이 일관성 있고 표준에 따라 작성되었는지 여부

1.2.3 확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술한다.

바. 안전 활동

1.3.1 SW 안전 분석

본 단계는 개발할 안전관련 시스템에 대한 위험분석 단계와 동시에 이루어 진다. 위험 분석에 대한 결과를 받아 소프트웨어에 반영하기 위한 준비를 한다.

전체 안전 요구사항에서는 필요한 기능안전성을 달성하기 위하여 안전관련 시스템, 기타 리스크 감소 설비에 대한 안전기능 요구사항과 안전무결성 요구사항으로 나타나는 전체 안전 요구사항 명세서를 개발한다.

전체 안전 요구사항 할당은 전체 안전 요구사항 명세(안전기능 및 안전 무결성 요구사항 모두)에 제시된 안전기능을 지정된 안전관련 시스템, 기타 리스크 감소수단에 할당하고 안전무결성 수준을 안전관련 시스템에 의해 수행될 각 안전기능에 할당한다.

1.3.2 SW 안전 계획서 작성

SW 위험 분석 절차을 활용하여 SW 의 위험을 분석한 후 안전을 확보하기 위한 SW 안전성 위험 분석에 관한 SW 안전계획서를 작성한다. SW 안전계획서는 SW 안전성 위험 분석 방법을 상세히 기술한다. 경우에 따라 SW 안전계획서는 시스템 안전계획서 내에 포함시킬 수 있다.

IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)

IEC61508-3 A.10 - Functional safety assessment

Assessment/Technique		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	점검표		R	R	R	R
2	의사 결정 / 진리표		R	R	R	R
3	고장 분석	IEC61508-3 B.4	R	R	HR	HR
4	다양한 소프트웨어의 공통 원인 실패 분석 (다양한 소프트웨어가 실제로 사용되는 경우)		---	R	HR	HR
5	신뢰성 블록 다이어그램		R	R	R	R
6	8 절의 요구 사항과 소프트웨어 기능 안전성 평가 계획 사이의 전방 추적성		R	R	HR	HR
• 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 A.10.1 : 점검표 (Checklists)

1. 목표

   안전 수명주기 단계별로 시스템의 모든 중요한 측면을 주의 깊게 파악하고 중요한 평가를 관리하며 정확한 요구 사항을 세우지 않고 포괄적인 범위를 보장한다.

2. 설명

   체크리스트를 수행하는 사람이 체크해야할 일련의 질문사항이다.
   대부분의 질문은 일반적이며 평가자는 평가되는 특정 시스템에 가장 적합한 것처럼 해석한다. 점검 목록은 전반적인 E / E / PE 시스템 안전 및 소프트웨어 안전 수명주기의 모든 단계에 사용할 수 있으며 특히 기능 안전성 평가를 돕는 도구로 유용하다. 검증되는 시스템의 다양한 변경을 수용하기 위해 대부분의 점검 목록에는 여러 유형의 시스템에 적용 할 수있는 질문이 있다. 반면 사용중인 점검 목록에 처리중인 시스템과 관련이 없고 무시해야하는 질문이 있을 수 있다. 특정 시스템의 경우, 다루고 있는 시스템에 구체적으로 지시 된 질문을 표준 체크리스트에 보충 할 필요가 있다. 어떤 경우든 체크리스트를 사용하는 것은 엔지니어가 체크리스트를 선택하고 적용하는 전문 지식과 판단에 크게 의존한다. 결과적으로 선택된 체크리스트에 관한 엔지니어의 결정 및 추가

또는 불필요한 질문은 모두 문서화되어야 한다. 목적은 다른 기준을 사용하지 않는 한 체크리스트 적용을 검토하고 동일한 결과를 얻을 수 있도록 보장하는 것이다.
체크리스트를 작성하는 데 필요한 객체는 가능한 간결해야한다. 정당성이 필요한 경우 추가 문서를 참조해야 한다. 합격, 불합격 및 결정적이지 않은 응답 또는 유사하게 제한된 응답 집합을 사용하여 각 질문의 결과를 문서화해야 한다. 이 간결성은 체크리스트 평가의 결과에 대한 결론에 도달하는 절차를 크게 줄인다.

3. 비고
   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.2.5 를 참조한다.

4. 분야별 가이드 참조 위치
   ☞ 철도 가이드: 부록 > B-7
   ☞ 자동차 가이드: -
   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..1 단계 0 안전등급분류 및 식별, 단계 2 소프트웨어 요구사항 분석

IEC61508-3 A.10.2 : 의사 결정 / 진리표 (Decision/truth tables)

1. 목표
   복잡한 논리적 조합과 관계에 대한 명확하고 일관된 명세 및 분석을 제공한다.

2. 설명
   2 차원 테이블을 사용하여 부울 프로그램 변수 간의 논리적 관계를 간결하게 설명한다. 테이블은 코드로 표현 된 복잡한 논리 조합을 분석하는 수단으로 적합하며 명세로 사용될 경우 잠재적으로 실행 가능합니다.

3. 비고
   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.6.1 을 참조한다.

4. 분야별 가이드 참조 위치
   ☞ 철도 가이드: 부록 > B-13
   ☞ 자동차 가이드: * PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.1 요구사항 분석 > 결정 테이블 테스팅(Decision Table Testing)

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..1 단계 0 안전등급분류 및 식별, 단계 2 소프트웨어 요구사항 분석

IEC61508-3 A.10.5 : 신뢰성 블록 다이어그램 (Reliability block diagram)

1. 목표

   도식화된 형태로 일어나는 일련의 사건과 시스템이나 작업의 성공적인 운영을 위해 충족되어야하는 조건을 모델링한다.

2. 설명

   분석 대상은 블록, 라인 및 논리 접합으로 구성된 성공 경로로 표현된다. 성공 경로는 다이어그램의 한쪽에서 시작하여 블록 및 교차점을 통해 다이어그램의 반대쪽까지 계속된다. 블록은 조건 또는 이벤트를 나타내며 조건은 참이거나, 아니라면 이벤트가 발생한 경우 경로가 전달할 수 있다. 접합부에 경로가 오면 접합부의 논리가 충족되면 경로가 계속된다. 정점에 도달하면, 모든 출력 라인을 따라 계속 될 수 있다.
   다이어그램을 통해 성공 경로가 하나 이상 존재하면 분석 대상이 올바르게 작동한 것이다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.6.4 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..1 단계 0 안전등급분류 및 식별, 단계 2 소프트웨어 요구사항 분석

IEC61508-3 B.4 – Failure analysis

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1a	원인 다이어그램		R	R	R	R
1b	이벤트 트리 분석		R	R	R	R
2	결함 트리 분석		R	R	R	R
3	소프트웨어 기능 장애 분석		R	R	R	R

- 소프트웨어를 가장 적절한 안전 무결성 레벨로 분류하기 위해 예비 위험 분석을 수행한다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.

IEC61508-3 B.4.1a : 원인 다이어그램 (Cause consequence diagrams)

일반 사항

1. 목표

   요약 이벤트를 조합 한 결과로 시스템에서 발생할 수있는 일련의 이벤트를 소형 다이어그램 형식으로 분석 및 모델링한다.

2. 설명

   폴트 트리와 이벤트 트리 분석의 조합이며, 시작과 같은 중요한 이벤트에서 시작하고 시퀀스 그래프는 일부 작업의 성공과 실패를 설명하는 YES / NO 게이트를 사용하여 표시된다. 이를 통해 사고 또는 마스터 된 상황을 유도하는 이벤트 시퀀스를 작성할 수 있다. 그리고 다음 각 실패에 대한 그래프 (즉, 폴트 트리)가 작성된다. 다음, 우발적인 상황에서 시작하여 역방향으로 진행하며, 우발적인 상황을 가장 큰 사건으로 여기는 글로벌 폴트 트리를 작성한다. 진행 방향에서 사건으로 인해 발생할 수있는 결과가 결정된다. 그래프는 정점과 다른 가지를 따라 전파하기 위한 조건을 설명하는 정점 기호를 포함 할 수 있으며 시간 지연도 포함될 수 있다. 이러한 조건은 오류 트리로 설명할 수 있다. 전파 라인을 논리 기호와 결합하여 다이어그램을 컴팩트하게 만들 수 있다. 원인 결과 다이어그램에 사용하기 위해 표준 기호 세트가 정의된다. 다이어그램은 폴트 트리를 작성하고 특정 비판적인 결과가 발생할 확률을 계산하는 데 사용할 수 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.6.6.2 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-6

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.4.1b : 이벤트 트리 분석 (Event tree analysis)

1. 목표

   개략적인 형태로 시작 이벤트 후 시스템에서 발생할 수있는 일련의 이벤트를 모델링하여 심각한 결과가 발생할 수 있음을 보여준다. 이벤트 트리는 처음부터 작성하기가 어렵고 결과 다이어그램을 사용하는 것이 작성에 도움이 된다.

2. 설명

   다이어그램의 맨 위는 시작 이벤트 다음에 진행되는 이벤트의 진행과 관련된 순서 조건이 기록된다. 분석 대상인 시작 이벤트에서 시작하여 첫 번째 조건으로 선이 그려진다. 거기서 다이어그램은 "예"와 "아니오" 가지로 나뉘어 미래 이벤트가 어떻게 상태에 의존하는지 설명한다. 가지들 각각에 대해, 비슷한 방식으로 다음 조건을 이어 나간다. 모든 지점이 모든 조건을 만족하는 것은 아니다. 하나는 시퀀스의 끝까지 계속되며, 이렇게 구성된 트리의 각 가지는 가능한 결과를 나타낸다. 시퀀스의 조건이 독립적인 경우 이벤트 트리를 사용하여 시퀀스의 조건 확률을 기반으로 다양한 결과의 확률을 계산할 수 있다. 조건이 거의 완전히 독립적이기 때문에 이러한 계산은 숙련 된 분석가가 수행해야 한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.6.6.3 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-22

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.4.2 : 폴트 트리 분석 (Fault tree analysis)

1. 목표

   사건의 분석이나 사건의 조합을 위해 위험이나 심각한 결과를 초래하는 사건의 확률 을 계산한다.

2. 설명

위험 또는 심각한 결과의 직접적인 원인인 이벤트에서 시작하여 이벤트의 원인을 식별하기 위한 분석을 수행한다. 논리 연산자 등을 사용하여 여러 단계로 수행된다. 중간 원인은 같은 방식으로 분석되며, 분석을 중지하는 기본 이벤트로 돌아간다. 이 분석은 그래픽으로 되어있으며 표준화 된 기호 세트가 폴트 트리를 그리는 데 사용된다. 분석이 끝나면 결함 트리는 기본 이벤트 (일반적으로 구성 요소 실패)를 상위 이벤트 (전체 시스템 장애)에 연결하는 논리적 기능을 표시한다. 주로 하드웨어 시스템 분석을위한 것이지만, 소프트웨어 오류 분석에 적용할 수 있다. 이 기법은 고장 분석 및 상위 이벤트의 확률적 계산에 사용할 수 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.6.6.5 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.5 안전분석(safety analysis) > 1.5.1 주요 요구사항 및 설명 > 1.5.1.1 소프트웨어 아키텍처 수준의 안전분석

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.4.3 : 소프트웨어 기능 장애 분석(Software functional failure analysis)

고장 모드,영향 및 중요도 분석

1. 목표

   설계 또는 작동 중에 특별한 주의와 제어 조치가 필요한 구성 요소를 결정하기 위해 단일 포인트 오류를 통해 부상, 손상 또는 시스템 저하를 초래할 수있는 구성 요소의 중요성을 순위 지정한다.

2. 설명

   FMEA 와 유사하지만 여러 가지 방법으로 순위를 매길 수 있는 중요도를 나타내는 하나 이상의 컬럼이 있다. 이 절차에서 모든 구성 요소에 대한 중요도 번호는 중요 모드에서 발생하는 백만 건의 작업 중 예상되는 특정 유형의 오류 수로 표시된다. 임계 수는 9 개의 매개 변수의 함수이며 대부분이 측정되어야 한다. 중요도 결정을위한 매우 간단한 방법은 구성 요소 오류의 확률에 생성 될 수있는 손상을 곱하는 것이다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.6.6.4 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.5 안전분석(safety analysis) > 1.5.1 주요 요구사항 및 설명 > 1.5.1.1 소프트웨어 아키텍처 수준의 안전분석

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

사. 사용 양식

- [SP-D-01] SW 개발계획서
- [SP-D-02] SW 안전계획서
- [CM-V-01] SW 확인검증 계획서
- [CM-V-02] SW 확인검증 보고서
- [CM-V-03] 형상관리 계획서

아. 적용 기법

해당 사항 없음

2. SW 요구분석(SR) 단계

Software Requirements Specification Phase

<table>
<tr><td>목적</td><td>SW 요구분석의 목적은 시스템 요구사항을 충족시키는 SW 요구사항을 필요한 SW 안전 무결성 수준 정의 및 SW 요구사항 테스트명세를 하기 위한 것이다.</td></tr>
<tr><td>시작 기준</td><td>시스템 요구사항이 확정됨
시스템 안전 요구사항 명세가 확정됨
시스템 구조 설계가 확정됨
SW 품질보증계획이 작성됨</td></tr>
<tr><td>활동</td><td>
• 시스템 요구사항 명세서
• 시스템 안전성 요구사항 명세서
• 시스템 구조 기술서
• SW 개발계획서
• SW 확인검증계획서
• SW 안전계획서

<table>
<tr><th>개발 활동</th><th>확인 검증 활동</th><th>안전 활동</th></tr>
<tr><td>2.1.1 요구사항명세
2.1.2 추적성 유지</td><td>2.2.1 SW 요구사항 평가
2.2.2 요구사항 테스트 명세서 작성
2.2.3 SW 요구사항 확인검증 작성</td><td>2.3.1 SW 요구사항 안전평가 수행
2.3.2 SW 안전기록 작성</td></tr>
</table>

[SR-D-01] SW 요구사항 명세서
[SR-D-02] SW 안전 기록
[SR-D-03] SW 요구사항 안전성분석 보고서
</td></tr>
</table>

확인	검증 담당자에 의해 SW 요구사항명세서가 검증된다. 이해당사자들이 SW 요구분석서 및 테스트명세서를 확인한다. 과제책임자가 SW 요구분석서 및 테스트명세서를 승인한다
종료기준	SW 요구사항명세서가 작성됨 SW 요구사항 테스트명세서가 작성됨 SW 요구사항 확인 검증 보고서가 작성됨

가. 목 적

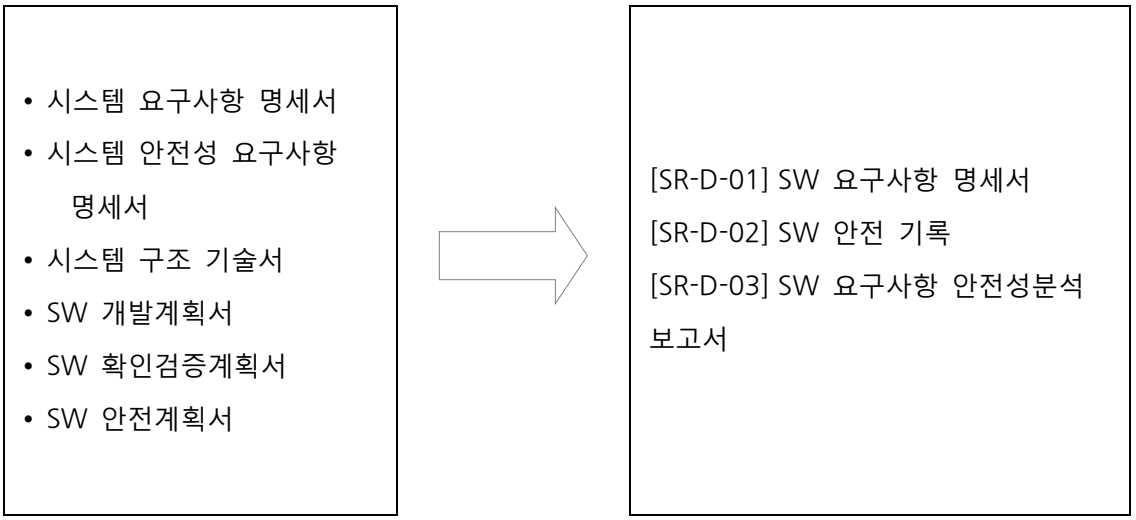
SW 요구분석의 목적은 시스템 요구사항을 충족시키는 SW 요구사항을 필요한 SW 안전 무결성 수준까지 정의하고, SW 요구사항 테스트명세서를 설명하기 위한 것이다.

나. 책임과 권한

책임 \ 역할		명세담당자	품질담당자	안전담당자
개발활동	2.1.1 요구사항명세서 작성	R		
	2.1.2 추적성 유지	R		
검증활동	2.2.1 SW 요구사항 평가	S	R	
	2.2.2 요구사항테스트명세서 작성	S	R	
	2.2.3 SW 요구사항확인 검증 보고서 작성	S	R	
안전활동	2.3.1 SW 요구사항 안전 평가 수행	S	S	R
	2.3.2 SW 안전 기록 작성	S	S	R

(Responsible: 담당, Approve:승인, Support:지원)

다. 입력물 및 출력물



라. 개발 활동

2.1.1 SW 요구사항명세서 작성

아래 내용들을 포함하여 SW 요구사항 명세서를 작성한다. 이 때 SW 요구사항 명세 기법을 활용한다. SW 요구사항명세서는 시스템의 전체 수명주기에 참여하는 사람들이 이해 가능하도록 표현한다.

SW 특성 명시

개발 대상 SW 에 요구되는 특성을 정의한다. (안전성을 제외하고) ISO/IEC 9126 에 정의된 이 특성에는 다음과 같은 것들이 있다.

- 기능성(용량과 응답 속도 성능을 포함)
- 신뢰성
- 유지보수성
- 효율성
- 사용성
- 이식성

SW 안전 무결성 수준 정의

안전성 기능 및 관련된 SW 안전 무결성 수준을 도출하여 기록한다. SW 안전 무결성 수준에서 요구하는 정도까지 기록한다. SW 안전 무결성 수준은 시스템 안전 무결성 수준에 기반하여 [RA-T-03] SW 무결성수준 결정기법을 사용하여 결정한다. SW 안전 무결성은 해당 SW 와 관련된 위험 수준에 근거하여 다음과 같이 5 개의 수준 중 하나로 결정한다.

SW 안전 무결성 수준	SW 안전 무결성 수준 설명
4	매우 높음
3	높음
2	중간
1	낮음
0	안전 무관

각 SW 구성요소가 서로 다른 안전 무결성 수준을 갖고 있다면 그 내용은 SW 구조 명세서에 명시한다. 안전과 관련된 요구사항은 안전 요구사항이라고 하고, 별도의 안전 요구사항 명세에 포함시킬 수 있다. 안전 요구사항은 다음 두 부분으로 나뉜다.

- 안전 기능적 요구사항: 실제 안전과 관련된 기능으로, SW 에서 수행되는 것임
- 안전 무결성 요구사항: 안전 관련 기능에 필요한 안전 무결성 수준을 정의함

중요도 분석은 안전과 관련된 SW 요구사항을 식별한다. 각 요구사항은 다양한 시스템 위험 분석(PHA 를 포함)에 대해 평가되어 잠재적인 수용 불가능한 위험들을 평가한다. 각 요구사항은 시스템 설계에 대해 평가되어 시스템 설계에서 할당된 SW 요구사항이 요구사항명세서에서 충족됨을 보장한다.

이전의 주요 작업 보고서에 존재하는 중요도 분석 결과를 SW 요구사항명세서를 이용하여 검토하고 갱신한다. 구현 방법론과 인터페이스 관련 기술은 주어진 SW 의

요소(요구사항, 모듈, 함수, 하위시스템, 다른 SW) 때문에 이전에 지정된 SW 무결성 수준을 상향 또는 하향 조절하는 원인이 될 수 있다. 수정된 SW 무결성 수준으로 검토에 의해 어떠한 불일치나 예상치 않는 SW 무결성이 발생하지 않음을 검증한다.

실행 시간, 클락 타임, 메모리 할당과 관련된 요구사항을 평가한다. 타이밍 분석에서는 하나 이상의 다음 기준을 충족하는 조건, 이벤트, 시간 간격을 식별한다.

- C 조건이 참이 된다면, 사건 A 가 T 초 이내에 발생되어야 한다.
- C 조건이 참이 된다면, 사건 A 는 T 초가 경과될 때까지 발생되어서는 안 된다.
- 사건 A 가 발생 후 T 초가 될 때까지 사건 B 가 발생되어서는 안 된다.

사전 성능 분석(preliminary performance analysis)이 수행될 수도 있다.

외부 인터페이스 식별

타 시스템과의 모든 인터페이스를 식별하여 문서화한다. 타 시스템은 운영자를 포함하여 통제 장치 내부 또는 외부에 있을 수 있고, 현재 직접 연결되어 있거나 향후 연결이 계획 된 것일 수 있다. 데이터 공유, 제공. 교환을 포함하는 기타의 요소들과의 관련성) 등 외부 연동 요구사항을 정의한다. 내부 데이터 요구사항 등은 내부 인터페이스 요구사항에 정의한다.

SW 운영 모드 및 행위 모드 정의

모든 관련된 운영 모드를 요구사항명세서에 상세히 기술한다. 또한 모든 관련된 프로그램 가능 소자의 행위 모드, 특히 고장 행위를 상세히 기술한다.

SW 가 다른 모드와 구별되는 요구사항을 갖고, 하나 이상의 모드의 운영을 요구한다면, 각 모드를 식별하고 정의하여야 한다. 모드의 예로서는, 유휴(idle), 준비(ready), 활동(active), 사용 후 분석, 교육, 일시적 이상상태(degraded), 비상사태, 백업, 전시, 평상시 등이 있다. 명세서에는 모드간 상관관계를 명시 해야 한다.

제약 사항 명시

HW 및 SW 사이의 모든 제약 사항을 식별하여 문서화한다.

SW 자체 점검 수준 명시

SW 자체 점검(self-checking) 수준과 SW 에 의한 HW 의 점검 수준을 명시한다. SW 자체 점검은 SW 의 고장과 오류가 발생하였을 때 SW 에 의해 검출되고 보고되는 것을 의미한다. 자체 점검 방법 및 주기를 명시한다.

기능 테스트 요구사항 명시

시스템 안전 요구사항 명세서에서 요구되는 정도까지 주기적인 기능 테스트에 대한 요구사항을 기술하고, 전체 시스템 운영 과정에서 모든 안전 기능이 테스트할 수 있도록 한다.

필요 SW 분석

요구되는 시스템의 안전 무결성 수준 달성과 관련된 기능을 수행하거나 안전과 무관한 기능을 수행하기 위해 SW 가 필요하다면, 그 SW 를 명확히 식별한다.

하나 이상의 SW 시스템이 통합될 경우 필요하다. 이러한 통합은 SW 안전 요구사항 분석에 요구되는 분석의 규모를 급격히 확장시킨다. 분리된 시스템에 할당된 SW 요구사항들에 대한 상세한 분석은 안전과 관련된 통합 및 인터페이스 오류를 감소시킬 수 있다. 이것은 시스템 위험이 부분적으로 두 개 이상 의 SW 시스템에 구현되는 요구사항이 있을 때 특히 중요하다.

2.1.2 추적성 유지

시스템 요구사항 분석과 SW 요구사항 분석 사이의 일관성을 보장하기 위해 추적성을 수립하고 유지한다.

마. 확인 검증 활동

2.2.1 SW 요구사항 평가

SW 요구사항을 내용, 인터페이스, 추적 가능성 측면에서 평가한다. 정확성, 일관성, 완전성, 정밀성, 가독성, 테스트가능성 등에 대해 SW 요구사항을 평가한다. 그 기준은 다음과 같다.

기준	설명
정확성	SW 요구사항이 시스템의 전제조건과 제약 범위 안에서 SW 에 지정된 시스템 요구사항을 만족하고 있는지 검증, 확인한다. SW 요구사항이 표준, 참고문헌, 규칙, 정책. 물리적 법칙, 사업적인 규칙을 따르고 있는지 검증한다. 전문적 기술계, 프로토타입의 결과, 공학적 원리. 그 밖의 원리와 관련 있는 논리적, 자료의 흐름을 이용하여 연속적 인 상태와 상태의 변화를 확인한다. 자료의 흐름과 제어가 기능적, 성능적 요구사항을 만족하고 있는지 확인한다. 자료의 사용법과 형식을 확인한다.
일관성	모든 용어와 개념이 일관성 있게 문서화 되었음을 검증한다. 기능적 상호작용과 가정이 일관적이며 시스템과 획득자의 요구사항을 만족하는지 검증한다. SW 요구사항 사이의 내부적 일관성과 시스템 요구사항과의 외부적 일관 성이 있음을 검증한다.
완전성	다음 요소가 시스템의 전제조건과 제약사항 내에서 요구사항명세서에 존재하는지 검증한다: ・ 기능(예, 알고리즘, 상태/방식 정의, 입력/출력 확인, 예외상황 처리, 보고, 기록) ・ 프로세스 정의 및 스케줄링 ・ HW, SW, 사용자 인터페이스 설명 ・ 성능 평가 기준(예, 타이밍, 사이징, 속도, 용량, 정밀성, 정확도, 안전성, 보안성); ・ 주요한 형상 데이터 ・ 시스템, 장치, SW 제어(예, 초기화. 트랜잭션과 상태 감시, 자체

	테스트).
정밀성	논리적, 계산적, 인터페이스의 정확도(예, 반올림, 버림)가 시스템 환경에서 요구사항을 만족하는지 확인한다. 모형화된 물리적 현상이 시스템의 정밀성 요구사항과 물리적 법칙을 따르는지 확인한다.
가독성	문서가 사용자에게 읽기 쉽고, 이해하기 쉬우며, 모호하지 않음을 검증한다. 문서가 모든 두문자 약어, 기억술. 약어, 용어. 기호를 정의하고 있음을 검증한다.
테스트 가능성	SW 요구사항명세서의 요구사항을 확인하는 승인 기준이 존재하는지 검증한다.

인터페이스 분석

HW, 사용자, 운영자, 다른 시스템과 함께 SW 인터페이스의 요구사항이 정확 하고, 일관성 있고, 완전, 정밀하며 테스트 가능한지 검증, 확인한다. 분석 기준은 다음과 같다.

기준	설명
정확성	외부와 내부 시스템, SW 인터페이스 요구사항을 확인한다.
일관성	인터페이스 설명이 요구사항 사이에서 일관성이 있는지 검증한다.
완전성	각 인터페이스를 설명하고 있으며, 자료 형식과 성능 평가 기준(타이밍, 대역폭, 정밀성, 안전성, 보안성 등)을 포함하고 있는지 검증한다.
정밀성	각 인터페이스가 정확한 정보를 제공하는지 검증한다.
테스트 가능성	인터페이스 요구사항을 확인하기 위한 객관적인 승인 기준이 존재하는지 검증한다.

추적 가능성 분석

안전 관련 시스템의 확인에서 요구사항까지의 추적성을 고려하여야 하며, 수명주기 모든 단계에 걸쳐 추적성을 입증할 수 있는 수단이 제공되어야 한다. 추적 불가능한 자료들은 시스템의 안전 또는 무결성과 관련이 없다는 것을 보여주어야 한다.

SW 요구사항을 추적하여 시스템 요구사항을 식별하고, 시스템요구사항을 추적하여 SW 요구사항을 식별한다. 정확성, 일관성, 완전성, 정밀성과의 관련성을 분석한다.

기준	설명

정확성	각 SW 와 그것에 관련된 시스템 요구사항 사이의 관계가 정확한지 확인한다.
일관성	SW 와 시스템 요구사항 사이의 관계에 일관성 단계가 명시되어 있는지 검증한다.
완전성	모든 SW 요구사항이 시스템 요구사항이 해당 시스템 요구사항에 일치함이 증명된 상세 시스템 요구사항을 유추할 수 있는지 검증한다. SW 와 관련있는 모든 시스템 요구사항이 SW 요구사항을 유추할 수 있음을 검증한다.
정밀성	시스템 성능과 운영 특성이 유추된 해당 SW 요구사항에 의해서 정확하게 명시되었음을 확인한다.

2.2.2 SW 요구사항 명세서 작성

SW 요구분석서로부터 SW 요구사항명세서를 작성한다. SW 요구사항명세서는 SW 요구사항명세서에 기술된 모든 요구사항을 검증하고, 완성된 SW 에 대한 테스트를 설명하는데 사용된다. SW 요구사항 명세서는 각 필요한 기능에 대해 다음을 포함한 테스트 사례를 식별한다.

- 필요한 입력 신호와 그 순서 및 값
- 기대되는 출력 신호와 그 순서 및 값
- 성능 및 품질 측면을 포함하는 수락 기준

2.2.3 확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술해야 한다. 확인 검증 보고서는 다음을 포함해서 작성한다.

- SW 요구분석서, SW 설계 명세서 또는 SW 모듈 설계 명세서에 부합하지 않는 항목
- SW 품질 보증 계획과 부합하지 않은 항목
- 문제에 잘 맞지 않는 모듈, 데이터, 구조 그리고 알고리즘
- 검출된 오류 또는 부족한 부분
- 검증된 항목의 식별 및 형상

바. 안전 활동

2.3.1 SW 요구사항 안전 평가 수행

SW 안전 요구사항 분석은 SW 및 인터페이스 요구사항을 평가하여, 위험을 야기하는 오류나 결함을 식별하기 위한 것이다. 분석은 요구사항 위험 분석 기법을 활용하고, 다음 사항을 수행한다.

- SW 검증 및 확인 계획을 검토하고 합의한다.
- SW 요구분석의 적절성을 입증한 결과를 검토한다.

2.3.2 SW 안전 기록 작성

SW 안전 기록은 다음을 포함하여 작성한다.

- 안전 분석 결과
  - 위험상태(hazard)의 식별 및 관련된 SW 요구사항
  - SW 안전 설계 제약사항 및 가이드라인
  - SW 안잔 관련 테스트 요구사항과 테스트 계획 프로세스의 입력
  - 설계(권장/필수/제한), 코딩, 테스트 기법
- 의심되거나 확인된 안전 문제점
- 안전 테스트 결과

IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)

IEC61508-3 A.1 - Software safety requirements specification

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	준 정형 기법	IEC61508-3 B.7	R	R	HR	HR
1b	정형 기법		---	R	R	HR

2	시스템 안전 요구 사항과 소프트웨어 안전 요구 사항 간의 전방 추적 성		R	R	HR	HR
3	안전 요구 사항과 인식 된 안전 요구 사항 간의 역 추적 성		R	R	HR	HR
4	위의 적절한 기술 / 조치를 지원하는 컴퓨터 지원 사양 도구		R	R	HR	HR

- 소프트웨어 안전 요구 사항 명세는 항상 자연 언어로 된 문제에 대한 설명과 응용을 반영하는 필요한 수학 표기법을 활용할 수 있다.
- 소프트웨어 안전 요구 사항을 명확하고 정확한 명세를 위해 추가요구사항을 반영한다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.

IEC61508-3 A.1.2 : 시스템 안전 요구사항과 소프트웨어 안전 요구사항간 전방 추적성 (Forward traceability between the system safety requirements and the software safety requirements)

1. 목표

   수명주기별 일관성을 유지한다.

2. 설명

   소프트웨어가 안전 관련 시스템의 올바른 작동을 위한 요구 사항을 충족 시키려면 수명주기 단계간에 일관성을 유지해야한다. 핵심 개념은 추적성이며 이는 본질적으로 초기 단계에서 내려진 결정이 이후 단계에서 적절하게 구현되었는지, 그 결정이 이전 단계의 결정이 적절하게 반영되었는지 확인하는 분석이다. 전방 추적성은 일반적으로 요구 사항이 수명주기 단계에서 적절히 처리되는지 확인하는 것이다. 전방 추적은 안전수명주기의 여러 단계에서 유용하다.

   - 시스템 안전 요구 사항부터 소프트웨어 안전 요구 사항까지
   - 소프트웨어 안전 요구 사항 사양부터 소프트웨어 아키텍처까지
   - 소프트웨어 안전 요구 사항 사양부터 소프트웨어 설계까지
   - 소프트웨어 설계 명세에서부터 모듈 및 통합 테스트 명세까지
   - 하드웨어 / 소프트웨어 통합을위한 시스템 및 소프트웨어 설계 요구 사항부터 하드웨어 / 소프트웨어 통합 테스트 규격까지
   - 소프트웨어 안전 요구 사항 명세부터 소프트웨어 안전성 검증 계획까지

- 소프트웨어 안전 요구 사항 사양부터 소프트웨어 수정 계획까지
- 소프트웨어 설계 명세부터 데이터 검증을 포함하는 소프트웨어 검증 계획까지
- IEC 61508-3 의 8 절 요구사항부터 소프트웨어 기능 안전성 평가 계획까지

역 추적성은 어떤 요구 사항에 의해 코드 구현뿐 아니라 모든 구현 명확히 결정되었는지 확인하는 것이다. 결정이 명확하지 않다면 구현에서 불필요한 요소 포함으로 복잡성이 추가되고 안전 관련 시스템의 명확한 요구 사항을 다루지 않을 수 있다. 역 추적성은 안전 라이프 사이클의 여러 단계에서 유용하다.

- 안전 요구 사항부터 인지 된 안전 요구까지
- 소프트웨어 아키텍처부터 소프트웨어 안전요구사항 사양까지
- 소프트웨어 세부 설계부터 소프트웨어 아키텍처까지
- 소프트웨어 코드부터 소프트웨어 세부 설계까지
- 소프트웨어 안전성 검증 계획부터 소프트웨어 안전성요구사항 사양까지
- 소프트웨어 수정 계획부터 소프트웨어 안전 요구 사항 명세까지
- 데이터검증을 포함하는 소프트웨어 검증 계획부터 소프트웨어 설계 명세까지

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.2.11 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-58, C-6.18

☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.2 SW 안전 요구사항 명세 > 1.2.1 주요 요구사항 및 설명

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.1~ 2 .8

IEC61508-3 A.1.4 : Technique&Measure 를 지원하는 컴퓨터 지원 명세 도구 (Computer-aided specification tools to support appropriate techniques/measures above)

컴퓨터 지원 명세 도구

1. 목표

모호성 및 완전성 자동 탐지용이를 위해 공식사양 기술을 사용한다

2. 설명

이 기술은 일관성 및 완전성을 평가하기 위해 자동으로 검사 할 수 있는 데이터베이스 형식의 명세를 생성한다. 명세 도구는 지정된 시스템의 다양한 측면을 사용자에게

보여준다. 이 Technique 은 명세 작성뿐만 아니라 프로젝트 수명주기의 설계 및 기타 단계를 지원한다. 명세 도구는 다음의 하위 절로 분류된다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.2.4 를 참조한다.

특정 방법을 지향하지 않는 도구

1. 목표

사용자가 관련 파트간 프롬프트 및 링크를 제공하여 올바른 사양을 작성한다.

2. 설명

명세 도구는 사용자로부터 작업을 이어 받아 프로젝트 관리를 지원한다. 특정 명세 방법론을 적용하지 않으며, 방법에 대한 독립성은 사용자에게 많은 자유를 허용하지만 사양을 만들 때 필요한 전문적인 지원을 제공하지 않는다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.2.4 를 참조한다.

계층적 분석을 포함한 모델 지향적 절차

1. 목표

불완전성, 모호성 및 모순을 방지하기 위해 사용자가 다양한 추상화 수준에서 동작과 데이터 설명 간의 일관성을 보장함으로써 명세 작성을 지원한다.

2. 설명

이 방법은 다양한 수준의 추상화(정밀도)에서 원하는 시스템(구조화 된 분석)의 기능적 표현을 제공한다. 모호성 및 완전성의 평가는 동일한 레벨상의 2 개의 기능 유닛 (모듈) 사이뿐만 아니라 계층 레벨 사이에서도 가능하다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.2.4 를 참조한다.

개체-관계-속성 데이터 모델

1. 목표

시스템 내 개체와 개체간의 관계에 초점을 맞춤으로써 사용자의 명세 작성을 돕는다.

2. 설명

시스템은 객체의 모음과 객체간의 관계로 설명된다. 이 도구를 사용하면 시스템이 해석할 수있는 관계를 판별 할 수 있다. 일반적으로, 관계는 계층적 구조, 객체, 데이터 흐름, 데이터 간의 관계 및 데이터의 대상이 되는 프로세스로 설명될 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.2.4 를 참조한다.

인센티브 및 답변

1. 목표

자극 – 반응 관계를 식별하여 사용자가 명세 작성을 돕는다.

2. 설명

시스템 객체 간의 관계는 "인센티브"와 "답변" 표기법으로 지정된다. 간단하고 쉽게 확장된 관계, 특성 및 구조를 나타내는 언어가 사용된다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.2.4 를 참조한다.

IEC61508-3 B.7 - Semi-formal methods

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	논리 / 기능 블록 다이어그램		R	R	HR	HR
2	시퀀스 다이어그램		R	R	HR	HR
3	데이터 흐름도		R	R	R	R
4a	유한 상태 기계 / 상태 전이 다이어그램		R	R	HR	HR
4b	시간 페트리 그물		R	R	HR	HR
5	엔터티 관련 특성 데이터 모델		R	R	R	R
6	메시지 시퀀스 차트		R	R	R	R

7	의사 결정 / 진리표		R	R	HR	HR
8	UML		R	R	R	R

- 논리/기능 블록 선도 및 순서도는 IEC 61131-3 을 참조한다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다. 대체 가능한 기술/조치는 번호 뒤에 오는 문자로 표시한다. 그 중 하나만 만족하면 된다.

IEC61508-3 B.7.4a : 유한 상태 기계 / 상태 전이 다이어그램 (Finite state machines/state transition diagrams)

1. 목표

   시스템의 제어 구조를 모델링, 확인, 지정 또는 구현한다.

2. 설명

   많은 시스템은 상태, 입력 및 동작과 관련하여 설명된다. 따라서, 상태 S1 에서, 입력 I 를 수신하면 시스템은 동작 A 를 수행하고 상태 S2 로 이동할 수 있다. 모든 상태에서 모든 입력에 대해 시스템의 동작을 설명함으로써 시스템을 완전하게 설명 할 수 있다.
   시스템의 결과 모델은 유한 상태 기계(finite state machine) 라고 불린다. 종종 시스템이 한 상태에서 다른 상태로 이동하는 방식을 나타내는 소위 상태 전이도 또는 차원이 상태 및 입력이고 행렬 셀에 동작 및 새 상태가 포함 된 행렬로 그려진다. 시스템이 복잡하거나 자연스러운 구조를 갖는 경우 계층화 된 유한 상태 기계에 이를 반영 할 수 있다. 상태차트는 중첩 된 상태가 허용되는 상태 전이도 유형이다. 상태 전이 표기법의 표현력에 높여주지만 안전 관련 시스템에서 복잡성을 높일 수 있습니다. 상태 차트에는 공식 명세가 있습니다. 상태 전이도는 전체 시스템이나 그 안에있는 어떤 객체나 요소에 적용될 수 있고 유한 상태 기계로 표현 된 명세 또는 설계를 확인할 수 있다.
   - 완전성 (시스템 또는 객체는 모든 상태의 모든 입력에 대해 동작 및 새로운 상태를 표시함)
   - 일관성 (각 상태 / 입력 쌍에 대해 하나의 상태 전이만 가능함)
   - 도달 가능성 (입력의 순서에 따라 하나의 상태에서 다른 상태로 갈 수 있는지의 여부)
   - 무한 루프 또는 막 다른 상태가 없음.

   이는 시스템에 중요한 속성이며 점검를 지원하는 도구는 개발되어 유한 상태 오토마타를 기반으로하는 다양한 모델을 사용할 수 있다. 유한 상태 기계 구현을 검증하거나 유한

상태 기계 모델을 애니메이션화하기위한 테스트 케이스의 자동 생성을 허용하는 알고리즘도 존재한다.
상태 전이도와 상태 차트는 다이어그램을 그려서 검사 할 수 있는 도구와 시스템을 구현하는 코드를 생성하는 도구로 지원된다. 고장 확률 계산에 사용될 수 있다 (B.6 및 C.6 참조).

3. 비고
이 측정 기법 및 기준은 IEC 61508-7 의 표 B.2.3.2 를 참조한다.

4. 분야별 가이드 참조 위치
☞ 철도 가이드: 부록 > B-27, C-6.1
☞ 자동차 가이드: * PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.1 요구사항 분석 > 상태전이 테스팅(State Transition Testing)
☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.7.4b : 시간 페트리 네트 (Time Petri nets)

1. 목표
시스템 동작 관련 측면을 모델링하고 분석 및 재설계를 통해 안전 및 작동 요구 사항을 평가하고 개선시킨다.

2. 설명
페트리 네트는 유한 상태 오토마타의 특별 경우인데, 동시성을 보이고 비동기적인 동작을하는 시스템에서 정보를 표현하고 흐름을 제어하는데 적합한 그래프 이론 모델의 클래스에 속한다. 페트리 네트는 장소와 전환점의 네트워크다. 장소는 "marked" 또는 "unmarked" 이며, 전환은 모든 입력 위치가 표시 될 때 "enabled"가 된다. 이 기능을 사용하면 "fire"할 수 있지만, 입력 전환 지점이 표시되지 않고 전환 지점의 각 출력 위치가 대신 표시된다. 잠재적 위험은 모델의 특정 상태로 표현된다. 페트리 네트 모델은 시스템의 타이밍 기능을 허용할 수 있다. "고전적인" 페트리 네트는 제어 흐름 측면에 집중하고 있지만 데이터 흐름을 모델에 통합하기 위한 몇 가지 확장이 제안되었다. 실패 확률 계산을 위해 몬테카를로 시뮬레이션을 수행을 효율적으로 지원한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.2.3.3 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-55

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.7.5 : 엔터티 관련 특성 데이터 모델 (Entity-relationship-attribute data models)

1. 목표

   사용자가 시스템 내의 개체와 개체간 관계에 초점을 맞춤으로써 명세작성을 돕는다.

2. 설명

   원하는 시스템은 객체와 그 관계의 모음으로 설명됩니다. 이 도구를 사용하면 시스템이 해석 할 수 있는 관계를 구별한다. 일반적으로 관계는 객체의 계층 적 구조, 데이터 흐름, 데이터 간 관계 및 특정 제조 프로세스의 적용을 받는 데이터를 설명한다. 사용자에 대한 검사 기능 및 지원은 다양한 관계에 따라 달라진다. 반면에, 많은 표현 가능성은 이 기법의 적용을 복잡하게 만든다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.2.4.4 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.7.6 : 메시지 시퀀스 차트 (Message sequence charts)

1. 목표

요구 사항 및 소프트웨어 아키텍처 설계를 포함한 소프트웨어 개발의 초기 설계 단계에서 시스템 요구 사항을 파악할 수 있다. UML 에서는 "System Sequence Diagram"이 사용된다.

2. 설명

메시지 시퀀스 차트는 시스템 앤터 (인간, 컴퓨터 시스템, 소프트웨어 요소 또는 객체 일 수 있음) 사이에서 발생하는 의사 소통 측면에서 시스템의 동작을 설명하기위한 메커니즘이다. 디자인 단계). 각 액터에 대해 다이어그램에 수직 "lifeline"이 그려지고 lifeline 사이의 화살표는 메시지를 나타낸다. 메시지 수신시 조치는 선택 사항으로 다이어그램에 상자로 표시된다. 시나리오 모음 (바람직한 동작과 바람직하지 않은 동작 모두를 설명)은 필요한 시스템 동작의 명세로 구축되며 시나리오에는 여러 용도가 있다. 최종 사용자에게 시스템 동작을 보여주기 위해 애니메이션을 적용 할 수 있다. 시스템의 실행 가능한 구현으로 변형 될 수 있고, 테스트 데이터의 기초가 될 수 있다. UML 은 시나리오를 나누고 반복 할 수 있는 선택 및 반복 구문의 형태로 Message Squence Chart 의 기본 개념을 확장하여보다 조밀 한 방법을 제공한다. 상위 레벨 시퀀스 다이어그램에서 참조 할 수 있는 하위 다이어그램을 정의 할 수 있으며 타이머 및 외부 이벤트도 표현할 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.14 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: -

☞ 자동차 가이드: -

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.7.8 : UML

1. 목표

복잡한 시스템에서 원하는 동작을 모델링하기 위한 포괄적인 표현을 제공한다.

2. 설명

UML 은 이름에서 알 수 있듯이 요구 사항 및 디자인 표기법 모음이다. 소프트웨어 개발에 대한 포괄적인 지원을 제공한다. 일부의 UML 은 다른 메소드에서 처음 도입 된 표기법 (예 : 시스템 시퀀스 다이어그램 및 상태 전이 다이어그램)과 기존 표기법을 제공한다. UML 모델로부터 코드를 생성하며 명세 및 설계에 적용 할 수있는 UML 표기법 안전 관련 시스템은 다음과 같다.

- 클래스 다이어그램
- 사용 사례
- 활동 다이어그램
- 상태 전이 다이어그램 (상태 차트)
- 시스템 시퀀스 다이어그램

다른 UML 표기법인 소프트웨어 아키텍처 디자인, 상태 전이도는 C.2.14 의 B.2.3.2 및 시스템 시퀀스 다이어그램에 설명되어 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.3.12 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-70, C-6.1, C-6.2

   ☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.4 SW 아키텍처 설계 > 1.4.2 소프트웨어 아키텍처 명세(안전측면) 예제 > 1.4.2.3 동적 설계, PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.14 환경조건과 운영 유즈케이스 분석

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

사. 사용 양식

- [SR-D-01] SW 요구사항 명세서
- [SR-D-02] SW 안전 기록
- [SR-D-03] SW 요구사항 안전성분석 보고서

아. 적용 기법

- [SR-T-01] SW 요구사항 명세 기법
- [SR-T-02] SW 요구사항 위험원 분석 기법

3. SW 설계(SD) 단계

Software Design Phase

<table>
<tr><td>목적</td><td>SW 요구분석서의 요구사항들에 대한 SW 적인 구조를 개발하고 설계한다.</td></tr>
<tr><td>시작 기준</td><td>SW 요구사항이 확정되고 시스템구조설계서가 준비됨</td></tr>
<tr><td>활동</td><td>
• SW 요구사항 명세서
• 시스템 안전성 요구사항 명세서
• 시스템 구조 기술서
• SW 개발계획서
• SW 확인검증계획서
• SW 안전계획서
<table>
<tr><th>개발 활동</th><th>확인 검증 활동</th><th>안전 활동</th></tr>
<tr><td>3.1.1 SW 구조 설계
3.1.2 SW 설계명세서 정의
3.1.3 SW 통합테스트 명세서 작성
3.1.4 SW 모듈설계 명세서 작성
3.1.5 SW 모듈테스트명세서 작성</td><td>3.2.1 SW 설계 평가
3.2.2 통합테스트명세서 검증
3.2.3 SW 모듈 테스트명세서 검증
3.2.4 확인검증 작성</td><td>3.3.1 SW 설계 안전평가 수행
3.3.2 SW 모듈안전평가 수행
3.3.3 SW 안전기록 작성</td></tr>
</table>
</td></tr>
</table>

	[SD-D-01] SW 설계 명세서 [SD-D-02] SW 단위시험 계획서 [SD-D-03] SW 통합시험 계획서 [SD-D-04] SW 코딩 메뉴얼 [SD-V-01] SW 설계 안전성분석 보고서
확인	검증담당자에 의해 SW 구조 및 설계가 검증된다. 안전담당자가 SW 설계의 안전을 분석한다.
종료기준	SW 구조명세서가 작성됨 SW 설계명세서가 작성됨 SW 모듈명세서가 작성됨 SW 설계 확인 검증 보고서 및 SW 안전 기록이 작성됨

가. 목 적

SW 요구사항명세서의 요구사항에 대해 SW 구조를 개발하고, 시스템 구조에 대한 SW 의 요구사항들을 검토하며, 안전성에 대한 시스템과의 상호 관계에 대한 증명 및 평가를 수행하며, 사전에 정의되지 않은 사항에 대한 설계 방안을 선택하기 위한 것이다. 또한 SW 구조 설계를 SW 단위가 되는 각 SW 모듈을 위한 상세한 설계로 분해하고 SW 단위 및 인터페이스를 SW 모듈 설계서로 문서화한다.

나. 책임과 권한

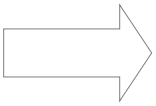
역할 / 책임	설계담당자	품질담당자	안전담당자

개발활동	3.1.1 SW 구조 설계서 정의	R		A
	3.1.2 SW 설계명세서 정의	R		A
	3.1.3 SW 통합 테스트 명세서 작성	R		A
	3.1.4 SW 모듈설계 명세서 작성	R		
	3.1.5 SW 모듈테스트명세서 작성	R		
검증활동	3.2.1 SW 설계 평가	S	R	
	3.2.2 통합 테스트명세서 검증	S	R	
	3.2.3 SW 모듈테스트명세서 검증	S	R	
	3.2.4 확인 검증 보고서 작성	S	R	
안전활동	3.3.1 SW 설계 안전 평가 수행	S	S	R
	3.3.2 SW 안전 기록 작성	S	S	R

(Responsible:담당, Approve:승인. Support:지원)

다. 입력물 및 출력물

- SW 요구사항 명세서
- 시스템 안전성 요구사항 명세서
- 시스템 구조 기술서
- SW 개발계획서
- SW 확인검증계획서
- SW 안전계획서



[SD-D-01] SW 설계 명세서
[SD-D-02] SW 단위시험 계획서
[SD-D-03] SW 통합시험 계획서
[SD-D-04] SW 코딩 메뉴얼
[SD-V-01] SW 설계 안전성분석 보고서

라. 개발 활동

3.1.1 SW 구조(아키텍처) 설계

SW 요구사항을 구현하는 SW 구성요소들을 기술하는 SW 설계명세서를 작성한다.

SW 구성요소 식별

SW 구조 명세서는 모든 SW 구성 요소를 식별하고, 이 구성 요소들에 대해 다음을 식별한다.

- 이 구성요소들이 새로운 것인지, 존재하던 것인지 또는 소유권이 있는 것인지
- 이 구성요소들이 사전에 확인되었는지, 확인되었다면 확인 조건이 무엇인지
- 구성요소에 대한 SW 안전 무결성 수준

주요 SW 구성요소의 예에는 데이터 저장 및 접근 (예: 데이터베이스), 의사소통 절차와 수단, 사업 로직 및 사용자 인터페이스를 포함한다.

SW 구조 및 인터페이스 정의

SW 요구사항을 SW 구조 설계로 변환할 때 정의된 기준에 따라 SW 구조 대안들을 평가하여야 한다. 현 SW 구조가 선정된 논리를 기록하여야 한 다. 평가 기준에는 품질 특성 (모듈화, 유지관리성, 확장성, 가변성, 신뢰성, 보안성 및 사용의 편리성) 및 개발-구매-재사용 분석의 결과가 포함될 수 있다.

SW 구조 명세서는 SW 구성요소 사이의 외부 및 내부 인터페이스를 명세화하고 문서화한다.

COTS 제약사항 파악 확인 검증

SW 안전 무결성 수준이 0 인 경우에는 COTS SW 는 별도의 예방책 없이 허용된다.

만약 COTS SW 가 SW 안전 무결성 수준 1~2 에서 사용되면, 그 SW 는 반드시 SW 확인 프로세스에 포함되어야 한다.

만약 COTS SW 가 SW 안전 무결성 수준 3~4 에서 사용되면, 다음 예방책을 취해야 한다.

- COTS SW 는 확인 테스트에 포함되어야 한다.
- COTS SW 의 고장 가능성에 대한 분석이 수행되어야 한다.
- COTS SW 의 고장을 검출하고, 이 고장으로부터 시스템을 보호하기 위한 전략이 정의되어야 한다.
- 보호 전략은 확인 테스트의 대상이 되어야 한다.
- 오류 기록(error logs)이 존재하고, 평가되어야 한다.
- 가능하다면 COTS SW 의 단순한 기능만을 사용하여야 한다.

기존 SW 제약사항 파악

이미 개발된 SW 가 설계에 포함되는 경우에는 이러한 SW 들은 명확하게 식별되고 문서화한다. SW 구조 명세서는 SW 요구분석서와 SW 안전 무결성 수준을 만족시키는데 적절함을 정당화해야 한다. SW 에 대한 변경이 시스템의 나머지에 미치는 영향은 재조사 및 재평가를 필요로 하는지 결정하기 위해 주의 깊게 고려되어야 한다. 재검증, 재확인, 재평가 중이 아닌 다른 모듈과의 인터페이스 명세가 이루어지고 있다는 증거가 있어야 한다.

가능하다면, 본 표준에 따라 개발된 기존의 검증된 SW 모듈은 설계에서 이용되어야 한다.

SW 개발 전략 및 개발기법 식별

SW 구조 명세서는 SW 안전 무결성 수준에서 요구하는 정도까지 SW 개발 전략을 식별한다.

SW 구조 명세서는 결함을 회피하고 결함을 처리하는 전략간 균형을 맞춘다.

SW 구조 명세서는 요구되는 SW 안전 무결성 수준에서 SW 요구분석서를 만족하기 위해 선택한 기법 및 수단을 명시한다.

3.1.2 SW 설계명세서 정의

SW 설계명세서는 SW 모듈 설계 명세서와 SW 모듈 테스트 명세서를 가지는 모듈들에 근거하여 SW 설계를 기술한다.

SW 설계 명세서는 다음을 포함한다.

- SW 구조 및 안전 무결성 수준으로 추적되는 SW 구성요소
- 환경과 함께 SW 구성요소의 인터페이스
- SW 구성요소 간의 인터페이스
- 데이터 구조
- 요구사항의 구성요소에의 분할
- 주 알고리즘과 순서
- 다이어그램

3.1.3 통합 테스트명세서(계획서) 작성

각 SW 구성요소(예, 프로그램 단위, 모듈)가 서로 통합하여 전체적인 SW 가 될 수 있도록 SW 가 SW 요구사항과 설계를 올바르게 구현했는지 확인 하기 위한 통합 테스트를 계획한다. SW 통합시험 계획서는 다음 사항을 반드시 문서화해야 한다.

- 테스트 케이스
- 수행되는 테스트의 형태
- 테스트 환경, 도구들, 설정 그리고 프로그램들
- 완성도 높은 판단을 위한 테스트 기준
- SW 설계가 요구사항을 충분히 반영하고 있음을 확인하기 위해 인터페이스 분석 및 추적 가능성을 분석결과

인터페이스 분석

정확성, 일관성, 완전성, 정밀성, 테스트 가능성에 대해 HW, 사용자, 운영자, SW 와 다른 시스템과의 SW 설계 인터페이스를 검증하고 확인한다. 그 기준은 다음과 같다:

- 정확성: 시스템 요구사항 범위 내에서 외부, 내부의 SW 인터페이스 설계를 확인한다.
- 일관성: 인터페이스 설계가 일관성이 있는지 검증한다.
- 완전성: 각 인터페이스가 데이터 형식과 성능 측정 기준(예, 타이밍, 대역폭, 정밀도, 안전성, 보안성)을 정확하게 설명하고 포함하는지 검증한다.
- 정밀성: 각 인터페이스가 요구하는 정확한 정보를 제공하는지 검증한다.
- 테스트성: 인터페이스 설계를 확인할 수 있는 객관적인 승인 기준이 존재하는지 검증한다.

추적 가능성 분석

설계 요소로써 요구사항을 유추하고, 요구사항을 가지고 설계요소를 유추한다. 정확성, 일관성, 완전성과의 관계를 분석한다. 그 기준은 다음과 같다: 요구사항추적표를 활용한다.

- 정확성: 설계 요소와 SW 요구사항의 연관성을 확인한다.
- 일관성: 설계 요소와 SW 요구사항의 연관성이 일관성이 있는지 검증한다.
- 완전성: 모든 설계 요소가 SW 요구사항으로부터 유추 가능한지 검증한다.
- 모든 SW 요구사항이 설계 요소에 포함되어 있는지 검증한다.

3.1.4 SW 모듈 설계 명세서 작성

SW 구조 설계를 모든 SW 단위인 각 SW 모듈을 위한 상세한 설계로 분해한다. SW 단위 및 인터페이스를 SW 단위시험 계획서로 문서화한다.

모듈당 하나의 SW 모듈 설계 명세서를 작성한다. SW 모듈 설계 명세서에는 다음 내용을 반드시 포함한다.

- 최하위 SW 구성 요소 (적용되는 표준 규격에서 언급된 모듈) 와 상위 레벨 구성 요소의 추적성 확인
- 외부 요인과의 상세 한 인터페이스와 다른 모듈들의 상세한 입력과 출력들
- 대상의 안전 무결성 수준
- 상세한 알고리즘과 데이터 구조

각각의 SW 모듈 설계 명세서는 해당하는 모듈의 코딩 시 다른 명세서의 내용 참조없이 독립적으로 적용이 가능해야 한다.

3.1.5 SW 모듈 테스트 계획 작성

각 SW 모듈을 테스트하기 위 한 SW 단위시험 계획서를 작성한다. 작성 시 다음을 고려한다.

- 테스트 결과에 대한 설명 과 각 모듈이 SW 단위시험 계획서의 요구사항을 만족시키는지 여부
- 모든 소스코드 명령문이 적어도 한번은 수행됨을 보여주면서, 각 모듈에 대해 테스트 커버리지에 대한 설명이 제공되어야 함
- 평가할 수 있는 형식으로 되어 있어야 함
- 테스트케이스 및 그 결과는 후속 분석을 위해 기계가 읽을 수 있는 언어 형태로 기록되어야 한다. 테스트는 반복 가능한 것이 바람직하며. 가능하다면 자동화된 수단에 의해 수행되도록 함

마. 확인 검증 활동

3.2.1 SW 설계 평가

SW 설계에 대한 내용을 평가하고 추적 가능성을 평가한다.

SW 설계 내용 평가

정확성, 일관성, 완전성. 정밀성, 가독성, 테스트 가능성에 대한 모듈을 평가한다. 작업 기준은 다음과 같다:

- 정확성: 원시 코드의 구성요소가 SW 설계를 만족하는지 검증, 확인한다. 원시코드의 구성요소가 표준, 참고문헌, 규약, 정책, 물리적 법칙, 사업적 규칙을 따르고 있는지 검증한다. 전문적 기술계, 프로토타입의 결과, 공학적 원리, 그 밖의 원리와 관련 있는 논리적, 자료의 흐름을 이용하여 원시 코드의 연속적인 상태와 상태의 변화를 확인한다. 자료의 흐름과 제어가 기능적, 성능적 요구사항을 만족하는지 확인한다. 자료의 사용법과 형식을 확인한다. 코딩 방법과 표준의 타당성을 평가한다.
- 일관성: 모든 용어와 코드가 일관성 있게 문서화 되었는지 검증한다. 원시 코드 구성요소 사이에 내부적 일관성이 있는지 검증한다.
- 완전성: 다음 요소가 시스템의 전제조건과 제약 안에서 SDD 에 존재하는지 검증한다.
  - 기능(예, 알고리즘, 상태/방식 정의, 입력/출력 확인, 예외 처리, 보고, 기록)
  - 정의와 스케줄링 처리
  - HW, SW, 사용자 인터페이스 설명
  - 성능 측정 기준(타이밍, 사이징, 속도, 용량, 정 1 기성, 정확도, 안전성, 보안성)
  - 주요한 형상 자료
  - 시스템, 장치, SW 제어(예, 초기화. 트랜잭션, 상태 감시. 자체 테스트)
- 정밀성: 논리적, 계산적, 인터페이스의 정확도(예, 반올림, 버림)가 시스템 환경에서 요구사항을 만족 하는지 확인한다. 모형화된 물리적 현상이 시스템의 정밀성 요구사항과 물리적 법칙을 따르고 있는지 확인한다.
- 가독성: 문서가 사용자에게 읽기 쉽고, 이해하기 쉬우며, 분명한지 검증한다. 문서가 문자 약어, 기억술 약어, 용어, 기호를 정의하고 있음을 검증한다.
- 테스트 가능성: 각 SW 설계 요소와 시스템 설계를 확인하는데 필요한 객관적 승인 기준을 검증한다. SW 설계 요소가 객관적 승인 기준을 통하여 테스트 가능한지 검증한다.

추적 가능성 분석

상위 수준의 설계 요소(구조)와 하위 수준(모듈)의 설계요소간 추적성을 분석한다. 정확성, 일관성, 완전성과의 관계를 분석한다. 그 기준은 다음과 같다.

- 정확성: 설계 요소 간의 연관성을 확인한다.
- 일관성: 상위 수준 설계 요소와 하위 수준 설계 요소가 연관성이 일관성이 있는지 검증한다.
- 완전성: 하위 수준 설계 요소가 상위 수준 설계요소로부터 유추 가능한지 검증한다. 상위 수준 설계 요소가 하위 수준 설계 요소에서 유추할 수 있는지 검증한다.

3.2.2 통합 테스트 명세서(계획서) 검증

통합 테스트 계획이 다음 기준을 만족하는지 확인한다.

- 시스템 요구사항을 유추할 수 있는지 여부
- 시스템 요구사항과 외부적 일관성이 있는지 여부
- 내부적 일관성이 있는지 여부
- SW 요구사항의 테스트 범위
- 테스트 표준과 방법의 타당성
- 예상하는 결과와의 일치
- SW 품질 테스트의 성공 가능성
- 운영과 유지관리의 성공 가능성(예, 사용자 요구사항을 따라서 운영되거나 유지관리될 수 있는 용량)

3.2.3 SW 모듈 테스트 명세서(계획서) 검증

개발자의 구성요소 테스트명세서가 테스트 문서의 목적, 형식, 내용을 정의하고 있는 프로젝트계획을 따르고 있는지 검증한다. 개발자의 구성요소 테스트명세서가 다음 기준을 만족하는지 확인한다.

- 단위 프로그램 사이에 내부적으로 일관성이 있는지 여부

- 단위 프로그램의 테스트 범위

모듈 테스트를 위한 테스트 사례 및 절차가 모듈테스트계획서의 기준을 만족하였는지 확인한다.

3.2.4 확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술해야 한다. 확인 검증 보고서는 다음을 포함해야 한다.

- SW 요구분석서, SW 설계 명세서 또는 SW 모듈 설계명세서에 부합하지 않는 항목
- SW 품질 보증 계획과 부합하지 않은 항목
- 문제에 잘 맞지 않는 모듈, 데이터, 구조 그리고 알고리즘
- 검출된 오류 또는 부족한 부분
- 검증된 항목의 식별 및 형상

바. 안전 활동

SW 설계에서의 안전과 관련된 부분이 요구사항을 정확히 구현하고 있으며 새로 운 위험을 초래하지 않음을 검증한다.

3.3.1 SW 설계 안전 평가 수행

SW 구조는 어플리케이션에서의 안전성 부분을 최소화시켜야 한다. 서로 상이한 안전 무결성 수준을 갖는 SW 의 구성 요소의 결함은 두 SW 가 서로 독립적이라는 명확한 근거가 없는 한 높은 안전 무결성 수준을 기준으로 처리되어야 한다.

이전의 주요 작업 보고서의 중요도 분석 결과를 설계를 이용하여 검토하고 갱신한다. 구 현 방법론과 인터페이스 관련 기술은 주어진 SW 의 요소(요구사항, 모듈, 함수, 하위 시스템, 다른 SW)에 지정된 SW 무결성 수준을 상향 또는 하향 조절할 수 있다. 수정된 SW 무결성 수준으로 인하여 어떠한 불일치나 예상치 못한 SW 무결성이 발생하지 않음 검증한다.

다음의 분석을 수행한다.

- 논리 분석을 통해 SW 설계에서의 안전 관련 수식, 알고리즘, 제어 로직을 평가한다.
- 데이터 분석에서는 SW 설계 내의 각 데이터 항목에 대한 설명과 사용 의도를 평가한다. 이 분석을 통해 데이터의 구조와 사용 의도가 위험 상태를 발생시키지 않음을 보장한다. 데이터 구조는 데이터 종속성을 위해 평가되어 위험을 고립, 분할, 안전에 영향을 미치는 결함 보유 이슈 및 또는 진이를 할 수 있도록 한다.
- 인터페이스 분석에서는 내외부 시스템의 타 구성요소와 SW 컴포넌트의 안전관련 인터페이스를 적절히 설계함을 검증한다. 인터페이스와 관련된 주된 관심 분야는 적절히 정의된 프로토콜 및 통제 및 데이터 연결이다. 외부 인터페이스는 설계에 서의 통신 프로토콜이 인터페이스 요구사항과 호환가능한지를 보여주기 위해 분석된 다. 인터페이스와 연관된 위험 상태는 어느 시점에서의 그 상태에 의해 정의되는 시스템 컨텍스트 및 환경 컨텍스트에 관련되어 있다. 인터페이스 분석에서는 이 시스템 및 환경 컨텍스트를 문서화하여야 한다. 인터페이스 분석은 시스템 수준에서의 위험 상태의 근원과 추가 분석이 필요한 분야를 알려준다.
- 제약사항 분석에서는 요구사항에 의해 선택된 설계에 부과된 제약 사항과 실세계 제약 사항의 안전을 평가한다.
- 기능 분석에서는 각 안전관련 SW 요구사항이 포함되었고, 각 SW 구성요소에 적절한 수준의 중요도가 할당됨을 보장한다.
- SW 구성요소 분석에서는 안전과 관련 없는 SW 구성요소를 검사하여 이 구성요소가 위험 상태를 야기하지 않음을 보장한다.

- SW 안전 요구사항 분석에서의 타이밍 및 사이징 분석 결과에 기반을 두어 타이밍 및 사이징 예측치를 수립하여 운영 환경 평가를 한다.
- 안전관련 SW 구성요소를 위해 신뢰성 예측이 이루어질 수도 있다. SW 안전 요구사항에 정의된 수용 가능한 위험 수준으로써 신뢰성 목표를 수립할 수 있다.

이전의 작업 보고서를 이용하여 위험 분석을 검토하고 갱신한다. 위험 요소를 제거, 감소, 약화시키기 위한 권장사항을 제시한다.

3.3.2 SW 모듈 안전 평가 수행

SW 모듈위험 분석기법을 활용하여 모듈 설계 및 관련된 자료 요소가 주요한 상위 설계 요소를 올바르게 구현하고, 어떠한 새로운 장애도 발생시키지 않음을 검증한다. 이전의 작업 보고서를 이용하여 위험 분석을 검토하고 갱신한다. 위험 요소를 제거, 감소, 약화시키기 위한 권장사항을 제시한다.

3.3.3 SW 안전 기록 작성

SW 안전 기록에는 다음을 포함한다.

- 안전분석 결과
- 의심되거나 확인된 안전 문제점
- 안전 테스트 결과

IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)

IEC61508-3 A.2 - Software design and development - software architecture design

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
	Architecture and design feature					

1	오류 감지		---	R	HR	HR
2	코드 감지 오류		R	R	R	HR
3a	오류 주장 프로그래밍		R	R	R	HR
3b	다양한 모니터 기술 (모니터와 모니터 기능이 동일한 컴퓨터에서 독립적 임)		---	R	R	
3c	다양한 모니터 기술 (모니터 컴퓨터와 모니터링되는 컴퓨터가 분리되어 있음)		---	R	R	HR
3d	다양한 중복성, 동일한 소프트웨어 안전 요구 사항 사양 구현		---	---	---	R
3e	기능적으로 다양한 중복, 다른 소프트웨어 안전 요구 사항 사양 구현		---	---	R	HR
3f	역방향 복구		R	R	---	NR
3g	무국적 소프트웨어 설계 (또는 제한된 상태 설계)		---	---	R	HR
4a	장애 복구 메커니즘 다시 시도		R	R	---	---
4b	단계별 성능저하		R	R	HR	HR
5	인공 지능 - 오류 수정		---	NR	NR	NR
6	동적 재구성		---	NR	NR	NR
7	모듈 방식	IEC61508-3 B.9	HR	HR	HR	HR
8	신뢰할 수 있고 검증 된 소프트웨어 요소의 사용 (있는 경우)		R	HR	HR	HR
9	소프트웨어 안전 요구 사항 사양과 소프트웨어 아키텍처 간의 포워드 추적성		R	R	HR	HR
10	소프트웨어 안전 요구 사항 사양과 소프트웨어 아키텍처 간의 역 추적성		R	R	HR	HR
11a	구조 다이어그램	IEC61508-3	HR	HR	HR	HR

		A.2.1				
11b	준 정형 기법	IEC6150 8-3 B.7	R	R	HR	HR
11c	정식 디자인 및 개선 방법		---	R	R	HR
11d	자동 소프트웨어 생성		R	R	R	R
12	컴퓨터 지원 사양 및 설계 도구		R	R	HR	HR
13a	최대 사이클 시간을 보장하는 주기적 동작		R	HR	HR	HR
13b	시간 트리거 아키텍처		R	HR	HR	HR
13c	이벤트 중심, 최대 응답 시간 보장		R	HR	HR	-
14	정적 리소스 할당		-	R	HR	HR
15	공유 리소스에 대한 액세스의 정적 동기화		-	-	R	HR

- 결함 허용성(장애 제어)에 관한 이 표의 측정 방법은 IEC 61508-2 에 기술된 프로그래밍 가능한 전자 장치의 하드웨어에 대한 장애 제어와 구조에 대한 요구 사항을 고려한다.
- 그룹 13 조치는 안전 타이밍 요구 사항이 있는 시스템 및 소프트웨어에만 적용된다. 14 번이 반드시 적용될 필요는 없다.

  컴파일러는 다음을 만족해야 한다.

  a) 모든 동적 변수와 객체에 대해 충분한 메모리가 런타임 이전에 할당되거나 메모리 할당 오류의 발생

  b) 응답 시간이 요구 사항을 만족함
- 4a 를 측정한다. 결함 복구를 위한 재시도는 모든 SIL 에서 적절하지만 재시도 횟수에 제한을 설정해야 한다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다. 대체 가능한 기술/조치는 번호 뒤에 오는 문자로 표시한다. 그 중 하나만 만족하면 된다.

IEC61508-3 A.2.1: 구조적 방법(Structured diagrammatic methods)

일반 사항

3. 목표

구조적 방법의 주요 목적은 라이프 사이클의 초기 부분에 중점을 두어 소프트웨어 개발 품질을 향상시키는 것이다. 이 방법은 정확하고 직관적인 절차와 표기법을 통해 논리적인

순서와 구조적인 방식으로 요구사항 및 구현 특징을 결정하고 문서화하는 것을 목표로 한다.

4. 설명

구조적 방법들은 범위가 존재한다. 전통적인 데이터 처리 및 트랜잭션 처리 기능을 위해 설계된 방법들도 있고, 프로세스 제어 및 실시간 응용프로그램에 더 중점을 두는 방법도 있다(보다 안전성이 중요한 경향이 있음). 구조적 방법은 기본적으로 문제 또는 시스템을 체계적으로 인식하고 분할하기 위한 "사고 도구(thought tool)"이다. 그 주요 특징은 다음과 같다.

- 큰 문제를 다루기 쉬운 단계로 나누기 위한 사고의 논리적 순서.
- 요구되는 시스템 및 환경을 포함하는 전체 시스템의 분석 및 문서화
- 요구되는 시스템 내의 데이터와 기능의 분해
- 체크리스트, 즉 일종의 분석이 필요한 목록
- 지식수준이 낮은 오버헤드
- 간단하고, 직관적이며, 실용적임
- 주로 의도된 시스템의 도식화된 모델을 개발하는 데 중점을 두고 있으며, 전체적인 방법에 대한 CASE 도구를 지원함

문제 및 시스템 개체(예: 프로세스 및 데이터 흐름)를 분석하고 문서화 하는 것을 지원하는 표기법은 정확해야 하나, 이 개체들에 의해 수행되는 처리 기능을 표현하기 위한 표기법은 비공식적인 경향이 있다. 일부 방법은 공식적인 표기법(예: 정규식 또는 유한 상태 기계)을 부분적으로 사용하기도 한다. 증가된 정밀도는 오해의 범위를 줄여주며, 자동 프로세싱을 위한 범위를 제공한다. 구조화된 표기법의 또 다른 이점은 가시성으로, 사용자가 강력히 알고 있지만 설명하기 어려운 지식과 비교하여 사용자가 명세나 설계를 직관적으로 점검할 수 있도록 하는 것이다.

3 비고

이 측정 기법 및 기준은 IEC 61508-3 의 표 A.2 와 A.4 를 참조한다.

통제 요구 사항 표현 (CORE)

1. 목표

모든 요구사항이 결정되고 표현되는 것을 보장한다.

2. 설명

이 접근법은 고객/최종 사용자와 분석가 간의 격차를 좁히기 위한 것이다. 수학적으로

정밀하지는 않지만 의사 소통을 돕는다. - CORE 는 명세보다는 요구사항 표현을 위해 설계되었다. 접근 방식은 구조적이며 표현은 다양한 수준의 상세 단계를 거친다. CORE 방법은 시스템이 사용되는 환경에 대한 지식 및 다양한 유형의 사용자에 대한 여러 관점 등을 가짐으로써 문제에 대하여 더 넓게 바라보는 것을 권장한다. CORE 에는 "총괄적 설계(grand design)"에서 출발하는 것을 인식하기 위한 가이드라인 및 활동이 포함되어 있다.

잭슨 시스템 개발 (JSD)

1. 목표

   요구 사항에서 코드에 이르는 소프트웨어 시스템 개발을 다루는 개발 방법으로 실시간 시스템을 특히 강조한다.

2. 설명

   JSD 는 개발자가 시스템 기능을 기반으로 실제 동작을 모델링하고, 필요한 작업을 결정하여 모델에 삽입하며, 결과 명세를 목표 환경에서 모델에서 실현 가능한 것으로 변환한다. 따라서 명세와 설계 및 개발의 전통적인 단계를 거치지만, 하향식이 아닌 전통적인 방식과는 다소 다른 시각을 가진다.

   더욱이, 이는 시스템을 구축하고 모델링 하는 것, 시스템에서 발생할 수 있는 일 등 현실에서의 실체(entities)를 발견하는 초기 단계에 중점을 둔다. 이러한 "현실 세계"에 대한 분석이 완료되고 모델이 생성되면, 시스템의 필수 기능이 분석되어 실제 모델에 어떻게 적용될 수 있는지를 결정한다. 결과적으로 생성된 시스템 모델은 모든 프로세스에 대한 구조적인 설명으로 보강되며 그 전체는 목표 소프트웨어 및 하드웨어 환경에서 운용 가능한 프로그램으로 변환된다.

Real-time Yourdon

1. 목표

   실시간 시스템의 사양 및 설계.

2. 설명

   이 기술의 바탕이 되는 개발 계획은 개발하게 될 시스템의 세 단계를 가정하는 것이다. 첫 번째 단계는 시스템에 요구되는 동작을 설명하는 "필수 모델"을 만드는 것이다. 두 번째 단계는 요구되는 동작을 구현할 때 필요한 구조와 메커니즘을 설명하는 구현 모델을 만드는 것이다. 세 번째 단계는 하드웨어 및 소프트웨어로 실제 시스템을 구축하는 단계이다. 세 단계는 전통적인 명세, 설계 및 개발 단계와 대체로 일치하지만 각 단계에서

개발자가 모델링 활동에 관여하는 사실에 보다 큰 중점을 둔다. 다음 두 부분은 중요한 모델이다.

- 시스템과 환경 사이의 경계에 대한 설명과 시스템이 응답해야 하는 외부 이벤트에 대한 설명을 포함하는 환경적 모델
- 이벤트에 대한 응답으로 시스템에 의해 수행되는 변환 구조에 대한 설명 및 응답을 위해 시스템이 보유해야 하는 데이터에 대한 설명을 포함하는 작동상태 모델

구현 모델 역시 개별 프로세스를 프로세서에 할당하고 프로세스를 소프트웨어 모듈로 분해하는 것을 다룰 수 있는 하위 모델로 나뉜다. 이러한 모델들을 포착하기 위해 이 기술은 데이터 흐름 다이어그램, 변환 그래프, 구조적인 영어, 상태 전이 다이어그램 및 페트리 넷과 같은 잘 알려진 기술을 결합한다. 또한, 이 방법은 종이나 기계적으로 그려진 모델로부터 제안된 시스템 설계를 시뮬레이션 하는 기술을 포함한다.

IEC61508-3 A.2.2 : 코드 감지 오류 (Error detecting codes)

1. 목표

   민감한 정보의 오류를 찾아서 수정한다.

2. 설명

   n 비트의 정보에 대해, k 비트의 코딩 된 블록이 생성되어, r 개의 에러가 검출되고 정정 될 수있게한다. 유형에는 해밍 코드 및 다항식 코드, 두가지가 있다. 안전 관련 시스템에서 오류의 미리 결정된 부분만 교정될 수 있기 때문에 오류 데이터를 수정하기보다는 오류 데이터를 삭제하는 것이 필요하다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 C.3.2 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-19

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 A.2.3a : 오류 주장 프로그래밍 (Failure assertion programming)

1. 목표

   프로그램의 실행 중 소프트웨어 설계 오류를 감지하여 시스템의 안전에 치명적인 오류를 방지하고 높은 신뢰성을 위해 작동한다.

2. 설명

   이 프로그래밍 방법은 명령문이 실행되기 전 사전 조건과 사후조건을 검사하는 것이다. 사전 조건 또는 사후 조건 중 하나라도 충족되지 않으면 오류를 보고한다.

   assert < pre-condition>;

   action 1;

   :

   :

   action x;

   assert < post-condition>;

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 C.3.3 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-24

   ☞ 자동차 가이드: -

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 A.2.3f : 역방향 복구 (Backward recovery)

1. 목표

   하나 이상의 오류가있는 경우 올바른 기능 작동을 제공합니다.

2. 설명

   오류가 감지되면 이전 상태로 시스템이 재설정 된다. 이 방법은 체크포인트의 내부 상태를 저장하는 것을 의미한다. 시스템은 저널링(추적), 보상(변경 무효화) 또는 외부 (수동) 상호 작용을 통해 발생한 변경 사항을 보완해야 한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.3.6 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-5

IEC61508-3 A.2.4a : 재시도 장애 복구 메커니즘 (Re-try fault recovery mechanisms)

1. 목표

재시도 메커니즘을 통해 탐지 된 오류 상태에서 기능적 복구를 시도한다.

2. 설명

감지 된 오류 또는 오류 조건이 발생하면 동일한 코드를 다시 실행하여 상황복구를 시도한다. 소프트웨어 재시작, 모니터링 작업 후 재시작, 스케줄링 및 재시작 작업을 수행을 시도하여 복구를 완료 할 수 있다. 재시도 기술은 일반적으로 통신 오류 또는 오류 복구에 사용되며, 조건은 체크섬과 같은 통신 프로토콜 오류 또는 응답 시간 초과에 의해 플래그가 지정 될 수 있습니다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.3.7 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-46

IEC61508-3 A.2.4b : 단계별 성능저하 (Graceful degradation)

1. 목표

덜 중요한 기능을 삭제하여 장애상태에서 중요한 시스템 기능을 유지한다.

2. 설명

시스템에서 수행 할 다양한 기능에 우선 순위를 부여한다. 이 설계는 낮은 우선 순위 기능에 자원을 할당하지 않고 보다 높은 우선 순위 기능을 수행한다. 예를 들어, 오류 및 이벤트 로깅 기능은 시스템 제어 기능보다 우선 순위가 낮을 수 있다 . 이 경우 오류 로깅이 실패하면 시스템 제어가 계속됩니다. 또한 시스템 제어가 실패하면 오류 로깅

하드웨어가 제어 기능을 대신한다. 이것은 주로 하드웨어에 적용되지만 소프트웨어를 포함한 전체 시스템에 적용 할 수 있고 이는 최상위 설계 단계부터 고려되어야 한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.3.8 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-31

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 A.2.8 : 신뢰할 수 있고 검증된 소프트웨어 요소의 사용 (Use of trusted/verified software elements(if available))

신뢰/검증된 소프트웨어 요소의 사용

1. 목표

소프트웨어 설계 및 요소가 새로운 응용 프로그램마다 광범위하게 재확인 및 재설계 되는 것을 방지한다. 형식적 또는 엄격하게 검증되지는 않았지만, 확인된 소프트웨어 요소를 설계에 활용힌다.

2. 설명

소프트웨어 요소가 오류 또는 작동 불능 상태로 부터 충분히 자유로운지를 확인한다. 유용한 기능을 제공하기 위해 기존에 개발 된 주요 하위 어셈블리 ("요소", IEC 61508-4, 3.4.5 및 3.2.8 참조)를 사용해야 하며, 일부 기능을 구현하기 위해 재사용 할 수 있어야 한다. 구조화 된 PES 는 구별되고 지정된 방식으로 상호 작용하는 소프트웨어 요소로 구성된다. 안전 관련 응용 프로그램의 경우 기존 요소가 포함 된 새 시스템에 필요한 안전 무결성이 있으며 기존 요소의 잘못된 동작으로 인해 안전이 손상되지 않는다는 것을 확인 해야한다. 기존 요소의 동작을 정확히 확인하기 위해 확인해야 할 사항은 다음과 같다.

- 요소의 포괄적인 운영 내역을 분석하여 해당 요소가 입증된 것을 확인
- 요소가 요구 사항을 충족하는지 확인하기 위해 요소의 동작에 대해 수집 된 증거를 평가

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.2.10 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

증명된 사용

1. 설명

드문 경우지만, 신뢰할 수있는 소프트웨어 요소가 필요한 안전 무결성을 달성한다는 것이 "증명된 사용"(IEC 61508-4, 3.8.18 참조) 이다. 운영체제와 같은 다양한 기능을 갖춘 복잡한 요소의 경우, 기능이 실제로 충분히 사용됨을 확인하는 것이 필수다. 예를 들어, 고장을 검출하기 위해 자가진단 루틴이 제공되는 경우, 작동 기간 내에 고장이 발생하지 않으면 고장 검출을 위한 자가진단 루틴을 사용중이라 볼 수 없다. 소프트웨어 요소는 다음 기준을 충족하는 경우 사용중인 것으로 간주된다.

- 변경되지 않은 사양
- 다른 응용 분야의 시스템
- 적어도 1 년 이상의 서비스 이력
- 안전 무결성 수준 또는 적절한 수요에 따른 작동 시간
- 비 안전 관련 실패율의 시연
- 수요 당 연간 10-2 건 95 % 신뢰의 300 건의 운영 필요
- 수요 당 연간 10-5 건 99,9 %의 신뢰의 690,000 건의 운영 필요

위 수치 계산 방법은 부록 D 를 참조한다. 통계적 접근에 대해서는 B.5.4 를 참조한다.

특정한 상황에서 안전과 관련되지 않은 고장이 다른 상황에서 안전과 관련되어 있을 수 있다. 소프트웨어 요소가 기준을 충족하는지 확인하려면 다음 사항을 문서화 해야한다.

- 버전 번호를 포함한 각 시스템 및 요소의 정확한 식별 (소프트웨어 및 하드웨어)
- 사용자의 신원 및 사용 시간
- 작동 시간
- 사용자 응용 시스템 및 응용 사례의 선택 절차
- 고장을 검출하고 등록하고 제거하는 절차

2. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.2.10 을 참조한다.

입증 증거의 시체 평가

1. 설명

기존 소프트웨어 요소 (IEC 61508-4, 3.2.8 참조)는 이미 개발되었지만 현재 프로젝트 또는 SRS 용으로 개발되지 않은 것을 뜻한다. 기존 소프트웨어를 통합하는 새로운 시스템의 안전 무결성을 평가하려면 기존 요소의 동작을 확인하는 검증이 필요하다. 이것은 공급 업체의 자체 문서 및 요소 개발 프로세스의 기록 문서거나 새로운 안전 관련 시스템의 개발자 또는 제 3 자를 통해 만들거나 보완 할 수 있다. 이것은 잠재적으로 재사용 가능한 소프트웨어 요소의 기능과 한계를 정의하는 "준수 품목 안전 수칙" 이다. 어떤 경우에도 준수 품목에 대한 안전 수칙을 지켜야하며, 특정 안전 기능의 무결성에 대한 평가를 가능하게하기에 충분하다. 이 표준은 안전 매뉴얼의 내용에 대한 특정 요구 사항을 준수하며 IEC 61508-2 Annex D 및 IEC 61508-3 Annex D 및 IEC 61508-3 7.4.2.12 및 7.4.2.13 을 참조한다. 내용에 대한 간단한 설명으로 준수 품목에 대한 안전 수칙은 다음과 같다.

- 요소의 설계가 알려지고 문서화되어 있음
- 요소는 모둔 요소 설계와 코드의 문서화된 테스트와 리뷰를 포함하는 체계적인 방법을 사용하여 검사 및 검증됨
- 안전 요구사항을 충족하며 요소의 사용되지 않거나 불필요한 기능으로 부터 새로운 시스템이 방해받지 않음
- 새로운 시스템에서 요소의 신뢰할 수있는 모든 고장 메커니즘이 구현되었는지 확인
- 새로운 시스템의 기능안전성 평가는 재사용 된 요소가 증거와 안전 수칙 준수에 따라 엄격히 적용되어야함.

IEC61508-3 A.2.11d : 자동 소프트웨어 생성 (Automatic software generation)

1. 목표

오류가 발생하기 쉬운 소프트웨어 구현 작업을 자동화한다.

2. 설명

시스템 설계는 기존의 실행 가능 코드보다 높은 추상화 수준에서 모델로 설명된다. 모델은 코드 생성기에 의해 실행 가능한 형태로 자동 변환된다. 오류가 발생하기 쉬운 수동 코딩 작업을 제거하여 소프트웨어 품질을 향상시킨다. 추상적인 수준에서 복잡한 설계를 할 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.4.6 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 ~ 2.7

IEC61508-3 A.2.13a : 최대 사이클 시간을 보장하는 주기적 동작 (Cyclic behaviour, with guaranteed maximum cycle time)

1. 목표

예측 가능한 동작으로 안전성이 중요한 실시간 시스템에 내고장성과 결합성을 구현한다.

2. 설명

TTA (Time-Triggered Architecture) 시스템에서는 모든 시스템 활동이 시작되고 전 세계적으로 동기화 된 시간을 기반으로 한다. 각 응용은 시간 트리거 버스에 시간 슬롯이 지정되며, 응용에는 정의 된 일정에 따라 교환 할 수 있는 각 응용 작업간에 교환되는 메시지가 포함되어 있다. 이벤트 중심 시스템에서 시스템 활동은 예측 할 수없는 시점에서 임의의 이벤트에 의해 시작된다. TTA 의 장점은 다음과 같다.

- 시스템 테스트 및 인증에 필요한 노력을 줄이는 결합성
- 내고장성을 투명하게 구현함으로써 안전성이 중요한 응용에 아키텍처를 권장함
- 분산 실시간 시스템의 설계를 용이하게하는 동기화 된 시간 기반의 제공

노드 간 통신은 정적 스케줄에 따라 시간 트리거 프로토콜 TTP / C 를 사용하여 메시지를 전송할 시기와 수신된 메시지가 특정 전자 모듈과 관련이 있는지 결정한다. 버스에 대한 액세스는 시간의 글로벌 개념에서 파생된 TDMA (time-division multiple access) 스키마에 의해 제어된다. TTP / C 프로토콜은 TTA 노드의 네트워크에서 네 가지 기본 서비스 (핵심 서비스)를 보장한다.

- 결정적이고 시기 적절한 메시지 전송

전송 요소의 출력 포트에서 미리 알려진 시간 범위 내에서 수신 요소의 입력 포트로 메시지를 전송한다. 시간 지연 방화벽 서비스를 통해 제공되는 시간 기반 통신 서비스는 결함허용전송 서비스를 제공한다. 이 서비스는 설계 상 제어 오류 전파를 제거하고 요소 간 결합을 최소화한다. 최소한의 대기 시간과 지터로 메시지를 적시에 전송하는 것이 실시간 애플리케이션의 제어 안정성을 달성한다.

- 내고장성 클록 동기화

통신 컨트롤러는 호스트 서브 시스템에 제공되는 폴트 허용 기준 (fulltolerant)의 글로벌 타임베이스에 동기화한다.

- 실패한 노드 (구성원 서비스)의 일관된 진단

통신 컨트롤러는 하나의 TDMA 라운드 미만의 지연을 갖는 클러스터 내의 모든 다른 SRU 의 상태에 대해 모든 SRU 에 알린다.

- 강력한 고장 분리

결함이있는 소프트웨어를 포함하는 호스트 하위 시스템은 잘못된 데이터 출력을 생성 할 수 있지만 TTP / C 클러스터의 나머지 작업을 방해할 수 없다. 시간 영역에서의 실패 침묵은 통신 제어기의 시간–촉발된 동작에 의해 보장된다. 시간 트리거 프로토콜은 FlexRay 및 TT 이더넷 (시간 트리거 이더넷) 이다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.3.11 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 B.9 - Modular approach

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	소프트웨어 모듈 크기 제한		HR	HR	HR	HR
2	소프트웨어 복잡성 제어		R	R	HR	HR
3	정보 숨기기 / 캡슐화		R	HR	HR	HR
4	매개 변수 개수 제한 / 고정 된 서브 프로그램 매개 변수 수		R	R	R	R
5	서브 루틴과 함수에서 한 개의 엔트리 포인트 / 한 개의 출구 포인트		HR	HR	HR	HR
6	완전히 정의 된 인터페이스		HR	HR	HR	HR

안전 무결성 수준에 따라 적절한 측정 기법 및 기준을 선택한다. 어떤 단일 기술만으로는 충분하지 않을 수 있다. 모든 적절한 기법을 고려한다.

IEC61508-3 B.9.1 : 소프트웨어 모듈 크기 제한 (Software module size limit)

1. 목표

   시스템의 복잡성을 제한하기 위해 소프트웨어 시스템을 이해 가능한 작은 부분으로 분해한다.

2. 설명

   모듈 방식 또는 모듈화는 소프트웨어 프로젝트의 설계, 코딩 및 유지 관리 단계에 대한 몇 가지 규칙을 포함한다. 규칙은 설계 중에 사용 된 설계 방법에 따라 다르지만 대부분의 메소드에는 다음과 같은 규칙이 있습니다.

   - 소프트웨어 모듈 (또는 이와 동등한 서브 프로그램)은 정의 된 단일 작업 또는 기능을 갖춘어야 한다.
   - 소프트웨어 모듈 간의 연결은 엄격하게 제한 및 정의되어야하며, 하나의 소프트웨어 모듈에서 일관성을 가져야 한다.
   - 여러 레벨의 소프트웨어 모듈을 제공하여 서브 프로그램 콜렉션을 만들어야 한다.
   - 서브 프로그램 크기는 지정된 값, 일반적으로 2-4 개의 화면 크기로 제한되어야 한다.
   - 서브 프로그램은 단일 항목과 단일 종료만을 가져야 한다.
   - 소프트웨어 모듈은 인터페이스를 통해 다른 소프트웨어 모듈과 통신해야한다.
   - 전역 변수 또는 공통 변수가 잘 구조화되어야하고, 액세스가 제어되어야하며, 각 인스턴스에서 그 사용이 정당화되어야 한다.
   - 모든 소프트웨어 모듈 인터페이스는 완전히 문서화되어야 한다.
   - 모든 소프트웨어 모듈의 인터페이스는 기능에 필요한 매개 변수만 포함해야 한다.

   프로그래밍 언어가 기본 매개 변수를 허용하거나 객체 지향 접근 방식이 사용되는 가능성에 따라 복잡해진다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.9 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.4 SW 아키텍처 설계 > 1.4.1 주요 요구사항 및 설명 > 1.4.1.2 소프트웨어 아키텍처 설계 원칙

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 B.9.2 : 소프트웨어 복잡성 제어 (Software complexity control)

1. 목표

   소프트웨어 자체의 속성 및 개발 또는 테스트 기록에서 프로그램의 특성을 예측한다.

2. 설명

   소프트웨어의 일부 구조적 특성을 평가하고 이를 신뢰성 또는 복잡성과 같은 원하는 특성과 연결한다. 소프트웨어 도구는 대부분의 측정 값을 평가해야 하며 적용 할 수 있는 측정 항목 중 일부는 다음과 같다.

   - 그래프 이론적인 복잡성: 수명주기의 초기에 적용되어 트레이드 오프를 평가할 수 있으며 프로그램 제어 그래프의 복잡성에 기반한다.
   - 특정 소프트웨어 모듈을 활성화하는 방법의 수 (액세스 가능성): 소프트웨어 모듈에 액세스 할 수 있으면 디버그 할 가능성이 높다.
   - Halstead 유형 메트릭 과학: 연산자 및 피연산자 수를 계산하여 프로그램 길이를 계산하고 미래 개발 자원을 추정 할 때 비교를 위한 기준선을 형성하는 복잡성과 규모의 척도를 제공한다.
   - 소프트웨어 모듈 당 출입구 수 - 출입구 수를 최소화하는 것은 구조화 된 설계 및 프로그래밍 기술의 특징이다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.13 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-38, C-6.10

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 B.9.3 : 정보 숨기기 / 캡슐화 (Information hiding/encapsulation)

1. 목표

   데이터 또는 절차에 대한 의도하지 않은 액세스를 방지하여 프로그램 구조를 지원한다.

2. 설명

   모든 소프트웨어 요소에 전역적으로 액세스 할 수있는 데이터는 실수로 또는 잘못 수정 될 수 있다. 데이터 구조를 변경하려면 코드 및 광범위한 수정에 대한 세부적인 조사가 필요하다. 정보 은닉은 이러한 어려움을 최소화하기 위한 일반적인 접근 방법입니다. 주요 데이터 구조는 "숨김" 이며, 정의 된 액세스 절차 집합을 통해서만 조작 할 수 있다. 소프트웨어의 기능적 동작에 영향을 미치지 않고 내부 구조를 수정하거나 절차를 추가 할 수 있다. 소프트웨어의 논리적 동작에 영향을 미치지 않고 액세스 절차 및 내부 데이터 구조를 재작성 (예 : 다른 조회 방법을 사용하거나 하드 디스크에 이름 저장) 할 수 있다. 이와 관련하여 추상 데이터 유형의 개념을 사용해야합니다

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.8 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-38, C-6.10

   ☞ 자동차 가이드: -

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 A.3 - Software design and development - support tools and programming language

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	적절한 프로그래밍 언어	IEC61508-3 A.3.1	HR	HR	HR	HR
2	강력한 형식의 프로그래밍 언어	IEC61508-3 A.3.2	HR	HR	HR	HR
3	언어 하위 집합		---	---	HR	HR
4a	공인 된 도구 및 공인 번역사		R	HR	HR	HR

4b	도구 및 번역자 : 사용으로 인한 자신감 증가	IEC61508-3 A.3.3	HR	HR	HR	HR
• 안전 무결성 수준에 따라 적절한 측정 기법 및 기준을 선택한다. • 대체가능하거나 동등한 측정 기법 및 기준은 번호 뒤에 오는 문자로 표시하며 이 중 하나를 선택한다. • 대체 측정 기법 및 기준은 특정 응용에 맞는 특성에 따라 적절하게 선택한다.						

IEC61508-3 A.3.1 : 적절한 프로그래밍 언어 (Suitable programming language)

1. 목표

국제 표준의 요구사항을 가능한 많이 지원하며 특히 방어적 프로그래밍, 명확한 자료형, 구조적 프로그래밍 및 표명(assertion)을 지원한다. 선택된 프로그래밍 언어는 최소한의 노력으로 쉽게 검증 가능한 코드를 생성할 수 있으며 프로그램 개발, 검증 및 유지보수를 용이하게 한다.

2. 설명

언어는 완전하고 명백하게 정의되어야 한다. 언어는 프로세서/플랫폼 지향적이 아닌 사용자 지향적이거나 문제지향적이어야 한다. 특수한 용도의 언어보다 범용 언어의 선택이 보다 일반적이다.

다음과 같은 특징을 언어에서 제공하면 좋다.

- 블록 구조
- 번역 시간 검사
- 런타임 유형 및 배열 바운드 확인

또한 언어는 다음을 장려하는 것이 좋다.

- 작고 다루기 쉬운 소프트웨어 모듈의 사용
- 특정 소프트웨어 모듈의 데이터 액세스 제한
- 가변 부분 범위의 정의
- 다른 유형의 오류 제한 구조

시스템의 안전한 운용이 실시간 제약 조건에 달려있다면 언어는 예외/인터럽트 처리를 제공해야 한다. 언어는 적절한 번역기, 기존 소프트웨어 모듈의 적절한 라이브러리, 버전 제어 및 개발을 위한 디버거와 도구에 의해 지원되는 것이 바람직하다. 이 표준을 개발할 당시, 객체 지향 언어가 다른 언어보다 선호되는 지는 명확히 알 수 없다.

검증을 어렵게 만들기 때문에 피해야 할 특징들은 다음과 같다:

- 서브루틴 호출을 제외한 무조건적 점프
- 재귀
- 포인터, 힙 또는 모든 유형의 동적 변수 또는 객체
- 소스 코드 수준에서의 인터럽트 처리
- 루프, 블록 또는 서브 프로그램의 다중 입구 또는 종료
- 암시적인 변수의 초기화 또는 선언
- 변형 기록 및 등가물
- 절차적 매개 변수들

특히 어셈블리 언어와 같은 저수준 언어는 프로세서/플랫폼 지향적인 특성으로 인해 문제가 발생한다. 바람직한 언어의 특징은 그 언어를 사용하여 설계함으로써 실행이 예측 가능한 프로그램을 만들 수 있는 것이다. 적절하게 정의된 프로그래밍 언어가 주어지면, 프로그램의 실행을 예측할 수 있도록 보장하는 부분집합이 있다. 실행의 예측 가능성이 높아지더라도 정적인 제한 요소가 많으면 이 부분집합을 정적으로 결정할 수는 없다(일반적으로). 이는 배열 인덱스가 범위 내에 있고 수치적인 오버플로우가 발생할 수 없다는 것 등에 대한 입증을 요구한다.

3. 비고

   이 기법 / 측정은 IEC 61508-3 의 표 A.3 에서 참조된다

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-54, C-6.13

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 A.3.2: 강력한 형식의 프로그래밍 언어 (Strongly typed programming language)

1. 목표

   컴파일러가 고수준의 검사를 실행하는 언어를 사용하여 결함 확률을 줄이는 것이다.

2. 설명

   강력한 형식의 프로그래밍 언어는 컴파일 될 때 프로시저 호출 및 외부 데이터 액세스와 같이 변수 유형을 사용하는 방식에 대한 많은 확인이 이루어진다. 미리 정의된 규칙을 준수하지 않으면 컴파일이 실패하고 오류 메시지가 생성된다.

이러한 언어는 일반적으로 기본적인 언어 데이터 유형(예: 정수, 실수)에서 사용자 정의 데이터 유형을 정의 할 수 있게 한다. 이는 기본 유형과 정확하게 같은 방식으로 사용할 수 있다. 정확한 유형이 사용되는 것을 보장하도록 엄격한 검사가 요구된다. 이러한 검사는 개별적인 단위로 컴파일 된 경우에도 전체 프로그램에 적용된다. 또한 검사는 개별적으로 컴파일 된 소프트웨어 모듈에서 참조된 경우에도 프로시저 인수의 수와 유형이 일치하는지 확인해야 한다.
강력한 형식 언어는 우수한 소프트웨어 공학 사례를 보여준다. 쉽게 분석 가능한 제어 구조(예: ..then else, do .. while 등)를 가져 잘 짜여진 프로그램으로 이어지는 것이 대표적인 사례이다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-3 의 표 A.3 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-49, C-6.13

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 A.3.4a : 공인된 도구 및 공인 번역사 (Certified tools and certified translators)

1. 목표

소프트웨어 개발의 여러 단계에서 개발자를 돕는 도구가 필요하고 출력물의 정확성과 관련하여 확신을 가질 수 있도록 도구를 인증해야 한다.

2. 설명

도구의 인증은 일반적으로 국가 별 또는 국제 표준과는 독립적으로 설정된 기준에 따라 독립적인 기관에 의해 수행된다. 모든 개발 단계 (사양, 디자인, 코딩, 테스트 및 유효성 검사)에서 사용되는 도구와 구성 관리에 사용되는 도구는 인증 대상이 된다. 현재까지 컴파일러만이 정기적으로 인증 절차를 밟고 있습니다. 이것들은 국가 인증 기관에 의해 규정되며, Ada 와 Pascal 과 같은 국제 표준에 대한 컴파일러를 사용한다. 공인 된 도구 및 컴파일러는 일반적으로 해당 언어 또는 프로세스 표준에 대해서만 인증 받으며 안전과 관련된 인증은 받지 않는다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.4.3 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 A.3.4b : 도구 및 번역기: 사용으로 인한 자신감 증가 (Tools and translators: increased confidence from use)

1. 목표

소프트웨어 패키지의 개발, 검증 및 유지 보수 중에 발생할 수 있는 번역기 오류로 인한 어려움을 피한다.

2. 설명

다수의 프로젝트에서는 성능이 부적절하다는 증거가 없는한 변역기가 사용되어 왔다. 운용된 이력이 없거나 심각한 결함이 있는 것으로 알려진 번역기는 올바른 성능에 대한 기타 보증이 없는 한 피해야 한다(예: C.4.4.1 참조). 번역기가 약간의 결함이라도 나타냈다면 관련 언어 구조를 기록하고 안전관련 프로젝트 중에는 해당 번역기의 사용을 피해야 한다. 이 기법의 또 다른 특징은 일반적으로 사용되는 언어의 기능만 제공하도록 제한하는 것이다. 불완전한 번역기는 소프트웨어 개발 중 심각한 장애를 일으킬 수 있으며, 안전과 관련한 소프트웨어의 개발을 불가능하게 한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.4.4 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 A.4 - Software design and development - detailed design

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	구조적 기법	IEC61508-3	HR	HR	HR	HR

		A.4.1 :				
1b	준 정형 기법	IEC61508-3 B.7	R	HR	HR	HR
1c	정식 디자인 및 개선 방법		---	R	R	HR
2	컴퓨터 지원 설계 도구		R	R	HR	HR
3	방어 프로그래밍		---	R	HR	HR
4	모듈 방식	IEC61508-3 B.9	HR	HR	HR	HR
5	디자인 및 코딩 표준	IEC61508-3 B.1	R	HR	HR	HR
6	구조화 된 프로그래밍	IEC61508-3 A.4.2	HR	HR	HR	HR
7	신뢰할 수 있고 검증 된 소프트웨어 요소의 사용 (있는 경우)		R	HR	HR	HR
8	소프트웨어 안전 요구 사항 사양과 소프트웨어 디자인 간의 포워드 추적성		R	R	HR	HR

- 안전 관련 시스템을 위한 객체 지향 소프트웨어 개발의 적합성에 대해 여전히 논의되고 있다.
- 안전 무결성 수준에 따라 적절한 측정 기법 및 기준을 선택한다.
- 대체가능하거나 동등한 측정 기법 및 기준은 번호 뒤에 오는 문자로 표시하며 이 중 하나를 선택한다.
- 대체 측정 기법 및 기준은 특정 응용에 맞는 특성에 따라 적절하게 선택한다.

IEC61508-3 A.4.2 : 컴퓨터 지원 설계 도구 (Computer-aided design tools)

1. 목표

보다 체계적으로 설계 단계를 수행하며 입증된 적절한 자동 설계 요소를 포함한다.

2. 설명

시스템의 복잡성으로 인해 필요하다고 판단되면 하드웨어 및 소프트웨어 설계시 컴퓨터 지원 설계 도구 (CAD)를 사용해야한다. 이러한 도구의 정확성은 특정 테스트, 사용의 광범위한 기록 또는 설계중인 특정 안전 관련 시스템에 대한 출력물의 독립적인 검증을

통해 입증되어야 한다. 지원 도구는 통합 정도에 따라 선택 되어야 하고 도구는 협업을 통해 하나의 도구의 출력이 후속 도구의 자동 입력을 위한 적합한 내용와 포맷을 갖도록 통합되면 재작업 중 인적오류가 발생할 가능성을 최소화한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 B.3.5 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 A.4.5 : 디자인 및 코딩 표준 (Design and coding standards)

1. 목표

검증이 용이하도록 팀 중심의 객관적인 접근 방식과 표준 설계 방법을 제공한다.

2. 설명

규칙은 참여자간 프로젝트 초기에 정하며 이는 설계 및 개발 방법 (예 : JSP, 페 트리 넷 등) 및 관련 코딩 표준 (C.2.6.2 참조)으로 구성된다. 이러한 규칙은 개발, 검증, 평가 및 유지 관리가 용이하게 하며 사용 가능한 도구, 특히 분석기 및 리버스 엔지니어링 도구를 고려해야한다

3. 비고

이 측정 기법 및 기준은 IEC 61508-3 의 B.1 및 IEC 61508-7 의 C.2.6 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-15, C-6.11

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계, 2.6 단계 5 소프트우웨어 구현 및 단위 테스트

IEC61508-3 B.1 - Design and coding standards

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	오류 가능성을 줄이기 위한 코딩 표준의 사용	IEC61508-3 B.1.1	HR	HR	HR	HR
2	동적 객체 없음		R	HR	HR	HR
3a	동적 변수 없음		---	R	HR	HR
3b	동적 변수 설치의 온라인 검사		---	R	HR	HR
4	제한된 인터럽트 사용		R	R	HR	HR
5	포인터의 사용 제한		---	R	HR	HR
6	제한된 재귀 사용		---	R	HR	HR
7	고수준 언어의 프로그램에서 구조화되지 않은 제어 흐름이 없음		R	HR	HR	HR
8	자동 유형 변환 없음		R	HR	HR	HR

- 측정 2, 3a 및 5 동적 객체 (예 : 실행 스택 또는 힙)의 사용은 사용 가능한 메모리와 실행 시간 모두에 대한 요구 사항이 존재할 수 있다. 컴파일러가 아래 사항을 만족할 때 2, 3a 및 5 번 항목은 적용하지 않아도 된다.
  a) 모든 동적 변수와 객체에 대해 충분한 메모리가 런타임 이전에 할당되거나 메모리 할당 오류의 발생
  b) 응답 시간이 요구 사항을 만족함
- 안전 무결성 수준에 따라 적절한 측정 기법 및 기준을 선택한다. 대체가능하거나 동등한 측정 기법 및 기준은 번호 뒤에 오는 문자로 표시하며 이 중 하나를 선택하여 만족해야 한다. 대체 측정 기법 및 기준은 부속서 C 에 명시된 특성 중 특정 응용에 맞는 특성에 따라 적절하게 선택한다.

IEC61508-3 B.1.1: 오류 가능성을 줄이기 위한 코딩 표준의 사용

1. 목표

안전 관련 코드에서 오류의 가능성을 줄이고 검증을 용이하게 한다.

2. 설명

프로그래밍 언어에 대하여 안전 관련 코딩 규칙이 어떻게 IEC 61508-3 규범 요구 사항을 준수하며 "바람직한 특성(부록 F 참조)"을 달성하는지 설명한다. 또한 사용할 수 있는 툴에 대하여 설명한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-3 의 표 B.1 을 참조한다.

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-15, C-6.11

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 B.1.2 : 동적 객체 없음 (No dynamic objects)

1. 목표

원치 않거나 감지되지 않은 오버레이 메모리 및 안전관련 런타임 병목현상을 없애기 위함이다.

2. 설명

동적 변수 및 동적 객체는 메모리가 할당되고 런타임에 절대 주소가 결정되는 변수 및 객체입니다. 할당 되는 메모리 값과 주소는 할당 시점의 시스템 상태에 따라 달라 지므로 컴파일러나 다른 오프라인 도구로는 확인할 수 없다. 동적 변수 및 객체의 수와 새로운 동적 변수 또는 객체를 할당하기위한 기존의 여유 메모리 공간은 할당 시점의 시스템 상태에 따라 달라 지므로 변수 또는 변수를 할당하거나 사용할 때 오류가 발생할 수 있다. 시스템에서 할당 한 위치의 여유 메모리 양이 부족할 경우 다른 변수의 메모리 내용을 우연히 덮어 쓸 수 있다. 동적 변수 또는 객체를 사용하지 않으면 이러한 오류가 발생하지 않는다. 일부 정적 분석에 의해 동적 운용을 정확하게 예측할 수없는 동적 객체의 사용에 대한 제한이 필요하므로 예측 가능한 프로그램 실행을 보장 할 수 없다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.6.3 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-15, C-6.11

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

IEC61508-3 B.1.4 : 제한된 인터럽트 사용 (Limited use of interrupts)

1. 목표

   소프트웨어를 검증 가능하고 테스트가 가능하도록 유지한다.

2. 설명

   인터럽트는 시스템을 간소화하는 데 사용될 수 있으며 사용은 제한되어야 한다. 인터럽트의 소프트웨어 처리는 실행 된 기능의 중요한 부분 (예 : 시간 변경, 중요한 데이터 변경)동안 금지되어야합니다. 인터럽트가 사용되면 인터럽트가 불가능한 파트는 지정된 최대 계산 시간을 가져야하므로 인터럽트가 금지되는 최대 시간을 계산할 수 있다. 인터럽트 사용과 마스킹은 철저히 문서화되어야 한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.6.5 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.2 단계 1. 개발 계획 수립

사. 사용 양식

- [SD-D-01] SW 설계 명세서
- [SD-D-02] SW 단위시험 계획서
- [SD-D-03] SW 통합시험 계획서
- [SD-D-04] SW 코딩 매뉴얼
- [SD-V-01] SW 설계 안전성분석 보고서

아. 적용 기법

- [SD-T-01] SW 설계 기법
- [SD-T-02] SW 구조 위험분석 기법
- [SD-T-03] SW 모듈 위험분석 기법

4. SW 구현(SC) 단계

Software Construction Phase

<table>
<tr><td>목적</td><td>SW 구현의 목적은 SW 설계를 적절히 반영하는 실행 가능한 SW 단위(모듈)을 생성한다.</td></tr>
<tr><td>시작 기준</td><td>SW 모듈설계가 완료됨
SW 모듈 테스트계획이 수립됨</td></tr>
<tr><td>활동</td><td>• SW 모듈설계 명세서
• 시스템 안전성 요구사항 명세서
• SW 개발계획서

<table>
<tr><th>개발 활동</th><th>확인 검증 활동</th><th>안전 활동</th></tr>
<tr><td>4.1.1 SW 모듈 구현</td><td>4.2.1 SW 구현 평가
4.2.2 확인검증 작성</td><td>4.3.1 SW 구현 안전평가 수행
4.3.2 SW 안전기록 작성</td></tr>
</table>

[SC-V-01] SW 코드 안전성분석 보고서</td></tr>
<tr><td>확인</td><td>검증담당자에 의해 SW 코드가 검증된다.
안전담당자가 SW 코드의 안전을 분석한다</td></tr>
<tr><td>종료기준</td><td>SW 모듈이 구축됨
SW 구현확인 검증 보고서 및 SW 안전 기록이 작성됨</td></tr>
</table>

가. 목 적

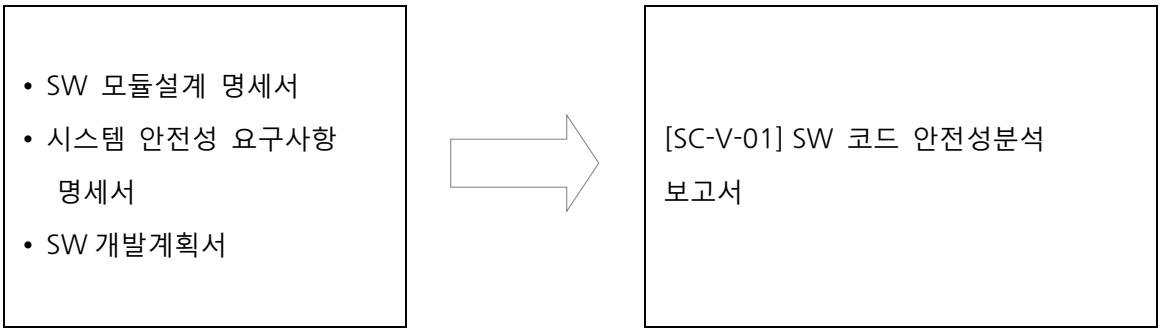
SW 모듈 설계에 따라 실행 가능한 SW 모듈을 생성하는 것이다.

나. 책임과 권한

책임 \ 역할		구현담당자	품질담당자	안전담당자
개발활동	4.1.1 SW 모듈 구현	R		
검증활동	4.2.1 SW 구현물 평가	S	R	
	4.2.2 확인 검증 보고서 작성	S	R	
안전활동	4.3.1 SW 모듈안전 평가 수행	S	S	R
	4.3.2 SW 안전 기록 작성	S	S	R

(Responsible:담당, Approve:승인, Support:지원)

다. 입력물 및 출력물



라. 개발 활동

4.1.1 SW 모듈 구현

각 SW 모듈을 실행 가능한 형태로 개발한다. 모듈 설계 시 정해진 표준에 따라 각 모듈을 구현한다.

마. 확인 검증 활동

4.2.1 SW 구현물 평가

원시코드, 추적 가능성, 인터페이스 분석을 통해 SW 구현물이 설계내용을 충분히 반영하고 있음을 평가한다.

원시 코드 평가

원시코드 구성요소(원시코드 및 문서화)를 정확성, 일관성. 완전성, 정밀성, 테스트 가능성을 가지고 평가한다. 작업 기준은 다음과 같다.

- 정확성: 원시코드 구성요소가 SW 설계를 만족하는지 검증, 확인한다. 원시코드 구성요소가 표준, 참고문헌, 규칙, 정책, 물리적 법칙, 사업적 규칙을 따르는지 검증한다. 전문적 기술계, 프로토타입의 걸과, 공학적 원리. 그 밖의 원리와 관련 있는 논리 적, 자료의 흐름을 이용하여 원시코드 구성요소의 연속적 상태와 상태의 변화를 확인한다. 데이터와 제어의 흐름이 기능적, 성능적 요구사항을 만족하는지 확인한다. 데이터의 사용법과 형식을 확인한다. 코딩 방법과 표준의 타당성을 평가한다.
- 일관성: 모든 용어와 코드의 개념이 일관성 있게 문서화 되었는지 검증한다. 원시코드 구성요소 사이에 내부의 일관성이 있는지 검증한다. SW 설계와 요구사항 사 이에 외면적 일관성이 있는지 검증한다.
- 완전성: 다음 요소가 시스템의 전제조건과 제약 안에서 원시코드에서 구현되었는지 검증한다.
  - 기능(예, 알고리즘, 상태/방식 정의, 입력/출력 확인, 예외 처리, 보고서, 기록)
  - 프로세스 정의 및 스케줄링
  - HW, SW, 사용자 인터페이스 설명
  - 성능 측정 기준(예, 타이밍, 사이징, 속도, 용량, 정확성, 안전성, 보안성);
  - 중요한 형상 자료
  - 시스템, 장치, SW 제어(예, 초기화, 트랜잭션과 상태 감시, 자체 테스트)
  - 원시 코드의 문서화가 특정 형상 관리 절차를 만족여부

- 정밀성: 논리적, 계산적, 인터페이스의 정밀성(예, 반올림, 버림)을 시스템 환경 하에서 확인한다. 모형화된 물리적 현상이 시스템의 정밀성 요구사항과 물리적 법칙을 따르는지 확인한다.
- 가독성: 문서가 사용자에게 읽기 쉽고, 이해하기 쉬우며, 분명한지 검증한다. 문서가 두문자 약어, 기억술, 약어, 용어, 기호를 정의하고 있음을 검증한다.
- 테스트 가능성: 원시코드 구성요소를 확인하기 위한 객관적인 승인 기준이 존재하는지 검증한다. 각 원시코드 구성요소가 객관적인 승인 기준으로 테스트할 수 있는지 검증한다.

추적 가능성 분석

설계 명세서를 기반으로 원시코드 구성요소를 유추한다. 정확성, 일관성, 완전성을 분석한다. 작업 기준은 다음과 같다:

- 정확성: 원시코드 구성요소와 설계 요소 사이의 관계를 확인한다.
- 일관성: 원시코드 구성요소와 설계 요소 사이의 관계가 일관성이 있는지 검증한다.
- 완전성: 모든 원시코드 구성요소가 설계 요소로부터 유추될 수 있는지 검증한다. 모든 설계 요소를 가지고 원시코드 구성요소를 유추할 수 있는지 검증한다.

HW, 사용자, 운영자, SW, 다른 시스템과 연관 있는 SW 원시코드의 인터페이스를 정확성, 일관성, 완전성. 정밀성, 테스트가능성을 기준으로 검증 및 확인한다.

- 정확성: 시스템 요구사항 하에서 외부, 내부의 SW 인터페이스 코드를 확인한다.
- 일관성: 인터페이스 코드가 원시코드 구성요소와 외부의 인터페이스(예, HW, 사용자, 운영자, 다른 SW) 사이에서 일관성이 있는지 검증한다.
- 완전성: 각 인터페이스를 설명하고 있는지, 자료 형식과 성능 측정 기준(예, 타이밍, 대역폭, 정밀도, 안전성. 보안성)을 포함하고 있는지 검증한다.
- 정확성: 각 인터페이스가 요구하는 정확한 정보를 제공하는지 검증한다.
- 테스트가능성: 인터페이스 코드를 확인하기 위한 객관적인 승인 기준이 존재하는지 검증한다.

4.2.2 확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술해야 한다. 확인 검증 보고서는 다음을 포함한다.

- SW 요구분석서, SW 설계 명세서 또는 SW 모듈 설계명세서에 부합하지 않는 항목
- SW 품질 보증 계획과 부합하지 않은 항목
- 문제에 잘 맞지 않는 모듈. 데이터, 구조 그리고 알고리즘
- 검출된 오류 또는 부족한 부분
- 검증된 항목의 식별 및 형상

바. 안전 활동

SW 설계에서의 안전과 관련된 부분이 요구사항을 정확히 구현하고 있으며 새로운 위험을 초래하지 않음을 검증한다.

4.3.1 SW 구현 안전 평가 수행

원시 코드를 사용해서 이전의 주요 작업 보고서의 중요도 분석 결과를 검토하고 갱신한 다. 구현 방법과 인터페이스 기술에 따라 주어진 SW 의 요소(예, 요구사항, 모듈, 함수, 하위시스템, 다른 SW) 에 지정된 SW 무결성 수준이 상향 또는 하향 조절될 수 있다. 수정된 SW 무결성 수준으로 인하여 어떠한 불일치나 예상하지 않는 SW 무결성이 발생하지 않음을 검증한다.

원시코드와 연관된 자료가 중요한 요구사항을 정확하게 구현했는지, 어떠한 장애도 발생 시키지 않았는지 검증한다. 장애 분석을 갱신한다.

이전의 작업 보고서를 이용하여 위험 분석을 검토하고 갱신한다. 위험을 제거, 감소, 약화 하기 위한 권장사항을 제시한다.

4.3.3 SW 안전 기록 작성

SW 안전 기록에는 다음을 포함시킨다.

- 안전 분석 결과
- 의심되거나 확인된 안전 문제점
- 안전 테스트 결과

사. 사용 양식

- [SC-V-01] SW 코드 안전성분석 보고서

아. 적용 기법

- [SC-T-01] SW 구현 안전 평가 기법

5. SW 통합(SI) 단계

Software Integration Phase

<table>
<tr><td>목적</td><td>SW 통합의 목적은 SW 모듈을 결합하여 일관되고 통합된 SW를 생성하고,
HW와 통합된 SW가 최초 명세한 요구사항을 만족함을 확인하는 것이다.</td></tr>
<tr><td>시작 기준</td><td>SW 모듈이 모두 구축되고, SW 통합 계획이 수립됨
SW 요구사항이 확정됨
SW 요구사항테스트명세서가 작성됨</td></tr>
<tr><td>활동</td><td>
• SW 모듈
• SW 통합계획서
• 시스템 안전성 요구사항 명세서
• SW 개발 계획서

<table>
<tr><th>개발 활동</th><th>확인 검증 활동</th><th>안전 활동</th></tr>
<tr><td>5.1.1 SW 모듈테스팅
5.1.2 SW 통합테스트 개발
5.1.3 SW 통합테스트 수행
5.1.4 SW/HW 테스트 수행</td><td>5.2.1 추적가능성 분석
5.2.2 모듈테스트 결과 확인
5.2.3 통합테스트 결과 검증
5.2.4 SW/HW 테스트 검증
5.2.5 확인검증 작성</td><td>5.3.1 SW 통합 안전평가 수행
5.3.2 SW 안전기록 작성</td></tr>
</table>

[SI-D-01] SW 단위시험 보고서
[SI-D-02] SW 통합시험 보고서
[SI-V-01] SW 시험 안전성분석보고서
</td></tr>
</table>

확인	검증담당자에 의해 SW 모듈이 검증된다. 검증담당자에 의해 HW 와 통합된 SW 가 검증된다. 안전담당자가 SW 모듈의 안전을 분석한다. 안전담당자가 HW 와 통합된 SW 의 안전을 분석한다.
종료 기준	SW 통합이 완성됨 SW 통합 테스트결과가 작성됨 SW 통합확인 검증 보고서 및 SW 안전 기록이 작성됨

가. 목 적

SW 모듈을 결합하여 SW 설계와 일관되고 통합된 SW 항목(item)을 생성하고, 기능적 및 비기능적 SW 요구사항을 만족하는 것을 보여준다. HW 와 통합된 SW 가 정의된 요구사항이 만족되었음을 확인한다.

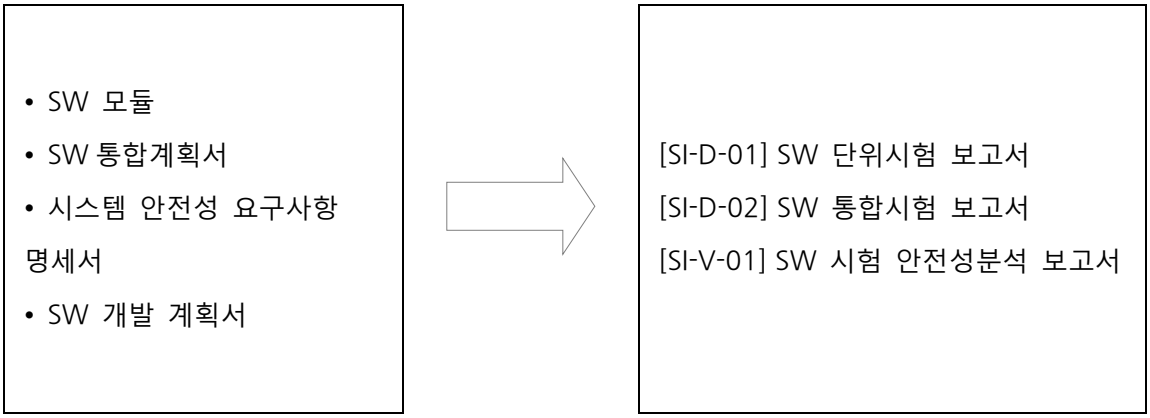
나. 책임과 권한

책임 \ 역할		통합담당자	품질담당자	안전담당자
개발활동	5.1.1 SW 모듈 테스팅	R		
	5.1.2 SW 통합 테스트 개발	R		
	5.1.3 SW 통합 테스트 수행	R		
	5.1.4 SW/HW 테스트	R		
검증활동	5.2.1 추적 가능성 분석	S	R	
	5.2.2 모듈테스트 결과 확인	S	R	
	5.2.3 통합 테스트 결과 검증	S	R	
	5.2.4 SW/HW 테스트 검증	S	R	
	5.2.5 확인 검증 보고서 작성	S	R	

안전활동	5.3.1 SW 통합 안전 평가 수행	S	S	R
	5.3.2 SW 안전 기록 작성	S	S	R

(Responsible:담당, Approve;:승인, Support:지원)

다. 입력물 및 출력물



라. 개발 활동

5.1.1 SW 모듈 테스팅

이 테스트는 각 모듈이 의도된 기능을 수행하는지 보여주기 위한 것이다. 이를 위해 이전 단계에서 작성한 SW 모듈 테스트 케이스를 활용하여 테스트를 수행하고. 그 결과로서 SW 모듈 테스트 보고서를 생성한다.

SW 항목을 구성하기 위한 통합 전략에 따라 SW 단위를 통합한다. SW 모듈들의 통합은 모듈 인터페이스 및 결합된 SW 가 통합 및 테스트 전에 적절히 확인될 수 있도록 각각의 사전에 테스트된 SW 모듈을 순차적으로 결합하는 방식으로 수행한다.

5.1.2 SW 통합 테스트 개발

SW 요구사항을 고려하는 SW 단위 통합 전략을 개발한다. SW 아키텍처에 기초한 SW 항목을 식별하고, 이들을 통합 및 테스트하기 위한 순서 또는 규칙을 정의한다.

각 통합된 SW 항목에 대해 수행되어야 하는 통합 테스트케이스를 개발한다. 테스트케이스에는 요구사항이 점검됨을 나타내기 위하여 인터페이스의 검증, 입력 데이터 및 검증 기준을 포함한다.

5.1.3 SW 통합 테스트 수행

통합 테스트를 수행한다. 테스트 결과가 테스트 계획 문서에서 테스트 추적 가능성의 테스트 기준을 따르는지 확인한다. 통합 테스트 계획에 의해서 요구되는 결과를 통합 테스트 보고서에 기록한다. SW 가 테스트 승인 기준을 만족하는지 확인하기 위하여 통합 테스트결과를 이용한다. 실제 테스트 결과와 예상했던 결과의 차이를 기록한다.

SW 통합 테스트보고는 다음 내용고려하여 작성한다.

- SW 통합 테스트 보고서는 테스트 결과와 SW 통합 테스트계획의 목적 및 기준이 충족되었는지를 기술하여야 한다. 통합이 실패할 경우 그 사유를 기록한다.
- SW 통합 테스트 보고서는 심사할 수 있는 형식이어야 한다.
- 테스트 사례와 그 걸과는 반드시 기록하며, 가능하면 후속 분석을 위해 기계에서 읽을 수 있는 형식으로 기록한다.
- 테스트는 반복 가능해야 하고, 가능하면 자동화된 수단으로 수행한다.
- 검증된 항목의 식별 및 형상을 포함하여야 한다.

5.1.4 SW/HW 통합 테스트

검증 기준에 따라 통합된 SW 제품을 테스트하고 그 결과 SW HW 통합 테스트보고서를 기록한다. 필요시 사용자 문서를 갱신한다.

마. 확인 검증 활동

5.2.1 추적 가능성 분석

구조 설계 명세서와 모듈 간의 정확성, 일관성. 완전성을 분석한다. 그 기준은 다음과 같다.

- 정확성: 구조 설계 명세서와 모듈 간의 관계를 확인한다. 확인 검증 테스트 계획, 설계, 사례, 절차 사이에 정확한 관계가 존재하는지 검증한다.
- 일관성: 구조 설계 명세서와 모듈 간의 관계가 일관성이 있는지 검증한다.
- 완전성: 모든 구조 설계 명세서와 모듈로부터 유추될 수 있는지 검증한다. 모든 모듈로부터 구조 설계를 유추할 수 있는지 검증한다. 모든 확인 검증 테스트 절차로 확인 검증 테스트 계획을 유추할 수 있는지 검증한다.

5.2.2 모듈 테스트 결과 확인

SW 가 테스트 승인 기준을 만족하는지 확인하기 위하며 개발자의 모듈 테스트 결과를 이용한다.

5.2.3 통합 테스트 결과 검증

통합 테스트 사례 및 테스트 절차가 통합 테스트를 위해 통합 테스트계획서의 기준을 만족 하였는지 확인한다.

5.2.4 SW/HW 테스트 검증

SW 가 테스트 승인 기준을 만족하는지 검증하기 위해서 개발자의 시스템 테스트 결과를 이용한다.

5.2.5 확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술해야 한다. 확인 검증 보고서는 다음을 포함해야 한다.

- SW 요구분석서, SW 설계 명세서 또는 SW 모듈 설계명세서에 부합하지 않는 항목
- SW 품질 보증 계획과 부합하지 않은 항목
- 문제에 잘 맞지 않는 모듈, 데이터, 구조 그리고 알고리즘
- 검출된 오류 또는 부족한 부분
- 검증된 항목의 식별 및 형상

바. 안전 활동

5.3.1 SW 통합 안전 평가 수행

통합테스트결과가 명시된 환경 하에서 안전 요구사항이 정확히 구현되었고 SW 가 안전하게 기능함을 입증한다. 테스트는 다음을 포함해서 수행한다.

- 인터페이스 테스팅
- 컴퓨터 SW 형상항목 테스팅
- 시스템 수준 테스팅
- 스트레스 테스팅
- 리그레션 테스팅

5.3.2 SW 안전 기록 작성

SW 안전 기록은 다음을 포함한다.

- 안전 분석 결과
- 의심되거나 확인된 안전 문제점

- 안전 테스트 결과

IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)

IEC61508-3 A.5 - Software design and development - software module testing and integration

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	확률론적 테스트		---	R	R	R
2	동적 분석 및 테스트	IEC61508-3 B.2	R	HR	HR	HR
3	데이터 기록 및 분석	IEC61508-3 A.5.1	HR	HR	HR	HR
4	기능 및 블랙 박스 테스트	IEC61508-3 B.3	HR	HR	HR	HR
5	성능 테스트	IEC61508-3 B.6	R	R	HR	HR
6	모델 기반 테스트		R	R	HR	HR
7	인터페이스 테스트		R	R	HR	HR
8	테스트 관리 및 자동화 도구		R	HR	HR	HR
9	소프트웨어 설계 명세와 모듈 및 통합 테스트 명세 간의 전향적 추적 성		R	R	HR	HR
10	정형 검증		---	---	R	R

- 소프트웨어 모듈 및 통합 시험은 검증 활동이다 (IEC61508-3 B.9 참조).
- 기법 9 형식 검증은 필요한 모듈 및 통합 테스트의 양과 범위를 줄일 수 있다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.

IEC61508-3 A.5.2 : 동적 분석 및 테스트(Dynamic analysis and testing)

1. 목표

   완성된 상태에서 프로토 타입의 동적 동작을 검사하여 명세 오류를 감지한다.

2. 설명

   안전 관련 시스템의 동적 분석은 안전 관련 시스템의 프로토 타입을 의도된 운영 환경의 입력 데이터에 적용함하여 수행한다. 안전 관련 시스템의 모든 오류는 수정 되어야하고 새로운 운영 버전에 재분석되어야 한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 B.6.5 및 IEC 61508-3 의 B.2 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-4, B-18, B-20, B-21, B-39, B-50, C-6.6, C-6.7, C-6.19

   ☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.5 백투백비교테스트(모델과 코드)/백투백테스트, 2.6.2 동등분할 , 2.6.3 경계값 분석, 2.6.4 에러 추정, 2.6.5 구문 커버리지, 2.6.6 분기/결정 커버리지, 2.6.7 변형 조건/결정 커버리지

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.5.3 : 데이터 기록 및 분석(Data recording and analysis)

1. 목표

   소프트웨어 프로젝트의 모든 데이터, 의사 결정 및 근거를 문서화하여 확인, 검증, 평가 및 유지관리 할 수 있도록 한다.

2. 설명

   프로젝트 도중 자세한 문서가 유지되며 다음 사항이 포함되어야 한다.

   - 각 소프트웨어 모듈에서 수행 된 테스트
   - 결정과 그 이론적 근거
   - 문제와 해결책

   프로젝트가 진행되는 동안 및 결론에서 이 문서를 분석하여 다양한 정보를 얻을 수 있다. 데이터 기록은 컴퓨터 시스템의 유지 관리에있어 매우 중요하다. 개발 프로젝트 동안

만들어진 특정 결정에 대한 이론적 근거가 유지 보수 엔지니어들에게 전달되지 않기 때문이다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.5.2 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-12, C-6.23

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.5.4 : 기능 및 블랙 박스 테스트 (Functional and black box testing)

기능 테스트

1. 목표

사양 및 설계 단계에서의 실패를 나타내고 구현 및 소프트웨어 및 하드웨어의 통합 중 실패를 방지한다.

2. 설명

기능 테스트 중에 시스템의 지정된 특성이 달성되었는지를 확인하는 검토가 수행된다. 시스템은 정상적으로 예상되는 동작을 적절하게 특성화하는 입력 데이터를 제공받는다. 출력은 관측되고 응답은 명세에 의해 주어진 것과 비교된다. 불완전한 명세의 표시 및 표시로부터의 편차가 문서화 되어있다. 다중 채널 아키텍처 용으로 설계된 전자 부품의 기능 테스트에는 사전 검증 된 파트너 부품으로 테스트 한 제조 부품이 포함됩니다. 이 외에도 제조 된 구성 요소는 동일한 배치의 다른 파트너 구성 요소와 함께 테스트하여 다른 방법으로는 가려진 공통 모드 오류를 표시하는 것이 좋다. 시스템의 작업 용량은 충분해야하며, C.5.20 을 참조한다

블랙 박스 테스트

1. 목표

실제 기능 조건에서 동적 동작을 확인하고 기능 명세 충족하지 못하고 유틸리티 및 견고성을 평가하는 데 실패했다는 사실이 표시한다.

2. 설명

시스템 또는 프로그램의 기능은 지정된 환경에서 실행되며 지정된 테스트 데이터는 설정된 기준에 따라 명세에서 체계적으로 추출된다. 시스템의 동작을 공개하고 명세와의 비교를 허용한다. 시스템의 내부 구조에 대한 지식이 테스트에 사용되지 않는다. 목적은 기능 단위가 명세에서 요구하는 모든 기능을 올바르게 수행하는지 여부를 확인하는 것이다. 입력 데이터 공간은 명세에 부합하게 특정 입력 값 범위 (등가 클래스)로 세분됩니다. 그런 다음 테스트 케이스는 다음 사항들로 구성된다.

- 허용 범위의 데이터
- 허용되지 않는 범위의 데이터
- 범위 한도의 데이터
- 극단 값
- 상위 클래스의 조합

다른 테스트 활동 (모듈 테스트, 통합 테스트 및 시스템 테스트)에서 테스트 케이스를 선택하기 위해서는 다른 기준이 효과적 일 수 있다. 예를 들어, "극한적인 작동 조건" 기준은 유효성 확인 프레임워크 내에서 시스템 테스트에 의존한다.

3. 비고

이 기법 / 측정은 IEC 61508-2 의 표 B.3, B.5 및 B.6 과 표 A.5, A.6, A.7, C.5, C.6 및 C 에서 참조된다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-4, B-6, B-18, B-42, B-43, C-6.6, C-6.7

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.1 요구사항기반 테스트, 2.6.1 요구사항 분석, 2.6.2 동등분할, 2.6.3 경계값 분석

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.5.6 : 모델 기반 테스트(Model based testing)

1. 목표

시스템 모델로부터 효율적인 자동 테스트 케이스 생성을 용이하게하고 고도로 반복 가능한 테스트 스위트를 생성한다

2. 설명

모델 기반 테스팅 (MBT)은 테스트 케이스 생성 (TCG) 및 테스트 결과 평가와 같은 일반적인 테스트 작업이 시스템 (애플리케이션) 테스트 (SUT) 모델을 기반으로하는 블랙 박스 접근 방식이다. 일반적으로 시스템 데이터와 사용자 행동은 Finite state machine, Markov 프로세스, 의사 결정 테이블 등을 사용하여 모델링된다. 모델 기반 테스트는 소스 코드 수준 테스트 범위 측정과 결합 할 수 있으며 기능 모델은 기존 소스 코드를 기반으로 할 수 있다. 모델 기반 테스트는 시스템 요구 사항 및 지정된 기능 모델을 사용하여 효율적인 테스트 케이스/절차를 자동 생성한다. 테스트가 매우 비싸기 때문에 자동 테스트 케이스 생성 도구에 대한 수요가 커지고 있습니다. 따라서 모델 기반 테스팅은 현재 매우 활발한 연구 분야이며 수많은 TCG (Test Case Generation) 도구를 사용할 수 있다. 툴은 전형적으로 모델의 행동 부분에서 테스트 스위트를 추출하여 특정 커버리지 요구 사항을 충족시킨다. 이 모델은 SUT (System Under Test)의 원하는 동작을 추상화 한 부분 표현이다. 이 모델에서 테스트 모델이 파생되어 추상적인 테스트 스위트를 구축한다. 테스트 케이스는이 추상적인 테스트 스위트에서 파생되고 시스템에 대해 실행되며 테스트는 시스템 모델에 대해서도 실행될 수 있다. TCG 가있는 MBT 는 공식 방법의 사용에 기반을두고 있으며 권장 사항은 안전 무결성 수준 (SIL)과 유사하다. 높은 SIL 의 경우 HR (권장)이며 낮은 SIL 의 경우 필요하지 않다. 구체적인 활동은 다음과 같다.

- 시스템 요구 사항에서 모델 구축
- 예상 입력 생성
- 예상 출력 생성
- 테스트 실행
- 예상 출력과 실제 출력 비교
- 추가 작업 결정 (모델 수정, 더 많은 테스트 생성, 소프트웨어의 안정성 / 품질 평가)

테스트는 사용자 / 시스템 동작 모델을 표현하는 다양한 방법 및 기법으로 도출 할 수 있다.

- 의사 결정 테이블
- 유한 상태 기계
- 문법을 사용
- Markov Chain 모델
- 상태 차트
- 정리 증명
- 제약 논리 프로그램

- 모델 검사
- 상징적 실행
- 이벤트 흐름 모델
- 반응 시스템 테스트 : 병렬 계층 적 유한 자동 기계

모델 기반 테스팅은 최근 안전 관련 핵심 영역을 대상으로 한다. 명세와 설계에서 모호성을 조기에 표시 할 수 있게 해주며, 비 반복적인 효율적인 테스트를 자동으로 생성하고 회귀 테스트 스위트를 평가하며 소프트웨어 안정성과 품질을 평가하고 테스트 스위트 업데이트를 용이하게 한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.5.27 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.5 백투백비교테스트(모델과 코드)/ 백투백테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.5.7 : 인터페이스 테스트(Interface testing)

1. 목표

서브 프로그램의 인터페이스에서 오류를 감지합니다.

2. 설명

테스트의 세부 수준 또는 완전성이 여러 단계 가능합니다. 다음을 위한 테스트는 가장 중요한 단계다.

- 모든 극한값의 인터페이스 변수;
- 개별적으로 극한값의 모든 인터페이스 변수와 함께 정상값의 다른 인터페이스 변수
- 각 인터페이스 변수의 도메인의 모든 값은 다른 인터페이스 변수와 함께 정상 값
- 조합 된 모든 변수의 모든 값 (이것은 작은 인터페이스에 대해서만 가능합니다);
- 각 서브 루틴의 각 호출과 관련된 지정된 테스트 조건.

인터페이스가 잘못된 매개 변수 값을 감지를 포함하지 않는 경우, 이 테스트는 특히 중요하며 또한 기존 서브 프로그램의 새로운 구성이 생성 된 후에도 중요하다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.5.3 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-34, C-6.20

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.2 인터페이스 테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..7. 소프트웨어 통합 및 통합테스트

IEC61508-3 A.5.8 : 테스트 관리 및 자동화 도구(Test management and automation tools)

1. 목표

소프트웨어 및 시스템 테스트에 대한 체계적인 접근을 권장한다.

2. 설명

적절한 지원 도구를 사용하면 시스템 개발에서 노동 집약적이고 오류가 발생하기 쉬운 작업을 기계화하고 체계적인 테스트 관리 방법을 제공 할 수 있다. 가능하다면 일반 및 회귀 테스트가 권장된다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.4.7 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 B.2 - Dynamic analysis and testing

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	경계 값 분석을 통한 테스트 케이스 실행		R	HR	HR	HR
2	오류 추측에서 테스트 사례 실행		R	R	R	R

3	오류 시드에서 테스트 사례 실행		---	R	R	R
4	모델 기반 테스트 케이스 생성으로부터 테스트 케이스 실행		R	R	HR	HR
5	성능 모델링		R	R	R	HR
6	동등한 클래스와 입력 파티션 테스트		R	R	R	HR
7a	구조 테스트 커버리지 (진입 점) 100 %	IEC61508-3 B.2.1	HR	HR	HR	HR
7b	구조 테스트 커버리지 (명세서) 100 %		R	HR	HR	HR
7c	구조 테스트 커버리지 (지점) 100 %		R	R	HR	HR
7d	구조 테스트 커버리지 (조건, MC / DC) 100 %		R	R	R	HR

- 시험 경우에 대한 분석은 서브 시스템 레벨에서 이루어지며, 명세 및 코드는 명세 또는 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.
- 100 % 적용 범위를 달성 할 수 없는 경우 (예 : 방어 명령문 적용) 적절한 설명이 제공되어야 한다.

IEC61508-3 B.2.1 : 경계 값 분석을 통한 테스트 케이스 실행 (Test case execution from boundary value analysis)

1. 목표

   매개 변수 한계 또는 경계에서 발생하는 소프트웨어 오류를 감지한다.

2. 설명

   프로그램의 입력 영역은 등가 관계 (C.5.7 참조)에 따라 여러 입력 클래스로 나눈다. 테스트는 클래스의 경계와 극한을 다루어야 한다. 테스트는 명세 입력 영역의 경계가 프로그램의 경계와 일치하는지 확인한다. 직접 및 간접 번역에서 0 값을 사용하면 종종 오류가 발생하기 쉽다.

   - 제로 제수
   - 공백 ASCII 문자

- 빈 스택 또는리스트 요소
- 전체 매트릭스
- 제로 테이블 엔트리

일반적으로 입력 경계는 출력 범위의 경계에 직접적으로 대응한다. 테스트 케이스는 출력을 제한된 값으로 강제 설정해야하며, 또한 출력이 스펙 경계 값을 초과하게하는 테스트 케이스를 지정할 수 있는지 고려해야 한다. 출력이 일련의 데이터 (예 : 인쇄 된 표) 인 경우 첫 번째 및 마지막 요소와 하나 및 두 개의 요소가 포함 된 목록에 주의해야한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.4 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-4, C-6.6

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.3 경계값 분석

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.2.2 : 오류 추측에서 테스트 사례 실행 (Test case execution from error guessing)

1. 목표

일반적인 프로그래밍 실수를 제거한다.

2. 설명

테스트 경험과 직관력과 함께 테스트중인 시스템에 대한 지식과 호기심이 결합되어 설계된 테스트 케이스 세트에 분류되지 않은 테스트 케이스가 추가 될 수 있다. 특수 값 또는 값 조합은 오류가 발생할 수 있다. 일부 테스트 케이스는 검사 체크리스트에서 파생 될 수 있다. 또한 시스템이 충분히 견고한 지 여부도 고려할 수 있다.

예 : 너무 빨리 또는 너무 자주 버튼을 전면 패널에 밀 수 있습니까?

두 개의 버튼을 동시에 누르면 어떻게됩니까?

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.5 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-20

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.4 에러 추정

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.2.5 : 성능 모델링 (Performance modelling)

1. 목표

시스템의 작업 용량이 지정된 요구 사항을 충족시키는지 확인한다.

2. 설명

요구 사항 명세에는 특정 기능에 대한 처리량 및 응답 요구 사항이 포함되며, 전체 시스템 리소스 사용 제약 조건을 포함한다. 제안된 시스템 설계는 다음에 의해 명시된 요구 사항과 비교할 수 있다.

- 시스템 프로세스 및 그들의 상호 작용 모델을 생성
- 프로세서 시간, 통신 대역폭, 저장 장치 등과 같은 각 프로세스 별 자원 사용 결정
- 평균 및 최악의 상황에서 시스템에 대한 요구 분포를 결정
- 개별 시스템 기능에 대한 평균 및 최악의 처리량 및 응답 시간 계산

간단한 시스템의 경우 분석 솔루션으로 충분할 수 있지만보다 복잡한 시스템의 경우 정확한 결과를 얻기 위해 시뮬레이션이 더 적합 할 수 있다.
상세한 모델링을하기 전에 모든 프로세스의 자원 요구 사항을 취합하기보다 간단한 "자원 예산"점검을 사용할 수 있다. 요구 사항이 설계된 시스템 용량을 초과하면 설계는 실행 불가능하다. 설계가 이 검사를 통과하더라도 성능 기량 모델링은 리소스 부족 때문에 과도한 지연 및 응답 시간이 발생한다. 이러한 상황을 피하기 위해 엔지니어는 전체 자원 중 일부 (예 : 50 %)를 사용하도록 시스템을 설계하여 자원 부족의 가능성을 줄인다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.20 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-39

☞ 자동차 가이드: -

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.2.6 : 동등한 클래스와 입력 파티션 테스트 (Equivalence classes and input partition testing)

1. 목표

최소한의 테스트 데이터를 사용하여 소프트웨어를 적절히 테스트한다. 테스트 데이터는 소프트웨어를 실행하는 데 필요한 입력 도메인의 파티션을 선택하여 얻는다.

2. 설명

이 테스트 전략은 입력 도메인의 파티션을 결정하는 입력의 등가 관계를 기반으로 한다. 테스트 케이스는 이전에 지정된 모든 파티션을 다루기 위해 선택된다. 적어도 하나의 테스트 케이스가 각 동등한 클래스에서 선택된다. 입력 파티셔닝에는 두 가지 기본 가능성이 있습니다.

- 명세에서 파생 된 동등한 클래스

- 명세의 해석은 입력 지향 일 수 있다. 예를 들어, 선택된 값이 같은 방식으로 처리되거나 출력 방향이 지정 될 수 있거나 값의 집합이 동일한 기능 결과를 가질 수 있다.

- 프로그램의 내부 구조에서 파생 된 동등한 클래스 - 동등한 클래스 결과는 프로그램의 정적 분석에서 결정된다. 예를 들어, 동일한 경로로 이어지는 값의 집합이 실행된다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.7 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-18, C-6.7

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.2 동등분할

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.2.7a : 구조 테스트 커버리지 (진입점) 100% ** (Structural test coverage (entry points) 100 % **)

1. 목표
   프로그램 구조의 특정 하위 집합을 사용하는 테스트를 적용합니다.

2. 설명
   프로그램 분석을 기반으로 입력 데이터 집합이 선택되어 프로그램 코드의 큰 비율 (종종 사전 지정된 목표)이 실행된다. 코드 적용 범위의 측정은 요구되는 엄격함의 수준에 따라 다음과 같이 달라질 수 있다. 모든 경우에, 선택된 커버리지 메트릭의 100 %가 목표가되어야한다. 100 % 적용 범위를 달성 할 수없는 경우, 달성 할 수없는 이유는 테스트 보고서 (예 : 하드웨어 문제가 발생할 경우에만 입력 할 수있는 방어 코드)에 기록해야 한다. 다음 목록의 처음 네 가지 기술은 IEC 61508-3 의 표 B.3 의 권장 사항에서 특별히 언급되며 테스트 도구로 지원되며 나머지 기술도 고려 될 수 있다.

   - 진입 점 (콜 그래프) 적용 범위 : 모든 서브 프로그램 (서브 루틴 또는 함수)이 적어도 한 번 호출 되었는지 확인한다. 객체 지향 언어에는 다이나믹 디스패치에 의해 호출 될 수있는 다형성 유형의 다른 변형 (서브 프로그램 오버라이딩)에 적용되는 같은 이름의 여러 서브 프로그램이 있을 수 있으므로 모든 오버라이딩 서브 프로그램을 테스트한다.

   - 명령문 : 코드의 모든 명령문이 적어도 한 번 실행 되었는지 확인한다.

   - 브랜치 : 모든 지점의 양측을 확인한다. 일부 유형의 방어 코드에서는 실용적이지 않을 수 있다.

   - 복합 조건 : 복합 조건부 분기 (즉, AND / OR 로 연결된)의 모든 조건이 실행된다.

   - LCSAJ : 선형 코드 시퀀스 및 점프는 조건문을 포함하여 점프에 의해 종료되는 코드 명령문의 선형 시퀀스다. 잠재적인 하위 경로는 이전 코드의 입력 데이터에 대한 제약으로 인해 실행 불가능하다.

   - 데이터 흐름 : 실행 경로는 데이터 사용량에 따라 선택된다.

- 기본 경로 : 처음부터 끝까지 유한 경로의 최소 집합 중 하나이며 모든 아크가 포함된다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.8 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-50, C-6.19

☞ 자동차 가이드: -

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 B.3 - Functional and black-box testing

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	원인 결과 다이어그램에서 테스트 사례 실행		---	---	R	R
2	모델 기반 테스트 케이스 생성으로부터 테스트 케이스 실행		R	R	HR	HR
3	프로토 타이핑 / 애니메이션		---	---	R	R
4	경계 값 분석을 포함한 동등한 클래스와 입력 파티션 테스트		R	HR	HR	HR
5	공정 시뮬레이션		R	R	R	R
• 시험 케이스에 대한 분석은 소프트웨어 시스템 수준에서 이루어지며 명세된 내용을 기반으로 한다. • 시뮬레이션의 완성도는 안전무결성 수준, 복잡성, 운용에 따른다. • 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 B.3.2 모델 기반 테스트 케이스 생성으로부터 테스트 케이스 실행

분야별 가이드 참조 위치

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.5 백투백비교테스트(모델과 코드)/ 백투백테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 B.3.4 경계 값 분석을 포함한 동등한 클래스와 입력 파티션 테스트

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-4, C-6.6

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.2 동등분할, 2.6.3 경계값 분석

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.3.5 공정 시뮬레이션

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-42

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.6 - Performance testing

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	눈사태 / 스트레스 테스트		R	R	HR	HR
2	응답 타이밍 및 메모리 제약	IEC61508-3 B.6.1	HR	HR	HR	HR
3	성능 요구 사항	IEC61508-3 B.6.2	HR	HR	HR	HR
• 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 B.6.1 : 눈사태 / 스트레스 테스트 (Avalanche/stress testing)

1. 목표

예외적으로 높은 워크로드로 테스트 객체에 부담을 주면서 테스트 객체가 정상적인 작업 부하를 쉽게 견딜 수 있는지 확인한다.

2. 설명

눈사태 / 스트레스 테스트에 적용 할 수있는 다양한 테스트 조건이 있으며 그 중 일부는 다음과 같다.

- 폴링 모드에서 작업하는 경우, 테스트 객체는 시간당 더 많은 입력 변경 사항을 가져온다.
- 요구에 따라 작업하는 경우, 시험 개체에 대한 시간당 요구수는 정상 조건을 초과 증가한다.
- 데이터베이스 크기가 중요한 역할을 하는 경우 정상 조건을 초과 증가한다.
- 영향력있는 장치는 각각 최대 속도 또는 최저 속도로 조정된다.
- 극단적인 경우, 가능한 모든 영향 요인이 경계 조건에 동시에 적용된다.

이 테스트 조건에서 테스트 객체의 시간 동작을 평가할 수 있고, 로드 변화의 영향을 관찰 할 수 있다. 내부 버퍼 또는 동적 변수, 스택 등의 올바른 차원을 검사 할 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.21 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-3

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.12 스트레스 테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.6.2 : 응답 타이밍 및 메모리 제약 (Response timings and memory constraints)

1. 목표

시스템이 일시적 및 메모리 요구 사항을 충족하는지 확인한다.

2. 설명

시스템 및 소프트웨어의 요구 사항 명세에는 특정 기능에 대한 메모리 및 응답 요구 사항이 포함되어 있으며, 전체 시스템 리소스 사용에 대한 제약 조건과 관련되어 있다.

분석은 평균 및 최악의 조건 하에서 할당 요구를 결정한다. 이 분석을 위해서 각 시스템 기능의 자원 사용 및 경과 시간을 예측해야 한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.22 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-45, C-6.4

   ☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.8 성능 테스트

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 B.6.3 : 성능 요구 사항 (Performance requirements)

1. 목표

   소프트웨어 시스템의 입증 가능한 성능 요구 사항을 수립한다.

2. 설명

   시스템 및 소프트웨어 요구 사항 사양 모두에 대한 분석을 수행하여 모든 일반 및 특정, 명시적 및 암시적 성능 요구 사항을 만들고 각 성능 요구 사항을 차례로 검사한다.

   - 획득 된 성공 기준
   - 성공 기준에 대한 조치가 획득 될 수 있는지 여부
   - 측정의 잠재적 정확성
   - 측정이 추정 될 수있는 프로젝트 단계
   - 측정이 가능한 프로젝트 단계

   성능 요구 사항, 성공 기준 및 잠재적 측정 목록을 얻기 위해 각 성능 요구 사항의 실행 가능성을 분석한다. 목표는 다음과 같다.

   - 각 성능 요구 사항은 적어도 하나의 측정과 관련됨
   - 가능한 한 개발 초기에 사용할 수있는 정확하고 효율적인 측정 선택
   - 필수 및 선택적 성능 요구 사항 및 성공 기준결정
   - 가능한 경우 하나 이상의 성능 요구 사항에 대해 단일 측정을 사용

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.19 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-40, C-6.5

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.8 성능 테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 A.6 - Programmable electronics integration (hardware and software)

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	기능 및 블랙 박스 테스트	IEC61508-3 B.3	HR	HR	HR	HR
2	성능 테스트	IEC61508-3 B.6	R	R	HR	HR
3	하드웨어 / 소프트웨어 통합 및 하드웨어 / 소프트웨어 통합 테스트 사양에 대한 시스템과 소프트웨어 설계 요구 사항 간의 전향 적 추적 성		R	R	HR	HR
• 프로그래머블 전자 장치 통합은 검증 활동이다 (IEC61508-3 A.9 참조). • 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 A.6.1~3

IEC61508-3 B.5.1, B.5.2: 기능 및 블랙박스 테스팅(Functional and black box testing) 참조

IEC61508-3 A.6.1 기능 및 블랙 박스 테스트

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-4, B-6, B-18, B-42, B-43, C-6.6, C-6.7

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.1 요구사항기반 테스트, 2.6.1 요구사항 분석, 2.6.2 동등분할, 2.6.3 경계값 분석

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.6.2 성능 테스트

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-3, B-40, B-45, C-6.4, C-6.5

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.8 성능 테스트

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 A.6.3 하드웨어 / 소프트웨어 통합 및 하드웨어 / 소프트웨어 통합 테스트 사양에 대한 시스템과 소프트웨어 설계 요구 사항 간의 전향 적 추적 성

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-58, C-6.18

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.1~ 2 .8

IEC61508-3 A.7 – Software aspects of system safety validation

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	확률론적 테스트		---	R	R	HR
2	공정 시뮬레이션		R	R	HR	HR
3	모델링	IEC61508-3 B.5	R	R	HR	HR
4	기능 및 블랙 박스 테스트	IEC61508-3 B.3	HR	HR	HR	HR
5	소프트웨어 안전 요구 사항 사양과 소프트웨어 안전 유효성 검사 계획 간의 전향 추적 성		R	R	HR	HR

6	소프트웨어 안전성 검증 계획과 소프트웨어 안전성 요구 사항 사양 간의 역추적성		R	R	HR	HR
• 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 A.7.2 : 공정 시뮬레이션(Process simulation)

1. 목표

   소프트웨어 시스템의 기능을 외부 세계와의 인터페이스와 테스트한다. 어떤식으로든 실제 세계를 수정하지 않아도된다.

2. 설명

   테스트 목적으로 만 통제중인 장비의 동작을 모방 한 시스템을 생성한다. 시뮬레이션은 소프트웨어 또는 소프트웨어와 하드웨어의 조합 일 수 있고 다음 사항을 따른다.

   - EUC 가 실제로 설치 될 때 존재 할 수 있는 입력과 동등한 입력을 제공한다.
   - 통제 대상 설비를 충실히 대표하는 방식으로 시험중인 소프트웨어의 출력에 응답한다.
   - 시험중인 시스템이 대응해야하는 모든 변화를 제공 할 수 있도록 운영자 입력을 제공한다.

   소프트웨어가 테스트 될 때, 시뮬레이션은 입력 및 출력과 함께 타겟 하드웨어의 시뮬레이션 일 수 있습니다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 C.5.18 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-42

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 A.4 : 기능 및 블랙 박스 테스트 (IEC61508-3 B.3)

IEC61508-3 B.5.1, B.5.2: 기능 및 블랙박스 테스팅(Functional and black box testing) 참조

IEC61508-3 A.7.3 모델링

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-11, B-13, B-27, B-28, B-39, B-43, B-51, B-55, B-65, B-66, B-67, C-6.1, C-6.2

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..6 ~ 2.8

IEC61508-3 A.7.4 기능 및 블랙 박스 테스트

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-4, B-6, B-18, B-42, B-43, C-6.6, C-6.7

☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드 가이드 > 2.5 테스트 메소드 > 2.5.1 요구사항기반 테스트, 2.6.1 요구사항 분석, 2.6.2 동등분할, 2.6.3 경계값 분석

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 A.7.5 소프트웨어 안전 요구 사항 사양과 소프트웨어 안전 유효성 검사 계획 간의 전향 추적 성

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-58, C-6.18

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.1~ 2 .8

IEC61508-3 A.7.6 소프트웨어 안전성 검증 계획과 소프트웨어 안전성 요구 사항 사양 간의 역추적성

분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-58, C-6.18

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.1~ 2 .8

IEC61508-3 A.8 - Modification

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	영향 분석	IEC61508-3 A.8.1	HR	HR	HR	HR
2	변경된 소프트웨어 모듈 재확인	IEC61508-3 A.8.2	HR	HR	HR	HR
3	영향을 받는 소프트웨어 모듈 재확인		R	HR	HR	HR
4a	전체 시스템 재 검증	IEC61508-3 A.7	---	R	HR	HR
4b	회귀 검증		R	HR	HR	HR
5	소프트웨어 구성 관리	IEC61508-3 A.8.3	HR	HR	HR	HR
6	데이터 기록 및 분석	IEC61508-3 A.8.4	HR	HR	HR	HR
7	소프트웨어 안전 요구 사항 사양과 소프트웨어 수정 계획 (재 검증 및 재 검증 포함) 간의 전향 적 추적 성		R	R	HR	HR
8	소프트웨어 수정 계획 (재 검증 및 재 검증 포함)과 소프트웨어 안전 요구 사항 명세 사이의 역 추적 성		R	R	HR	HR

- 그룹 4. 영향 분석은 회귀 검증의 필수적인 부분이다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다. 대체 기술 또는 이에 상응하는 기술/조치는 번호 뒤에 오는 문자로 표시된다. 대체 또는 동등한 기술/조치 중 하나만 만족하면 된다.

IEC61508-3 A.8.1: 충격 분석(Impact analysis)

1. 목표

   소프트웨어 시스템의 변경 또는 개선이 소프트웨어 시스템의 다른 소프트웨어 모듈, 시스템에 미칠 영향을 결정한다.

2. 설명

소프트웨어에 대한 수정 또는 개선을 수행하기 전에 영향을 파악하고 영향을 받는 소프트웨어 시스템 및 소프트웨어 모듈을 알아내기 위한 분석을 수행한다.
분석이 완료된 후 소프트웨어 시스템의 재검증에 관한 결정이 필요하다. 이는 영향을 받는 소프트웨어 모듈의 수, 영향을 받는 소프트웨어 모듈의 중요성 및 변경의 성격에 따라 다르다. 가능한 결정은 다음과 같다.
- 변경된 소프트웨어 모듈만 재검증한다.
- 영향을 받는 모든 소프트웨어 모듈들을 재검증한다.
- 전체 시스템을 재검증한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-3 의 표 A.8 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-32, C-6.22

☞ 의료 가이드: Part 2> 제 4 장 소프트웨어 유지 보수 프로세스 > 4.2 문제 및 변경 사항 분석

IEC61508-3 A.8.3: 소프트웨어 형상 관리(Software configuration management)

1. 목표

소프트웨어 형상 관리는 해당 산출물이 변경 될 때 개발 산출물 그룹의 일관성을 보장한다. 형상 관리는 일반적으로 하드웨어 및 소프트웨어 개발에 모두 적용되며 소스코드, 문서, 인터페이스 등 각종 결과물을 대상으로 한다.

2. 설명

소프트웨어 형상 관리는 개발 과정에서 사용되는 기법이다 (IEC 61508-3, 6.2.3 참조).
본질적으로 형상의 모든 버전과 형상의 서로 다른 버전 간의 모든 관계를 문서화한다.
결과 문서는 개발자가 하나의 형상(특히 그 요소 중 하나)에 대한 변경이 다른 산출물에 미치는 영향을 결정할 수 있게 한다. 변경 요청이 있을 경우 변경 여부와 변경 활동을 통제하고 현재 상태와 변경 항목들이 제대로 반영되었는지 여부를 확인한다. 형상 관리가 잘 이루어 질 경우, 시스템 또는 하위 시스템은 일관된 버전 구성에서 안정적으로 재구축 될 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-3 의 표 A.8 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 2> 제 4 장 소프트웨어 유지 보수 프로세스 > 4.2 문제 및 변경 사항 분석

IEC61508-3 A.8.4b : 회귀 검증 (Regression validation)

1. 목표

회귀 테스트에서 유효한 결론을 도출한다.

2. 설명

크거나 복잡한 시스템의 완전한 회귀 테스트에는 많은 노력과 자원이 필요하다. 시스템 개발의 해당 지점에서 직접 관심사인 시스템 측면만을 포괄하도록 회귀 테스트를 제한하는 것이 바람직하다. 회귀 테스트에서 부분 테스트의 범위를 명확하게 이해하고 테스트 된 시스템 상태에 대한 유효한 결론을 이끌어내야한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.5.25 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.8 단계 7 소프트웨어 시스템 테스트

IEC61508-3 A.8.5 : 소프트웨어 형상 관리(Software configuration management)

1. 목표

소프트웨어 형상 관리는 해당 산출물이 변경 될 때 개발 산출물 그룹의 일관성을 보장하는 것을 목표로 하며, 하드웨어 및 소프트웨어 개발에 모두 적용됩니다.

2. 설명

소프트웨어 형상 관리는 개발 과정에서 사용된다(IEC 61508-3, 6.2.3 참조). 모든 중요한 제품의 모든 버전의 생산과 다양한 제품의 서로 다른 버전 간의 모든 관계를 문서화해야 한다. 결과 문서는 개발자가 하나의 제품에 대한 변경을 다른 제품에 미치는 영향을

결정할 수 있게 한다. 특히 시스템 또는 하위 시스템은 요소 버전의 일관된 세트에서 안정적으로 재 구축 될 수 있다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 C.5.24 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-48

☞ 의료 가이드: Part 2> 제 6 장 소프트웨어 형상관리 프로세스

IEC61508-3 A.9 - Software verification

Technique/Measure		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	정식 증명		---	R	R	HR
2	명세 및 디자인의 생기		R	R	R	R
3	정적 분석	IEC61508-3 B.8	R	HR	HR	HR
4	동적 분석 및 테스트	IEC61508-3 B.2	R	HR	HR	HR
5	소프트웨어 설계 사양과 소프트웨어 검증 (데이터 검증 포함) 계획 간의 포워드 추적 성		R	R	HR	HR
6	소프트웨어 검증 (데이터 검증 포함) 계획과 소프트웨어 설계 명세 간의 역 추적 성		R	R	HR	HR
7	오프라인 수치 분석		R	R	HR	HR
소프트웨어 모듈 테스트 및 통합		See IEC61508-3 A.5				
프로그래밍 가능한 전자 통합 테스트		See IEC61508-3 A.6				
소프트웨어 시스템 테스트 (검증)		See IEC61508-3 A.7				

- 이 표는 모든 검증 활동이 정리되어 있으나 표 A.5 와 표 A.6 의 동적 검증 요소에 대한 추가적인 요구 사항을 제시하지 않는다. 또한 이 표는 소프트웨어 검증 (표 B.7 을 참조한다) 이외에 검증 테스트를 요구 하지 않는다. 이 표는 안전 요구 사항 명세에 대한 적합성의 사례(demonstration)이다(end-end verification).
- 검증은 IEC 61508-1, IEC 61508-2 및 IEC 61508-3 과 모두 관련된다. 안전 관련 시스템의 초기 검증은 이전의 시스템 레벨 명세와 상반된다.
- 소프트웨어 안전 생명주기에서 초기 단계에는 정적 검증 방법이 사용되고, 코드가 생성되면 동적 테스트가 진행된다. 정적 검증 방법은 소프트웨어 검사, 워크 스루 (walk-through), 정적 분석, 정식 증명 등이 있다. 동적 테스트는 기능 테스트, 화이트 박스 테스트, 통계 테스트가 있다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.

IEC61508-3 A.9.2 : 명세 및 디자인의 생기 (Animation of specification and design)

1. 목표

   명세를 체계적으로 검사하여 소프트웨어를 검증한다.

2. 설명

   실행 가능한 코드보다 더 추상적인 소프트웨어의 표현(명세 또는 상위 레벨 설계)은 최종 실행 가능한 소프트웨어의 동작을 결정하기 위해 검사된다. 시험은 실행 가능한 소프트웨어의 동작 및 출력을 시뮬레이션하기 위해 (상위 수준 표현의 성격 및 추상 수준에 따라 제공되는 가능성에 따라) 자동으로 수행된다. 이 접근 방식의 한 응용 프로그램은 나중에 실행 가능한 소프트웨어에 적용될 수있는 테스트를 생성하여 테스트 프로세스를 적정 수준으로 자동화한다. 또 다른 응용 프로그램은 사용자 인터페이스를 애니메이션화하여 최종 사용자가 소프트웨어 개발자가 작업 할 사양의 자세한 의미를 이해할 수 있게 한다. 이것은 두 그룹 간의 소통 방법을 제공한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.26 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 4 소프트웨어 상세 설계

IEC61508-3 A.9.3 : 정적 분석 (Static analysis)

1. 목표

   조기 또는 수년간의 작동 후 고장을 초래할 수있는 시험중인 시스템의 고장을 방지한다.

2. 설명

   체계적이고 가능한 컴퓨터 지원 접근법은 프로토 타입 시스템의 특정 정적 특성을 검사하여 해당 요구 사항 (예: 설계 가이드라인, 시스템 사양 및 기기 데이터 시트)에 대한 완전성, 일관성, 모호성을 보장한다. 정적 분석은 재현 가능하며 정의 된 단계에 도달 한 프로토 타입에 적용된다. 하드웨어 및 소프트웨어에 대한 정적 분석의 몇 가지 예는 다음과 같다.

   - 데이터 흐름의 일관성 분석 (예: 데이터 객체가 어디에서나 같은 값으로 해석되는지 테스트)
   - 제어 흐름 분석 (예: 경로 결정, 액세스 할 수없는 코드 결정)
   - 인터페이스 분석 (다양한 소프트웨어 모듈 간의 가변 전송 조사 등)
   - 변수를 생성, 참조 및 삭제하는 시퀀스를 탐지하는 데이터 흐름 분석
   - 특정 가이드라인 (예: 연면 거리 및 공간 거리, 조립 거리, 물리적 단위 배열, 기계적으로 민감한 물리적 단위, 소개 된 물리적 단위의 독점적 사용)을 준수하는지 테스트한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.6.4 와 IEC 61508-3 의 표 B.8 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-4, B-7, B-8, B-10, B-20, B-56, C-6.6, C-6.15, C-6.16, C-6.17

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 A.9.7 : 오프라인 수치 분석 (Offline numerical analysis)

1. 목표

수치 계산의 정확성을 보장한다.

2. 설명

수치 적 부정확성은 이상적인 함수와 수의 유한 표현을 사용하여 수학적인 함수를 계산할 때 발생할 수 있다. 절단 오류는 함수가 푸리에 급수와 같은 무한 수열의 한정된 수의 항으로 근사 될 때 발생한다. 반올림 오류는 컴퓨터에서 실수를 정확하게 표시함으로써 발생한다. 가장 단순한 계산 이외의 다른 연산이 부동 소수점에서 수행되는 경우 계산의 유효성을 검사하여 응용 프로그램에서 요구하는 정확도가 실제로 달성되는지 확인해야한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.13 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2..1 단계 0 안전등급분류 및 식별, 단계 2 소프트웨어 요구사항 분석

IEC61508-3 B.8 – Static analysis

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	경계 값 분석		R	R	HR	HR
2	점검표		R	R	R	R
3	제어 흐름 분석		R	HR	HR	HR
4	데이터 흐름 분석		R	HR	HR	HR
5	오류 추측		R	R	R	R
6a	특정 기준을 포함한 공식 검사		R	R	HR	HR
6b	워크 쓰루 (소프트웨어)		R	R	R	R
7	상징적 실행		---	---	R	R
8	디자인 검토	IEC61508-3 B.8.1	HR	HR	HR	HR
9	런타임 오류 동작의 정적 분석		R	R	R	HR

10	최악의 실행 시간 분석		R	R	R	R
• 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다.						

IEC61508-3 B.8.1: 설계 검토(Design review)

1. 목표

   소프트웨어 설계의 결함을 찾는다.

2. 설명

   설계 검토는 설계 요구 사항을 평가 및 충족시키며 문제를 식별하고 해결책을 제시한다. 이는 설계 기능을 평가하기 위한 소프트웨어 설계에 대한 공식적이고 문서화된 포괄적이고 체계적인 검토를 의미한다.
   설계 검토는 입력 요구 사항에 대한 설계 상태를 평가할 수 있는 방법과 향후 개선 기회를 식별 할 수 있는 방법을 제공한다. 개발 라이프 사이클 활동이 진행되고 주요 세부 설계 일정이 충족되면 모든 인터페이스 측면을 검토한다. 설계가 요구 사항을 충족하고 안전요구사항에 부합하는지 설계 검토를 실시한다. 이는 주로 설계자의 작업을 검증하기 위한 것이고 확인(confirmation)과 세부 활동(refining activity)으로서 다룬다.
   잠재 장애 분석(sneak circuit analysis)과 같은 엄격 검사 기법은 예상불가능 경로 또는 논리 흐름, 의도하지 않은 출력, 잘못된 타이밍, 원하지 않는 동작 등과 같은 잘못된 소프트웨어 동작을 탐지하기 위해 사용한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-3 의 표 B.8 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-4, C-6.6

   ☞ 자동차 가이드: PART 3: SW 안전 개발 가이드 > PART 2. 테스트 메소드(Test Method) 가이드 > 2.6 테스트 기법(Test Techniques) 가이드 > 2.6.3 경계값 분석

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.3 : 제어 흐름 분석 (Control flow analysis)

1. 목표

   부족하거나 잠재적으로 잘못된 프로그램 구조를 탐지한다.

2. 설명

   제어 흐름 분석은 좋은 프로그래밍 습관을 따르지 않는 의심스러운 코드 영역을 찾는 정적 테스트 기술이다.

   - 액세스 할 수 없는 코드 (예 : 코드 블록을 도달 할 수없는 무조건 점프)
   - 매듭이있는 코드, 잘 구조화 된 코드에는 단일 노드에 대한 연속적인 그래프 축소로 줄일 수 있는 제어 그래프가 있다. 대조적으로, 제대로 구조화되지 않은 코드는 여러 노드로 구성된 매듭으로만 축소 된다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.9 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-8, C-6.15

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.4 : 데이터 흐름 분석 (Data flow analysis)

1. 목표

   부족하거나 잠재적으로 잘못된 프로그램 구조를 탐지한다.

2. 설명

   데이터 흐름 분석은 제어 흐름 분석에서 얻은 정보를 코드의 다른 부분에서 읽거나 쓰는 변수에 대한 정보와 결합한 정적 테스트 기술이며, 분석을 통해 다음을 확인할 수 있다.

   - 값을 할당 받기 전에 읽을 수 있는 변수는 새로운 변수를 선언 할 때 값을 할당함으로써 피할 수 있다.
   - 읽지 않고 한 번 이상 쓰여진 변수는 생략 된 코드로 나타낼 수 있다.
   - 쓰여 있지만 읽지 않은 변수는 중복 코드로 나타낼 수 있다.

   데이터 흐름 이상이 항상 프로그램 오류와 직접적으로 연결되는 것은 아니지만 예외가 발생하지 않으면 코드에 오류가 있을 가능성이 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.10 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-10, C-6.16

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.6a : 특정 기준을 포함한 공식 검사 (Formal inspections, including specific criteria)

1. 목표

   소프트웨어 요소의 결함을 밝힌다.

2. 설명

   공식 검사는 결함을 발견하고 생산자가 자료를 개선 할 수 있도록 자료를 생산하는 사람의 동료가 수행하는 소프트웨어 자료를 검사하는 구조화 된 프로세스다. 생산자는 숙지 단계에서 검사원에게 간단한 설명을 하는 것 이외에는 검사 과정에 아무런 영향을 미치지 않는다. 정식 검사는 소프트웨어 개발 생명주기의 모든 단계에서 생산 된 특정 소프트웨어 요소에서 수행 된다. 검사를 받기 전에 검사원은 검사 할 재료에 익숙해야하고 검사 과정에서 역할은 명확해야 한다. 검사 일정을 준비하고 시작 및 종료 기준은 소프트웨어 요소에 필요한 속성을 기반으로 정의한다. 시작 기준은 검사가 실시되기 전에 충족되어야하는 기준 또는 요구 사항이며 종료 기준은 특정 프로세스를 완료하기 위해 충족되어야하는 기준 또는 요구 사항이다. 검사하는 동안 검사의 결과는 중재자에 의해 형식적으로 기록되어야 한다. 모든 조사관은 결과에 대해 합의해야 한다. 결함은 인수 전에 수정을 요구하거나 주어진 시간 / 공정표에 의해 수정을 요구하는 것으로 분류되어야 한다. 확인 된 결함은 검사가 완료된 후에 후속 수정을 위해 생산자에게 보고되어야 한다. 확인 된 결함의 수와 범위에 따라 중재자는 소프트웨어 자료의 추가 검사를 위해 필요하다고 판단 할 수 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.14 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.6b : 워크 쓰루 (소프트웨어) (Walk-through (software))

1. 목표

   명세과 구현 간의 불일치를 밝힌다.

2. 설명

   워크 쓰루는 비공식적인 기술로서, 소프트웨어 요소 제작자가 소프트웨어 요소의 결함을 찾는 목적으로 동료들과 함께 수행한다. 소프트웨어 개발 생명주기의 모든 단계에서 생산된 특정 소프트웨어 요소에서 수행 될 수 있다. 안전 관련 시스템이 명세에 주어진 요구사항에 부합하는지 확인하기 위해 안전 관련 시스템의 특정 기능을 조사하고 평가한다. 제품의 구현 및 사용에 관한 의문점은 문서화되어 해결 할 수 있다. 공식 검사와 달리 작성자는 워크 스루 절차 중에 검사한다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.15 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-56, C-6.17

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.8 : 디자인 검토 (Design review)

1. 목표

   소프트웨어 설계의 결함을 밝힌다.

2. 설명

   설계 검토는 설계 요구 사항을 평가하고 요구 사항을 충족하여 문제를 식별하고 해결책을 제안 할 수있는 설계 기능을 평가하기 위한 소프트웨어 설계에 대한 공식적이고 문서화된 포괄적이고 체계적인 검토이다.

   설계 검토는 입력 요구 사항에 대한 설계 상태를 평가할 수 있는 수단을 제공하고 향후

개선 기회를 식별 할 수 있는 수단을 제공한다. 개발 라이프 사이클 활동이 진행되고 주요 세부 설계 일정이 충족되면 모든 인터페이스 측면을 검토하고 설계가 요구 사항을 충족하는지 확인하고 설계가 가장 적합한 지 확인하도록 설계 검토가 실시되어야 한다. 설계는 안전 요구 사항과 일치해야 하며 검토는 주로 설계자의 작업을 검증하기위한 것이므로 확인과 정제활동이 있어야 한다. "새치기 회로 분석"과 같은 엄격한 검사 기법을 사용하여 예상치 못한 경로 또는 논리 흐름, 의도하지 않은 출력, 잘못된 타이밍, 원하지 않는 동작 등과 같은 잘못된 소프트웨어 동작을 탐지 할 수 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.16 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-56, C-6.17

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

IEC61508-3 B.8.9 : 런타임 오류 동작의 정적 분석(Static analysis of run time error behaviour)

1. 목표

   공식 방법은 수학적 추론의 원칙을 기술 시스템의 명세 및 구현으로 옮겨서 명세나 구현의 완전성, 일관성 또는 정확성을 높인다.

2. 설명

   공식 방법은 명세 및 구현 단계에서 시스템에 대한 설명을 개발하는 수단을 제공한다. 이러한 공식 설명은 시스템 기능 및 구조의 수학적 모델이다. 모호하지 않은 시스템 기술이 달성되어 기본 시스템의 이해를 높일 수 있다. 적절한 공식 방법을 선택하는 것은 시스템, 개발 프로세스 및 사용 가능한 수학 모델의 범위를 완전히 이해해야하는 어려운 작업이다.

   모델의 속성은 시뮬레이션보다 높은 신뢰를 제공한다. 즉 시스템의 선택된 동작을 관찰하는 시스템에 대해 보증한다. 정형 기법의 단점은 다음과 같다.

   - 고정 된 추상화 수준
   - 주어진 단계에서 관련된 모든 기능을 포착하기위한 한계
   - 구현 엔지니어가 모델을 이해해야하는 어려움
   - 시스템의 수명주기 동안 모델을 개발, 분석 및 유지하기위한 높은 노력

- 모델의 구축 및 분석을 지원하는 효율적인 도구의 가용성
- 모델을 개발하고 분석 할 수있는 직원의 가용성

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 B.2.2 와 C.2.4 를 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.6 단계 5 소프트웨어 구현 및 단위 테스트

사. 사용 양식

- [SI-D-01] SW 단위시험 보고서
- [SI-D-02] SW 통합시험 보고서
- [SI-V-01] SW 시험 안전성분석 보고서

아. 적용 기법

- [SI-T-01] SW 단위시험 기법
- [SI-T-02] SW 통합시험 기법

제 3 장. 안전관련 SW 개발 기법

[SR-T-01] SW 요구사항 명세 기법

가. 개요

SW 요구사항명세(SRS: Software Requirement Specification)는 컴퓨터에 설치될 SW 에 대한 요구사항을 기술한 문서로서 SW 사용자나 SW 설계 담당자에 의해 만들어지는 개념 문서를 바탕으로 개발할 SW 에 기대되는 요구사항에 대해 기술한다. SRS 에는 SW 의 컴포넌트는 물론 SW 가 운용될 환경에 대 한 특정한 요구사항 및 SW 의 기능직인 특징 및 요구되는 특성 등을 기술한다. SRS 는 컴퓨터의 외적인 관점으로 SW 의 행위에 대해 어떻게 구현할 것인지에 대해 기술하지 않고 무엇을 할 것인가에 대해 블랙박스 개념을 도입하여 기술한다. 또한 SW 의 외부 이벤트의 행위에 대한 내부적 행위의 혹은 상태에 대한 기술도 포함할 수 있다.

SW 요구사항명세서에 기술되어야 할 사항은 다음과 같다.

- 컴퓨터 시스템과 컴포넌트로 이루어진 시스템 사이의 인터페이스를 정의한다. 이러한 인터페이스의 입력/출력은 시스템 입출력으로 하나의 변수명으로 명명한다.
- 컴퓨터 시스템과 개발되는 SW 컴포넌트 사이의 인터페이스를 정의한다. 이 인터페이스의 입출력은 SW 입출력으로 하나의 SW 변수명으로 명명한다.
- 시스템 입출력 변수명과 SW 입출력 변수명과의 관계를 정의한다.
- SW 입출력 변수에 의해 제어되는 SW 의 행위를 정의한다. 행위적 관점으로서 STATEMATE 의 경우 상태도(State chart)로 기술되고, 제어흐름도 등을 활용할 수 있다.
- SW 의 에러나 결함에 대해 요구되는 반응을 정의한다.

- SW 로 구현할 때 필요한 설계 제약 조건에 대해 기술한다. 이 제약 조건에는 신뢰성, 유지 보수성, 안전성 흑은 보안상의 요구사항을 기술하고 참고한 특정 표준을 나열한다.
- 항목은 기능적 관점으로서 STATEMATE 의 경우 액티비티차트(Activity chart), 컨텍스트다이어그램(Context Diagram) 및 자료흐름도로 기술할 수 있다.

나. 특성

- SW 요구사항은 정확해야 한다. 이를 위해 모든 기술된 명세 등과 상치되는 면이 없는지를 비교하여 확인한다.
- SW 요구사항은 모호하지 않아야 한다. 즉, 명세된 요건은 단일한 의미로 설명 가능해야 하고, 다중적인 의미로 해석되거나 모호하지 않아야 한다.
- SW 요구사항은 완전해야 한다. 이는 중요한 모든 요구 조건들이 기술되어야 하고, 유효한 입력뿐 아니라 유효하지 않은 SW 의 응답에 대해서도 기술되어야 함을 의미한다.
- SW 요구사항은 일관성을 유지해야 한다. 즉, 내부적으로 각각의 요구조건들이 서로 상충되지 않아야 한다.
- SW 요구사항은 검증 가능해야 한다. 즉, 구현된 SW 가 요구조건을 만족하는 지 확인할 수 있어야 하는데 이를 위해서는 모호한 요구조건이 없어야 한다.
- SW 요구사항은 쉽게, 완전하게, 일관성 있게 수정 가능해야 한다.
- SW 요구사항은 추적성이 있어야 한다. 이를 위해 모든 요구사항 명세는 전후단계에서 추적 가능하도록 참조 번호를 필요로 하며, 앞 단계의 명세를 참고하는 경우 이에 대해 언급한다.

다. SW 의 요구사항 작성 방법

컴퓨터 시스템에 대하여 SW 요구분석서를 작성하기 위한 절차는 다음과 같다.

- 컴퓨터 시스템과 SW 의 경계를 구별한다.
- 입력변수와 출력변수를 정의한다.

- 컴퓨터 시스템의 기능적인 행위를 정의한다.
- 컴퓨터 시스템의 시간적 요구사항을 정의한다.
- SW 의 요구되는 특성과 관련된 요구사항(예, Safety 등)을 정의한다.
- 사전과 목차를 정의한다.
- 원래의 요구사항의 추적성을 보인다.
- SRS 에 대한 임관성과 완전성에 대한 검사를 수행한다.

(1) 컴퓨터 시스템과 SW 경계의 구별

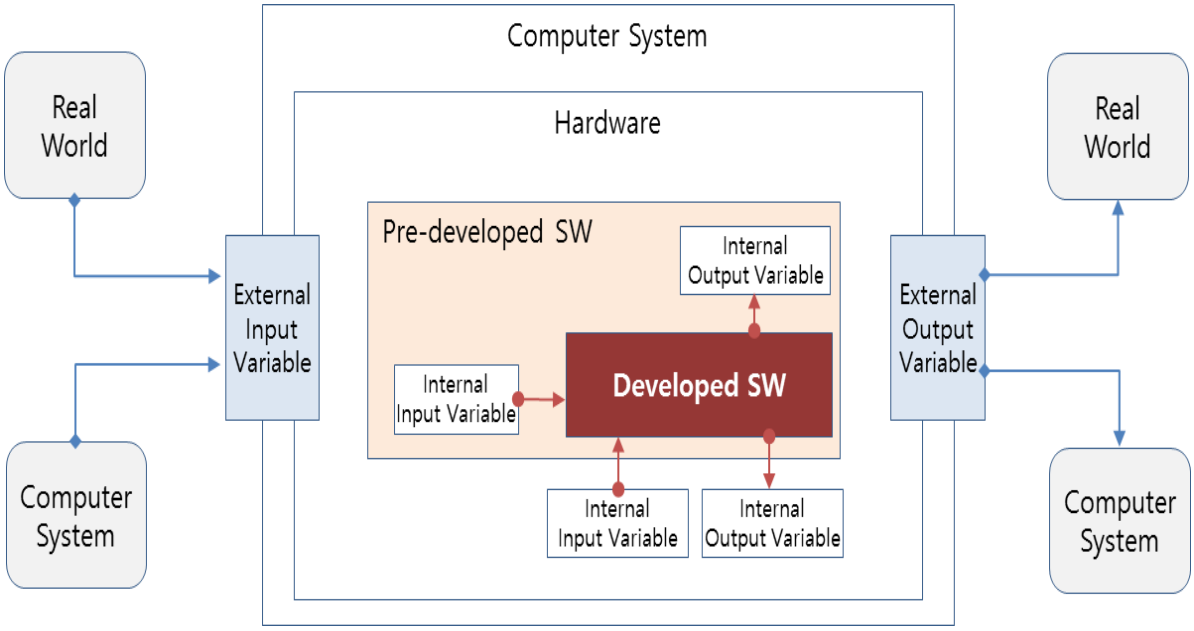
컴퓨터 시스템과 SW 의 경계를 명확하게 구별한다. 즉 컴퓨터 시스템 경계에서는 입력 및 출력변수를 결정하고 SW 의 경계에서는 내부입력변수 및 내부출력 변수를 결정해야만 한다. 컴퓨터 시스템의 경계는 개념서에 미리 정의할 수 있다. 이것은 SW 요구사항 작성자나 혹은 제 3 자가 모두 직관적으로 이해할 수 있어야 한다. SW 요구사항 작성자는 외부입력변수와 내부입력변수와의 관계의 복잡성과 내부출력변수 및 출력변수와의 관계의 복잡성이 컴퓨터 시스템 내에 포함되는 외부 컴포넌트들이 증가함에 따라 증가 할 수 있음을 인식해야 한다. 즉 외부 컴포넌트들의 복잡성이 입력 및 출력변수의 복잡성을 좌우함으로 인식해야 한다.

다음과 같은 원칙이 컴퓨터 시스템의 경계를 선택하는데 도움이 될 수 있다.

- 하위 시스템의 분할에 대해 컴퓨터 시스템의 경계가 개념서에 기술되어 있을 때에는 이를 따른다.
- 컴퓨터 시스템이 미리 정해진 하위 시스템의 경계에 의해 제한되지 않는다면, 컴퓨터 시스템의 경계는 그 경계에서의 입력 및 출력변수가 확실히 구별될 수 있는 위치에서 정해져야 한다. 컴퓨터 시스템의 행위는 입력 및 출력변수 측면에서 기술한다.
- 컴퓨터 시스템의 경계는 SDS-SRS 의 검증을 고려하여 선택되어야 한다.
- 특별한 고려 사항이 없을 때는, 컴퓨터 시스템의 경계는 컴퓨터 시스템의 HW 의 한계에 존재해야 한다.

개발될 SW 의 경계의 선택은 다소 직관적이다. SW 의 경계는 개발될 SW 와 컴퓨터 시스템(이미 개발된 SW 와 컴퓨터 HW)의 나머지 부분 사이에 존재하게 된다. 그림

IV.3.1 을 참조한다. 컴퓨터 시스템에 의해 계측되고 제어될 컴퓨터 시스템과 SW 경계, 그리고 외부 장치와 같은 시스템 문맥(System Context)은 액티비티차트(Activity Chart)와 컨텍스트다이어그램(Context Diagram) 등으로 기술할 수 있다.



개발될 SW 는 기존의 개발된 SW 와 HW 그리고 외부 입출력 인터페이스와 구별되어야 하며 이러한 문맥(Context)은 SRS 작성 초기에 정의한다.

(2) 입력 변수와 출력변수의 정의

컴퓨터 시스템과 SW 의 경계가 정해지면 컴퓨터 시스템의 인터페이스를 구성하는 입력변수와 출력변수를 정의한다. 입력 및 출력변수는 컴퓨터 시스템의 외부 장치의 인터페이스로부터 입출력 되는 외부 입출력 변수와, 컴퓨터 시스템(이미 개발된 SW 혹은 HW)으로부터 개발되는 SW 로 입출력 되는 내부 입출력 변수로 구분된다.

외부입출력변수로는 온도 혹은 압력 등의 아날로그 변수와 디지털 입출력값 혹은 버튼 등의 디지털 변수들이 포함된다. 일반적으로 외부 입출력변수의 값은 컴퓨터 시스템 경계에 위치하게 된다.

내부입력변수는 컴퓨터 시스템 내의 현재 시간이나 컴퓨터 시스템의 "FAIL" 신호가 될 수 있다. 내부출력변수로는 HW 를 제어하는 상태 변수가 된다.

각각의 입력변수는 다음 사항을 포함해서 기술한다.

- 입력변수 인터페이스에 대한 외부 장치 혹은 컴퓨터 시스템 내부 장치들
- 입력변수를 정의하는 특성(attribute)
- 입력변수와 관련된 내부입력변수와 SW 경계에서의 그 내부입력변수를 정의하는 특성
- 입력변수와 내부입력변수와의 관계

출력변수는 다음 사항을 포함해서 기술한다.

- 출력변수 인터페이스에 대한 외부 장치 혹은 컴퓨터 시스템 장치들
- 출력변수를 정의하는 특성(attribute)
- 출력변수와 관련된 내부출력변수와 SW 경계에서의 내부출력변수를 정의하는 특성
- 내부출력변수와 출력변수 사이의 관계.

(3) 컴퓨터 시스템 기능적 행위의 정의

컴퓨터 시스템의 기능적 행위는 과거 혹은 현재의 입력변수의 값에 대한 출력변수의 값을 정의함으로 표현될 수 있다. 컴퓨터 시스템의 기능적 행위의 명세는 주로 액티비티차트(Activity Chart) 및 자료흐름도로 기술되며, 이들 간의 시작 및 종료 혹은 각 액티비티의 수행 조건 들을 기술하는 컨트롤차트(Control Charts) 및 제어흐름도로 기술된다. 또한 컴퓨터 시스템의 기능의 보다 구체적인 행위에 대한 요구사항은 Statecharts, SDL, 제어흐름도 등으로 기술된다. 다음과 같은 절차를 포함한다.

- 기능들을 필요한 만큼 적절하게 분할한다.
- 분할된 기능을 (정형적으로) 기술한다.
  - 기능의 입력변수와 출력변수의 의존관계를 기술한다.
  - 기능들과의 관계를 기술한다.
  - 각각 출력변수에 대해, 기능에 대한 의존성을 기술한다.
  - 기능의 시작 및 출력, 수행 조건을 기술한다.
- 기능에 대해 일관정과 완전성을 체크한다.

각각에 대한 자세한 설명은 다음과 같다.

(가) 기능들의 분할

전체의 SW 의 기능은 세분화된 기능들로 분할할 필요가 있다. 이는 매우 중요한 절차 중 하나이다. 이는 명세자의 작업을 쉽게 할 뿐 아니라, 명세된 행위의 이해와 검토를 용이하게 한다. 분할을 너무 적게 하면 각각의 기능들은 상당히 복잡해지고, 반대로 분할을 너무 지나치게 하면 각각의 기능들은 이해하기 쉬우나 각각의 기능적 의존도가 복잡해진다. 따라서 명세자, SW 설계 담당자, 검토자가 모두 컴퓨터 시스템의 행위의 완전한 기능을 이해하는 것이 중요하다.

다음과 같은 이유로 기능들은 세분화 되어야 한다.

- 특정 기능을 분할 없이 이해하기는 상당히 어렵다. 따라서 SRS 작성자는 어느 정도가 복잡한지를 스스로 결정해야 한다.
- 기능 내에는 타이밍 요구사항을 보일 필요가 있다.
- 채널화와 같은 시스템의 요구사항은 필요로 한 만큼 상당히 원하는 만큼 분할 할 수 있다.
- 기능들은 공통된 기능적 요구사항을 공유한다.
- 문제의 영역 내에서 분할이 이루어져야 한다.

가능한 분할은 하향식으로 전개하여 그 기능을 보는 사람에게 SW 가 완전하고 내부적으로 일관성 있고, 행위적인 명세에 대해 정확성을 보여준다.

예를 들어, 경보 시스템은 다음과 같이 그 기능성을 명세할 수 있다.

- 경보시스템은 외부 센서로부터 입력을 받는다.
- 연속적으로 신호들을 측정하고 계산하여 결과를 출력한다.
- 연산기에 의해 결정된 특정한 범위 안에 결과 값이 있는지를 체크한다.
- 만약 그 값이 범위를 벗어난다면, 그 시스템은 경고 메시지를 운영자를 위한 화면에 보여주고 알람을 울릴 것이다.
- 만약 주어진 시간 내에 운영자가 반응을 하지 않으면, 시스템은 출력기구에 fault 메시지를 보여주고 신호의 모니터링을 중단한다.

경고 시스템의 기능적 명세의 첫 번째 단계는 먼저 문장으로 적는 것이다. 위의 기능을 기술하면 다음과 같다.

- SETUP: 운영자로부터 범위의 한계를 입력 받는다.
- PROCESS_SIGNAL: 센서로부터 계산되지 않는 신호를 입력 받고, 범위의 한계와 비교될 값으로 프로세싱한다.
- COMPARE: 범위의 한계와 프로세싱된 값과 비교한다.
- DISPLAYJ'AIJLT: 운영자 화면에 경고 메시지를 보내고, 알람을 울린다.
- PRINT_FAULT: Fault 메시지를 출력팅 기계에 출력한다.

이러한 명세에는 이러한 기능들이 다루게 될 데이터를 포함하는 행동들의 명세가 포함된 다. 즉 입력 데이터와 출력 데이터를 포함하고, 다른 기능들에 의해 이러한 입/출력 데이터들이 사용되는지 기술한다.

즉 PROCESS_SIGNAL 은 입력을 받아 COMPARE 에 의해 비교될 수 있는 값으로 프로세싱 되지 않는 데이터를 변환한다.

(나) 분할된 기능의 (정형적) 명세

① 기능의 의존성 및 입력 및 출력 변수의 의존성 구별

우선 기능들이 어떠한 관계를 가지고 진행되는지 기술한다. 즉 어떤 기능들이 또 다른 어떤 기능에 영향을 주는지를 기술하게 된다. 각 기능에 대한 입출력 흐름을 기술한다. 입출력 변수의 의존성을 구별하고 파악해야 한다. 입력변수와 출력변수의 의존성을 구별하기 위해서, 각각의 출력변수에 대하여 그 행위를 결정짓는 입력변수를 찾아야 하고 또한 각각의 입력변수는 어떤 출력 변수에 영향을 주는 지를 구별해야 한다. 어떤 입력변수가 어떤 출력변수에 혹은 영향을 주는지 혹은 어떤 출력변수가 또 다른 입력변수에 영향을 주는지에 대한 의존성 명세를 한다.

② 기능에 대한 행위 명세

각각의 입출력 변수에 대하여 그 기능의 행위를 기술할 필요가 있다. 입출력 변수를 기준으로 기능들 간의 의존 관계 및 기능의 시작 및 종료, 기능의 수행의 조건과 같은 기능들을 제어하는 행위를 기술한다. 하지만 이러한 명세는 그 수준에 따라 추상화 될 수 있으며 추상화 수준은 SRS 작성자에 따라 다르다.

(4) 컴퓨터 시스템의 시간적 요구사항에 대한 정의

시스템의 시간에 대한 요구사항은 입력변수에 대한 출력이 나오는 것에 대한 시간적 제약 혹은 어떠한 기능들의 수행이 만족해야 하는 시간의 제약을 가리킨다. 즉 어떤 입력에 대해 어떠한 출력이 나오기까지의 시간적 제약을 기술하는 것이다. 이것은 순차다이어드램(Sequence Diagram) 등을 이용하여 특정 입력에 대한 시간의 제약 조건을 기술하면서 구체화한다.

(5) SW 설계 제약사항 정의

SW 설계에 고려해야하는 제약조건을 기술한다.

- HW 환경과 관련 있는 제약사항

- SW 환경과 관련 있는 제약사항
- 신뢰성 요구사항
- 유지보수성 요구사항
- 안전성 요구사항

(가) HW 환경의 제약사항 정의

- 컴퓨터의 연산속도
- 메모리 크기
- 사용 가능한 인스트릭션 수
- 외부 포트의 수
- 컴퓨터와 인터페이싱 하는 아날로그 혹은 디지털 입력 혹은 출력 수

(나) SW 환경의 제약사항 정의

- 사용되는 프로그램 언어
- 사용되는 컴파일러
- 사용되는 프로그램 기능의 제약사항

(다) 신뢰성 요구사항 정의

신뢰성에 대한 요구사항은 SW 가 잘못된 반응을 하게 될 확률적 기술이다. 이것은 SW 요구사항 작성자와 안전성 요구사항 작성자들이 의논한다. 이것은 단순히 통계적 확률적 수치를 의미할 수 있다.

(라) 안전성 요구사항 정의

안전성 요구사항은 SW 혹은 시스템이 안전한 상태가 아닌 상태로 이끄는 고장의 결과를 막거나 완화시키는 데 영향을 준다. 즉 시스템의 안전을 위해 특별히 중요한 어떤 입력에 대해 반드시 출력되어야 하는 출력과 그 출력과 관련되어 수행되어야 하는 기능을 기술한다.

[SR-T-02] SW 요구사항 위험원 분석 기법

가. 요구사항 위험 분석

SW 요구사항 위험 분석에서는 SW 요구사항명세서(Software Requirement Specficication) 가 시스템 위험원에 미칠 영향을 조사한다. 요구사항은 일반적으로 여러 개의 세트로 구분될 수 있으며, 각 세트는 SW 의 어느 한 관점을 기술하게 된다. 이와 같은 세트들을 총칭해서 품질속성(qualities)이라고 부른다. SW 위험 분석에서 고려되어야 할 품질 속성들은 정확도 (accuracy), 용량(capacity), 기능성(functionality), 신뢰도(reliability), 강인도(robustness), 안전성(safety), 보안(security) 등이 있다. 이들 속성을 특정한 응용분야에 적용하려면 해당 분야 및 시스템에 맞게 수정해서 사용한다.

나. 방법

SW 요구사항 위험 분석의 목적은 각 품질속성과 그 속성의 각 요구사항을 조사하고, 위험원에 미칠 영향을 평가하는 것이다. HAZOP 기법에서는 영향을 평가하기 위한 지침 단어(guide words)의 사용을 주천하고 있다. 또한 일련의 지침 문구(phases)가 각 품질속성에 대해 제공되며 그 속성은 관련된 각 요구사항의 위험원에 미치는 영향을 평가하는데 사용될 수 있다.

지침 문구는 표 IV.2.1 에 있다. 이는 특정한 SW 품질속성에 관련한 각 요구사항에 대해 조사되어야 할 개념들을 제안하고 있다. 어떤 경우는 한 요구사항이 여러 개의 품질속성에 영향을 미칠 수 있다. 어떤 경우에서는 품질 속성들이 다양성 관점에서 더욱 세분화하고 있다. 셋째 칼럼은 지침 문구의 사용을 추천하는 수명주기 단계 코드를 기술하고 있다.

표 IV.2.2 SW 품질속성에 관한 지침 문구(NUREG/CR-6430)

품 질	관 점	단 계	지침 문구
정확도 (accuracy)	센 서	RADC	모든 영점에서 Stuck
		RADC	모든 1 점에서 Stuck
		RADC	아무 곳에서 Stuck
		RADC	최소 범위 이하
		RADC	최대 범위 이상
		RADC	범위 안에 있으나, 잘못됨
		RADC	물리 단위가 부정확함
		RADC	잘못된 데이터 유형이나 데이터 크기
		RADC	모든 영점에서 Stuck
	작동기	RADC	모든 1 점에서 Stuck
		RADC	아무 곳에서 Stuck
		RADC	최소 범위 이하
		RADC	최대 범위 이상
		RADC	범위 안에 있으나, 잘못됨
		RADC	물리 단위가 부정확함
		RADC	잘못된 데이터 유형이나 데이터 크기
	운전원 입력 및 출력	RA	허용 범위 이하의 수치
		RA	허용 범위 이상의 수치
		RA	허용 범위 안에 있으나, 잘못됨
		RA	수치가 잘못된 물리 단위를 가짐
		RA	수치가 잘못된 데이터 유형이나 데이터 크기를 가짐
		RA	수치가 아닌 값이 부정확함
		RDC	계산결과가 허용 오차 한계를 벗어남(너무 낮음)
		RDC	계산결과가 허용 오차 한계를 빗어남(너무 높음)
	계 산	RDC	공식 또는 방정식이 잘못됨
		RDC	물리 단위가 부정확함
		RDC	잘못된 데이터 유형이나 데이터 크기
용량 (capacity)	메시지	RADC	메시지 양이 정해진 최소 값보다 낮음
		RADC	메시지 양이 정해진 최대 값을 초과함
		RADC	메시지 양이 일정치 않음
		RADC	메시지 비율이 정해진 최소 값보다 낮음

	타이밍	RADC	메시지 비율이 정해진 최대 값올 초과함
		RADC	메시지 비율이 일정치 않음
		RADC	메시지 내용이 부정확하나, 그런대로 쓸 만함
		RADC	메시지 내용이 엉망임
		RADC	입력 신호가 도착 실패함
		RADC	입력 신호가 너무 자주 발생함
		RADC	입력 신호가 너무 늦게 발생함
		RADC	입력 신호가 예상치 않게 발생함
		RADC	시 스 템 거동이 결정 적 (deterministic) 이 지 못함
		RADC	출력 신호가 작동기까지 도착 실패함
		RADC	출력 신호가 너무 빠르게 도착함
		RADC	출력 신호가 너무 늦게 도착함
		RADC	출력 신호가 예상치 않게 도착함
		R	운전원 조치를 위한 충분한 시간이 주어지지 않음
		AD	처리(processing)가 부정확한 순서에 따라 이루어짐
		DC	코드가 비종결(non-terminating> 루프에 빠짐
		DC	데 드 록(deadlock) 이 발 생 함
		C	인터럽트가 제어 정보률 상실함
		C	인터럽트가 데이터를 상실함
기능성 (functionality)		RA	기능이 규정대로 수행되지 않음
		RA	기능이 실행되기 전에 적절히 초기화되지 않음
		RA	기능이 개시조건이 만족되지 않을 때 실행함
		RA	개시조건은 만족되었으나, 기능이 실패함
		RA	기능이 종료조건이 되었는데도 계속 실행함
		RA	종료조건이 만족되지 않았으나, 기능이 종결함
		RA	기능이 필요한 동작, 계산, 사건 등이 종결되기 전에 종결함
		R	기능이 부정확한 운전모드에서 실행됨
		R	기능이 부정확한 입력을 사용함
		R	기능이 부정확한 출력을 생산함
신뢰도(reliability)		RA	SW 가 요구보다 덜 신뢰함
		RA	SW 가 요구보다 더 신뢰함
		RA	시스템이 사용에 들어가도 SW 신뢰도가 밝혀지지 않음
		RA	SW 가 요구대로 점진적으로 저하되지 않음
		RA	SW 결함허용요건이 만족되지 않음

	RA	신뢰도가 운전모드에 따라 변함
	R	SW 가 가동 중 시험에서 실패함
	R	SW 가 고장 남
	A	HW 장치가 고장 남
	A	SW 고장이 관련되지 않은 공정으로 전파됨
	A	SW 가 고장으로부터 복구 실패함
	A	HW 또는 SW 고장이 운전원에게 보고되지 않음
	A	SW 가 부적당한 운전원 동작을 검출하지 못함
	AD	데이터가 부정확한 공정으로 전달됨
강인도(robustness)	RA	SW 가 예상치 못한 입력 데이터 출현 시 고장
	RA	SW 가 부정확한 입력 데이터 출현 시 고장 남
	RA	SW 가 비정상 조건 발생 시 고장 남
	RA	SW 가 요구 시에 자체 복구를 실패함
	RA	SW 가 메시지 과부하에서 고장 남
	RA	SW 가 메시지 상실시 고장 남
안전성(safety)	RA	SW 가 시스템을 위험한 상태로 몰고 감
	RA	SW 가 시스템올 위험한 상태로부터 위험하지 않은 상태로
	RA	SW 가 비상시 안전정지 개시률 실패함
	RA	SW 가 위험한 원자로 상태를 인식하지 못함
보완(security)	RA	비인가 된 사람이 SW 시스템을 함부로 이용함
	RA	비인가 된 SW 변경사항이 함부로 이루어짐
	RA	비인가 된 데이터가 함부로 이루어짐

지침 문구에 사용된 문자는 다음과 같다.

- R: 요구사항
- A: 구조 설계
- U: 상세 설계
- C: 코딩

수록된 문구들 외에도 안전 분석 담당자는 그 요구사항이 위험원에 미치는 영향을 조사해야 한다. SW 요건명세서에서 언급된 몇 가지 품질 속성들은 표 Ⅲ.3.1 에 포함되어 있지 않다. 즉 완전성, 일관성. 정확성, 추적성, 명확성 등에 대한 분석과

확인도 필요하다. 이는 요구사항 분석과 확인의 일부로써, 위험원 분석의 대상은 아니다.

예를 들어, 센서의 정확도를 살펴보면, 특정한 센서에 대한 정확도 요건, 즉 "센서 123 의 값이 100 에서 500 사이에 있어야 하고, 그 오차가 5% 보다 크지 않아야 한다'" 라면 다음 과 같은 질문들이 제기될 수 있다.

- 만약 그 센서의 값이 100 보다 적다면, 위험원에 어떤 영향을 미치는가?
- 만약 그 센서의 값이 500 보다 크다면, 위험원에 어떤 영향을 미치는가?
- 만약 그 센서의 값이 100 에서 500 사이에 있지만, 실제 값에 5% 이내에 들지 않으면, 위험원에 어떤 영향을 미치는가?
- 만약 센서 판독이 위의 요건을 만족한다면, 위험원에 어떤 영향을 미치는가? 특히 그 판독치가 실제 값에서 5% 벗어나면 어떤 영향이 있는가?
- 만약 그 센서가 zero 값에서 움직이지 않는다면, 위험원에 어떤 영향을 미치는가?
- 만약 그 센서가 완전히 고장 나면, 위험원에 어떤 영향을 미치는가?

이 분석에서 그러한 상황이 어떻게 발생할 수 있는가 또는 그러한 상황이 불가능한 것으로 생각하고 문제를 회피하는 것은 적절치 않다. 위험원분석에서는 그러한 상황이 발생 할 수 있다고 가정하고 그 결과를 면밀히 조사하여야 한다.

(1) 위험 분석 필요정보

다음과 같은 정보가 요구사항 위험 분석의 수행에 필요하다.

- 예비 위험원 목록 (PHL)
- 예비 위험원 분석 (PHA)
- 안전성 분석 보고서 (SAR)
- 보호 장치의 설계 내용
- . SW 요건 명세서 (SRS)

(2) 위험 분석 절차

다음과 같은 절차가 요구사항 위험 분석시 수행된다.

- SW 가 어떤 방식으로든지 대처해야 할 위험원들을 파악한다. 이것은 각 위험원과 관련된 리스크 추정치를 포함한다.
- 각 위험원과 제어범주에 관한 SW 필수성(criticality) 수준을 확인한다.
- 각 요구사항에 대한 필수성 수준을 배정하기 위하여 SW 요건명세서의 각 안전성 필수 요구사항을 시스템 위험원과 위험원 범주들에 대해 서로 대응시킨다.
- 'R'로 표시된 지침 용어들을 이용해서 각 요구사항을 분석한다.
- 분석결과를 문서화한다.

이와 같은 위험원분석 동안에 수집된 정보는 나중에 SW 개발 중에 지속적으로 활용될 수 있다. 각종 SW 요구사항에 배정된 필수성 수준과 지침 문구 분석의 결합으로 인해 수집된 정보는 개발, 확인 및 시험 단계에 자원 배정 시 유용하게 사용될 수 있다. 때로는 SW 에 의한 위험원을 줄이기 위해 응용 시스템을 다시 설계해야 할 필요성을 제기할 수도 있다.

SW 요구사항 위험 분석은 시스템 설계에서 어떠한 사항이 변경되어야 할 것인지를 추가로 제시할 수 있다. 특히, SW 에 배정된 시스템 요구사항은 HW 으로 구현해서 더 나은 결과를 얻을 수 있을 수 있다. 또한 요구사항 위험 분석에서는 어떠한 위험원도 초래하지 않은 요구사항을 찾아서 안전성에 영향을 미치지 않는다는 결론을 내리고, 그러한 요구사항들을 다음의 분석단계에서는 고려할 필요가 없다.

위의 분석 단계에서 여러 가지 분석 기법들이 적용될 수 있다. 가장 보편적인 기법은 고장수목분석(FTA)이다. 또한 수목의 정점 사건으로서 지침 문구들을 이용해서 결말에 이르기까지 그 수목을 확장해 나가는 사건수목분석(ETA)을 고려할 수도 있다. 기법의 선정은 어떤 정보가 분석가에게 제공 가능한지, 또한 어떤 정보가 수집되었는가에 따라서 달라진다.

(3) 위험 분석 결과물

요구사항 위험 분석 결과는 다음과 같다.

- SW 위험원의 목록
- SW 에 의해 영향을 받을 수 있는 각 위험원에 대한 필수성 수준
- 각 SW 요구사항에 대한 필수성 수준
- SW 가 각 요구사항에 견주어 정확하게 또는 부정확하게 동작할 때에 그 SW 의 위험원에 미치는 영향 분석

[SD-T-01] SW 설계 기법

가. 개요

SW 설계명세(Software Design Specification: 이하 SDS)는 SW 시스템에 대하여 IEEE Std 1016-1998 에 기술된 요구조건에 따라 기술한다. 이것은 SW 의 요구사항을 SW 구조, SW 의 컴포넌트, 인터페이스 및 구현단계에서 필요한 데이터와 관련된 사항으로 바꾸어 기술한다. 특히 SW 설계 사항은 구현작업을 위한 구체적인 알고리즘을 제시하고 요구사항으로부터 추적될 수 있어야 한다. SW 설계 단계의 주목적은 SW 요구사항을 근거로 SW 요구사항으로부터 추적할 수 있는 사항을 만들어 내는 것이다. 또한 이로부터 SW 코드를 만들어 낼 수 있도록 완전하고(complete), 일관성(consistent)있고, 올바르고(correct), 시험가능하고 (testable), 이해할 수 있는(understandable) 정보를 만드는 것이다. 이 과정에 서는 SW 요구사항 등 이전 단계에서 개발된 문서가 사용된다. SW 설계 단계는 SW 설계 단계의 직접적인 목적과 SW 를 보증하기 위한 절차가 동시에 수행된 다음 마치게 된다.

다음은 SW 설계 단계에 수행해야 할 작업이다.

- SW 요구사항을 각각 설계 컴포넌트로 분할한다.
- SW 시스템을 코딩을 위해 필요로 한 만큼 자세하게 SW 컴포넌트들로 분할한다.
- 각 컴포넌트의 외부 및 내부 인터페이스를 기술한다.
- 안전성 및 보안성과 관련된 컴포넌트들을 구분하여 표시한다.
- 그 SW 에 대한 가정, 가정에 대한 응답 및 시스템에 요구된 다른 알고리즘과 안전성 체크 및 시스템 수행에 역행하지 않는 fault tolerance protection 을 설계한다.

- 추적성 메커니즘을 구현한다. 즉 SW 설계와 SW 요구사항 및 SW 문서 간의 연결 고리를 만드는 절차를 수행한다.
- SW 설계가 그 요구사항 및 품질을 평가 하는데 사용될 기준을 정의한다.
- SW 설계에 대하여 이해성(understandability), 정확성(correctness), 시험가능성(testability), 일관성(consistency), 완전성(completeness) 및 SW 요구사항 단계에서 정의된 다른 품질에 관련된 성결과 관련하여 SW 설계사항을 분석한다.
- 시험 가능성에 대하여 SW 설계사항을 평가한다.
- SW 설계와 관련된 주목한 만한 문제점을 SW 설계단계 결과에 포함한다.
- 필요하다면 SW 설계 사항을 수정한다.
- SW 설계 사항을 도출한다.

나. 방법

(1) SW 설계 개체

SW 설계 개체는 설계의 요소(elements)이며 이 설계 요소는 구조적으로 혹은 기능적으로 다른 요소등과 구별되고 각각 다르게 이름이 붙여진다. 설계 개체는 SW 요구사항을 분할함으로써 얻어지는데 이러한 분할의 목적은 전체 시스템을 보다 작은 컴포넌트들로 나누고 다른 요소들과의 영향을 최소화하면서 테스트하기 위함이다. 나누어진 컴포넌트를 가지고 실제 구현을 하게 된다.

개체는 전체 시스템, 서브시스템, 데이터 저장, 모듈, 프로그램. 절차로서 존재한다. 설계를 분할하기 위해 요구된 이 개체의 개수와 타입은 시스템의 복잡도 혹은 설계의 기술 또는 프로그래밍 환경과 같은 다양한 요소들에 의존한다. 각 개체들은 서로 다른 성질을 가지고 있기도 하고 서로 공통된 성질을 지니기도 한다. 각 설계 개체는 하나의 이름, 목적, 기능을 갖는다. 그리고 인터페이스 혹은 공유데이터와 같은 관계 개체 간의 공통 관계도 존재하는데, 개체 간의 공통 특징은 설계 개체 속성으로 기술한다.

(2) 설계 개체 속성

설계 개체 속성은 설계 개체의 특징이나 성질로서 각 개체에 대한 사실을 기술한다. 개체의 속성에 대한 기술은 요구사항에서 고려한 제약들과 개체가 어느 곳에 적용되어서 동작하는지에 대한 가정들을 포함해야 한다. 즉 개체들이 활동할 수 있는 환경에 대해서 기술하는 것이다.

개체의 속성 및 속성과 관련된 정보에 대해 표 IV.3.1 과 같이 구분하여 기술할 수 있다.

표 IV.3.1 설계 개체 속성

개체 속성	관련 정보
(1) Identification	각 개체의 이름, 다른 개체와 구별되는 이름이어야 한다. 그리고 이 이름은 그 속성을 대표할 만한 이름이어야 한다.
(2) Type	각 개체의 타입에 대한 기술이다. 타입 속성은 개체의 특성을 기술한다. 예를 들어, 서브프로그램], 모듈, 프로시저, 프로세스 혹은 데이터스토어와 같은 종류의 이름으로 단순히 이름이 붙여진다.
(3) Purpose	목적은 각 개체들의 역할이 무엇인지에 대해서 기술한다. 이 속성은 각 개체의 생성을 위한 근본적인 이유를 제시하고 개체가 만들어 졌을 때를 위한 특수한 기능 혹은 수행 요구사항을 제시하게 된다. 또한 각 개체 가 만족시켜야 하는 특정 요구사항 역시 기술되어야 한다.
(4) Subordinates	각 개체간의 종속관계를 기술해야 한다. 종속관계는 각 개체간의 관계를 규명한다. 이것은 수직적인 종속관계를 의미하며 이 정보를 통해 개체를 설계로부터 요구사항으로의 추적에 이용된다. 또한 SW 를 분할하는데 있어서 상하 구조적인 관계를 규명하는데 이용된다.
(5) Dependencies	다른 Entities 와 특정 개체와의 관계에 대한 기술이다. 의존 속성은 하나의 개체를 위한 관계의 유무의 사용 혹은 요구를 규명한다. 이러한 관계는 structure charts 나, data flow diagrams, 혹은 transition diagram 을 도식적으로 기술될 수 있다. 의존 속성은 또한 상호 작용을 위한 시간적 혹은 조건적인 성질을 포함한 상호작용의 특징을 묘사한다. 이러한 상호작용은 개체의 초기화, 수행의 순서, 자원의 공 유, 생성, 복제, 사용, 저장 혹은 파괴를 포함할 수 있다.
(6) Interface	어떻게 하나의 특정 개체가 다른 Entities 와 상호작용 하는가에 대한 기술이다. 인터페이스 속성은 이러한 상호 작용을 하는 방법 및 규칙을

	기술해야 한다. 상호작용의 방법은 개체를 깨우거나 가로채는 메커니즘, 매개변수, 공유 데이터 영역. 혹은 메시지를 통하여 통신하는 메커니즘 혹은 내부 데이터를 직접적으로 접근하는 메커니즘을 포함할 수 있다. 상호작용을 제어하는 규칙은 통신 프로토콜, 데이터 포맷, 수용 가능한 데이터. 각 값의 의미를 포함한다. 이 속성은 또한 입력 범위, 입출력의 의미, 각 입출력의 타입과 포맷 그리고 출력 에러 코드를 제공한다.
(7) Resource	설계에 대하여 외적이 개체에 의해 사용되는 요소에 대한 기술이다. 이 속성은 물리적인 외부 장치를 가리키는 것으로 SW 와 관련 있는 입출력 장치를 가리킨다. 이 속성에는 어떤 자원을 획득할 때의 프로세스 시긴', 버퍼 사용의 물리적인 크기 등을 기술할 수 있다. 이 속성은 잠재적인 경쟁과 대드락 조건을 규명해야 한다, 그에 더하여 자원의 관리 기능도 기술해야 한다.
(8) Processing	개체의 기능을 수행하기 위해 개체에 의해 사용되는 규칙을 기술한다. 이 기술에 는 어떠한 기능을 수행하기 위한 알고리즘이 기술된다. 이 속성에는 타이밍, 이벤트 혹은 프로세스의 순서, 프로세스 초기화의 전제조건. 이벤트의 우선순위, 처리레벨, 실체 처 리 과정, path conditions 및 순환 및 순환을 종료하기 위한 기준이 포함된다. 그리고 사고의 처리는 오퍼플로우 혹은 validation check failure 와 같은 경우의 처리를 기술한다.
(9) Data	개체에 대하여 내부적인 데이터요소를 기술한다. 데이터 속성은 표현 방법, 초기 값. 사용, 의미, 포맷 및 내부 데이터로서 수용 가능한 값을 기술한다. 데이터에 대한 기술은 데이터 사전의 형태로 될 수 있다. 이는 모든 데이터 요소 의 내용. 구조 및 사용을 기술한다.

[SD-T-02] SW 구조 위험 분석 기법

가. 개요

SW 설계 위험 분석은 두 부분으로 나누어진다. 하나는 컴퓨터시스템 구조를 조사하는 것이고, 다른 하나는 SW 상세 설계를 조사하는 것이다. 컴퓨터시스템의 구조는 세 부분으로 이루어진다. 즉 HW 구조, SW 구조와 그것들의 매핑(mapping)이다. HW 구조는 여러 가지 HW 컴포넌트, 즉 프로세시, 메모리, 디스크 구동기, 표시기와 통신선로 등을 포함한다. SW 구조는 다양한 SW 공정, 데이터 저장, 스크린 배치와 논리적 통신경로 등이다. 매핑은 SW 가 HW 에서 어떻게 동작할 것인지 기술한다. 이것은 절차들이 다른 어떤 프로세서들 상에서 동작할 것이고. 각종 데이터 등이 어디에 저장되고, 각종 화면이 어디에서 디스플레이 되고, 그리고 논리적 통신은 물리적 통신선로 상에서 어떻게 이루어질 것인지를 기술한다.

어떤 구조는 다른 구조들이 갖지 않는 복잡한 기능을 갖거나, 고장모드들을 포함할 수도 다. 이것은 설계 명세서에 따라 생긴 추가적인 위험원들이며 앞의 위험원 분석에서는 확인되지 않은 것들이다. 구조설계문서는 요구사항과 설계요소들 간에 양방향 추적이 가능해야 한다. 즉 각 요구사항은 그 요구사항을 구현하는 설계요소들로 추적되고, 각 설계요소는 그 구현을 요구하는 요구사항으로 역추적이 가능해야 한다. 만약 이와같은 추적이 가능하지 않다면, 구조설계 위험원분석이 착수되기 이전에 추적이 가능하게 만들어야 한다.

여기서 분석은 요구사항 위험 분석에 이어서 SW 구조에 이르기까지 확장해 간다. 유사한 분석이 HW 구조와 컴퓨터시스템 전체구조(즉 HW, SW 와 매핑)에 대해서도 수행되는 것이 바람직하다.

나. 방법

(1) SW 구조 위험 분석의 입력

다음과 같은 정보가 SW 구조 위험 분석을 수행하는데 필요하다.

- 예비위험원목록 (PHL)
- 예비위험원분석(PHA)
- 안전성분석보고서 (SAR)
- SW 요건명세서 (SRS)
- SW 요구사항 위험 분석
- 구조추적매트릭스에 대한 요건
- SW 구조내용

(2) 분석 절차

다음과 같은 절차로 SW 구조 위험 분석을 수행한다.

1. 각 SW 구조요소에 대해 그 요소에 의해 영향을 받은 모든 요구사항들을 결정한다. 이는 표 III.3.3 의 추적매트릭스로부터 나온다.
2. SW 구조요소에 의하여 영향을 받은 모든 요구사항들과 관련된 리스크를 기반으로 각 SW 구조요소에 대한 리스크 수준을 배정한다. 표 IV.3.2 는 이를 위한 한 가지 방법을 보여주고 있다. 전 단계에서에서 구한 리스크 수준을 사용하고, 그 요소에 대한 리스크를 배정하기 위해 그 요소에 의해 영향을 받은 다양한 리스크의 요구사항 수량을 고려한 것이다. 다음과 같은 알고리즘이 제안되어 있다.

    ① 한 개의 요구사항을 선정한다. 구조요소의 심각도 수준을 그 요구사항의 것과 동일하게 배정한다. 만약 그 요구사항이, 예를 들어 중간 정도의 심각도를 갖는다면, 그러면 초기 요소수준이 또한 "중간"이 된다.

    ② 각 추가적인 요구사항에 대해 만약 모든 확인된 요구사항들이 동시에 만족하지 못하였다면 결말의 심각도를 추정해서 구조요소의 심각도 추정치를 누적한다.

③ 그 구조요소에 의해 영향을 받은 모든 요구사항들이 고려될 때까지 계속한다. 최종 구조요소의 리스크 수준은 구조요소의 설계 고장확률과 그 고장에 따른 누적된 심각도를 곱한다.

3. 분석결과를 문서화한다.

표 IV.3.2 구조 리스크 수준 결정을 위한 매트릭스

구조요소의 리스크 수준	요구사항 추가에 따른 리스크 수준		
	높음	중간	낮음
매우 높음	매우 높음	매우 높음	매우 높음
높음	매우 높음	높음	높음
중간	높음	중간	중간
낮음	높음	중간	낮음

분석과정에서 수집된 정보는 SW 요구사항 위험 분석의 정보에 추가될 수 있다. 특히, 만약 여러 구조요소들이 매우 높은 리스크를 갖는 것으로 분류된다면, 그 구조를 다시 설계해서 SW 구조로 인한 리스크를 낮추거나, 또는 전체시스템의 리스크를 낮추기 위해 보상 수단을 고려하여야 한다. 요구사항 위험 분석을 실시해서 그에 따른 추가적인 개발, 확인 및 시험을 위한 자원의 배정은 구조 위험 분석 결과를 기반으로 정한다.

구조 위험 분석에서는 어떤 구조요소들이 위험하지 않다는 것을 입증할 수 있다. 그 분석 에서는 구조요소에서 생긴 고장이 시스템 위험을 초래하지 않다는 것을 검증한다. 그와 같은 요소들은 설계 및 구현 위험 분석 과정에서 크게 주의할 필요가 없다.

만약 고장수목분석(FTA) 또는 사건수목분석(ETA)이 요구사항 위험 분석 과정에서 사용 되었다면 그것은 SW 와 HW 구조까지 확대될 수 있다. 그 수목들의 값은 대개가 수목의 구조에 있는 정보에서 나온다.

(3) SW 구조 위험 분석의 결과

구조 위험 분석의 결과는 다음과 같다.

- SW 구조 설계요소들과 그 배정된 리스크 수준의 목록
- 어떤 정해진 구조가 사용될 때 그 SW 의 위험원에 미치는 영향 분석
- 선정된 구조와 관련한 위험원들을 완화하는데 필요한 설계제약사항과 코딩제약사항 목록
- SW 콤포넌트들의 위험원 필수싱 수준을 줄일 수 있는 설계변경사항 권고
- 상세 설계 확인 및 검증, 코드 확인 및 검증, 그리고 최종 시스템에 대한 검증분석 및 시험수행 기간에 수행되어야 할 분석 및 시험 수행에 대한 권고

[SD-T-03] SW 모듈 위험 분석 기법

가. 개요

상세설계 문서는 SW 요구사항, SW 구조, 그리고 상세설계 내용에 대하여 두 가지 추적 방식을 갖고 있어야 한다. 첫째는 각 요구사항이 그 요구사항을 구현하는 구조와 상세한 설계요소까지 추적되는 방식이다. 둘째는 각 상세한 설계요소가 그것을 구현하는 구조와 요구사항에 이르기까지 역추적하는 방식이다. 만약 이와 같은 추적들이 안되면, 위험 분석이 시작되기 이전에 이루어져야 한다.

나. 방법

(1) SW 상세설계 위험 분석의 입력

다음과 같은 정보가 상세설계 위험 분석을 수행하는데 필요하다.

- 예비위험원목록 (PHL)
- 예비위험원분석 (PHA)
- 안전성분석보고서 (SAR)

- SW 요건명세서 (SRS)
- SW 구조설명서 (SAD)
- SW 상세설계 설명서
- SW 요구사항 및 구조 위험 분석
- 요구사항에서 구조와 상세설계에 이르기까지 추적매트릭스

(2) 분석 절차

SW 상세설계 위험 분석 다음과 같은 절차로 수행한다.

- 각 SW 구조요소에 대해서 구조요소와 함께 상세설계 요소들의 목록을 작성한다. 어떤 설계요소들은 한 개 이상의 구조요소들이 사용될 수 있다. 예를 들면, 하위 통신 SW 는 그 구조의 거의 모든 요소들에서 사용될 수 있다.
- 각 설계요소에 대해서 구조요소와 관련된 위험원이 변경하였는지를 결정하기 위해 표 III.3.6 에 'D'로 표기된 지침 문구를 사용한다. 이것은 만약 설계요소, 설계 규칙, 설계 도구, 또는 설계기법들이 두 개 이상의 구조요소들에서 공통모드 고장메커니즘을 갖는다면 생길 수 있다. 만약 그렇게 되면 앞의 위험 분석들은 다시 수행되어야 할 필요가 있다.
- 분석 결과를 문서화한다.

만약 모든 설계요소들을 분석할 만한 자원들이 없다면 매우 높은 또는 높은 리스크를 갖는 구조요소들로 이루어진 부품 및 관련 부품을 선택한다. 통신 모듈, 장치 구동기 또는 파일 관리자 등은 일반적으로 높은 리스크를 갖는 구조요소에 속한다.

상세단계의 분석이 필요한 경우 시스템의 위험도에 영향을 끼치지 않도록 주의를 기울여야한다. 다시 말하면, 요구사항 및 구조설정 단계에서 위험원들을 확인, 관리 그리고 완화하는 데에 있어서 이슈사항이 발생했다면, 상세설계 단계에서 분석해야 할 대상이 있다는 의미가 될 수 있고, 그 부분에 대해서는 시스템 위험 요소 관점에서 전반적으로 수행의 정확성에 좀 더 중요성하게 다루어져야한다는 것을 의미한다.

분석과정에서 수집된 정보는 상세 설계에 의해 새로운 위험원이 발생하지 않았다는 것을 보증하는데 쓰일 수 있다. 그것은 또한 코딩과 시험을 위한 자원의 배정에도 도움이 될 수 있다.

(3) SW 상세설계 위험 분석의 결과

SW 상세설계 위험 분석의 결과는 문서화된 분석결과이다.

[SC-T-01] 구현 안전 평가 기법

가. 개요

SW 문서는 상세한 설계요소와 그 설계요소를 구현하는 코드간에 두 가지 방식의 추적이 가능해야 한다. 만약 이러한 추적이 안 된다면, 코드 위험 분석이 시작되기 전에 이루어져 한다. 앞의 3 가지 분석들이 잘 수행되었다면 위험 분석보다는 이 시점에서는 정확성(correctness)이 현안이 된다. 주요 역점사항은 코드 변경 이전에 수행된 분석결과를 변경하지 않고, 새로운 위험원을 만들지 않았는지를 확증하는 것이다. 앞의 분석 결과들이 가장 필수적인 코드 요소들에 대한 직접적인 확인과 시험에 사용될 수 있다.

나. 방법

(1) SW 코드 위험 분석의 입력

다음과 같은 정보가 코드 위험 분석을 수행하는데 필요하다.

- 예비위험원목록 (PHL)
- 예비위험원분석 (PHA)
- 안전성분석보고서 (SAR)
- SW 요건명세서 (SRS)
- SW 구조설명서 (SAD)
- SW 상세설계 설명서
- 코드
- SW 요구사항. 구조 및 설계 위험 분석
- 요구사항에서 구조, 설계 및 코드까지 추적매트릭스

(2) 분석 절차

다음과 같은 단계들이 코드 위험 분석을 수행하는데 사용될 수 있다.

- 1. 각 코드요소에 대해서 설계 위험 분석의 결과가 수정되어야 할 필요가 있는지, 또는 새로운 위험원이 생겨났는지를 결정하기 위해 표 III.3.6 에서 'C'로 표기된 지침 문구를 사용한다. 만약 그렇다면 앞에서 수행된 분석의 일부 또는 전부가 다시 수행되어야 할 필요가 있다.
- 2. 모든 모듈들에서 공통모드고장을 초래할 가능성이 있는 도구, 컴퓨터 언어, 그리고 코딩 기법들을 조사한다, 위험한 도구 특성 또는 코딩 기법들을 회피하는 코딩규칙 또는 도구-활용규칙들을 명시한다. 만약 기존의 운영체제(OS)가 사용된다면, 회피해야 할 위험한 특성 또는 기능들을 명시한다.
- 3. 분석결과를 문서화한다.

(3) SW 코드 위험 분석의 결과

SW 코드 위험 분석의 결과는 문서화된 분석결과이다.

[SI-T-01] SW 단위시험 기법

가. 개요

단위 시험을 하기 위해서는 시험환경이 엄격한 형상관리 통제 하에 있어야 하며 기준선을 근거로 모든 통제가 이루어져야 한다.
시험 전 단위시험 환경의 시험장비가 정상적으로 작동되는지를 확인하여야 하는데 이러한 것들을 확인하기 위한 내용을 기술한다. 시험을 지원하는 도구와 같은 시험환경을 준비한다. 시험장비는 시험 전 사전 점검이 되어 있어야 한다. 수정이 안된 부분 및 시험장비의 사양을 기록해 둔다. 모든 시험에 관한 상세정보 (설정 , 시험장비 검정 및 교정 기록 , 피시험기기 구성품 목록, 형상정보 등)를 기록한다.

나. 방법

시험환경 구축 절차

단위 시험에 앞서 시험 시 사용될 장비들에 대한 준비 차원에서 시험 환경을 점검한다. 호스트, 타겟, 에뮬레이터 , 연결케이블, 측정 장비 등을 종합 점검한다.
단위 시험을 위한 준비로써 운영체제 , 컴파일러 , 시험결과측정 S/W(시험도구소프트웨어) , 런타임 라이브러리, 시험데이터, 시험 스크립트 등을 시험에 앞서 준비한다.

SW 단위시험 절차서 작성

각각의 시험 대상 단위에 대하여 같은 형태의 절차끼리 그룹핑하여 절차를 작성한다.
각각의 단위 별로 시험사례를 표 IV.3.3 의 양식으로 작성한다.

표 IV.3.3 단위시험 시험양식

번호	시험항목	입력값	입력 변수	예상결과값	결과값	Pass/Fail	시험횟수

- 시험항목 : 단위의 기능, 입출력파라메터 및 로직을 반영
- 입력값 : 단위 및 단위간 고려
- 예상결과값 : 시험데이터에 따른 예상결과 기입
- 결과값 : 오류검출 또는 오류추측을 위한 데이터 생성 고려

절차의 승인

이 절에서는 절차서 자체에 대한 공식적인 검토와 그 절차를 승인해야할 개인의 목록을 정하고, 그 승인을 받아야 한다

[SI-T-02] SW 통합시험 기법

가. 개요

SW 통합 시험을 하기 위해서는 시험환경이 엄격한 형상관리 통제하에 있어야 하며 기준선을 근거로 모든 통제가 이루어져야 한다. 모든 통합 시험은 데이터베이스, 리파지토리에서 가져오고 결과도 리파지토리에 저장하여 일관성을 보장하고 접근통제가 이루어져야 한다.

SW 통합 시험 환경의 시험장비가 정상적으로 작동되는지를 시험 전에 반드시 확인해야 하고 미흡한 부분에 대해 기록해서 관리한다. 모든 시험에 관한 상세정보(설정, 시험장비 검정 및 교정 기록, 피시험기기 구성품 목록 및 형상정보 등)를 기록한다.

나. 방법

SW 통합시험 절차

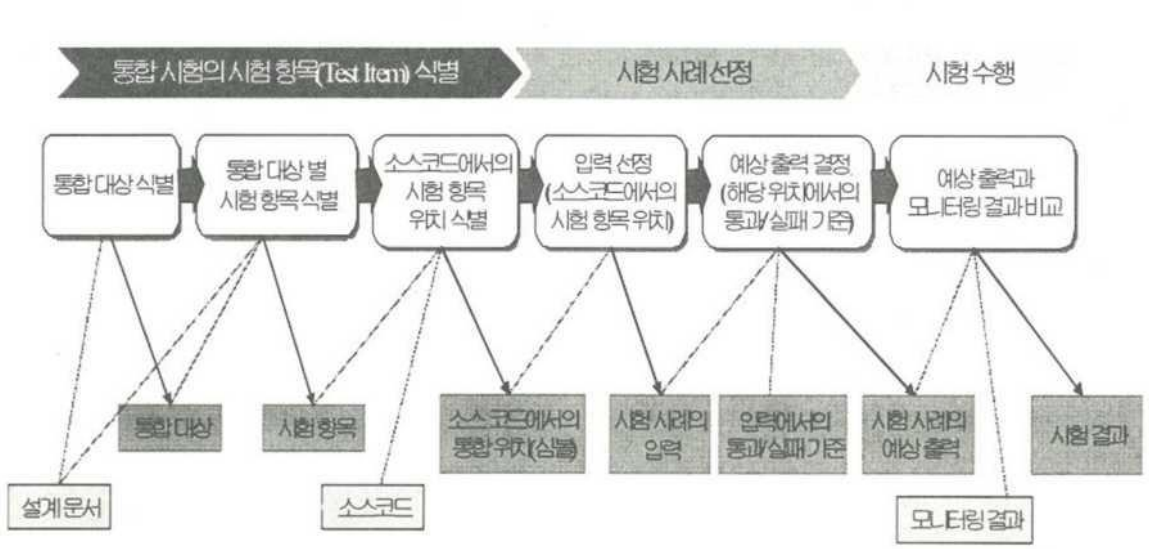
SW 와 SW 의 통합 접근은 그림 IV.3.2 와 같이, 통합시험 시험항목 식별, 시험사례 선정, 시험 수행 별 관련된 시험 작업들을 포함한다.

시험 항목 식별 단계에서는 설계 명세서를 토대로 통합 시험의 대상이 되는 통합 대상을 같이 식별하고, 각 통합 대상별로 수행되어야 하는 통합 시험항목을 식별한 후, 식별한 시험 항목이 소스코드에 매핑되는 시험 항목의 위치를 선정한다.

입력과 예상출력으로 이루어진 시험사례 선정 단계에서는 소스코드에서 식별한 시험항목의 위치를 입력으로 정하고, 이 입력에 통과 및 실패 기준을 적용하여 예상 출력을 결정한다.

시험수행 단계에서는 각 시험 사례를 수행하면서 모니터링된 결과와 예상 출력을 비교하면서 시험을 수행한다.

그림 IV.3.1 SW 통합시험 절차



시험환경 구축 절차

SW 통합시험을 위해서는 주요 통합요소를 설치, 장착 또는 통합 범위를 설정한다. 통합 환경에 필요한 것들을 준비하고 이를 측정할 측정장비를 연결한다.

컴파일된 모듈 코드들은 통합된 코드로 병합하여 새로운 이름을 부여한 다음 재컴파일한 후 심볼을 다운로드를 한다.

SW 통합시험 절차서 작성

다음과 같은 양식으로 통합 유형별로 그룹핑하여 절차서를 작성한다. 이때 절차서 가 본문에 들어가기에 볼륨이 클 경우 부록으로 작성할 수도 있다.

통합시험 대상별 시험사례를 기술한다. 이때, 표 IV.3.4 와 같이 통합시험사례 ID, 통합시험 항목 ID, 입력, 예상출력을 기술한다.

표 IV.3.4 통합시험 양식

통합시험 사례 ID	통합시험항목 ID	입력	예상출력

문서화 및 시험 보고서 생성

시험이 시험절차대로 종료되면 시험 결과값을 정리하여 시험결과 보고서에 포함되도록 해야 한다. 여기에는 시험 과정의 모든 이력정보를 포함하여 시험 실패, 시험시 사고, 시험결과분석, 재시험 등 상세정보를 기록한다.

절차의 승인

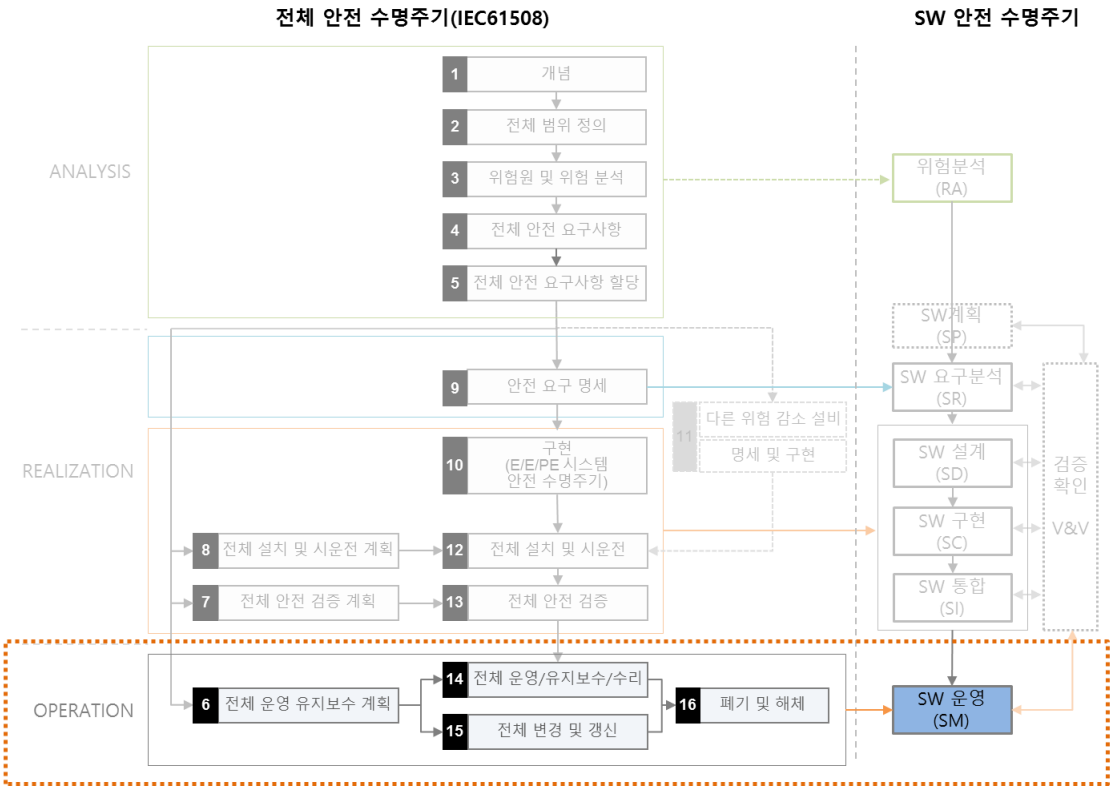
이 장에서는 절차서 자체에 대한 공식적인 검토와 절차를 승인해야 할 개인의 목록을 기술하고, 그 승인을 받아야 한다

V. SW 운영(Operation)

제 1 장. 개 요

SW 운영 단계의 목적은 SW 운영, 유지보수 및 수리, 변경 및 갱신 처리, 폐기 및 해체 처리를 원활하기 수행하기 위한 것이다.

그림 V.1.1 SW 운영 수명주기



제 2 장. 주요 활동

1. 개발 활동

SW 운영, 유지보수 및 수리

SW 에 요청된 요구사항을 처리하고, 안전관련 시스템의 기능안전성이 지정된 수준으로 유지되는지를 확인한다. 안전관련 시스템의 전체 운영, 유지보수 및 수리에 필요한 기술적 요구사항이 규정되고, 안전관련 시스템의 미래 운영, 유지보수 및 수리에 대한 책임이 있는 사람들에게 제공되는지를 확인한다.

변경 및 갱신 처리

SW 변경 및 갱신이 이루어진 후에 안전 관련 시스템에 대한 기능안전성이 적절한지 확인한다.

폐기 및 해체 처리

SW 의 폐기 또는 해체 활동 중과 후에 안전관련 시스템에 대한 기능안전성이 상황에 적절한지 확인하기 위해 필요한 절차를 수행한다.

2. 확인 및 검증 활동

SW 운영 평가

원시코드, 추적 가능성, 인터페이스 분석을 통해 SW 구현물이 설계내용을 충분히 반영하고 있음을 평가한다.

확인 검증 보고서 작성

각 검증 활동의 종료 후에 작성하는 SW 확인 검증 보고서는 SW 검증 합격 유무 또는 불합격의 원인에 대하여 서술해야 한다. 확인 검증 보고서는 다음을 포함한다.

- SW 요청서, 변경 신청서, 폐기 신청서에 맞는지 여부 확인
- SW 유지보수 계획과 부합하지 않은 항목
- 검출된 오류 또는 부족한 부분
- 검증된 항목의 식별 및 형상

3. 안전 활동

SW 운영에서의 안전과 관련된 부분이 요청사항을 정확히 접수 및 처리하고 있으며 새로운 위험을 초래하지 않음을 검증한다.

SW 구현 안전 평가 수행

기존 코드와 연관된 자료가 고객의한 요청사항을 정확하게 구현했는지, 어떠한 장애도 발생 시키지 않았는지 검증한다. 장애 분석을 갱신한다.

이전의 작업 보고서를 이용하여 위험 분석을 검토하고 갱신한다. 위험을 제거, 감소, 약화 하기 위한 권장사항을 제시한다.

SW 안전 기록 작성

SW 안전 기록은 다음 내용을 포함한다.

- 안전 분석 결과
- 의심되거나 확인된 안전 문제점
- 안전 테스트 결과

IEC 61508-3 표준에 맞는 기능 안전 검증 기법(Technique/Measures)

IEC61508-3 B.5 - Modelling

Technique/Measure		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	데이터 흐름도		R	R	R	R
2a	유한 상태 기계		---	R	HR	HR
2b	정형 기법		---	R	R	HR
2c	시간 페트리 그물		---	R	HR	HR
3	성능 모델링		R	HR	HR	HR
4	프로토타이핑 / 애니메이션		R	R	R	R
5	구조 다이어그램		R	R	R	HR

- 특정한 기술이 고려 사항에 열거되지 않은 경우. 그것은 이 표준을 따라야 한다. 표는 제외된다고 가정해서는 안된다.
- 확률적 정량화는 요구되지 않는다.
- 안전 무결성 수준에 따라 적절한 기술/조치를 선택한다. 대체 기술 또는 이에 상응하는 기술/조치는 번호 뒤에 오는 문자로 표시된다. 대체 또는 동등한 기술/조치 중 하나만 만족하면 된다.

IEC61508-3 B.5.1 : 데이터 흐름도 (Data flow diagrams)

1. 목표

   다이어그램 형식의 프로그램을 통한 데이터 흐름을 설명한다.

2. 설명

데이터 흐름도는 데이터 입력이 출력으로 변환되는 방식을 설명하며 다이어그램의 각 단계는 고유 변경을 나타낸다. 데이터 흐름도에는 세 가지 측면이 있습니다.

- 주석 화살표: 데이터가 무엇인지를 설명하는 주석이있고 데이터 흐름을 나타낸다.
- 주석이 버블: 변경을 문서화하는 주석을 나타낸다.
- 연산자(and, xor): 연산자는 주석 화살표를 연결에 사용된다.

데이터 흐름도의 각 버블은 독립 실행 형 블랙 박스이며 입력이 가능 해지면 해당 출력을 출력으로 변환한다.

주요 이점은 변환이 어떻게 구현되는지에 대한 가정없이 표시한다는 것이다. 순수한 데이터 흐름도에는 제어 정보 또는 시퀀싱 정보가 포함되어 있지 않지만 Yourdon (C.2.1.4 참조) 에서 처럼 표기법의 실시간 확장을 통해 제공된다. 데이터 흐름도의 준비는 시스템 입력을 고려하여 시스템 출력 방향으로 작업하는 것이 좋다. 다이어그램의 전체 구조를 결정하고 데이터 흐름 다이어그램을 구성하는 규칙은 시스템 설계 측면 중 하나다. 모든 설계와 마찬가지로, 초기 다이어그램을 생성하기 위해 단계적 개선을 반복한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.2 를 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-11

☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.4 SW 아키텍처 설계 > 1.4.2 소프트웨어 아키텍처 명세(안전측면) 예제 > 1.4.2.3 동적 설계

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.5.4 : 프로토 타이핑 / 애니메이션 (Prototyping/animation)

1. 목표

주어진 제약에 대해 시스템을 구현할 타당성을 점검하고 오해를 찾기 위해 시스템에 대해 담당자의 해석을 고객에게 전달한다.

2. 설명

   시스템 기능, 제한 조건 및 성능 요구 사항의 서브 세트가 선택되며 프로토 타입은 고급 도구를 사용하여 작성된다. 이 단계에서는 대상 컴퓨터, 구현 언어, 프로그램 크기, 유지 관리 가능성, 안정성 및 가용성과 같은 제약 조건은 고려사항이 아니다. 프로토 타입은 고객의 기준에 따라 평가되며 시스템 요구 사항은 이 평가에 따라 수정될 수 있다.

3. 비고

   이 측정 기법 및 기준은 IEC 61508-7 의 표 C.5.17 을 참조한다.

4. 분야별 가이드 참조 위치

   ☞ 철도 가이드: 부록 > B-43

   ☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

IEC61508-3 B.5.5 : 구조 다이어그램 (Structure diagrams)

1. 목표

   프로그램의 구조를 도표로 표시한다.

2. 설명

   구조 다이어그램은 데이터 흐름도을 보완하는 방법이다. 프로그래밍 시스템과 파트의 계층 구조를 설명하고 그래픽으로 표시한다. 데이터 흐름도의 요소를 프로그램 단위의 계층 구조로 구현할 수있는 방법을 제공한다. 구조 다이어그램은 프로그램 모듈 간의 관계를 보여 주며 순서에 대한 정보는 제공하지 않는다. 다음 네 가지 기호를 사용하여 작성한다.

   - 모듈의 이름이 주석 된 사각형
   - 직사각형을 연결하는 구조를 만드는 선
   - 구조선 차트의 요소로 전달되는 데이터의 이름이 주석로 표시된 원형 화살표 (원은 비어있으며 안에 있는 화살표는 차트의 사각형을 연결하는 선과 평행)
   - 구조 다이어그램에서 한 모듈에서 다른 모듈로 전달되는 제어 신호의 이름이 표시된 원형(채워진) 화살표는 두 모듈을 연결하는 선과 평행하게 그려진다. 데이터 흐름도에서 여러가지 구조의 차트를 만들 수 있다.

데이터 흐름도는 시스템의 정보와 기능 간의 관계를 나타내고 구조 다이어그램은 시스템의 요소가 구현되는 방식을 나타낸다. 두 기술 모두 시스템의 유효성은 있지만 다른 관점을 제시한다.

3. 비고

이 측정 기법 및 기준은 IEC 61508-7 의 표 C.2.3 을 참조한다.

4. 분야별 가이드 참조 위치

☞ 철도 가이드: 부록 > B-51

☞ 자동차 가이드: PART 3: SW 안전 프로세스와 단계 별 T&M > 1. 소프트웨어 설계 가이드 > 1.4 SW 아키텍처 설계 > 1.4.2 소프트웨어 아키텍처 명세(안전측면) 예제 > 1.4.2.2 정적 아키텍처 설계

☞ 의료 가이드: Part 3 >제 2 장 의료기기 소프트웨어 개발 생명주기 단계별 가이드 > 2.3 단계 2 소프트웨어 요구사항분석, 2.4 단계 3 소프트웨어 아키텍쳐 설계, 2.5 단계 3 소프트웨어 상세설계

제 3 장. 사용 양식

1. SW 유지보수계획서

SW 의 유지보수계획서가 가질 수 있는 목차는 아래와 같다. 제시 한 목차는 유지보수계획서를 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경우 SW 의 특성에 따라 다른 목차를 사용할 수 있다.

가. 목 차

6.4 구 현

6.5 시 험

6.6 인 계

7. 보고 요건

8. 행정 요건

8.1 부적합 사항(Anomaly) 해결 및 보고

8.2 일 탈(Deviation) 정 책

8.3 통제 절차

8.4 표준, 시행령 및 약정

8.5 성능 추적

8.6 유지보수계획서의 품질 관리

9. 문서화 요건

나. 작성법

1. 목 적	유지보수에 대한 목적을 기술한다.
2. 범 위	유지보수 활동에 대한 범위를 지정한다.
3. 참고문서	
3.1 적용 법규	안전성 분석에 사용된 법규를 기술한다.
3.2 기술 표준	안전성 분석에 사용된 기술 표준을 기술한다.
3.3 설계 문서	안전성 분석에 사용된 설계 문서를 기술한다.
4. 용어 정의 및 약어	
4.1 용어정의	안전성 분석에 사용된 용어를 정의한다.
4.2 약 어	안전성 분석에 사용된 약어를 정의한다.
5. 개 요	
5.1 유지보수조직	SW 유지보수 활동을 수행하는 조직을 기술한다. 내부 유지보수 조직과 연계된 외부 조직을 기술하고, 유지보수 활동에 의해 발생된 문제점들을 해결할 수 있는 조직, 그리고 유지보수 제품을 승인할 권한이 있는 조직에 대해서도 기술한다.
5.2 유지보수우선순위	유지보수 활동이 어떻게 작업패키지로 구성될 것인지를 기술하고 작업패키지에 우선순위가 할당되는 공정 및 자원이 우선순위 작업에 어떻게 할당되는지 기술한다.
5.3 유지보수기준선	유지보수 활동을 위한 기준선을 정의하고 기준선부터 출발해서 유지보수 활동을 할 것임을 정의하여야 한다.
5.4 유지보수담당자원	유지보수 활동을 위한 시설, 도구, 재정, 절차요건과 같은 유지보수 자원에 대해 요약하여야 한다
5.5 유지보수책임	유지보수 활동에 대한 조직의 책임과 역할에 대해 기술한다.

5.6 방법론, 도구 및 기법	문서, 유지보수 도구, 기법, 방법, 운영 및 시험 환경들을 기술한다. 각 도구에 대한 획득, 훈련, 지원, 품질 정보를 포함한다.
6. 유지보수공정	
6.1 SW 문제/수정 식별, 분류 및 우선순위	
6.2 분 석	유지보수를 해야 하는 타당성 분석을 수행하는 방법 및 내용에 대해 기술한다. 구현해야 하는 사례 및 난이도, 대략적인 구현 시간을 통한 대략적인 자원 평가를 기술하고 현 사용자와 추후 사용자들에게 미칠 수 있는 영향들도 분석한다. 사용자에게 영향을 최소활 할 수 있는 수정일정도 분석한다. 추가적인 문제점 분석이 요구되는 지를 결정하고 제안된 변경에 대한 수락 및 거 절에 대해 기록한다. 설계 단계동안 고려되어야 하는 것과 변경에 따른 결과에 대 한 SW 와 HW 의 제약사항들을 평가한다. 가능한 현재 사용되는 설계를 다시 사용하는 것이 좋다.
6.3 설 계	설계단계에서 수행하여야 하는 유지보수 활동에 대해 기술한다. 설계공정의 특성은 과제마다 공정이 다양하다. 제품의 특징은 SW 의 모듈이 어떻게 변화될 것인지 결정하는 설계를 마쳤을 경우에 고려할 수 있다.
6.4 구 현	구현단계에서 수행하여야 하는 유지보수 활동에 대해 기술한다. 구현단계에서의 입 력은 설계단계의 모든 결과물이고 다른 입력물도 요구될 수 있다. 승인되거나 통제 된 요구사항과 설계 문서, 유지보수 일원에 의해 사용되는 승인된 구현 표준, 구현 단계에서 적용될 수 있는 설계 측정값, 세부적인 구현일정, 얼마나 코드 검토를 수행할 것인지에 대한 내용을 포함할 수 있다.
6.5 시 험	시스템 시험과 인수 시험에 대한 유지보수 활동을 기술한다.
6.6 인 계	SW 에 대한 인계를 위한 절차 및 역할들을 기술한다. 형상관리와의 원만한 소통을 통해 인계활동이 정확히 이루어지도록 정의한다.
7. 보고 요건	정보가 어떻게 수집되고 제공되는 지를 기술한다. 완성된 작업, 진행중인 작업, 수정된 작업, 적체된 작업들을 기술한다.
8. 행정 요건	
8.1 부적합 사항(Anomaly) 해결 및 보고	부적합 사항 보고 기준, 분배 리스트, 해결 권한들을 포함하는 방법들을 기술한다.
8.2 일탈(Deviation) 정책	계획에서 배제되는 절차와 형식들을 기술한다.
8.3 통제 절차	유지보수 수행동안 적용된 통제 절차를 기술한다. 이러한 절차는 SW 제품과 유지보수 결과가 어떻게 형상화되고 보호되고 저장되는지를 포함한다.
8.4 표준, 시행령 및 약정	유지보수 활동에 대한 표준, 시행령, 약정사항들을 기술한다. 약정사항들은 내부표준, 내부 시행령, 내부 정책들을 포함한다.
8.5 성능 추적	모든 SW 유지보수 단계의 성능 추적을 위한 절차를 기술한다.
8.6 유지보수계획서의 품질 관리	계획의 정확성을 보장하기 위해 유지보수계획서가 어떻게 검토되고, 향상되고, 승인되는지 기술한다.
9. 문서화 요건	유지보수 활동을 통해 생산되는 결과물을 기록하는 절차를 기술한다.

2. SW 운영 안전성분석보고서

SW 에 운영에 대한 안전성 분석 보고서의 작성과 관련하여 안전성분석 보고서에 포함되어야 할 내용들에 대한 최소요건들을 제시하였고, SW 의 운영 안전성분석 보고서에 포함되어야 할 기본 내용을 기술하고 있으며 자세한 내용은 다음과 같다.

가. 목 차

나. 작성법

1. 목적	안전성분석의 수행 목적을 정의한다.
2. 범위	안전성분석의 수행 범위를 정의한다.
3. 용어 정의 및 약어	안전성분석 보고서에 사용된 어떤 용어 및 약어를 정의하고 설명한다.
4. 참고 문서	안전성분석을 위해서 사용된 근거법령, 근거기준, 설계문서, 기타문서들을 기술한다.
5. 시스템 개요	대상 운영 시스템의 상위수준 시스템 구조를 기술한다. 시스템, 하위시스템, 그리고 연계관계 등의 기능을 기술한다. 고객의 요청 및 문제 발생에 의해 변경해야 할 대상 코드의 내용을 기술한다.
6. 안전성	SW 운영 안전성분석에 사용된 입력자료를 기술한다.

분석기법	SW 안전성분석에 사용된 기법을 기술한다. SW 안전성분석에 사용된 절차를 기술한다. SW 안전성분석에 사용된 도구를 기술한다.
7. 안전성 분석결과	SW 운영 시험 적합성분석의 결과를 기술한다. SW 운영 시험 위험원분석의 결과를 기술한다.
8. 안전성 검토	요청된 고객의 운영 중 요구사항에 대해 안전성분석의 결과를 시스템 개발조직, 시스템 안전성분석조직 및 소프트웨어 설계조직과의 공식적인 회의를 통해서 SW 코드에 포함된 위험원 의 수정 및 보완의 결과를 기술한다.
9. 승인	안전성보고서 자체에 대한 공식적인 검토와 승인해야 할 개인의 목록을 규정해야 하고 서명을 받는다.

VI. 부 록

Appendix 1. SW 안전 수명주기 산출물 양식

[SP-D-01] SW 개발계획서

SW 의 개발계획서가 가질 수 있는 목차를 참고하여 아래와 같이 구성할 수 있고, 목차는 개발계획을 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경우 SW 의 특성에 따라 적절히 변경하여 사용할 수 있다.

가. 목 차

6. 기술 프로세스 계획

6.1 프로세스 모형

6.2 방법, 도구와 기법

6.3 기반 구조 계획

6.4 제품 수락 계획

7. 지원 프로세스 계획

7.1 형상 관리 계획

7.2 검증과 확인 계획

7.3 문서화 계획

7.4 품질 보증 계획

7.5 검토와 감사 계획

7.6 문제 해결 계획

7.7 하청업자 관리 계획

7.8 프로세스 개선 계획

8. 추가 계획

나. 작성법

1. 개요	목적, 범위. 프로젝트 목표, 프로젝트 가정사항 및 제약사항, 프로젝트 인도물 목록, 프로젝트 일정과 예산의 요약, 그리고 변경 계획을 제공한다.
1.1 프로젝트 요약	1.1.1 목적, 범위와 목표 프로젝트의 목적, 범위, 목표와 인도될 제품이 정의 된다. 또한 결과 산출 물이나 프로젝트로부터 배제될 목표나 범위에 대한 고려사항이 설명되도록 한다. 범위 설 정 문장은 프로젝트 협정서와 다른 관련 시스템 수준 또는 사업 수준 문서에서의 유사한 문장과 일치시킨다. 1.1.2 가정 사항과 제약사항 프로젝트가 전제로 하는 가정 사항을 기술하여야 하고, 일정. 예산, 자원,

	재사용되는 SW, 타 SW 와 결합되는 SW, 다른 제품과의 인터페이스와 같은 프로젝트 요소 상에 부과된 제한 사항을 기술한다. 1.1.3 프로젝트 인도물 획득자에게 전달된 작업 산출물, 인도 날짜. 인도 장소, 프로젝트 협약서를 만족하는 요구된 양을 목록화한다. 1.1.4 일정과 예산 요약 SW 프로젝트 일정과 예산에 대한 요약을 제공한다.
1.2 계획 변경	계획에 대한 예정된 또한 예정되지 않은 수정본을 생성하기 위한 계획을 명시한다.
2. 참고문헌	모든 문서와 계획서 내에 참조된 다른 정보 자료들의 완전한 목록이 제공한다.
3. 정의	본 계획서를 정확하게 이해하기 위해 요구되는 머리글자 어와 모든 용어에 대한 정의를 포함하는 문서, 참고 문헌을 정의하거나 제공한다.
4. 프로젝트 조직	프로젝트 외부 인터페이스를 식별하고, 프로젝트 내부 조직 구조를 서술하고, 프로젝트를 위한 역함과 책임을 정의한다.
4.1 외부 인터페이스	프로젝트와 외부 실체간의 조직적 경계를 서술한다.
4.2 내부 구조	SW 개발 팀 단위간의 인터페이스를 포함한 프로젝트 조직의 내부 구조를 기술한다.
4.3 역할과 책임	각 주요 작업 활등과 지원 프로세스의 성질을 식별하여 서술하고 해당 프로세스와 활동을 위해 책임을 갖는 조직단위를 정의한다.
5. 관리 프로세스 계획	프로젝트를 위한 프로젝트 관리 프로세스를 명시한다.
5.1 초기 계획	평가 계획, 팀원 계획, 자원 획득 계획, 훈련 계획을 명시한다. 5.1.1 산정 계획 프로젝트 비용, 일정, 자원 요구사항, 관련된 신뢰 수준을 산정하기 위해 사용되는 방법, 도구 및 기법뿐 아니라 프로젝트 수행을 위한 비용과 일정을 명시한다. 5.1.2 팀원 계획 해당 기술 수준에 맞는 직원의수, 다수의 인원과 기술 유형이 필요한 프로젝트 단계, 필요한 기간을 명시하여야 한다. 추가로 직원 인원의 출처를 명시한다. 5.1.3 자원 획득 계획 성공적으로 프로젝트를 완수하기 위해서 필요한 인력에 추가로 자원 획득 계획을 명시한다. 5.1.4 프로젝트 팀원 훈련계획 SW 프로젝트를 성공적으로 수행하는데 이용 가능한 필수적 기술 수준 확보를 위해 필요한 훈련을 명시한다.
5.2 작업계획	SW 프로젝트를 위한 일정, 자원, 예산 세부항목을 명시한다.

	5.2.1 작업 활동 SW 프로젝트에서 수행되는 다양한 작업활동을 명시한다. 5.2.2 일정 할당 시간 순차적 제약을 표시하고, 병행작업 활동기회를 예시하는 방법으로 작업 활동간 일정관계를 제공한다. 5.2.3 자원 할당 프로젝트 작업 분할 구조 내에 각 주요 작업 활동을 위해 할당되는 자원의 세부항목을 제공한다. 5.2.4 예산 할당 작업 분할 구조내의 각각의 주요 작업 활동에 대해 필요한 자원 예산의 상세한 세부항목을 제공한다
5.3 통제 계획	제품 요구사항, 프로젝트 일정, 예산, 자원, 작업 프로세스와 작업 산출물의 품질을 측정, 보고 및 통제에 필요한 매트릭, 보고체계, 제어절차를 명시한다. 5.3.1 요구 통제 계획 제품 요구사항에 대한 변경을 측정, 보고, 통제하기 위한 체계를 명시한다. 5.3.2 일정 통제 계획 프로젝트 이정표에서 완료된 작업 진행 정도를 측정하고, 계획된 진도와 실제 진도를 비교하고, 계획된 진도대로 수행되지 않았을 때 수정 활동에 대한 체계를 명시한다. 5.3.3 예산 통제 계획 완료된 작업의 비용을 측정하고, 계획된 비용과 비교하고, 실제 비용이 예산과 일치하지 않을 때 수정 활동에 대한 체계를 명시한다. 5.3.4 품질 통제 계획 작업 프로세스와 작업 산출물의 품질을 측정하고 통제하는 체계를 명시한다. 5.3.5 보고 계획 보고 체계와 보고서 양식을 명시한다. 5.3.6 매트릭 수집 계획 프로젝트 매트릭을 수집하고 보유하는데 사용되는 방법, 도구 및 기법을 명시한다.
5.4 위험 관리 계획	프로젝트 위험 요소의 식별, 분석, 순위 결정을 위한 위험 관리 계획을 명시한다.
5.5 종료 계획	SW 프로젝트의 순차적인 종료 보장에 필요한 계획을 포함한다.
6. 기술 프로세스 계획	개발 프로세스 모형, 기술적 방법, 도구와 여러 작업 산출물 개발에 사용되는 기법들 프로젝트 기반구조를 수립하고 유지 보수하는 계획, 제품 수락 계획 등을 명시한다.
6.1 프로세스 모형	주요 작업 활동들 간의 관계, 정보 흐름을 명시하는 지원 프로세스, 활동과 기능들간의 작업 산출물, 작업 산출물이 생성되는 시기, 검토와

	성취되는 주요 이정표, 설정되는 기준선, 완성되는 프로젝트 인도물, 프로젝트 기간 동안에 필요한 승인사항을 정의한다.
6.2 방법, 도구와 기법	개발 방법론, 프로그래밍 언어와 다른 표기법, 명세와 설계와 구현과 시험과 통합과 문서화와 인도와 수정과 유지 보수를 위해 사용되는 도구와 기법들을 명시한다.
6.3 기반 구조 계획	SW 프로젝트를 수행하는데 필요한 정책. 절차, 표준, 설비, 개발 환경(HW. 운영체제, 네트워크와 SW)을 설정하고 유지 보수하는 계획을 명시한다.
6.4 제품 수락 계획	SW 프로젝트에서 생성된 인도 가능한 작업 산출물에 대한 획득자 수락을 위한 계획을 명시한다.
7. 지원 프로세스 계획	SW 프로젝트 기간 동안의 지원 프로세스를 위한 계획을 포함한다.
7.1 형상관리 계획	형상 식별, 통제, 상태 설명, 평가, 배포 관리를 제공하기 위하여 사용될 수 있는 방법 포함하여 SW 프로젝트를 위한 형상 관리 계획을 포함한다.
7.2 검증과 확인 계획	검증과 확인 작업 활동을 위한 범위, 도구, 기법, 책임을 포함하는 SW 프로젝트를 위한 검증과 확인 계획을 포함한다.
7.3 문서화 계획	인도할 작업 산출물과 인도하지 않을 작업 산출물을 위한 계획을 포함하는 SW 프로젝트를 위한 문서화 계획을 포함한다.
7.4 품질보증 계획	SW 프로젝트가 요구 명세서, 본 계획서, 지원 계획, 표준, 절차, 지침 에 명시되어있는 SW 프로세스와 제품을 위한 사항을 완료하였는가를 보장하기 위한 계획을 제공한다.
7.5 검토와 감사 계획	프로젝트 검토와 감사를 수행하는데 사용되는 일정, 자원, 방법, 절차가 명시한다.
7.6 문제해결 계획	프로젝트 수행 시 생성된 SW 문제 보고서를 알리고. 분석하고, 우선 순위를 결정하고, 처리하는데 사용되는 절차, 자원, 방법, 도구, 기법을 명시한다.
7.7 하청업자 관리 계획	해당 SW 프로젝트를 위한 작업 산출물을 생성할 수 있는 하청업자를 선정하고 관리하기 위한 계획을 포함한다.
7.8 프로세스 개선 계획	주기적인 프로젝트 심사, 개선 분야 결정, 개선 계획 수행을 위한 계획을 명시한다.
8. 추가 계획	프로젝트 요구사항과 계약 항목을 위한 추가 계획을 작성한다.

[SP-D-02] SW 안전계획서

SW 에 대한 안전계획서의 작성과 관련하여 SW 안전계획서에 포함되어야 할 내용들에 대한 최소요건들을 규정하고 있다. SW 안전계획서를 작성함에 있어서 참고문서로서 IEEE Std. 1228 이 참고모델이 될 수 있고 만약 SW 개발자가 원한다면 SW 안전계획을 시스템 안전계획 안에 포함할 수 있다.

가. 목 차

4.1 SW 안전 분석 준비

4.2 SW 안전 요구사항 분석

4.3 SW 안전 설계 분석

4.4 SW 안전 코드 분석

4.5 SW 안전 테스트 분석

4.6 SW 안전 변경 분석

5. 개발 후 과정

5.1 교육훈련

5.2 적용

5.3 모니터링

5.4 유지 보수

5.5 폐기 및 공지

6. 계획 승인

나. 작성법

1. 목적	본 계획을 준수함으로써 기대되는 안전 목표를 포함하여 안전 계획의 목적 및 범위를 기술한다. SW 프로젝트에 특정한 위험 및 안전 목적을 명시한다.
2. 용어 정의, 약어 및 참조문헌	계획서에서 사용되는 용어, 두문자, 약어 및 참조문헌을 명시한다.
3. SW 안전 관리	안전 관련 SW 를 개발할 때 사용되는 조직, 일정, 자원, 책임, 도구, 기법 및 방법론을 명세한다.
3.1 조직 및 책임	전체조직에서 SW 안전 활동을 기술하고, 안전 이슈와 관련된 기조직 관계 및 기능 관계, 의사소통 체계. 승인체계를 설명한다. SW 안전프로그램을 책임지는 한 사람을 식별하고, 이 사람은 조직에서 자율성을 갖고 있어야 한다. SW 안전프로그램 관리는 다음을 책임진다. • 계획 준비 • 계획을 효과적으로 구현하기 위한 자원을 획득하고 할당함

	• 안전 활동 계획을 다른 조직 구성요소, 기능(개발, 시스템 안전, SW 품질보증, SW 신뢰성. 형상관리, 확인 검증, SW 테스팅) 등과 조정함 • 전체 시스템 안전 프로그램 내에서 SW 안전활동을 조정함 • SW 안전과 관련된 기술적 이슈를 조정함 • SW 안전 활동 실행에 대한 적절한 기록을 유지함 • SW 안전 계획 실행 감사에 참여 • 안전 담당자에 대한 방법, 도구, 기법을 교육함
3.2 자원	안정 활동에 요구되는 자원 요구사항과 그 할당에 대한 내용을 명시한다.
3.3 직원 자격요건 및 교육훈련	다음 활동을 수행함 인원에 대한 자격을 명시한다. • 안전 요구사항 정의 • 시스템의 안전-critical 부분을 설계하고 구현함 • SW 안전 분석 업무를 수행함 • 안전 중요 특성을 테스트함 • SW 안전 계획 구현을 심사함 • 프로세스 인증을 수행함
3.4 SW 수명주기	SW 안전 활동 간의 관계를 명시한 수명주기를 설명한다.
3.5 문서화 요구사항	SW 안전과 관련되어 준비되어야 할 문서를 내용을 포함하여 명시한다. 이러한 문서에는 다음이 포함된다. • SW 프로젝트 관리 • SW 형상관리 • SW 품질 보증 • SW 안전 요구사항 • SW 안전 설계 • SW 개발 방법론, 표준, 프랙티스, 메트릭 및 컨벤션 • 시험 문서 • SW 검증 및 확인 • 안전 검증 및 확인 보고 • SW 사용자 문서 • SW 안전 요구사항 분석 결과 • SW 안전 설계 분석 결과 • SW 안전 코드 분석 결과 • SW 안전 테스트 분석 결과 • SW 안전 변경 분석 결과
3.6 SW 안전 프로그램 기록	다음을 포함하여 생성되고 유지되어야 할 SW 안전 프로그램 기록을 명시 한다. • 분석 결과(확인 검증, 요구사항, 설계, 코드, 테스트 및 기타 기술적 문서를 포함) • 사전 릴리즈 되었거나 구축된 시스템에서 의심되거나 확인된 안전 문제에 관한 정보 • SW 안전 프로그램 활등에 대해 수행된 평가 결과

	• 전체 시스템 전체 또는 일부에 대해 수행된 안전 테스트 걸과 • SW 안전 프로그램 인원에게 제공된 교육훈련기록 • 수행된 모든 인증 결과
3.7 SW 형상관리 활동	안전 관련 SW 가 형상관리 계획에 따라 어떻게 관리되는지 명시한다.
3.8 SW 품질보증 활동	핵심 SW 안전 프로그램 활동이 적절히 수행됨을 확인하기 위해 SW 품질 보증의 역할을 명시한다. 이 절에서는 다음 사항을 설명한다. • 어떻게 계획이 준비되고, 승인되고, 변경되고, 이전 문서와 일관성이 있는가? • SW 안전 업무로부터 나온 기술적 권고사항이 어떻게 검토되고, 변경되고, 구현되는가? • 검토 및 평가가 SW 안전 관심사항, 요구사항, 지침, 프로세스 인증을 어떻게 포함하는가? • SW 안전 프로그램 실행이 어떻게 모니터 되는가?
3.9 SW 검증 및 확인 활동	다음을 보장하기 위해 각 수명주기 활동의 결과가 시스템 안전 요구사항 및 시스템 위험 분석과 대응되는 방법을 명시한다. • 모든 시스템 안전 요구사항이 수명주기 단계에 의해 충족되었음 • 수명주기 활동 동안 업무에 의해 추가적인 위험이 도입되지 않았음
3.10 도구 지원 및 승인	CASE 제품, 컴파일러, 에디터, 결함나무 생성기, HW 및 SW 시험환경과 같은 도구를 선정하고, 승인하고, 통제하기 위한 프로세스와 기준을 명시한다.
3.11 사전 개발되었거나 구매한 SW	사전에 개발되었거나 구매한 SW 를 명시하고, 이 SW 가 안전 활동의 요구사항을 충족시키는지에 대해 설명한다.
3.12 협력업체 관리	협력업체에서 만들어진 SW 가 SW 안전 프로그램 요구사항을 충족시킴을 명시한다.
3.13 프로세스 인증	SW 제품이 계획서에 명시된 프로세스에 따라 만들어졌음을 인증하는데 사용된 방법을 명시한다.
4. SW 안전 분석	
4.1 SW 안전 분석 준비	SW 안전을 분석하기 위한 준비사항을 명시한다.
4.2 SW 안전 요구사항 분석	SW 안전 요구사항을 분석하기 위해 사용되는 분석 종류, 분석 결과 제공방법, 공식 검토 및 인스펙션 등을 명시한다.
4.3 SW 안전 설계 분석	SW 안전 설계 내용을 분석하기 위한 방법, 분석 종류, 분석 결과를 문서화하는 방법, 검토 방법 등을 명시한다.
4.4 SW 안전 코드 분석	SW 안전과 관련된 코드를 분석하는 방법, 분석 결과를 문서화하는 방법, 검토 방법 등을 명시한다.
4.5 SW 안전 테스트 분석	SW 안전과 관련된 테스트를 분석하는 방법, 분석 결과를 문서화하는 방법. 검토 방법 등을 명시한다.
4.6 SW 안전 변경 분석	가정, 명세, 요구사항, 설계. 코드, 장비, 시험계획, 환경, 사용자 문서, 교육훈련자료에서 발생한 변경내용을 반영하기 위한 활동을 정의한다.

5. 개 발 후 (post development)	
5.1 교육훈련	안전과 관련된 인원을 교육 훈련시키기 위한 내용을 명시한다.
5.2 적용	5.2.1 설치 개발된 SW 를 설치하기 위한 요구사항을 명시한다. 5.2.2 시작(startup) 및 이전 (transition) 신규 시스템을 시작하거나 구시스템을 이전할 때 요구사항을 명시한다. 5.2.3 운영 지원 운영지원을 위해 제공되어야 하는 모든 문서 또는 매뉴얼을 명시한다.
5.3 모니터링	SW 의 운영을 모니터링 하는 절차를 설명한다.
5.4 유지보수	개발 후 안전 관련 변경을 유지 보수하는 방법을 설명한다.
5.5 폐기 및 공지	운영 SW 에 대한 폐기 절차 및 방법에 대해 설명한다.
6. 계획 승인	계획서에 대해 공식적으로 검토하기 위한 요구사항과 계획을 승인하는 사람들의 목록을 명시한다.

[CM-V-01] SW 확인 검증 계획서

SW 에 대한 확인 및 검증계획서의 작성과 관련하여 SW 확인 및 검증계획서에 포함되어야 할 내용들에 대한 최소요건들을 규정하고 있고 이는 IEEE std. 1012-2004 를 참고문헌으로 하여 구성하였다. SW 의 확인 및 검증계획서의 내용은 아래와 같이 작성할 수 있고, 예시한 목차는 확인 및 검증계획서를 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경우 SW 의 개발공정 및 검증 특성에 적합하게 목차를 재구성 할 수 있다.

가. 목 차

7.4 통제 절차

7.5 표준 및 관행 설명

8. 확인 검증 문서화 요구사항

나. 작성법

1. 목적	SW 확인 검증 의 목적, 목표, 범위를 설명한다.
2. 참고 문헌	준수사항 문서, 참조되는 문서, 보완하거나 실행하기 위한 문서를 확인한다.
3. 정의	예외 사항을 분류하기 위한 기준을 포함하여 사용되는 모든 용어들을 정의하고, 모든 약어와 표기법도 함께 설명한다.
4. 확인 검증 개요	SW 확인 검증 를 수행하기 위한 조직, 스케줄, SW 무결성 수준 스키마, 자원, 책임, 기구, 기술, 방법론을 명시한다.
4.1 조직	확인 검증 를 위한 조직 구조를 설명한다. 확인 검증 프로세스와 개발, 프로젝트 관리, 품질보증, 형상관리와 같은 다른 프로세스와의 관계를 설명한다. 확인 검증 시도, 확인 검증 작업에 의해서 유발되는 문제를 해결할 수 있는 권한, 확인 검증 결과물을 승인할 수 있는 권한 내에서 의사소통 방법에 관하여 명시한다.
4.2 주요 일정	프로젝트의 수명주기와 이정표를 기술한다. 확인 검증 작업의 스케줄과 개발, 조직적이고 지원 적인 프로세스(즉, 품질보증 또는 형상 관리)의 피드백으로서 작업 결과를 요약한다. 확인 검증 작업은 작업 반복 정책에 따라서 재수행되기 위해 스케줄링 되어야 한다.
4.3 SW 무결성 수준	시스템에 적합하게 선택된 SW 무결성 수준을 명시한다. 프로그램에서 지정된 SW 무결성 수준과 다르게 각 구성 요소(예를 들면, 요구사항, 세부 기능, SW 모듈, 하위시스템, 또는 다른 SW 부분)에 지정된 SW 무결성 수준을 문서화한다.
4.4 자원 개요	직원, 시설, 기구, 재정 또는 특정 요구사항(즉, 보안, 접근권한, 문서제어)을 포함한 확인 검증 자원을 요약 정리한다.
4.5 책임	확인 검증 작업을 위한 조직적 구성원과 책임에 관하여 명시한다.
4.6 도구, 기술 및 방법론	문서, HW, 확인 검증 SW 도구, 기술, 방법론과 확인 검증 프로세스에서 사용되는 운영 및 시험 환경에 관하여 명시한다.
5. 확인 검증 프로세스	수행한 확인 검증 활동과 작업을 문서화한다. 모든 SW 수명주기 프로세스의 확인 검증 활동과 작업의 개요를 포함한다. 각 확인 검증 활동에 대하여 다음 8 가지 내용을 포함한다. 가) 확인 검증 작업 확인 검증을 위한 최소의 작업, 작업 기준, 요구되는 입력과 출력을 명시한다. 또한 각 SW 무결성 수준을 위해 수행해야하는 최소의 확인

	검증 작업을 정의한다. 확인 검증 작업들은 하나 이상의 SW 무결성 수준이 적용될 수 있다. 작업을 수행하고 문서화하는 엄격함의 정도는 SW 무결성 수준을 따른다. SW 무결성 수준이 감소함에 따라. 확인 검증 작업에 수반되는 범위, 강도, 엄격함도 감소된다. 나) 방법론과 절차 온라인 접근, 개발 과정의 감시/평기에 대한 조건을 포함하여, 각 작업의 방법론과 절차에 관하여 설명한다. 작업 결과의 평가기준을 명시한다. 다) 입력 각 확인 검증 작업에서 요구되는 입력을 확인한다. 반드시 각 입력의 형식과 출처를 명시한다. 라) 출력 각 확인 검증 작업에서 요구되는 출력을 확인해야 한다. 각 출력의 목적, 형식과 수취인을 명시한다. 확인 검증 관리와 확인 검증 작업으로부터의 출력은 반드시 뒤따르는 프로세스와 활 동의 적절한 입력이 된다. 마) 스케줄 확인 검증 작업의 스케줄을 설명한다. 각 작업을 시작, 종료하고. 각 입력에 대한 인수와 기준, 각 출력의 인도에 관한 명확한 이정표를 작성한다. 바) 자원 확인 검증 작업 수행에 필요한 자원을 정의해야 한다. 종류별로(예, 직원, 기구, 설비, 이동, 훈련) 자원을 명시한다. 사) 위험과 가정 확인 검증 작업에 관련되는 위험원(예, 스케줄, 자원, 기술적 문제)와 가정을 확인한다. 위험원을 제거하고, 축소하며, 완화시키기 위한 권고사항을 제공한다. 아) 역할 및 책임 확인 검증 작업 수행을 위하여 조직체와 구성원에게 책임을 부여한다.
6. 확인 검증 보고 요구사항	확인 검증 보고서는 작업 보고서, 확인 검증 활동 요약 보고서, 예외사항 보고서, 확인 검증 최종 보고 서로 구성된다. 작업 보고서, 확인 검증 활동 요약 보고서, 예외사항 보고서는 각 SW 결과물과 처리에 관해서 SW 개발 처리에 대한 피드백으로 제공된다. 확인 검증 보고서는 전문 연구 보고서와 같은 선택적 보고서도 포함할 수 있다. 확인 검증 보고서 의 형식과 분류는 사용자가 정할 수 있다. 요청된 확인 검증 보고서는 다음과 같이 구성된다. 가) 작업 보고서: 확인 검증 작업은 확인 검증 작업의 결과와 상태를 문서화하며, 기술적인 발표에 적합한 형식이어야 한다. 다음은 작업 보고서의 예이다. 1) 예외사항 평가 2) 기준선 변경 평가

	3) 개념 문서 평가 4) 형상 관리 평가 5) 계약 검증 6) 중요도 분석 7) 새로운 제약조건 평가 8) HW/SW/사용자 요구사항 할당 분석 9) 장애 분석 10) 설치 점검 1J) 설지 형상 강사 12) 인터페이스 분석 13) 이동 평가 14) 운영 절차 평가 15) 의도적 변경 평가 16) 권고사항 17) 검토 결과 18) 위험 분석 19) SW 설계 평가 20) SW 무결성 수준 21) SW 요구사항 평가 22) 원시 코드와 원시 코드 문서 평가 23) 시스템 요구사항 검토 24) 시험 결과 25) 추적 가능성 분석 나) 확인 검증 활동 요약 보고서: 활동 요약 보고서는 획득지원, 계획, 개념, 요구사항, 설계, 구현, 시험, 설치. 점검과 같은 각 확인 검증 활동에서 수행되는 확인 검증 작업의 결과를 요약한다. 운영 활동과 유지관리 활동 기간 동안에 확인 검증 활동 요약 보고서는 이전의 확인 검증 활동 요약 보고서나 기타 문서를 갱신시킬 수 있다. 각 확인 검증 활동 요약 보고서는 다음 내용을 포함한다. 1) 수행된 확인 검증 작업 설명 2) 작업 걸과 요약 3) 예외사항과 해결방법 요약 4) SW 품질 평가 5) 기술적, 관리적 위험의 확인 및 평기- 6) 권고사항 다) 예외사항 보고서: 예외사항 보고서는 확인 검증 시도에 의해서 발견된 각 예외사항을 문서화한다. 각 예외사항은 SW 시스템에 미치는 영향과 중대성에 따라 평가된 다. 확인 검증 활동과 작업의 범위와 응용은 예외사항과 위험의 원인을 규명하기 위해 수정 될 수 있다. 각

	예외사항 보고서는 다음 내용을 포함한다. 1) 문서나 코드에서 설명과 위치 2) 영향 3) 예외사항의 원인과 오류발생의 설명 4) 예외의 중대성 5) 권고사항 6) 확인 검증 최종보고서: 확인 검증 최종보고서는 설치와 점검 활동의 마지막 시점 또는 확인 검증 시도를 내리는 시점에서 작성된다. 확인 검증 최종보고서는 다음 내용을 포함한다. (1) 모든 수명주기 확인 검증 활동 요약 (2) 작업 결과 요약 (3) 예외사항과 해결방법 요약 (4) 전체적인 SW 품질 평가 (5) 교훈/최고의 실무 (6) 권고사항 선택적 보고서는 다음을 포함할 수 있다. 가) 특정 연구 보고서 이 보고서는 SW 수명주기 기간에 수행된 특정 확인 검증 연구를 포함한다. 이 보고 서의 제목은 주제에 따라 다양하게 지정될 수 있다. 보고서는 기술적, 관리적 작업의 결과를 문서화하며. 다음 내용을 포함한다. 1) 목적과 목표 2) 접근방법 3) 결과 요약 나) 기타 보고서 이 보고서는 SW 확인 검증 절차(SVVP)에서 정의되지 않은 작업의 결과를 기술한다. 이 보고서의 제목은 주제에 따라 다양하게 지정될 수 있다. 예를 들어, 이러한 작업 보고서는 품질보증 결과, 사용자 시험 결과, 안전성 평가 결과, 형상 및 데이터 관리 상태 결과 등을 포함할 수 있다.
7. 확인 검증 관리적 요구사항	관리적 확인 검증 요구사항은 예외사항의 해결방법과 보고, 작업 반복정책, 일탈정책, 제어 절차, 표준, 관행, 용어 설명에 관하여 명시한다.
7.1 예외사항 해결방법 및 보고	예외사항 보고의 기준, 예외사항을 보고하는 분배 목록, 예외사항을 해결하는 권한과 시간에 관한 방침을 포함한 예외사항을 해결하고 보고하는 방법에 관하여 명시한다.
7.2 작업 반복 정책	작업의 입력이 변경되고, 작업 절차가 바뀔 때 확인 검증 작업이 어느 정도 재수행되는 정도를 결정하기 위해 사용되는 기준에 관하여 명시한다. 이러한 기준은 변화의 평가, SW 무결성 수준, 예산, 스케줄, 품질에 미치는 영향 등을 포함할 수 있다.

7.3 일탈 정책	계획에서 변경된 절차와 기준에 관하여 기술한다. 일탈에 필요한 정보는 작업 정의, 논리적 근거, SW 품질에 미치는 영향 등이 포함된다. 일탈을 승인하기 위해서 책임 있는 권한을 확인한다.
7.4 통제 절차	확인 검증 시도에 적용되는 통제 절차를 정의한다. 이러한 절차는 SW 결과물과 확인 검증 결과가 어떻게 형상화 되고, 보호되며, 저장되는지를 명시한다.
7.5 표준 및 관행 설명	조직 내부의 표준, 관행, 정책을 포함하여 확인 검증 작업의 성능을 결정하는 표준, 관행, 규약을 검증한다.
8. 확인 검증 문서화 요구사항	시험 문서의 목적, 형식, 내용을 정의한다. 이러한 시험문서의 형식에 관한 설명은 IEEE Std 829-1983 을 참조할 수 있다. 다음과 같은 확인 검증 시험 문서를 위해 목적, 형식, 내용에 관하여 설명한다. 가) 시험 계획 나) 시험 설계 다) 시험 사례 라) 시험 절차 마) 시험 결과 모든 확인 검증 의 결과와 결론은 확인 검증 최종 보고서에 나타나 있다.

[CM-V-02] SW 확인 검증 보고서

SW 확인 검증 절차에서 요구하는 SW 확인 및 검증 보고서(즉, 확인 및 검증 작업보고서, 확인 및 검증 활동 요약 보고서, 확인 및 검증 비정상 보고서, 그리고 확인 및 검증 최종 보고서)는 다음과 같은 내용으로 구성되고, 필요한 경우 SW 확인 및 검증 특성에 적합하게 목차를 재구성할 수 있다.

가. 목 차

나. 작성법

1. 작업 보고서	확인 검증 작업은 확인 검증 작업결과 및 상태를 문서화하여야 하고, 기술적으로 명확하게 작성한다. 작업보고서는 다음 내용을 포함한다. (1) 비정상 평가(evaluation) (2) 기준선 변경 평가 (3) 개념 문서 평가 (4) 형상관리 평가 (5) 계약 확인 (6) 중요도 분석 (7) 신규 제한사항의 평가 (8) HW/SW/사용자 요건 할당 분석 (9) 위험원 분석 (10) 설치 점검 (11) 설치 형상 실사 (12) 인터페이스 분석

	(13) 이진 평가 (14) 운영 절차서 평가 (15) 제안된 변경 평가 (16) 권고내용(recommendations) (17) 검토결과 (18) 위험성 분석 (19) SW 설계 평가 (20) SW 무결성 수준(SIL) (21) SW 요건 평가 (22) 소스코드 및 소스코드 문서 평가 (23) 장치 요긴 검토 (24) 시험결과 (25) 추적성 분석
2. 확인 검증 활동 요약 보고서	확인 검증 활동에 대하여 수행된 확인 검증 작업의 결과를 요약한다. 확인 검증 활동에는 취득 지원, 계획, 개념, 요건, 설계, 이행, 시험 및 설치/점검이 포함된다. 운영 활동 및 유지보수 활동에 대해, 확인 검증 활동 요약 보고서는 이전 확인 검증 활동 요약 보고서 또는 다른 분리문서를 개정할 수 있다. 다음 내용을 포함한다. (1) 수행된 확인 검증 작업 설명 (2) 작업결과의 요약비정상 및 해결내용에 대한 요약 (3) SW 품질 평가 (4) 기술 및 관리 측면의 위험성 식별 및 평가 (5) 권고내용
3. 비정상 보고서	확인 검증 노력에 의해 검출된 각 비정상 이슈에 대해 문서화한다. SW 장치 측면의 영향을 평가(evaluate)하고 중요도를 평가한다. 확인 검증 활동 및 작업의 범위는 도출된 비정상 및 위험성의 원인을 다루기 위해 개정되어야 한다. 다음 내용을 포함한다. (1) 문서 또는 코드 내에서의 설명 및 위치 (2) 영 향 (impact) (3) 비정상의 원인 및 오류 시나리오의 설명 (4) 비정상 중요도 정도(level) (5) 권고내용
4. 확인 검증 최종 보고서	설치/점검 활동의 마지막 부분 또는 확인 검증 노력의 종결부분에 포함한다. 다음 내용을 포함한다. (]) 모든 수명주기 확인 검증 활동 요약 (2) 작업결과의 요약 (3) 비정상 및 해결의 요약 (4) 개괄적인 SW 품질의 평가 (5) 교훈/최적기법 (6) 권고내용

	선택적 보고서는 다음을 포함할 수도 있다. (1) 특별 연구 보고서 SW 수명주기 동안 수행한 연구대상과 결과를 기술한다. 이 보고서의 제목은 주제에 따라 변경될 수 있다. 이 보고서에는 기술적/관리적 작업의 결과를 다음 내용을 포함하여 문서화한다. 목적 접근방법 결과 요약 (2) 기타 보고서 SW 확인 검증 절차(SVVP)에 정의하지 않은 작업결과에 대해 기술한다. 이 보고서의 제목은 주제에 따라 변경될 수 있다. 품질보증 결과, 최종 사용자 시험 결과, 안전성 평가 보고서, 또는 형상 및 데이터 관리 상태 결과 등이 될 수 있다.

[CM-V-03] 형상관리 계획서

IEEE 828 을 참고하여 작성한 SW 형상관리계획서 작성시의 목차를 권고한다.

가. 목 차

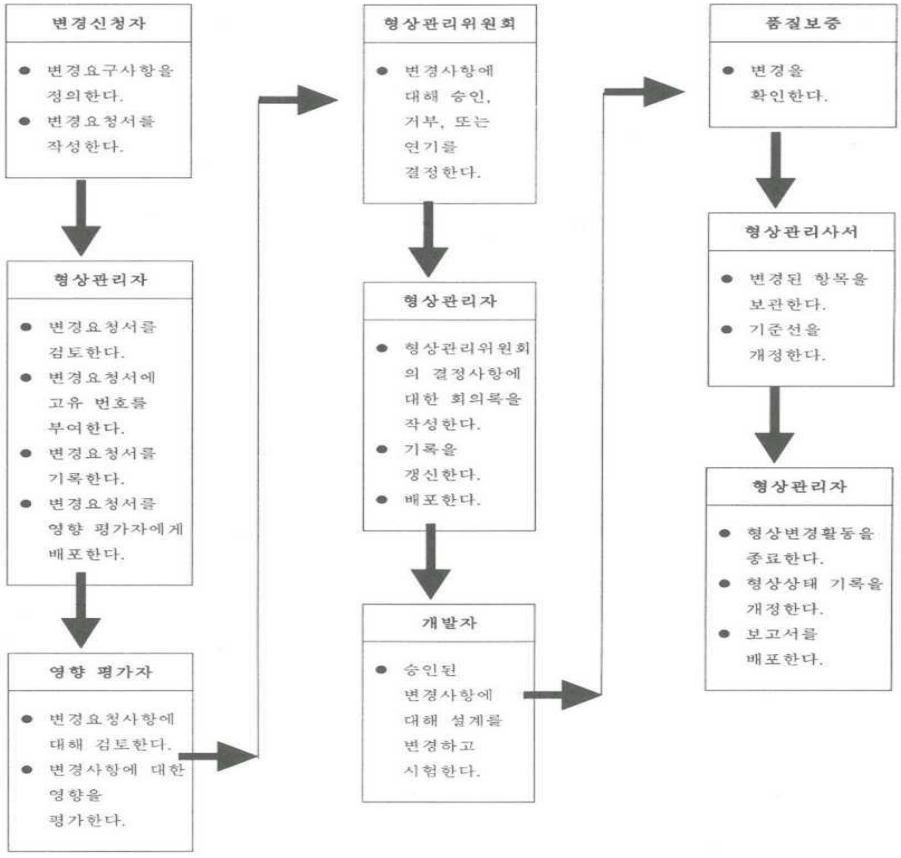
나. 작성법

1. 목적	소프트웨어 형상관리 계획서의 목적을 기술한다. SW 형상관리 계획서를 작성하는 이유와 SW 형상관리계획서가 어떠한 역할을 수행할 것인지에 대해 간략히 기술한다.

2. 범위	SW 형상관리계획서가 포함해야 하는 영역을 지정한다. SW 형상관리계획서를 통해 형상관리계획서의 전체적인 구조를 파악할 수 있도록 기술한다.
3. 참고문서	SW 형상관리계획서를 작성하는 데 참고가 된 법규, 기술기준, 설계문서들을 기술한다.
4. 용어 및 약어정의	SW 형상관리계획서에서 나오는 용어와 약어에 대해서 정의한다. 문서의 전문전인 용어는 자세하게 설명하여 전체내용을 이해하는데 도움이 될 수 있도록 구성한다.
5. 형상관리 개요	SW 형상관리활동이 적용되는 사업조직, 이들 조직의 형상관리 책임, 사업에 적용되는 형상관리 정책 및 지침 등을 명시한다.
5.1 조직	SW 형상관리 활동에 참여하거나, 책임이 있는 모든 조직단위, 사업 구조내의 조직 단위들의 기능적 역할, 조직 단위 간의 관계 등을 명시한다. 조직단위는 공급자와 고객, 주 계약자와 하청계약자, 또는 단일 조직 내의 여러 다른 그룹으로 구성될 수도 있다.
5.2 책임	조직 단위에 대한 SW 형상관리 활동의 책임을 규정한다.
5.3 정책, 방향 및 절차	관련된 정책, 지침, 절차에 의해 계획서에 부과되는 모든 외부 강제사항을 명시한다. 각각의 강제사항이 계획서에 미치는 영향 및 효과 등을 기술한다.
6. 형상관리 업무	계획서의 적용범위에서 규정된 SW 의 형상을 관리하기 위해 요구되는 모든 기능과 업무를 명시한다. 형상관리활동은 일반적으로 형상식별, 형상통제, 상황보고, 형상감사 및 검토 등 4 개의 기능으로 구분된다.
6.1 형상식별	SW 사업에서 관리되어야 할 각종 코드, 시방서, 설계 문건 및 데이터들에 대해 물리적, 기능적 특성을 식별(identification), 명명(naming) 및 문서화 한다. 관리 대상 항목들은 중간 결과물 또는 최종 결과물 (예를 들면, 실행코드, 소스코드, 사용자매뉴얼, 프로그램 리스팅, 데이터베이스, 테스트 사례 및 시방서 등) 및 각종 지원 환경(예를 들면, 컴파일러, 운영시스템, 프로그램 툴 및 테스트 베드)을 포함한다. 다음은 형상관리 식별 절차이다. -형상항목 식별 (Identifying CIs) SCMP 에서는 통제되어야 할 형상항목들 각각에 대한 정의를 명시한다. SCMP 는 과제에 대해서 형상항목과 구성과 유지방안에 대해 명시한다. -형상항목 명명 (Naming) SCMP 는 통제되는 각 항목에 해당하는 특정한 식별 표시를 할 수 있는 식별 체계를 명시한다. SCMP 는 저장, 검색, 추적, 재생산 및 배포를 목적으로 통제되는 형상항목들에 대해서 명명하는 방법을 기술한다. -형상항목 획득(Acquiring) 과제에 대한 통제 대상 SW 라이브러리를 식별하고 코드, 문서, 그리고 식별된 기준선의 데이터를 어떻게 적절한 라이브러리 내에서 물리적으로 통제할 수 있는지 명시한다.

	물리적인 표식 및 형상항목에 대한 라벨링(labeling)을 포함함 문서와 자기 미디어(magnetic media)의 실제적인 보관을 위한 절차를 기술한다. 라이브러리 저장소로부터 통제된 형상항목을 어떻게 검색하고 재 생성할 것인지 기술한다.
6.2 형상통제	형상통제란 형상 기준선 상에 해당되는 형상항목들에 대해 변경 사항(에러 수정이나 기능 향상)을 요청, 평가, 승인/불허 또는 시행하는 업무이다. 다음의 단계적인 상세 업무에 대해 기술한다. 가. 변경사항 필요성 식별 및 문서화 나. 변경 요청에 대한 분석 및 평가 다. 변경 요청에 대한 승인 또는 비승인 라. 변경에 대한 확인, 시행 및 출시 SW 개발 팀에서 변경사항이 필요하면 SW 변경의뢰서(SCR: Software Change Request) 를 작성해서 형상관리위원회 (CCB: Configuration Control Board)에 제출하여 승인을 받는다. CCB 에서 검토 후 승인이 나면 개발 팀은 변경사항을 반영하고 SW 변경 승인서 (SCA: Software Change Authorization)을 작성해서 주 라이브러리(master libraries)를 변경하고 제출된 SCR 에 대해 최종적인 마감을 한다.
6.3 형상상태 기록	형상관리 항목들의 형상상태에 대해 기록하고 유지, 보고하는 업무를 말한다. 다음 내용을 포함한다. 가. 기준선 및 변경사항에 대해 추적 및 보고해야 하는 데이터 요소 나. 상태보고서의 형식 및 보고 빈도 다. 정보의 수집, 저장, 처리 및 보고방법 라. 상태 데이터의 접근통제 방법
6.4 형상감사 및 검토	형상 감사는 실제적인 형상항목이 요구되는 물리적, 기능적 특성을 얼마만큼 만족하는가를 검토하는 업무이다. 형상검토는 형상기준선을 구축하기 위한 관리상의 방법이다. 형상감사 및 검토에는 사업의 형상감사 및 형상검토 계획을 기술하고 다음 사항을 포함한다. 가. 목적 나. 감사 혹은 검토할 형상항목 다. 감사 혹은 검토 업무일정 라. 감사 혹은 검토 수행 절차 마. 참석자 직책 바. 검토에 이용 가능해야 하거나, 혹은 감사나 검토 지원을 위해 필요한 문서 사. 부적합사항의 기록 및 수정조치의 보고 절차 아. 승인기준 및 승인 후 특정조치
6.5 연계성 통제	연계성 통제활동에서는 사업의 형상항목의 변경에 대하여 계획서의 범위 외의 연계의 변경을 조정한다. 사업에 영향을 미칠 수 잠재적 연계를 확인하기 위해 다른 사업과 인도할 제품뿐만 아니라 HW, 운영 SW, 지원 SW 도 조사한다.

	연계성 통제에는 사업 SW와 연계하는 외부 항목을 식별하고, 각 연계에 대해 다음과 같은 사항을 규정해야 한다. 가. 연계속성 나. 영향을 받는 조직 다. 연계 코드, 문서 및 자료의 통제방법 라. 연계통제 문서의 승인 및 규정된 기준선으로의 배포 방법
6.6 하도급/공급자 통제	하청/공급자 통제 활동에서는 사업환경 외부에서 개발된 항목을 사업의 형상항목으로 통합한다. 하청계약 된 SW에 대해서 사항에 대해 기술한다. 가. 형상관리 계획을 포함하여, 하청계약서의 일부가 되어야 할 형상관리의 요건 나. 하청계약자의 관리방법 다. 하청계약 항목의 형상감사 및 검토 라. 외부코드, 문서 및 데이터의 시험, 확인, 인수 및 사업 SW로의 통합 마. 정보의 보안과 소유권의 추적을 위한 재산항목의 취급방법 바. 하청계약자의 참여를 포함한 변경사항 처리방법
7. 형상관리 일정	모든 형상관리 활동사항에 대한 연속 및 종속성과 사업 이정표 또는 이벤트에 대한 주요 형상관리 활동사항의 관계에 대해 명시한다. 일정계획의 기간과 형상관리활동사항에 대한 사업의 모든 주요한 이정표에 대해 기술한다. 형상관리 이정표는 형상 기준선 구축, 변경 통제 절차의 시행, 형상감사의 시작 및 종료일에 대한 내용을 포함한다. 일정은 절대적인 날짜로 표시하거나 형상관리 또는 사업 일정에 상대적인 날짜로 또는 간단한 이벤트의 순서로 표시한다. 그래픽 표현이 특히 이 정보를 보이는데 적합하다.
8. 형상관리 자원	형상관리 활동의 수행을 위해 필요한 SW 도구, 기법, 기기, 인원 및 훈련 등을 명시하고, 형상관리도구 선정 과정을 기술한다. 형상관리 도구가 적용되는 HW 및 SW 환경, 그리고 필요한 통제사항을 기술한다.
9. 형상관리 계획서 유지관리	사업의 생명주기 동안 형상관리 계획서의 유지관리를 위해서 필요한 활동과 책임을 기술하며, 다음 사항을 포함한다. 가. 계획의 감시 책임자 나. 개정작업의 빈도 다. 계획 변경의 평가 및 승인방법 라. 계획 변경의 수행 및 전달방법 이 지침은 SW 형상항목의 변경을 위한 변경절차에 대한 지침을 제공하고 있다. 그림 1은 형상항목의 변경절차를 보여준다. **그림 1. 형상항목 변경절차**



해당 형상항목에 대한 변경의 필요성을 인지한 사람이 변경신청자가 될 수 있다.

형상관리자는 변경요청서를 검토하여 관련 SW 개발조직, 확인 및 검증조직, 안전성 분석조직에게 변경요청서를 배포하여 해당 형상항목이 변경되었을 경우의 영향을 평가한다. 형상항목의 변경에 대한 영향평가는 다음과 같은 관점에서 이루어져야 한다.

- 시스템 기능에 주는 영향정도
- 시스템 연계에 주는 영향정도
- 형상항목 변경에 드는 경제적, 인적 및 시간적 비용
- 형상항목 변경에 따른 일정의 변경정도
- SW 안전성, 신뢰성, 유지보수성에 미치는 영향정도

[SR-D-01] SW 요구사항 명세서(SRS)

SW 의 요구사항명세서가 가질 수 있는 목차를 IEEE std. 830-1984 표준을 참고하여 아래와 같이 구성할 수 있고, 목차는 요구사항명세서를 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경우 SW 의 특성에 따라 다른 목차를 사용할 수 있다.

가. 목 차

12. 기타

나. 작성법

1. 목적	SRS 의 목적을 기술한다.
2. 범위	SRS 의 적용 범위를 기술하고, 세부 장, 절의 내용에 대해 간단하게 기술한다.
3. 용어정의 및 약어	컴퓨터 시스템의 인터페이스 및 기능을 기술하는데 사용되는 단어들의 정의와 약어 및 표기법을 설명한다. 약어 및 표기법은 입출력 변수 및 기능 혹은 함수들의 이름을 짓는 규칙과 같은 SRS 내에 사용되는 이름을 짓는 규칙 및 각종 표기법의 의미를 기술한다.
4. 참고문헌	SKS 작성 시에 활용한 참고문헌과 SKS 내용의 이해에 필요한 참고문헌을 기술한다.
5. 컨텍스트	전체 시스템의 개략적인 내용을 보여준다. 시스템의 전체적인 구성을 설명하고, 컴퓨터 시스템과 외부 장치 간의 인터페이스에 중점을 두고 기술한다. 컴퓨터 시스템과 전체 시스템 간의 경계를 분명하게 정의한다. 전체 시스템을 구성하는 각각의 기능들을 기술하고 컴퓨터 시스템의 입/출력, 입/출력에 따른 컴퓨터 시스템의 동작 특성 등을 기술한다. 전체 시스템과 그 시스템 내의 HW 및 이미 개발된 SW 그리고 개발할 SW 를 구분 지어 기술하고 각 부분 사이의 인터페이스를 기술한다. 특히 개발되는 SW 가 반드시 분명하게 드러나야 한다. 또한, 컴퓨터 시스템이 어떠한 제약조건(Constraints)에서 동작하는가를 기술한다. 제약조건에는 사용자 인터페이스, HW 인터페이스, SW 인터페이스, 통신 인터페이스, 메모리 인터페이스 및 메모리 제약사항, 운전 및 설치 요구사항 등이 포함된다.
6. 인터페이스 요구사항	컴퓨터 시스템과 전체 시스템 간의 인터페이스 요구사항을 정의한다. 컴퓨터 시스템의 인터페이스는 입력변수와 제어변수 간의 관계로서 정의한다.
6.1 입력변수	입력변수에 대한 다음과 같은 내용을 기술한다. 가) 입력변수에 직접 연결되어 있는 외부장치를 정의한다. 1) 인터페이스 포인터에 연결되어 있는 외부 장치 나) 입력변수를 정의한다. 2) 입력변수의 이름 3) 입력 변수가 가리키는 인터페이스 포인터 4) 입력 타입: 아날로그 신호, 디지털 입력, 혹은 시리얼 입력 등 5) 입력변수가 갖는 유효범위 6) 5) 항목에서 기술된 유효범위의 단위

	다) 입력변수와 관련 있는 내부입력변수를 정의한다. 7) 내부입력변수의 이름 8) 내부입력변수의 접근하는데 필요한 SW 정보: 레지스터 주소 등 9) 내부입력변수의 값의 형식과 범위 라) 입력변수와 감시 변수간의 연관성을 기술한다. 10) 입력 변수에서 감시 변수로의 입력 혹은 복사를 일으키는데 필요한 액션 혹은 이벤트의 상세내역 11) 입력변수 값과 감시 변수 값 사이의 관계
6.2 출력변수	출력변수에 대한 다음과 같은 내용을 기술한다. 가) 제어변수에 의해 기술되는 인터페이스 포인터와 직접 연결되어 있는 외부장치를 정의한다. 1) 인터페이스 포인터에 연결되어 있는 외부 장치 나) 출력변수를 정의한다. 2) 출력변수의 이름 3) 출력변수가 가리키는 인터페이스 포인터 4) 출력변수의 타입: 아날로그 신호, 디지털 입력, 혹은 시리얼 입력 등 5) 출력변수가 가정하는 유효범위 6) 5) 항목에서 기술된 유효범위의 단위를 기술한다. 다) 출력 변수와 관련 있는 내부출력 변수 7) 내부출력변수의 이름 8) 내부출력변수의 접근하는데 필요한 SW 정보: 레지스터 주소 9) 변수의 값의 형식과 범위 라) 제어변수와 출력 변수 간의 연관성 10) 제어변수에서 출력변수로의 전이를 일으키는데 필요한 액션 혹은 이벤트에 대한 세부내용 11) 출력변수 값과 제어변수 값 사이의 관계
7. 기능(Function) 요구사항	컴퓨터 시스템의 원하는 행위를 기술한다. 이 절에서는 컴퓨터 시스템의 기능을 기술하는 기능적 요소와 그 기능 요소를 수행하는데 대한 행위적 요소의 혼합된 형태로 기술한다. 기능 구성 요소들은 입력변수와 출력변수 혹은 이들을 각각 연결하고 각각의 기능간의 이벤트 및 데이터를 주고받는 관계로 기술된다. STATEMATE 를 활용하는 경우 액티비티 차트 배경도 또는 자료흐름도를 사용하여 기술할 수 있다. 또한 기능들의 제어, 즉 기능들의 시작 및 종료, 기능의 수행 조건 및 데이터 이벤트의 처리 등을 기술한다.
8. 성능 요구사항	시간적 요구사항 SW 실행 시, 시스템과 SW 간, 혹은 SW 내부적으로 만족해야 할 시간적 제약을 기술한다. 입력변수 출력변수

	출력이 나오기까지의 제한된 시간 성능에 대한 시간적 제약의 요구사항은 순차도(Sequence Diagram) 등을 통해 특정 입력 변수에 대해 출력변수의 시간적 제약 표시함으로 기술할 수 있다.
9. 설계 제약사항	
9.1 HW 환경	자연어로 다음 사항을 기술한다. · 컴퓨터의 연산속도 · 메모리 크기 · 사용 가능한 인스트럭션 수 · 외부 포트의 수 · 컴퓨터와 인터페이스 하는 아날로그 혹은 디지털 입력 혹은 출력 수
9.2. SW 환경	자연어로 다음 사항을 기술한다. · SW 개발 언어 · 개발 컴파일러 정보 · SW 개발 환경 · 개발되는 SW 언어의 제약사항 SW 의 환경.
10. 신뢰성 요구사항(Reliability Requirements)	SW 가 제대로 동작하여 신뢰할 수 있는 상태에 대해 확률적 혹은 통계적으로 정의한다.
11. 안전성 요구사항(Safety Requirements)	SW 가 안전한 상태를 유지하기 위한 필요한 모든 요구사항을 다음 요소들을 포함하여 기술한다. 입력변수 처리기능 안전한 상태 혹은 출력변수의 값(Safety state or value of output variable) 입력 변수는 특정한 상태에서 시스템의 안전에 영향을 줄 수 있는 입력변수를 가리킨다. 처리 기능은 시스템의 안전에 영향을 줄 수 있는 입력에 대해 처리하는 기능을 말한다. 안전한 상태는 시스템의 안전에 영향을 줄 수 있는 입력을 추리한 후, 갖게 되는 안전한 상태를 가리킨다. 예를 들어 어떠한 입력의 조합이 들어왔을 때, 반드시 수행해야 하는 기능과 그리고 그러한 기능들에 의해 처리된 후의 안전한 상태 혹은 안전한 상태를 가리키는 출력을 기술한다. 이것은 Sequence Diagram 으로 기술할 수 있다.
12. 기타	기타 항목은 모두 자연어로 식별하기 쉽게 기술한다.

다. IEC 61508 에서 제시하는 요건

IEC 61508 에서 제시하는 안전기능 요구사항명세 포함 내역 (IEC 61508 - 2:2010 7.2.3)
a) 각 안전기능에 대하여 요구되는 기능안전성을 달성하기 위해 필요한 모든 안전기능에 대한 기술 - E/E/PE 안전관련 시스템의 설계와 개발에 충분할 정도로 광범위하고 자세하여야 한다. - E/E/PE 안전관련 시스템이 EUC 안전 상태를 달성하고 유지할 수 있는 방법에 대한 설명이 포함되어야 한다. - EUC 의 안전 상태를 확보, 유지하기 위해 계속적인 통제 필요 여부와 통제기간을 명시하여야 한다. 그리고 - 저요구 작동모드이거나 고요구 작동모드 또는 연속적인 작동모드의 방식으로 운영할 때 해당 안전기능이 E/E/PE 안전관련 시스템에 적합한 지 여부를 명시하여야 한다. b) 처리량과 응답시간 성능 c) 기능안전성을 달성하는 데 필요한 E/E/PE 안전관련 시스템과 운영자의 인터페이스 d) 기능안전성에 관련되어 E/E/PE 안전관련 시스템의 설계에 영향을 미칠 수 있는 모든 정보 e) E/E/PE 안전관련 시스템과 기타 시스템 사이의 모든 인터페이스(EUC 내부와 직접 관련되거나 EUC 외부적인 것.) f) 다음을 포함하는 EUC 의 모든 관련 운영 모드 - 설치와 조정 등의 사용 준비 - 시동, 자동, 수동, 반자동, 일정 상태의 운영 - 비운영 지속 상태, 재설정, 중단, 유지 - 합리적 예측이 가능한 비정상 조건 비고 1 합리적으로 예측 가능한 비정상 조건이란 개발자나 사용자가 예측할 수 있는 조건이다. 비고 2 특정 형태의 운영(예 : 설치, 조정, 유지)에는 이러한 운영이 안전하게 수행될 수 있도록 하기 위하여, 추가적 안전기능을 요구해도 된다. g) E/E/PE 안전관련 시스템에 요구되는 모든 모드 - 특히, E/E/PE 안전관련 시스템의 고장 행태와 그에 필요한 대응(예 : 경고, 자동 중단 등)을 자세하게 기술하여야 한다. h) 모든 하드웨어/소프트웨어 상호작용의 중요성 - 필요시, 하드웨어와 소프트웨어 사이에 필요한 모든 제약조건을 확인하고 기록하여야 한다. 비고 3 설계가 완료되기 전에 이런 상호작용들을 알 수 없는 경우에는, 일반적 제약조건만 언급할 수 있다. i) E/E/PE 안전관련 시스템과 관련 서브시스템의 제약과 제한조건(예 : 시간 제약) j) E/E/PE 안전관련 시스템을 시동, 재시동하는 절차에 관련된 구체적인 요구사항
IEC 61508 에서 제시하는 안전무결성 요구사항명세 포함 내역
a) 각 안전기능에 대한 안전무결성수준과 필요 시, 안전기능에 대해 요구되는 목표 고장 기준

비고 1 안전기능의 안전무결성수준은 IEC 61508 - 1 에 따라, 안전기능에 대한 목표 고장 기준을 결정한다.

비고 2 안전기능을 위해 필요한 리스크 감소가 정량적 방법을 통해 도출된 경우, 안전기능의 목표 고장 기준을 명시해야 할 필요가 있다.

b) 각 안전기능의 작동모드(저요구 작동모드, 고요구 작동모드 또는 연속적인 작동모드)

c) E/E/PE 하드웨어에 실시되어야 하는 증명시험을 가능하게 하기 위한 요구사항, 제약, 기능, 시설

d) 제조, 보관, 이송, 시험, 설치, 작동점검, 운영, 유지 등의 E/E/PES 안전수명주기 동안 일어날 수 있는 모든 극한의 환경 조건

e) 전자기 적합성을 얻기 위해 필요한 전자기 내성 한계(IEC 61000 - 1-1) - 전자기 내성 한계를 도출하기 위해 전자기 환경(IEC 61000 - 2-5)과 요구되는 안전무결성수준을 모두 고려하는 것이 좋다.

[SR-D-02] SW 안전 기록

SW 안전 기록이 가질 수 있는 목차를 참고하여 아래와 같이 구성할 수 있고, 목차는 안전 기록을 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경우 SW 의 특성 및 단계별 관리할 포인트에 따라 적절히 변경하여 사용할 수 있다.

가. 목 차

나. 작성법

1. 도입	이 항목은 위험원 로그의 목적을 설명하고 시스템 안전 특징과 연관된 환경과 안전 요구사항을 제시한다. 그리고 다음과 같은 내용을 포함한다. • 프로젝트에 참여하는 모든 사람들이 이해할 수 있도록 충분한 상세자료를 제시하며 안전 로그의 목적, 취지. 목차 • 위험원 로그와 관련된 시스템의 단일 식별자, 시스템에 대한 설명과 영역에 대한 참조 • 안전계획서 참조(프로젝트 초기 단계에는 제외될 수도 있음) • 시스템 안전 요구사항서나, 이것이 아직 작성되어 있지 않은 경우에는 안전 분석 문서를 참조 • 위험원 로그 변경과 같은 위험원 로그 관리 프로세스와 신규 입력사항 발생 시의 승인 프로세스
2. 일지	일지에는 위험원 로그가 복잡해짐에 따라 이에 대한 날짜순서 기록을 제공하고 추적성을 제공하기 위해 위험원 로그에 일어나는 모든 변경사항을 기술한다. 변경할 때마다 다음 사항을 기록한다. • 수정일자

	• 입력 식별번호 • 수정하는 자 • 수정내용과 수정사유에 대한 설명 • 변경된 위험원 로그 섹션 위험원 로그를 데이터베이스에 저장해 두는 경우 자동화된 도구로 관리할 수 있다.
3. 디렉토리	안전 기록로그라고 알려진 디렉토리는 프로젝트에서 작성되고 사용된 모든 안전문서에 대한 업데이트된 내용과 참조문서이다. 다음과 같은 참조 문서를 제공한다. (아래 리스트에 더 추가 가능). • 안전계획서 • 안전 요구 사양서 • 안전 표준 • 안전 문서 • 사건/사고 경위서 • 분석, 평가 및 심사보고서 • 종합안전대책기술서 • 관련 안전 관리 기관의 회신 각 문서에 대해 디렉토리는 다음 항목들을 포함한다. • 고유 식별 번호 • 문서 제목 • 현재 버전 번호와 작성일자 • 원장 보관 위치 디렉토리는 위험원 로그와 별도로 관리하는 것이 편리할 수도 있다. 디렉토리를 프로젝트 문서관리 시스템과 통합해 두는 경우도 있다

4. 위험한 데이터	도출된 위험원 위험원에 대해 다음의 항목 정보 포함해 기록한다. 위험원 분석과 리스크 평가 과정에서 입수된 정보는 보고서 작성 이 완료 후 위험원 로그에 기록한다.
	• 고유 식별 번호
	• 영향을 받는 시스템 기능 또는 구성요소와 위험원을 형성하는 상태를 기술하는 위험원에 대한 간략한 설명
	• 위험원으로 도출된 원인
	• 위험원에 대한 전체 설명과 분석 참조
	• 분석이 근거하는 가정과 분석의 제약사항
	• 위험원이 기여 인자로 작용하여 발생되는 사고의 심각성
	• 위험원 발생 가능성, 사고발생 가능성
	• 위험원과 관련된 예상 리스크발생 가능성 목표위험원의 상태
	  • 미결: 위험원을 종결하고자 하는 행동이 취해지지 않는 상태
	  • 취소: 이벤트가 위험원이 아니거나 전적으로 다른

	위험원내에 포함되어 있는 것 으로 결정됨 • 해결: 위험원을 종결하고자 하는 행동은 취해졌으나 아직 완료되지 않은 상태 • 종결: 위험원을 종결하고자 하는 행동이 완료됨 • 위험원이 종결되지 않거나 취소일 때 종결까지 진행시키는 책임을 맡은 사람이나 회사의 이름 • 위험원을 제거하거나 시스템에서 허용 가능한 수준으로 리스크를 감소시키기 위해 취해져야 하는 조치를 언급하거나 참조
5. 위험원	도출된 위험원에 대해 다음 내용을 포함하여 기록한다. • 위험원이 저지되었는지 아니면 추가적인 조치가 필요한지에 대한 언급(추가적인 조치를 취하지 않는다면 그 이유에 대해 설명) • 취해야 할 리스크 감소조치에 대한 상세 내용 • 리스크 대체수단에 대한 논의 • 리스크 감소와 더불어 고려는 이루어졌지만 실행에 옮겨지지 않은 조치에 대한 이유 설명 • 리스크 감소조치 이후 사고 발생 재평가 필요성에 대한 언급 • 조치의 결과로 변경할 설계문서에 대한 언급 • 모든 관련 안전 요구사항에 대한 언급
6. 사건 데이터	시스템이나 장비의 라이프사이클 동안 발생된 모든 사고를 기록 한다. 각 사건과 그 사건을 유발시키는 위험원을 연결시켜 일련의 이벤트를 다음 내용을 포함하여 기술한다. • 고유 식별 번호 • 사건의 간략한 설명 • 사건 경위 조사 보고서에 대한 언급 재발 방지를 위해 취해진 조치에 대한 설명이나 조치를 취하지 않았다면 그 이유 설명
7. 사고 데이터	도출된 모든 사고를 기록하는데 사용한다. 각 사고와 사고를 유발하는 위험원을 연결시켜 일련의 이벤트를 다음 내용을 포함하여 도출한다. • 고유 식별 번호 • 사고의 간략한 설명 • 사고 발생건수 상세설명 및 분석 보고서에 대한 언급 • 사고 심각성에 대한 카테고리 구분과 사고발생 허용 상한율 (사고 확률 목표) 사고를 유발할 수 있는 위험원과 관련된 사고전과 목록

[SR-D-03] SW 요구사항 안전성분석 보고서

SW 에 요구사항에 대한 안전성 분석 보고서의 작성과 관련하여 SW 요구사항 안전성분석 보고서에 포함되어야 할 내용들에 대한 최소요건 들을 규정하고 있다.

SW 의 요구사항 안전성분석 보고서에 포함되어야 할 내용은 다음과 같다.

가. 목 차

나. 작성법

1. 목적	안전성분석의 수행 목적을 정의한다.
2. 범위	안전성분석의 수행 범위을 정의한다.
3. 용어 정의 밑 약어	안전성분석 보고서에 사용된 어떤 용어 및 약어를 정의하고 설명한다.

4. 참고 문서	안전성분석에 사용된 근거법령, 근거기준설계문서, 기타 문서들을 기술한다.
5. 시스템 개요	다음 내용을 기술한다. - 상위수준의 시스템 구조 - 시스템, 하위시스템, 그리고 연계관계
6. 안전성 분석기법	다음 내용을 기술한다. SW 안전요구사항 SW 안전성분석에 사용된 입력자료 SW 요구사항 안전성분석에 사용된 기법 SW 요구사항 안전성분석에 사용된 절차 SW 요구사항 안전성분석에 사용된 도구
7. 안전성 분석결과	다음의 안전성분석의 결과를 기술한다. SW 안전요구사항 적합성분석의 결과 SW 안전요구사항 위험원분석의 결과
8. 안전성 검토	시스템 개발조직, 시스템 안전성분석조직 및 소프트웨어 개발조직과 안전성 분석 검토회의를 수행 후, SW 요구사항명세서에 포함된 위험원의 수정 및 보완이 발생한 경우 이에 대해 기술한다
9. 승인	요구사항 안전성보고서 자체에 대한 공식적인 검토와 승인해야할 개인의 목록을 규정해야 하고 서명을 받는한다.

[SD-D-01] SW 설계 명세서

SW 의 설계명세서가 가질 수 있는 목차를 아래와 같이 구성할 수 있고, 제시한 목차는 설계명세서를 기술하기 위한 일반적인 사항들로 구성되어 있으며 필요한 경 우 SW 의 특성에 따라 다른 목차를 사용할 수 있다.

가. 목 차

6.4 상세 설계기술(Detailed Description) 6.4.1 모듈 상세 설계 (Module Detailed Design) 6.4.2 데이터 상세 설계(Data Detailed Design)

나. 작성법

1. 목적	SW 설계명세서의 목적을 기술한다. SW 설계명세서를 작성하는 이유와 설계명세서가 어떠한 역할을 수행할 것인지에 대해 간략히 기술한다.
2. 범위	SW 설계 명세에서 포함해야 하는 영역을 지정한다. SW 설계명세를 통해 SW 시스템의 전체적인 구조를 파악할 수 있고 구현이 가능한 범위까지 기술한다.
3. 참고문서	SW 설계 명세서를 작성하는 데 참고가 된 표준규약들과 계획단계, 요구단계에서 생성된 문서들을 기술하도록 한다.
4. 용어 및 약어정의	SW 설계 명세서에서 나오는 용어와 약어에 대해서 정의한다. 문서의 전문적인 용어는 자세하게 설명하여 전체내용을 이해하는 데 도움이 될 수 있도록 구성한다.
5. SW 설계 정보	
5.1 SW 설계 개체	SW 설계 개체는 설계의 요소(elements)이며 이 설계 요소는 구조적으로 혹은 기능적으로 다른 요소들과 구별되고 각각 다르게 이름을 붙인다. 설계 개체는 SW 요구 사항을 분할함으로써 얻어지는데 이러한 분할의 목적은 전체 시스템을 보다 작은 컴포넌트들로 나누고 다른 요소들과의 영향을 최소화 시킨 후 테스트하기 위함이다. 나누어진 컴포넌트를 기준으로 구현을 한다. 개체는 전체 시스템, 서브시스템, 데이터 저장, 모듈, 프로그램, 절차로서 존재한다. 설계를 분할하기 위해 요구된 이 개체의 개수와 타입은 시스템의 복잡도 혹은 설계의 기술, 또는 프로그래밍 환경과 같은 다양한 요소들에 의존한다. 각 개체들은 서로 다른 성질을 가지고 있기도 하고 서로 공통된 성질을 가질 수 있다. 각 설계 개체는 하나의 이름, 목적, 기능을 갖는다. 그리고 연계 혹은 공유데이터와 같은 관계 개체간의 공통 관계도 존재하는데, 개체간의 공통 특징은 설계 개체 속성에 의해 기술한다.
5.2 SW 설계 개체 속성	설계 개체 속성은 설계 개체의 특징이나 성질로서 각 개체에 대한 사실을 기술한다. 개체의 속성에 대한 기술은 요구사항에서 고려한 제약들과 개체가 어느 곳에 적용되어서 동작하는지에 대한 가정들을 포함한다. 즉 개체들이 활동할 수 있는 환경에 대해서 기술한다.

	개체의 속성 및 속성과 관련된 정보에 대 해서 아래와 같이 구분하여 기술할 수 있다. -개체명 (Identification) 각 개체의 이름, 다른 개체와 구별하고 속성을 대표하는 이름 -개체 타입 (Type) 각 개체의 타입에 대한 기술이다. 타입 속성은 개체의 특성을 기술한다. 예를 들어, 서브 프로그램, 모듈, 프로시저, 프로세스 혹은 데이터 스토어와 같은 종류의 이름으로 단순히 이름이 붙여진다 -목적 (Purpose) 각 개체들의 역할에 대해 기술한다. 이 속성은 각 개체의 생성을 위한 근본적인 이유를 제시하고 개체가 만들어 졌을 때를 위한 특수한 기능 혹은 수행 요구사항을 제시한다. 또한 각 개체 가 만족시켜야 하는 특정 요구 사항을 기술한다. -종속관계(Subordinates) 각 개체간의 종속관계를 기술한다. 종속관계는 각 개체 간의 관계를 규명한다. 이것은 수직적인 종속관계를 의미하며 이 정보를 통해 개체를 설계로부터 요구 사항으로의 추적에 이용된다. 또한 SW 를 분할하기 위한 상하 구조 관계를 파악하는데 이용된다. -의존관계 (Dependencies) 개체들 사이의 관계에 대한 기술이다. 의존 속성은 하나의 개체를 위한 관계 유무의 사용 혹은 요구를 기술한다. 구조도(structure charts), 데이터 흐름도(data flow diagrams), 혹은 전이도(transition diagram) 등으로 도식적으로 표현할 수 있다. 상호작용을 위한 개체의 초기화, 수행의 순서, 자원의 공유, 생성, 복제, 사용, 저장 혹은 삭제 등의 상호의존 관계에 대해 기술한다. -상호작용 (Interface) 개체들 사이의 상호작용 에 대한 기술이다. 상호 작용을 하는 방법 및 규칙을 기술한다. 상호작용의 방법은 개체 호출하는 방법, 매개변수, 공유 데이터 영역, 혹은 메시지를 통하여 통신하는 방법, 또는 내부 데이터를 직접적으로 접근하는 방법을 포함할 수 있다. 상호작용을 제어하는 규칙은 통신 프로토콜, 데이터 포맷, 수용 가능한 데이터, 각 값의 의미를 포함한다. 이 속성은 또한 입력 범위, 입출력의 의미, 각 입출력의 타입과 포맷 그리고 출력 에러 코드를 제공한다. -자원 (Resource) 설계에 대하여 개체에 의해 사용되는 요소에 대한 기술이다. 이 속성은 물리적인 외부 장치를 가리키는 것으로 SW 와 관련 있는 입출력 장치가 될 수 있다. 이 속성에는 어떤 자원을 획득할 때의 프로세스 시간, 버퍼 사용의 물리적인 크기 등을 기술할 수 있다. 이 속성은 잠재적인 경쟁과

	데드락 조건을 기술한다, 그에 더하여 자원의 관리 기능도 기술한다. -처리 (Processing) 개체의 기능을 수행하기 위해 개체에 의해 사용되는 규칙을 기술한다. 특정 기능을 수행하기 위한 알고리즘을 기술한다. 타이밍, 이벤트 혹은 프로세스의 순서, 프로세스 초기화의 전제조건, 이벤트의 우선순위, 처리레벨, 처리과정, 순환 및 순환을 종료하기 위한 기준이 포함된다. -데이터 (Data) 개체의 내부 데이터요소를 기술한다. 데이터 속성은 표현 방법, 초기값, 사용, 의미, 포맷 및 내부 데이터로서 수용 가능한 값을 기술한다. 데이터 사전의 형태로 기술할 수 있다.
6. SW 설계	개체 속성정보는 여러 방법으로 기술할 수 있고, 이러한 기술을 토대로 사용자는 시스템의 다양한 핵심포인트를 세부적으로 관찰할 수 있다. SW 설계를 다양한 관점에서 바라보고 기술하기 위한 방법은 분해 (Decomposition), 의존성 (Dependency), 상호작용 (Interface), 상세기술 (Detailed Description) 등이 있다. 이러한 설계 관점을 가지고 SW 를 표현하여 설계의 완성도를 높일 수 있게 된다.
6.1 분해 명세(Decomposition Description)	분해 명세(Decomposition Description) 방법은 SW 시스템을 설계 개체들로 나누고 개체들의 목적과 기능들을 기술하는 방법이다. 또한 이 방법은 상세명세(Detailed Description)에 참고한다. 설계 개체들은 특별한 기능을 수행하고 요구사항에 대한 추적이 가능해야 한다. 설계 담당자 및 유지보수 담당자는 이러한 목적을 가 지고 있는 설계 개체들을 식별하기 위해서 분해 명세를 사용한다. 설계 개체들은 개체들이 가지고 있는 속성이나 정보들의 공통된 특징을 묶기 위하여 그룹화될 수 있고 개체들의 표현에 대한 완벽성을 추구하기 위해 재검토되기도 한다. 이러한 표현을 위해 모듈 분해(Module Decomposition) 과 데이터 분해(Data Decomposition)을 서로 분리해서 표현할 수도 있다. 프로젝트 관리자는 분해명세를 참고하여 설계를 계획 하고 감시하고 통제하는 역할을 수행할 수 있다. 프로젝트 관리자는 SW 의 컴포넌트와 목적 그리고 기본적인 기능을 식별할 수 있는데, 이러한 설계 정보를 통해 다른 프로젝트의 비용과 인력, 그리고 일정에 대해서 서로 협력할 수 있어 개발노력을 덜어줄 수 있게 된다. 다이어그램으로 계층적으로 도식화하고 각각의 개체의 기능과 목적에 대한 기술은 자연어로 표현할 수 있다.
6.1.1 모듈 분해 (Module Decomposition)	모듈분해는 SW 를 구성하고 있는 모듈을 분해하는 방법이다. SW 를 구성하고 있는 요소나 하나의 기능을 수행하는 함수들이 모듈 분해의 개체가 될 수 있다.
6.1.2 동시성	SW 와 동시적으로 동작하는 개체들을 기술한다. 동시성 프로세스에는

프로세스 분해(Concurrent Process Decomposition)	시간과 동 기화가 포함되어 동작하는 개체이다. SW 가 실시간 시스템 속성을 만족하여야 한다면 시간과 동기화를 수행하는 주체들이 하나의 프로세스로 표현될 수 있다. 이러한 프로세스들을 분해하여 실 시간적 분해를 표현해야 한다.
6.1.3 데이터 분해(Data Decomposition)	SW 에서 사용되는 데이터에 대한 분해를 기술한다. SW 의 모듈이나 함수를 수행하는데 필요한 데이터를 표현한다.
6.2 의존성 명세(Dependency Description)	의존성 명세는 개체들 간의 관계를 기술한다. 개체의 종속적인 관계를 식별하고 개체들간의 응집도를 기술하고 필요한 자원에 대해서 식별한다. 설계대상 개체들 간의 상호활동을 위한 전략을 정의하고 시스템의 행동이 언제, 어디서, 어떻게 일어나는지를 쉽게 인지하기 위해 필요한 정보를 제공한다. 공유된 정보와 실행순서, 매개변수 연계와 같은 개체들 사이에 존재 하는 관계의 형태를 기술한다. 의존성 명세는 시스템이 요구사항과 설계 변화에 영향을 평가하기 위해서 어떻게 작동하는지에 대한 전체적인 그림을 제공한다. 이를 통해 유지보수 담당자는 시스템의 실패나 자원 장애의 원인이 되는 개체들을 고립시켜 시스템의 안전하고 원활한 작동에 도움을 줄 수 있다. 또한 먼저 개발되어야 하는 개체와 다른 개체에 의해 필요한 개체들을 구분하기 위한 시스템 통합 계획 수립에 활용할 수 있다. 이 방법은 통합 시험에 사용되어 통합 시험 케이스를 만드는 데에 활용할 수 있다. 동일한 개체 안에 있는 요소 사이의 관계를 최대화하여 개체 사이의 관계를 최소화하는 여러가지 방법이 있고 이러한 방법은 모듈 결합도를 낮추고 모듈 응집도를 높이는 작 용을 한다. 정형 명세 언어는 시스템의 기능과 데이터, 그들간의 상호작동, 입/출력, 그리고 다른 시스템의 측면을 기술한다. 설계개체 사이의 관계는 데이터 흐름도(Data Flow Diagram), 구조도(Structure Chart), 전이도(Transaction Diagram)로 표현할 수 있다.
6.2.1. 내부 모듈 의존성 (Inter-module dependency)	SW 의 내부에서 동작하는 모듈들간의 의존관계에 대해 기술한다. 독립적으로 수행하는 모듈및 의존관계에 있는 모듈에 대해서도 기술한다.
6.2.2. 내부프로세스 의존성 (Inter-process dependency)	프로세스 단위로 분해했던 개체들간의 의존관계에 대해서 기술한다. 내부모듈 의존성과 마찬가지 방법으로 프로세스간의 의존성에 대해서 기술 한다.
6.2.3 데이터 의존성(Data dependency)	데이터 간의 의존성 속성이 존재할 수 있다. 하나의 모듈이나 프로세스에서 변경한 데이터가 다른 모듈이 나 프로세스의 데이터 입력 값이나 출력 값에 영향을 준다면 데이터 의존성이 존재한다고 할 수 있다. 이 절에서는 이러한 데이터 의존성에 대해서 기술 한다.

6.3 상호용 (Interface) 설계기술	상호작용은 개체들에 의해서 제공되는 기능들을 설계 담당자, 구현 담당자시험 담당자에게 제공하기 위해 기술한다. SW 요구사항에서 제공하지 않은 외부연계에 대한 사항들을 자세하게 기술한다. 상호작용 개체의 개체명, 기능 및 속성들이 포함한다. 설계 담당자, 구현 담당자, 그리고 시험 담당자 사이의 협의사항을 결합하는 서비스를 제공하고 상세 설계 이전 필요한 정보들을 제공한다. 기존 프로젝트, 외부 자원(고객 포함) 그리고 다른 개발자에 의해서 이미 작성된 개체를 사용하는 경우에도 상호작용을 파악하고 기술한다. 개체 연계는 여러 사람에 의해서 개발된 개체들이 서로 부드럽게 통합되고 쉽게 유지보수 될 수 있도록 기술한다. 상호작용에 기술하는 사용자 인터페이스의 입출력과 같은 데이터에 기반한 개체들은 데이터 사전에 기술한다. 사용자에게 시각적으로 표현되어야 하고 고객이 시스템을 인지하는 방법을 세부적으로 기술한다. 이러한 시스템에는 기능모델, 동작 시나리오, 세부적인 도식 집합 그리고 상호작동 언어를 포함한다.
6.3.1 모듈 연계(Module Interface)	SW 의 모듈이 사용자나 외부장치와 연계 하는 방법을 기술한다.
6.3.2 프로세스 연계 (Process Interface)	프로세스가 외부 장치와의 연계를 통해 동작하는 순서와 서로 공유하는 정보들에 대 해서 기술한다. 또한 SW 가 HW 와 서로 통합되어 시스템이 동작되는 순서를 기술한다.
6.4 상세명세(Detailed Description)	설계 대상 개체의 세부사항들을 기술하는 것이다. 자료식별, 데이터, 데이터 처리등과 같은 속성 들도 포함한다. 상세명세는 구현에 앞서 프로그래머가 필요한 세부사항들을 제공한다. 단위시험 계획을 세우는데 활용할 수 있다. 설계 개체의 세부사항들을 기술하기 위해 사용되는 도구들이 많이 있다. 프로그램 설계 언어는 개체의 입력, 출력, 데이터, 그리고 알고리즘을 표현하기 위해서 사용되고, 개체의 논리적인 구조를 표현하기 위한 기술적인 방법은 구조적인 언어나 순서도 등의 도식적인 방법을 사용할 수 있다. 해당 SW 에 가장 적합한 설계방법을 사용하여 SW 설계 관점을 표현한다.
6.4.1 모듈 상세 설계 (Module Detailed Design)	모듈에 대한 세부적인 설계를 작성한다. 여러 개의 함수가 모듈을 구성하고 있다면 함수에 대한 세부적인 흐름을 설계한다. 순서도로 표현하거나 클래스 다이어그램을 이용해 모듈내의 함수들을 자세히 기술한다.
6.4.2 데이터 상세 설계(Data Detailed Design)	데이터 단위로 분해했던 구조를 데이터의 세부 단위로 나누어서 자세히 기술한다. 데이터가 가지는 형태나 표현방식 그리고 데이터가 모듈이나 함수에 어떤 정보를 제공해야 하는지에 대해 기술한다.

[SD-D-02] SW 단위시험 계획서

IEEE std. 829-2008 에 SW 단위시험 계획서에 포함되어야 할 내용들에 대한 최소요건들을 규정하고 있으며 , 이를 기반으로 본 가이드는 다음의 목차를 권고한다.

가. 목 차

나. 작성법

1. 목적	단위시험의 목적을 기술한다.
2. 범위	단위 시험적용 범위를 기술하며 단위시험 계획서 각 영역의 구성에 대해 기술한다.
3. 참고문서	단위시험 계획서 작성시 참고한 법규, 기준 및 인용한 참고문서를

	기술한다.
4. 용어정의 및 약어	단위시험 계획서에서 사용된 용어 및 약어의 정의를 기술한다. 용어정의는 IEC, IEEE 등 에서 사용되는 표준용어를 사용하여 기술하여야 하며 적절한 표준용어가 없을 경우 일반 IT 에서 사용하는 표준용어로 대체할 수 있다.
5. 시험대상	시험대상을 기능 및 성능 엔터티로 구분하였을 때 단위시험은 설계명세서를 근간으로 기능 위주로 구분하였을 경우 정보를 추출하여 시험하고자 하는 대상을 기술한다. 시험대상과 그에 따른 시험항목이 정해지면 각각의 시험항목에 시험 ID 를 부여하여 시험 사례와 일치해야한다. 시험대상 및 시험항목이 정해지면 이들을 각각의 별도 섹션으로 구분하여 기술한다.
6. 시험항목	시험대상과 시험항목은 대분류/중분류 소분류 관계로 이전 단계에서 분류한 시험대상에 대한 구체적인 시험항목을 기술한다.
7. 접근법	단위시험 방법에 대한 접근법을 기술한다. 예를들면 설계명세서로부터 기능을 추출하고 추출된 기능을 세부 기능으로 분류하여 시험항목을 만든다. 이때 시험사례에 따라 입력데이터, 예상결과 및 통과기준을 정하고 결과를 확인하고 최종 분석하는 일련의 단계를 그림으로 도식화 하고 이에 대한 설명을 기술한다. 기능의 주요특징이 누락되는 일이 없도록 주의를 하여야 하며 시험 시 고려하여야 할 특성과 시험 시 고려하지 않아도 될 특징(또는 시험불가)을 잘 구분해서 기술한다.
8. 시험통과 및 실패기준	단위시험 환경 하에서 선정된 시험항목을 실행하였을 경우 예상출력과 허용오차 범위 안에 있으면 시험통과 기준을 만족하는 것이고 예상결과 범위 밖이면 실패로 판단한다.
9. 시험환경 요건	상위시스템, HW 자원 및 SW 자원의 목록(장비명, 수량, 사양, 기능 등)을 테이블 형태로 기술한다. 이때 이들의 검•교정 날짜가 누락되지 않도록 주의한다. 상위시스템, HW 자원 및 SW 자원에 대한 각각의 설명과 함께 시험환경 구성도를 알기 쉽도록 그림으로 도식화한다. 시험환경의 인터페이스 오류를 방지하기 위하여 연결장비 및 케이블에 대한 사양도 빠지지 않도록 주의한다.
9.1 SW 자원	SW 환경, 운영체제, 에디터, 디버거, 컴파일러/어셈블러, 링커 등 소프트웨어 자원을 열거한다. 특히 SW 시험도구를 사용할 경우 시험 Harness, 시험드라이버, 시험 스터브, 시험 사례의 자동생성, 시험 스크립트 등 도구가 지원하는 특징을 기술한다.
9.2 HW 자원	HW 장비에 대한 시스템, 호스트 및 타겟 환경 등을 기술한다.
9.3 시험장비 및 시험장비 구성	단위시험에 사용되는 SW 자원 및 HW 자원을 종합한 시험장비들을 기술한다.
10. 책임 및 권한	단위 시험과 관련된 개발팀의 역할과 책임, 검증팀의 역할과 책임 그리고 시험 중 오류가 발생하였을 경우 이를 해결하는 과정 및 책임사항들을 책임사항 및 역할 위주로 간략히 기술한다.
11. 시험일정	시험일정은 시험계획서 작성부터 시작하여 시험절차서,

	시험수행, 시험결과 분석 및 시험 결과 보고서 작성까지의 일정을 기술하는데 시험일정과 연계하여 시험 설 계 및 시험 계획서는 설계단계에서, 시험사례생성 및 시험절차서 작성, 시험실행은 구현단계에서 실행하는 일정에 맞춘다. o 업무: 시험을 위해 필요한 세부업무를 기술 o 선행업무: 해당업무의 선행업무를 기술 o 필요기능: 해당업무를 수행하기 위해 요구되는 기능을 기술 o 예정시간: 해당업무 수행 예정시간을 기술 o 완료시기: 해당업무의 완료 시기를 기술
12. 계획의 승인	계획서 자체에 대한 공식적인 검토와 그 계획을 승인해야 할 개인의 목록을 규정하고, 그 승인을 받는다.

[SD-D-03] SW 통합시험 계획서

IEEE std. 829-2008 에 SW 통합시험 계획서에 포함되어야 할 내용들에 대한 최소요건들을 규정하고 있고 이를 참고하여 본 가이드는 다음의 목차를 권고한다.

가. 목 차

나. 작성법

1. 목적	SW 통합시험의 목적을 기술한다.
2. 범위	SW 통합시험 적용 범위를 기술하며 SW 통합시험 계획서 각 장의 구성에 대해 기술한다.
3. 참고문서	SW 통합시험 계획서 작성 시 참고한 법규, 기준 및 인용한 참고문서를

	기 술 한다.
4. 용어정의 및 약어	SW 통합시험 계획서에서 사용된 용어를 기술한다. 용어정의는 IEC 또는 IEEE 등의 SW 분야에서 사용되는 표준용어를 기술하여야 하며 적절한 표 준용어가 없을 경우 일반 IT 에서 사용하는 표준용어로 대체할 수 있다.
5. 시험 대상	SW 와 SW 의 통합을 위해서는 SW 설계명세서를 토대로 독립적으로 실행가능한 부분과 이질적으로 종속관계에 있는 통합대상을 식별한다. 이를 근간으로 그림 1 과 같이 전체적으로 트리구조를 그려서 SW 통합 시나 리오를 작성한다. 이때 그림으로 통합되어 가는 부분을 실선으로 나타내어(또는 트리구조 등 다른 방법으로 구분) 구분한다. SW 통합 시험대상과 그에 따른 SW 통합 시험항목이 정해지면 각 각의 SW 통합시험항목에 시험 ID 를 부여하여 통합시험케이스가 어떻게 병합되어 가는지를 시험 ID 만 보아도 쉽게 알 수 있도록 조합의 형태로 표기하여야 한다. 이때 역으로 통합시험항목에 해당되는 설계사양서 (구조명세서 부분)의 해당 색션 또는 설계사양 ID(ex, SWD_XXX)도 부여한다. 시험대상 및 시험항목이 정해지면 이들을 각각의 별도 섹션으로 구분하여 기술한다. 시험대상은 표 1 과 같이 기술한다. SW 통합시나리오에 따라 SW 통합이 합쳐지는 과정을 알 수 있도록 작성한다. **그림 1. SW 통합구조(예)** **표 1. SW 통합시험 대상** 구분 | 통합된 모듈 | | | 주 1) 중첩되는 부분은 별도로 표시한다. 주 2) 상기의 표 대신 통합 시나리오를 트리구조로 대체 표기할 수 있다.

6. 시험 항목

통합되는 모듈의 개수만큼 표 2 와 같이 시험항목을 나열한다.

표 2. 통합 시험 항목

통합시험 ID	통합시험 항목 ID	기능설명

7. 통합 방안

SW 통합은 다음과 같은 방법으로 수행한다.
SW 와 SW 의 통합은 그림 2 와 같이, 통합 시험 의 시험 항목 식별, 시험 사례 선정, 시험 수행 별로 관련된 시험 작업들을 수행한다.
그림 2 의 시험 항목 식별은 설계 명세서를 토대로 통합 시험의 대상이 되는 통합 대상을 식별하고 각 통합 대상별로 수행되어야 하는 통합 시험 항목을 식별 한 후, 식별한 시험 항목이 소스코드에서 매핑되는 시험 항목의 위치를 선정하는 단계이다. 그림 2 의 입력과 예상출력으로 이루어진 시험 사례 선정에서는 소스코드에서 식별 한 시험 항목의 위치를 입력으로 선정하고, 이 입력에 통과 및 실패 기준을 적용하여 예상 출력을 결정한다.
그림 2 의 시험 수행에서는 각 시험 사례를 수행하면서 모니터링되는 결과와 예상 출력을 비교하여 시험을 수행한다.

그림 2. 통합 시험 방법

시험 통과 및 실패 기준
SW 통합시험 환경 하에서 선정된 시험항목을 실행하였을 경우 예상출력과 허용오차. 범위 안에 있으면 시험통과 기준을 만족하는 것이고 예상결과 범위 밖이면 실패로 판단한다. 시험통과와 실패 기준은 표 3 과

<table>
<tr><td></td><td>같이 기재한다.

표 3. SW 통합시험 통과/실패 기준
<table>
<tr><th>구분</th><th>기준</th><th>비고</th></tr>
<tr><td>Pass</td><td>시험결과가 시험항목에 기재된 예상결과와 일치하는 경우 ‘Pass’로 기재함
시험 수행횟수가 1 회 이상으로 지정된 경우, 모두 성 공하여야 ‘Pass’로 기재함</td><td></td></tr>
<tr><td>Fail</td><td>‘Pass’ 이외의 결과는 ‘Fail’ 처리 후 사유 기재
시험 수행횟수가 1 희 이상으로 지정된 경우, 1 회 이상 Fail 이 발생하면 시험결과를 ‘Fail’로 기재하고 해당 시험 을 중단함</td><td></td></tr>
</table></td></tr>
<tr><td>8. 중단 및 재개 기준</td><td>SW 시험이 중단되어야 할 조건 및 재개 조건을 기술 한다.
통합 시험 중 작성 되어야 할 문서
통합 시험 조직은 시험 기간 동안 다음과 같은 문서들을 작성하여 통합 시험 종료 후 형상관리 조직에 이를 제출한다.
o 통합 시험 계획서
o 통합 시험 절차서
o 통합 시험 결과 보고서</td></tr>
<tr><td>9. 시험환경 요건</td><td>상위시스템, HW 자원 및 SW 자원에 대한 장비들의 목록(장비명, 수량, 사양 및 기능)을 테이블 형태로 기술한다. 이때 이들의 검·교정 날짜가 누락되지 않도록 주의한다. 상위시스템, HW 자원 및 SW 자원에 대한 각각의 설명과 함께 시험환경 구성도를 알기 쉽도록 그림으로 도식화 한다. 시험환경의 인터페이스 오류를 방지하기 위하여 연결장비 및 케이블에 대한 사양도 빠지지 않도록 기술한다.</td></tr>
<tr><td>9.1 SW 자원</td><td>SW 환경, 운영체제, 에디터, 디버거, 컴파일러/어셈블러, 링커 등의 소프트웨어 자원을 열거한다. 특히 SW 시험도구를 사용할 경우 테스트 Harness, 테스트드라이버, 테스트 스터브, 테스트 케이스의 자동생성, 테스트 스크립트 등 도구가 지원하는 특징을 기술한다.</td></tr>
<tr><td>9.2 HW 자원</td><td>HW 장비에 대한 시스템, 호스트 및 타겟 환경 등을 기술한다.</td></tr>
<tr><td>9.3 시험장비 및 시험장비 구성</td><td>시험에 사용되는 SW 자원 및 HW 자원을 종합한 시험장비들을 기술한다.</td></tr>
<tr><td>10. 위험 및 가정사항</td><td>통합시험을 수행하는데 있어 요구되는 개발 산출물이 지연될 경우, SW 관 리 계획서에 명시된 정책에 따라 관리한다. (예산. 인력, 예기치 못한 상황 등 기 술)
‘통합시험이 제대로 이루어 지지 않을 경우 제 3 전문가를 활용’ 등의 문구를 기술한다.
‘통합시험 수행에 있어 지연이 발생할 경우, 인력을 더 투입한다’ 등의 문구를 기술한다.
‘추가의 새로운 통합시험 요구가 발생될 때 인력을 더 투입한다.’ 등의</td></tr>
</table>

	문구를 기술한다.
11. 책임 및 권한	통합 시험과 관련된 개발팀의 역할과 책임, 검증팀의 역할과 책임 그리고 통합시험 중 오류가 발생하였을 경우 이를 해결하는 과정 및 책임사항들을 책임사항 및 역 할 위주로 간략히 기술한다.
12. 시험일정	시험일정은 시험계획서 작성부터 시작하여 시험절차서, 시험수행, 시험결과 분석 및 시험 결과 보고서 작성까지의 일정을 기술하는데 시험일정과 연계하여 시험 설 계 및 시험 계획서는 설계단계에서, 시험사례생성 및 시험절차서 작성, 시험실행은 통합시험단계에서 실행하는 일정을 기준에 맞게 일치시킨다. ㅇ 업무: 시험을 위해 필요한 세부업무를 기술한다. ㅇ 선행업무: 해당업무의 선행업무를 기술한다. ㅇ 필요기능: 해당업무를 수행하기 위해 요구되는 기능을 기술한다. ㅇ 예정시간: 해당업무 수행 예정시간을 기술한다. ㅇ 완료시기: 해당업무의 완료 시기를 기술한다.
13. 계획의 승인	이 장에서는 계획서 자체에 대한 공식적인 검토와 그 계획을 승인해야할 개인의 목록을 규정하고 관리자의 승인을 받는다.

[SD-D-04] SW 코딩 매뉴얼

코딩 규칙은 언어별로 차이가 있으므로 여기에서는 JAVA 언어를 기준으로 코딩 규칙을 설명한다.

가. 목 차

나. 작성법

1. 개요	SW 코딩 표준 매뉴얼의 개요를 기술한다.
2. 목적	SW 표준 메뉴얼의 목적을 기술한다. 언어 및 산업계 표준이 있으면 해당 기준에 맞게 코딩 표준을 정한다.
3. 소스파일 기본	파일이름, 확장자, 소스파일의 인코딩, 공백문자, 특수 문자 등에 대한 가이드라인을 제시한다.
4. 소스 파일 구조	라이선스, 패키지 문, 임포트문, 클래스 선언 등에 대한 내용을 기술한다.
5. 포멧팅	중괄호, Line-Wraping, 공백처리(공백 라인, 공백문자, 변수 정렬 등), 기타(들여쓰기, 변수선언, 라인당 글자수 등) 등에 대해 기술한다.
6. 네이밍	형식(패키지, 클래스, 메소드, 변수, 상수 등)에 대해 기술한다.
7. 주석문	주석문 작성법(구조, 띄어쓰기, 특수문자, 공백, 주석 내용 등)에 대해 기술한다.

[SD-V-01] SW 설계 안전성분석 보고서

SW 에 설계사항에 대한 안전성 분석 보고서의 작성과 관련하여 안전성분석 보고서에 포함되어야 할 내용을 다음과 같이 권고한다.

가. 목 차

나. 작성법

1. 목적	안전성분석의 수행 목적을 정의한다.
2. 범위	안전성분석의 수행 범위를 정의한다.
3. 참고 문서	안전성분석을 위해서 사용된 근거법령, 근거기준, 설계문서, 기타 문서들을 기술한다.
4. 용어 정의 및 약어	안전성분석 보고서에 사용된 어떤 용어 및 약어를 정의하고 설명한다.
5. 시스템 개요	다음과 같은 내용을 기술한다. 상위수준의 시스템 구조를 기술한다. 시스템, 하위시스템, 그리고 연계관계 등의 기능을 기술한다. SW 구조설계의 내용을 기술한다.
6. 안전성	다음과 같은 내용을 기술한다.

분석기법	SW 안전성분석에 사용된 입력자료를 기술한다. SW 안전성분석에 사용된 기법을 기술한다. SW 안전성분석에 사용된 절차를 기술한다. SW 안전성분석에 사용된 도구를 기술한다.
7. 안전성 분석결과	다음과 같은 안전성분석의 결과를 기술한다. SW 설계 적합성분석의 결과를 기술한다. SW 설계 위험원분석의 결과를 기술한다.
8. 안전성 검토	안전성분석의 결과를 시스템 설계조직, 시스템 안전성분석조직 및 소프트웨어 설계조직과의 공식적인 회의를 통해서 SW 설계명세서에 포함된 위험원의 수정 및 보완의 결과를 기술한다.
9. 승인	안전성보고서 자체에 대한 공식적인 검토와 승인해야 할 개인의 목록을 규정하고 서명을 받는다.

[SC-V-01] SW 코드 안전성분석 보고서

SW 에 코드에 대한 안전성 분석 보고서의 작성시 포함해야 할 내용에 대한 최소요건을 다음과 같이 권고한다.

가. 목 차

나. 작성법

1. 목적	안전성분석의 수행 목적을 정의한다.
2. 범위	안전성분석의 수행 범위을 정의한다.
3. 용어 정의 및 약어	안전성분석 보고서에 사용된 어떤 용어 및 약어를 정의하고 설명한다.
4. 참고 문서	안전성분석을 위해서 사용된 근거법령, 근거기준, 설계문서, 기타문서들을 기술한다.
5. 시스템 개요	상위수준의 시스템 구조를 기술한다. 시스템, 하위시스템, 그리고 연계관계 등의 기능을 기술한다. SW 구현코드의 내용을 기술한다.
6. 안전성 분석기법	SW 안전성분석에 사용된 입력자료를 기술한다. SW 안전성분석에 사용된 기법을 기술한다.

	SW 안전성분석에 사용된 절차를 기술한다. SW 안전성분석에 사용된 도구를 기술한다.
7. 안전성 분석결과	SW 코드 적합성분석의 결과를 기술한다. SW 코드 위험원분석의 결과를 기술한다.
8. 안전성 검토	안전성분석의 결과를 시스템 개발조직, 시스템 안전성분석조직 및 소프트웨어 설계조직과의 공식적인 회의를 통해서 SW 코드에 포함된 위험원 의 수정 및 보완의 결과를 기술한다.
9. 승인	안전성보고서 자체에 대한 공식적인 검토와 승인해야 할 개인의 목록을 규정해야 하고 서명을 받는다.

[SI-D-01] SW 단위시험 보고서

본 지침은 IEEE std. 829-2008 을 참고하여 SW 단위시험 보고서에 포함되어야 할 내용의 최소요건을 다음과 같이 권고한다.

가. 목 차

나. 작성법

1. 목적	단위 시험의 목적을 기술한다.
2. 범위	단위 시험적용 범위를 기술하며 단위시험 계획서 각 장의 구성에 대해 기술한다.
3. 참고문서	단위시험 계획서 작성시 참고한 법규, 기준, 인용한 참고문서를 기술한다.
4. 용어 정의 및 약어	단위시험 계획서에서 사용된 용어 및 약어의 정의를 기술한다.
5. SW 단위시험 환경	상위시스템, HW 자원 및 SW 자원에 대한 장비들의 목록(장비명, 수량, 사양 및 기능)을 테이블 형태로 기술한다. 이때 이들의 검정 및 교정 날짜가 누락되지 않도록 주의한다. 상위시스템, HW 자원 및 SW 자원에

	대한 각각 의 설명과 함께 시험환경 구성도를 알기 쉽도록 그림으로 도식화 한다. 시험환경의 인터페이스 오류를 방지하기 위하여 연결장비 및 케이블에 대한 사양도 빠지지 않도록 주의한다. 단위시험계획서의 시험환경 요건을 인용한다.
6. SW 단위시험결과	주어진 시험환경에서 수행한 시험결과를 기능, 비기능(성능) 등을 구분하여 단위 시험결과를 기술한다.
7. SW 단위시험 결과분석	단위시험결과를 통계적으로 분석하여 시험결과를 알기 쉽도록 도표 또는 그래프 하여 최종적으로 정리한 내용을 기술한다. 시험도구에 의하여 자동생성된 시험결과도 여기에 첨부한다.
8. SW 단위시험 결과요약 및 권고사항	전체의 시험결과를 요약하고 권고사항을 기술한다.
9. 결과의 승인	이 절에서는 결과 보고서 자체에 대한 공식적인 검토와 그 결과를 승인해야 할 개인의 목록을 규정하고, 그 승인을 받는다.

[SI-D-02] SW 통합시험 보고서

본 지침은 IEEE std. 829-2008 을 참고하여 SW 통합시험 보고서에 포함되어야 할 내용에 대한 최소요건을 다음과 같이 권고한다.

가. 목 차

나. 작성법

1. 목적	SW 통합시험의 목적을 기술한다.
2. 범위	SW 통합시험 적용 범위를 기술하며 SW 통합시험 계획서 각 색션 의 구성에 대해 기술한다.
3. 참고문서	SW 통합시험 계획서 작성 시 참고한 법규, 기준 및 문서를 기술한다.
4. 용어 정의 및 약어	SW 통합시험 계획서에서 사용된 용어를 기술한다. 용어정의는 IEC 또는 IEEE 등의 SW 에서 사용되는 표준용어를 기술하여야 하며 적절 한 표준용어가 없을 경우 일반 IT 에서 사용하는 표준용어로 대체할 수도

<table>
<tr><td></td><td>있다.</td></tr>
<tr><td>5. SW 통합시험 환경</td><td>상위시스템, HW 자원 및 SW 자원에 대한 장비들의 목록(장비명, 수량, 사양 및 기능)을 테이블 형태로 기술한다. 이때 이들의 검.교정 날짜가 누락되지 않도록 주의한다. 상위시스템, HW 자원 및 SW 자원에 대한 각각 의 설명과 함께 시험환경 구성도를 알기 쉽도록 그림으로 도식화 한다. 시험환경의 인터페이스 오류를 방지하기 위하여 연결장비 및 케이블에 대한 사양도 빠지지 않도록 기술한다.</td></tr>
<tr><td>SW 자원</td><td>소프트웨어 환경, 운영 체제, 에디터, 디버거, 컴파일러/어셈블러, 링커 등의 소프트웨어 자원을 열거한다. 특히 SW 시험도구를 사용할 경우 테스트 Harness, 테스트드라이버, 테스트 스터브, 테스트 케이스의 자동생성, 테스트 스크립트 등 도구가 지원하는 특징을 기술한다.</td></tr>
<tr><td>HW 자원</td><td>HW 장비에 대한 시스템, 호스트 및 타겟 환경 등을 기술한다.</td></tr>
<tr><td>시험장비 및 시험장비 구성</td><td>모듈시험에 사용되는 SW 자원 및 HW 자원을 종합한 시험장비들을 기술한다.
소프트웨어 통합시험 계획서에 기술된 시험환경을 그대로 인용한다.</td></tr>
<tr><td>6. SW 통합시험결과</td><td>SW 통합시험 절차서에 따라 기술된 각 통합시험 시나리오별 결과를 기술 한다. 시험결과는 표 1 과 같은 포맷으로 기술한다.

표 1. 통합시험 결과
<table><tr><th>Test ID</th><th>Test Case</th><th>Input</th><th>Sub Instance</th><th>Expected Output</th><th>결과값</th></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>
통합시험 결과에 따라 설계명세서가 변경이 필요할 경우 형상관리절차에 따라 변경한다.</td></tr>
<tr><td>7. SW 통합시험 결과분석</td><td>시험결과를 통계적으로 분석하여 시험결과를 알기 쉽도록 표 2 와 같이 도표 또는 그래프로 기술한다.
표 2. 통합시험결과 분석
<table><tr><th colspan="2">통합시험 항목</th><th rowspan="2">시험사례의 수</th><th rowspan="2">Pass 개수</th><th rowspan="2">Fail 갯수</th></tr><tr><th>통합된</th><th>각 모듈명</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td colspan="2">합 계</td><td></td><td></td><td></td></tr></table>시험도구에 의하여 자동 생성된 시험결과도 여기에 첨부한다.
SW 통합시험 결과요약 및 권고사항 전체의 통합 시험결과를 요약하고 권고사항을 기술한다.</td></tr>
<tr><td>9. 보고의 승인</td><td>이 장에서는 보고서 자체에 대한 공식적인 검토와 보고서를 승인해야할 개인의 목록을 기술하고, 그 승인을 받는다.</td></tr>
</table>

[SI-V-01] SW 시험 안전성분석 보고서

본 가이드는 안전성분석 보고서에 포함되어야 할 내용에 대한 최소요건을 다음과 같이 권고한다.

가. 목 차

나. 작성법

1. 목적	안전성분석의 수행 목적을 정의한다.
2. 범위	안전성분석의 수행 범위를 정의한다.
3. 용어 정의 및 약어	안전성분석 보고서에 사용된 어떤 용어 및 약어를 정의하고 설명한다.
4. 참고 문서	안전성분석을 위해서 사용된 근거법령, 근거기준, 설계문서, 기타문서들을 기술한다.
5. 시스템 개요	상위수준의 시스템 구조를 기술한다. 시스템, 하위시스템, 그리고 연계관계 등의 기능을 기술한다. SW 구현코드의 내용을 기술한다.
6. 안전성 분석기법	SW 안전성분석에 사용된 입력자료를 기술한다. SW 안전성분석에 사용된 기법을 기술한다. SW 안전성분석에 사용된 절차를 기술한다.

	SW 안전성분석에 사용된 도구를 기술한다.
7. 안전성 분석결과	SW 시험 적합성분석의 결과를 기술한다. SW 시험 위험원분석의 결과를 기술한다.
8. 안전성 검토	안전성분석의 결과를 시스템 개발조직, 시스템 안전성분석조직 및 소프트웨어 설계조직과의 공식적인 회의를 통해서 SW 코드에 포함된 위험원 의 수정 및 보완의 결과를 기술한다.
9. 승인	안전성보고서 자체에 대한 공식적인 검토와 승인해야 할 개인의 목록을 규정해야 하고 서명을 받는다.

Appendix 2. 안전대책 기술서(Safety Case)

가. 개요

안전 대책기술서는 프로젝트에 의해 제시된 시스템 혹은 장비의 안전성에 대한 개요를 입증하기 위한 서류이다. 안전대책기술서는 가능하다면 모든 위험 요인 확인 분석, 안전 요구 사항, 위험성 평가 증거 자료를 포함한 추가 서류로 제출할 수도 있다.

안전대책기술서의 설명에 있어서, 시스템의 항목은 작업 시스템, 장비, 증거 자료, 작업 구성물, 작업 방법을 포함하는 안전과 관련된 모든 프로젝트에 즉각적인 적용이 가능해야 한다.

나. 지침

1) 책임 사항

프로젝트 관리자는 안전대책기술서의 준비와 지속, 그리고 안전검토그룹의 제출에 대해 책임을 갖는다. 프로젝트 관리자는 이 역할을 프로젝트 안전 매니저에 위임하는 것은 가능하지만 총체적인 책임은 프로젝트 관리자의 몫이다.

안전 승인 단체는 안전대책기술서의 승인에 대해 책임이 있다.

2) 제출

안전대책기술서는 검열과 승인을 위해 적합한 안전 승인 단체에 제출되어야 한다. 안전 인증서는 승인절차 이후 발행된다. 제출 과정은 시스템검토위원회와 상호 연관되도록 작업 지시 사항에 명시되어야 한다.

안전대책기술서는 개정이 가능하며 프로젝트 주기 동안 수정이 가능하다.

안전대책기술서는 설치와 인도 단계로 가기에 앞서 처음에 면밀히 검토한 다음 제출한다.

안전대책기술서는 설치와 인도 단계 동안 수정된다. 만약 더 앞선 안 전 관련 정보가 있다면 운영과 유지 보수 단계 전에 통용되고 승인된다.

안전대책기술서는 안전 관련 정보가 얼마나 확보 가능한가에 따라 다른 주기 단계를 따라 수정되어야 한다.

3) 일반 요구 사항

안전대책기술서는 안전 요구 사항에 따르는 설계와 시스템의 장비의 확 실성을 입증함에 그 목적을 둔 증거 사항들을 형식적으로 기술한 것이다. 그 사항들은 다음과 같다.

가) 안전 요구 사항에 적합할 것
나) 시스템 설계, 개발, 분석의 과정의 적절한 적용
다) 시스템은 안전 요구 사항을 충족시켜야 하며 사용하기에 안전할 것

안전대책기술서는 프로젝트팀의 다음과 같은 사항들을 확실히 명시해야 한다.

가) 모든 예측 가능한 위험원들을 명백히 밝힐 것
나) 사고 발생 시 ALARP 원칙에 따라 위험원을 확인하고 위험성의 확 인을 위한 적절한 행동을 취할 것

안전대책기술서는 시스템에서 발생할 수 있는 가능한 최소의 위험 (ALARP 원리)이 발생했을 경우를 명시해야 한다. 독립형과 상급 시스템 또는 과정의 일부 양 쪽 모두 시스템과 관련되는 위험들을 최소로 하게 하기 위한 합리적인 행동이 취해졌음에 대한 보증을 제공해야 한다. (ALARP 원리) 그것은 적절한 안전 요구 사항에 의해 명시된 것과 안전 요구 사항에 의해 예측 가능한 위험원들을 억제하고 제거하기 위한 모든 합리적인 단계들을 보여야 한다.

안전대책기술서는 어떠한 해결 불가능한 위험원이나 안전 요구 명세와 안전계획과 일치하지 않는 것에 대해서 명시해야 한다.

안전대책기술서는 목적을 이루기 위해 상호 작용하는 하드웨어, 소프트웨어, 컴퓨터 처리, 인력의 종합적인 전체 시스템과 관련된 안전성도 고려해야 한다. 또한 그것은 대중과 환경적 측면에서 인력에 대한 위험원이 시스템에 미치는 영향과 같은 시스템 외부에서의 안전 문제들을 고려해야 한다

안전대책기술서는 위험원목록과 같은 다른 프로젝트 참고 문헌에 자세하게 기술되며 수준 높은 정보가 기입되어 있어야 한다. 어떤 참고 문헌은 규격에 맞추어 문제화되고 배치한다. 참고 문헌은 참고 문헌 번 호, 제목, 판 번호, 날짜를 포함하여 정확하고 포괄적으로 기술한다.

안전대책기술서는 그 합리화 과정을 뒷받침하는 참고 증거를 제시해야 한다. 그 증거는 비록 안전대책기술서가 위험원목록과 안전도 평가, 안전 감사 보고서에 크게 의존한다 할지라도 많은 출처로부터 인용할 수 있다.

안전대책기술서는 다른 프로젝트 참고 문헌으로부터의 정확한 정보를 인용할 수도 있다.

비록 안전대책기술서가 시스템이나 장비의 안전성에 대해 전문가에 의해 우선적으로 사용된다고 하더라도, 안전대책기술서는 많이 읽혀지고 있으며 이것은 안전대책기술서 작성시 고려해야한다. 안전대책기술서의 잠재적 독자들은 다음과 같다.

가) 프로젝트 수행자
나) 전문가
다) 시스템검토위원회의 안전성활동 참가자들
라) 사무국
마) 안전성 감사자, 평가자

다. 안전대책기술서의 내용

안전대책기술서는 하단에 열거된 바와 같은 항목을 포함해야 하며 이후의 항목에도 열거 되어야 한다. 항목이 적용되지 않을 때는 "적용 불가" 라는 문구가 삽입하여야 하며 항목을 삭제하여서는 안된다.

1 개요
2 도입
3 시스템 정의
4 안전 요구 사항
5 품질 관리
6 안전 관리
7 기술적 안전 평가
8 관련된 안전대책기술서
9 결론

라. 항목별 작성 내용

1 개요	개요에서는 안전대책기술서에 포함된 중요한 정보를 요약한 것으로 다음 항목들을 포함하여야 한다. 가) 시스템, 구조 혹은 다른 프로젝트의 출력물, 그들의 목적, 기능성, 위치에 대한 간결한 기술 나) 안전 설계와 개발 절차 개요 다) 평가, 감사 절차 개요 라) 시험, 운영상 경험의 개요 마) 확보 증거와 미해결 위험원과 관련한 현재의 안전 상태 바) 도입 안전대책기술서의 계획 의도와 목적, 범위, 구성을 명시한다.
2 도입	
3 시스템 정의	안전대책기술서는 발생된 안전 문제의 이해를 위해 전달되어야 할 시스템, 구조와 다른 프로젝트의 개괄을 제공한다. 안전대책기술서는

	감사를 거친 항목들인 목적, 기능성, 구조, 설계 운영과 관련된 참고문헌을 포함한다. 다음 사항을 포함한다. 가) 물리적 위치를 포함한 시스템의 명시 나) 다른 시스템, 서비스, 시설에 대한 전제를 포함한 시스템의 주변, 경계에 대한 명시 다) 계약이 종료되면 구성 세부 시스템의 확인, 세부 시스템의 안전대책기술서에 대한 참고 개관은 시스템의 정상 운영을 포함해야 하며 운영 직원의 규칙을 포함하고 정형화 유지와 기술의 원조를 요구한다. 정상 운영이 다음에 의존하는 정도를 명시한다. 가) 자동화 나) 절차 다) 판단 라) 외부 장비, 서비스, 시설 시스템 특징, 대체 시스템 방식, 비교 운영 절차의 항목에서 시스템의 비 정상 운영이 명시되어야 한다. 아래 항목들과 관련된 계획들이 지속적인 작업과 모든 위험원의 증가와 관련된 안전 관련성에 따라 표시한다. 가) 컴퓨터 하드웨어와 소프트웨어를 포함한 기계적, 전기적 시스템의 기능장애 나) 인간 실수를 포함한 절차상의 고장 다) 화재, 파괴 등으로 인한 외부적인 긴급 상황 안전대책기술서 적용에 대한 시스템의 구성은 정확하게 명시되어야 한다. 안전대책기술서는 안전계획에 있어 요구되는 규정에 참고되어 시스템의 효과적으로 구성 관리되고 제어되도록 변화시키는 것에 대하여 적합한 것임을 입증해야 한다.
4 안전 요구 사항	모든 안전요구사항은 정확하게 명시되어야 하고, 참조문헌으로 사용되어서는 안 된다. 단지 안전요구사항 명세서에 명시된 것만 포함되어야 한다. 프로젝트에 영향을 주는 각 요구사항을 어떻게 지정하는지에 대한 요구사항의 안전 관계에 대한 논의 가 포함되어야 한다. 모든 가정은 명백히 공표 되고 정당화되어야 한다. 안전대책기술서는 안전요구사항에 관련이 있다는 증거를 입증해야 한다.
5 품질 관리	효과적인 기술을 위한 전제 조건으로써 안전대책기술서는 시스템, 세부 시스템이나 장비의 품질에 대해 주기 기간 동안 효과적인 품질 관리 시스템 (QMS: Quality Management System)에 의해 제어되고 유지되어야만 한다.
6 안전 관리	1) 도입 가) 안전대책기술서는 프로젝트 수행의 측면에서 어떻게 안전 관리를 할 것인지에 대해 명시해야 한다. 그것은 안전계획에 명시된 활동으로 언급,

	요약 될 수 있다. 안전계획과 첨부된 증거는 이러한 활동이 적절하게 수행되고 계획되었다는 것을 보이는 것이다. 나) 위험원 목록은 계획 활동의 수행을 위한 증거의 주된 출처이다. 다) 다음과 같이 안전 관리의 쟁점이 기술되었다. 규칙과 책임 사항 안전 주기 안전 규정 계약인 관리 안전도 제어 구성 관리 프로젝트 안전 훈련 2) 규칙과 책임 사항 가) 안전대책기술서는 안전계획에 명시된 규칙을 수행하는 프로젝트에 있어 안전 관리 직원의 중요성을 입증할 수 있는 증거를 제공할 수 있어야 한다. 나) 안전대책기술서에 명시된 능력과 경험의 참고가 가능한 중요한 안전 인원을 임명을 한다. 3) 안전 주기 안전계획에 명시된 안전 주기가 일반적 주기와 다르다면 프로젝트 수행 기간 동안 본 계획의 안전 주기를 지킨다. 4) 안전 규정 안전대책기술서는 안전계획의 절차와 규정에 따른다는 증거를 제공해야 하며 어떤 불일치성이라도 정당화해야 한다. 5) 계약인 관리 안전대책기술서는 계약인의 안전계획에서 명확하게 서술된 계약인의 업무를 보여준다. 6) 안전 제어 안전대책기술서는 안전계획에 명시된 안전 제어가 적용된다는 증거를 제시해야만 한다. 7) 구성 관리 가) 안전대잭기술서는 구성 관리 시스템이 정당한 것이며 올바르게 수행된다는 것을 증명 해야만 한다. 나) 구성 관리 하에 있는 모든 안전 관련 프로젝트 항목의 증거가 명시되어야만 한다. 8) 프로젝트 안전 훈련 안전대책기술서는 안전 관련 활동을 수행하는 직원이 정의된 훈련 계획에 따라 적합하게 훈련되었다는 것을 보여야만 한다.
7 기술적 안전 평가	기술적 안전 평가는 설계 원칙, 계산, 시험법 명시, 결과, 안전 분석 등의 모든 지원되는 증거를 포함하여 설계의 안전을 보증하는 기술적 원리가 설명되어야만 한다. 정확한 참고문헌이 문서에 포함되어 있다면 참고문헌을 모두 제시할 필요는 없다. 다음 사항은 기술 안전 평가의

	구성을 위한 지침으로 제공되어야 하는 내용이다. · 도입 · 안전 분석 · 안전 기술 · 안전 감사, 평가 · 안전 관련 적용 상태 · 안전 요구사항 허가 · 기타 특수 안전 문제 작업 활동의 측면에 따른 다음 사항들은 적당한 위치에 위에 열거된 사항들을 포함하여 기입해야만 한다. · 정밀 기능 운영 · 오류의 영향 · 외부 영향의 작용
8 관련된 안전대책기술서	안전대책기술서에는 다른 안전대책기술서에 대한 참고문헌을 포함하고 있어 야만 한다. 이 안전대책기술서는 관련 안전대책기술서의 어떠한 가정이나 제한에 대한 실례를 포함한다.
9 결론	안전대책기술서는 안전 요구 사항과 관련한 시스템의 안전성 허용 가능성에 대해 설명할 수 있어야 한다. 가) 안전대책기술서의 가정 목록은 안전 요구 사항을 위해 특별히 작성된 것임을 확인 나) 시스템에 의해 존재하는 잔존 위험의 제시 다) 시스템 정의의 제시 라) 모든 해결 불가능한 위험 요소의 명시 마) 안전을 위한 작업 방해 요인 혹은 절차상의 부과 문제 바) 안전 관련 작업이 수행되는 것에 대한 명시 혹은 권고 사항 결론부에서는 안전 인증서의 해당 내용과 운영상 용도에서 필요하다고 생각되는 기타 제한을 권고한다.

Appendix 3. IEC 61508 Technique/Measure List

Category		Technique/Measure	한글명(Technique/Measure)	SIL 1	SIL 2	SIL 3	SIL 4
IEC61508-3 A.1 - Software safety requirements specification	1a	Semi-formal methods	준 정형 기법	R	R	HR	HR
	1b	Formal methods	정형 기법	---	R	R	HR
	2	Forward traceability between the system safety requirements and the software safety requirements	시스템 안전 요구 사항과 소프트웨어 안전 요구 사항 간의 전방 추적 성	R	R	HR	HR
	3	Backward traceability between the safety requirements and the perceived safety needs	안전 요구 사항과 인식 된 안전 요구 사항 간의 역 추적 성	R	R	HR	HR
	4	Computer-aided specification tools to support appropriate techniques/measures above	위의 적절한 기술 / 조치를 지원하는 컴퓨터 지원 사양 도구	R	R	HR	HR
IEC61508-3 A.2 - Software design and development - software architecture design	1	Fault detection	오류 감지	---	R	HR	HR
	2	Error detecting codes	코드 감지 오류	R	R	R	HR
	3a	Failure assertion programming	오류 주장 프로그래밍	R	R	R	HR
	3b	Diverse monitor techniques (with independence between the monitor and the monitored function in the same computer)	다양한 모니터 기술 (모니터와 모니터 기능이 동일한 컴퓨터에서 독립적임)	---	R	R	
	3c	Diverse monitor techniques (with separation between the monitor computer and the monitored computer)	다양한 모니터 기술 (모니터 컴퓨터와 모니터링되는 컴퓨터가 분리되어 있음)	---	R	R	HR
	3d	Diverse redundancy, implementing the same software safety requirements specification	다양한 중복성, 동일한 소프트웨어 안전 요구 사항 사양 구현	---	---	---	R
	3e	Functionally diverse redundancy, implementing different software safety requirements specification	기능적으로 다양한 중복, 다른 소프트웨어 안전 요구 사항 사양 구현	---	---	R	HR
	3f	Backward recovery	역방향 복구	R	R	---	NR
	3g	Stateless software design (or limited state design)	무국적 소프트웨어 설계 (또는 제한된 상태 설계)	---	---	R	HR
	4a	Re-try fault recovery mechanisms	장애 복구 메커니즘 다시 시도	R	R	---	---
	4b	Graceful degradation	단계별 성능저하	R	R	HR	HR
	5	Artificial intelligence - fault correction	인공 지능 - 오류 수정	---	NR	NR	NR

	6	Dynamic reconfiguration	동적 재구성	---	NR	NR	NR
	7	Modular approach	모듈 방식	HR	HR	HR	HR
	8	Use of trusted/verified software elements (if available)	신뢰할 수 있고 검증 된 소프트웨어 요소의 사용 (있는 경우)	R	HR	HR	HR
	9	Forward traceability between the software safety requirements specification and software architecture	소프트웨어 안전 요구 사항 사양과 소프트웨어 아키텍처 간의 포워드 추적 성	R	R	HR	HR
	10	Backward traceability between the software safety requirements specification and software architecture	소프트웨어 안전 요구 사항 사양과 소프트웨어 아키텍처 간의 역 추적 성	R	R	HR	HR
	11a	Structured diagrammatic methods	구조 다이어그램	HR	HR	HR	HR
	11b	Semi-formal methods	준 정형 기법	R	R	HR	HR
	11c	Formal design and refinement methods	정식 디자인 및 개선 방법	---	R	R	HR
	11d	Automatic software generation	자동 소프트웨어 생성	R	R	R	R
	12	Computer-aided specification and design tools	컴퓨터 지원 사양 및 설계 도구	R	R	HR	HR
	13a	Cyclic behaviour, with guaranteed maximum cycle time	최대 사이클 시간을 보장하는 주기적 동작	R	HR	HR	HR
	13b	Time-triggered architecture	시간 트리거 아키텍처	R	HR	HR	HR
	13c	Event-driven, with guaranteed maximum response time	이벤트 중심, 최대 응답 시간 보장	R	HR	HR	-
	14	Static resource allocation	정적 리소스 할당	-	R	HR	HR
	15	Static synchronisation of access to shared resources	공유 리소스에 대한 액세스의 정적 동기화	-	-	R	HR
IEC61508-3 A.3 - Software design and development - support tools and programming language	1	Suitable programming language	적절한 프로그래밍 언어	HR	HR	HR	HR
	2	Strongly typed programming language	강력한 형식의 프로그래밍 언어	HR	HR	HR	HR
	3	Language subset	언어 하위 집합	---	---	HR	HR
	4a	Certified tools and certified translators	공인 된 도구 및 공인 번역사	R	HR	HR	HR
	4b	Tools and translators: increased confidence from use	도구 및 번역자 : 사용으로 인한 자신감 증가	HR	HR	HR	HR
IEC61508-3 A.4 - Software design and development - detailed design	1a	Structured methods	구조적 기법	HR	HR	HR	HR
	1b	Semi-formal methods	준 정형 기법	R	HR	HR	HR
	1c	Formal design and refinement methods	정식 디자인 및 개선 방법	---	R	R	HR
	2	Computer-aided design tools	컴퓨터 지원 설계 도구	R	R	HR	HR
	3	Defensive programming	방어 프로그래밍	---	R	HR	HR
	4	Modular approach	모듈 방식	HR	HR	HR	HR

	5	Design and coding standards	디자인 및 코딩 표준	R	HR	HR	HR
	6	Structured programming	구조화 된 프로그래밍	HR	HR	HR	HR
	7	Use of trusted/verified software elements (if available)	신뢰할 수 있고 검증 된 소프트웨어 요소의 사용 (있는 경우)	R	HR	HR	HR
	8	Forward traceability between the software safety requirements specification and software design	소프트웨어 안전 요구 사항 사양과 소프트웨어 디자인 간의 포워드 추적 성	R	R	HR	HR
IEC61508-3 A.5 - Software design and development - software module testing and integration	1	Probabilistic testing	확률 론적 테스트	---	R	R	R
	2	Dynamic analysis and testing	동적 분석 및 테스트	R	HR	HR	HR
	3	Data recording and analysis	데이터 기록 및 분석	HR	HR	HR	HR
	4	Functional and black box testing	기능 및 블랙 박스 테스트	HR	HR	HR	HR
	5	Performance testing	성능 테스트	R	R	HR	HR
	6	Model based testing	모델 기반 테스트	R	R	HR	HR
	7	Interface testing	인터페이스 테스트	R	R	HR	HR
	8	Test management and automation tools	테스트 관리 및 자동화 도구	R	HR	HR	HR
	9	Forward traceability between the software design specification and the module and integration test specifications	소프트웨어 설계 명세와 모듈 및 통합 테스트 명세 간의 전향 적 추적 성	R	R	HR	HR
	10	Formal verification	정형 검증	---	---	R	R
IEC61508-3 A.6 - Programmable electronics integration (hardware and software)	1	Functional and black box testing	기능 및 블랙 박스 테스트	HR	HR	HR	HR
	2	Performance testing	성능 테스트	R	R	HR	HR
	3	Forward traceability between the system and software design requirements for hardware/software integration and the hardware/software integration test specifications	하드웨어 / 소프트웨어 통합 및 하드웨어 / 소프트웨어 통합 테스트 사양에 대한 시스템과 소프트웨어 설계 요구 사항 간의 전향 적 추적 성	R	R	HR	HR
IEC61508-3 A.7 - Software aspects of system safety validation	1	Probabilistic testing	확률 론적 테스트	---	R	R	HR
	2	Process simulation	공정 시뮬레이션	R	R	HR	HR
	3	Modelling	모델링	R	R	HR	HR
	4	Functional and black-box testing	기능 및 블랙 박스 테스트	HR	HR	HR	HR
	5	Forward traceability between the software safety requirements specification and the software safety validation plan	소프트웨어 안전 요구 사항 사양과 소프트웨어 안전 유효성 검사 계획 간의 전향 추적 성	R	R	HR	HR
	6	Backward traceability between the software safety validation plan and the	소프트웨어 안전성 검증 계획과 소프트웨어 안전성 요구 사항 사양 간의 역	R	R	HR	HR

		software safety requirements specification	추적 성				
IEC61508-3 A.8 - Modification	1	Impact analysis	영향 분석	HR	HR	HR	HR
	2	Reverify changed software module	변경된 소프트웨어 모듈 재확인	HR	HR	HR	HR
	3	Reverify affected software modules	영향을받는 소프트웨어 모듈 재확인	R	HR	HR	HR
	4a	Revalidate complete system	전체 시스템 재 검증	---	R	HR	HR
	4b	Regression validation	회귀 검증	R	HR	HR	HR
	5	Software configuration management	소프트웨어 구성 관리	HR	HR	HR	HR
	6	Data recording and analysis	데이터 기록 및 분석	HR	HR	HR	HR
	7	Forward traceability between the Software safety requirements specification and the software modification plan (including reverification and revalidation)	소프트웨어 안전 요구 사항 사양과 소프트웨어 수정 계획 (재 검증 및 재 검증 포함) 간의 전향 적 추적 성	R	R	HR	HR
	8	Backward traceability between the software modification plan (including reverification and revalidation)and the software safety requirements specification	소프트웨어 수정 계획 (재 검증 및 재 검증 포함)과 소프트웨어 안전 요구 사항 명세 사이의 역 추적 성	R	R	HR	HR
IEC61508-3 A.9 - Software verification	1	Formal proof	정식 증명	---	R	R	HR
	2	Animation of specification and design	명세 및 디자인의 생기	R	R	R	R
	3	Static analysis	정적 분석	R	HR	HR	HR
	4	Dynamic analysis and testing	동적 분석 및 테스트	R	HR	HR	HR
	5	Forward traceability between the software design specification and the software verification (including data verification) plan	소프트웨어 설계 사양과 소프트웨어 검증 (데이터 검증 포함) 계획 간의 포워드 추적 성	R	R	HR	HR
	6	Backward traceability between the software verification (including data verification) plan and the software design specification	소프트웨어 검증 (데이터 검증 포함) 계획과 소프트웨어 설계 명세 간의 역 추적 성	R	R	HR	HR
	7	Offline numerical analysis	오프라인 수치 분석	R	R	HR	HR
	Software module testing and integration		소프트웨어 모듈 테스트 및 통합	See IEC61508-3 A.5			
	Programmable electronics integration testing		프로그래밍 가능한 전자 통합 테스트	See IEC61508-3 A.6			
	Software system testing (validation)		소프트웨어 시스템 테스트	See IEC61508-3			

			(검증)	A.7			
IEC61508-3 A.10 - Functional safety assessment	1	Checklists	점검표	R	R	R	R
	2	Decision/truth tables	의사 결정 / 진리표	R	R	R	R
	3	Failure analysis	고장 분석	R	R	HR	HR
	4	Common cause failure analysis of diverse software (if diverse software is actually used)	다양한 소프트웨어의 공통 원인 실패 분석 (다양한 소프트웨어가 실제로 사용되는 경우)	---	R	HR	HR
	5	Reliability block diagram	신뢰성 블록 다이어그램	R	R	R	R
	6	Forward traceability between the requirements of Clause 8 and the plan for software functional safety assessment	8 절의 요구 사항과 소프트웨어 기능 안전성 평가 계획 사이의 전방 추적성	R	R	HR	HR
IEC61508-3 B.1 - Design and coding standards	1	Use of coding standard to reduce likelihood of errors	오류 가능성을 줄이기위한 코딩 표준의 사용	HR	HR	HR	HR
	2	No dynamic objects	동적 객체 없음	R	HR	HR	HR
	3a	No dynamic variables	동적 변수 없음	---	R	HR	HR
	3b	Online checking of the installation of dynamic variables	동적 변수 설치의 온라인 검사	---	R	HR	HR
	4	Limited use of interrupts	제한된 인터럽트 사용	R	R	HR	HR
	5	Limited use of pointers	포인터의 사용 제한	---	R	HR	HR
	6	Limited use of recursion	제한된 재귀 사용	---	R	HR	HR
	7	No unstructured control flow in programs in higher level languages	고수준 언어의 프로그램에서 구조화되지 않은 제어 흐름이 없음	R	HR	HR	HR
	8	No automatic type conversion	자동 유형 변환 없음	R	HR	HR	HR
IEC61508-3 B.2 - Dynamic analysis and testing	1	Test case execution from boundary value analysis	경계 값 분석을 통한 테스트 케이스 실행	R	HR	HR	HR
	2	Test case execution from error guessing	오류 추측에서 테스트 사례 실행	R	R	R	R
	3	Test case execution from error seeding	오류 시드에서 테스트 사례 실행	---	R	R	R
	4	Test case execution from model-based test case generation	모델 기반 테스트 케이스 생성으로부터 테스트 케이스 실행	R	R	HR	HR
	5	Performance modelling	성능 모델링	R	R	R	HR
	6	Equivalence classes and input partition testing	동등한 클래스와 입력 파티션 테스트	R	R	R	HR
	7a	Structural test coverage (entry points) 100 %	구조 테스트 커버리지 (진입점) 100 %	HR	HR	HR	HR
	7b	Structural test coverage (statements) 100 %	구조 테스트 커버리지 (명세서) 100 %	R	HR	HR	HR
	7c	Structural test coverage (branches) 100 %	구조 테스트 커버리지 (지점) 100 %	R	R	HR	HR
	7d	Structural test coverage (conditions, MC/DC) 100 %	구조 테스트 커버리지 (조건, MC / DC) 100 %	R	R	R	HR
IEC61508-3	1	Test case execution from	원인 결과 다이어그램에서	---	---	R	R

B.3 - Functional and black-box testing		cause consequence diagrams	테스트 사례 실행				
	2	Test case execution from model-based test case generation	모델 기반 테스트 케이스 생성으로부터 테스트 케이스 실행	R	R	HR	HR
	3	Prototyping/animation	프로토 타이핑 / 애니메이션	---	---	R	R
	4	Equivalence classes and input partition testing, including boundary value analysis	경계 값 분석을 포함한 동등한 클래스와 입력 파티션 테스트	R	HR	HR	HR
	5	Process simulation	공정 시뮬레이션	R	R	R	R
IEC61508-3 B.4 - Failure analysis	1a	Cause consequence diagrams	원인 다이어그램	R	R	R	R
	1b	Event tree analysis	이벤트 트리 분석	R	R	R	R
	2	Fault tree analysis	결함 트리 분석	R	R	R	R
	3	Software functional failure analysis	소프트웨어 기능 장애 분석	R	R	R	R
IEC61508-3 B.5 - Modelling	1	Data flow diagrams	데이터 흐름도	R	R	R	R
	2a	Finite state machines	유한 상태 기계	---	R	HR	HR
	2b	Formal methods	정형 기법	---	R	R	HR
	2c	Time Petri nets	시간 페 트리 그물	---	R	HR	HR
	3	Performance modelling	성능 모델링	R	HR	HR	HR
	4	Prototyping/animation	프로토 타이핑 / 애니메이션	R	R	R	R
	5	Structure diagrams	구조 다이어그램	R	R	R	HR
IEC61508-3 B.6 - Performance testing	1	Avalanche/stress testing	눈사태 / 스트레스 테스트	R	R	HR	HR
	2	Response timings and memory constraints	응답 타이밍 및 메모리 제약	HR	HR	HR	HR
	3	Performance requirements	성능 요구 사항	HR	HR	HR	HR
IEC61508-3 B.7 - Semi-formal methods	1	Logic/function block diagrams	논리 / 기능 블록 다이어그램	R	R	HR	HR
	2	Sequence diagrams	시퀀스 다이어그램	R	R	HR	HR
	3	Data flow diagrams	데이터 흐름도	R	R	R	R
	4a	Finite state machines/state transition diagrams	유한 상태 기계 / 상태 전이 다이어그램	R	R	HR	HR
	4b	Time Petri nets	시간 페 트리 그물	R	R	HR	HR
	5	Entity-relati ᵒ nship-attribute data models	엔터티 관련 특성 데이터 모델	R	R	R	R
	6	Message sequence charts	메시지 시퀀스 차트	R	R	R	R
	7	Decision/truth tables	의사 결정 / 진리표	R	R	HR	HR
	8	UML	UML	R	R	R	R
IEC61508-3 B.8 - Static analysis	1	Boundary value analysis	경계 값 분석	R	R	HR	HR
	2	Checklists	점검표	R	R	R	R
	3	Control flow analysis	제어 흐름 분석	R	HR	HR	HR
	4	Data flow analysis	데이터 흐름 분석	R	HR	HR	HR
	5	Error guessing	오류 추측	R	R	R	R
	6a	Formal inspections, including specific criteria	특정 기준을 포함한 공식 검사	R	R	HR	HR
	6b	Walk-through (software)	워크 쓰루 (소프트웨어)	R	R	R	R
	7	Symbolic execution	상징적 실행	---	---	R	R
	8	Design review	디자인 검토	HR	HR	HR	HR

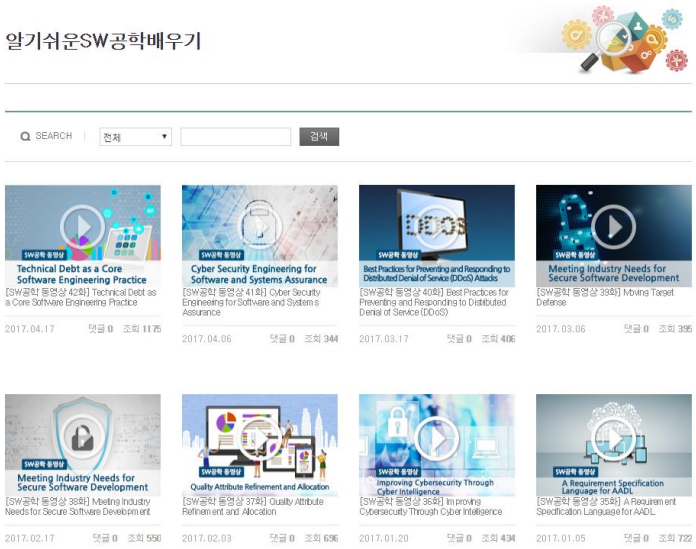
	9	Static analysis of run time error behaviour	런타임 오류 동작의 정적 분석	R	R	R	HR
	10	Worst-case execution time analysis	최악의 실행 시간 분석	R	R	R	R
IEC61508-3 B.9 - Modular approach	1	Software module size limit	소프트웨어 모듈 크기 제한	HR	HR	HR	HR
	2	Software complexity control	소프트웨어 복잡성 제어	R	R	HR	HR
	3	Information hiding/encapsulation	정보 숨기기 / 캡슐화	R	HR	HR	HR
	4	Parameter number limit / fixed number of subprogram parameters	매개 변수 개수 제한 / 고정된 서브 프로그램 매개 변수 수	R	R	R	R
	5	One entry/one exit point in subroutines and functions	서브 루틴과 함수에서 한 개의 엔트리 포인트 / 한 개의 출구 포인트	HR	HR	HR	HR
	6	Fully defined interface	완전히 정의 된 인터페이스	HR	HR	HR	HR

Appendix 4. SW 공학 및 품질관련 유용한 사이트

정보통신산업진흥원 소프트웨어공학포탈(구 소프트웨어공학센터)에는 수년간 축적한 유용한 SW 공학 및 개발에 도움이 되는 자료가 많이 있음.

지식마당 > 알기쉬운 SW 공학배우기

http://www.sw-eng.kr/member/customer/Learn/BoardList.do



지식마당 > SW 개발도우미

http://www.sw-eng.kr/member/00000000000000028542/Cms/BoardView.do