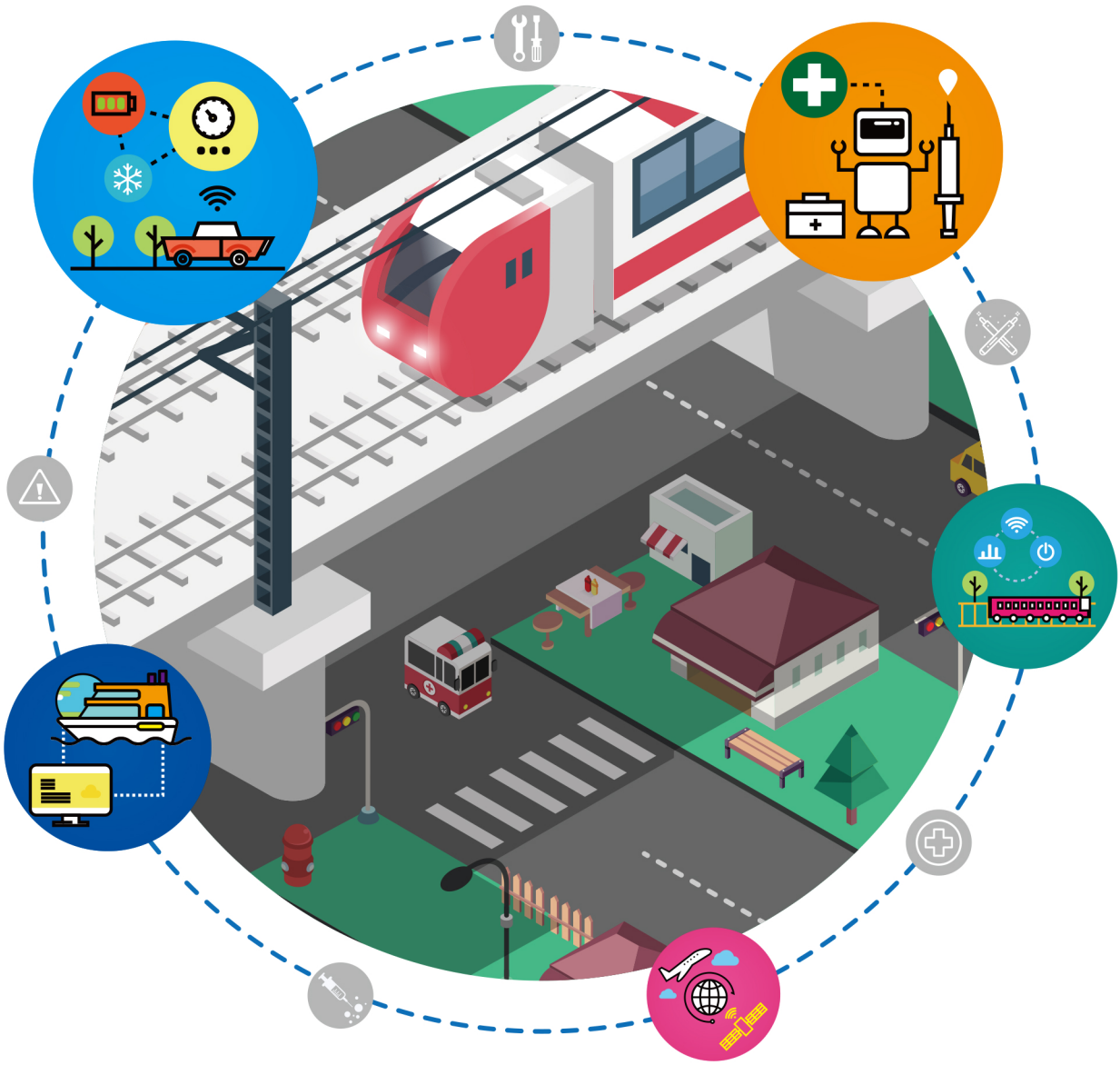
SW안전가이드

철도 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

철도 분야 SW안전가이드를 집필하며

최근 소프트웨어가 사회 각 분야를 주도하는 소프트웨어 중심사회가 도래하면서 국민 안전 확보를 위한 핵심요소로 부각되고 있으며, 철도 산업 분야의 경우 철도 관련 시스템의 규모 및 복잡성의 증가로 인해 최근 철도 소프트웨어 개발 시 국제 안전 표준을 준수한 소프트웨어 안전성 확보 활동을 의무화 하고 있다.

국내 철도 산업은 정부의 주도적인 발주가 없으면 일부 업체만이 살아남는 포화 상태이기 때문에 해외 시장으로 눈을 돌려야 하는 형편이지만 해외 시장에 진입하기 위해서는 고 수준의 위험도 분석, 제어 및 검증 기술력을 통한 안전성을 확보해야 한다. 그러나 국내 업체들 대부분 안전과 관련된 위험도 분석, 제어 및 검증 기술력이 낮고 기술력이 축적되어 있지 않는 실정이며, 실무에서 시스템 및 소프트웨어 개발자들이 국제 안전 표준을 준수해 안전 활동 수행 시 각 활동에 대한 요건을 이해하고 적용하는데 많은 어려움이 존재함으로 개발 현장의 실무차원에서 안전 표준을 보다 쉽게 이해하고 안전 목표 수준을 달성할 수 있도록 구체적인 수행 지침과 쉽게 따라할 수 있도록 실 사례 등을 포함한 철도 분야 SW안전가이드를 개발하였다.

본 과제에서 제시하는 철도 분야 안전가이드는 크게 표준 IEC 62278 기반의 시스템 안전성 분석 가이드와 IEC 62279 기반의 철도 소프트웨어 개발가이드로 구성되어 있으며, SW안전가이드의 시범적용을 통해 가이드의 적정성 검증과, 시범적용 시 확보된 산출물을 실제 개발 현장에서 활용 가능한 수준으로 사례를 가이드에 포함해 제공함으로써 가이드의 사용성과 적용성을 제고하였다.

본 철도 분야 SW안전가이드를 활용함으로써 인력이나 기술 수준이 영세한 철도 소프트웨어 개발관련 업체들이 안전성 높은 소프트웨어 시스템을 개발하기 위해서 무엇이 필요하고 어떻게 수행되어야 하는지를 이해할 수 있을 것이다.

본 가이드를 통해서 안전 소프트웨어 개발의 주요 활동 및 산출물 양식을 이해하는 것이, 업체의 경쟁력 강화에 큰 도움이 될 것이다.

경기대학교 산학협력단, (주)에스피아이디 컨소시엄
책임자: 권 기 현 교수

목 차

표 목 차

그림 목차

제 1 장　서　론

제 1 절 배경 및 필요성

철도는 대량의 승객을 한 지역에서 다른 지역으로 빠르게 이동시키는 교통수단이다. 뿐만 아니라, 자동차에 비해서 철도는 미리 정해진 철로를 이용하기 때문에 출발 및 도착 시간이 정확하다. 이러한 편리함으로 인해서 철도가 현대인의 교통수단으로 널리 활용되고 있으나, 만약 사고가 일어나면 그 피해 규모가 매우 크기 때문에, 철도 시스템을 개발할 때에는 안전성을 반드시 고려해야 한다.

기본적으로 철도 시스템의 안전성은 신호 제어에 달려있다. 철로에 있는 다른 열차의 존재 유무, 선행 열차와의 거리 간격 등을 고려하여 신호를 보내면, 기관사 또는 무인 시스템은 신호에 따라서 열차의 운행 속도를 제어해야 한다. 사고조사위원회 보고서에 따르면, 2014년에 발생되었던 상왕십리역 추돌사고는 신호제어를 수행하는 소프트웨어 오류가 원인이었던 것으로 밝혀졌다.

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 50129를 먼저 만들었다. 그 후에, 이들 유럽 표준은 국제 표준 IEC 62278[1], IEC 62279[2], IEC 62425[3]로 각각 제정되었다. [그림 1]은 국제 표준들 간의 관련성이다.

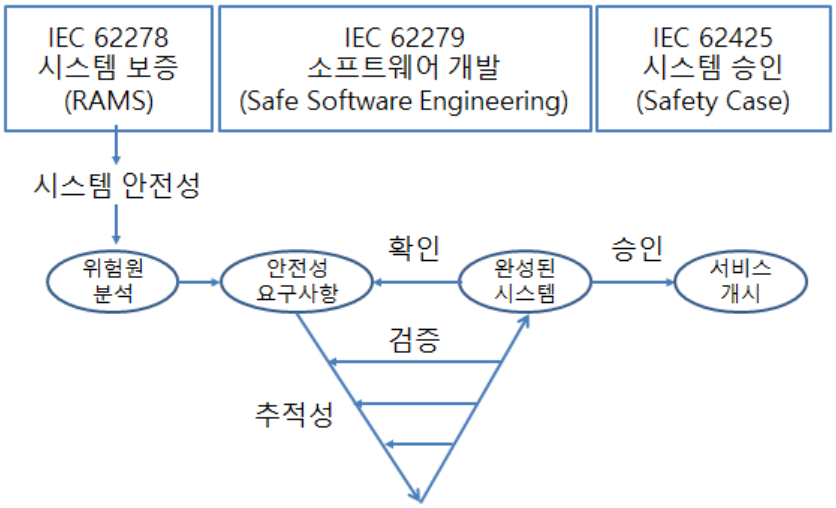


그림 1 철도 국제 표준들

IEC 62278은 철도 시스템을 믿고 사용할 수 있는지를(dependability) 보증하기 위해 네 가지 시스템 보증 요소인 RAMS (Reliability, Availability, Maintainability, Safety)를 다룬다. 본 과제의 목표가 안전 가이드 개발이기 때문에, 본 과제에서는 네 가지 요소 중에

1) IEC 62278(EN 50126) : 철도신호분야 RAMS
2) IEC 62279(EN 50128) : 철도신호분야 안전관련 소프트웨어 개발 및 유지보수
3) IEC 62425(EN 50129) : 철도신호분야 통신 및 안전관련 문서 증거인 종합안전대책보고서

서 안전성을 다룬다. 안전성과 관련된 중요한 활동 중의 하나는 안전 요구사항을 식별하는 위험원 분석이다. IEC 62279는 안전성 높은 소프트웨어 개발을 돕는 표준이다. 안전 요구사항 명세로부터 V 모델에 따라서 소프트웨어를 개발하는 절차, 산출해야 할 문서 및 각 단계마다 따라야 하는 기술적인 요구사항을 정의하고 있다. 안전성 높은 소프트웨어 개발을 보증하기 위해서는 검증, 확인 및 추적성 등의 활동이 필수적이다. IEC 62425는 시스템의 안전성 승인이다. 개발된 시스템은 안전성 승인을 받은 후에 공공에 사용되어야 한다. 안전성 승인을 받기 위해서 인허가 당국에 안전성 입증과 관련된 모든 증거를 종합안전대책보고서(safety case) 형식으로 제출한다.

하지만 이러한 국제 표준은 안전성을 달성하기 위해 필요한 활동들에 대한 지침과 이를 만족하기 위한 기법만을 제시하고 있어 실무에서 시스템 및 소프트웨어 개발자들이 이해하고 적용하는데 많은 어려움이 존재한다. 따라서 이와 같은 문제점을 해소하기 위해 시스템 개발 현장의 실무차원에서 안전 표준을 쉽게 이해하고 안전 목표 수준을 달성 할 수 있도록 구체적인 수행 지침이 필요하며, 이론적이고 절차적인 설명에서 벗어나 쉽게 참조하고 따라할 수 있는 실 사례 등이 포함된 가이드의 개발이 필요하다. 이를 위하여 2016년 NIPA “철도 분야 소프트웨어 신뢰·안전성 확보를 위한 가이드 개발과 시범적용 용역” 과제를 통해서 철도 안전 소프트웨어 가이드가 개발되었다. 이 가이드는 실무자들이 관련 국제 표준을 이해해서 안전 요구사항을 철저히 식별하고, 안전 요구사항에 부합하는 소프트웨어 개발을 도움으로서, 철도 시스템의 안전성 향상에 기여하고자 제작되었다. 본 과제는 2016년의 후속 과제로서, 이미 작성된 철도 안전 소프트웨어 가이드의 시스템 안전성 분석 및 위험도 평가 부분을 상세히 다듬고, 개발뿐만 아니라 통합 및 유지보수에 이르기까지 소프트웨어 개발 전 단계를 포함하도록 기존 가이드를 확장한다.

제 2 절 목 적

철도 안전 가이드의 개발 목적은 철도관련 시스템을 설계하고 개발을 담당하는 조직 및 이해관계자들이 철도 분야 국제 표준을 실무 수준에서 쉽게 이해하고 적용 가능하도록 시스템 개발 생명주기 각 단계 별 안전활동의 구체적인 수행 절차와 지침, 실 사례 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 확장하고자 한다. 참고로, 기존 가이드의 구성은 그림 2와 같다.

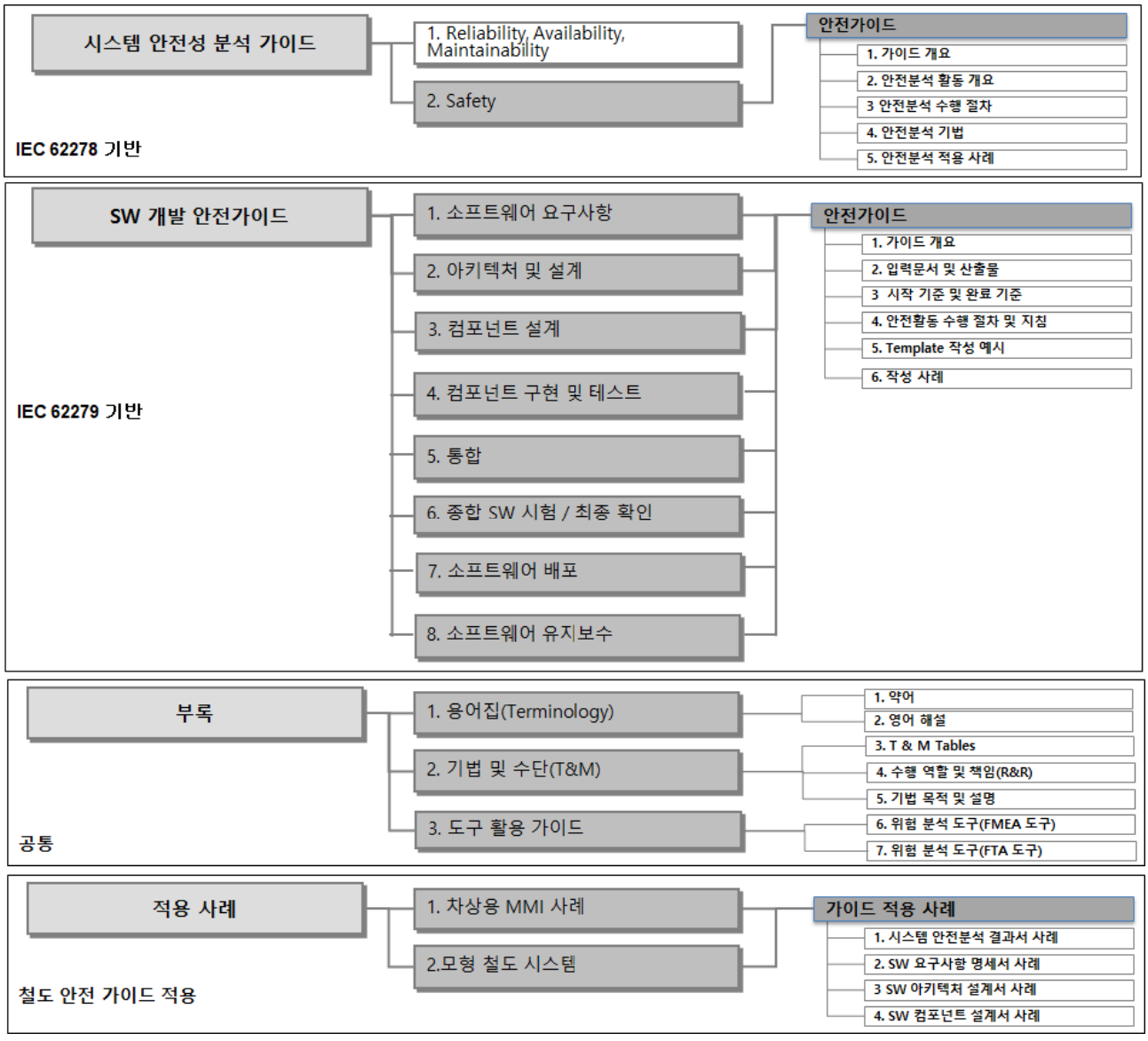


그림 2 2016년에 제작된 철도 안전 가이드의 구성

확장의 핵심은 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, FRACAS을 상세하게 안내하고 해설한다. 이전 가이드에서는 일부 기법만을 다루었으나, 본 과제에서는 실무에서 사용되는 안전성 분석 및 위험도 평가와 관련된 거의 모든 기법을 다룬다. 둘째, IEC 62279에 명시된 소프트웨어 전 단계를 지원하도록 기존 가이드를 확장한다. 이전 가이드에서는 시간 제약으로 인해서 요구사항, 아키텍처 및 설계, 컴포넌트 설계, 컴포넌트 구현 및 테스팅 단계만을 다루었다. 즉, V 모델로 얘기하면, 위에서 아래로 내려가는 왼쪽 영역을 다루었다. 본 과제에서는 통합 단계, 종합 테스트 단계, 배포 단계, 유지보수 단계를 추가함으로서 IEC 62279 소프트웨어 개발 전 단계를 지원한다.

제 3 절 과제 범위 및 수행방법

1. 과제 범위

본 과제는 소프트웨어 안전성을 확보할 수 있도록 주요 소프트웨어 개발 단계 별 수행 지침, 활동 절차, 수행기법 등으로 구성된 가이드를 개발하고, 그리고 이러한 가이드를 실제 개발 현장에서 활용 할 수 있도록 적용 사례 등을 제시한다.

표 1 과제 범위

구 분	범 위
현황 분석 및 기존 가이드의 개선점 파악	▪ 철도 형식 승인 및 철도기술기준의 의의 및 대응 ▪ 철도 업체 목록 조사 ▪ 현장적용지원사업 적용을 통한 기존 가이드의 개선점 도출
철도 안전가이드	▪ IEC 62278 기반의 시스템 안전성 분석 가이드 개발(지침, 수행절차, 수행기법, 산출물 양식, 적용 사례 등) ▪ IEC 62279 기반의 소프트웨어 개발가이드 개발(지침, 수행절차, 수행기법, 산출물 양식, 적용 사례 등) ▪ 가이드의 내용과 국내 철도기술기준 및 국제 표준과의 대응을 나타내는 조견표 ▪ 빈번하게 사용되는 안전 기술 및 대책(T&M) 정리
적용 사례	▪ 가이드 적용 철도 시스템 개요 ▪ 주요 소프트웨어 가이드 적용에 따른 산출물 예시

2. 과제 수행방법

○ 현황분석 시 기 작성 된 철도 관련 안전성 분석 연구결과 보고서, 문헌조사, 보도자료 및 현업 전문가 등을 통해 철도 안전 가이드 개발에 대한 방향성과 핵심 요구사항을 도출하였다.

○ 안전가이드 개발 시 현황조사에 따른 개선사항 및 요구사항의 반영과 자동차, 항공 등 타 산업부분의 안전 표준 및 가이드의 안전 지침 등을 참조해 안전활동의 수행 절차와 기법 등을 가이드에 반영하였다.

○ 안전가이드의 주요 내용은 철도 분야의 소프트웨어 시스템을 개발하는데 있어 필요한 안전활동과 이에 관련한 수행 기법을 구체화 하는 것이다. 기본적으로 철도 소프트웨어 안전성은 시스템 수준에서의 안전성 분석 결과를 바탕으로 확보되어야 하므로 시스템 수준에서의 안전성 분석을 위한 지침, 상세 수행 절차, 수행 기법들을 IEC 62278 표준에 기반하여 시스템 안전성 분석 가이드를 개발하였다.

- 안전가이드 개발 시 정기적인 철도, 자동차, 항공, 산업일반 분야의 소프트웨어 안전 전문가 검토 회의를 통해 가이드의 내용과 품질을 검증하였다.
- 철도 안전가이드의 실무적용 사용성 등을 검증하기 위해 적용 철도 시스템을 선정하고 시범적용을 통해 가이드의 활용 수준을 검증하다. 시범적용 시 안전가이드를 통해 작성된 산출물을 사례로 포함함으로써 가이드의 현장 실무 적용 편의성을 제고하였다.
- 또한 모형철도 시스템의 안전기능 개발에 안전가이드를 추가적으로 적용해 산출물을 작성하고 실 사례를 보고서에 포함함으로써 가이드의 이해도와 활용성을 제고하였다.

제 4 절 보고서 구성

본 보고서의 구성은 크게 철도분야 안전 관련 현황분석, 시스템 개발생명 주기에 따른 시스템 안전성 분석 가이드 및 소프트웨어 개발을 위한 안전가이드, 가이드 적용 사례, 가이드 적용 시 참조해야 할 부록으로 구성되어 있다.

1. 현황분석

철도안전법에 따라 2017년 6월부터 실시되는 철도기술기준의 의의 및 기업의 대응방안, 정부의 지원방안을 다루고, 국내 철도 업체의 목록을 기술한다. 또한, 이전에 만들어진 가이드를 현장적용지원사업에 적용하여 얻었던 경험 및 개선 사항을 기술한다.

2. 철도 안전가이드

철도 안전가이드는 시스템 수준에서 잠재적 위험원의 식별 및 분석, 식별된 고 수준의 위험을 허용 가능 수준으로 낮추는 시스템 안전대책 도출을 위한 시스템 안전성 분석 가이드와, 식별될 시스템 안전대책으로 부터 철도 분야 소프트웨어 시스템을 개발함에 있어 안전성 확보를 위한 요구사항 명세, 설계, 구현, 검증, 통합, 종합 테스트, 배포, 유지보수 단계 별 8개의 소프트웨어 개발 가이드로 구성된다.

3. 적용사례

철도 안전가이드 적용 시 가이드의 실무 적용성 및 활용성 제고를 위해 실제 철도 관련 시스템의 분석 및 설계 산출물 사례를 제공한다. 사례로 선정한 모형 철도의 ACC[4] 개발에 가이드를 적용하여 작성된 분석 및 설계 산출물 등을 수록하였다.

4. 부 록

안전가이드 적용 시 참조해야 할 용어집과 각 단계 별 가이드의 활동 수행에 따라 준용 또는 적용해야 할 기술적인 기법과 측정, 주요 기법의 설명, 그리고 기법적용 시 활용 가능한 도구 가이드 설명으로 구성된다. 또한 현황분석 시 사용한 인터뷰 설문 항목도 포함하였다.

4) ACC (Adaptive Cruise Control) : 앞 열차와 안전 속도를 유지하는 정속주행 시스템이다.

제 5 절 철도 안전가이드 적용 범위

본 철도 안전가이드는 크게 IEC 62278 기반의 시스템 안전성 분석 가이드와 IEC 62279 기반의 소프트웨어 개발 가이드로 구성되어 있다.

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 있다.

다음 그림은 IEC 62278 및 IEC 62279 표준을 기반으로 시스템을 개발 시 본 철도 안전가이드의 적용 범위를 나타낸 그림이다.

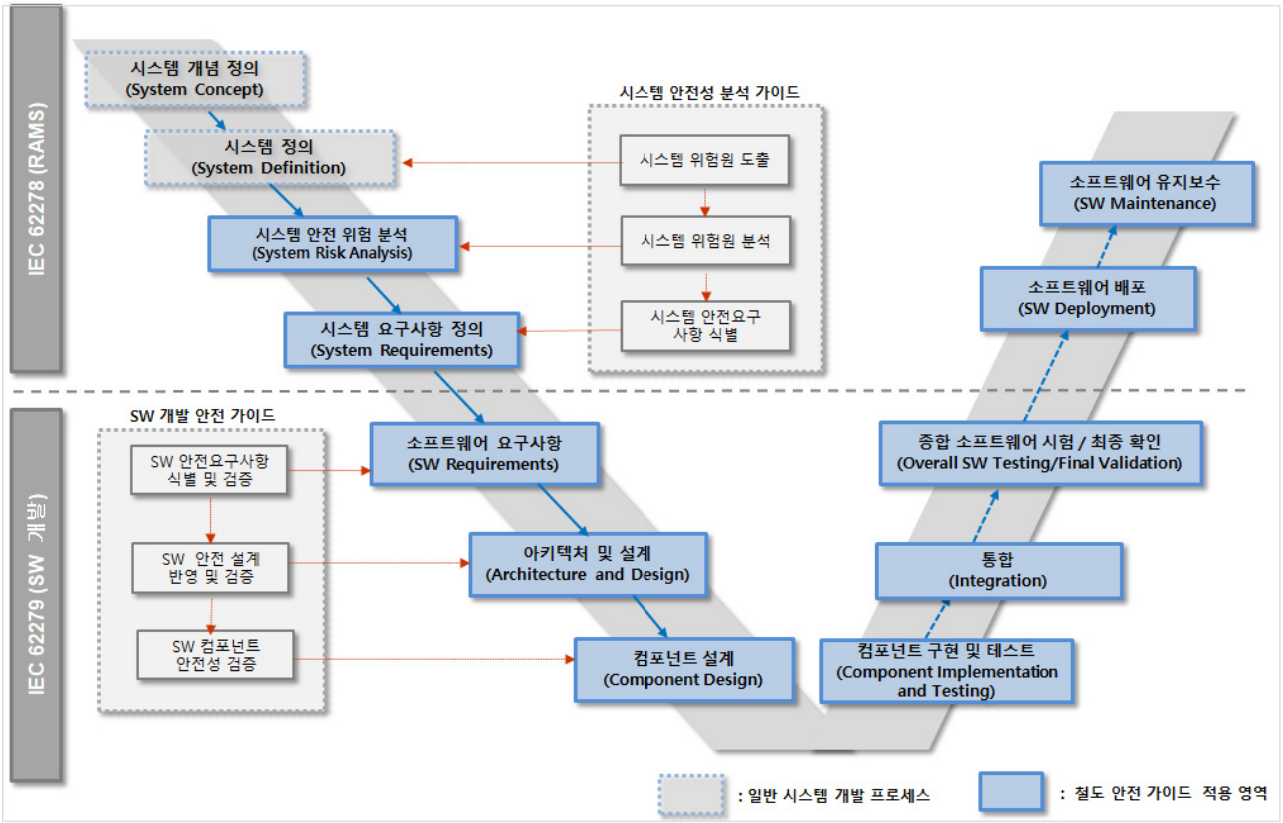


그림 3 철도 안전가이드 적용 범위

제 6 절 단계 별 철도 안전가이드 활용 표

시스템 및 소프트웨어 개발 가이드를 적용함에 있어 안전성 분석에 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 분석의 전반적인 내용을 용이하게 이해할 수 있다.

표 2 시스템 안전성 분석 가이드 표

단 계	입력 문서	주요 활동	분석 기법	산 출 물
시스템 안전성 분석	▪ 시스템 요구사항 명세서 ▪ 시스템 아키텍처 명세서	▪ 구조분석 ▪ 기능분석 ▪ 오류분석 ▪ 위험원 식별 ▪ 위험도 평가 ▪ 고장모드 식별 ▪ 기능기반 고장 영향도 평가	▪ PHA ▪ SHA ▪ FTA ▪ FMEA ▪ HAZOP	▪ 예비위험원분석서 ▪ 시스템 안전성 분석서 ▪ 고장모드 영향 분석서 ▪ FT 분석서

표 3 소프트웨어 개발 안전가이드 표

단 계	입력 문서	주요 활동	점검 방안	산 출 물
소프트웨어 요구사항 명세	▪ 시스템 요구사항 명세서 ▪ 시스템 안전 요구사항 명세서 ▪ 시스템 아키텍처 기술서 ▪ 외부 인터페이스 명세서 ▪ 소프트웨어 품질 보증 계획 ▪ 소프트웨어 확인 계획	▪ 요구사항 명세서 작성 ▪ 요구사항 테스트 명세서 작성 ▪ 요구사항 검증 보고서 작성 ▪ 추적성 유지 ▪ 가독성, 테스트 가능성 ▪ 소프트웨어 요구사항의 안전성 분석	▪ 소프트웨어 요구사항 명세 점검목록	▪ 소프트웨어 요구사항 명세서 ▪ 전체 소프트웨어 테스트 명세서 ▪ 소프트웨어 요구사항 검증 보고서
소프트웨어 아키텍처 및 설계 명세	▪ 소프트웨어 요구사항 정의서 ▪ 시스템 설계 기술서 ▪ 외부	▪ 컴포넌트 식별 ▪ 인터페이스 식별 ▪ 소프트웨어 아키텍처 및 설계 안전성 분석	▪ 소프트웨어 아키텍처 명세 점검목록 ▪ 소프트웨어 인터페이스 명세	▪ 소프트웨어 아키텍처 명세서 ▪ 소프트웨어 인터페이스 명세서 ▪ 소프트웨어 설계 명세서

단 계	입력 문서	주요 활동	점검 방안	산 출 물
	인터페이스 명세서		점검목록 ▪ 소프트웨어 설계 명세 점검목록	▪ 소프트웨어 통합 테스트 명세서 ▪ 소프트웨어/하드웨어 통합 테스트 명세서 ▪ 소프트웨어 아키텍처 및 설계 검증 보고서
소프트웨어 컴포넌트 설계 명세	▪ 소프트웨어 설계 명세서	▪ 컴포넌트 상세 설계 ▪ 인터페이스 상세 설계 ▪ 소프트웨어 컴포넌트 설계 안전성 분석	▪ 소프트웨어 컴포넌트 설계 점검 목록	▪ 소프트웨어 컴포넌트 설계 명세서 ▪ 소프트웨어 컴포넌트 테스트 명세서 ▪ 소프트웨어 컴포넌트 설계 검증 보고서
소프트웨어 컴포넌트 구현 및 테스팅	▪ 소프트웨어 컴포넌트 설계 명세서 ▪ 소프트웨어 컴포넌트 테스트 명세서	▪ 소스 코드의 구현 ▪ 소스 코드의 용량 및 복잡도 조절 ▪ 정적 분석 (코딩 규칙 준수 및 런타임 오류(RTE) 검출)	▪ 정적 분석 (코딩 규칙(MISRA-C/C++) 준수, RTE 검출)	▪ 소프트웨어 소스 코드 및 지원 문서 ▪ 소프트웨어 컴포넌트 테스트 보고서 ▪ 소프트웨어 소스 코드 검증 보고서
통합	▪ 소프트웨어/하드웨어 통합 테스트 명세서 ▪ 소프트웨어 통합 테스트 명세서	▪ 코딩표준 가이드 검토 ▪ 컴포넌트 구현 ▪ 소스코드 정적 테스트 ▪ 소프트웨어 / HW 통합 ▪ 통합 테스트	▪ 통합 테스트 보고서 점검 ▪ 통합 테스트 결과서 점검 ▪ 통합 검증 보고서 점검	▪ 소프트웨어 통합 테스트 결과서 ▪ 소프트웨어/하드웨어 통합 테스트 결과서 ▪ 소프트웨어 통합 검증 보고서
종합 소프트웨어 시험 / 최종 확인	▪ 소프트웨어 요구사항 명세서 ▪ 종합 소프트웨어 테스트 명세서 ▪ 소프트웨어 검증 계획서 ▪ 소프트웨어 확인 계획서 ▪ 기 검증 결과들을 포함한 모든 하드웨어 및 소프트웨어 문서	▪ 종합 소프트웨어 테스트 ▪ 소프트웨어 확인 ▪ 종합 소프트웨어 검증 확인	▪ 종합 소프트웨어 테스트 보고서 점검 ▪ 소프트웨어 확인 보고서 점검	▪ 종합 소프트웨어 테스트 보고서 ▪ 종합 소프트웨어 테스트 검증 보고서 ▪ 소프트웨어 확인 보고서 ▪ 배포(Release) 노트

단 계	입력 문서	주요 활동	점검 방안	산 출 물
	▪ 시스템 안전 요구사항 명세서			
소프트웨어 배포	▪ 배포와 관련된 모든 디자인, 개발 및 분석 문서	▪ 소프트웨어 릴리스 및 배포 계획 수립 ▪ 소프트웨어 배포 매뉴얼 작성 ▪ 소프트웨어 릴리스 ▪ 소프트웨어 배포 ▪ 소프트웨어 배포 검증	▪ 소프트웨어 릴리스 및 배포 계획서 점검 ▪ 소프트웨어 배포 매뉴얼 점검	▪ 소프트웨어 릴리스 및 배포 계획서 ▪ 소프트웨어 배포 매뉴얼 ▪ 릴리스 노트 ▪ 배포 기록 ▪ 배포 검증 보고서
소프트웨어 유지보수	▪ 모든 디자인, 개발 및 분석 문서	▪ 소프트웨어 유지보수 계획 ▪ 소프트웨어 변경 ▪ 소프트웨어 유지보수 ▪ 소프트웨어 유지보수 검증	▪ 소프트웨어 유지보수 계획서 점검 ▪ 소프트웨어 유지보수 기록 점검 ▪ 소프트웨어 유지보수 검증	▪ 소프트웨어 유지보수 계획서 ▪ 소프트웨어 변경 기록 ▪ 소프트웨어 유지보수 기록 ▪ 소프트웨어 유지보수 검증 보고서

제 2 장 철도 현황 조사

제 1 절 철도 산업 현황

1. 국내 철도 시장 현황

1.1. 철도 업체 현황

국내 철도 산업은 철도엔지니어링, 철도건설, 철도제조업으로 구분할 수 있다. 국내 철도엔지니어링 기업은 철도분야뿐만 아니라 도로, 환경, 도시계획 등의 다양한 분야에 대해 담당하고 있다. 철도부문을 포함하고 있는 국내 엔지니어링 기업은 총 73개이며[5], 주로 국내철도 관련 타당성조사, 설계, 감리 등에 대한 사업을 담당하고 있으며 철도건설의 경우 국내 중대형 건설기업이 담당하고 있다.

엔지니어링 및 건설기업을 제외 한 본 연구와 관련 된 철도장비제조 및 관련 제품을 생산하는 업체 수는 많으나, 철도 제품만을 생산하는 업체 수는 많지 않기 때문에 정확한 종사자 규모를 파악하기는 어렵다.

우리나라의 철도장비제조 관련 사업체 수는 2014년도 경제총조사 기준으로 252개이며, 종사자 수는 6,745명으로 조사되었다. 기관차 및 기타 철도차량 제조업에 11개 사업체, 철도차량부품 및 관련 장치물 제조업에 241개 사업체가 있는 것으로 나타난다. 분야별 주요 업체는 [표 4]와 같다.

표 4 국내 주요 철도제조업체 현황

구 분	주요업체
철도 차량	▪ 현대로템(고속열차, 기관차, 전동차, 객차, 경전철, 자기부상열차 등) ▪ 경량전철 : 우진산전, 로윈 등 ▪ 모터카 : 대동ENT, 신성시스템 등 ▪ 화차 : 태양중공업, 고려차량
대차/연결기	▪ 다모이엠티, 영풍아이알, 성신RST , 벽진 등
구체 제작	▪ 동성중공업, ATX , 성진테크, 진양테크 등
의장 설비	▪ 한국화이바, 가본, 화인테크폴리머, 홍일기업, 진성산업 등
기장/제동	▪ 유진기공, 주은기공, 하나글로텍, 크노르코리아 등
전기/전장	▪ 갑을 오토텍, 샬롬엔지니어링, 한일전원공업, 우진테크 등
선로부품	▪ 삼표이엔시, AVT , 엔트켐, 협성실업, 팬드롤코리아 등
신호제어/관제	▪ 대아티아이, 그린시스템, 경인기술, 한터기술 등

종사자 규모별 사업체 수를 살펴보면 종사자수가 1~4명인 사업체가 106개(42%)로 가장 많고, 5~9명인 사업체가 64개(25.4%), 10~29명인 사업체가 51개(20.2%), 30~49명인 사업체가 15개(6%)로 50명 미만인 사업체가 전체 252개 중 236개(93.6%)로 대부분을 차지하고 있다. 이에 반해 100~999명 이상인 사업체는 7개(2.8%), 1,000명 이사인 사업체는 1개에 불과한 것으로 조사되었다.

5) 한국엔지니어링 협회에 철도면허 등록 기업 기준(2011년 12월 말)

종사자 규모별로 살펴보면 종사자 수 1,000명 이상인 사업체의 종사자 수는 2,170명으로 전체의 32.2%를 차지하고 있는 반면에 50명 미만인 사업체의 종사자 수는 2,196명으로 32.5%를 차지하고 있다.

표 5 2014년도 경제총조사 기준 철도장비제조업 종사자 규모별 현황

구 분	사업체 수(개)	비율(%)	종사자 수(명)	비율(%)
계	252	100	6,745	100
1~4명	106	42	293	4.3
5~9명	64	25.4	409	6.1
10~29명	51	20.2	933	13.8
30~49명	15	6	561	8.3
50~99명	8	3.2	509	7.5
100~499명	6	2.4	1,001	14.8
500~999명	1	0.4	869	13
1,000명 이상	1	0.4	2,170	32.2

자료: 통계청(http://kostat.go.kr), 2014년도 경제총조사

1.1.1. 철도차량제작회사 현황

종사자 수 1,000인 이상인 현대로템은 기존 3사(현대정공, 대우·현대중공업) 경쟁구도에서 1999년부터 현대로템으로 통합 된 이 후 국내 철도시장의 85%를 점유하는 독점 형태를 보이고 있다. 현대로템은 고속열차, 기관차, 전동차, 객차, 경전철, 자기부상열차 등을 제작하고 있다. 법적으로 진입장벽은 없으나 운영기관의 입찰평가기준이 신규 사업자에게 불리하여 현실적으로 진입이 곤란한 실정이다.

철도운영기관은 입찰 시 납품실적을 요구하기 때문에 납품실적이 없는 신규사업자의 진입이 원천적으로 어려운 상황이다. 하도급 등을 통하여 전동차를 직접 제작한 실적이 있는 경우에도 입찰평가 과정에서 실적으로 인정하지 않고 있다.

기존제품과의 엄격한 호환성을 요구하고 있는데, 호환이 가능하도록 차량을 제작 할 경우 기존업체 특허권 침해가 불가피한 실정이다. 이는 국내에서 호환성에 대한 명확한 개념정의 없이 기존제품과 거의 동일한 수준의 경우만 호환성을 인정하기 때문이다. 외국의 경우 전동차 공개경쟁입찰 시 호환성을 요구하는 예는 없는 것으로 조사되었다.

1.1.2. 철도부품회사 현황

내수시장은 도시철도차량 내구연한 연장 정책에 따라 대·폐차 물량이 줄어들고, 표준규격이 권장사항으로 규정되어 있어 소량 다품종 생산으로 인해 설비투자 및 생산비용이 증가하고 있다. 또한 철도 부품 수요처의 무리한 A/S 요구, 생산제품에 대한 빈번한 사양변경 및 설계변경 요구로 인해 채산성이 저하되어 대부분을 차지하고 있는 영세 업체의 생존이 위협 받고 있다.

철도 운영기관들이 철도부품 구매 시 저가 입찰방식을 선호하기 때문에 품질저하 및 기술개발 의욕이 저하되고 있다. 또한 신규 업체가 우수한 제품을 개발하여도 기존의 업체를 발주자가 선호하기 때문에 납품이 어려운 실정이다. 아울러 철도용품/시스템 입찰 시 요구되는 현장부설시험 및 인증 기반이 제대로 적용되고 있지 않아 개발된 제품의 상용화에 상당한 제약이 되고 있다.

기존에 도입된 차량들의 부품 국산화율이 낮았기 때문에 철도 운영기관에서 기존 외국부품을 그대로 사용하려는 경향이 많은 것으로 조사되었다. 종합제어장치, 자동운전장치, 인버터, 견인전동기, 보조전원장치 등 주요장치 대부분을 완제품형태로 수입하며, 인프라 부품(체결장치 등), 신호시스템 등은 해외 기술에 의존하고 있는 실정이다. 따라서 주요 부품 고장 시, 수리비용이 높고, 장기간이 소요될 뿐만 아니라, 수입 부품 단종· 고가 구입 등의 문제가 발생하고 있다.

1.2. 국내 철도 수·출입 현황

국내 철도운영기관이 발주한 철도제품 물량 중 철도차량 규모는 평균적으로 5천억 원 내외이며 2012년과 2014년의 경우 고속철도 사업으로 9천억 원 내외를 기록했지만 그 외에는 연도별로 편차가 크고, 차종별로도 편차가 큰 편이다. 이에 반해 철도유지보수 부품 발주 물량은 철도차량에 사용되는 전장품이 다수를 차지하고 있으며, 매년 2천억 원 내외를 유지하고 있으며 철도차량에 비해 꾸준하게 발주 물량이 유지되고 있다.

표 6 국내 운영기관 철도제품 구매 현황

(단위: 억 원)

구 분		2012년	2013년	2014년	2015년	2016년(추정)
철도차량	고속철도	6,691	-	7,441	-	3,093
	전기기관차	-	-	-	-	-
	디젤기관차	37	13	-	-	-
	간선형전기동차	-	-	-	-	-
	전동차	1,636	1,192	2,546	4,434	1,279
	경전철	-	1,066	-	-	-
	화차	-	-	-	-	-
	기타	-	-	-	105	-
철도유지보수부품		1,704	2,570	1,883	2,569	2,109
합계		10,068	4,841	11,870	7,108	6,481

자료: 국내 운영기관 발주물량 기준[6)]

철도차량 및 부품의 해외수출 규모는 2009년부터 증가 추세를 보이고 있으나, 2012년 기준 약 7억8천만 달러의 규모를 보인 이후 2013년에는 3억6천만 달러로 줄었다가 2015년 기준 다시 7억만 달러 수준으로 회복되고 있는 추세이다. 수입의 경우는 2013년 약 2억5천만 달러 수준으로 증가하였다가 2014년부터 감소 추세를 보이고 있으며 2016년 8천5백만 수준으로 줄어들었다.

표 7 철도차량 및 부품 수출입 현황

(단위: 백만 달러)

구 분	2010년	2011년	2012년	2013년	2014년	2015년	2016년
수 출	644	729	786	367	402	699	458
수 입	146	179	126	249	162	119	85
무역수지	498	550	660	118	240	580	373

자료: 한국무역협회(KITA)

6) KORSIA(한국철도차량산업협회) 발간 자료

해외시장에서 적용되고 있는 각종 규제 및 기술적 적용규격에 대응할 수 있는 기술개발이 부족하여 진입 가능한 시장이 제한적인 실정이다. 개발된 철도 제품에 대한 시험, 인증과 관련한 업체의 이중 부담에 비해 인센티브는 거의 없는 편으로 조사되었다. 철도 부품업체는 발주처의 검사와는 별도로 인증기관의 검사를 받을 경우 검사 시간 및 비용이 추가로 발생하나, 품질인증에 따른 인센티브가 충분치 않은 실정이다. 그리고 시험· 인증을 위한 국내 인프라가 충분치 않아 외국 인증기관에 비싼 비용을 지불해야 하는 사례가 발생하고 있다.

철도차량 및 부품 수출 실적을 자세히 살펴보면 인도, 터키 미국에 대한 수출 의존도를 보이고 있지만 인도를 제외하고는 점차 감소하고 있는 추세를 보이고 있다.

표 8 철도차량 및 부품 나라별 수출 실적

(단위: 천 달러)

구 분	2010년	2011년	2012년	2013년	2014년	2015년	2016년
총 계	644,071	728,856	785,683	366,577	401,810	699,339	452,170
인 도	74,815	112,745	44,536	7,975	117,083	222,476	159,663
이 집 트	0	0	4	10	26,651	164,413	33,220
브 라 질	79,299	1,775	16,633	561	14,038	116,325	155,990
터 키	107,242	221,630	126,113	136,377	69,419	54,229	26,966
뉴질랜드	20,944	66,046	55,063	382	520	50,673	33,740
미 국	69,561	74,726	52,659	105,751	45,263	33,451	12,255
홍 콩	113	58	125	156	162	24,864	13,170
이 란	35,051	68,071	0	23,311	20,190	13,838	-
중 국	12,740	5,699	840	584	8,453	2,972	1,875
우크라이나	-	-	276,618	2,158	360	2,245	1,331
카자흐스탄	84,932	-	-	-	2	2,136	-
일 본	11,531	1,232	4,893	5,029	1,487	1,401	5,708
말레이시아	25,961	422	361	1,648	391	1,365	985
파키스탄	-	-	31	10,113	313	1,040	-
대 만	563	2,427	1,151	3,140	4,643	582	4,878
태 국	219	2,207	4,774	120	1,257	981	1,213
멕 시 코	172	419	534	1,011	1,249	874	1,176

자료: 한국무역협회(KITA)

수입 실적의 경우는 2010년과 2011년도 프랑스, 2013년도 일본을 제외하고는 중국과 독일로부터 꾸준하게 수입을 하고 있다.

표 9 철도차량 및 부품 나라별 수입 실적

(단위: 천 달러)

구 분	2010년	2011년	2012년	2013년	2014년	2015년	2016년
총 계	145,982	177,166	125,622	248,635	162,136	119,261	66,394
중 국	11,312	11,713	20,576	22,046	33,484	19,491	22,984
독 일	35,631	26,244	9,781	28,112	23,069	18,694	9,304
체 코	6,872	7,920	7,468	16,361	17,748	15,088	9,624
프 랑 스	35,311	89,233	11,773	27,144	23,283	14,536	8,992
인 도	2,861	1,253	1,241	263	5,082	12,182	-
오스트리아	6,059	453	562	9,891	6,005	10,336	7,703
일 본	10,884	11,274	47,284	86,706	20,506	6,309	7,787

자료: 한국무역협회(KITA)

2. 해외 철도 시장 현황

2.1. 해외 업체 현황

독일 지멘스, 프랑스 알스톰이었다. 2015년 기준 해외 철도시장 점유율을 보면, 기존의 3강 체제에서 벗어나 중국중처그룹(CRRC)이 1위를 하고 있으며 캐나다의 롬바르디아와 미국의의 트리니티(Trinity)이 2, 3위를 기록하고 있다. 프랑스 알스톰은 4위 그리고 미국의 제너럴일렉트릭(GE)와 독일의 지멘스가 5위와 6위를 형성하고 있다. 현대로템의 경우 2012년도 기준 13위에서 2014년부터 10위로 올라갔으며 2.3%의 시장 점유율을 기록하고 있다.

중국의 경우 세계철도 시장의 최강자가 되기 위해 세계 1, 2위에 있던 중국북차집단공사(CNR)와 중국남차집단공사(CSR)를 합병하여 압도적인 세계 철도 시장의 최강자가 되었다. 이런 세계 시장 재편에 따라 일본의 히타치제작소는 같은 해에 이탈리아 핀메카니카 철도 사업부문 인수하여 8위의 시장 점유율을 기록하고 있다.

표 10 세계 철도차량시장 업체별 점유현황 (2015년 기준)

순 위	차량제조업체	국 가	2015년 철도 시장 점유율	
			매출액 (단위: 백만 유로)	점유율
1	CRRC	중국	17,033	33.4%
2	롬바르디아 (Bombardier)	캐나다	4,855	9.5%
3	트리니티 (Trinity)	미국	3,863	7.6%
4	알스톰 (Alstom)	프랑스	3,145	6.2%
5	제너럴일렉트릭 (GE)	미국	2,460	4.8%
6	지멘스 (Siemens)	독일	2,252	4.4%
7	스타들러 (Stadler)	스위스	1,566	3.1%
8	히타치 (Hitachi)	일본	1,470	2.9%
9	Greenbrier	미국	1,186	2.3%
10	현대로템	대한민국	1,174	2.3%
11	트랜스마쉬홀딩 (TMH)	러시아	1,157	2.3%
12	카와사키 (Kawasaki)	일본	983	1.9%
13	CAF	스페인	822	1.6%

자료: 한국철도차량산업협회, “철도차량” 협회지

2.2. 해외 시장 규모

세계 철도기술 시장은 2015년도 기준으로 토목, 건축 등 인프라 구조물 시장을 제외하고 1,620억 유로(약 232조원) 규모이며, 이 중 철도차량 시장 규모는 유지보수 분야와 함께 60%(976억 유로)를 차지하고 있다. 전력, 궤도, 신호, 통신 등 철도시스템(E&M) 분야는 40%(646억 유로) 규모를 차지하고 있다.

표 11 분야별 철도시장 규모 (2015년 기준)

분 야		규모(단위: 억 유로)	비 중(%)
철도차량	철도차량	488	30
	차량유지보수	488	30
철도시스템 (E&M)	전력, 궤도	468	29
	신호, 통신	178	11
합 계		1,622	100

자료: KORSIA(한국철도차량산업협회) 발간 자료, 2016년도

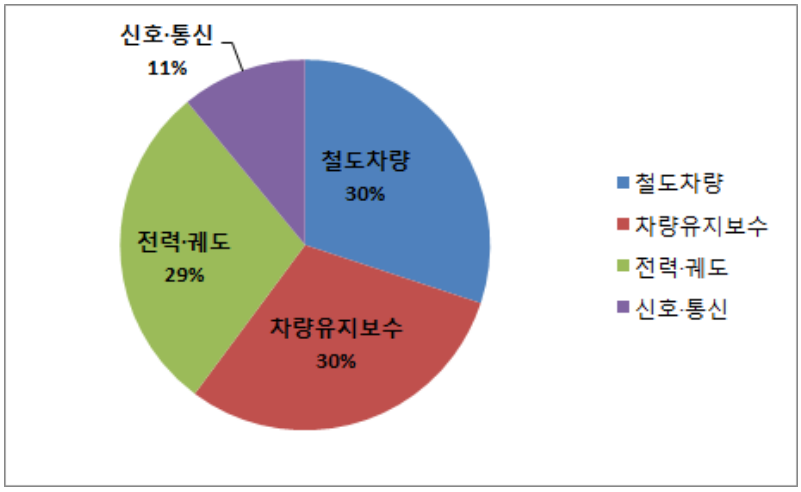


그림 4 분야별 철도시장 비중

2.2.1. 철도차량 시장규모

전체 철도차량의 시장규모는 488억 유로 규모이며, 열차 종류별로는 화차 119억 유로, 전동차 74억 유로, 고속전철 73억 유로, 메트로 56억 유로, 디젤기관차 50억 유로, 전기기관차 45억 유로, 객차 40억 유로, 경전철 21억 유로, 디젤동차 11억 유로 순으로 조사되고 있다. 철도차량 시장규모는 신규 차량과 개량 차량을 모두 포함하며, 차량 유지보수와 관련 된 시장 규모도 포함하고 있다.

표 12 차종별 시장 규모

분 야	규모 (단위: 억 유로)	비 중(%)
화 차	119	26
전 동 차	74	15
고속전철	73	15
메 트 로	56	11
디젤기관차	50	10
전기기관차	45	9
객 차	40	8
경 전 철	21	4
디젤동차	11	2
합 계	488	100

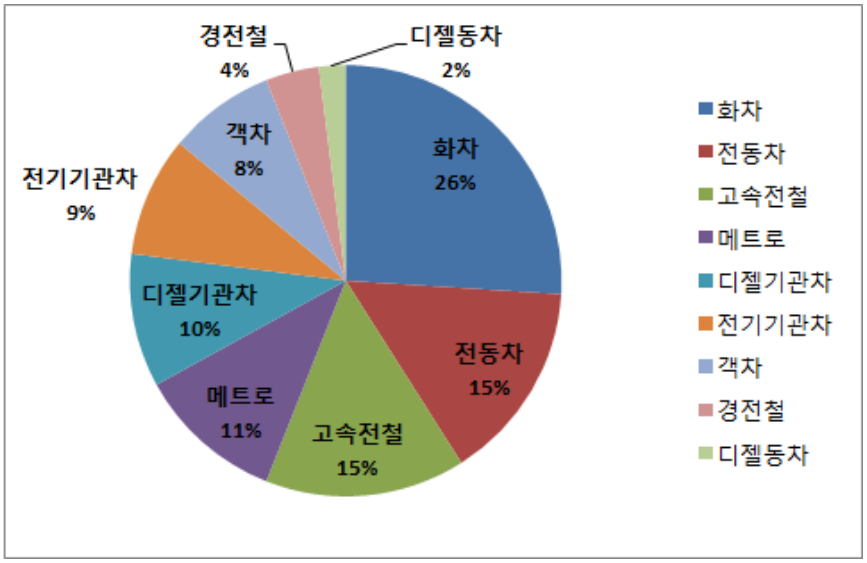


그림 5 차종별 시장 비중

지역별로는 아시아 175억 유로, 서유럽 89억 유로, CIS 87억 유로, 북미 72억 유로 동유럽 21억 유로, 아프리카/중동 18억 유로, 중남미 15억 유로, 오세아니아 12억 유로의 순이다. 중국, 베트남 등의 아시아 지역 철도 운영의 확대로 서유럽과 북미 등 2000년대 초반 강세 지역을 제치고 가장 높은 시장 규모를 차지하고 있다.

표 13 지역별 시장 규모

지 역	규모 (단위: 억 유로)	비 중(%)
아 시 아	175	36
서 유 럽	89	18
CIS	87	18
북 미	72	15
동 유 럽	21	4
아프리카/중동	18	4
중 남 미	15	3
오세아니아	12	2
합 계	488	100

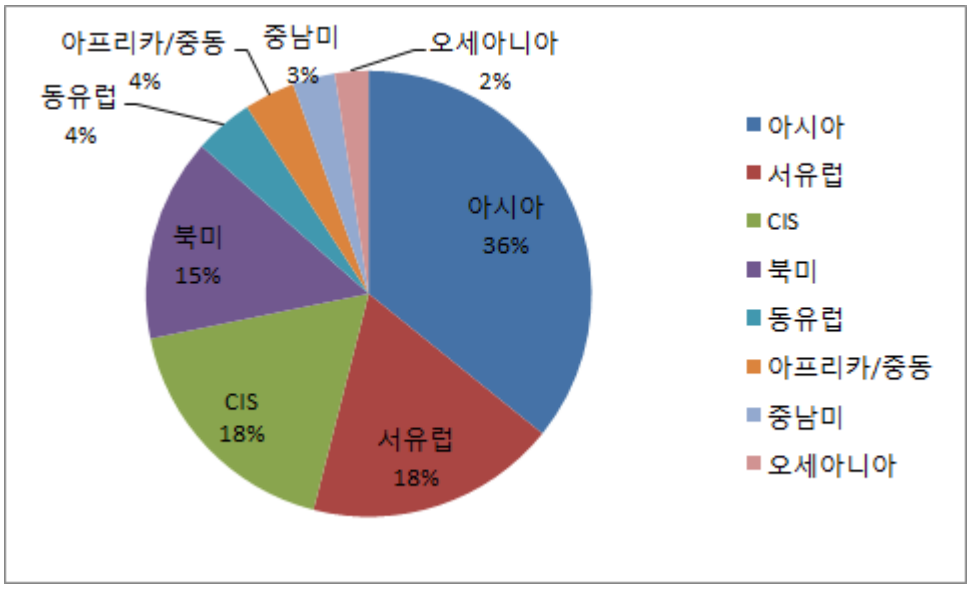


그림 6 지역별 시장 비중

3. 국내 철도 시장 구조

3.1. 철도 사업 발주 기관

3.1.1. 국토교통부

국토교통부는 직접적으로 철도와 관련 된 기획 및 인프라 관련 업무를 수행하는 정부기관으로 2차장 산하의 교통물류실과 철도국에서 주요 업무를 수행한다. 철도국에서 기획재정부와 협의하여 국책 사업을 진행하면, 산하 기관인 한국도시철도공단에서 인프라 구축을 수행하고 완공 된 노선은 한국철도공사(KORAIL)가 정부로부터 위임 받아 운영한다.

고속전철이나 화물 운반용 전국단위 시설의 경우 한국철도공사가 직접 운영하고, 지방단위 노선이나 민간기업 개발 노선의 경우 별도의 운영 주식회사를 설립하여 운영권을 위임한다.

그림 7 도시철도 및 광역철도 운영기관

3.1.2. 철도기술연구원

사단법인 한국철도기술연구원은 철도분야의 기술개발 및 정책연구를 통한 철도교통의 발달과 철도산업의 경쟁력 강화를 목적으로 설립된 대한민국 유일의 철도종합연구기관이다. 과기정통부 산하 기타공공기관으로 지정되어 있으며 아래의 업무를 중심으로 국내철도연구 관련 사업을 발주한다.

- 고속철도, 일반철도, 도시철도 및 경량전철 시스템 연구개발
- 차세대 대중교통시스템 연구개발
- 철도안전, 표준화, 철도정책 및 물류 기술 연구개발
- 남북철도 및 대륙철도 연계기술 연구개발
- 철도 핵심원천기술 연구개발
- 기술정책 수립 지원, 시험평가 인증, 인력양성, 기술지원 등

그림 8 한국철도기술연구원 조직도

3.2. 발주구조

국가계획에 반영되어 있는 철도망의 경우 한국도시철도공단이나 지자체에서 예비타당성조사를 요청하면, 국토교통부의 교통물류실과 철도국에서 이를 반영하여 우선순위를 결정하고 우선순위가 높은 경우 기획재정부에 예비타당성 조사를 요구한다.

기획재정부는 국토교통부의 요청을 받아 재정위원회를 개최하여 추진여부를 결정한다. 추진이 되면 예비타당성 조사를 시행하고 타당성이 있다고 결론이 나면 국토교통부에서 역의 위치나 길이, 편의 시설 등의 기본계획을 수립하고 한국도시철도공단에 기본설계, 실시설계 및 시공을 위임한다.

한국도시철도공단에서는 시공을 위해 발주를 내고 업체를 선정하여 시공을 진행한다. 시공이 완료되고 운영 허가가 떨어지면 한국철도공사에서 운영하거나 지자체 또는 별도의 운영주식회사를 설립하여 운영을 위임한다.

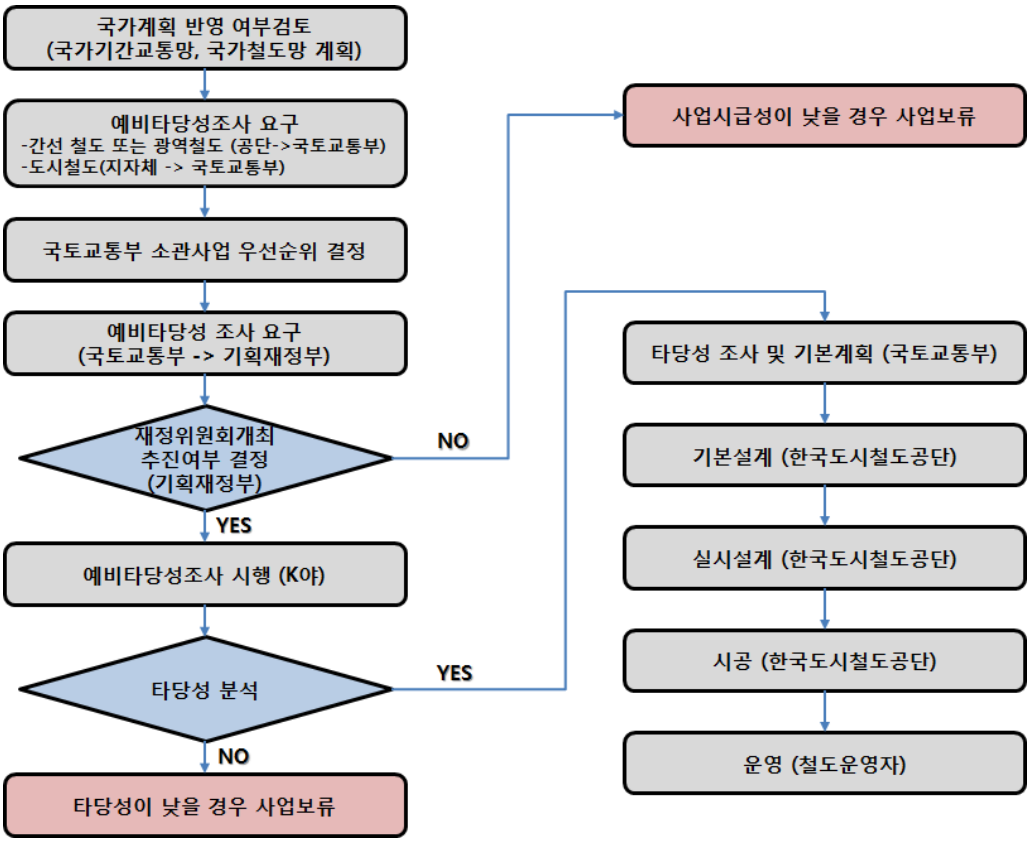


그림 9 국내 철도 사업 발주 절차

이와는 다르게 한국철도기술연구원에서는 정부가 제시하는 기술 계획에 따라 자체 연구 인력 제공과 더불어 시장 활성화를 위해 과제를 발주하며, 대형 규모의 R&D가 아닌 경우는 대부분 중소규모의 업체가 수주하여 업무를 재분배 하는 경우가 많다. 대부분의 철도 관련 사업이 정부 주도로 이루어지며, 기업차원에서 수익을 따져 자체 연구 등을 위해 별도의 발주를 하는 경우는 극히 드물다.

제 2 절 철도 관련 법 체계

1. 철도 관련 법

1.1. 제·개정 연혁

철도 관련 주요 법은 1960년대 철도법으로 제정되어 적용되어 오다가 2004년도 12월 철도사업법 제정으로 철도법이 폐지되었고, 지하철도건설촉진법이 1979년 4월에 제정되어 1990년 12월 도시철도법으로 개정되었으며 2007년 7월에는 표준규격, 안전기준, 성능시험 등의 항목이 추가되어 개정되었다.[7)]

이 후 철도산업 구조 개혁으로 철도의 건설과 운영이 분리됨에 따라 기존의 철도안전 관리체계 정비의 필요성이 제기되고, 고속철도 개통, 도시철도 및 일반철도 건설 확대 등으로 철도시설, 차량 및 철도운영에서의 안전 위협요소가 증가되고, 다양한 철도 관련 사고에서 드러난 현행 차량 및 안전관리체계의 한계 및 문제점에 대한 보완이 필요하게 되면서 기존 철도법이 폐지되고 철도건설법, 철도사업법, 철도안전법 등의 3개 법으로 정비되었다.

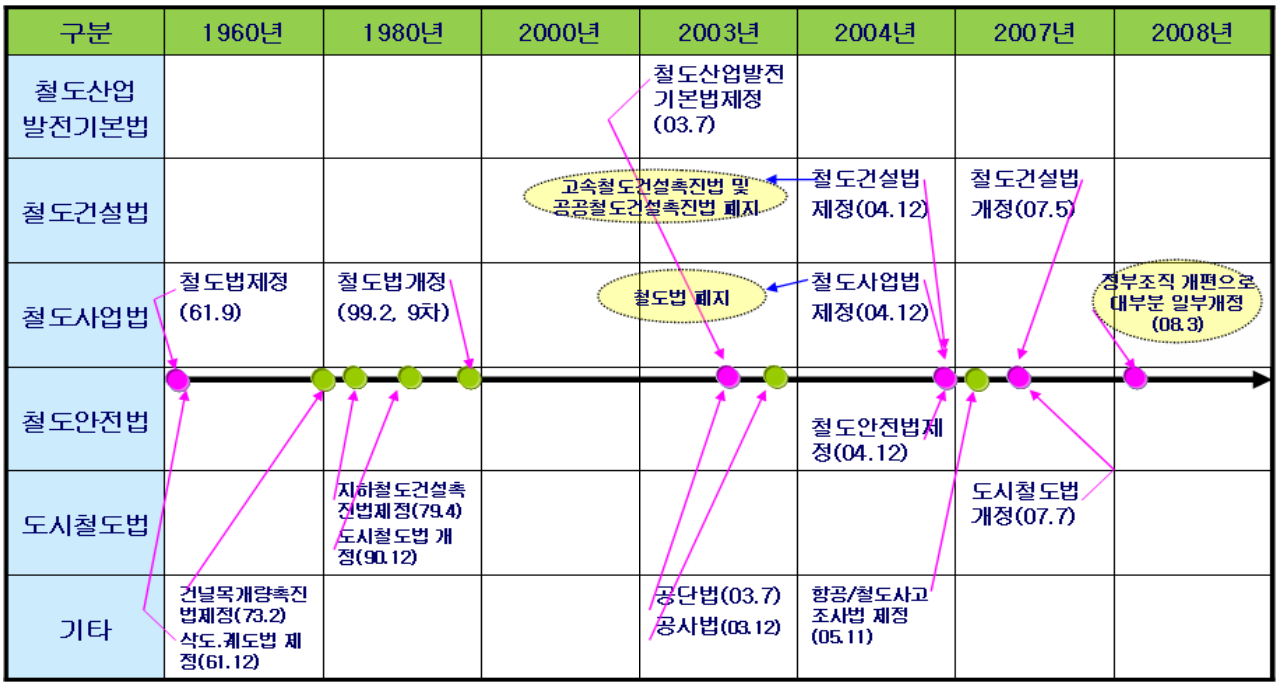


그림 10 철도 관련 법령의 변천 연혁

7) 국토교통부, 철도 관련 법제 개선 연구, 2009

1.2. 철도 관련 법

철도관련법으로는 철도산업발전기본법, 철도건설법, 철도안전법, 철도사업법, 도시철도법, 궤도운송법 등이 있다. 철도산업발전기본법이 개념적으로 철도 관련법의 상위에 있어, 철도안전법(철도산업발전기본법 제14조)의 근간법으로 위치하지만, 철도건설법, 철도사업법, 도시철도법, 궤도운송법 등과는 독립적이다. 철도 관련법 적용 대상은 다음과 같다.

- 지역 간 철도(일반, 고속) 및 광역철도: 철도산업발전기본법, 철도건설법, 철도사업법, 철도안전법
- 도시철도: 도시철도법, 철도안전법, 철도사업법, 궤도운송법
- 삭도 및 궤도: 궤도운송법

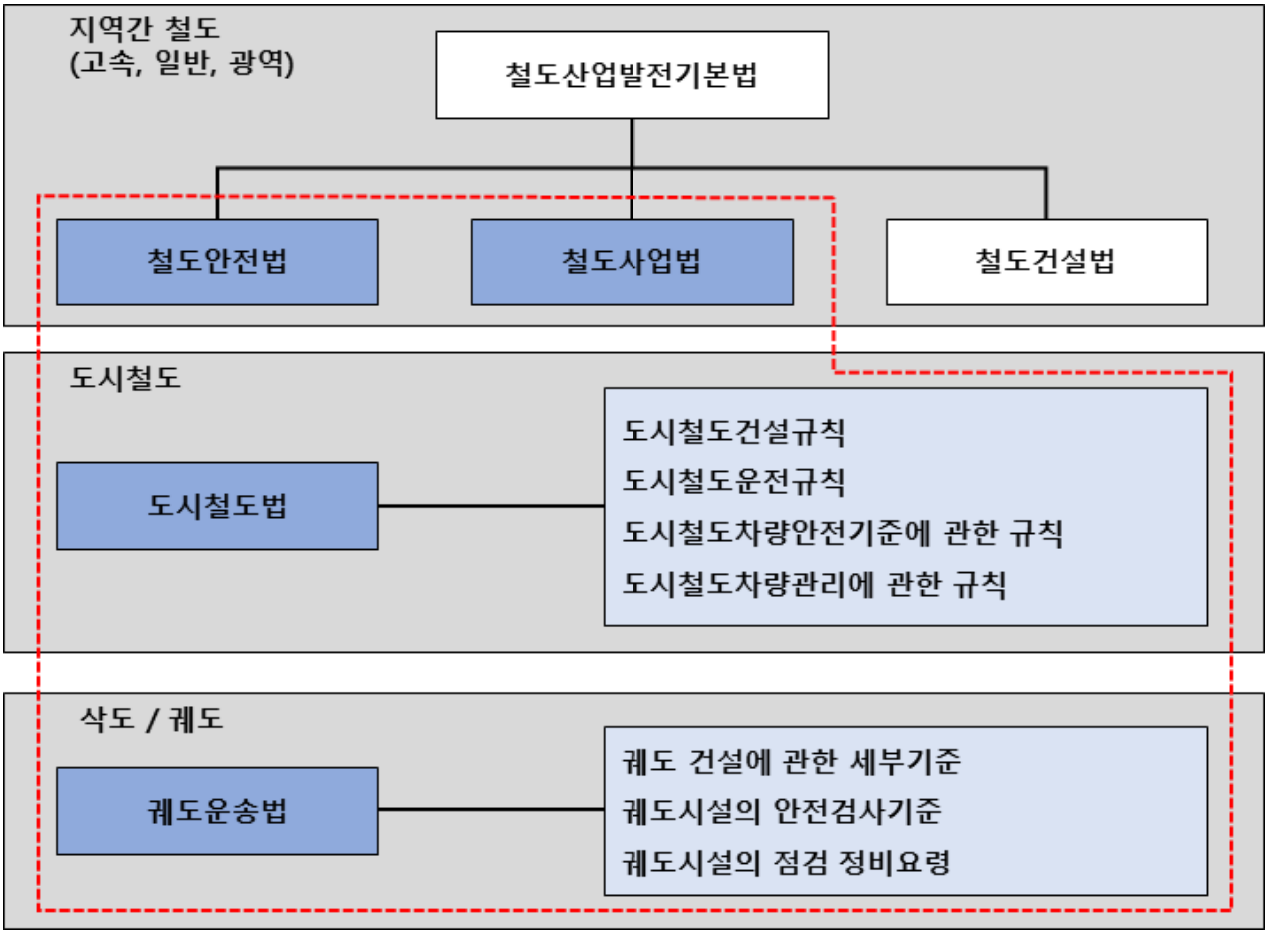


그림 11 철도 안전 관련 법 체계

1.2.1. 철도산업발전기본법

타 법령 또한 철도산업과 철도산업발전시책과의 관련성이 낮거나 미흡하며, 철도산업발전시책과의 관련성 분석결과에 따르면 철도산업구조의 선진화와 철도시설 확충 이외에는 시책과 연계성이 없다.

표 14 철도산업발전기본법 개요

구 분	주요 내용
법 령	철도산업발전기본법
목 적	철도산업의 경쟁력을 높이고 발전기반을 조성하여 철도산업의 효율성 및 공익성을 향상시키고 국민경제의 발전에 공헌
적용범위	국가철도망(일반철도, 광역철도)
법 률	**주요 내용**
제 2 장	- 철도산업 발전기반의 조성
제 1 절	- 철도산업발전 기본계획 수립과 추진체계
제 2 절	- 철도산업의 육성 · 철도시설투자 확대 · 철도산업의 지원 · 철도산업의 인력양성 및 철도기술 진흥 · 철도산업의 정보화 추진 · 국제협력 및 해외진출
제 3 장	- 철도안전 및 이용자 보호 · 철도 안전, 철도 서비스의 품질 개선 등 · 철도이용자의 권익보호
제 4 장	- 철도산업구조개혁의 추진 · 철도산업의 상하분리 · 철도시설 : 국가(한국철도시설공단) · 철도운영 : 시장경제원리(한국철도공사) · 자산 · 부채 · 인력처리 · 철도시설 관리권 · 공익적 기능 유지

1.2.2. 철도건설법

철도건설법은 철도시설과 철도의 개발과 이용 중 역세권 개발과는 정합성이 존재하나, 철도산업발전 시책 중 철도시설확충과 철도망의 기능강화와는 관련성이 낮다.

표 15 철도건설법 개요

구 분	주요 내용
법 령	철도건설법
목 적	이 법은 철도망의 신속한 확충과 역세권개발사업의 활성화를 위하여 철도망구축계획의 수립, 철도건설, 역세권개발에 관한 사항을 규정하여 철도교통망의 효율적인 확충과 공공복리의 발전에 이바지함을 목적으로 한다.
적용범위	국가철도망(일반철도, 광역철도), 도시철도
법 률	**주요 내용**
제 2 장	- 철도의 건설
제 1 절 제 2 절 제 3 절	- 국가철도망구축계획 - 철도의 건설체계 - 철도건설 비용부담
제 3 장	- 역세권 개발 · 역세권개발구역 지정 · 역세권개발사업의 시행 등

1.2.3. 철도사업법

철도영업에 관한 내용과 정합성을 이루고 있으나 제3자 시장참여에 관한 조항이 전무하다.

표 16 철도사업법 개요

구 분	주요 내용
법 령	철도사업법
목 적	이 법은 철도사업에 관한 질서를 확립하고 효율적인 운영여건을 조성하여 철도사업의 건전한 발전과 철도이용자의 편의를 도모하고 국민경제의 발전에 이바지함을 목적으로 한다.
적용범위	국가철도망(일반철도, 광역철도), 도시철도
법 률	**주요 내용**
제 2 장 제 3 장 제 4 장 제 5 장	철도사업의 관리 철도서비스 향상 전용철도 국유철도시설의 활용 · 지원 등

1.2.4. 철도안전법

철도안전법은 철도산업과의 정합성이 높지만, 철도정책과 연계한 철도차량 기술개발 및 현장에 대한 침투성이 낮다.

표 17 철도안전법 개요

구 분	주요 내용
법 령	철도건설법
목 적	이 법은 철도안전을 확보하기 위하여 필요한 사항을 규정하고 철도안전관리체계를 확립하여 공공복리의 증진에 기여함을 목적으로 한다.
적용범위	국가철도망(일반철도, 광역철도), 도시철도
법 률	**주요 내용**
제 2 장	철도안전관리체계
제 3 장	철도종사자의 안전관리
제 4 장	철도시설 및 철도차량의 안전관리
제 5 장	철도차량운행안전 및 철도보호
제 6 장	철도사고조사 · 처리
제 7 장	철도안전기반구축

1.2.5. 도시철도법

도시철도법은 각 철도산업과 정합성이 존재하며 건널목개량촉진법 또는 법률이 목적한 바와 정합성을 확보하고 있다.

표 18 도시철도법 개요

구 분	주요 내용
법 령	도시철도법
목 적	이 법은 도시교통권역의 원활한 교통소통을 위하여 도시철도의 건설을 촉진하고 그 운영을 합리화하며, 도시철도차량 등을 효율적으로 관리함으로써 도시교통의 발전과 도시교통 이용자의 안전 및 편의증진에 이바지함을 목적으로 한다.
적용범위	도시철도
법 률	**주요 내용**
	도시철도기본계획수립체계 도시철도사업의 관리(면허제) 역세권 개발 및 지하보상 등 도시철도의 건설 및 운영을 위한 자금조달, 정부지원

1.2.6. 법령과 철도산업과의 정합성

철도 관련 법령들과 철도산업과의 정합성을 살펴보면 철도산업발전기본법은 철도시설과 관련이 있고, 철도건설법은 철도시설과 철도의 개발 이용 산업에 관련이 있다. 철도안전법은 철도사업, 철도운영, 철도차량 기술에 관련이 있으며, 철도사업법은 철도운송의 철도사업 분야에 관련이 있다. 도시철도법은 철도사업, 철도운영 철도시설, 철도의 개발과 이용에 모두 관련이 있고, 건널목개량촉진법은 철도시설과 밀접한 관련이 있다.

표 19 법령과 철도산업과의 연관관계

산 업 / 법 령	철도운송		철도시설	철도차량 (차량기술개발 포함)	철도의 개발과 이용
	철도사업	철도운영			
철도산업발전기본법	×	△	○	△	×
철도건설법	×	×	●	×	○(역세권)
철도안전법	○	●	△	○	×
철도사업법	●	△	×	△	×
도시철도법	○	○	○	△	○
건널목개량촉진법	×	×	●	×	×

철도 차량 기술 개발기술과 관련해서 철도 안전과 밀접한 철도안전법과 도시철도사업법을 상세하게 살펴보고 두 사업법의 내용을 비교해보도록 한다.

2. 철도안전법

우리나라 법은 아래 [그림 12]와 같이 헌법을 근간으로 하여 국회에서 필요한 법률을 제정하고 이를 시행하기 위해 행정부의 수반인 대통령이나 총리 또는 각 기관의 장이 (명)령을 제정하여 공포한다. 법률에서 행정규칙으로 갈수록 범위와 내용이 구체적이며 상세한 내용을 다루게 된다. 반대로 행정규칙에서 법률로 갈수록 강한 효력을 가진다.

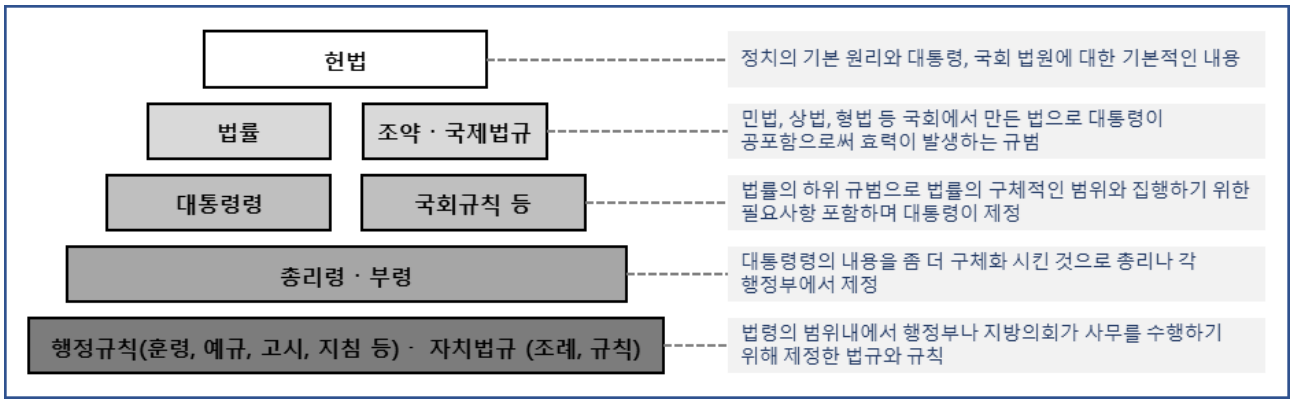


그림 12 우리나라 법령의 체계

철도안전법은 우리나라의 법령 체계에 맞게 철도안전법 밑에 시행령과 시행규칙을 두고 있으며 형식승인을 수행하기 위한 기술기준을 두고 있다. 철도안전법은 기본적으로 철도산업발전기본법에 명시된 철도를 대상으로 하나, 철도안전에 관하여 다른 법률에 특별한 규정이 있는 경우를 제외하고는 철도안전법 규정을 따르도록 하고 있다.

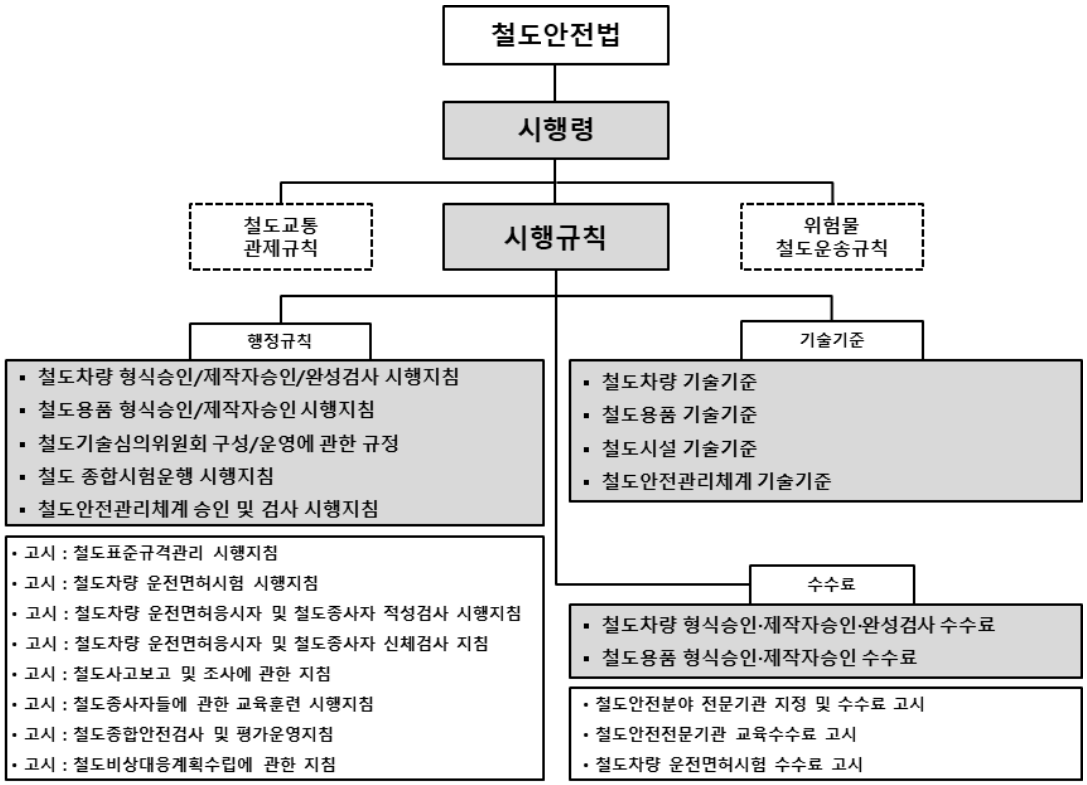


그림 13 철도안전법 체계

2.1. 철도안전법 주요 내용 및 규정

철도안전법 제2장 철도안전관리체계에서는 철도안전종합계획수립과 시행에 대한 사항과 철도 운영자 등이 준수해야 할 안전관리규정, 비상대응계획, 종합안전심사 제도에 대한 사항을 규정하고 있다.

철도안전법 제3장 철도종사자의 안전관리에서는 철도차량운전면허 취득과 관련된 사항을 규정하고 있으며, 운전 및 관제 업무 종사자 관리 등을 규정하고 있다.

철도안전법 제4장 철도시설 및 철도차량의 안전관리에서는 철도시설 및 차량의 안전기준 준수, 철도용품 품질인증 및 표준화, 철도차량에 대한 성능시험, 제작검사 및 내구연한, 종합시험운행에 관한 사항을 규정하고 있다.

- 철도시설관리자는 철도시설안전기준에 맞춘 철도시설 설치 및 유지보수 수행
- 철도운영자는 철도차량안전기준에 맞춘 철도차량 운행 및 유지보수 수행
- 철도용품 품질인증 및 표준화는 권장사항으로 운영 중
- 철도차량 제작자는 철도차량의 성능과 구조· 장치의 형상 및 규격 등이 안전 및 기능 확보에 적합한지 여부에 대해 국토해양부장관이 실시하는 성능시험을 받도록 규정
- 철도차량 제작자는 철도차량의 제작에 착수한 때부터 철도차량의 품질 및 안전성이 확보되고 있는지의 여부에 대하여 국토해양부장관이 실시하는 제작검사를 받도록 규정
- 철도 운영자 등은 사용내구연한을 초과한 철도차량을 운행할 수 없으나, 정밀진단을 받아 안전운행에 적합하다고 인정되는 경우에는 사용내구연한을 연장할 수 있도록 규정
- 철도시설관리자는 철도노선을 새로 건설하거나 기존노선을 개량하여 운영하고자 할 때에는 정상운행하기 전에 종합시험운행을 실시하도록 규정

철도안전법 제5장 철도차량운행안전 및 철도보호에서는 열차운전 및 관제 관련종사자 의무, 열차이용자 의무, 위험물 운송, 철도시설 보호 등 에 관한사항을 규정 하고 있다.

- 열차운전 및 관제 관련 종사자는 업무 수행 중 음주 등을 금지토록 규정
- 위험물은 철도 수송을 할 수 없으며, 철도 수송이 가능한 위험물에 대해서는 안전에 관한 사항을 준수토록 규정
- 철도차량의 안전운행 및 철도보호를 위하여 철도경계선으로부터 3 0 미터 이내의 지역인 철도보호지구에서의 행위 관련 사항을 규정

○ 철도보호 및 질서유지를 위하여 여객열차 이용자 및 일반인들의 금지행위 관련 사항 및 철도종사자의 직무상 지시 준수 등을 규정

철도안전법 제6장 철도사고조사 및 처리에서는 철도사고 발생 시 철도 운영자 등의 사고처리를 위한 조치 및 보고에 관한 사항을 규정하고 있다.

철도안전법 제7장 철도안전기반 구축에서는 철도안전기술 진흥, 전문기관 육성, 안전지식의 보급, 안전정보 종합관리, 철도안전 관련기관에 대한 재정지원 등의 내용을 규정하고 있다.

철도안전법 제8장 보칙에서는 철도안전관련 제도와 관련된 권한의 위임 및 위탁, 수수료 등에 관한 사항을 규정하고 있다. 제9 장 벌칙에서는 벌칙, 양벌규정, 과태료 등의 사항을 규정하고 있다.

2.2. 하위법령 위임사항

철도안전법 시행령과 시행규칙 등의 위임사항은 아래의 내용과 같다. 이를 기반으로 해서 법령을 수행하게 된다.

○ 철도안전법 시행령 위임사항

- 안전운행 또는 질서유지 철도종사자의 정의
- 철도안전종합계획의 경미한 변경 내역 및 시행계획의 수립절차 등
- 종합안전심사의 시기, 방법 · 절차, 평가 등
- 철도차량 운전면허 종류, 운전면허를 받을 수 없는 신체장애인 등
- 적성검사기관의 지정절차, 지정기준 등
- 전문교육훈련기관의 지정절차, 지정기준 등
- 운전면허 갱신, 취득절차 일부 면제 등
- 신체검사를 받아야 하는 철도종사자
- 철도안전기준을 준수해야하는 철도차량의 구조 및 장치
- 품질인증절차를 면제할 수 있는 철도용품
- 품질인증기관의 지정절차, 지정기준, 업무범위 등
- 성능시험을 면제할 수 있는 철도차량
- 성능시험기관의 지정절차, 지정기준, 업무범위 등
- 제작검사를 면제할 수 있는 철도차량
- 제작검사기관의 지정절차, 지정기준, 업무범위 등
- 정밀진단기관의 지정절차, 지정기준, 업무범위 등
- 음주 등이 제한되는 철도종사자
- 탁송 및 운송금지 대상 위험물 또는 운송취급주의 위험물 등
- 철도보호지구안에서의 행위 신고절차, 안전운행 저해행위, 안전조치, 행위제한에 따른 손실보상 등
- 철도보호 또는 질서유지 등을 위해 위반자 또는 물건에 대한 퇴거지역의 범위
- 철도사고 등의 발생 시 조치사항 및 국토해양부장관에게 즉시 보고하여야 하는 철도사고 등
- 철도안전 전문 인력의 구분, 자격기준 등
- 보고 및 검사에 관한 절차
- 위임, 위탁되는 권한 내용 및 대상 기관
- 과태료의 부과기준

○ 철도안전법 시행규칙 위임사항

- 안전관리규정의 내용, 경미한 변경
- 비상대응계획의 내용, 경미한 사항의 변경, 비상대응훈련의 평가

- 종합안전심사의 시기변경 등 (시행령 위임)
- 교육훈련 철도차량 등의 표지 (시행령 위임)
- 운전면허의 종류에 따라 운전할 수 있는 철도차량의 종류 (시행령 위임)
- 신체검사 방법 · 절차 · 합격기준, 지정취소 · 정지 등
- 적성검사 방법 · 절차 및 합격기준 등
- 적성검사기관의 지정절차 및 세부지정기준(시행령 위임), 지정취소 · 업무정지 (법 위임)
- 교육훈련기관의 지정절차 및 세부지정기준(시행령 위임), 지정취소 · 업무정지 (법 위임)
- 운전면허시험의 과목 및 절차
- 운전면허증의 교부, 기재사항 변경, 운전면허의 갱신 안내통지, 절차 및 갱신에 필요한 경력 등
- 운전면허의 취소 및 효력정지처분의 세부기준, 통지 등
- 운전업무수행의 필요요건 등
- 관제업무수행의 필요요건 등
- 운전업무종사자 등에 대한 신체검사의 실시, 적성검사의 실시 등
- 철도종사자의 안전교육 방법 등
- 품질인증절차의 면제범위 등 (시행령 위임)
- 철도기술심의위원회의 설치
- 품질인증의 신청, 인증품의 표시, 품질인증의 대상 및 기준 등
- 품질인증기관의 지정절차 및 세부지정기준 (시행령 위임), 지정취소 및 지위승계의 신고 (법 위임)
- 품질인증의 사후관리를 하는 자의 증표
- 철도표준규격의 제정 등
- 성능시험의 대상, 기준, 절차, 경미한 변경 (법 위임), 성능시험의 면제범위(시행령 위임)
- 성능시험기관의 지정절차, 세부지정기준 (시행령 위임), 지정취소(법 위임)
- 제작검사 대상, 기준, 절차, 경미한 변경 (법 위임), 제작검사 면제범위(시행령 위임)
- 제작검사기관의 지정절차, 세부지정기준 (시행령 위임), 지정취소(법 위임)
- 철도차량의 사용내구연한, 정밀진단의 실시 등
- 정밀진단기관의 지정절차, 세부지정기준 (시행령 위임), 지정취소(법 위임)
- 종합시험운행의 실시시기 · 절차 등
- 음주제한의 기준, 위해물품 휴대금지 예외, 위해물품의 종류, 여객출입금지장소, 여객열차 안에서의 금지행위, 폭발물 등 적치금지 구역, 적치금지 폭발물, 출입 금지 철도시설, 열차운행에 지장을 줄 수 있는 유해물, 질서유지를 위한

금지행위, 철도 사고 등의 보고
- 철도안전 전문 인력의 교육훈련, 자격부여 절차(시행령 위임)
- 검사공무원의 증표

○ 철도차량안전 기준에 관한 규칙 위임 사항
- 철도차량의 화재안전기준, 전기안전기준 관련 사항
- 철도차량한계, 축 중, 중량분포 등에 관한 사항
- 철도차량 주행안전기준에 관한 사항
- 철도차량의 충돌안전기준에 관한 사항
- 철도차량의 차체 및 장치에 대한 기술기준
- 철도차량의 주행장치에 대한 기술기준
- 철도차량의 제동장치에 대한 기술기준
- 철도차량 추진 및 보조전원장치에 대한 기술기준
- 철도차량 차상신호장치 및 운전자보안장치에 대한 기술기준
- 철도차량 종합제어장치에 대한 기술기준
- 철도차량 연결 장치에 대한 기술기준
- 철도차량 관련 기타 장치에 대한 기술기준
- 철도차량 유지관리계획 수립, 유지보수 시행 및 기록관리 등에 관한 철도차량 관리자의 의무

○ 철도시설안전기준에 관한 규칙 위임사항
- 철도시설 설계 시 철도시설의 신설· 유지· 보수 및 운영 등의 전반에 대하여 안전성 분석 시행 의무화
- 선로 안전에 관한 일반기준 및 시설· 설비에 관한 기술기준
- 노반 안전에 관한 일반기준 및 시설에 관한 기술기준
- 철도교량 안전에 관한 일반기준 및 시설에 관한 기술기준
- 터널 안전에 관한 일반기준 및 시설· 설비에 관한 기술기준
- 역 시설 안전에 관한 일반기준 및 시설· 설비에 관한 기술기준
- 철도건널목 안전에 관한 일반기준 및 설비에 관한 기술기준
- 전철전력 안전에 관한 일반기준 및 설비에 관한 기술기준
- 철도신호설비 및 통신설비 안전에 관한 일반기준, 구조 관련 기술기준
- 철도시설 유지관리 계획수립·시행 및 기록관리 등 철도시설 관리자 의무

○ 철도차량운전 규칙 위임사항

- 철도차량 운행과 관련된 종사자에 대한 교육 훈련에 관한 사항
- 열차에 탑승해야 하는 철도 종사자에 관한 사항
- 차량의 적제 제한, 특대화물 수송에 관한 사항
- 열차 조성 및 제동에 관한 사항
- 열차 운전, 속도 및 속도제한에 관한 사항
- 열차 간 안전거리 확보를 위한 폐색방식 유형 및 준수사항
- 신호기의 유형 및 준수사항
- 운전관련 표지설치에 관한 사항 등

2.3. 철도안전법 및 하위 법령의 연혁

2.3.1. 철도안전법

철도안전법은 철도산업구조개혁의 추진과 고속철도의 개통 등 철도분야에서 기술적, 사회적 안전을 위협하는 요소가 증가함에 따라 철도에서의 안전관리체계를 구축하기 위해 2004년에 제정되었다. 철도안전법 제정문에 명시된 주요 사항은 다음과 같다.

- 철도안전종합계획 및 연차별 시행계획의 수립・시행(법 제5조 및 제6조)
- 안전관리규정 제정 및 비상대응계획 수립・시행(법 제7조 및 제8조)
- 철도차량운전업무 종사자의 요건(법 제10조 및 제21조)
- 철도시설 및 철도차량의 안전기준(법 제25조 및 제26조)
- 철도용품의 품질인증제도 도입근거 마련(법 제27조)
- 철도차량의 성능시험 및 제작검사(법 제35조 및 제36조)
- 열차 안에서의 유해물질 휴대금지(법 제42조)

철도안전법은 2009년에 제80조 양벌규정의 내용을 변경하는 일부개정이 이루어 졌다. 당초 양벌규정은 영업주가 종업원 등에 대한 관리·감독상 주의의무를 다하였는지에 관계없이 영업주를 처벌하도록 하고 있어 책임주의 원칙에 위배될 소지가 있으므로, 영업주가 종업원 등에 대한 관리·감독상 주의의무를 다한 때에는 처벌을 면하게 함으로써 양벌규정에도 책임주의 원칙이 관철되도록 하고 있다.

2.3.2. 철도안전법 시행령

철도안전법이 제정되어 철도 운영자 등에 대한 종합안전심사, 철도차량운전면허, 철도차량에 대한 성능시험・ 제작검사 등 철도에서의 안전체계 강화를 위한 제반제도가 도입됨에 따라 동법에서 위임된 사항과 그 시행에 관하여 필요한 사항을 정하기 위해 2005년에 철도안전법 시행령을 제정하였다. 철도안전법 시행령의 주요 내용은 다음과 같다.

- 철도 운영자 등에 대한 종합안전심사 (영 제6조 내지 제9조)
- 철도차량운전면허의 종류 구분 (영 제11조)
- 철도사고조사위원회의 조사대상인 철도사고 등 (영 제57조 및 제58조)

2009년 6월 일부 개정에서는 철도차량 제작검사기관 지정 기준을 변경하였다. 이전 시행령에서는 철도차량 등 제작검사기관의 지정기준에 제작검사수행실적이 포함되어 있어 신규기관의 진입이 원천적으로 봉쇄되는 불합리한 측면이 있었기 때문에 이를 개선

하기 위하여 제작검사수행실적 요건 대신 「국가표준기본법」제23조에 따라 검사기관으로 인정받을 것을 지정기준에 포함시키도록 변경하였다.

2009년 12월 일부 개정에서는 철도차량 제작검사기관 지정 시 「국가표준기본법」에 따라 검사기관으로 인정받을 요건을 폐지하여 신규 제작검사기관의 참여가 용이하도록 진입부담을 해소하고, 철도시설의 보호 및 차량의 안전운행을 위하여 철도보호지구 안에서의 행위에 대한 신고 및 안전조치 등에 관한 세부사항을 국토해양부장관이 고시하도록 신설하였다.

2011년 2월 일부 개정에서는 인·허가제도 선진화 방안에 따라, 불필요한 재인증으로 인한 기업의 부담을 완화하고 행정력의 낭비를 방지하기 위해 철도용품 품질인증의 유효기간을 전제로 하는 재인증 관련 규정을 삭제하였다.

2.4. 철도안전법 시행규칙

철도안전법 시행규칙은 철도안전법 및 철도안전법시행령에서 위임된 사항과 운전업무종사자에 대한 신체검사 및 적성검사의 실시, 철도기술심의위원회의 설치, 종합시험운행의 시행 등에 필요한 사항을 정하기 위해 2005년에 제정하였다.

2008년 12월 철도안전법 시행규칙 일부 개정에서는 질병 발생 현황 및 철도의안전운행과 관련성 등을 고려하여 철도차량운전·관제 등 업무에 종사하는 자에 대한 신체검사의 항목 및 기준을 정비하였다.

- 신체검사의 불합격 기준을 최초검사·특별검사와 정기검사로 구분하여 정하고,
- 신체검사의 항목 중 비·구강·인후 계통과 중복되는 치아계통과 정신지체에 포함되는 지능결함을 삭제하며,
- 발생빈도가 높은 퇴행성질환과 철도안전에 중대한 영향이 있는 수면장애·공황장애를 추가하며,
- 질환이 있는 사실만으로 불합격 처리하던 것을 의사가 증상에 따라 업무수행 가능 여부를 판단하여 불합격 여부를 정하도록 함

2009년 2월 철도안전법 시행규칙 일부 개정에서는 신체검사지정병원 등 철도안전 관련 지정기관이 법령을 반복적으로 위반하는 경우 가중 처분되는 기준 기간을 현행 2 년에서 1 년으로 단축하고, 경미한 위반행위에 대하여는 업무정지 처분 이전에 경고를 하여 자발적으로 시정할 수 있는 기회를 부여하는 등 행정처분 기준을 변경하였다.

2009 년 6월 철도안전법 시행규칙 일부 개정은 철도차량 제작검사기관 지정기준 과 관련하여 「철도안전법 시행령」이 개정됨에 따라, 같은 영에서 위임된 철도차량 제작검

사기관의 지정관련 내용을 변경하였다.

- 「국가표준기본법」제2 3 조에 따른 검사기관으로 인정받아야 시행할 수 있는 철도차량 검사항목 신설
- 철도차량 제작검사기관 지정 신청 시 제출해야하는 서류 등을 정하고, 기술인력 보유기준을 업무분야별로 조정

2009년 12월 철도안전법 시행규칙 일부 개정은 철도차량 제작검사기관 지정과 관련한 「철도안전법 시행령」의 변경에 따라 관련 제출서류에 관한 사항을 변경하였다.

2010년 3월 철도안전법 시행규칙 일부 개정은 철도차량운전면허를 취득하기 위한 집합식 이론교육을 폐지하고 기능교육훈련은 현행 이수시간 내에서 탄력적으로 운용하도록 하여 철도차량운전면허의 취득이 용이하도록 하였다. 그리고 운전업무 수행경력이 없는 면허소지자가 면허갱신 시 받아야하는 교육시간을 단축하고, 실무수습 · 교육계획 및 교육시간 등을 철도운영자가 자율적으로 정하여 실시하도록 하여 필요시 철도차량 운행이 용이하도록 교육훈련 제도를 변경하였다.

- 운전면허시험 응시를 위한 서류 중 적성검사기관에서 교부한 적성검사판정서는 당초 응시원서 접수일 이전 2 년 이내인 것에서 1 0 년 이내인 것으로 변경
- 운전면허시험 응시를 위한 서류 중 교육훈련기관에서 교부한 교육훈련수료증명서는 당초 응시원서 접수일 이전 2 년 이내인 것에서 유효기간 삭제
- 운전면허 갱신에 필요한 경력과 관련하여 당초에는 이론교육 및 기능교육을 각각 20시간 이상 받아야 하나, 통합하여 20시간으로 변경

2010년 9월 철도안전법 시행규칙 일부 개정에서는 철도차량의 제작기술과 소재 및 유지보수기술의 발전 등을 감안하여 철도차량의 사용내구연한 기준 및 사용 내구연한 연장기간을 개선함으로써 차량운영의 효율성을 제고함은 물론 철도차량의 조기교체에 따른 경제적 손실을 방지하도록 하였다.

- 당초 : 철도차량에 대한 정밀진단 결과 당해 철도차량이 안전운행에 지장이 없는 것으로 판정된 때에는 5 년의 범위 내에서 그 사용내구연한의 연장기간을 지정
- 변경 : 철도차량에 대한 정밀진단 결과 안전운행에 적합하다고 인정된 철도차량에 대한 사용내구연한의 연장은 1 5 년의 범위에서 정밀진단기관이 인정하는 기간까지로 지정(사용내구연한이 5 년을 초과하여 연장된 철도차량에 대하여는 5 년마다 철도운영자가 자체 안전진단 실시)

3. 도시 철도법

3.1. 도시철도법 주요 내용 및 규정

도시철도란 도시교통의 원활한 소통을 위하여 도시교통권역에서 건설·운영하는 철도·모노레일·노면전차·선형유도전동기·자기부상열차 등 궤도에 의한 교통시설 및 교통수단으로서, 도시철도법에 의해 건설·운영되는 철도가 해당된다.

도시철도법은 도시철도 사업 전반에 대한 사항을 규정하고 있는 기본법으로서, 도시철도의 안전에 관한 규정 또한 포함하고 있다. 도시철도의 건설· 운영에 관하여는 다른 법률에도 불구하고 도시철도법을 적용하는 것을 원칙으로 한다. 다만 도시철도법 내에 특별한 규정이 없는 경우에는 철도안전법 등의 관계 규정을 준용토록 규정하고 있다. 현행 도시철도법에 규정되어 있는 철도안전 관련 규정은 다음과 같다.

- 도시철도의 건설 및 운전에 관한 사항은 국토해양부령 준수(도시철도 건설규칙, 도시철도운전규칙)
- 도시철도 차량 및 시설 표준규격 제정 및 사용 권고
- 도시철도 차량 및 시설 관련 안전기준에 대한 국토해양부령 준수 (도시철도 차량 안전기준에 관한 규칙, 도시철도 시설안전기준에 관한 규칙)
- 도시철도 차량 및 시설에 관한 성능시험 시행
- 도시철도용품의 품질인증 권장
- 도시철도차량의 내구연한 연장 및 정밀진단에 관한 국토해양부령 준수 (도시철도 차량관리에 관한 규칙)
- 표준규격, 안전기준, 성능시험기준, 품질인증기준 및 정밀진단기준의 제정·개정 및 폐지에 관련된 전문적인 기술 검토 및 개선방안 마련에 관한 업무를 체계적·효율적으로 추진하기 위하여 전담기관을 지정하여 운영

3.2. 도시철도법 하위법령

도시철도법 시행령에 위임된 철도안전 관련 규정은 다음과 같다.

- 철도표준규격 제정 절차
- 안전기준을 준수해야 하는 도시철도차량 및 시설의 구조와 장치
- 성능시험 대상, 절차, 기준, 방법 및 시험기관 지정에 관한 사항
- 품질인증 대상, 절차, 기준, 방법 및 시험기관 지정에 관한 사항
- 정밀진단기관 지정에 관한 사항

도시철도법 시행규칙에 위임된 철도안전 관련 규정은 다음과 같다.

- 표준규격, 안전기준, 성능시험기준, 품질인증기준 및 정밀진단기준 관련 전담기관의 업무범위 및 운영에 관한 사항

도시철도운전규칙에 위임된 철도안전 관련 규정은 다음과 같다.

- 도시철도 안전관련 종사자에 대한 적성검사, 교육훈련 등의 관리
- 열차의 안전한 운행을 위해 도시철도시설 안전점검 등의 조치 시행
- 응급복구에 필요한 기구 및 자재 등에 관한 보관 및 정비
- 안전운전계획 수립 및 시행
- 도시철도 신설구간 등에서의 시험운전 실시
- 선로의 보전, 점검· 정비 시행 및 공사 후 사용 제한
- 전력설비 보전, 전차선 점검, 전력설비 검사 시행 및 공사 후 사용 제한
- 통신설비의 보전, 검사 시행 및 공사 후 사용 제한
- 운전보안장치의 보전, 검사 시행 및 공사 후 사용 제한
- 선로, 전력설비, 통신설비, 운전보안장치 점검 내역 기록보존
- 건축한계 내에 물품유치 금지
- 열차 보전, 차량검사· 시험운전 시행 및 기록 보존
- 열차운전 관련 사항 : 열차 편성, 비상제동거리, 제동장치 및 시험, 운전 시운전면허 소지(운전면허 내용은 철도안전법 적용), 무인운전 시 안전 확보, 운전진로, 폐색구간, 추진운전과 퇴행운전, 동시출발 및 도착 금지, 정거장외 승하차 금지, 선로의 차단, 열차 정지· 서행· 진행 등
- 차량의 결합· 해체에 관한 사항
- 선로전환기 취급에 관한 사항
- 열차운전 속도 및 속도 제한에 관한 사항
- 차량의 구름방지에 관한 사항
- 폐색방식 유형 및 준수사항
- 신호기의 유형 및 준수사항
- 운전관련 표지설치에 관한 사항

도시철도차량관리에 관한 규칙은 도시철도차량의 성능시험 및 정밀진단, 도시철도용품의 품질인증 등에 관하여 다음의 사항을 규정하고 있다.

- 성능시험 면제대상, 시험절차 일부 면제 대상, 성능시험 신청/기관지정 신청 관련 서식
- 품질인증 신청 시 제출서류
- 도시철도 차량내구연한, 정밀진단 절차, 기관지정 신청 관련 서식

도시철도차량안전기준에 관한 규칙은 도시철도차량의 구조 및 장치의 안전운행에 필요한 기준을 규정하고 있다.

- 시설물과 적합한 도시철도차량의 발주 또는 운영에 관한 사항
- 철도차량한계, 축 중, 중량분포, 차량 표지 등에 관한 사항
- 철도차량의 화재안전기준, 전기안전기준 관련 사항
- 철도차량의 충돌안전기준에 관한 사항
- 철도차량의 차체 및 장치에 대한 안전관련 기준
- 철도차량의 주행장치에 대한 안전관련 기준
- 철도차량의 제동 장치에 대한 안전관련 기준
- 철도차량 추진제어 및 보조전원장치에 대한 안전관련 기준
- 철도차량 신호보안 장치에 대한 안전관련 기준
- 철도차량 연결 장치, 통신장치, 집전장치 등 기타 장치에 대한 안전관련 기준
- 도시철도시설안전기준에 관한 규칙은 도시철도차량의 안전운행을 위한 선로시설, 전철전력설비, 신호 및 열차제어설비의 안전기준을 규정하고 있다.
- 도시철도차량과의 상호 연관성을 고려한 선로시설, 전철전력설비, 신호 및 열차제어설비 설치・운영에 관한 사항
- 선로 안전에 관한 일반기준 및 시설・ 설비에 관한 기술기준
- 노반 안전에 관한 일반기준 및 시설에 관한 기술기준
- 철도교량 안전에 관한 일반기준 및 시설에 관한 기술기준
- 터널 안전에 관한 일반기준 및 시설・ 설비에 관한 기술기준
- 전철전력설비의 전기안전기준, 화재안전기준, 설계 및 설치에 관한 사항
- 신호 및 열차 제어설비의 전기안전기준, 화재안전기준, 설계 및 설치에 관한사항
- 도시철도시설 유지관리계획 수립・시행 및 기록관리 등 철도운영자 의무에 관한 사항

4. 철도안전법과 도시철도법의 규정 비교

도시철도의 경우, 철도안전과 관련된 사항 중 자체 법규에 규정되어 있는 사항은 철도시설과 철도차량에 대한 안전기준, 성능시험, 품질인증, 표준화, 정밀진단, 종합시험운행 등이다. 이는 도시철도 차량 및 시설 유형이 다양하고 기술적 특성이 일반철도와 다르기 때문으로 판단된다. 일례로 도시철도 시스템에서는 차량과 시설간의 인터페이스가 중요하기 때문에 철도시설에 대한 성능시험도 법령에 규정되어 있다.

한편, 도시철도차량에 대한 제작검사는 철도안전법의 규정을 준용하고 있다. 이는 제작검사가 당초 인증된 철도차량과의 일치성을 확보할 수 있도록 제작과정 자체에 주안점을 두기 때문에 도시철도만의 기술적 특성을 고려한 별도의 규정이 필요하지 않은 것으로 판단된다.

도시철도의 경우, 기술적 특성이 차별화되지 않은 안전 분야에 대해서는 대부분철도안전법 규정을 적용하고 있다. 일례로 철도 운영자 등이 국토교통부장관으로 부터 승인받아야 하는 안전관리규정, 비상대응계획과 2년 단위로 심사를 받는 종합안전심사 등이 대표적이다. 그리고 철도운전자가 취득해야 하는 면허 관련 사항이나, 철도이용자들이 준수해야 하는 철도보호 및 질서유지 관련 조항들도 철도안전법의 적용을 받고 있다.

표 20 도시철도 안전관련 사항 적용 법령

철도안전법 적용	도시철도법 적용
철도안전종합계획	운전・관제업무 종사자 관리
안전관리 규정	철도시설 안전기준
비상대응 계획	철도차량 안전기준
종합안전심사	철도용품 품질인증
철도차량 운전면허	철도 표준화
철도차량 제작검사	철도차량 성능시험
철도차량 운행안전	철도차량 내구연한 및 정밀진단
철도보호지구 내 행위	종합시험운행
철도보호 질서유지	
철도사고조사 처리	
철도안전 기반구축	

5. 철도 기술기준

5.1. 철도 기술기준 체계

철도안전법 개정 전 철도차량 관련 지침과 시행지침은 아래 [그림 14]와 같이 성능 시험 위주로 되어 있어 해외 선진 기술기준에 부합하지 못하였다. 따라서 도시철도와 일반철도로 분산 된 안전기준, 표준규격 및 각종 시험기준 등을 통합 및 재편하고, 해외 선진 기술기준과 부합되도록 보강하여 [그림 15]의 철도 기술기준 체계를 구축하였다.

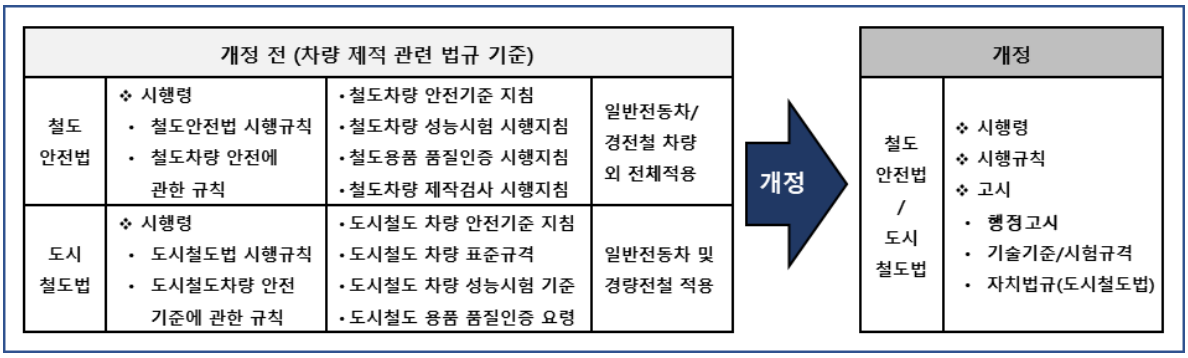


그림 14 철도 관련 법규 체계 개정 전·후

철도 기술기준 체계는 철도차량, 철도용품, 철도시설, 철도안전관리체계로 구성되어 있다. 철도차량은 고속철도, 일반철도, 도시철도, 특수철도 차량 등 차종별 기술기준과 제작자 승인기준으로 구성되며, 철도용품은 차량분야, 궤도분야, 전철전력분야, 신호통신분야의 철도용품 기술기준과 철도용품 제작관리 및 품질유지의 제작자 승인 기준으로 구분되어 있다.

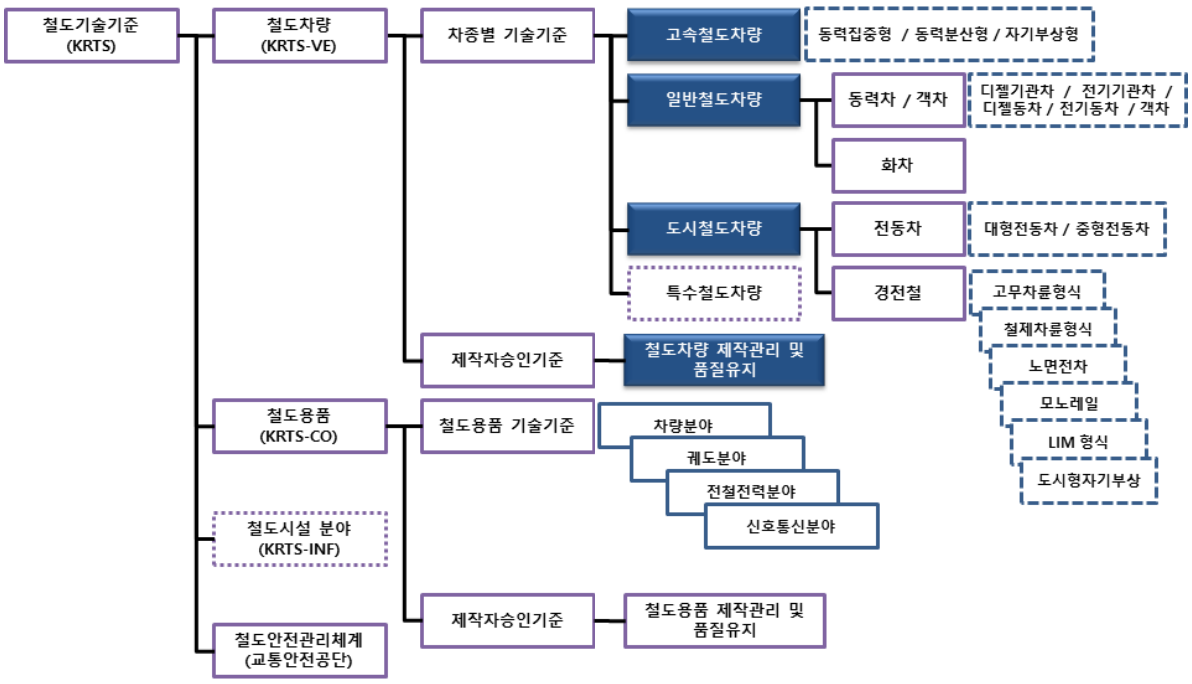


그림 15 철도 기술기준 체계도

5.2. 철도차량 기술기준

철도안전법 체계 개정 전 차량의 검사는 최초 차량 성능검사가 제작검사(차량제작) 이후 시행되므로 제작 전 사전 성능검증 취지가 퇴색되었으며, 성능시험은 설계적합성의 검증보다는 부품 및 완성차량의 시험 합격여부에 치중되어 있었다. 또한 제작자가 아닌 제3자(정부/검사기관)가 결함을 찾아 시정하는 것에서 한계를 보였고, 사후관리 관련 규정이 전무하여 제작사에 시정요구, 사용정지 등의 안전 확보를 위한 수단이 미비하였다.

이에 철도안전법 체계를 개정하면서 기존 체계의 문제점을 개선하고자 차량, 용품, 시설, 안전관리체계에 대한 안전 승인을 위한 기준으로 각각의 기술기준을 제정하였다. 철도차량 기술기준은 철도안전법 제26조(철도차량 형식승인), 제26조의 3(철도차량 제작자 승인) 및 제26조 6(철도차량 완성검사)에 근거하여 형식승인 및 제작자승인에 필요한 기술기준을 위해 제정되었다.

철도차량 기술기준의 주요 내용은 개요, 필수요구사항, 주요장치별 기준, 시험규격서로 구성되어 있다. 필수요구사항은 안전, 성능, 인터페이스, 운영 및 유지관리, 운용한계 등으로 구성되며, 주요장치별 기준은 철도차량을 구성하는 주요장치별 설계 및 구조에 관한 요구사항이 규정된다. 시험규격은 설계 적합성을 시험으로 입증 할 경우의 표준화 된 시험규격으로 부품, 구성품, 완성차, 예비주행, 시운전단계로 구분하고 있다.

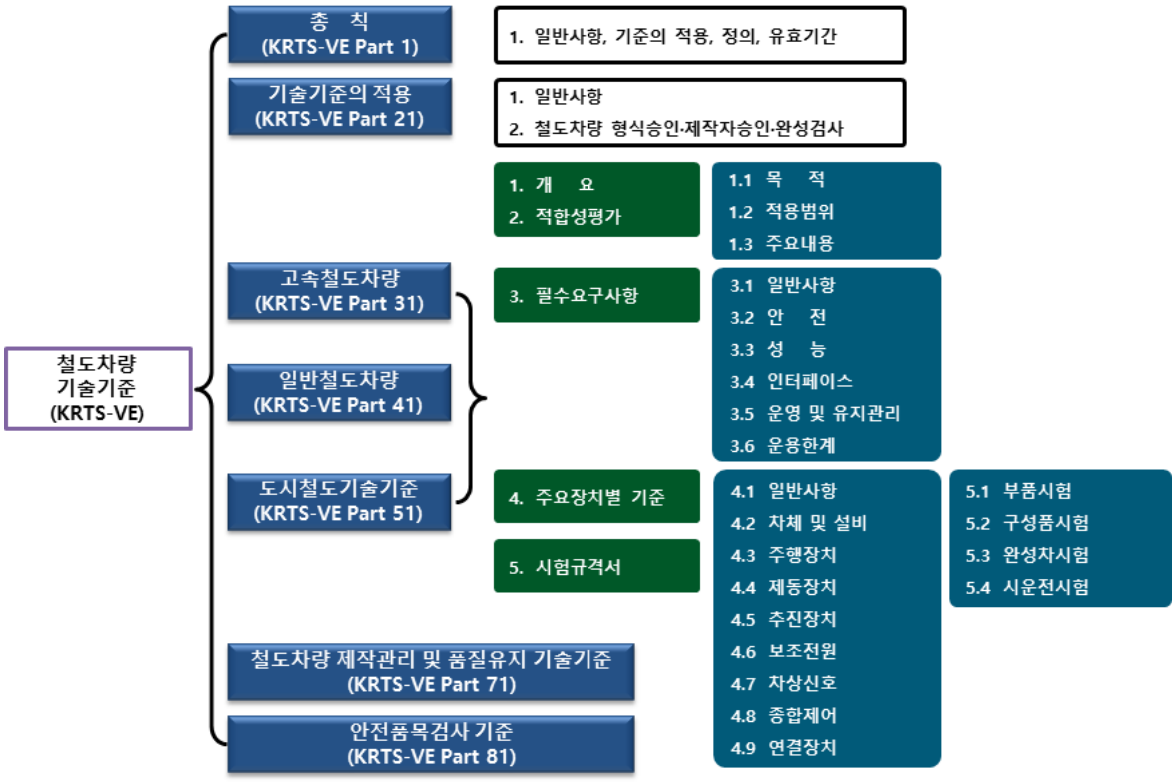


그림 16 철도차량 기술기준 구성도

5.2.1. 필수 요구사항

철도차량 기술기준의 내용 중 필수요구사항은 일반사항, 안전, 성능, 인터페이스 등으로 구성되어 있으며 이 중 안전 부분에는 차량한계, 주행안전, 충돌 및 전복, 화재안전, 전기안전, 위험도분석, 철도소프트웨어에 대한 기준으로 구성되어 있다.

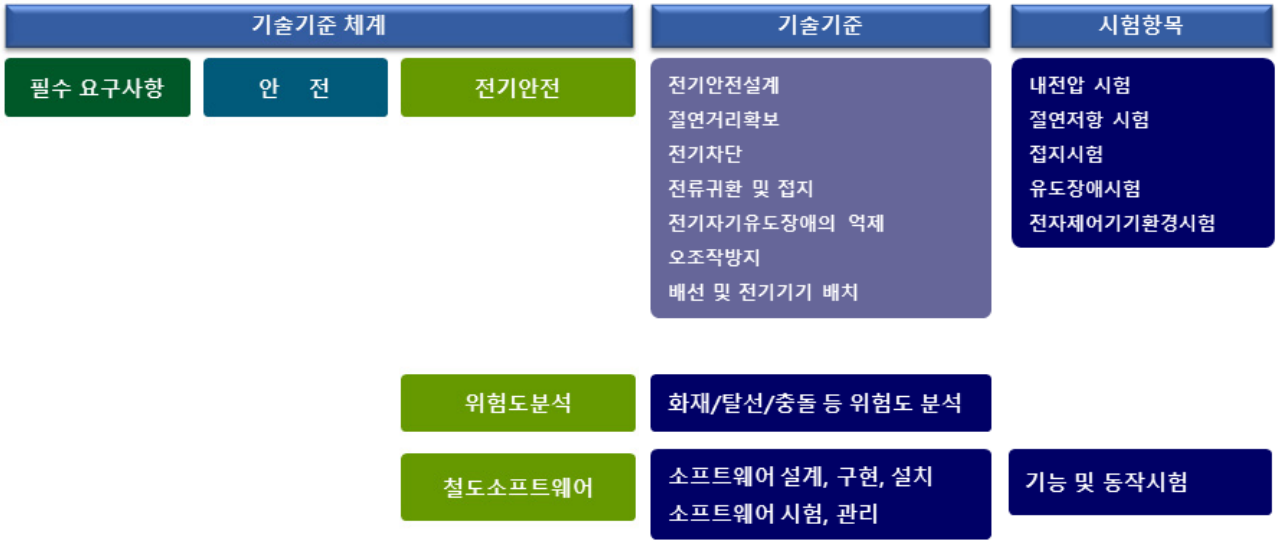


그림 17 철도차량 기술기준 필수 요구사항

5.2.2. 위험도분석

철도 차량의 설계·제작·유지보수 및 운영환경 전반에 걸친 위험도분석을 수행하여 필요한 안전대책을 제시하고, 해당 철도차량의 위험도가 설계단계부터 적절한 수준으로 제어되고 있는지를 입증해야 한다.

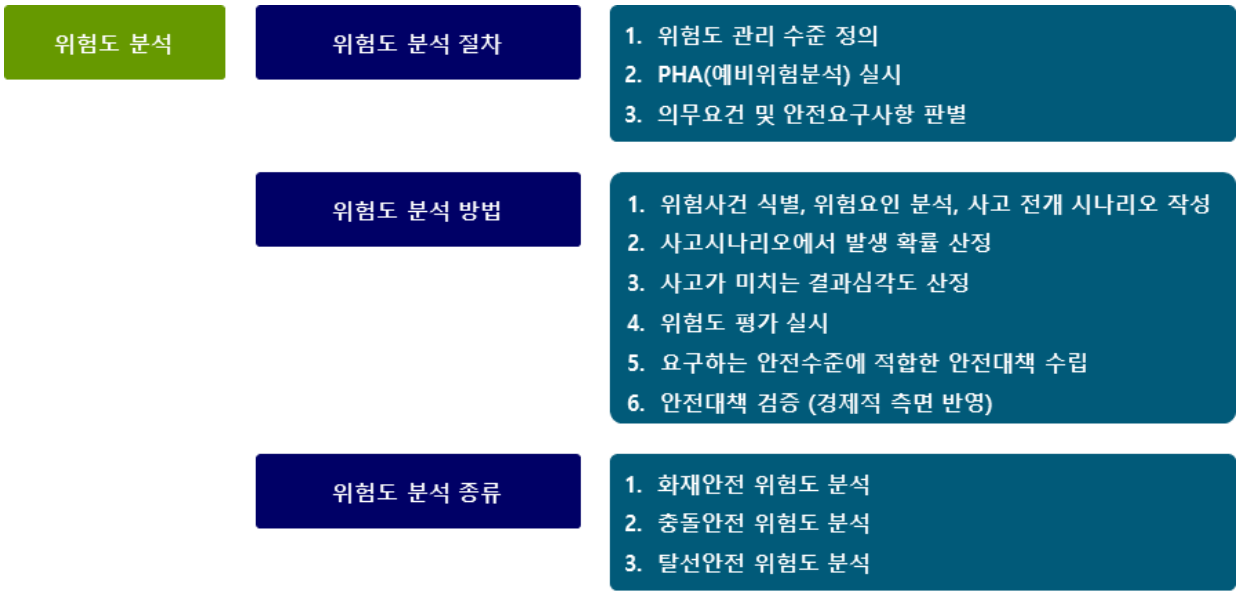


그림 18 철도차량 기술기준 위험도분석

5.2.2.1. 적용범위

철도차량의 위험도분석은 신규로 제작・조립・수입되는 기관사, 승무원 또는 승객이 탑승하는 동력차・부수차(제어차 포함)에 대하여 실시한다. 다만, 위험도분석을 시행한 철도차량과 동일한 운영조건에서 동일한 구조 시스템 및 등가 재질을 갖는 철도차량에 대해서는 위험도분석을 실시하지 않아도 된다.

5.2.2.2. 분석절차

철도운영자는 도입하려는 철도차량에 대한 위험도 관리수준을 정하고 예비위험분석(PHA)을 실시하여 기술기준에서 정한 의무요건의 확인과 승객, 공중, 직원의 안전 및 운행안전을 보장하는 요구사항을 판별하여야 한다. 이 후 예비위험분석 결과를 반영하여 철도차량을 설계하고, 제작・시험평가・운영 및 유지관리 전반에 걸친 위험도분석 보고서를 작성하여 검사기관 또는 전문기관에 제출하여야 한다.

5.2.2.3. 분석방법

철도차량의 위험도분석은 다음의 절차에 따라 실시하며, 타당한 사유와 합리적인 근거가 있는 경우는 해당 분석방법을 수행하지 않아도 된다.

(1) 사고를 유발할 수 있는 위험사건을 식별하고 위험요인을 분석하여, 분석 된 위험요인이 사고로 전개 될 수 있는 사고 시나리오를 작성한다.

(2) 각각의 사고시나리오에서 위험사건의 발생확률을 산정한다.

(3) 각각의 사고시나리오에서 사고가 피해에 미치는 영향을 분석하여 결과심각도를 산정한다.

(4) 각각의 사고시나리오에서 위험사건의 발생확률과 결과심각도를 산출하여 위험도 평가를 실시한다.

(5) 위험도평가 결과가 요구하는 안전수준을 만족하지 못하는 경우에는 그에 대한 원인을 분석하고 요구하는 안전수준에 적합하도록 안전대책을 수립한다.

(6) 수립된 안전대책이 요구하는 안전수준에 적합하며, 경제적 측면에서 적정한 것임을 확인하여 안전대책의 검증을 실시한다.

5.2.2.4. 고려사항

철도차량의 위험도 분석을 실시하는 경우에는 다음의 내용을 고려하여야 하며 산출물 작성 시에도 해당 고려사항을 반영해서 작성해야 한다.

(1) 위험도 분석을 위한 기본 자료는 타당성을 입증할 수 있도록 충분히 조사·기술하여야 한다.

(2) 위험도 분석은 가능한 정량적인 방법으로 실시하며, 정량적인 방법이 곤란한 경우에는 기존의 경험 또는 사례를 이용하거나 정성적인 방법을 적용한다.

(3) 자료의 조사 및 위험도평가는 가능한 가장 최근에 확립된 방법 및 기술을 사용하여 실시하며, 적용된 방법 및 기술을 명시하고 인용된 자료 또는 가정은 그 출처를 분명히 하여야 한다.

(4) 위험요인을 분석하고 사고시나리오를 작성하는 경우 철도차량에서의 위험요인을 승객탑승구역, 승객접근구역, 접근제한구역 등으로 분류하여 승객, 직원 및 공중에 대한 인명피해와 중대한 재산상의 손실 우려가 있는 위험요인 및 사고시나리오를 작성하여 검토해야 한다.

(5) 철도차량의 위험도 분석은 철도차량 또는 열차의 운행조건과 선로변 또는 승강장의 안전시설 등과의 상호관련성, 비상사태가 발생하는 때에 승객 및 승무원의 구조를 위한 철도운영기관, 소방기관, 의료기관 등의 제반 활동사항을 포함하여야 한다.

(6) 위험도분석의 세부기준에 관하여 기술기준에서 정해지지 않은 사항은 관련된 국가규격 및 국제규격을 중용한다.

5.2.2.5. 분석대상

○ 화재안전 위험도분석

철도차량의 화재안전을 위한 위험도분석은 철도차량의 위험요인(Hazard)으로부터 발생 가능한 사고시나리오의 발생확률과 결과심각도에 의하여 수행하며, 결과심각도는 통계적 기법을 활용한 사고사례 영향분석 또는 전산화재 대피모사 또는 실물(모형) 화재시험 등으로 수행한다. 세부적인 분석은 [그림 19]와 같다. (전체 사항은 철도차량 기술기준 3.2.6.3 위험도분석 방법 참조)

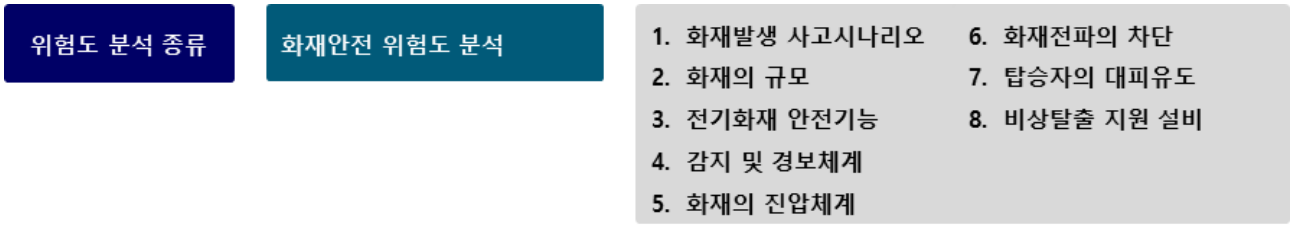


그림 19 화재안전 위험도 분석 시 고려사항

○ 충돌안전 위험도분석

운행제어장치, 제동장치 등 열차나 철도차량의 기술적 결함, 신호지시위반이나 과속운행과 같은 운전취급 오류 등에 의한 열차간의 충돌사고, 선로지장물이나 외부장애물과 같은 장애물 충돌사고의 위험성을 고려하여 분석을 수행한다.

소프트웨어와 관련 한 사항은 [그림 20]과 같으며, 다음의 탈선안전 위험도분석에서도 동일하게 적용하여 분석을 수행한다. (이외 고려 사항은 철도차량 기술기준 3.2.6.3 위험도분석 방법 참조)

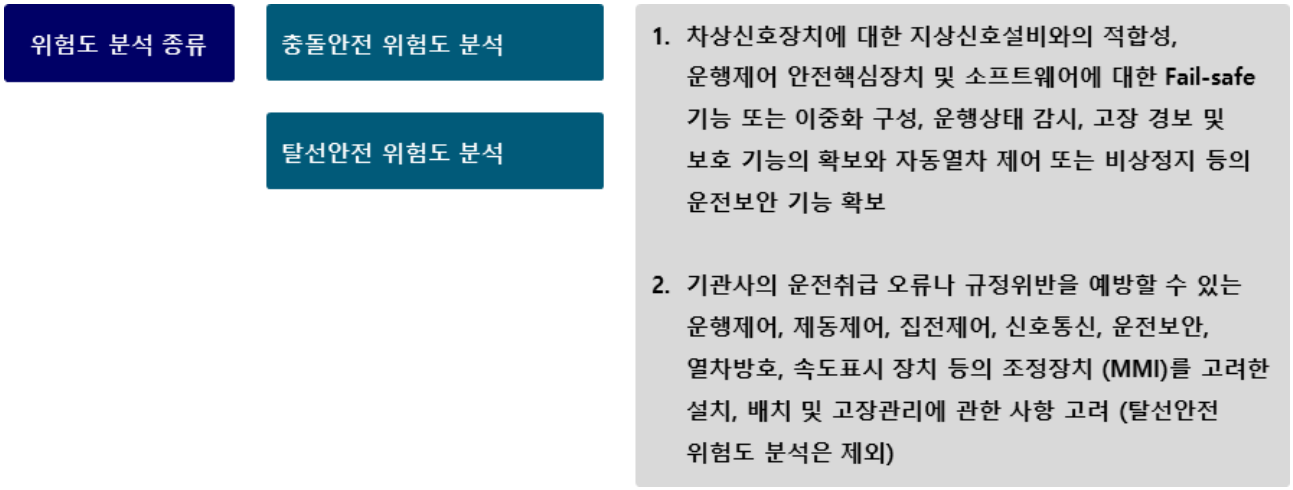


그림 20 충돌안전, 탈선안전 위험도 분석 시 고려사항

○ 탈선안전 위험도분석

주행장치, 제동장치 등 열차나 철도차량의 기술적 결함, 신호지시위반이나 과속운행과 같은 운전취급 오류 등에 의한 열차탈선사고, 선로지장물이나 외부장애물과 같은 장애물에 의한 탈선사고의 위험성을 고려한다.

차량한계의 확보, 중량분포 및 축중에 대한 기준 만족, 최소곡선구간 통과 등 열차의 정지 및 운행상황에서의 탈선 위험방지설계에 관한 사항을 고려한다.

규정된 열차 운행조건에서 차량과 선로 간 작용력을 최소화하여 주행안전성을 확보하고 곡선구간에서의 전복을 방지할 수 있는 구조설계와 장치의 안전성능 확보에 관한 사항을 고려한다.

소프트웨어와 관련 한 사항은 충돌안전 위험도 분석과 같이 [그림 20]의 내용을 따른다. (이외 고려 사항은 철도차량 기술기준 3.2.6.3 위험도분석 방법 참조)

5.2.2.6. 결과기록

위험도분석 결과를 아래와 같이 차량 설계 및 제작에 필요한 항목에 대해 기술하되 관련 항목을 조정하여 기록할 수 있다. 또한 위험도분석 결과보고서는 체계적으로 관리하여야 한다.

(1) 철도운영개요, 기반시설 내용 및 주요특성, 열차운행계획 등 도입되는 열차나 차량운행에 관한 전반적인 개요

(2) 철도운영과 기반시설에 의한 제약사항, 차량 및 열차의 안전요구 특성, 열차운행 및 유지관리 조건 등 철도차량 또는 열차의 전반적인 운행안전에 영향을 미칠 수 있는 위험요인을 확인하고 안전성을 검증할 수 있도록 설계에 관한 요구사항을 상세하게 기술

(3) 실시된 위험도분석 결과를 상세하게 기술하고, 이를 기반으로 철도차량의 안전운행을 위한 안전대책 실행계획 수립에 반영

(4) 위험도분석에 활용한 참고자료 및 인용문헌 등을 기술

5.2.3. 철도소프트웨어

철도소프트웨어는 소프트웨어 개발, 확인, 및 검증, 안전성분석으로 구별되며, 각 활동은 생명 주기별로 계획, 요구사항 정의, 설계, 구현, 시험, 배포 및 설치, 유지보수의 각 단계마다 수행한다. (자세한 내용은 철도차량 기술기준 3.2.7 철도소프트웨어 참조)

5.2.3.1. 소프트웨어 안전

철도 차량에 사용되는 응용소프트웨어, 운영소프트웨어, 펌웨어 등의 소프트웨어에 대하여 소프트웨어 개발 및 사용 전에 품질보증계획을 수립하고 이를 문서화 한다. 또한 정기적으로 감사를 실시하여 품질보증계획에 따라 업무가 수행되었는지 여부를 확인하고 계획의 유효성을 평가한다.

소프트웨어가 의도 된 기능을 수행할 수 있도록 소프트웨어 개발 요건, 확인 및 검증 요건, 안전요건 등의 품질보증 요건에 맞추어 소프트웨어에 대한 계획, 설계, 구현, 시험, 설치, 운영 및 유지보수를 수행하며, 개발 및 사용 전에 소프트웨어 형상관리 계획을 수립하여 이를 문서로 작성한다.

소프트웨어 생명주기 동안에 형상관리 계획에 따라 형상항목 식별업무, 형상항목 통제 업무, 형상상태 기록 및 보고 업무의 형상관리 활동을 수행하며, 형상관리 계획서에 따라 정기적으로 소프트웨어의 형상감사를 실시하여 소프트웨어 형상관리계획서에 업무가 수행되었는지 여부를 확인하고 계획의 유효성을 평가하여야 한다.

소프트웨어 안전 확보를 위해 철도소프트웨어에 대한 국제표준 IEC 62279를 기반으로 안전활동을 수행한다. 소프트웨어 전부 또는 일부가 신청자와 독립된 전문기관으로부터 국제표준 IEC 62279 기준으로 독립평가를 받은 경우 신청자는 독립평가 대상소프트웨어에 대하여 일부 항목에 대한 검사면제를 요청할 수 있으며, 검사기관은 요청에 따라 대상, 범위, 평가항목을 검토하여 대응 항목의 검사를 면제할 수 있다. 단, 검사면제 요청 시 제출되는 독립평가보고서의 발행기관은 ISO 17020 및 ISO 17065에 따른 검사 업무 자격기관으로 제한한다.

소프트웨어 안전활동은 제출된 서류를 바탕으로 검사하며, 필요 시 검사기관은 신청자와 협의를 거쳐 현장방문을 통한 검사를 시행할 수 있다.

5.2.3.2. 적용범위

철도 차량에 사용되는 응용소프트웨어, 운영소프트웨어, 펌웨어 등의 소프트웨어를 대상으로 하며 이들 중 형식승인검사 또는 완성검사 대상을 제안하고 이에 대한 산출물을 [표 21]과 같이 작성・유지한다. 모든 생명주기 활동에서는 위에서 다룬 소프트웨어 안전에 관한 활동을 공통적으로 수행한다.

표 21 철도차량 기술기준 소프트웨어 구성 체계

생명주기	소프트웨어 개발활동	확인 및 검증활동	안전성분석 활동
	소프트웨어 품질보증 및 형상관리		
계 획	계획수립	계획 검증 및 확인	소프트웨어 안전계획 수립
요구사항	요구사항 정의	요구사항 검증 및 확인	요구사항의 안전성 분석
설 계	구조 및 상세설계	구조 및 상세설계 검증 및 확인	설계 안전성 분석
구 현	소스코드 구현 및 통합	소스코드 검증 및 확인	소스코드의 안전성 분석
시 험	소프트웨어 시험	소프트웨어 시험의 확인 및 검증	소프트웨어 시험 안전성 분석
	- 단위시험		
	- 통합시험		
	- 시스템 시험		
설 치	소프트웨어 배포・설치・인계	사용자 요구사항 만족 여부 확인 (Validation)	소프트웨어 설치 안전성 분석
유지보수	소프트웨어 유지보수	소프트웨어 변경 관리	소프트웨어 변경의 안전성 분석

5.2.3.3. 계획수립

(1) 위험도분석에 따른 철도차량에 대한 예비위험도 분석을 통해 위험도 관리수준을 결정하고, 정해진 관리수준에 따라 신청자는 철도차량 소프트웨어 안전관리를 실시한다.

(2) IEC 62279에서 권고한 조직 구성 및 조직간 독립성을 확보한다.

(3) 소프트웨어 품질보증 계획, 소프트웨어 형상 관리 계획 및 소프트웨어 검증 및 확인 계획을 수립하고, 이를 문서로 작성한다.

(4) 수립된 계획의 현황과 적합성을 정기적으로 점검하고, 점검결과에 따라 필요하다면 소프트웨어 계획을 수정 및 보완한다.

(5) 소프트웨어 개발계획에 따라 다음의 사항을 포함한 확인 및 검증계획을 수립하고 이를 문서로 작성한다.

- 확인 및 검증 업무에 참여하는 주요 조직, 기능 및 책임사항
- 확인 및 검증 업무의 주요일정 및 필요한 자원
- 생명주기의 각 단계에서 수행되어야 할 확인 및 검증의 업무, 절차 및 기법
- 확인 및 검증 업무의 보고

(6) 확인 및 검증계획의 현황과 적합성을 정기적으로 점검하고, 점검결과에 따라 필요하면, 확인 및 검증계획을 수정 및 보완한다.

(7) 소프트웨어 개발 전에 다음의 사항을 포함한 소프트웨어 안전계획을 수립하고, 이를 문서로 작성한다.

- 안전성 분석에 참여하는 주요조직, 기능 및 책임사항
- 안전성 분석의 업무에 필요한 교육 및 훈련에 관한 사항
- 생명주기의 각 단계에서 수행되어야 할 안전성 분석 업무 및 업무의 관리내용

(8) 소프트웨어 안전계획의 현황과 적합성을 정기적으로 점검하고, 점검결과에 따라 필요하다면, 소프트웨어 안전계획을 수정 및 보완한다.

5.2.3.4. 요구사항 정의

소프트웨어에 대한 다음의 요구사항을 정의하고, 그 결과를 문서로 작성한다.

- ○ 소프트웨어 기능
- ○ 소프트웨어의 성능
- ○ 소프트웨어의 외부 연계
- ○ 소프트웨어의 신뢰성관련 요구사항
- ○ 소프트웨어의 안전성관련 요구사항
- ○ 소프트웨어의 보안성관련 요구사항

소프트웨어 요구사항에 대하여 다음의 소프트웨어 확인업무를 수행하고 그 결과를 문서로 작성한다.

- 발주자의 요구사항과 소프트웨어 요구사항과의 추적성 분석
- 생명주기 단계별 요구사항의 적합성 확인
- 하드웨어, 사업자 및 기타 시스템과의 연계요구사항에 대한 적합성 확인

5.2.3.5. 소프트웨어 설계

정의된 요구사항을 바탕으로 소프트웨어 구조 및 상세 설계를 수행하며, 그 결과를 문서로 작성한다. 또한 소프트웨어의 구조설계 및 상세설계에 대하여 다음의 확인업무를 수행하고 그 결과를 문서로 작성한다.

- 소프트웨어 요구사항이 소프트웨어 설계에 정확하게 반영되었음을 확인하는 추적성 분석
- 설계요소 생명주기 단계별 요구사항의 적합성 확인
- 하드웨어, 사업자 및 기타 시스템과의 소프트웨어 연계설계의 적합성 확인

5.2.3.6. 소프트웨어 구현

소프트웨어 상세설계 내용을 소프트웨어 소스코드로 구현하며, 수립된 통합계획에 따라 소프트웨어 통합을 수행한다. 소프트웨어의 소스코드에 대하여 다음의 확인업무를 수행하고 그 결과를 문서로 작성한다.

- 소프트웨어 구조 및 상세설계가 소스코드로 정확하게 구현되었음을 확인하는 추적성 분석
- 소스코드 구성요소에 대한 생명주기 단계별 요구사항의 적합성 확인
- 소스코드와 하드웨어, 사업자 및 기타 시스템과의 연계 적합성 확인

5.2.3.7. 소프트웨어 시험

단위시험, 통합시험, 시스템시험에 대한 계획 및 절차를 수립하고, 이를 문서로 작성하며, 시험계획 및 절차에 따라 시험 한 수행결과 또한 문서로 작성한다. 그리고 시험에 대한 시험계획서, 절차서 및 보고서 내의 상호관계에 대한 추적성을 분석하며, 그 결과를 문서로 작성한다.

5.2.3.8. 소프트웨어 설치

완성차시험 및 시운전시험을 통하여 소프트웨어가 시스템에 정확히 설치되었고 요구되는 기능을 정확히 수행하는지를 확인하고 관리하며 소프트웨어를 시스템에 설치할 경우에 설치관련 안전요구사항의 준수여부를 확인하며, 그 결과를 문서로 작성한다.

5.2.3.9. 소프트웨어 유지보수

유지보수 계획에 따라 다음의 유지보수 활동을 수행한다.

- 소프트웨어 변경요구의 확인
- 문제점 보고
- 소프트웨어 생명주기 활동의 재수행

소프트웨어 운영 중에 요구되는 소프트웨어의 변경에 대하여 다음의 업무를 수행한다.

- 소프트웨어 운영 시 부적합사항으로 인한 영향을 평가
- 소프트웨어의 변경사항에 대한 확인 및 검증업무의 반복 정도를 판단
- 승인된 변경사항에 적합하도록 확인 및 검증계획을 개정
- 소프트웨어 생명주기에 따라 확인 및 검증업무를 재수행

5.2.3.10. 적용대상

철도차량 기술기준은 완성차에 대한 기술기준으로 차량을 구성하는 각 부품들에 대한 기술기준을 포함하고 있다. 이 중 소프트웨어가 사용되는 부품은 다음과 같으며, 해당 부품들에 대한 형식승인 시 철도소프트웨어에 대한 기술기준을 적용한다.

- 차량-신호 인터페이스: 지상신호장치의 인터페이스, 차상신호장치의 인터페이스
- 제동장치: 비상제동, 상용제동, 주차제동, 구원운전 시 제동, 압축공기 공급장치 등
- 보조전원장치: 보호기능 등
- 차상신호장치: ATS, ATP/ATP, CBTC, ETCS or ATC 등
- 종합제어장치: 운행상태 확인 장치, 출입문제어 장치 등

6. 철도차량 형식승인

개정 전 철도차량 분야에서의 안전관리는 (기술검토 → 부품시험 → 구성품시험 → 완성차 시험 → 시운전)의 절차를 가지는 성능시험(최초)과 (기술검토 → 입고검사 → 공정검사 → 최종검사)의 절차를 가지는 제작검사(양산)의 두 가지로 수행되었다. 하지만 이는 설계에 대한 검증이 부족하고 검사기관의 차량 결함 발견 능력에 한계가 보이게 되었고, 사후 결함이 발견되어도 제작자에 대한 제재나 시정이 곤란하였다. 이에 차량 제작과정 위주 검증을 설계단계부터 제작과정 품질관리시스템, 완성품 사후관리까지 확대하여 형식승인(설계단계), 제작자 승인(제작단계)과 완성검사(양산단계)의 세 가지로 개정되었다.

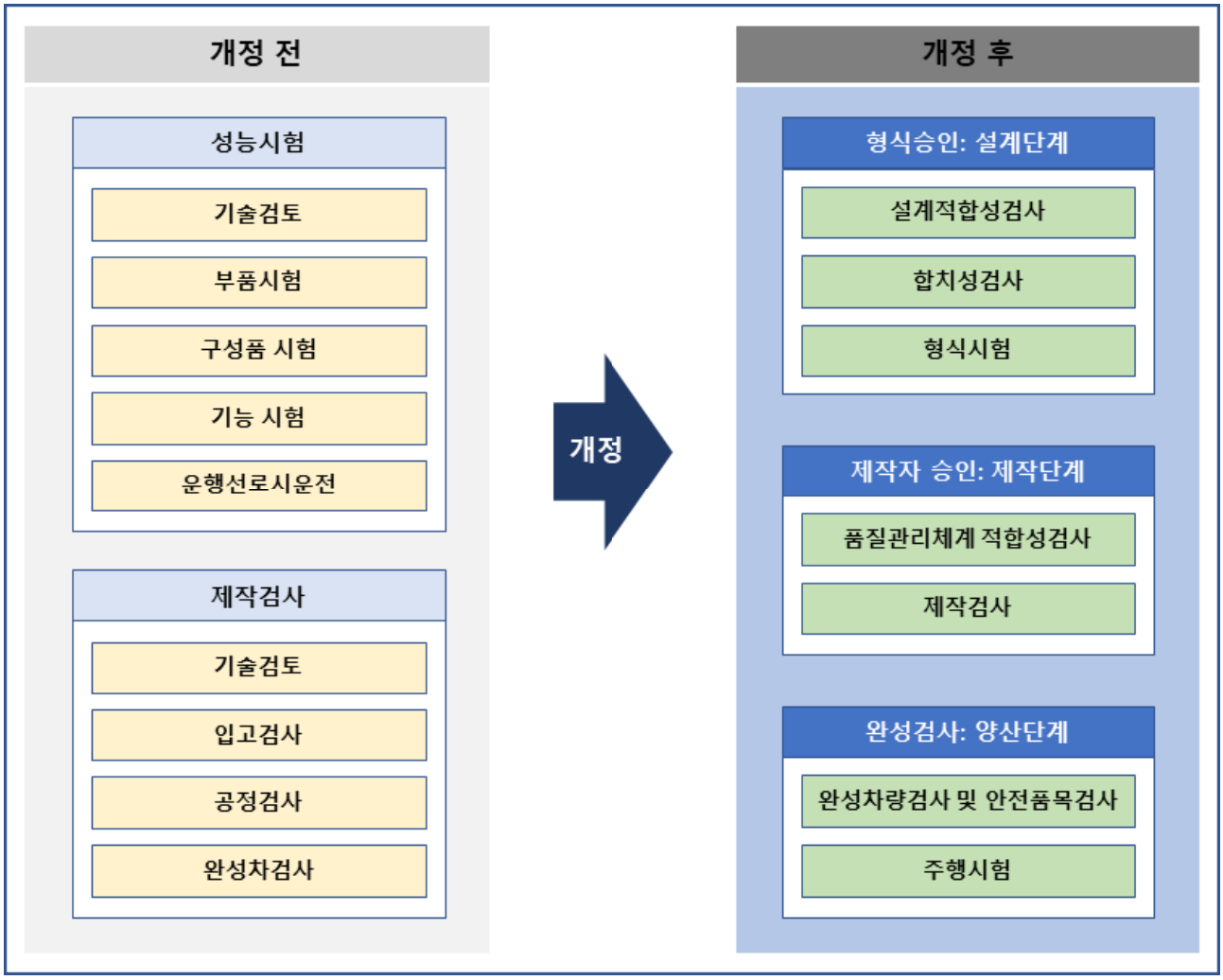


그림 21 철도차량 형식승인 개정 전·후

양산단계 이 후 안전 및 품질확인, 점검을 통해 지속적인 사후관리를 진행하며 품질에 대한 문제와 위반이 발생하는 경우 제재를 할 수 있다.

6.1. 기술분류

형식승인은 철도차량 기술기준에서 제시하고 있는 차량시스템, 차량주요장치, 공통적용기술(전자제어장치, 시운전), 제작자 승인에 대해 [표 22][8]와 같이 진행한다.

표 22 형식승인 기술분류

분 류		주요 분야
차 량 시 스 템	차량/안전	차량한계, 주행안전, 충돌안전, 화재안전, 전기안전, 위험도분석, 소프트웨어
	차량/성능	운행조건, 운행성능
	차량/인터페이스	차량-전력, 차량-신호, 차량-통신, 차량-궤도, 차량-기관사
	차량/운영 유지관리	유지보수기준, 유지보수성, 유지보수자료
	차량/운용한계	안전운행, 신뢰성 및 가용성, 보건 소음, 구원운전, 공기역학적 특성
차 량 주요장치	차량/차제・설비	구조체설계, 구조체안전, 실내기압변화, 리프팅, 장애물제거기, 부식억제, 출입문, 승무원출입문, 출입문-스크린도어, 차량간 통로문, 냉난방 환기장치, 등구류, 의자 및 선반, 전면 유리창, 측면 유리창 및 기타, 운전실 및 비상탈출구, 승객용 비상출구, 경적, 열차비상용품, 고압가스운송차량특수장치
	차량/주행장치	주행장치설계, 주행장치틀, 윤축및차륜특성, 축상조립장치, 현가장치, 차체지지장치, 구동장치
	차량/제동장치	제동장치설계, 제동요구사항, 비상제동, 상용제동, 주차제동, 기초제동, 압축공기공급장치, 활주방지, 구원운전시제동, 제동상태표시
	차량/추진장치	설계요구사항, 인버터/컨버터, 견인전동기, 내연기관구조, 내연기관장치, 보호기능, 집전장치, 비상운전, 피뢰기, 주퓨즈, 차단기, 필터리액터
	차량/보조전원장치	보조전원장치설계, 보호기능, 연장급전, 유도장애의억제, 보조전원용인버터, 축전지
	차량/신호	시스템일반, 자동열차정지장치, 자동열차방호장치, 자동열차제어장치, 자동열차운전장치
	차량/종합제어	종합제어장치설계, 운전상태확인장치, 열차운행기능, 출입문제어, 무인운전
	차량/연결장치	연결기, 통로연결장치
공통적용기술 (전자제어장치 시운전)	전자파환경	
	소음	
	진동/충격	
	온도환경	
	공력특성	
제작자승인	품질관리체계	차량 및 차량용품, 궤도용품, 신호통신용품, 전력용품, 전차선용품

8) http://krts.krri.re.kr 철도형식승인 홈페이지

6.2. 철도차량 형식승인

6.2.1. 진행절차

철도차량 형식승인은 [그림 22]와 같이 제작자가 철도차량 형식승인기관(이하 “검사기관”)에 입증계획서를 제출하면 검사기관에서 사전검토를 진행한 후 계획서에 맞게 인증조직을 구성한다. 조직구성이 완료 되면 제작자에서 제공하는 입증자료를 기반으로 다음과 같이 3단계의 인증 및 승인을 진행한다.

- 형식승인(설계적합성, 합치성, 형식시험): 설계자료 검증
- 제작자승인: 제작 차량 검증
- 완성검사: 양산 차량 검증

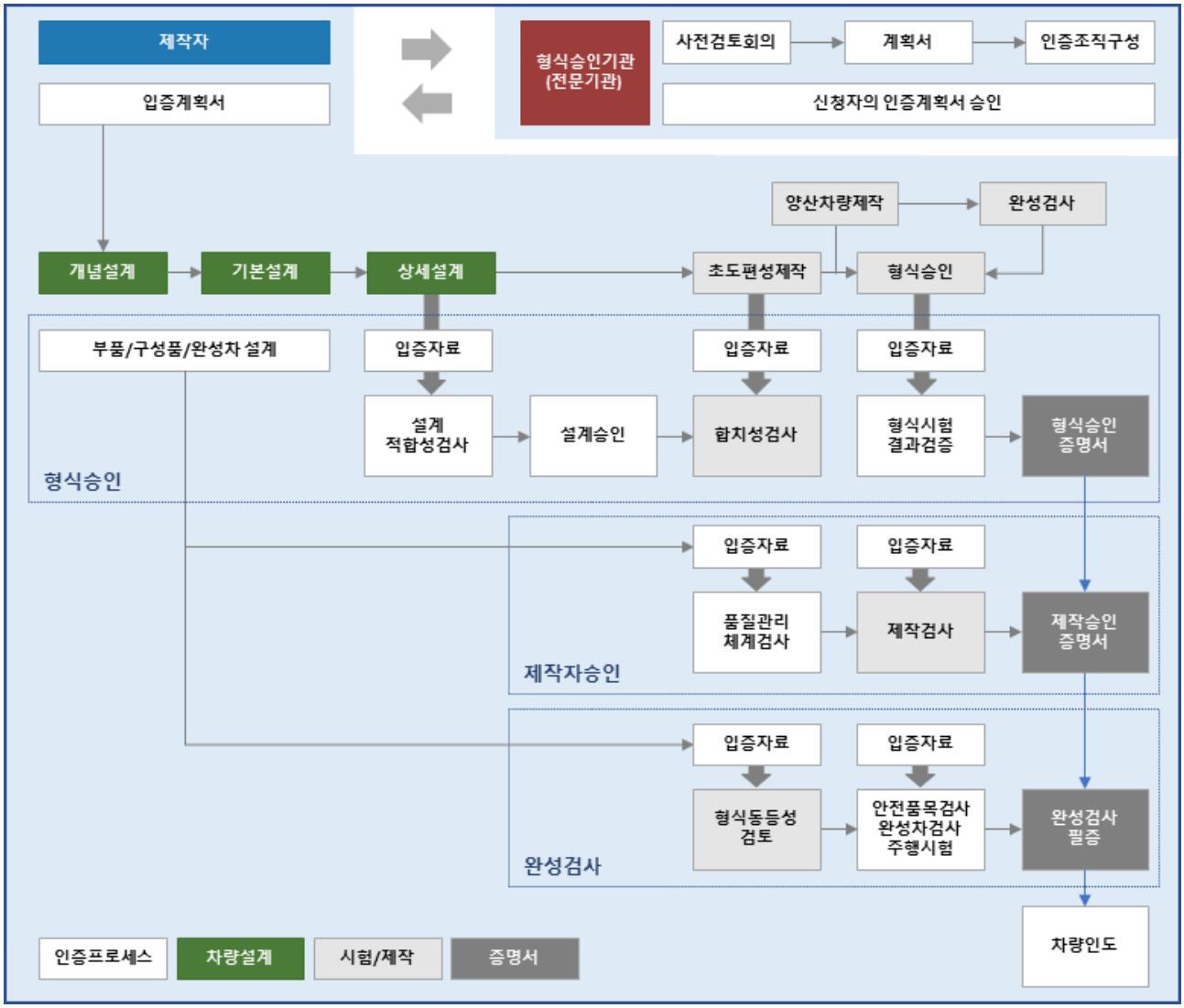


그림 22 철도차량 형식승인 단계별 절차

철도차량 형식승인은 현재 철도기술연구원에서 진행하고 있으며 [그림 23]과 같이 조직을 구성하여 인증 및 승인을 진행한다. 검사원의 경우 전문가 Pool을 활용하여 각 단위의 전문가를 활용하여 전문성을 강화하여 진행한다.

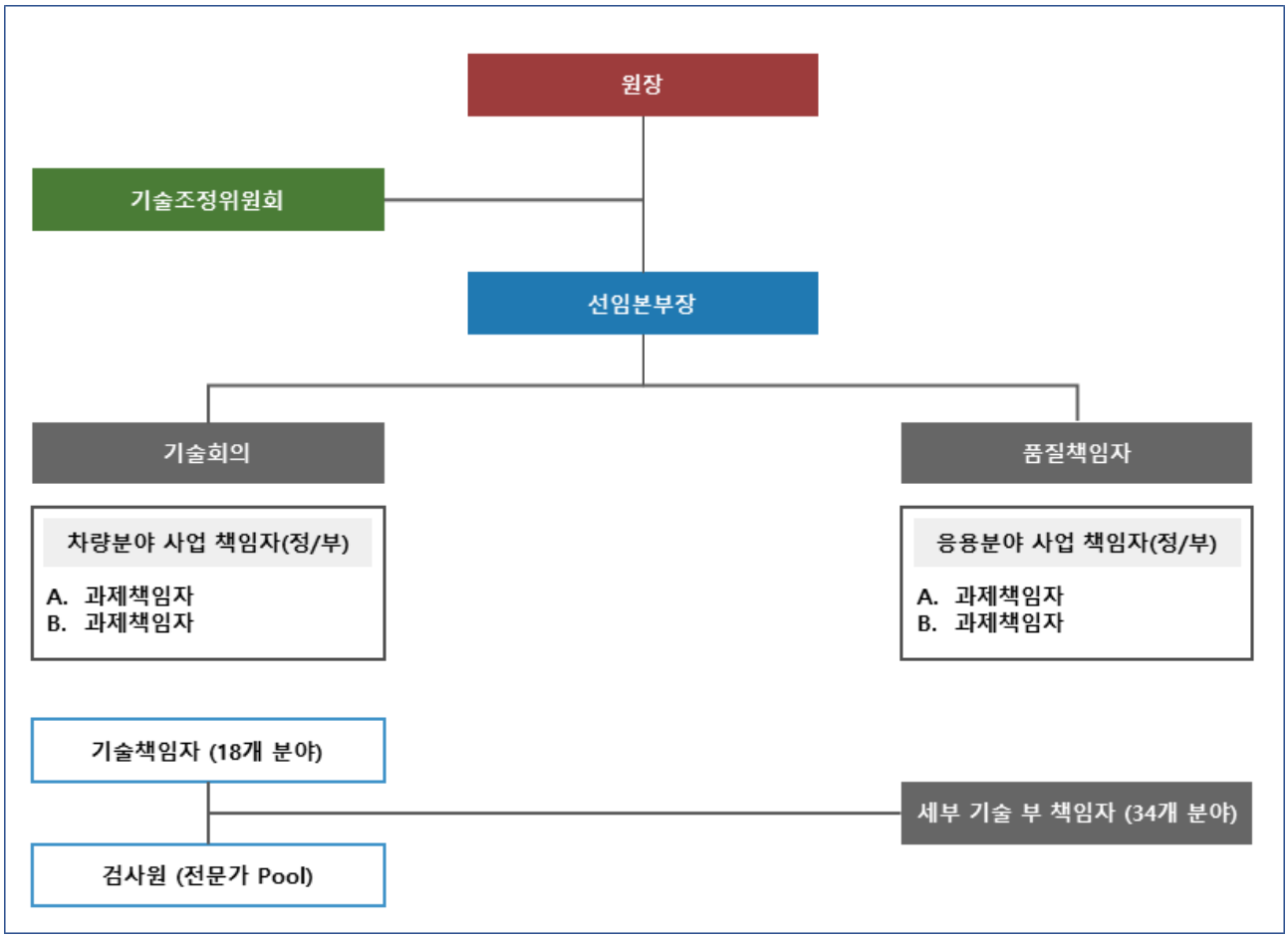


그림 23 철도차량 형식승인 조직구성 (철도기술연구원)

6.2.2. 형식승인

최초로 제작된 철도차량의 형식(Type)이 국가가 규정한 철도기술기준에 적합한지 여부를 확인하기 위하여 아래의 검사를 국토교통부장관이 검사하고 승인하는 것으로, 사유가 있는 경우에는 형식 승인을 면제, 취소, 변경승인 할 수 있다.

(1) 설계적합성검사

(2) 합치성검사

(3) 형식시험

형식승인은 제작되는 철도차량의 설계가 기술기준에 적합한지를 검사 이에 대한 결과를 인증하는 것으로 본 가이드와 관련 된 단계별 위험도분석과 철도소프트웨어에 대한 검토 산출물 자료는 [표 23]과 같다.

표 23 철도차량 형식승인 평가항목 체계

설계적합성 평가항목	설계적합성검사			형식시험
	기술검토서	도면	해석서(계산서)	
3.2.6 위험도분석				
1. 화재안전 위험도	○			
2. 충돌안전 위험도	○			
3. 탈선안전 위험도	○			
3.2.7 철도소프트웨어				
1. 소프트웨어 설계	○			
2. 소프트웨어 구현	○			
3. 소프트웨어 시험				기능 및 동작시험
4. 소프트웨어 설치	○			
5. 소프트웨어 유지보수	○			

6.2.2.1. 설계적합성검사

부품단계, 구성품단계, 완성차단계에서 철도차량의 설계가 기술기준에 적합한지를 검사하고 이를 객관적으로 문서화하는 검사를 의미한다. 신청자는 형식승인을 받으려는 철도차량의 설계도면 및 기술 자료가 기술기준에 부합하는지 여부를 입증하고, 검사기관은 이를 확인한다.

○ 검사절차 (시행지침 제12조)

- 신청자는 형식승인을 받으려는 철도차량의 설계도면 및 기술 자료가 기술기준에 부합하는지 여부를 입증하여야 한다.
- 신청자는 형식승인을 받으려는 철도차량의 설계도면 및 기술 자료가 기술기준에 부합하는지 여부를 입증하여야 한다.
- 검사기관은 신청자가 제출한 입증자료를 통하여 설계적합성을 확인하게 되며, 입증자료가 불충분하다고 판단되는 경우 신청자에게 추가적인 자료 보완을 요청할 수 있으며, 검사기관은 신청자에게 불충분 사유를 기술한 자료를 통보해 주어야 한다.
- 검사기관은 신청자가 제출한 입증자료, 설명서의 검토만으로는 설계적합성의 확인이 불가능하다고 인정되는 경우에는 신청자의 동의하에 현장실사를 할 수 있다.
- 신청자는 설계적합성을 입증하기 위하여 공학적(Engineering) 해석을 실시할 수 있으며 이 경우 신청자는 검사기관에 공학적 해석 자료를 제출하여야 한다.
- 검사기관은 신청자가 설계적합성을 입증하였다고 판단되는 경우에는 신청자에게 철도차량 설계적합성 검사 확인서를 발급한다.

○ 관련문서

- 철도차량 설계적합성확인서 <시행지침 서식1>

○ 관련법령

- 시행규칙: 제48조 (철도차량 형식승인검사의 방법 및 증명서 발급 등) 제1항 제1호
- 시행지침: 제12조 (설계적합성검사 등) 제1항 ~ 제5항
- 기술기준: Part31 고속철도, Part41 일반철도, Part51 도시철도

6.2.2.2. 합치성검사

형식승인의 일부로서 철도차량이 부품단계, 구성품단계, 완성차단계에서 설계적합성검사로 확인된 설계와 합치하게 제작되었는지 검사하고 이를 객관적으로 문서화하는 것이다.

○ 검사절차 (시행지침 제13조)

- 신청자는 부품, 구성품, 완성차에 대해 시행지침 제12조 제5항에 따라 설계적합성 확인을 받은 철도차량 설계와의 일치여부 (이하 “합치성”이라 한다.)를 입증하여야 한다.
- 신청자는 차량형식 시험의 완성차 시험을 착수하기 전에 부품, 구성품에 대한 합치성 입증을 완료하여야 한다. 단, 시행지침 제9조에 따른 형식승인계획서에 반영된 경우에는 그러지 아니하여도 된다.
- 신청자는 시행지침 별지 제2호 서식의 철도차량 합치성검사 요청서를 다음 각 호의 서류를 첨부하여 합치성검사 시작 7일 전까지 검사기관에 제출하여야 한다. 다만, 외국 등에서 검사가 필요한 경우에는 검사기관과 신청자가 협의하여 그 기간을 조정할 수 있다.
  - 검사의 실시 일정 및 장소
  - 검사의 기준 및 방법
  - 철도차량 설계적합성검사 확인서
- 검사기관은 신청자가 합치성을 입증하였다고 인정되면 철도차량 합치성검사 확인서를 발급한다.

○ 관련문서

- 철도차량 설계적합성검사 확인서 <시행지침 서식1>
- 철도차량 합치성검사요청서 <시행지침 서식2>
- 철도차량 합치성검사확인서 <시행지침 서식3>

○ 관련법령

- 시행규칙: 제48조 (철도차량 형식승인검사의 방법 및 증명서 발급 등) 제1항 제2호
- 시행지침: 제13조 (합치성검사 등)
- 기술기준: Part31 고속철도, Part41 일반철도, Part51 도시철도

6.2.2.3. 형식시험

형식승인검사의 일부로서 철도차량의 부품단계, 구성품단계, 시운전단계에서 철도차량 기술기준에 적합한지를 확인하는 시험을 의미한다. 소프트웨어의 경우 완성검사는 소프트웨어 시험에 대한 산출물 대신 철도차량 기술기준의 5.3.17 “기능 및 동작 시험”으로 대체한다. ([표 23] 참조)

○ 검사절차 (시행지침 제14조)

- 신청자는 형식승인을 받으려는 철도차량이 형식승인기준에 적합한지의 여부를 입증하여야 한다.
- 차량형식 시험은 부품, 구성품, 완성차 시험 “시운전시험(예비주행시험 포함)”으로 구성된다.
- 검사기관은 차량형식시험 절차서에 따른 시험절차의 준수 여부 및 해당 시험 과정에서 계측장비를 통해 획득한 결과의 유효성을 확인하기 위해 차량형식시험에 입회할 수 있다.
- 신청자는 별지 제4호 서식의 철도차량 형식시험 요청서에 다음의 서류를 첨부하여 차량형식 시험 시작 7일 전까지 검시기관에 제출하여야 한다. 다만, 외국 등에서 검사가 필요한 경우 검사기관과 신청자가 협의하여 그 기간을 조정할 수 있다.
  - 차량형식 시험의 실시 일정 및 장소
  - 제11조 제2항 보완된 차량형식시험 절차서
- 신청자는 검시가관에 다음 각 호의 내용이 포함된 차량형식 시험 보고서를 제출하여야 합니다.
  - 차량형식시험 결과가 형식승인기준에 적합한지를 입증하는 분석 자료
  - 차량형식시험의 실시 경위 및 경과
  - 차량형식시험 과정에서 참여한 사람에 대한 인적사항
- 검시가관은 차량형식시험 결과 해당 철도차량이 형식승인기준에 적합하다고 판단되는 경우에는 신청자에게 형식시험 결과 확인서를 발급한다.

○ 관련문서

- 철도차량 형식시험요청서 <시행지침 서식4>
- 철도차량 형식시험결과확인서 <시행지침 서식5>
- 철도차향 형식승인증명서 <시행규칙 서식28>

○ 관련법령

- 시행규칙: 제48조 (철도차량의 형식승인검사의 방법 및 증명서 발급 등) 제1항 제3호
- 시행지침: 제14조 (차량형식 시험 등), 제15조 (차량형식시험 대상 등)

6.2.2.4. 부적합사항

- ○ 검사기관은 형식승인검사 과정에서 다음에 해당하는 부적합 사항 (이하 “형식승인 부적합사항 이라 한다.)이 발생하였을 때 신청자에게 시정조치를 요구할 수 있다.
  - 입증자료・설명자료 등의 위조・변조가 발생한 경우
  - 검사절차의 오류가 있는 경우
  - 신청자가 형식승인검사 과정에서 입증한 사실이 형식승인기준과 상이한 경우
- ○ 시정조치 요구를 받은 신청자는 시정조치계획 또는 그 결과를 검사기관에 제출하여야 한다.
- ○ 검사기관은 신청자가 시정조치를 모두 완료한 경우에 한하여 설계적합성검사 확인서, 합치성검사 확인서, 차량형식 시험 확인서를 발급한다.
- ○ 부적합사항에도 불구하고 검사기관은 형식승인 부적합사항이 경미하여 안전에 중대한 영향이 없고, 시정조치계획이 타당하다고 인정하는 경우에는 일정기간 이내에 시정조치를 완료하는 조건으로 형식승인 검사 확인서를 교부할 수 있다.
- ○ 다만, 신청자는 기한 내에 시정조치를 완료하지 못하였거나, 시정조치계획에 따라 시정조치를 이행하지 못한 경우에는 형식승인검사 확인서를 검사기관에 반납하여야 한다.

6.2.2.5. 형식승인검사 중단

- ○ 검사기관은 부적합사항에 해당하는 경우에는 형식승인검사의 중단을 국토교통부장관에게 요청할 수 있다. 이 경우 검사기관은 형식승인검사의 중단 필요성과 그 사유를 국토교통부장관에게 보고하여야 한다.
- ○ 형식승인검사 중단 요청을 받은 국토교통부장관은 해당 사실을 확인하여 형식승인검사를 중단 할 수 있다. 이 경우 국토교통부장관은 신청자에게 형식승인검사의 중단 사실을 서면으로 통보하여야 한다.
- ○ 국토교통부장관은 형식승인검사를 중단하기 위하여 신청자의 의견을 청취하여야 한다.
- ○ 국토교통부장관은 신청자가 형식승인검사의 재개를 요청하였을 때, 중단의 원인이 된 형식승인 부적합사항의 해소여부를 확인 후 재개여부를 결정하여야 한다. 이 경우 국토교통부장관은 검사기관에게 형식승인 부적합사항의 해소여부를 확인토록 지시할 수 있다.

6.2.3. 제작자승인

국토교통부장관은 제작자가 형식승인을 받은 철도차량을 균일한 품질로 제작할 수 있는 능력을 갖추었는지를 확인하기 위하여 인력, 설비, 장비, 기술 및 제작검사 등 철도차량의 적합한 제작을 위한 유기적 체계를 갖추고 있는지에 대하여 아래와 같이 검사하고 승인한다. (자세한 내용은 “철도차량 기술기준, Part71 제작자승인 기술기준” 참조)

(1) 품질관리체계 적합성 검사

(2) 제작검사

- 형식승인을 받은 철도차량을 제작(외국에서 대한민국에 수출할 목적으로 제작하는 경우를 포함한다.)하려는 자는 국토교통부령으로 정하는 바에 따라 철도차량의 제작을 위한 인력, 설비, 장비, 기술 및 제작검사 등 철도차량의 적합한 제작을 위한 유기적 체계(이할 “철도차량 품질관리체계”라 한다.)를 갖추고 잇는지에 대하여 국토교통부장고나의 제작자승인을 받아야 한다.
- 국토교통부장관은 제작자승인을 하는 경우에는 해당 철도차량 품질관리체계가 국토교통부장관이 정하여 고시하는 철도차량이 제작관리 및 품질유지에 필요한 기술기준에 적합한지에 대하여 국토교통부령으로 정하는 바에 따라 제작자승인 검사를 하여야한다.
- 우리나라가 체결한 협정 또는 가입한 협약에 따라 제작자승인이 면제되는 경우 등 대통령령으로 정하는 경우에는 제작자승인 대상에서 제외하거나 제작자승인검사의 전부 또는 일부를 면제할 수 있다.
- 제작자승인을 위해서는 다음에 대한 계획서를 작성하여야 한다.
  - 철도차량 품질관리체계의 중요한 특성
  - 제작자승인기준 (임시기준 포함, 자세한 사항은 시행지침 제25조 참조)
  - 제작자승인 검사의 면제범위 및 필요성
  - 제작자승인 검사 수행조칙
  - 부적합사항 처리절차

6.2.3.1. 품질관리체계 적합성 검사

철도차량 형식승인을 요청한 철도차량의 품질관리체계가 철도차량제작자승인기술기준에 적합한지 여부를 검사한다.

○ 검사절차 (시행지침 제26조)

- 신청자는 철도차량 품질관리체계가 제작자승인에 적합한지 여부 (이하 “품질관리체계의 적합성” 이라 한다.)를 입증하여야 한다.
- 검사기관은 신청자가 제출한 입증자료, 설명서의 내용이 불충분하다고 판단되는 경우에는 신청자에게 추가적인 자료 및 보완 설명을 요구할 수 있다. 이 경우 검사기관은 신청자에게 불충분 사유를 기술한 자료를 통보하여야 한다.
- 검사기관은 신청자가 품질관리체계의 적합성을 입증하였다고 인정되는 경우에는 <시행지침 서식9>의 철도차량 품질관리체계적합성 확인서를 신청자에게 발급하여야 한다.

○ 관련문서

- 철도차량 품질관리체계적합성 확인서 <시행지침 서식9>

○ 관련법령

- 철도안전법: 제26조3 제2항에 따른 철도차량 제작자승인검사
- 시행규칙: 제53조 (철도차량 제작자승인검사의 방법 및 증명서 발급 등)
- 시행지침: 제26조 (품질관리체계 적합성검사 등)
- 기술기준: Part71 제작자승인

6.2.3.2. 제작검사

철도차량에 대한 품질관리체계의 적용 및 유지여부 등을 확인하는 검사로 입증자료 등의 현장적용·운용 여부 등을 확인하기 위하여 철도차량 제작공정 등에 대한 현장 검사 및 제작관리, 제작공정 등을 확인한다.

○ 검사절차 (시행지침 제27조)

- 검사기관은 신청자가 제출한 품질관리체계의 입증자료 등의 현장적용·운용 여부 등을 확인하기 위하여 철도차량 제작공장 등에 대한 현장 검사를 실시할 수 있다.
- 검사기관은 제작검사 결과 품질관리체계가 제작 현장에 적용·운용 되고 있다고 인정되는 경우에는 철도차량 제작검사 확인서를 발급한다.
- 검사기관은 국토교통부장관이 철도차량 제작자승인증명서를 신청자에게 교부한 이후 90일 이내에 철도차량 제작자승인 보고서를 작성하여 신청자에게 통보해야 한다.
- 철도차량 제작자승인 보고서는 다음의 사항을 포함해야 한다.
  - 제작자승인계획서에 따라 진행된 제작자승인검사의 결과
  - 부적합사항에 대한 시정조치 내용
  - 그 밖에 제작자승인 결과와 관련된 자료
- 검사기관은 철도차량 제작자승인 보고서를 전자문서의 형태로 보관해야 한다.

○ 관련양식

- 철도차량 제작검사 확인서 <시행지침 서식10>
- 철도차량 제작자승인증명서 <시행규칙 서식32>

○ 관련법령

- 철도안전법: 제26조의3 제2항에 따른 철도차량 제작자승인검사
- 시행규칙: 제53조 (철도차량 제작자승인검사의 방법 및 증명서 발급 등)
- 시행지침: 제27조 (제작검사 등)
- 기술기준: Part71 제작자승인

6.2.4. 완성검사

국토교통부장관은 제작자 승인을 받은 자가 철도차량을 판매하기 전에 형식승인을 받은 대로 제작되었는지를 완성검사를 통하여 확인하다. 완성검사를 신청하면 검사기관은 완성검사 신청 이전에 사전기술검토를 실시하여 다음의 사항을 검토 한다.

- 완성검사의 일정 등에 관한 사항
- 신청자가 검사기관등과의 사전 협의가 반드시 필요하다고 요구하는 사항

발주자는 신청자 또는 검사기관 등의 요청이 있는 경우 사전기술검토에 참석할 수 있으며 신청자나 검사기관이 검토하는 사안에 대하여 필요한 정보를 제공하여야 한다. 그리고 신청자는 사전기술검토에서 다음의 정보를 검사기관에 제공해야 한다.

- 완성검사를 받으려는 철도차량과 동일한 철도차량에 대하여 발급된 형식승인의 내용
- 주요 협력업체 현황 및 그 협력업체에서 제공하는 부품, 구성품 및 용역의 내용

6.2.4.1. 완성차량검사 및 안전품목검사

형식승인과 제작자승인을 받은 철도차량에 대하여 출하하기 전에 철도차량이 철도차량 기술기준에 적합(안전과 직결된 주요 부품의 안전성 확보)하고, 형식승인 받은 설계대로 제작되었는지를 확인하는 검사이다.

○ 검사절차 (시행지침 제38조)

- 완성차량검사는 다음의 검사로 구성된다.
  - 완성차검사: 완성검사를 받으려는 철도차량이 형식승인을 받은 대로 제작되었는지 여부를 확인한다.
  - 안전품목검사: 철도차량 운행 중 분리, 탈선, 전복, 화재 등 열차사고와 밀접한 관련이 있는 품목에 대한 안전성 및 성능을 확인한다.
- 안전품목검사의 대상은 “철도차량기술기준”에서 정한 안전품목검사 항목에 따르며, 완성차검사의 대상은 “철도차량기술기준”에서 정한 차종별 완성차시험을 따른다.
- 안전품목검사 대상 중에서 법 제27조에 따라 형식승인을 받은 철도용품에 대해서는 검사를 면제한다. 다만, 검사기관은 철도안전사고와 직결되는 용품에 대해서는 검사를 실시할 수 있으며, 그 검사대상 및 범위는 시행지침 [표 24]와 같다.
- 신청자는 안전품목에 대한 성능 및 안전성 확보여부와 완성검사를 받으려는 철도차량이 형식승인을 받은 대로 제작되었는지 여부를 입증해야 한다.
- 전문기관은 완성차량검사를 위하여 필요한 경우 신청자에게 상세한 설명을 요구할 수 있다.
- 검사기관은 전문기관이 요청한 경우 완성차량에 필요한 자료를 전문기관에 제출하고, 완성차량검사 업무를 지원·협조해야 한다.
- 전문기관은 신청자가 입증한 사실이 기술기준에 적합하고 형식동등성이 인정되는 경우에 신청자에게 완성차량검사 확인서를 발급해야 한다. 이 경우 전문기관은 검사기관에서 완성차량검사 확인서 발급사실을 통지해야 한다.

○ 관련양식

- 완성차량검사확인서 <시행지침 서식11>

○ 관련법령

- 철도안전법: 제26조6의 제1항
- 시행규칙: 제57조 (철도차량 완성검사의 방법 및 검사필증 발급 등)
- 시행지침: 제38조 (완성차량검사의 실시 등)
- 기술기준: Part31 고속철도, Part41 일반철도, Part51 도시철도

표 24 형식승인을 받은 철도용품 중 완성차량검사의 대상 품목 및 검사항목

품 목 명		검사항목	면제되는 검사
차륜		소재검사, 비파괴검사	외관검사, 가공치수검사, 표면조도검사
차축(일반차축)		소재검사, 비파괴검사	외관검사, 가공치수검사, 표면조도검사
오일댐퍼		감쇄력검사, 작동(댐퍼성능)검사	재료시험, 작동유시험, 치수외관검사, 완충고무특성시험, 온도시험, 내구성시험
1차 스프링	코일스프링	소재시험, 스프링상수 검사	외관/치수검사, 염수분모시험, 비파괴시험, 게재물검사, 탈탄시험
	고무스프링	소재시험, 스프링상수 검사	외관/치수검사, 온도시험, 내마모성시험, 피로시험, 압축영구줄음 시험, 반발탄성 시험
2차 스프링 (공기스프링)		소재시험, 스프링상수 검사	외관/치수검사, 내용적시험, 피로시험, 마모시험, 기밀시험
연결기		소재검사, 비파괴검사	외관검사, 가공치수검사, 연결기헤드압축시험, 연결기헤드인장시험, 호환성시험, 공기누설시험, 전기적특성, 온도상승시험, 살수시험, 기능검사, 외관검사, 가공치수검사
제동실린더		기능 및 작동검사	외관검사, 치수검사, 재료시험, 온도시험, 진동시험, 내구성시험, 압력강도시험
제동차말제(합성수지)		소개검사, 특성검사	외관검사, 치수검사, 압축강도시험, 충격강도시험, 결합력시험, 최고온도측성시험
제동디스크(합금소재)		소재검사, 특성검사	외관검사, 치수검사, 허용 잔류편심 확인 시험
신호보안장치		전기특성시험	외관구조검사, 치수검사, 저온시험, 고온시험, 고온 · 고습시험, 온도사이클시험, 진동시험, 충격시험, 방수시험

6.2.4.2. 주행시험

철도차량이 형식승인을 받은 대로 성능과 안전을 확보하였는지 운행선로 시운전 등을 통하여 최종 확인하는 검사로 예비주행시험과 시운전시험으로 구성된다.

○ 시험절차 (시행지침 제39조)

- 검사기관은 신청자의 요청이 있는 경우 15일 이내에 주행시험을 수행해야 한다. 다만, 신청자의 동의가 있거나, 외국에서의 검사업무가 진행되는 경우에는 주행시험 착수시기를 조정할 수 있다.
- 검사기관은 완성차량 검사확인서가 신청자에게 발급되었는지를 확인한다.
- 예비주행시험과 시운전시험의 대상 및 최소 주행거리에 관한 사항은 다음의 사항을 준용할 수 있다.
  - 예비주행시험: 시험선로에서 5,000킬로미터 ([표 26]의 경우 1,000 킬로미터 이상)
  - 시운전시험: [표 25]를 따른다.
- 검사기관은 신청자가 주행시험 절차서에 따라 주행시험을 실시하였는지 여부 및 결과를 확인하기 위해 예비주행시험과 시운전시험에 현장 입회할 수 있다.
- 검사기관은 주행시험의 결과 해당 철도차량이 형식승인을 받은 대로 성능과 안전성을 확보하였을 경우 신청자에게 확인서를 발급한다.

○ 관련양식

- 철도차량 주행시험확인서 <시행지침 서식12>
- 철도차량 완성검사필증 <시행규칙 서식35>

○ 관련법령

- 철도안전법: 제26조6 (철도차량 완성검사)의 제1항
- 시행규칙: 제57조 (철도차량 완성검사의 방법 및 검사필증 발급 등)
- 시행지침: 제39조 (주행시험의 실시 등)
- Part31 고속철도, Part41 일반철도, Part51 도시철도

표 25 차종별 시운전 주행거리

차 종	최고운행속도(킬로미터/시간)	주행거리(킬로미터)
고속철도차량	300 이상	35,000 이상
	200~300 미만	15,000 이상
일반철도차량	151~200 미만	5,000 이상
	150 이하	1,000 이상
도시철도차량	151~200 미만	5,000 이상
	150 이하	1,000 이상

비고: 일반철도차량 중 객차 및 화차는 검사기관이 차량발주자와 협의하여 주행거리를 별도로 정할 수 있다.

6.3. 면제조건

(1) 철도안전법 제26조 제4항에 따라 국토교통부장관은 다음의 어느 하나에 해당하는 경우에는 철도차량 형식승인검사의 전부 또는 일부를 면제할 수 있다. <개정 2013.3.23.>

○ 시험・연구・개발 목적으로 제작 또는 수입되는 철도차량으로서 대통령령으로 정하는 철도차량에 해당하는 경우

○ 수출 목적으로 제작 또는 수입되는 철도차량으로서 대통령령으로 정하는 철도차량 [표 26]에 해당하는 경우

○ 대한민국이 체결한 협정 또는 대한민국이 가입한 협약에 따라 형식승인검사가 면제되는 철도차량의 경우

○ 그 밖에 철도시설의 유지・보수 또는 철도차량의 사고복구 등 특수한 목적을 위하여 제작 또는 수입되는 철도차량으로서 국토교통부장관이 정하여 고시하는 경우

(2) 철도안전법 시행규칙 제47조 (철도차량 형식승인의 경미한 사항변경)에 해당하는 경우에는 철도차량 형식승인검사의 전부 또는 일부를 면제할 수 있다.

○ 법 제26조 제2항 단서에서 “국토교통부령으로 정하는 경미한 사항을 변경하려는 경우” 란 다음의 어느 하나에 해당하는 변경을 말한다.

- 철도차량의 구조안전 및 성능에 영향을 미치지 아니하는 차체 형상의 변경
- 철도차량의 안전에 영향을 미치지 아니하는 설비의 변경
- 중량분포에 영향을 미치지 아니하는 장치 또는 부품 배치 변경
- 동일 성능으로 입증할 수 있는 부품의 규격 변경
- 그 밖에 철도차량의 안전 및 성능에 영향을 미치지 아니한다고 국토교통부장관이 인정하는 사항의 변경

○ 법 제26조 제2항 단서에 따라 경미한 사항을 변경하려는 경우에는 다음의 서류를 첨부하여 국토교통부장관에게 제출하여야 한다.

- 해당 철도차량의 철도차량 형식승인증명서
- 제1항 각 호에 해당함을 증명하는 서류
- 변경 전후의 대비표 및 해설서
- 변경 후의 주요 제원
- 철도차량기술기준에 대한 적합성 입증자료 (변경되는 부분 및 그 와 연관되는 부분에 한정한다.)

표 26 특수한 목적으로 제작 또는 수입되는 철도차량 (철도차량 형식승인 면제차량)

분 류	명 칭	용 도
사 고 복구용차	사고복구차 (Two Way Motor Car)	선로 및 육로주행겸용 사고복구용차
	사고복구용 기중기 (Crane)	사고복구용 기중기
	화재진압용 소방차 (Firefighting Train)	철도차량 및 철도노선 화재사고 복구용차
작 업 차	복합 침목교정차 (Multiple Tie Tamper)	충다지기 및 궤도틀림상태 (줄맞춤, 면맞춤, 수평 등) 정정 작업차량
	자갈 다지기차 (Ballast Compactor)	도상자갈표면 및 어깨달고 다지기 작업차량
	자갈 제거차 (Ballast Cleaner)	일반 및 분기부 도상 전단면 자갈치기 작업차량
	분기기 자갈 제거차 (Switch Cleaner Track Undercutter)	분기부 도상 전단면 자갈치기 작업차량
	자갈 정리차 (Ballast Regulater)	도상 자갈 정리 및 단면 형성 작업차량
	분기기 침목교정차 (Switch Tie Tamper)	분기부 침목 다지기 및 궤도틀림 정정 작업차량
	궤도 안정기 (Dynamic Track Stabilizer)	궤도 안정화 작업차량
	컨베어 호퍼차 (Conveyor Hopper Car)	자갈, 토사적재 운반 하역 작업차량
	가선차 (OverHead Wiring Car)	전차선 가선 작업차량
	가선 보조작업차 (Worker's Operation Trolley)	가선작업 보조 작업차량
	전주 작업차 (Crane Working Car)	전주건식, 중량물 들기 작업차량
	전주 적재차 (Master Loading Trolley)	전주 적재 작업차량
	전선 적재차 (Flat Car)	전선드럼 적재 작업차량
	골재차(2대1조) (Aggregate Loading Trolley)	골재 적재차량
	콘크리트 믹서카 (Concrete Mixing Car)	콘크리트 믹서 작업차량
	모터카 (Motor Car)	궤도보수, 전철보수, 점검 및 검측 현장 작업차량
	굴삭차 (Drilling Hammer Car)	현장 암반 및 토공 굴착 작업차량
	작업자 침식차 (Small Couchette Trolley)	작업자 침식차량
	유조트롤리 (Fuel Tank Loading Trolley)	보선장비 연료조달
	살수트롤리 (Water Tank Loading Trolley)	터널내 지하공간 물청소작업

제 3 절 해외 철도 안전 체계

1. 유럽

1.1. 유럽 철도 안전 체계

유럽은 유럽연합에서 철도 안전 및 상호운용 관련 지침(Directive 2004/49/EC, 2008/57/EC)을 제정하고, ERA(European Railway Agency)에서 유럽 철도 산업 및 관련 제조사들이 필수적으로 준수해야 할 요구사항들을 상호운용을 위한 기술 사양서(TSI: Technical Specifications for Interoperability) 형식으로 제공하며, 각 회원국은 관련 지침에 의거해서 자국의 철도관련 법/제도를 개정하고 TSI를 근간으로 기술적으로 준수해야 할 요구사항을 제정하여 운영한다. 이를 통해, 유럽연합 가입국들은 점진적으로 자국 실정을 반영한 표준화된 철도 안전 법/제도 및 필수 준수 사항을 보유하게 되고, TSI에서 규정된 주요 표준 및 기술 사양 등을 자연스럽게 유럽 각 회원국 및 관련 철도 산업이 준수해야 하는 요구사항으로 정착시켜 유럽 철도의 상호 운용성을 보장한다.

유럽연합 철도 안전 지침에 따라 정의된 철도 안전 관련 주요 관련자들은 Notified Body(정부에서 인증 받은 철도 인증 조직), Independent Safety Assessor(ISA: 독립 안전 평가자), National Safety Authority(정부 안전 관련 기관), Investigating Body(사고 및 사건 조사 조직), Infrastructure Manager(철도 인프라 및 안전 관리 조직), Railway Undertaking(철도 산업 종사자)등이 있다.

특히 독립 안전 평가자는 미국과는 차별화된 철도 안전 검증 단체인데, 이들은 정부 철도 기관 등에 고용되어 철도 관련 사업자들이 개발하는 안전 필수 시스템이 안전 필수 요구사항 및 표준을 만족하는 지를 평가하는 역할을 한다. 즉, 철도 산업 종사자들은 철도 인증을 득하기 위해서는 TSI의 요구사항에 따라 개발해야 하며, 이를 독립 안전 평가자나 철도 인증 조직이 검사하는 체계로 운영된다.

EN 50128은 주요 시스템의 필수 기술 사양서(TSI)에 소프트웨어 관련 준수 표준으로 되어 있어서, 관련 사업자들은 TSI의 요구사항 준수에 대하여 평가를 받고 철도 인증을 받아야 하므로 EN 50128 표준은 법/제도적으로 강제 적용된다. 예를 들면, “Applicable standards in CR Control-Command and Signaling TSI(2006/679/EC)”의 Section 6.1.2 Index A2 Interoperability Constituents Modules 및 Section 6.2.2.3 Index A2, “Applicable standards in HS Rolling stock subsystem TSI(208/232/EC)”의 Section 4.2.7.13 부분에서 소프트웨어 개발 시 EN 50128 준수 항목이 명시되어 있다.

유럽 연합에서 규정한 지침(Directive) 및 TSI가 유럽 연합 회원국에 적용되는 방식을 간략히 살펴보자면, 독일의 경우 독일 교통부 산하 독립 조직으로 독일연방철도국

(Eisenbahn-Bundesamt: EBA)이 있으며, 독일연방철도국은 유럽연합지침(Directive 2004/49/EC) 및 TSI에 따라 안전 규정을 제정하고 안전 인증을 수행하며, 소프트웨어 안전 표준에 대해서는, 철도 건설 및 운영 규정(Die Eisenbahn-Bau-und Betriebsordnung: EBO)에 철도차량용 안전 관련 소프트웨어 개발은 EN 50128을 준수하도록 강제되어 있다.

1.2. 철도차량 및 용품 승인 절차

상호운용성 검사 전문기관인 NoBo(Notified Body)는 TSI(유럽연합 상호운용성 인증 기준)에 의거하여 검사 업무를 수행하고, 국내운영성 검사 전문기관인 DeBo (Designated Body)는 NNRT(Notified National Technical Rule)에 의거하여 검사 업무를 수행한다. 각각의 검사 결과는 NSA(유럽연합 가맹국 철도안전 주무부처)에 차량 및 용품승인 결과를 보고하고 NSA는 검사 결과를 바탕으로 승인여부를 결정하여, 승인 완료 된 제품은 등록 후 ERA(유럽철도국, European Railway Agency)에 보고하는 것으로 마무리 된다.

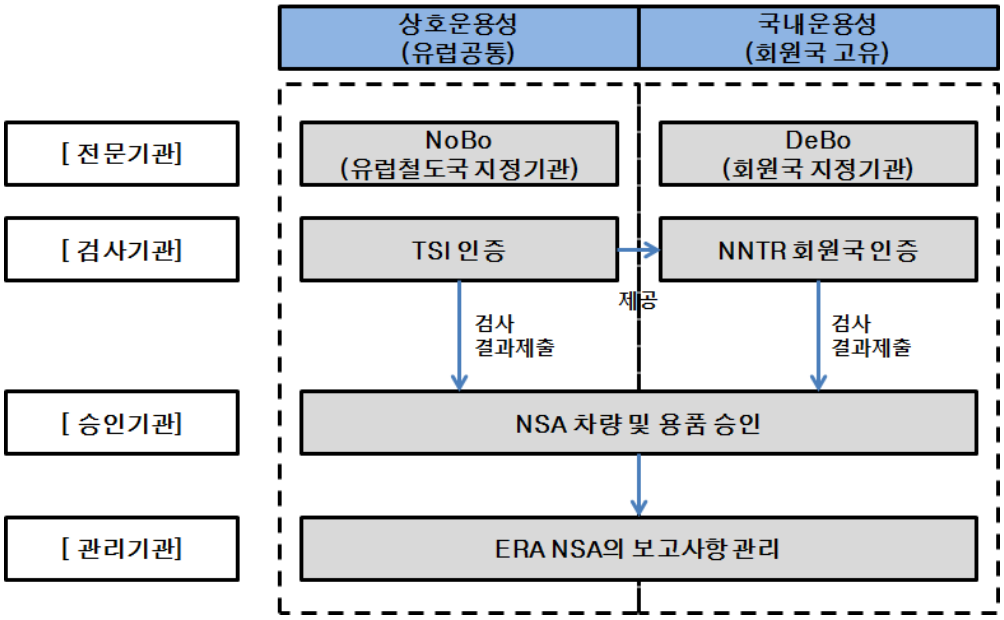


그림 24 유럽연합의 철도차량 및 용품 승인체계와 절차

1.3. 철도운영 및 시설관리

유럽연합의 철도운영 및 시설관리 안전승인체계는 철도차량 및 용품과는 달리 국내운영성과 상호운용성에 대한 검증을 분리하여 시행하지 않는다. 유럽연합의 철도운영 및 시설관리기관 안전승인은 승인신청자가 자체 수립한 안전관리체계에 대한 검사와 승인으로 구성되며, 국가가 설립한 독립기관이 검사와 승인을 일원적으로 수행한다.

승인절차는 철도운영 및 시설관리기관이 NSA에 안전승인을 신청하면 시작된다. 기관이 NSA에 신청을 하면 NSA는 국가안전규칙에 의거하여 안전관리체계의 검사와 승인을 수행하고 그 결과를 ERA에 보고하면 마무리 된다.

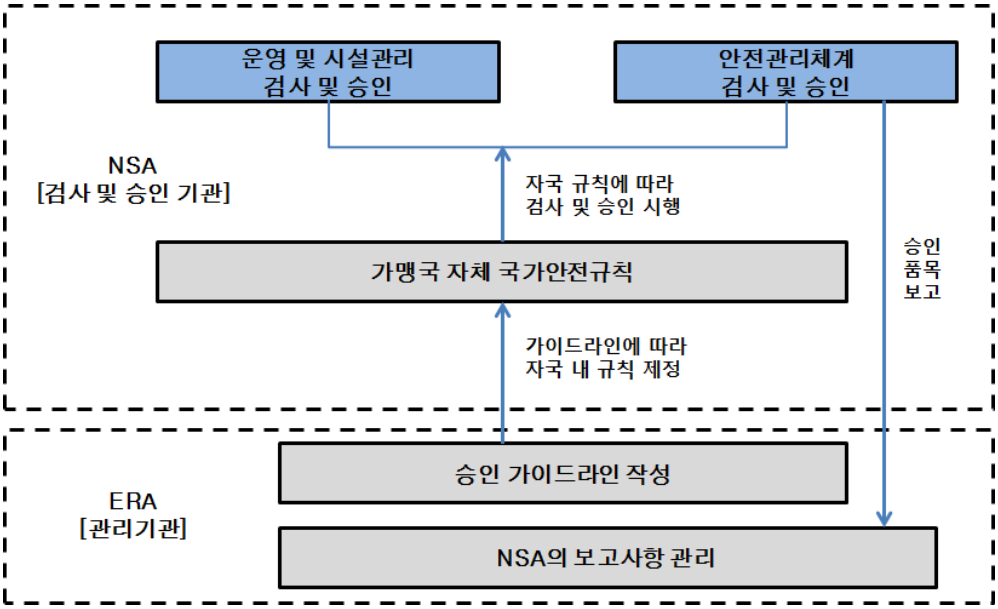


그림 25 유럽연합의 철도운영 및 시설관리 승인체계와 절차

2. 미국

2.1. 철도안전 관련 법체계

미국의 철도안전에 관한 사항은 교통법(US Code Title 49, 이하 USC 49)와 하위법령(CFR Title 49, 이하 CFR 49)에 규정되어 있다. USC 49 철도프로그램 중 철도안전에 관한 내용은 파트 A에 규정되어 있다.

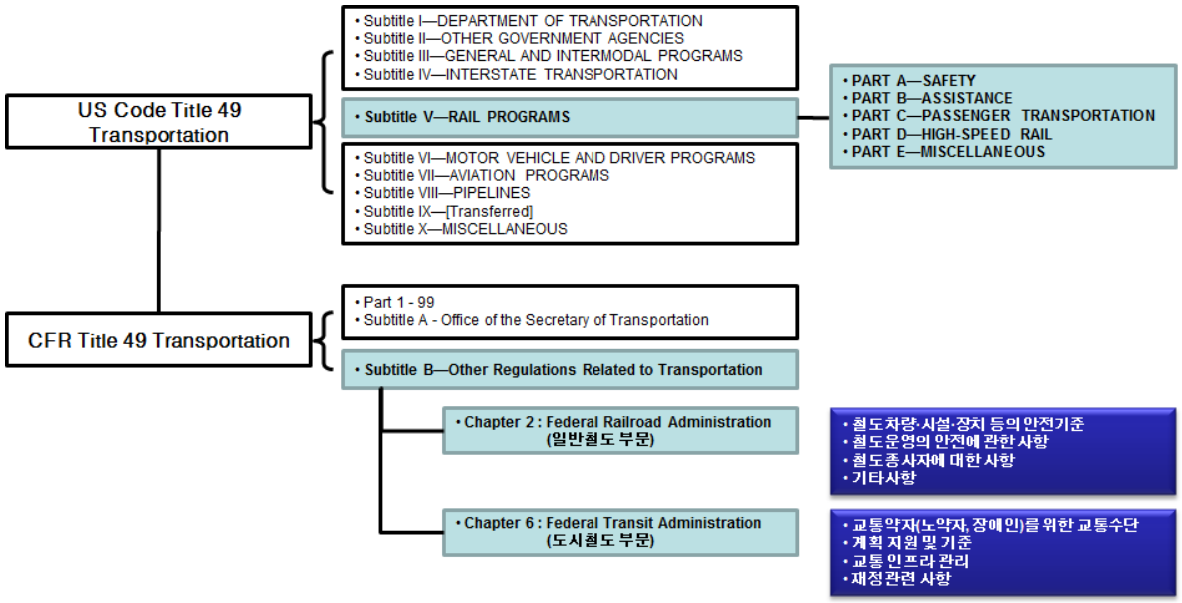


그림 26 미국 철도안전 관련 법체계

2.1.1. USC 49 철도안전 주요내용

총칙에서는 철도안전에 관한 국가 또는 주정부의 권한 및 역할 등에 관한 사항을 규정하고 있다.

- ○ 국가는 철도안전 분야 전반에 대해 공공의 의견을 수렴하고, 안전에 관한 정보와 기준 등을 고려하여 관련 법령을 수립한 후, 이를 집행해야 한다.
- ○ 국가는 철도장비, 시설, 차량, 운영에 대해 적용하도록 발급된 안전규정 및 훈령을 집행하기 위해 필요한 조사 및 감독의 활동을 규정할 수 있다. 주 위원회에 의해 철도장비, 시설, 차량, 운영이 규제될 경우, 주정부는 조사 및 감독 활동에 참여할 수 있다.
- ○ 국가는 공공기관 또는 자격을 갖춘 개인에게 철도장비, 시설, 차량, 운영에 대한 검사, 시험, 테스트 등을 위탁할 수 있으며, 검사 시작과 함께 이들은 미국정부의 직원이 된다.

안전에 관한 특별한 사항에서는 철도차량, 장비, 시설, 운영 등에 대한 제도적 요구사항들을 규정하고 있다.

- 여객수송 철도차량에 대한 최소안전기준, 평면교차문제 해결방안, 기관사 운전면허 또는 자격증명, 자동열차제어시스템의 필요조건 등
- 열차운행의 안전에 관한 모니터링 내용에 대한 기록, 부당한 변경 또는 기능이 부족한 열차운행 모니터링 장치의 사용 금지 등
- 음주 또는 금지 약물 복용 여부에 대한 검사
- 파워브레이크, 궤도안전기준, 기관차 및 철도차량의 시인성 등 기술적 사항
- 공중에 의한 철도 훼손 가능성에 대한 경고, 철도통과 및 시설훼손을 방지하기 위한 전략 등 철도보호에 관한 사항
- 종사자들의 근무환경 관련 사항 등

철도용품에 관한 조항에서는 철도차량에 사용되는 용품에 관한 기술적 요구조건들을 규정하고 있다.

신호시스템에 관한 조항에서는 신호시스템의 검사, 시험, 조사 및 설치 등에 관한사항을 규정하고 있다.

기관차에 관한 조항에서는 차량의 검사, 수리, 조사 및 사용을 위한 요구조건 등 에 관한 사항을 규정하고 있다.

사건 및 사고에 관한 조항에서는 철도사고에 대한 조사, 보고서 작성 등에 관한 사항을 규정하고 있다.

철도서비스 시간에 관한 조항에서는 철도종사자 근무 의무시간의 제한에 관한 사항을 규정하고 있다.

2.1.2. CFR 49 철도안전 주요내용

CFR 49 는 2장 FRA(연방철도국) 에서 철도 전반에 관한 사항을 규정하고 있으며, 6장 FTA (연방대중교통국)의 고정궤도시스템 조항에서 철도안전에 관한 사항을 규정하고 있다.

CFR 49-2 장에서는 수록된 철도안전 관련 조항은 FRA(연방철도국)와 연관된 철도를 대상으로 하며, 철도운전, 종사자 관리, 철도차량, 시설, 장치의 기술적 요구조건 등을 규정하고 있다. 도시철도 등은 이 규정의 적용대상에서 제외된다.

- 철도차량, 시설, 장치 등에 관한 안전기준 등을 규정 : 궤도, 철도화차, 철도차량 안전유리, 기관차, 철도안전용품, 여객장치, 제동장치, 신호시스템 등
- 철도운영 안전에 관한 사항 등을 규정 : 철도운영에 관한 규칙 및 실행, 철도사건 사고 조사 및 보고, 여객열차 비상대응, 철도작업장 안전 등
- 철도종사 관련 사항을 규정 : 알콜 및 약물사용에 대한 통제, 철도종사자 근무시간, 철도기관사 면허 및 자격증명,
- 기타 사항 : 철도 경찰공무원, 철도안전 집행 절차

CFR 49-6장에서는 고정궤도시스템 조항의 적용을 받는 철도는 도시화 구역에서 건설・운영되거나 또는 FTA의 재정지원을 받는 경전철, 중전철, 모노레일, 트롤리 등이며, FRA 에 의해 규제를 받는 철도는 해당되지 않는다.

- 철도시스템 안전프로그램 및 보안계획에 관한 사항 : 기준, 요구조건, 항목 등
- 안전에 대한 검토 및 교정을 위한 시행계획 등

2.2. 인증 대상 및 기관

일반철도의 경우 연방 차원의 안전 승인은 없으며, 연방정부가 설립한 관리, 감동기관이 일반철도운영 및 시설관리 관련 연방법령 준수여부를 점검 및 조사한다. 도시철도의 경우 승인 신청자가 수립한 안전관리체계와 자체 안전심사결과에 대하여 주 정부가 설립한 전문기관이 검사를 수행하고 연방정부 산하기관이 승인을 수행한다.

2.2.1. 승인 기관

연방교통부(DOT) 산하 FRA는 전문기관인 교통기술센터(TTCI)를 AAR과 공동창립하고 철도차량 및 용품 연방규정(CFR 49)을 제정하여 관리, 감독을 위탁한다.

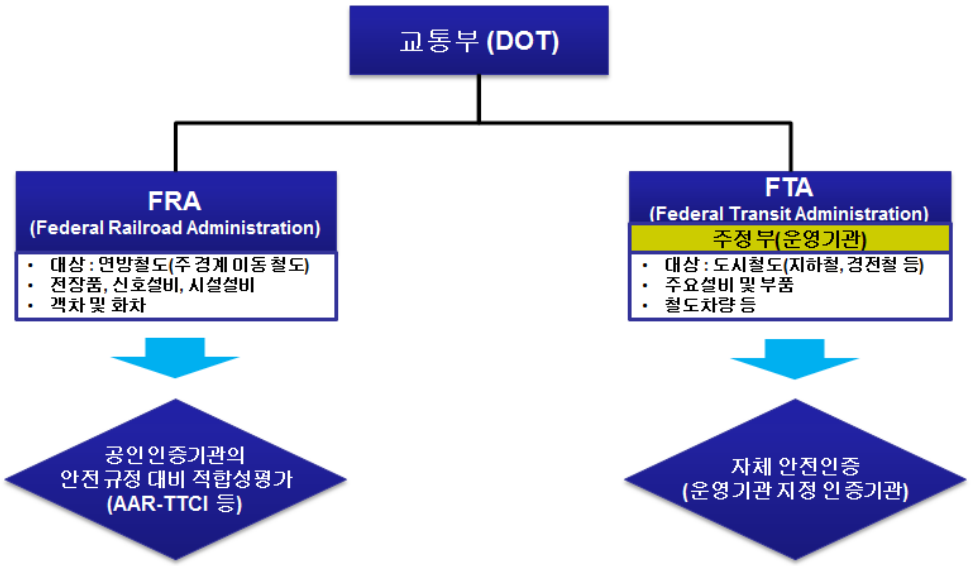


그림 27 미국 철도 안전 인증 대상 및 기관

연방철도국(FRA)는 철도차량 및 용품/철도운영 및 시설관리/철도종사자 등 철도 전 분야의 안전기준을 설정하는 기본지침인 철도안전 연방규정 CFR 49를 제정하고, 주 경계를 넘는 연방 차원의 여객 및 화물철도의 안전업무를 관할한다.

연방대중교통국(FTA)는 주 내부 대중교통(도시철도 포함)의 안전업무를 관할하며, 주 안전감독국(SSO)이 작성한 안전 및 보안 프로그램을 승인한다. 또한 이를 기준으로 주 내부 도시철도운영 및 시설관리기관의 안전관리체계를 승인 후 관리 감독한다.

2.3. 철도차량 및 용품 승인 절차

- 일반철도는 철도운영 및 시설관리기관은 안전관리체계를 수립하고 자체 안전심사를 수행함. FRA (연방철도국) 산하 ORS (철도안전사무소, Office of Railroad Safety)에서는 일반철도의 안전관련 기본지침인 연방규정(CFR 49 제2장)을 기준으로 일반철도운영기관의 안전관리체계에 대한 관리·감독을 수행한다.
- 도시철도는 철도운영 및 시설관리기관이 SSPP(시스템 안전 프로그램 계획)와 SSP (시스템 보안 계획)를 작성하여 안전관리체계를 수립하고 자체 안전심사를 수행하여 이를 SSO (주 안전감독국)에 제출한다. SSO는 SSPS (안전 및 보안 프로그램 규격에 의거하여 검사를 수행하고 결과를 FTA (연방대중교통국)에 제출 하고, FTA는 검사결과를 검토하여 최종적으로 승인여부를 결정하며 SSO를 통하여 사후 관리·감독을 수행한다.

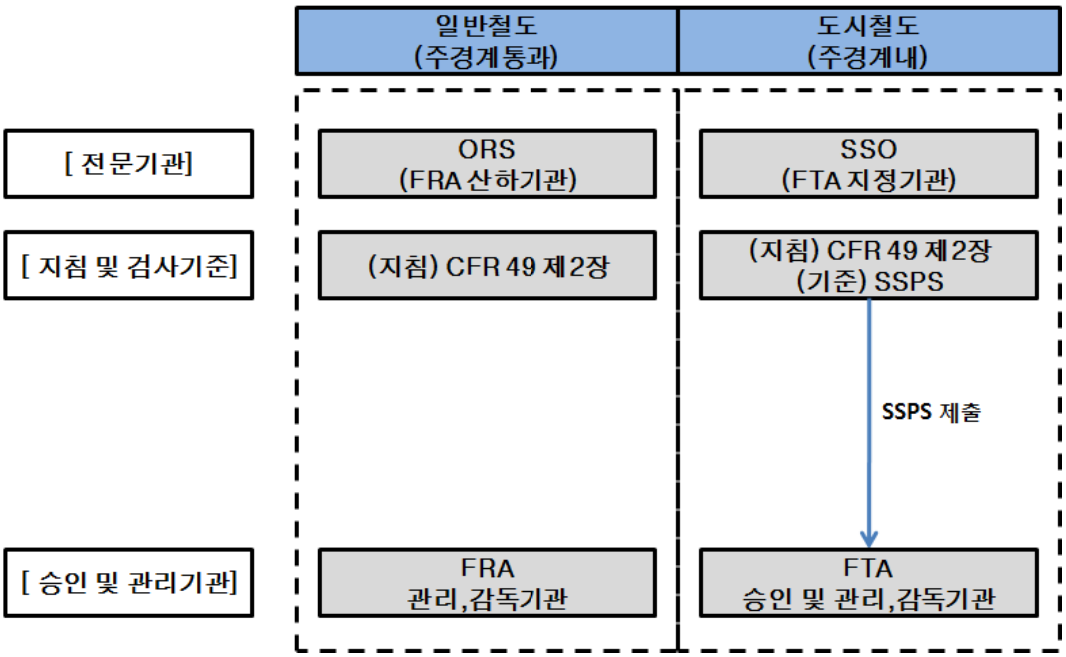


그림 28 미국 철도운영 및 시설관리 안전승인체계 및 절차

2.4. 철도 기술 기준

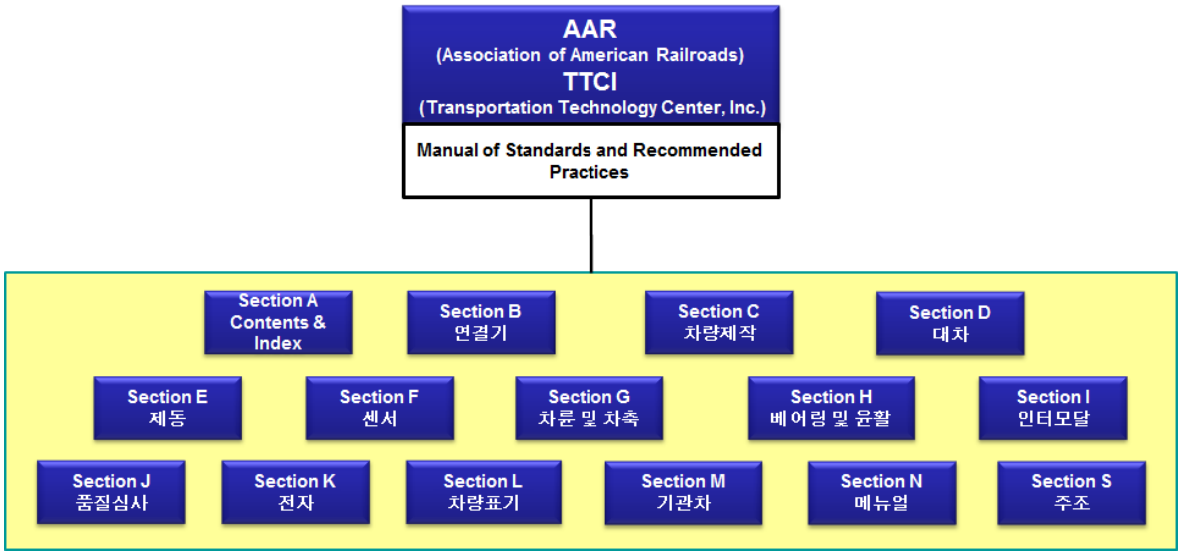


그림 29 미국 철도 기술 기준

3. 일본

3.1. 철도안전 관련 법체계

3.1.1. 철도차량 및 용품

국토교통성에서는 철도사업법 국토교통성령(한국의 시행규칙에 해당)을 통하여 철도차량 및 용품, 궤도 및 기타 시설에 대한 국내 기술기준을 규정하여 공표한다. 기술기준에 대한 철도차량 및 용품제작사와 철도운영 및 시설관리기관의 이해를 돕기 위하여 별도 해석기준(강제력 없음)을 설정한다.

철도차량 인증검사에 필요한 시행지침은 검증을 직접 수행하는 철도운영기관(인증대상 철도차량의 발주자)이 자체적으로 수립한다.

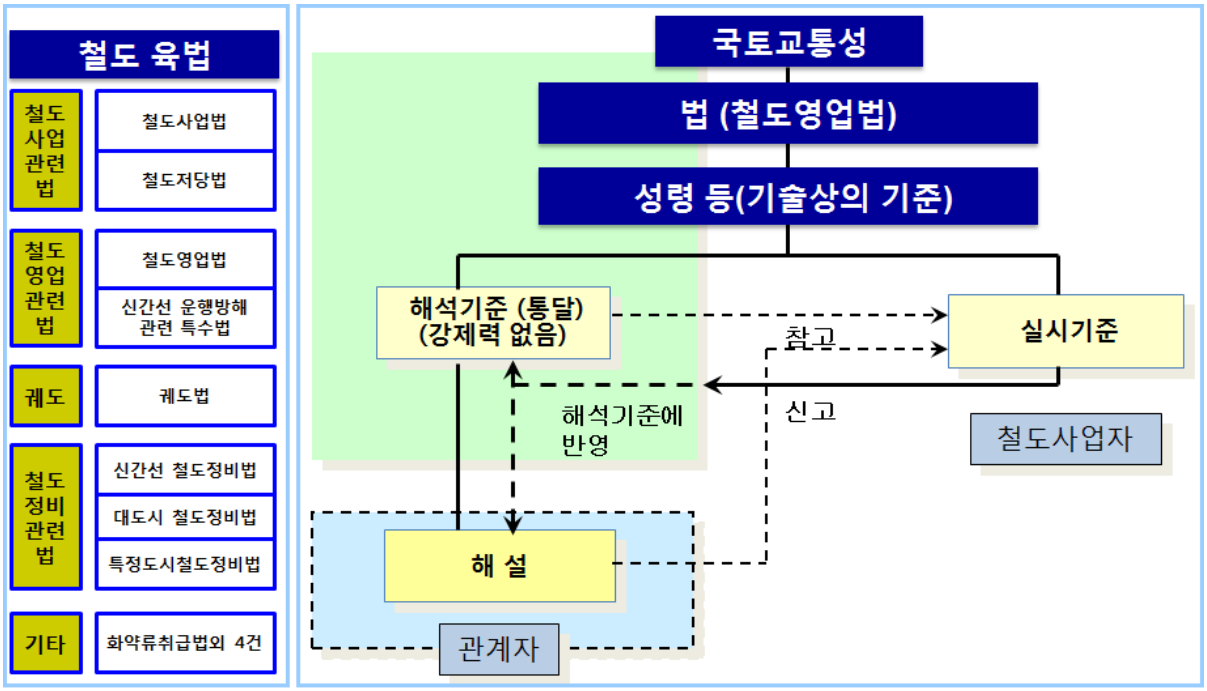


그림 30 일본 철도안전 법체계

3.1.2. 철도운영 및 시설관리

철도사업법 국통교통성령 및 규칙은 철도 운영, 시설, 차량, 운전, 보안, 안전지침을 규정하고 있다. 철도운영 및 시설관리기관은 국토교통성 기술기준을 실제 현장에서 적용하기 위한 별도의 실시기준을 자체적으로 작성하여 적용한다.

3.2. 인증대상 및 기관

3.2.1. 철도차량 승인

국토교통성(MLIT)는 철도 분야 법령 및 규제의 집행, 정책 결정 및 지원을 총괄한다. 철도차량 제작자 및 운영기관의 최초 차량 형식에 대한 승인신청을 검토 후 승인하며, 이후 해당 차량의 양산 단계에서 철도차량 제작사와 운영기관이 합동으로 수행한 자체 검사 결과를 보고받고 사후 관리, 감독을 수행한다.

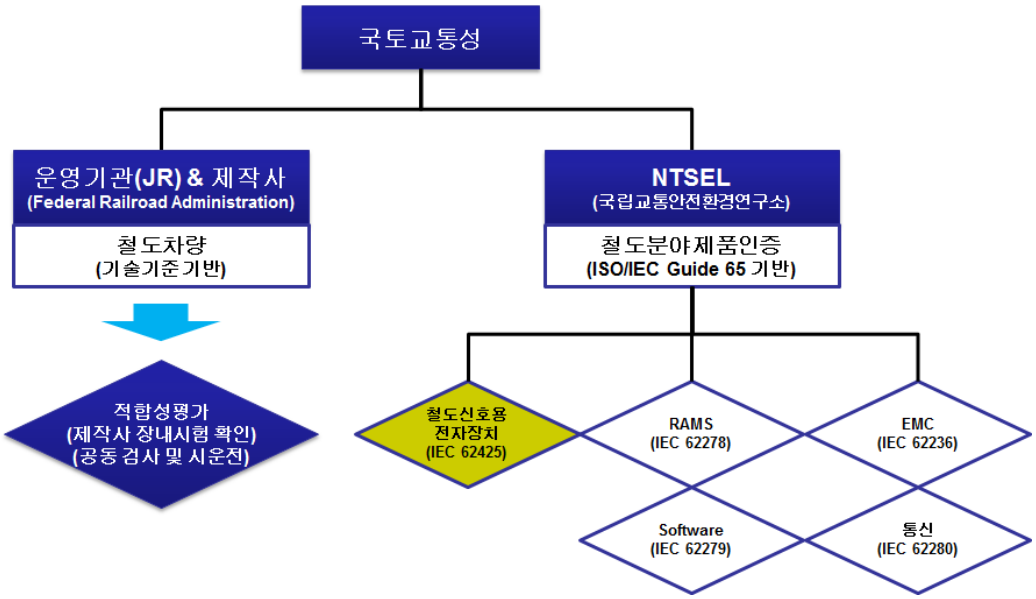


그림 31 일본 철도안전 승인 기관 및 내용

○ 철도용품 승인

국립 교통안전환경연구소(NTSEL)은 일본정부 예산으로 운영되는 독립행정법인(한국의 기타공공기관과 유사하다.)으로, 한국의 교통안전공단과 유사한 설립 배경을 가진다. 산하기관인 NTCC에서 수행한 철도용품에 대한 인증검사 결과를 검토 후 최종적으로 승인한다.

철도 시스템 수출 시 개별 분야의 기술적 안전성과 신뢰성에 관한 규격(IEC 62278, RAMS) 등 타 철도 관련 규격 인정 및 상호인증을 추진하여, 일본 철도차량 및 용품 제조업체들의 해외 진출 시 안전 규격 인증 심사 기간 및 비용 절감을 기대하고 있다.

3.3. 철도차량 및 용품 승인 절차

철도운영 및 시설관리에 대한 일괄적인 안전승인이 존재하지 않으며 운영 및 시설관리 기관은 자체 안전관리 및 검증을 수행하고 국토교통성에 신고한다. 국토교통성은 이를 토대로 관리, 감독을 실시하며 결과에 따라 시정 및 제제조치를 취할 수 있다.

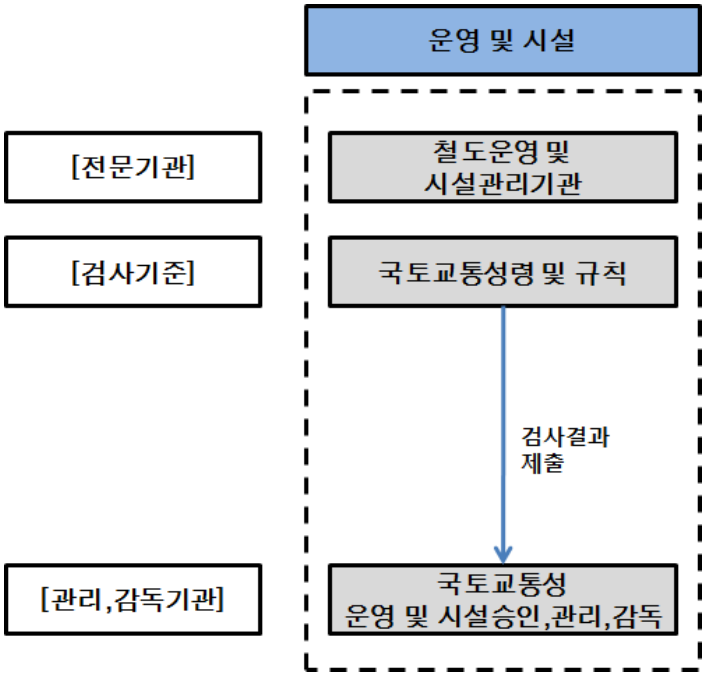


그림 32 일본 철도운영 및 시설 안전관리체계 구성요소

3.4. 철도 기술 기준

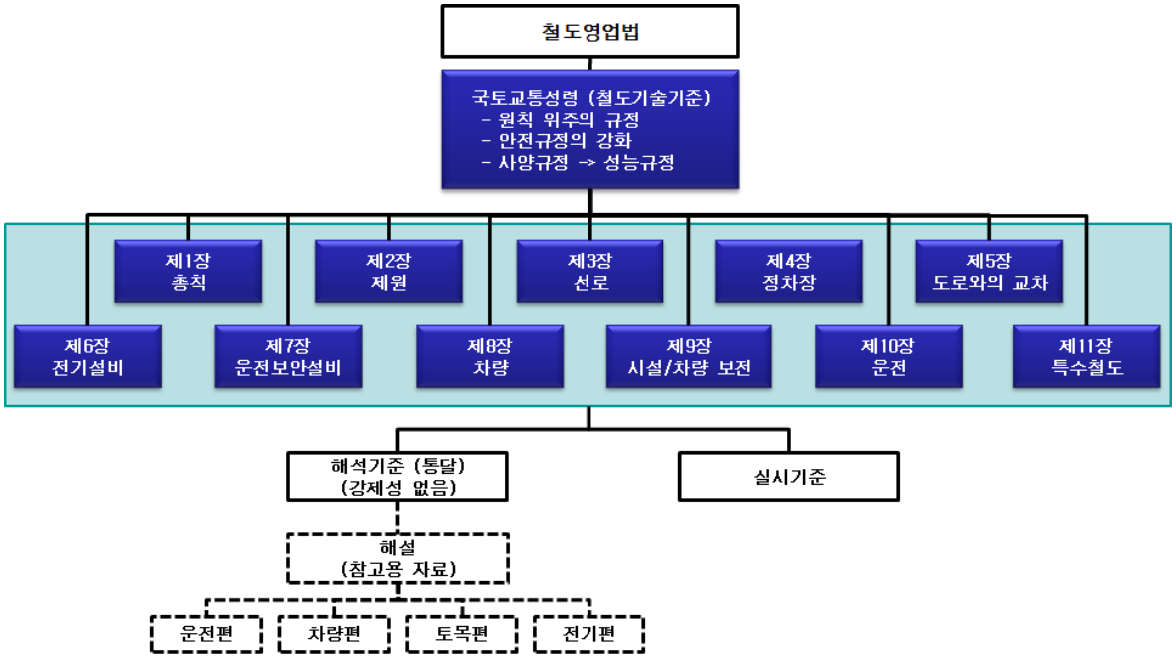


그림 33 일본 철도 기술기준

4. 중국

4.1. 철도 안전 관련 법 체계

4.1.1. 중화인민공화국철도법

중화인민공화국철도법(이하 철도법)은 1991년 최초로 주석령 제32호로 반포되어 중국 철도 운송과 철도건설 및 관리에 대한 기본법의 역할을 하고 있다. 이후 2009년 한차례 개정되어 지금까지 유지되고 있다.

철도법은 철도 운수와 철도건설의 순조로운 운영을 보장하고, 사회주의 현대화 건설과 인민 생활의 수요에 적응하기 위한 목적으로 제정되었다. 또한 철도법의 적용 범위는 중국 철도 전체를 대상으로 국가철도, 지방철도, 전용 철도 또는 철도 전용선을 포함한다.

철도법 하위에는 행정법규가 있고 행정법규 하위에는 법규성 문건이 있으며, 최종적으로는 우리나라의 시행령과 같은 규칙이 있다.

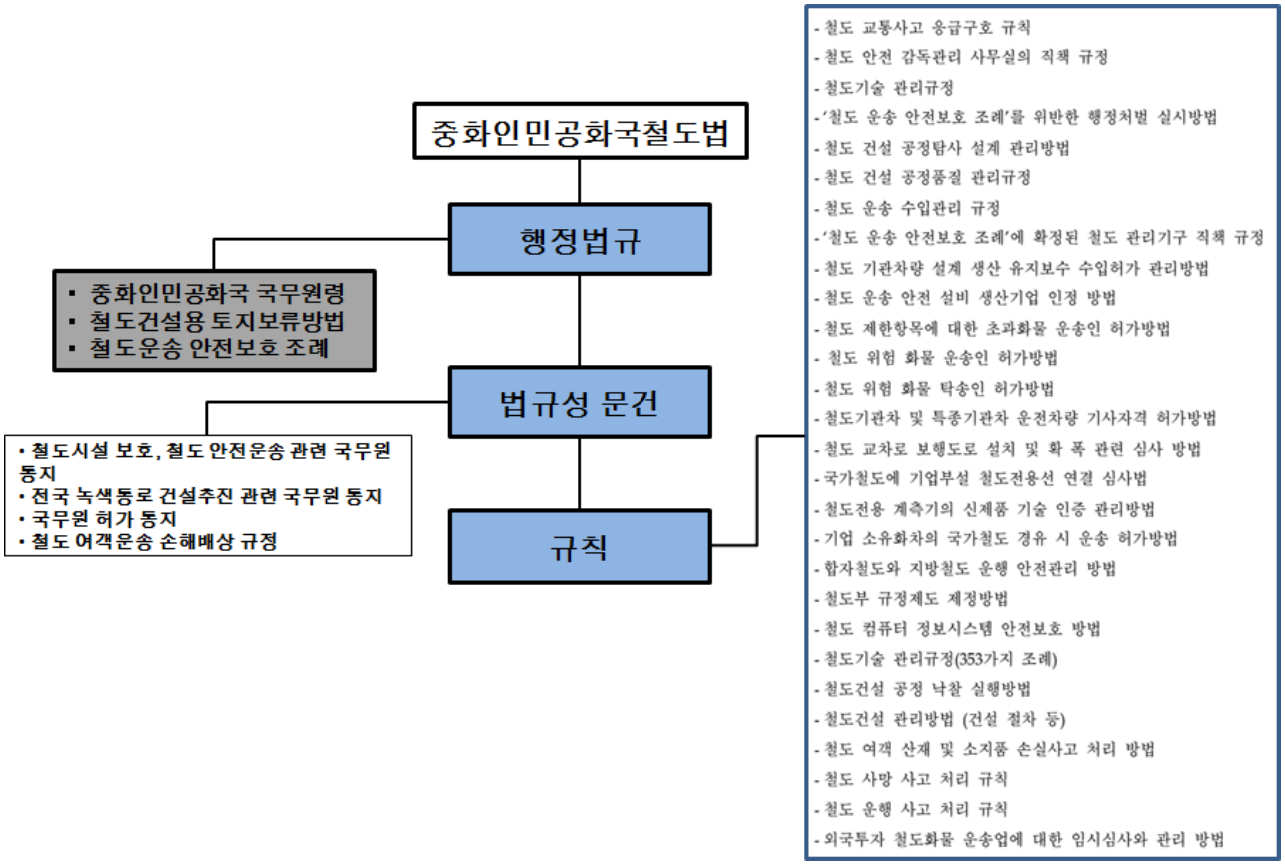


그림 34 중국 철도 안전 관련 법체계

4.2. 철도 관련 조직 및 업무내용

중국의 철도 관련 조직은 국내 조직구성과 유사하며, 수행하는 업무 또한 유사하다. 아래의 표와 같이 대부분의 조직이 국내의 철도 관련 조직과 유사하며, 조직별 수행 업무도 유사하다.

표 27 중국 철도 관련 조직 및 업무내용

구 분	내 용	국내비교
철 도 부	우리나라 국토부와 성격 및 기능이 유사한 최상위 조직으로 중국 정부를 대표하여 중국 철도의 건설, 운영, 기술개발업무를 총괄하고 있는 기관	국토교통부 철도국
공정관리중심	한국철도시설공단과 성격 및 기능이 유사한 철도부 직속 사업단위 산하기관으로 철도프로젝트의 건설 및 관리 업무를 수행하고 있음	한국철도시설공단
노선별 여객전용선 유한책임공사/ 주식유한공사	공정관리센터가 철도부에서 계획하는 전 사업에 대하여 업무를 수행한다면, 노선별 여객전용선 유한책임공사 / 주식유한회사는 해당노선 건설 사업에 대한 실질적인 건설관련 업무를 담당	
지방철도국	한국철도공사와 기능이 유사하며 실질적인 철도운영을 담당하고 있음	한국철도공사
중국철도과학연구원	한국철도기술연구원에 해당되는 국영기업으로, 철도기술에 대한 연구개발 및 기술인력 육성 목적	한국철도기술연구원
설 계 원	중국 철도의 설계를 담당하고 있는 민영기관으로 총 5개의 설계원이 있으며, 중국 전역에 대한 설계를 분할하여 담당하고 있음	국내설계사 유신, KRTC 등

4.3. 인증체계

4.3.1. 국제철도산업표준(IRIS) 인증체계

국제철도산업표준(이하 IRIS)은 철도산업분야에서 세계적으로 인정받는 국제표준으로서, 사업관리 시스템의 평가를 통한 철도산업의 품질 향상과 유지를 그 목표로 하고 있다. IRIS 시스템은 심사 결과에 대한 국제적인 인정과 수용, 복수의 사업관리 시스템에 대한 심사를 배제하여, 아래와 같이 비용 대비 효과를 제고하고 있다.

- IRIS 요건(표준)에 기반을 둔 심사의 품질 제고
- 심사 기록과 결과물을 중앙 데이터베이스에 수집 및 저장

제 4 절 사고사례

1. 국내 사고사례

1.1. 부산역 KTX 열차 충돌사고

1.1.1. 사고개요

부산역 KTX 열차 충돌사고는 2007년 11월 3일 오전 6시 29분경 부산역에서 부산발 서울행 제 110 KTX 고속 열차(이하 110호)를 부산 차량기지에서 출고된 제 H112 KTX 고속열차(이하 H112호)의 정지신호가 현시된 것을 확인하지 못하여 충돌한 사고이다. 6시 30분 부산역에서 출발 예정이었던 110호는 승객 190명을 태운 상태였으며 부산역 구내 9번 선에서 대기 중이었다. H112호는 부산역에서 승객을 태우고 서울역으로 운행하기 위하여 정시보다 18분 빠른 6시 7분에 기지에서 출발, 6시 29분경 부산역에 진입하였다. 정지신호를 확인하지 못하고 진입한 뒤 약 40m 전방, 9번 선에서 출발 대기 중인 110호를 발견하고 비상제동을 시도하였으나 정면으로 충돌하여 KTX 고속 열차 동력차 2량의 앞부분이 파손되었다.

이 사고로 부상자 1명이 발생하였으며, 물적 피해는 약 25만 1,600만 원이 발생하였다. 또한, 110호가 전 구간 운휴되었으며 이후 7개 열차가 24분에서 49분가량 지연되었다. 지연료 반환은 모두 4,863건으로 총 5,961만 2,400원이 발생하였다.

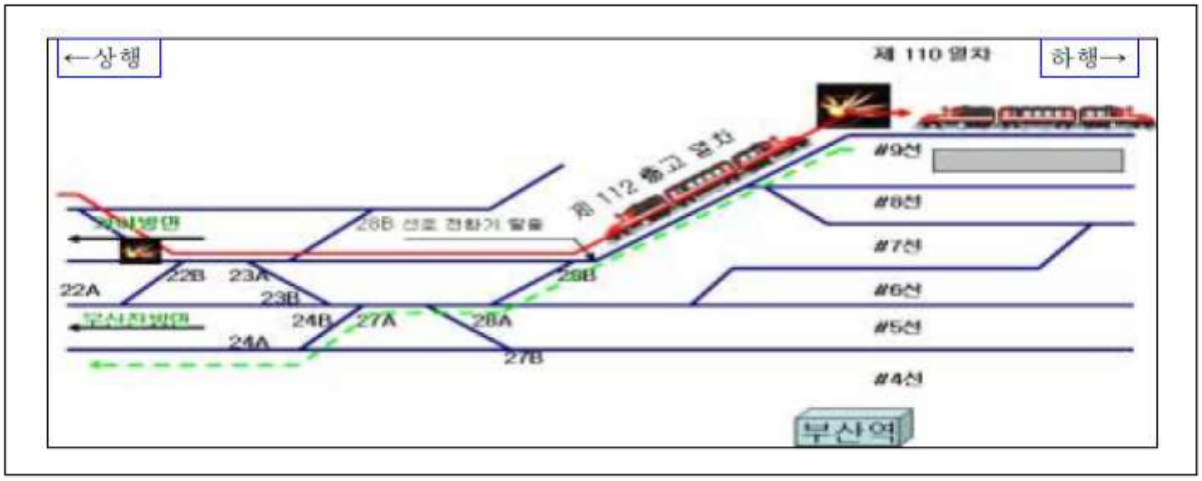


그림 35 부산역 KTX 열차 충돌 사고

1.1.2. 사고원인 분석

부산역 KTX 열차 충돌사고는 복합적인 오류에 근거한다. 먼저 사고 열차 H112호의 기장은 잡념 및 졸음으로 인해 부산역 장내 신호기의 정지신호를 무시하고 진입하였다. 당시 운전자 경계 장치와 ATS가 작동 했으나 무의식적으로 확인 버튼을 눌러 보안 장치의 기능은 무력화된 상태였다.

부산역 장내신호기 전방 3.3m 지점에 설치되어 있는 경고용 ATS 지상자의 경고용 ATS 경보와 장내신호기 정지에 의한 ATS 경보를 착각하여 ATS를 복귀시킨 것으로 판단된다. 이외에도 부산역과 가야역의 운전취급자와 관제사 역시 관제사의 승인 없이 조기 출발을 지시한 것, 사고 열차에 대해 기외정차 통보를 하지 않은 것 등 운전취급규정과 철도 운행에 관한 안전지침을 위반한 것으로 드러났다.

1.2. 상왕십리역 추돌사고

1.2.1. 사고개요

서울메트로 2호선 내선에 운행하던 제2258호 선행 전동열차가 상왕십리역에서 승강장 안전문(PSD[9])을 여닫느라 출발이 지체되고 있는 상황에서 제2260호 후속 전동열차 기관사는 신당역을 출발하여 상왕십리역 방향 ③번과 ②번 폐색신호기의 진행신호(녹색, Green) 현시에 따라 진행하면서 우곡선(500R)을 돌아 ①번 상왕십리역 장내신호기가 정지신호(적색, Red) 로 현시된 것을 확인하고 즉시 제동장치(상용 + 비상 + 보안)를 체결하였으나 68㎞/h속도에서 제동거리(2) 부족으로 선행열차 후부에 충돌 · 탈선된 사고이며, 복구하는 동안 약 8시간 48분간 내선방향의 열차운행이 중단되었다.

그림 36 사고 당시 신호 상황 ③폐색신호(G) ②폐색신호(G) ①장내신호(R)

이 사고로 인한 인명피해가 중상 22명과 경상 68명, 경미한 피해가 387명으로 인명피해가 총 477명으로 집계되었고, 물적 피해는 전동차 차체의 굴곡 등으로 약 28억 2천 6백만 원 정도로 집계되었다.

9) 승강장 안전문 (PSD: Platform Safety Door)

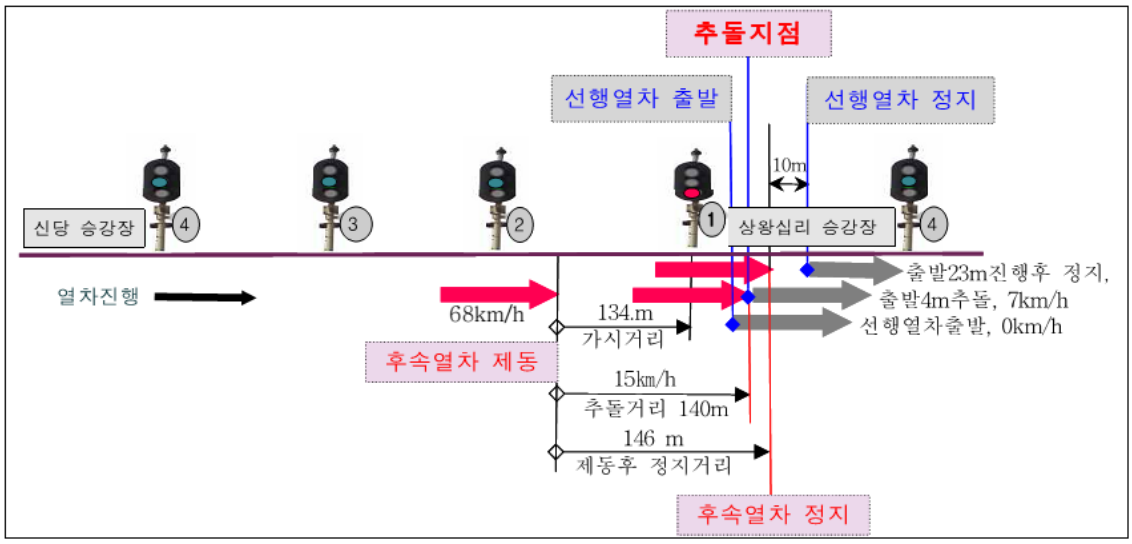


그림 37 선행열차(2258)와 후속열차(2260)의 충돌사고 상황

1.2.2. 사고원인 분석

상왕십리역 승강장에 선행열차가 정차하고 있는데도 불구하고 ②번(435) 폐색 신호기가 정지신호(적색)로 현시되지 아니하고 진행신호(녹색)로 잘못 현시되어 후속열차가 상왕십리역에 접근하면서 ①번(437)신호기의 정지신호(적색) 현시에 따라 기관사가 비상제동을 작동하였으나 제동거리가 충분하지 못하여 선행열차를 추돌하였다.

구 분	Count	Data			
내 용	데이터 개수	① 437신호기	② 437궤도	③ 435궤도	④ 436궤도
정상정보	**0 4** (0000 0100)	(R현시) **0 0** (0000 0000)	(점유) **0 0** (0000 0000)	(비점유) **0 1** (0000 0001)	(비점유) **0 1** (0000 0001)
사 고 시	**0 4** (0000 0100)	**0 4** (0000 0100) "Y현시"	**1 E** (0001 1110) "미정의"	**0 6** (0000 0110) "미정의"	**0 7** (0000 0111) "미정의"
신호제어 처리결과	신호제어	Y현시로 인식	비점유로 처리	비점유로 처리	비점유 처리 436신호기 G현시 출력
		435신호기, 433신호기 진행현시 출력			
	화면표시	표시안함 (Y신호로 인식)	점유(적색)	점유(적색)	비점유(백색)

그림 38 사고당시 왕십리에서 을지로입구로 송신한 통신데이터

사고 직후 현장의 통신장애가 복구되지 않은 상태에서 신당역에서 프로토콜 분석기로 수집된 데이터를 분석한 결과, 왕십리 현장제어장치는 을지로입구역으로 5개의 데이터 모두를 [그림 38과 같이 프로토콜에 정의되지 않은 임의의 데이터 또는 잘못 된 데이터를 전송하였고 이를 수신한 을지로입구역 연동장치는 수신한 데이터를 그대로 신당역 단말제어장치로 전송하였다.

2번째부터 4번째 데이터는 궤도회로 정보를 표현하는 것으로 모두 프로토콜에 정의되지 않은 데이터를 수신하였으나 연동장치는 8개의 비트가 모두 “0”(16진수 “00”)일 때에만 열차점유로 처리하도록 프로그래밍 되어있어 현장상태와 관계없이 437T, 435T, 436T는 모두 비점유로 처리하였다.

이에 따라 사고당시 437T궤도회로에는 선행열차가 정차 중이었으나 해당구간에 열차가 없는 것으로 인식하여 433신호기, 435신호기는 모두 진행신호(녹색, Green)를 현시하도록 출력되었다.

신호기를 제어하는 장치는 고장 시 안전 측 동작원칙(Fail Safe)에 의하여 설계·제작 되어야 하나 을지로입구역 연동 제어 장치와 신당역 단말제어장치의 소프트웨어는 열차점유정보 8개 비트 모두 “0”일 경우에만 열차가 점유하는 것으로 처리하고, 그 외 255개에 해당하는 경우는 모두 열차가 없는 것으로 처리토록 한 소프트웨어의 결함으로 분석되었다.[10)]

10) 항공·철도사고조사위원회, 서울매트로 2호선 상왕십리역 전동열차 충돌탈선 조사보고서, 2014. 9. 30 발행

2. 해외 사고사례

2.1. 중국 원저우 고속철 추돌 탈선사고

2.1.1. 사고개요

중국 원저우 고속철 추돌 탈선사고는 2011년 7월 23일 현지 시각 20시 34분경 중국 남동부에 위치한 저장성(浙江省) 원저우(溫州)에서 고속 열차 '둥처(動車)'가 추돌하여 탈선한 사고를 말한다.

사고의 경위는 다음과 같다. 상해 철도국이 관리하는 닝버(寧波)-원저우(溫州) 노선을 운행하던 D3115호는 벼락을 맞고 동력을 상실하여 정지하였다. 또한, 벼락으로 인해 해당 구간의 신호시스템에 이상이 발생하여 뒤따라오던 D301호가 추돌하였다. 이로 인해 객차 8량이 탈선하고 4량이 20m 높이의 교량 밑으로 추락하였다

이 사고로 여객 37명, 승무원 3명을 포함한 총 40명의 사망자가 발생하였으며 부상자는 총 172명으로 집계되었다. 물적 피해로는 7대의 차량이 폐기되고 22대 의 차량의 파손되었으며 직접적 경제 손실로 총 1억 9371.65만 위엔(약 360억 원)이 발생하였다. 또한, 전 구간이 32시간 35분 운휴되는 간접적인 피해를 초래했다(王群, 2013)

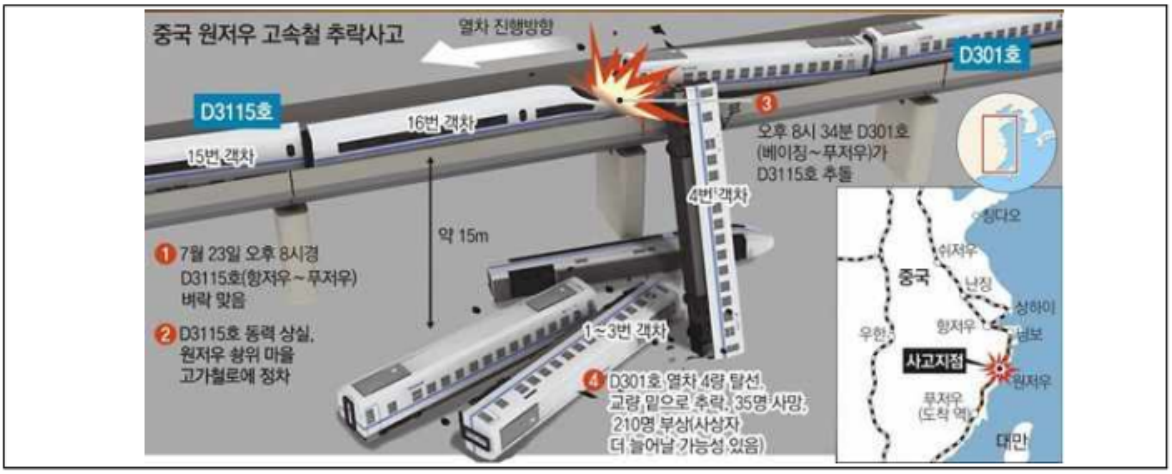


그림 39 중국 윈저우 고속철 추돌 탈선사고

2.1.2. 사고원인 분석

원저우 고속철 추돌 탈선사고의 주요 사고원인으로 지목된 것은 다음과 같다.

첫째, 기상 악화로 인해 열차 제어 시스템에 오류가 발생하였다. 고속철 D3115호는 낙뢰로 인한 동력 상실로 20m 높이의 고가 교량 위에 정차하였다. 마찬가지로 낙뢰로 인해 오류가 발생한 신호기는 후속 열차인 D301호에 대해 정지 신호 대신 진입을 유도하였다. D301호 기관사가 육안으로 정지한 선행 열차를 확인한 뒤 급정지를 시도하였

으나 사고를 막지 못하였다. 사고 발생 당일 2011년 7월 23일 19시 27분부터 19시 34분 사이의 기상 상황은 낙뢰 횟수가 340회를 넘었으며 이 중 전류 100kA가 넘는 번개는 총 11회로, 낙뢰에 의한 사고 발생이 충분히 의심될 상황이었다. 이에 대한 안전장치나 매뉴얼이 부재한 상황에서 운행을 지속했다는 점은 문제가 될 수 있다. 또한, 낙뢰 1회 로 열차 제어 시스템에 오류가 발생했다는 것으로 미루어볼 때 시스템 설계 자체에 심각한 결함이 있었다고 보인다.

둘째, 철도부의 관리가 부실했다는 문제점이 있었다. 사고 조사 과정에서 철도부 전임 책임자의 설비 입찰과 기술심사가 규정에 어긋나게 집행되었다는 점과 사고 당시 철도부 요원들의 과실이 지적된 것이다.

2.1.3. 사고결과

사고 발생 직후 중국 정부의 후진타오 국가 주석과 원자바오 총리는 인명 구조와 피해 복구를 철저히 시행할 것을 지시하였다. 그러나 중국 철도부는 사고 원인 규명 시 결함 및 과실이 드러날 것을 우려하여 열차 잔해를 제대로 조사하지 않고 땅에 묻어버렸다.

또한 중국 정부는 중국철도부 상해철도국의 고위 관리 3명에게 사고의 책임을 묻고 물러나게 하였다. 그러나 사고 이전 철도장관이 철도 건설 관련 수뢰혐의로 구속된 사건 및 중국철도부 차관보 역시 해외 계좌로 횡령하여 체포된 사례 등을 볼 때 고질적인 부패 문제가 있음에도 이에 대한 규명이 명확하게 이루어지지는 못했다.

당시 중국 정부는 중국 고속철도의 해외 진출에 대해 적극적인 입장을 취하고 있었기 때문에 해당 사고 사항이 영향을 줄 것을 우려하였다. 결과적으로 중국철도의 사고조사위원회는 사고조사에 대한 발표 내용을 미루었으며, 중국 정부에 의해 각 미디어의 사고조사 및 발표에 대한 언론 통제가 이루어졌다.

원저우 고속철 추돌 탈선사고 이후 중국 정부는 모든 고속철도의 속도를 50km/h 감속하도록 했다. 최고 시속 350km로 설계된 것은 시속 300km로, 시속 250km와 시속 200km로 설계된 것은 각각 시속 200km와 시속 160km로 조정하였다. 또한, 중국 정부는 감속으로 인해 운행시간이 늘어나는 것을 감안하여 운임료 5%를 인하하였다. 이는 고속철도 안전에 대한 불신으로 수요가 낮아지는 것을 방지하기 위한 대책임을 알 수 있다.

대외 적으로 중국정부의 적극적인 해명에도 불구하고 국제적으로 중국 고속철도 기술에 대한 우려가 확대되었다. 이로 인해 중국정부가 공을 많이 들였던 사우디아라비아의 고속철도 사업에서 중국 고속철도가 배제되었다.

2.2. 스페인 갈라시아 고속열차 탈선사고

2.2.1. 사고개요

스페인 갈리시아 고속열차 탈선사고는 2013년 7월 24일 현지 시각으로 20시 41분경 스페인 북서부 갈리시아 지역의 산티아고 데 컴포스텔라(Compostela) 역 3.2km 지점에서 수도 마드리드 발 페롤 행 여객열차가 곡선 구간에서 13량 전 편성이 탈선하여 전복된 사고이다.

사고 당시 열차는 고속선 구간을 지나 재래선 구간으로 진입하고 있었으나, 급곡선 선호 제한속도 80km/h의 약 두 배에 해당하는 속도인 192km/h로 운행하던 중 정상적으로 제동하지 못하고 선로 외측으로 탈선하였다. 이 사고로 탑승객 218명 중 79명이 사망하고 130여 명이 부상을 입었다.

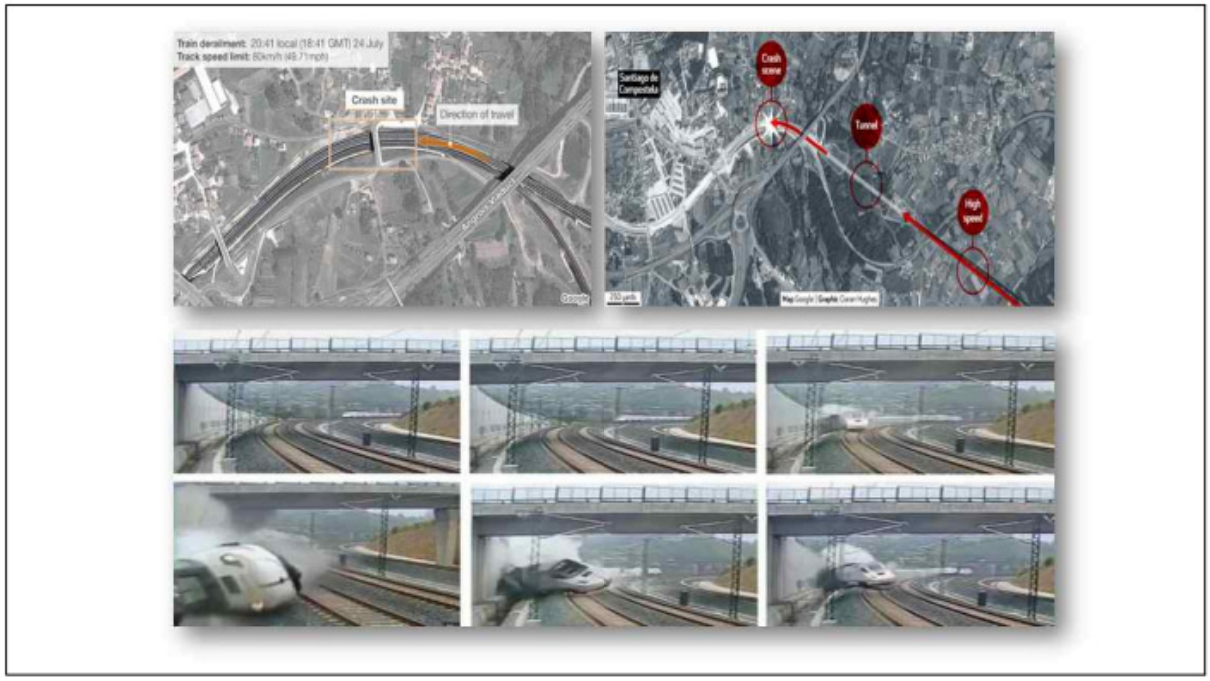


그림 40 스페인 갈라시아 고속열차 탈선사고

○ 사고원인 분석

사고의 원인이 되었던 과속에 대한 책임은 기관사에게 있었던 것으로 밝혀졌다. 고속구간을 192km/h로 운행하던 기관사는 회사 무전을 받던 중 곡선 진입 수초 전에 제동하였다.

기관사의 제어로 정상적인 제동이 가능하려면 탈선 지점으로부터 약 4km 전방에서 제동을 취급했어야 했다는 분석이다. [표 28]을 살펴보면 통화 종료 후 기관사가 비상 제동하여 어느 정도 속도를 경감하였으나 사고 발생을 막기에는 턱없이 부족한 153km/h로 곡선 구간에 진입하면서 탈선한 것으로 보인다.

그러나 일부 전문가들은 사고 구간이 유럽형 열차제어시스템인 ETCS에서 스페인이 자체적으로 운용하는 신호 및 자동제어시스템인 ASFA 시스템으로 교체되는 구간이었음을 고려할 때, 제어시스템 상에 문제가 발생했을 수도 있다고 지적하였다. ETCS 신호시스템 구간은 자동으로 속도제어가 가능하지만 ASFA 신호시스템은 자동 속도제한 없이 해당 구간에서 초과 속도로 운행하면 경보로 주의를 주는 시스템이기 때문에, ETCS 구간에서 ASFA 구간으로 교체되는 사고 구간은 다른 곳보다 주의해야 하는 구간이라고 할 수 있다.

표 28 사고 당시 상황

구 분	전화통화	1, 2차 속도경고	통화종료	3차 속도경보	비상제동	사고발생
시 간	약 2분전		11초전	6초전	2초전	20:41
거 리	약 6Km 전방					
속 도	192km/h				179km/h	153km/h
신호시스템	ETCS-1	ASFA				

2.2.2. 사고결과

사고 발생 직후 스페인 총리는 긴급 관계 장관회의를 소집하였으며, 다음날 신속하게 사고 장소에 방문하여 희생자 가족들을 위로하였다. 정부 최고위층에서 희생자 가족을 위로하고 사고 조사에 집중하는 좋은 선례를 남겼다고 볼 수 있다.

또한, 정부의 의지로 빠른 시간 내 사고결과가 발표될 수 있도록 노력하여 스페인 철도사고조사위원회인 CIAF는 사고 발생 다음 해인 2014년 8월 2일에 약 266쪽 분량의 최종 사고 조사 보고서를 발표할 수 있었다. 이후 스페인 정부는 전 철도 노선에 대한 안전점검을 시행하고, 열차운행 중 무선전화를 금지하였다. 또한, 사고의 또 다른 원인으로 지목되었던 고속선의 ERTMS 시스템과 일반 노선에서의 신호시스템 변경 시의 오류를 막기 위해 Fail-Safe 안전시스템을 마련하였다.

제 5 절 국내 철도 소프트웨어 현황 조사 결론

1. 철도 현황 조사 정리

본 현황 조사는 철도 소프트웨어 신뢰·안전성 확보를 위한 가이드 개발을 위해 국내·외 철도 시장 현황을 조사하고, 우리나라 철도 산업 시장의 구조, 법체계, 인증 체계를 비교하기 위해 해외 주요 국가들의 철도 산업 체계 및 법체계 조사 분석 하였다.

철도 소프트웨어 신뢰·안전성 확보를 위한 가이드의 필요성을 확인하기 위해 공식적으로 소프트웨어 오류로 인해 발생한 사고사례를 찾았지만 대부분의 경우가 기기 이상이나 운영자의 조작 미숙으로 인한 사고로 분류되어 있고 자세한 원인이 기록되어 있지 않아서 운영자 조작 미숙으로 인한 사고 또는 시설물 오류로 인한 사고 중 소프트웨어와 연관이 있을 것이라고 생각되는 사고들을 모아서 정리하였다.

국내외 사고사례 보고서의 경우 대부분 사고의 원인과 책임 소재를 명확하게 규명하고 있지 않으며, 공식 발표와 다른 원인 분석에 무게가 실리는 경우가 자주 확인되었다. 이는 다양한 이유가 있겠지만 소프트웨어 오류로 인한 사고의 경우 명확하게 원인을 규명하기가 어렵고 설사 원인을 확인하였다고 해도 재현하기가 어렵기 때문에 소프트웨어 오류로 인한 사고라고 단정하는 것이 쉽지 않다.

최근 고속철도의 비중이 높아지는 현 철도 산업에서 소프트웨어의 오류로 인해 발생하는 사고는 대규모 인명 피해를 발생시키지만 개발 단계에서부터 철저한 준비와 검증이 이루어지지 않으면 이를 예방하기가 더욱 어려워지고 있다.

유럽, 중국, 미국 등 철도 강국인 주요국들은 자국민의 안전과 철도 산업의 경쟁력 확보를 위해 강력한 규제와 인증 제도를 시행하고 있다. 이들 규제와 인증 제도는 안전성 확보를 최우선으로 하고 있으며, 이를 확보하기 위해 IEC 62278과 IEC 62279 표준을 준수하도록 하고 있다. 이들 표준의 핵심은 시스템 개발 초기부터 안전성 분석을 수행하고 소프트웨어 개발 단계에서는 철저하게 소프트웨어 공학 프로세스를 준수하여 안전한 결과물을 만들어내게 하는 것이다.

포화 상태인 국내 철도 시장에서 살아남기 위해 해외 시장을 바라보고 있는 국내의 철도 산업 관련 업체들의 대부분은 영세하기 때문에 강화되고 있는 해외 철도 시장의 규제를 따르기 위해 표준을 지키며 제품을 생산하는 것이 쉽지 않다.

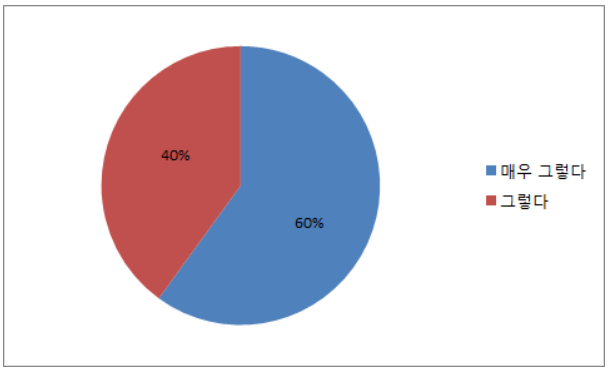
2. 현장 인터뷰 및 설문 조사

철도 안전가이드 개발을 위해서는 가이드 활용의 대상이 되는 철도 산업 현업 개발 담당자들로부터 개발 실태와 가이드 개발 시 필요한 요구사항을 파악해야 했다. 이를 위해 본 과제에서는 설문 항목을 작성하여 현업 담당자들로부터 답변을 받았고 필요한 경우 현장 인터뷰를 통해 설문 조사 내용을 보강하여 조사하였다. 인터뷰 및 설문 결과 중 수치화가 용이한 객관식 항목들에 대해서 설문 결과와 분석을 정리 하였다.

2.1. 소프트웨어 공학 및 표준의 이해

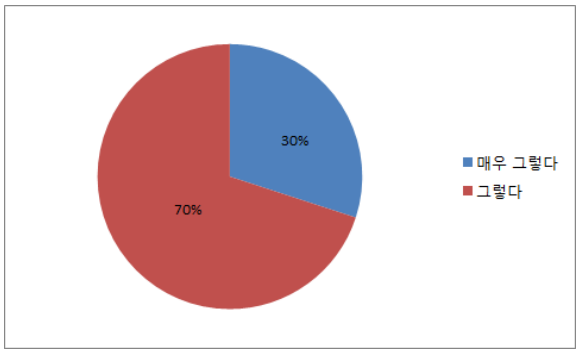
1. 소프트웨어 개발업무에 있어서 체계적인 소프트웨어공학 기법의 적용이 중요하다고 생각하십니까? ① 매우 그렇다 ② 그렇다 ③ 보통 ④ 아니다 ⑤ 매우 아니다

○ 대부분의 경우 소프트웨어 개발업무에 있어서 체계적인 소프트웨어 공학 기법의 적용이 중요하다고 생각하고 있다.



2. 체계적인 업무 수행을 위해서라면 업무가 다소 가중되어도 소프트웨어 공학기법을 적용 할 의향이 있으십니까? ① 매우 그렇다 ② 그렇다 ③ 보통 ④ 아니다 ⑤ 매우 아니다

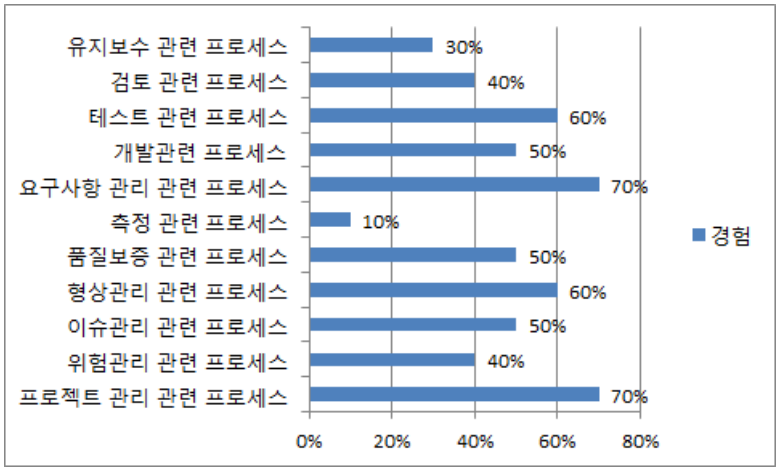
○ 대부분의 경우 체계적인 소프트웨어 개발 업무를 위해서는 업무가 가중되더라도 적용해야 한다고 생각하고 있다.



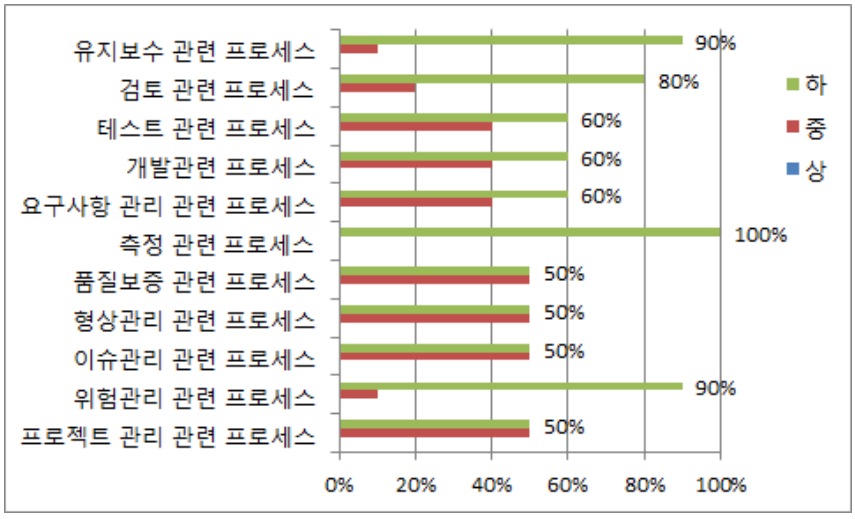
3. 다음 프로세스의 적용 경험과 본인의 이해 수준에 대해 모두 체크해 주시기 바랍니다.

프로세스 항목	적용경험	이해수준
프로젝트 관리 관련 프로세스	유 / 무	상 / 중 / 하
위험관리 관련 프로세스	유 / 무	상 / 중 / 하
이슈관리 관련 프로세스	유 / 무	상 / 중 / 하
형상관리 관련 프로세스	유 / 무	상 / 중 / 하
품질보증 관련 프로세스	유 / 무	상 / 중 / 하
측정 관련 프로세스	유 / 무	상 / 중 / 하
요구사항 관리 관련 프로세스	유 / 무	상 / 중 / 하
개발관련 프로세스(분석, 설계, 구현)	유 / 무	상 / 중 / 하
테스트 관련 프로세스	유 / 무	상 / 중 / 하
검토(Review) 관련 프로세스	유 / 무	상 / 중 / 하
유지보수 관련 프로세스	유 / 무	상 / 중 / 하

○ 소프트웨어 공학의 다양한 프로세스 중 프로젝트 관리, 요구사항 관리, 테스트 등 일반적으로 널리 알려진 프로세스의 경우 경험자가 많은 반면 측정이나 유지보수 관련 프로세스에 대해서는 대부분의 경우 경험도가 낮게 나타나고 있다.



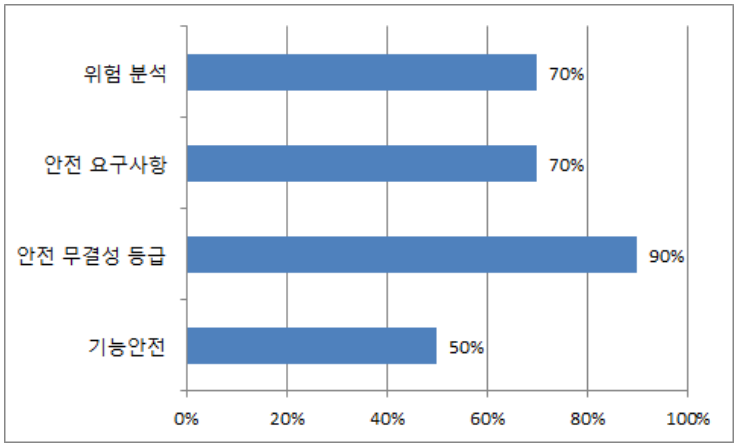
○ 대부분의 소프트웨어 공학 프로세스에 대한 이해수준은 낮은 것으로 나타나고 있다.



2.2. 안전성 분석

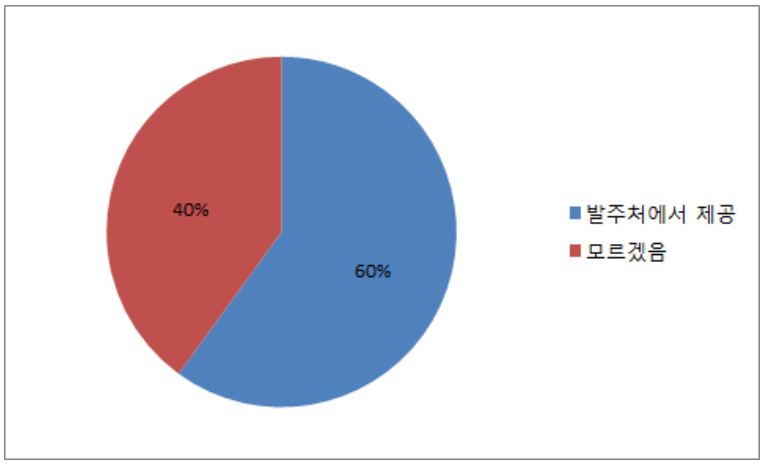
6. 현업에서 사용하시거나 알고 계신 용어를 모두 선택해 주십시오.
   ① 기능 안전 (Functional Safety)
   ② 안전 무결성 등급 (SIL, Safety Integrity Level)
   ③ 안전 요구사항 (Safety Requirement)
   ④ 위험 분석 (Risk Analysis)

○ 안전성 분석과 관련해서 용어에 대한 이해도는 높은 편이다.



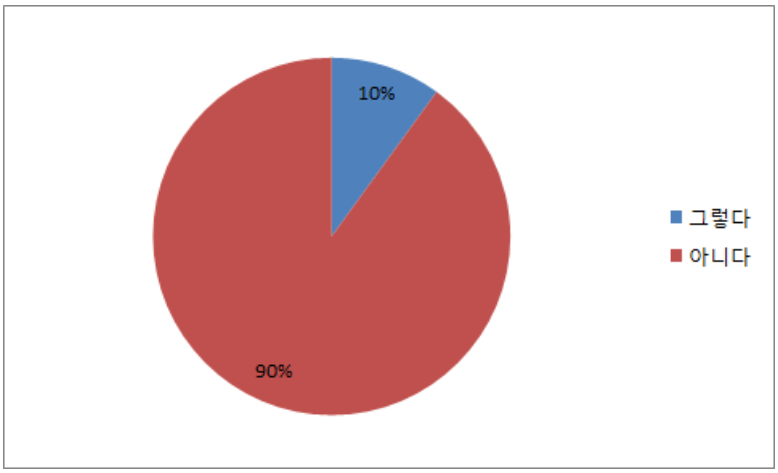
8. 제품 개발 시 안전 무결성 등급(SIL)은 어떻게 산정하십니까?
   ① 발주처에서 제공
   ② 기존 관행대로 선정
   ③ 상황에 맞게 계산 및 선정
   ④ 모르겠음

○ 제품 수주 시 안전 무결성 등급은 대부분 발주처로부터 받거나(60%) 관련한 사항을 개발 시 반영하지 않고 있다.(40%)



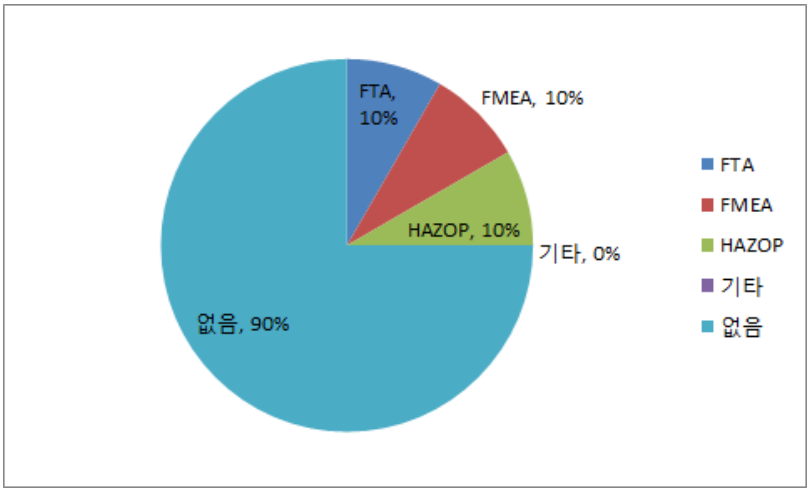
9. 제품 개발 시 사전에 위험 분석 프로세스를 수행 하십니까?
① 그렇다
② 아니다

○ 제품 수주 후 개발 시 대부분의 경우 위험 분석을 수행하지 않고 개발을 진행한다. (90%)



12. 위험 분석을 위해 사용하고 있는 기법을 모두 선택해 주시기 바랍니다.
① FTA (Fault Tree Analysis)
② FMEA (Failure Mode and Effects Analysis)
③ HAZOP (HAZard and OPerability study)
④ 기타 (
⑤ 없음

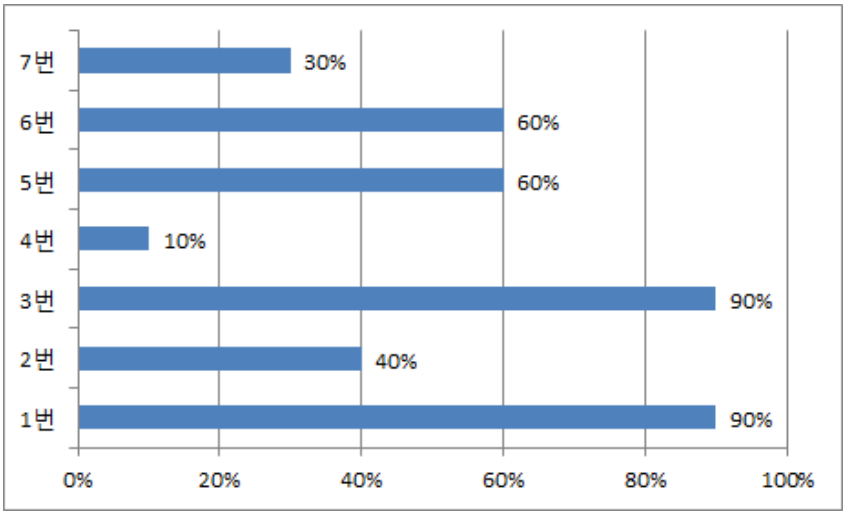
○ 대부분의 경우 위험 분석을 수행하지 않으며(90%), 수행하는 경우 일반적인 FTA, FMEA, HAZOP 분석 기법을 사용한다.



2.3. 소프트웨어 개발 방법

15. 다음 중 요구사항 정의 시 정의하는 정보를 모두 선택해 주시기 바랍니다.
    ① 고객이 제시한 명시적인 요구사항 정리
    ② 과거경험, 벤치마킹, 과거 문제점 등을 고려하여 요구사항 정의
    ③ 구현하여야 하는 모든 기능 리스트
    ④ 비-기능 요구사항
    ⑤ 제약 사항
    ⑥ 인터페이스 대상
    ⑦ 위험 분석 결과 도출 된 안전 기능
    ⑧ 해당하는 사항 없음

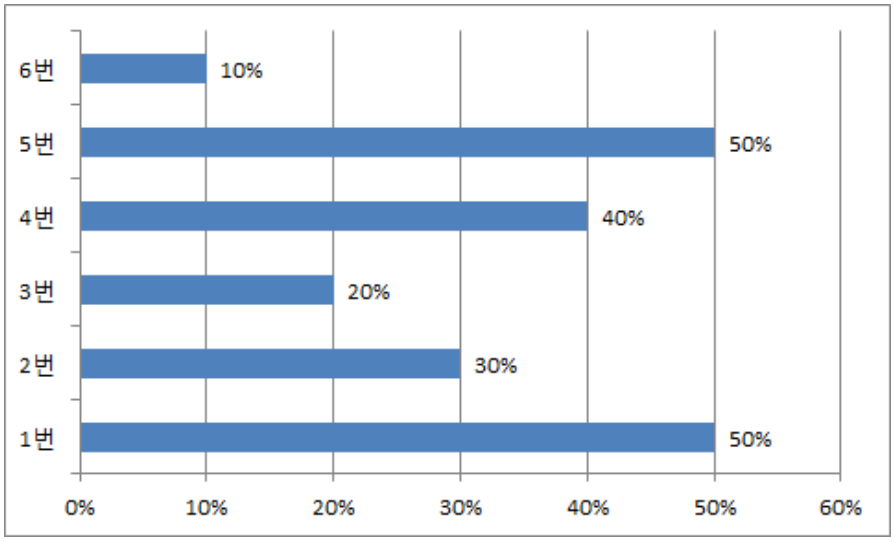
○ 대부분 비-기능 요구사항(10%)과 안전 기능 요구사항(30%)을 식별하지 않으며, 고객이 제시한 요구사항을 기능 요구사항으로 반영하고 있다.



17. 소프트웨어 설계를 위해 수행하는 활동을 모두 선택해 주시기 바랍니다.

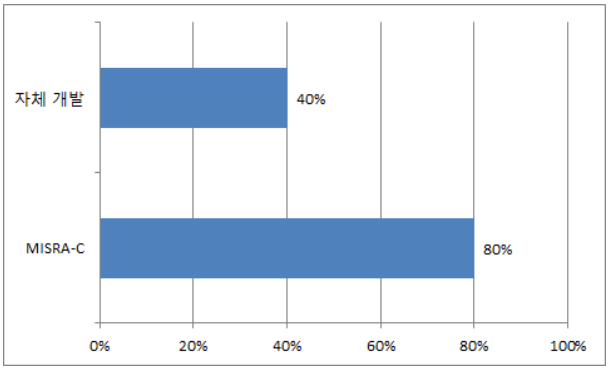
① 자체 개발 부분, 솔루션 도입 부분, 기존 시스템 재사용 영역 식별
② 아키텍처 설계
③ 상위 설계 및 데이터 설계 활동을 수행하고 문서화 함
④ 상세 설계 및 데이터 설계 활동을 수행하고 문서화 함
⑤ 내부 시스템 간 또는 외부 시스템과의 인터페이스에 대한 설계를 수행하고, 문서화
⑥ 위의 활동들을 수행하지 않음

○ 소프트웨어 설계 시 기존 시스템의 기능을 재사용하는 빈도가 높으며, 외부 시스템과의 인터페이스에 대한 식별과 반영을 중요하게 생각하고 있다.



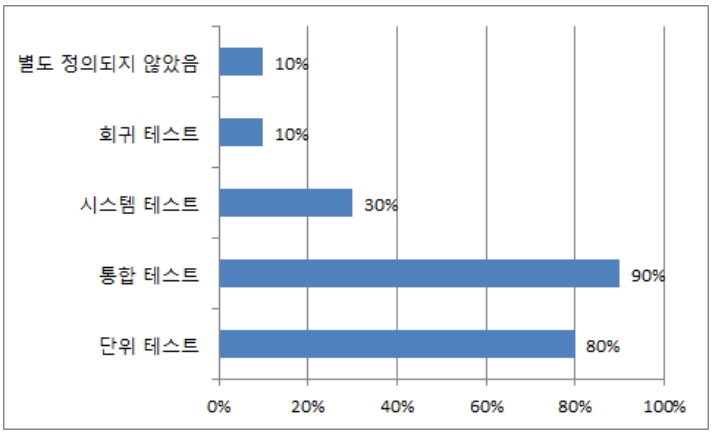
20. 소프트웨어 개발 시 사용하는 코딩 가이드라인을 선택해 주시기 바랍니다.
① MISRA-C
② 자체 개발
③ 컨설팅 업체 제안
④ 사용하지 않음

○ 대부분의 업체에서 코딩 가이드라인으로 MISRA-C를 적용하고 있으며, 자체 개발한 코딩 가이드라인과 MISRA-C를 조합해서 사용하는 경우가 있다.



21. 제품 개발 시 수행하는 테스트를 모두 선택해 주시기 바랍니다.
① 단위 테스트
② 통합 테스트
③ 시스템 테스트
④ 회귀 테스트
⑤ 별도 정의되지 않았음

○ 테스트 단계에서 대부분 단위 테스트와 통합 테스트를 수행한다. 하지만 인터뷰를 통해 확인해 본 결과 단위 테스트는 구현 된 기능이 동작하는지 개발자만이 사용하는 테스트였고, 통합 테스트는 분산 개발 된 기능들이 제대로 통합 되는지 통합 후 기능 시연을 하는 단계라는 것을 확인했다. (테스트 프레임워크나 프로세스를 정식으로 사용하는 경우는 거의 없었다.)



3. 현황 조사 결론

“소프트웨어 중심사회”가 도래하면서 소프트웨어가 국민 안전 확보를 위한 핵심요소로 부각되고 있다. 철도 산업 분야의 경우 철도 관련 시스템의 규모 및 복잡성의 증가로 인해 소프트웨어의 안전성이 더욱 중요한 요소로 부각되고 있다.

해외 철도 산업에서도 소프트웨어의 중요성이 날로 증가하고 있으며 자국민의 안전과 철도 산업의 경쟁력 강화를 위해 법체계 확립하고 제품 개발 시 국제 안전 표준을 준수하도록 강제하고 있다. 포화 상태인 국내 철도 시장에서 살아남기 위해 해외 시장을 바라보고 있는 국내의 철도 산업 관련 업체들의 대부분은 영세하기 때문에 강화되고 있는 해외 철도 시장의 규제를 따르기 위해 표준을 지키며 제품을 생산해야 한다.

국제 안전 표준은 안전활동 수행 시 준수해야 할 활동에 대한 요건과 요건을 만족하기 위한 기법만을 제시하고 있다. 표준을 준수하여 산출물을 작성하기 위해서는 소프트웨어 공학 기법을 요구하고 있어 이 분야에 대한 지식과 경험이 없을 경우 실무에서 표준을 이해하고 적용하는데 많은 어려움이 존재한다. 또한 실제로 설문 조사를 분석한 결과, 국내 철도 산업 업체들의 소프트웨어 공학의 이해도나 관련된 기술 수준이 낮은 것으로 파악되었다.

따라서 이와 같은 문제점을 해소하기 위해 실무차원에서 표준을 쉽게 이해하고 목표 수준을 달성 할 수 있도록 구체적인 수행 지침과 쉽게 참조하고 따라할 수 있는 예제 및 적용 순서가 포함된 가이드 개발이 필요하다.

제 3 장 철도 시스템 안전성 분석 가이드

제 1 절 IEC 62278 표준 개요

1. 목적 및 적용범위

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 절차에 대해 다룬다. 또한 표준은 철도 차량 뿐만 아니라 신호시스템, 전력설비 등 철도 시스템 전반을 대상으로 다음과 같은 기본적인 개념을 다룬다[7].

- 생명주기 각 단계에서 RAMS 관리를 위한 절차와 실시항목을 규정한다.
- RAMS 요구사항 및 RAMS 요구사항을 충족하는지 확인하는 절차를 규정한다. 이를 위해 필요한 분석 및 작성해야 할 산출물을 규정한다.
- 철도 시스템의 안전에 미치는 영향에 대하여 안전성 분석을 규정한다. 하지만 안전성 분석을 수행할 것을 요구하고 있으나 명확한 수치에 대해서는 규정하지 않는다.
- RAMS 관리를 위한 일련의 과정에 있어서 관련 조직과 관계자의 역할 및 자격 등에 대해서는 유럽의 체제를 전제로 한다[1].

따라서 철도용 전기 설비의 개발 시 표준에서 제시한 RAMS 요구사항을 충족하는 표준 생명주기 프로세스의 가이드라인을 준수한다. 본 시스템 안전성 분석 가이드의 범위는 생명주기에서 시스템 수용 단계까지이며 시스템 개량 및 폐기단계는 범위에서 제외된다. “RAMS” 라는 단어는 아래와 같이 4개의 알파벳으로 시작하는 축약어를 의미하며 각각의 의미는 다음과 같다.

- 신뢰성(Reliability; R) : 운용 환경에서 요구 품질을 유지하며 얼마나 오래 사용할 수 있을까?
- 가용성(Availability; A)) : 운용해야 할 시점에 정상적인 기능을 유지하고 있나?
- 유지보수성(Maintainability; M) :고장이 났을 때 수리해서 다시 운용하는데 얼마나 걸리나?
- 안전성(Safety; S) : 발생 가능한 위험을 방지하기 위한 관리는 잘 되어 있는가?

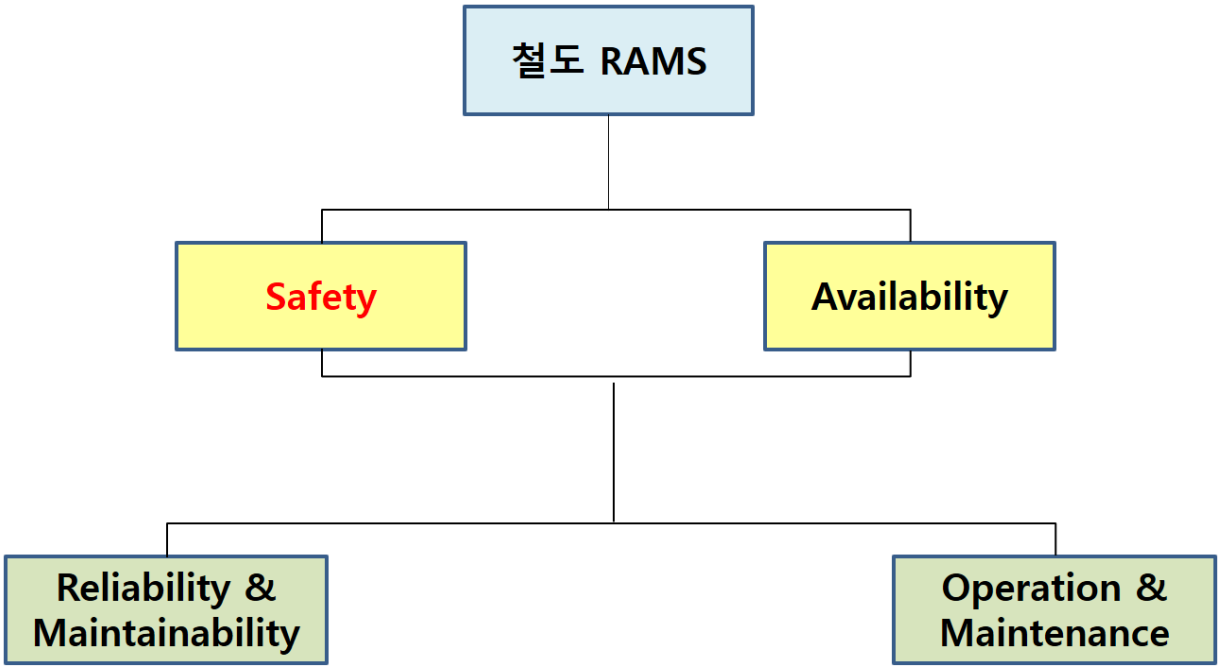


그림 41 철도 안전과 RAMS [50]

표준에서는 철도 RAMS은 [그림 41]와 같이 안전성(Safety)과 가용성(Availability) 그리고 유지보수성(Maintainability) 영역으로 나뉘게 된다. 철도 RAMS의 목표는 철도 안전성, 가용성 확보 및 향상이다. 안전성과 가용성의 확보 및 향상은 신뢰성, 유지보수성, 운용 및 유지보수 활동에 의해 좌우됨을 알 수 있다.

2. 철도 시스템의 안전성과 RAM 분석 및 관리

철도 시스템은 제안 요청서(Request for proposal; RFP) 또는 시스템 설계 시 다른 성능 목표와 함께 RAMS 목표를 구체적으로 정의하고 서브시스템에 할당해야 한다. RAMS 관리는 철도 운용 시 발생 가능한 시스템 장애, 인명사고 및 물적 손실을 초래할 수 있는 컴포넌트 고장 및 위험원들을 식별한다. 이후 이를 적절한 수준으로 관리하기 위하여 철도 시스템의 계획부터 설계, 제작, 시험 및 운용단계에 이르기 까지 전 생명주기에 걸쳐 적용되는 프로세스 및 단계별 활동을 의미한다. 즉, RAMS 관리는 RAMS 활동을 통해 안전성 분석 영역과 RAM 관리 영역이라는 두 가지의 총체적 관점을 [그림 42]에 정리하였다.

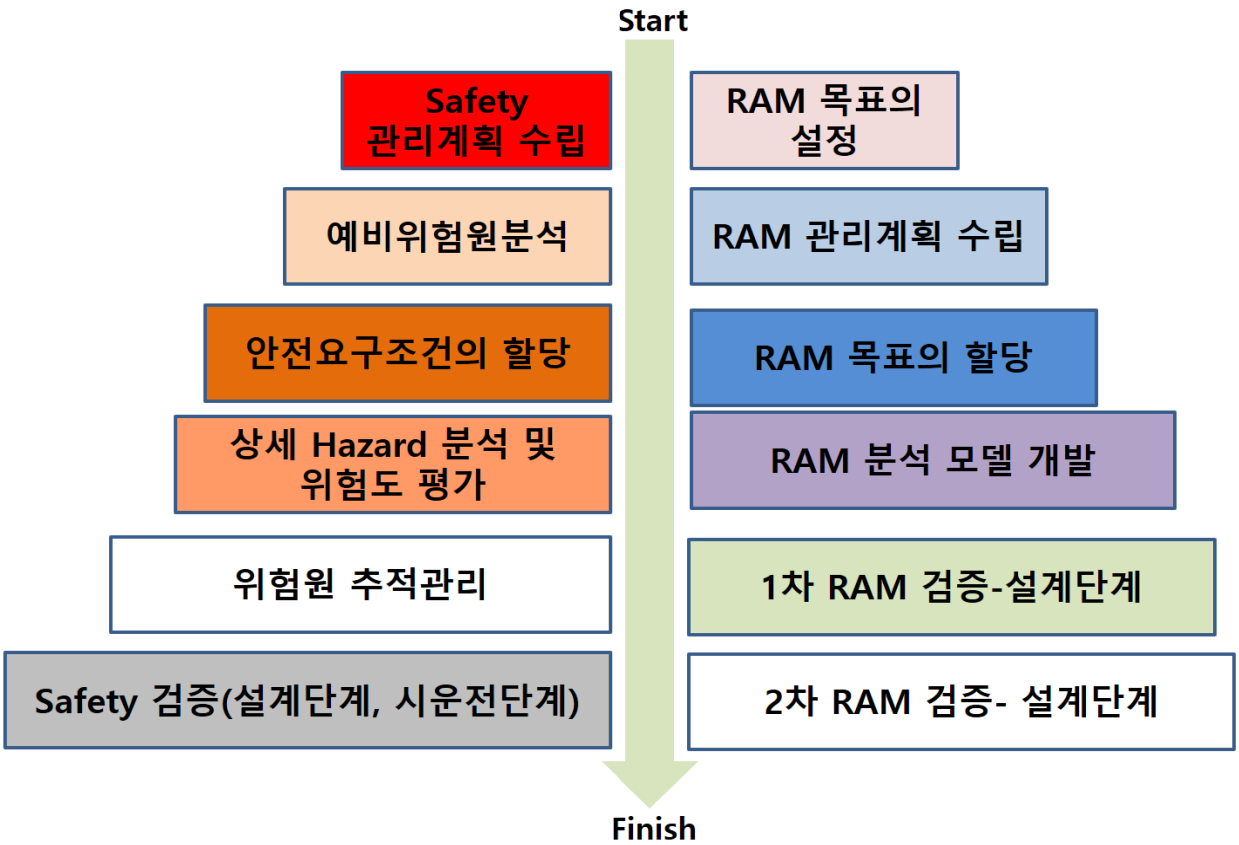


그림 42 철도시스템의 안전성 분석 영역과 RAM 관리 영역 [51]

3. 철도 시스템 RAMS 활동의 수행절차

앞서, 안전성 분석 영역과 RAM 관리 영역을 살펴보았다. 안전성 분석 영역과 RAM 관리 영역 또한 생명주기를 따라 수행된다. 이러한 생명주기에 따른 철도 시스템 RAMS 활동의 수행 절차를 [그림 43]에 나타내었다.

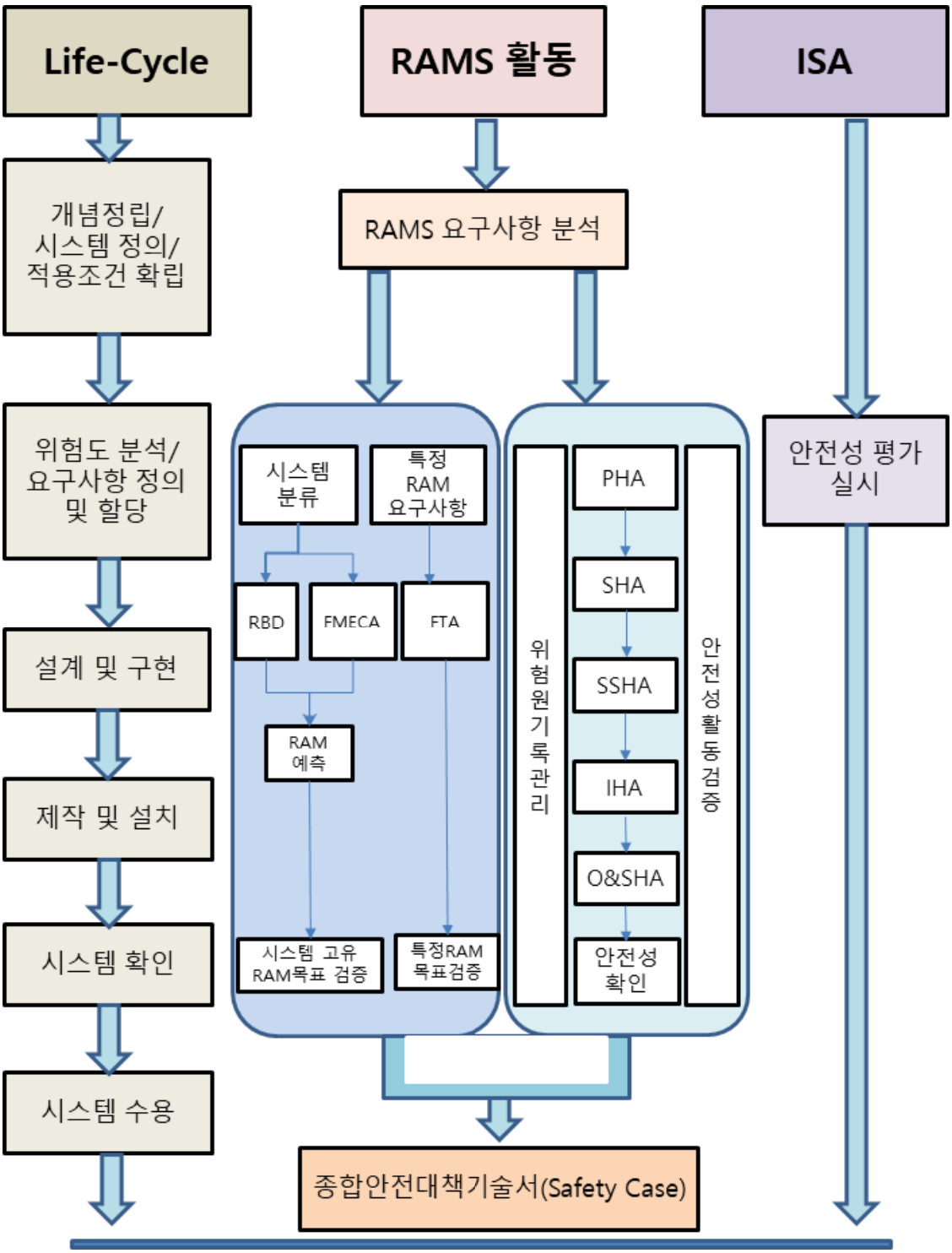


그림 43 철도시스템 전 생명주기에서 요구되는 RAMS 활동 개요

철도 시스템 생명주기의 첫 단계인 개념 및 시스템 정의와 적용 조건 확립 단계에서 수행되는 RAMS 요구사항 분석 결과를 근거로 RAM 분석 업무가 생명주기를 따라 수행된다. RAMS 분석을 위한 대상 선정 및 분석 수준을 결정하기 위해 시스템 분류 업무를 우선 수행한다.

시스템 분류 업무는 시스템 아키텍처 정보를 바탕으로 수행되어야 한다. 이러한 정보를 바탕으로 수행된 시스템 분류를 통해 정의된 분석대상(최하위 수준)을 기준으로 Failure mode, effects and criticality analysis(FMECA)를 수행한다. 이후 시스템 기능 블록 다이어그램(Function block diagram)을 근거로 한 신뢰성 블록 다이어그램(Reliability block diagram; RBD, 이후 RBD)를 작성해야 한다. 또한 FMECA 및 RBD를 근거로 시스템의 신뢰성 및 유지보수성을 예측한다. 그리고 특정 RAM 요구사항을 예측하기 위해 Fault tree analysis(FTA)를 수행하며, 정량적인 예측결과를 도출하기 위해 FMECA에서 분석된 결과를 활용한다.

제작 및 설치 이전에 시스템의 신뢰성 및 유지보수성 예측결과와 특정 RAM 요구사항에 대한 예측결과가 RAM 요구사항에 충족 하는가에 대한 검증업무를 수행해야 한다. 또한 입증 업무의 범위는 설계단계에서 예측된 RAM 분석 데이터와 요구사항간의 충족여부 검토가 해당되며, 시운전 단계나 운용단계에서 수행되는 통계적 입증은 포함되지 않는다.

철도 시스템 생명주기의 첫 단계인 개념 및 시스템 정의와 적용 조건 확립단계에서 수행되는 RAMS 요구사항 분석결과를 근거로 안전성 분석은 생명주기를 따라 수행한다. 예비위험원분석(Preliminary hazard analysis; PHA, 이하 PHA) 결과를 근거로 철도 시스템의 위험원 관리를 계층적으로 수행해야 한다. 우선 철도 시스템의 시스템 수준에서 위험원 관리를 위해 시스템 위험원 분석(System hazard analysis; SHA, 이후 SHA)을 수행한다. 수행된 SHA 결과를 근거로 철도 시스템을 구성하는 서브시스템에 대한 위험원 관리를 위해 서브시스템 위험원 분석(Subsystem hazard analysis; SSHA, 이후 SSHA)을 수행한다. 또한 서브시스템 간의 인터페이스에서 발생 가능한 위험원의 관리를 위해 인터페이스 위험원 분석(Interface hazard analysis; IHA, 이후 IHA)을 수행한다. 유지보수를 포함한 시스템 운용상의 위험원 관리를 위해 운용 및 지원상의 위험원 분석(Operation & support hazard analysis; O&SHA, 이후 O&SHA)을 수행한다.

설계 및 구현, 제작 및 설치단계에서 수행되는 위험원 관리는 설계 및 기타 운용조건의 변경에 따라 생명주기의 시스템 확인단계에서 수행되는 시험 및 시운전 단계 이전까지 갱신하여 관리한다. 시스템 확인 단계, 즉 시험 및 시운전 단계에서는 설계 및 구현, 제작 및 설치단계에서 도출된 위험원 관리가 설계 및 운용절차에 반영 되었는가를 확인하는 안전성 확인 업무를 수행한다. 본 시스템 안전성 분석 가이드에서 제외되는

내용이지만 RAMS 활동과는 독립적으로 수행되는 독립안전평가업무의 결과를 검토, 반영함으로써 안전성 분석 검증을 수행해야 한다. 생명주기에 따라 수행된 안전성 분석 및 RAM 관리에 대한 결과를 총괄하여 종합안전대책보고서(Safety Case)를 수립함으로써 철도 시스템에 대한 RAMS 활동을 종료한다.

본 시스템 안전성 분석 가이드는 표준에서 다루고 있는 안전성 분석 영역과 RAM 관리 영역 중에서 안전성 분석 영역에 관한 부분을 다룬다. 안전성 분석 영역에서 안전성 분석의 수행에 대하여 기술한다. 또한 표준은 위험원 식별(Hazard identification), 위험원 분석(Hazard analysis), 위험원 관리(Hazard management)를 위해서 안전성 분석에 필요한 분석 절차와 기법을 제시한다. 따라서 본 시스템 안전성 분석 가이드의 목적은 철도 시스템의 안전성 분석을 수행하는데 필요한 정보를 제공하는데 있다.

4. 관련 국제표준

최근 철도산업의 국제적 추세는 유럽을 중심으로 표준화 되고 있다. 특히 철도 시스템과 컴포넌트 사양, 컴포넌트의 개발 및 운용의 각 프로세스에 대해 안전 요구사항을 규정하는 European Norm(EN)계열의 안전성 표준들과 철도 시스템의 안전성을 확보하기 위한 RAMS 활동에 대한 요구사항들이 International Electrotechnical Commission(IEC) 표준화 되었다. 철도 RAMS 관련 표준 및 적용현황에 대해서 [그림 44], [그림 45]에서 확인할 수 있다.

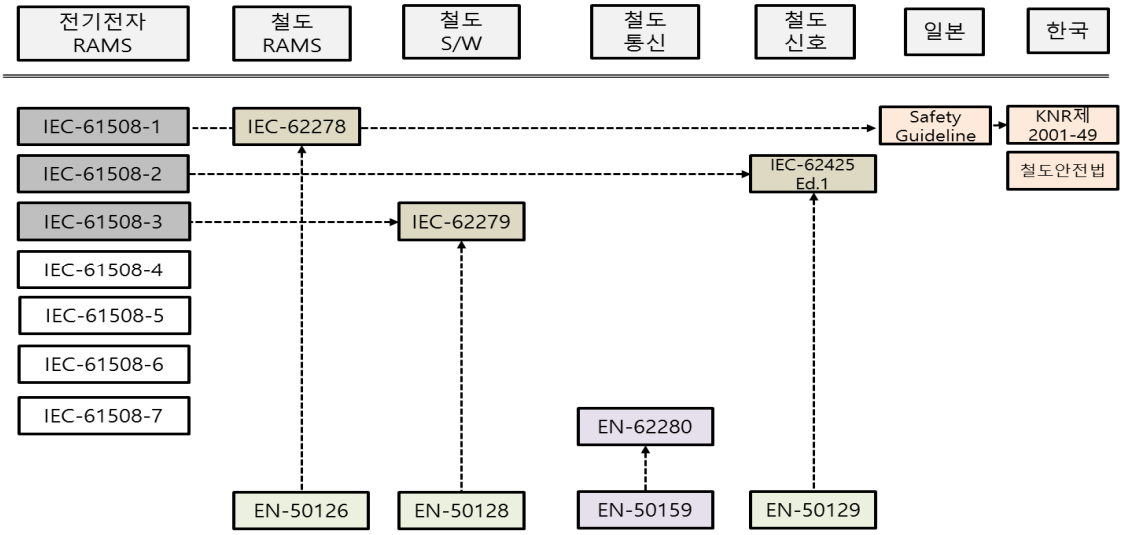


그림 44 철도 RAMS 관련 표준 [10]

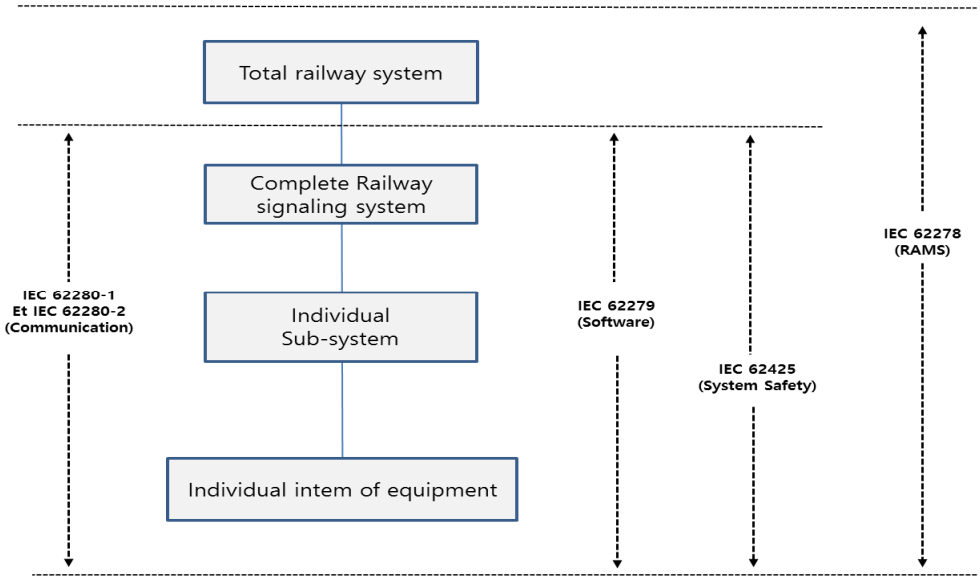


그림 45 IEC의 주요 철도분야 적용 표준현황 [10]

5. 안전성 분석 표준 연계 개요

철도 시스템에 관련된 안전 표준에는 다음 네 가지가 존재하고 있다.

- IEC 62278(EN 50126) : RAMS
- IEC 62279(EN 50128) : 소프트웨어
- IEC 62425(EN 50129) : 안전성 인증
- IEC 62280(EN 50159) : 통신

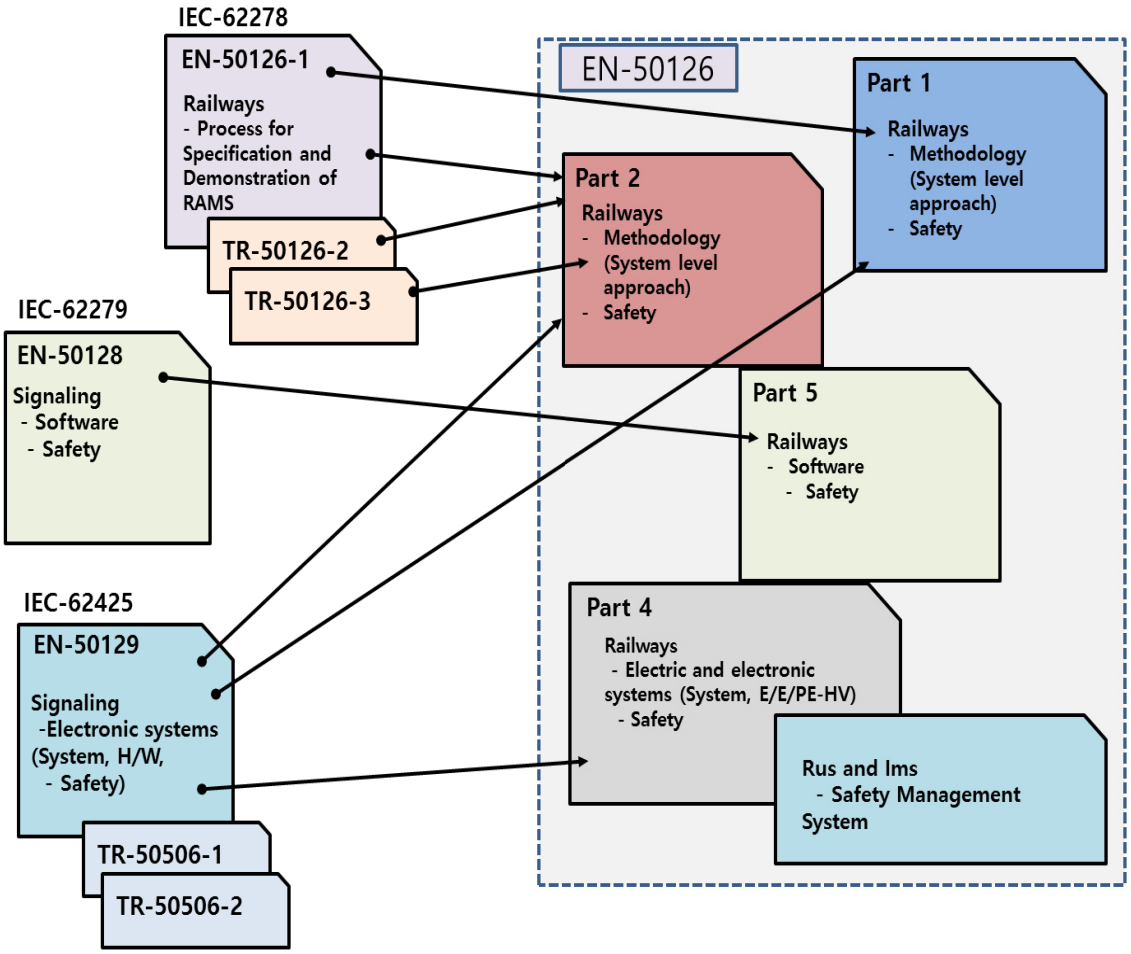


그림 46 유럽에서의 RAMS 표준 재편성 개요 [7, 50]

[그림 46]의 네 가지 표준은 일반산업 분야에 있어서 컴퓨터제어기능을 대상으로 한 안전 표준인 IEC 61508을 기초로 해서 UIC(French: Union Internationale des Chemins de fer or International Union of Railways : 국제철도연합)의 기술지침과 유럽 각국의 철도 시스템의 기술요구사항을 통합하였다. IEC 61508에서는 안전성 분석 생명주기의 범위를 개념설계에서 폐기까지 모든 단계를 포함하였으며 요구되는 안전성 수준에 맞는 기술요구사항을 정하는 안전 무결성 등급(Safety integrity level; SIL, 이후 SIL), 이 두 가지의 개념을 도입하였다. 이는 생명주기의 각 단계를 엄밀히 구분하고 관리함으로써

체계적으로 수행함과 동시에 요구되는 안전성 수준에 맞추어 적합한 안전성 기준을 결정할 수 있도록 하였다. 이 같은 개념에 기초한 안전성 분석이 철도를 포함하여 많은 분야에 있어서 향후 주류가 될 것으로 여겨진다.

표 29 철도분야 안전표준 주요특징 및 설명

표 준 명	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 14단계에 대해서 세부적인 RAMS 활동에 대해 정의하고 있다. 따라서 RAMS 요구사항을 개발하고 이행하기 위한 기준을 제공한다.
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.
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.

제 2 절 안전성 분석 개요

표준의 안전성 분석은 철도 시스템에 내재하고 있는 잠재적 위험원이나 결함을 찾아 제거하거나 발생확률을 허용수준 이하로 줄일 수 있도록 하드웨어, 소프트웨어, 설비, 환경, 운영, 문서를 고려한 모든 일련의 활동을 포함한다. 다시 말해 철도 시스템의 안전성 확보를 위해 철도 시스템의 생명주기 동안 내재한 위험원 및 결함을 도출하고, 도출된 위험원 및 결함을 제거하거나 허용수준 이하로 제어하는 일련의 과정과 입증하는 단계로 구성된다.

1. 생명주기 관점에서의 시스템 안전성

본 시스템 안전성 분석 가이드에서는 표준에서 제시하는 생명주기를 토대로 안전성 분석 기법을 활용한 방법에 관하여 기술한다.

그림 47 IEC 62278 시스템 생명주기 [7]

표준에서 시스템 생명주기는 [그림 47]과 같이 크게 14단계로 구성되어 있다. 표준에서 안전성 분석은 생명주기 내에서 시스템의 안전성 향상 및 입증을 위해 각 단계별로 수행되어야 한다. 생명주기 단계별 주요 내용을 기술하고 수행되어야 할 안전성 분석 기법에 대해 [표 30]에 기술하였다.

표 30 시스템 생명주기에 따른 주요 안전성 분석

안전성 분석 단계	주요 안전성 분석	IEC 62278 생명주기 단계
시스템 및 예비위험원분석단계	1. 시스템 기능 요구사항, 인터페이스 요구사항, 운영시나리오를 토대로 위험 분석이 수행된다. 2. 과거 유사시스템을 통한 위험원 리스트가 작성된다. 3. PHA 기법을 통한 위험원에 대한 위험도가 예측된다.	생명주기 1, 2 단계
시스템 위험원 분석 및 도출 단계	1. 세부 설계 자료를 토대로 SHA 기법 수행을 통해 시스템으로 인한 사고를 예측한다. 2. SHA, FMEA, HAZOP 기법 등을 통한 위험원을 확인한다. 3. 위험 및 고장률을 산출하기 위한 위험원을 분석 한다.	생명주기 3단계
위험 분석 및 안전성 목표수립 단계	1. 정량적 또는 정성적 방법을 통한 위험 분석이 수행된다. 2. 분석된 위험을 통해 기능 또는 위험원별 SIL 또는 Tolerable hazard rate(THR)이 할당 된다. 3. 시스템 안전 요구사항이 작성된다.	생명주기 4, 5단계
저감대책 수립 및 활동 단계	1.안전성 측면의 고장률은 SIL 등급별 위험측 고장률에 위험측 고장관련 기능 및 모델링 하여 관리 하여야 한다.	생명주기 6,7,8 단계
안전성 검증 및 확인 단계	1. 시스템의 기능 및 성능요구사항의 준수여부를 확인 하여야 한다. 2. 시스템의 시스템요구사항 및 안전 요구사항의 준수여부를 검증 하여야 한다.	생명주기 9단계
안전성 인증 단계	1. 안전성 분석의 완성도의 판단을 위임하는 기관에서 작성하는 평가보고서에 따라 안전성 인증을 진행한다.	생명주기 10단계
안전성 관리 단계	1. 시스템이 운영을 시작한 이후에 작용하는 신뢰성 및 안전성 측면의 유지보수 및 철거 프로그램으로써, 유지보수는 신뢰성과 안전성을 모두 고려하여 주기적 보수 및 고장에 의한 유지보수를 실시되어야 한다.	생명주기 11,12,13,14 단계

생명주기 관점에서의 시스템 안전성은 위험원을 제거하거나 위험원으로부터 발생할 수 있는 사고 확률을 줄일 수 있도록 체계적으로 설계에 적용하는 과정이다. 또한 시스템에 내재된 위험원에 의해 발생할 수 있는 사고를 의도적으로 허용 가능한 수준으로 제어하는 것이다. 따라서 시스템 설계에 있어서 반드시 시스템 설계 초기부터 저감대책을 검토 및 수립해야 한다.

본 시스템 안전성 분석 가이드의 기준인 표준에서 제시하는 시스템 생명주기를 근간으로 안전성 분석 영역과 RAMS 관리 영역의 활동은 [그림 48]와 같이 구분할 수 있다. 안전성 분석 영역만을 바라본다면, 안전성 분석은 철도 시스템 설계 초기에 위험도 허용 기준을 수립한다. 그리고 이를 기반으로 안전관리 계획서에 반영하여 안전성 분석 수행 및 안전요구사항 도출 과정을 거치게 된다. 위험도 평가를 반영한 안전요구사항은 설계 반영을 통해, 운용단계에서 시험 및 시운전 과정의 검증을 거치게 된다. 최종적으로 철도 운영기관을 통해 안전성 관리를 받게 된다.

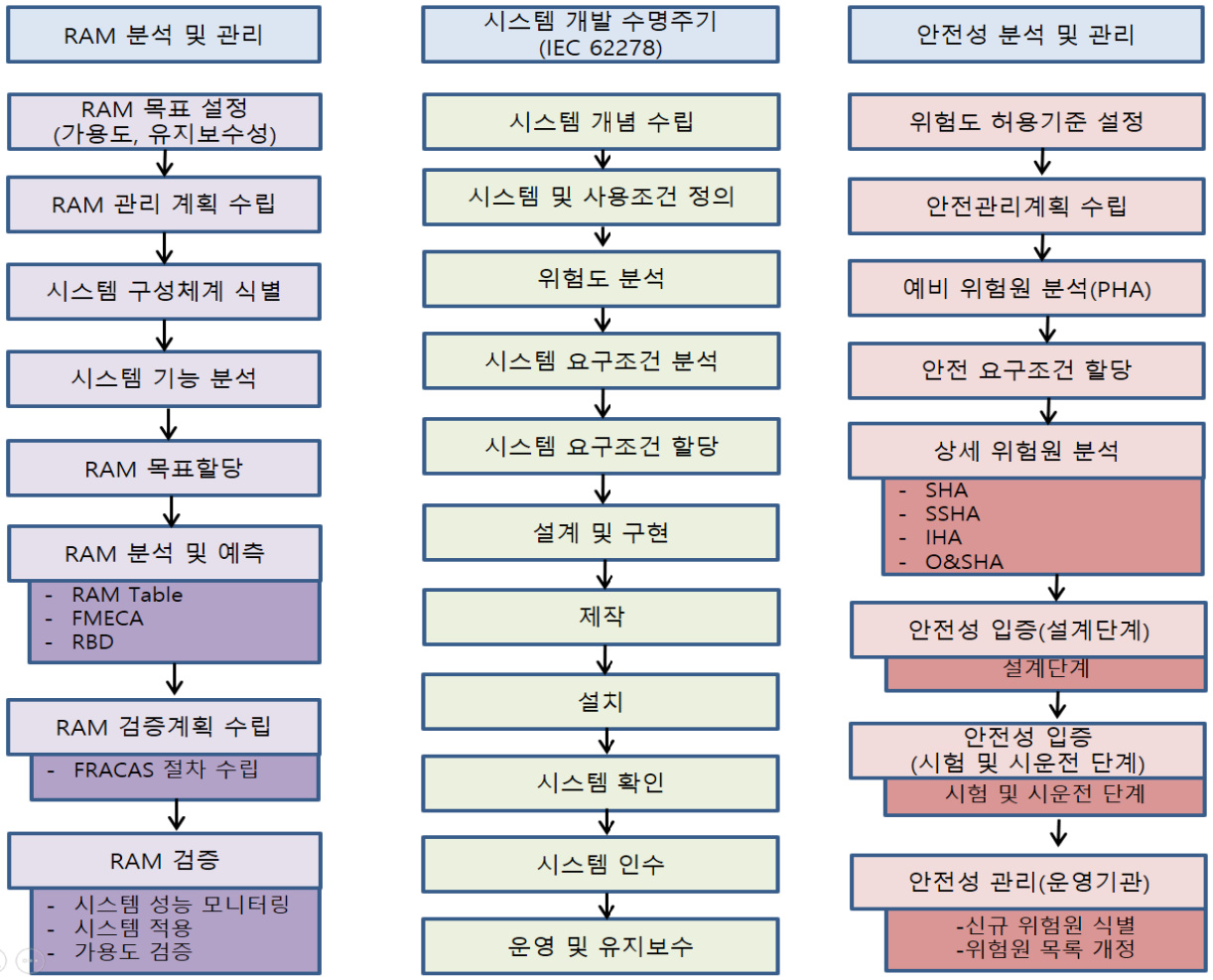


그림 48 생명주기와 RAMS & 안전성 분석 및 관리 [50]

2. 소프트웨어 안전성

소프트웨어 안전성은 시스템 수준에서의 안전성 분석 결과를 바탕으로 수행되어야 한다. 이를 토대로 시스템 수준에서 식별된 물리적 컴포넌트가 소프트웨어적 특성을 보유하였다면, 소프트웨어 안전성 분석을 수행해야 한다. 소프트웨어 안전성은 IEC 62279와 소프트웨어 생명주기 프로세스를 기초로 한다. 또한 시스템 설계에 적용되는 설계 프로세스와 장치, 통합, 시운전 시험 등을 거쳐야 한다. 이때 철도 시스템이 지닌 소프트웨어 설계의 민감한 요소들을 파악하는 고장계통 분석을 통해 정확한 구현이 가능해야 한다. 그리고 안전성 분석 결과를 필수 시스템(Vital System)과 서브시스템 요구사항 명세서 내에 안전 요구사항 형태로 문서를 정리해야 한다.

소프트웨어 안전성 분석의 핵심은 코드에 대한 소프트웨어 개발 공학 관점에서의 안전성 분석에 있다. 이러한 확인을 위한 선행 조건들은 다음과 같다.

- 안전 요구사항을 비롯한 시스템과 서브시스템 소프트웨어 요구사항 명세서를 위험원과 안전성 분석, 점검 전반에 걸쳐 작성한다. 또한 요구사항의 정확성, 완성도, 서브시스템 간 일관성 등을 확인해야 한다.
- 서브시스템 소프트웨어의 안전 요구사항을 해당 소프트웨어 설계까지 추적 가능해야 한다. 안전성 분석과 검토를 안전에 필수적인 기능들에 대해 수행하여 이들의 정확성, 완성도, 일관성 등을 확인해야 한다.
- 소스코드가 소프트웨어 안전성 설계 기준에 부합하는지 확인해야 한다.
- 소프트웨어 상세 설계 준수 여부, 검토 목표는 소스코드가 정확하며 소프트웨어 상세 설계 관점에서 완벽한지 여부와 소스코드에 문서화 되지 않은 기능들이 구현되어 있는지 여부를 확인하는데 있다.
- 일관성 확인: 데이터 조작 전에 데이터가 일관성을 유지하고 있는지를 확인한다. (제로 속도 확인, 신호기 이동명령 확인 등)
- 타당성 확인: 관련 데이터들이 논리적으로 정확하게 합당하며 데이터 범위는 예상 범위 내에 있는지 확인한다.
- 하드웨어 감시확인: 내장 시험, 출력의 교차 비교하여 확인한다.
- 적정 수준의 통합 시험과 확인 시험을 계획하여 소프트웨어 코드 리뷰에 따라 실행하여 구현 내용이 설계와 일치하며 안전 요구사항을 만족하는지 확인한다.
- 단, 상용 소프트웨어(COTS)를 사용할 경우에는 소프트웨어 안전성 분석도 수행되어야 한다(단, 본 가이드라인에서는 COTS에 대한 안전성 보증 활동은 제외함). 하지만, 제외된 사항에 대해 동일 사양 제품의 적용사례 또는 카탈로그 등과 같은 증빙자료를 제시해야 한다.

철도 시스템의 소프트웨어 개발과 관련해서는 IEC 62279 기반 안전성 인증을 받기 위해 아래와 같은 검증기법이 요구된다.

- 철도 소프트웨어 인증지원 도구 즉, 기능 안전성의 블랙박스 테스팅이 가능하도록 최종 개발한 시스템에 활용하는 통신 인터페이스를 직접 모의하고 실제 신호를 발생, 주입함으로써 시스템 체계를 구성하는 하위 시스템의 모든 소프트웨어 영역을 검증할 수 있도록 기술 지원이 필요로 한다.
- 또한, 자동차 산업계의 MISRA-C, MISRA-C++, 항공 산업계의 JSF AV C++, 각종 보안 관련 분야의 CERT-C, CERT-C++ 등을 참고하여 철도 시스템을 위한 소프트웨어 코딩 규칙 표준 및 지침이 필요로 한다.
- 이를 적용함으로써, 시스템의 소프트웨어 안전성・신뢰성 확보가 가능해진다.

소프트웨어 시험은 제작 시제품의 소프트웨어 테스팅을 수행하여 품질 및 안전성 검증의 적합성을 문서화하여 제시하여야 한다.

- 코딩규칙 준수(MISRA-C 코딩규칙, IEC 61508 3 코딩규칙, IEC 62279 코딩 규칙) 여부 입증
- 동적 테스팅(MC/DC 테스트, 제어흐름 테스트 또는 경로 테스트)을 수행
- 기능 테스팅(기능 요구사항 테스팅, 경계값 분석 포함) 수행

제 3 절 표준(IEC 62278) 기반 안전성 분석

1. 안전성 분석 수행 시점

시스템 안전성 분석 가이드에서는 철도 시스템의 안전성 분석에 대한 기법들에 대하여 다루고 있다. 이러한 기법들이 시스템 생명주기 상에서 어느 시점에서 활용해야 하는지 사용자가 해당 정보를 인지할 수 있도록 [그림 49]에 제시하였다.

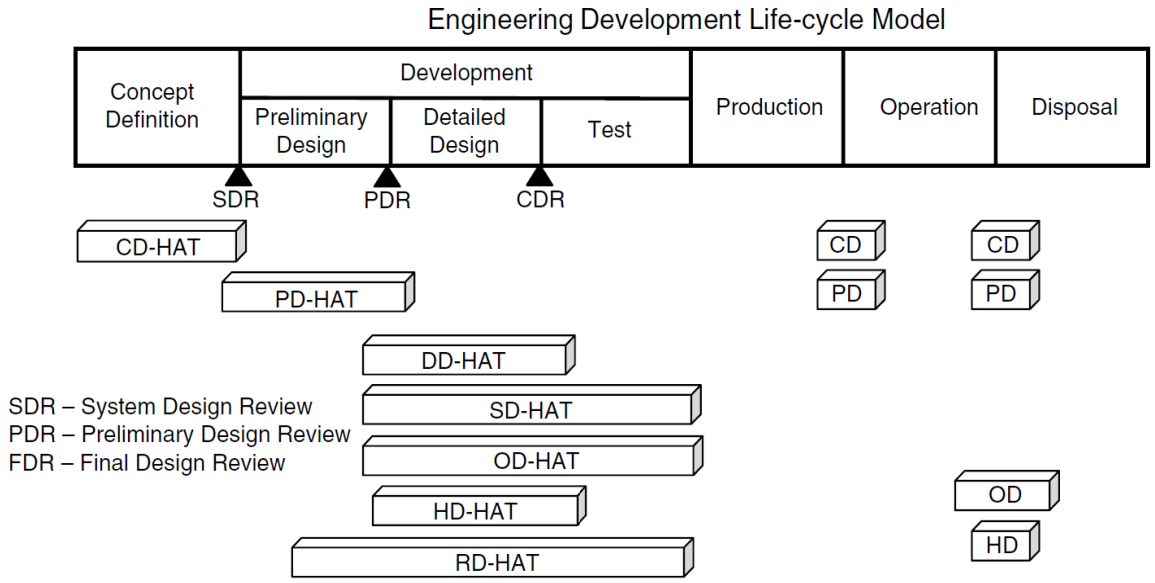


그림 49 안전성 분석 기법의 수행시점 [52]

시스템 생명주기는 Concept Definition, Development, Production, Operation, Disposal의 5단계로 구성되어 있으며, 대부분의 안전성 분석 수행은 시스템 설계 초기에 수행함으로써 안전성을 확보하는데 주력하고 있다. 특히 Development 단계에서 세부적으로 Preliminary Design, Detailed Design, Test 단계를 통해 많은 안전성 분석을 동시 및 지속적으로 수행한다. 시스템 생명주기 상의 주요 마일스톤(Milestone)에는 SDR(System Design Review), PDR(Preliminary Design Review), CDR(Critical Design Review) 등이 있다.

주요 마일스톤에서 각 단계마다 수행되는 안전성 분석에 대한 검토가 이뤄진다. 검토를 통해 해당 단계 수준에서 확보해야할 안전성에 대한 분석이 제대로 되었는지 확인한다. 시스템 생명주기 상에서 각 시점마다 수행되는 안전성 분석 기법들에 대한 Type에는 CD-HAT, PD-HAT, DD-HAT, SD-HAT, OD-HAT, HD-HAT, RD-HAT가 있다. 해당 Type들마다 여러 안전성 분석 기법이 포함될 수 있으며, 상세한 내용은 [그림 50]에서 다룬다.

Technique	Type	Identify Hazards	Identify Root Causes	Life-Cycle Phase	Qualitative/ Quantitative	Skill	Level of Detail	I/D
PHL	CD-HAT	Y	N	CD-PD	Qual.	SS	Minimal	I
PHA	PD-HAT	Y	P	CD-PD	Qual.	SS	Moderate to in-depth	I-D
SSHA	DD-HAT	Y	Y	DD	Qual.	SS, Engr., M&S	In-depth	I-D
SHA	SD-HAT	Y	Y	PD-DD-T	Qual.	SS, Engr., M&S	In-depth	I-D
O&SHA	OD-HAT	Y	Y	PD-DD-T	Qual.	SS, Engr., M&S	In-depth	I-D
HHA	HD-HAT	Y	Y	PD-DD-T	Qual.	SS, Engr., M&S	In-depth	I-D
SRCA	RD-HAT	P	N	PD-DD	Qual.	SS	In-depth	N/A
FTA	SD-HAT, DD-HAT	P	Y	PD-DD	Qual./Quant.	SS, Engr., M&S	Moderate to in-depth	D
ETA	SD-HAT	P	P	PD-DD	Qual./Quant.	SS, Engr., M&S	Moderate to in-depth	D
FMECA	DD-HAT	P	P	PD-DD	Qual./Quant.	SS, Engr., M&S	In-depth	I
FaHA	DD-HAT	Y	P	PD-DD	Qual.	SS, Engr., M&S	In-depth	I
FuHA	SD-HAT, DD-HAT	Y	P	CD-PD-DD	Qual.	SS, Engr., M&S	Moderate to in-depth	I
SCA	SD-HAT, DD-HAT	P	Y	DD	Qual.	SS, Engr., M&S	Moderate to in-depth	D
PNA	SD-HAT, DD-HAT	P	N	PD-DD	Qual./Quant.	SS, Engr., M&S	In-depth	D
MA	SD-HAT, DD-HAT	P	N	PD-DD	Qual./Quant.	SS, Engr., M&S	Moderate to in-depth	D
BA	SD-HAT	Y	P	PD-DD	Qual.	SS, Eg	Moderate to in-depth	I
BPA	DD-HAT	Y	P	PD-DD	Qual.	SS, Engr., M&S	In-depth	D
HAZOP	SD-HAT, DD-HAT	Y	P	PD-DD	Qual.	SS, Engr., M&S	Moderate to in-depth	I
CCA	SD-HAT, DD-HAT	Y	P	PD-DD	Qual./Quant.	SS, Engr., M&S	Moderate to in-depth	D
CCFA	SD-HAT, DD-HAT	Y	P	PD-DD	Qual.	SS, Engr., M&S	Moderate to in-depth	D
MORT	SD-HAT, DD-HAT	Y	P	PD-DD	Qual./Quant.	SS, M&S	Moderate to in-depth	D
SWSA	SD-HAT, DD-HAT	Y	P	CD-PD	Qual.	SS, Engr., M&S	Moderate to in-depth	N/A

[a]Abbreviations: Y = yes, N = no, P = partially; Skill required: SS = system safety; Engr. = engineering electrical/mechanical/software; M&S = math & statistics; life-cycle phase: CD = conceptual design, PD = preliminary design, DD = detailed design, T = testing; I-inductive, D = deductive.

그림 50 안전성 분석 기법 선정을 위한 속성정보 요약 [52]

앞서 시스템 생명주기 상에서 안전성 분석 수행 시점을 Type을 통해 제시하였다. 해당 Type에는 여러 기법들이 포함되어 있으며, 기법마다 가지고 있는 속성정보들의 요약을 [그림 50]에 기술하였다. [그림 50]에서 여러 안전성 분석 기법을 확인할 수 있으며, 이러한 안전성 분석 기법들을 시스템 생명주기에 따라 각 수행 시점에 활용하여 효과적인 결과를 얻는데 중점을 둔다. 안전성 분석 수행 시점에 따른 Type 중 가장 많이 활용되는 기법들은 PHA, SSHA, SHA, FTA, Failure mode and effects analysis(FMEA), Hazard and operability study(HAZOP) 등이 있다. 예를 들어 PHA의 경우 PD-HAT Type으로서 시스템 생명주기 상에서 Preliminary Design 단계에서 수행되는 것을 알 수 있다. 따라서 사용자가 철도 시스템 설계에서 생명주기 어느 시점 상에서 안전성 분석을 수행할 것인지에 따라 해당하는 기법을 선정하여 수행하는데 필요한 정보를 제시하였다.

2. 철도 시스템 안전성 분석의 수행절차

철도 시스템의 안전성 확보 및 입증을 위한 안전성 분석은 우선적으로 시스템의 잠재적인 위험원을 도출하고 이에 대한 분석 및 평가를 수행해야 한다. 따라서 최종적으로 도출된 위험원이 안전성 분석을 통해 허용수준 이하로 제어되었는지 입증해야 한다. 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 절차는 7단계로 되어 있으며 [표 31]에 기술하였다.

표 31 안전성 분석 절차 및 주요 산출물

안전성 분석	주요 내용	주요 기법 및 산출물
1. RFP 안전 요구사항 분석/계획	- RFP 를 통한 안전과 관련된 안전 요구사항 분석 수행 - 안전성 분석 수행 계획 수립	안전성 계획 수립 및 안전 계획서 작성
2. 시스템/위험원 정의 및 평가	- 발생 가능한 잠재 위험원에 대한 도출/분석 수행(PHA) - 시스템의 구성, 요소, 기능분석 - 실제 동작시의 운영조건, 기술 특성 검토 - 시스템의 기본적인 기능사양 해석 - 위험요인 판별 - 시스템의 신뢰성/안전성에 영향을 미치는 위험원의 위험을 등급 테이블에 근거하여 정성적으로 평가(SHA) - 각 단계별 위험원인 분석 및 위험 평가 - 위험에 따른 안전요건의 검증 및 이에 따른 위험 수용 여부 결정	- 예비위험원분석(PHA) 수행 - 시스템 위험원 분석 (SHA) 수행 - 서브시스템 위험원 분석 수행 (SSHA) 수행 - FMEA (Failure Mode & Effects Analysis) 수행
3. 시스템 안전 요구사항 정의	- 위험의 흐름을 추적하며, 안전관련 요구사항 도출 - 안전 요구사항, 환경 요구사항, 설비 요구사항, 소프트웨어 요구사항, 기능상의 안전 요구사항 정의	- FRACAS(Failure Reporting, Analysis and Corrective Action System) 수행
4. 안전성 구현	- 정의된 안전 요구사항 설계, 제작에 적용 - 안전 요구사항 추적 관리 - 저감대책 적용 확인 - 위험원 로그 관리 - 운영/유지보수 매뉴얼에 따른 O&SHA 갱신	- O&SHA(Operating & Support Hazard Analysis) 수행
5. 안전성 검증 및 인수	- 안전성 보증활동에 따른 품질, 안전, 기술관리 내용을 반영 - 안전성 보증활동에 따른 결과를 반영한 기술적인 안전체계를 기술 - 설계, 구현, 시험단계에서 신뢰성/안전성을 확보하기 위한 대책 적용 및 최종적인 보고서 산출 - 안전 요구사항 검증(Verification) 수행	- O&SHA 수행 및 V&V 수행
6. SIL 입증	- 최종 작성되는 종합안전대책보고서에 의한 품질관리 조건, 안전관리 조건, 기술적 안전조건, 정량화된 안전 목표 평가 - SIL 만족여부 확인	- 종합안전대책보고서(Safety Case Report) 작성
7. 유지보수	- 감시 및 수정 - 최종 작성되는 종합안전대책보고서에 의한 품질관리 조건, 안전관리 조건, 기술적 안전조건, 정량화된 안전 목표 평가 - SIL 만족여부 확인	

앞서, 철도 시스템을 대상으로 안전성 분석을 수행하기 위한 안전성 분석 기법 식별과 다양한 기법들을 생명주기 각 단계별 수행 시점 및 절차에 관해 간략히 설명 하였다. 철도 분야에서 주로 쓰이는 안전성 분석 기법에는 PHA, HAZOP, FMEA 등이 있다. 여러 기법들 중에 Preliminary Hazard List(PHL)과 PHA는 시스템 설계 초기 위험원 식별을 위한 방법들이며, 초기 식별된 위험원을 바탕으로 상세한 위험원 도출 방법으로 FMEA, HAZOP이 사용된다. 일반적으로 시스템 내부의 위험원 식별은 하나의 안전성 분석 기법만으로는 완벽하게 식별될 수 없다. 따라서 개별 안전성 분석 기법이 지니는 특성을 상호 보완적으로 사용하는 것이 바람직하다. 이러한 과정을 표준에서는 Failure Reporting Analysis & Corrective Action System(FRACAS)를 통해 고장 데이터를 수집하고, 고장 원인을 결정하기 위한 절차와 수행된 개선 조치에 대한 문서를 제공하는 목적을 갖는다.

지금까지 수행한 안전성 분석을 통해 안전 기능 및 안전 무결성 등급을 포함한 상세 요구사항들은 시스템 사양서로 문서화 된다. 또한 관련한 상세 요구사항은 (1) 위험원 식별 및 분석, (2) 위험도 평가 및 분류, (3) 안전 무결성 등급 할당의 정보를 요구한다. 안전성 분석은 시스템 생명주기 동안 지속적이고 반복 수행을 통해 갱신해야 한다. 이러한 과정을 거쳐 갱신되는 모든 안전과 관련된 요구사항과 저감대책은 시스템 설계 및 제작에 반영된다.

위험원을 제거 하거나 위험도를 감소시키기 위한 모든 안전성 분석은 시스템 생명주기 동안 위험원 로그(Hazard Log)를 통해 추적 및 관리되어야 한다. 위험원 로그는 변경 과정을 거치며, 다음과 같은 항목이 발생될 때마다 갱신한다.

- 관련 위험원이나 잠재적 사고가 도출 될 때
- 관련 사고가 발생할 때
- 기존 위험원과 사고에 관련된 추가 정보가 특별히 이슈화 될 때
- 안전문서가 만들어 졌거나 다시 발행될 때

RAMS 관리자 또는 이를 입력할 수 있는 권한을 가진 철도 시스템 설계 구성원과 함께 위험원 로그 갱신을 수행한다. 위험원 로그가 갱신될 때 항상 철도 시스템 설계 관리자 또는 관리자의 대리인으로부터 승인을 받는다. 위험원 로그 관리는 설계 시 파악한 잠재적 안전 위험요소를 파악, 분석, 제어, 정리하는데 목적이 있다. 위험원 로그는 시스템 생명주기에 걸쳐 지속적으로 갱신해야 하며, 또한 설계가 변경될 경우에도 이에 해당하는 사항이 위험원 로그에 기록되며, 설계변경에 따른 프로세스는 품질보증 계획서에 따른다. 그리고 설계변경에 대한 상태감시 사항이 기록해야 한다.

안전성 검증은 생명주기 각 단계에서 기록된 안전 요구사항이 추적/관리되고 저감대책에 대한 적용과 시험이 정확하게 수행되었는지를 확인하는 활동이다. 그리고 안전 요구사항에 대해 완벽하게 시스템 및 서브시스템이 개발되었는지 검증하는 활동이다. 또한 통합된 시스템의 분석과 시험을 통해 시스템이 의도된 설계 목적에 적합한지와 안전 무결성 등급에 따른 안전 관련 사항의 만족여부를 확인하는 활동이다. 따라서 안전성 검증은 생명주기 각 단계별로 추적 관리되지 않거나 적절한 시험 및 평가가 이루어지지 않은 요구사항은 반드시 검증단계에서 확인해야 하고 문서화해야 한다.

또한 부적합 사항은 반드시 보고서에 포함되어 작성해야 한다. 이러한 검증 및 확인 활동은 만일 시스템에 어떤 일련의 수정 혹은 추가가 되는 경우에도 적절하게 반복되어 수행해야 한다. 검증 및 확인에 대한 상세한 사항은 검증 및 확인 계획서를 참고한다. 안전성 검증 및 확인은 안전 무결성 등급에 따라 독립성이 확보된 팀에 의해 검증 및 확인 계획서에 따라 수행한다. 안전성 검증 및 확인 보고서에는 최소한 다음의 내용들이 포함해야한다.

- 활동수행 일자 및 수행전문가 정보
- 적용방법 및 결과, 조치사항 및 현재 상태
- 테스트, 분석, 검증을 통한 안전 요구사항의 충족 증거
- 적용한 규격 및 표준의 부합 여부
- 사용한 도구의 적절성

시스템 설계와 테스트가 완료되고 나서 해당 시스템 또는 서브시스템이 지니고 있는 목표 안전 무결성 등급에 대한 최종적인 입증 문서로서 종합안전대책보고서를 작성해야 한다. 인증 단계는 수행된 모든 안전성 분석에 대해 독립된 인증 조직에 의해 평가 및 인증을 받는 단계로서, 평가 및 인증을 위해 준비해야 할 활동에 대해 기술 한다. 인증은 고객의 요구에 의해 수행되기도 하며, 자체적으로 인증을 취득하기 위해 수행되기도 한다. 평가 및 인증을 받기 위해서는 종합안전대책보고서를 제출한다. 철도 시스템이 안전을 확보했음을 증명하기 위해서는 컴포넌트나 서브시스템 또는 시스템 수준에서 다음과 같은 조건들을 만족해야 한다.

- 품질관리에 대한 근거 확보
- 안전관리에 대한 근거 확보
- 기능적/기술적 안전에 관한 근거 확보

3. 안전성 분석에서의 위험원 역할

시스템의 안전성을 확보하기 위해서는 위험원이 제거 또는 완화되어야 한다. 따라서 위험원 식별 및 도출은 시스템 안전성 확보에 있어서 매우 중요한 요소이다. 안전성 분석은 시스템의 안전성 확보를 위한 기본적인 토대를 제공하며, 위험원을 식별 및 도출하는데 필요한 기법을 적용하려면 위험원의 본질, 위험원과 사고와의 관계, 그리고 시스템 설계에 미치는 영향을 이해하는 것이 필요하다.

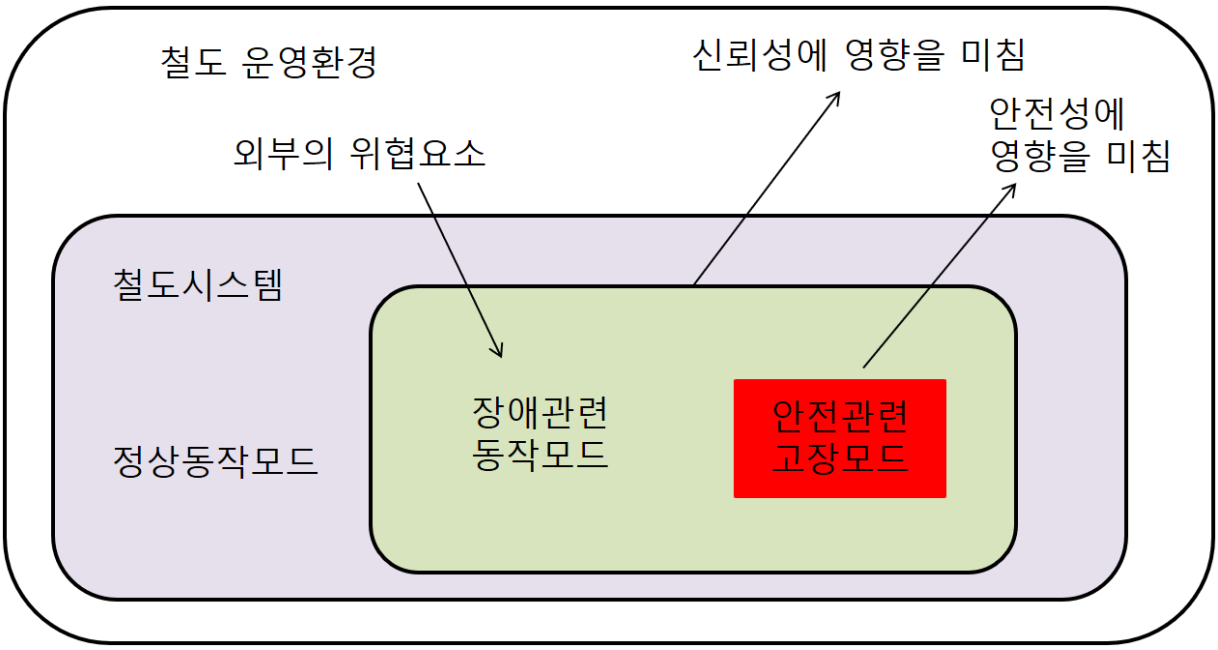


그림 51 철도 시스템의 시스템 안전성 확보를 위해 고려해야할 환경요소 [53]

본 시스템 안전성 분석 가이드에서는 철도 시스템 설계와 관련한 안전성 분석 수행을 목표로 한다. 따라서 철도 시스템 안전성 확보라는 목표를 달성하기 위한 철도 시스템 안전성 분석에 대한 가이드를 제공한다. 철도 시스템의 안전성을 확보하기 위해서는 [그림 51]에 도식화한 철도 시스템이 지니고 있는 정상동작모드와 고장모드에 대한 시스템 정의를 올바로 수행해야 한다. 이러한 과정을 거쳐야 단순히 해당 위험원으로부터의 고장이 장애만 일으키는지, 또는 안전과 직결된 고장인지에 대한 식별 및 분석이 가능해지기 때문이다.

[그림 52]는 본 시스템 안전성 분석 가이드에서 다루는 안전성 분석을 지원하기 위한 안전성 분석 방법의 개략적인 절차를 나타내었다. 철도 시스템의 설계 범위를 통해 시스템 경계를 파악할 수 있으며, 시스템 경계 내부간의 기능/인터페이스, 내・외부간의 기능/인터페이스 등 여러 장치 및 서브시스템을 포함한다. 시스템 내・외부간의 기능/인터페이스 등 여러 장치 및 서브시스템에는 잠재적인 결함이 존재하고 있다.

이러한 결함들이 안전성 분석을 통해 위험원으로 식별될 경우 해당 위험원을 제거한다. 해당 위험원을 제거하기 어려운 경우 저감대책을 통해 위험원의 발생빈도 및 심각도를 낮춘다. 이러한 발생빈도 및 심각도를 통해 해당 위험원의 사고의 발생확률을 도출하여 위험도 평가를 수행한다. 그리고 해당 위험도 결과를 통해 저감대책을 통해 안전성을 확보하였는지 평가하는 것이 핵심이다. 따라서 본 시스템 안전성 분석 가이드를 참고하는 실무자들은 [그림 51]와 [그림 52]을 참조하여 개략적인 이해와 전반적인 활용 절차를 이해 할 수 있을 것이다.

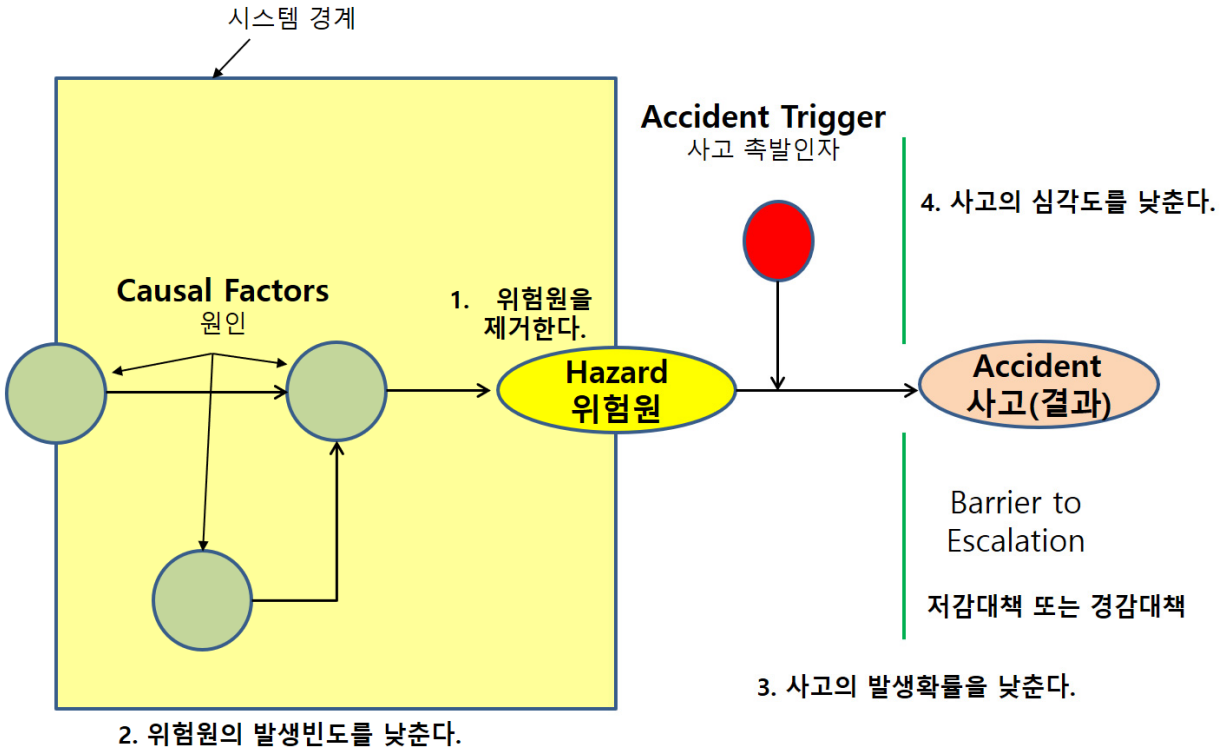


그림 52 안전관리와 위험도저감 [54]

4. 안전 무결성 등급(Safety Integrity Level)

표준에서 안전 무결성(Safety Integrity)은 안전관련 시스템이 일정기간 내에 모든 일정 조건에서 요구되는 안전기능을 성공적으로 수행할 확률로 정의하고 있다. 안전 무결성 등급은 안전성과 관련된 기능에 대하여 정의될 수 있으며, 안전 기능 목표를 달성하기 위해서는 우선 어떤 안전기능을 추가할 것인지를 결정한다. 이후 안전기능의 달성 가능한 정도를 안전 무결성 등급(Safety Integrity Level, 이하 SIL 이라 함)으로 결정한다. 이러한 안전기능을 구성하는 서브기능별로 가장 낮은 수준의 Level 0부터 가장 높은 수준의 Level 4까지 할당 될 수 있다. 안전 무결성 할당의 개념은 아래의 [그림 53]과 같다.

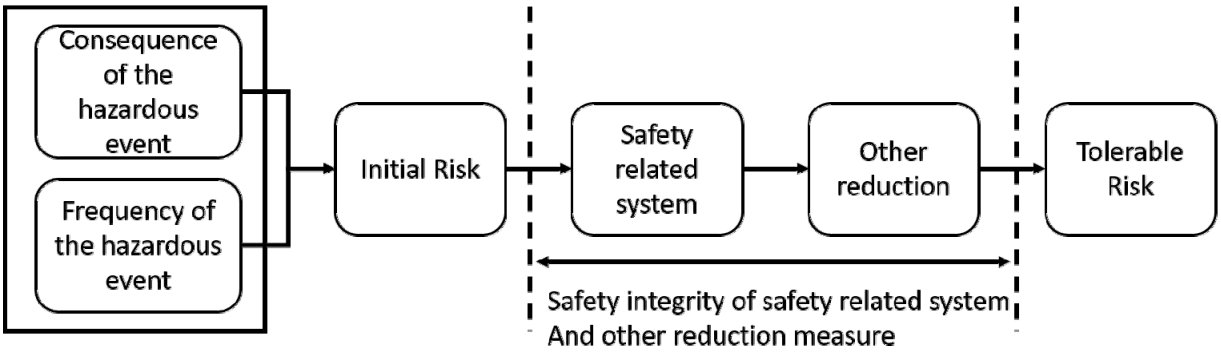


그림 53 안전 무결성 할당의 개념 [55]

또한, 상기의 안전 무결성 할당의 개념과 동시에 각 안전 무결성 등급에 따른 정량적 요구수준으로서 위험 고장 발생빈도(Probability of dangerous Failure per Hour, 이하 PFH 이라 함)를 [표 32]와 같이 정의하고 있다.

표 32 안전 무결성 등급(SIL)과 위험 고장 발생빈도(PFH) [55]

안전 무결성 등급 (SIL)	위험 고장 발생빈도 (PFH)
4	≥ 10-9 to <10-8
3	≥ 10-8 to <10-7
2	≥ 10-7 to <10-6
1	≥ 10-6 to <10-5

[그림 53], [표 32]와 같이 안전 무결성 등급은 안전성 관련 시스템에 대한 위험도 평가 결과에 기반하여 초기 평가된 위험도를 허용 가능한 수준으로 저감시키기 위해 요구되는 시스템 또는 기능에 대한 무결성 등급에 따라 할당되어야 한다.

표 33 SIL 할당을 위한 위험 고장 발생률 [55]

SIL	Range of λ (failures per hour)	Range of λ (failures per year)
1	$1E^{-9} \leq \lambda < 1E^{-8}$	$1E^{-5} \leq \lambda < 1E^{-4}$
2	$1E^{-8} \leq \lambda < 1E^{-7}$	$1E^{-4} \leq \lambda < 1E^{-3}$
3	$1E^{-7} \leq \lambda < 1E^{-6}$	$1E^{-3} \leq \lambda < 1E^{-2}$
4	$1E^{-6} \leq \lambda < 1E^{-5}$	$1E^{-2} \leq \lambda < 1E^{-1}$

[표 32]에서 정의된 PFH는 위험 고장률(λ)로 다시 표현될 수 있으며, 1년을 10000시간으로 정의하여 연간 위험 고장률(λ)으로 나타내면 [표 33]과 같다.

5. 주요 안전성 분석기법의 위험도 분석 평가 기준

급변하는 국내 철도산업에서 안전성 분석의 중요성은 보다 증대되고 있다. 이로 인해, 국외에서 발행된 표준의 역할이 국내 철도 산업 환경에 큰 영향을 주고 있는 실정이다. 대표적인 표준으로는 미국과 유럽을 중심으로 개발된 IEC 또는 EN 표준 등이 있다. 이러한 표준의 특징은 철도시스템의 개념설계부터 운용 단계에 이르기 까지 전 생명주기적 관점에서 위험원으로부터 발생할 수 있는 사고에 대한 위험도 평가 수행을 하여야 한다. 국외 표준뿐만 아니라, 국내 철도 분야 산업 환경에서도 위험도를 정량·정성적 평가에 따라 위험도를 등급별로 산정하고, 이에 대한 등급별 사전 대응 및 조치를 요구하고 있다.

위험도를 평가하기 위해서는 이에 대한 근거로 제시할 기준이 필요하다. 하지만, 국내 철도 분야에서는 위험도 평가 기준에 대해 일관적인 기준을 사용하지 못하고, 운영기관별로 평가 기준에 대한 차이를 보이고 있는 실정이다. 이로 인해, 발생되는 문제는 안전성 분석 및 SIL 인증을 위한 대응에 있어서 일관된 산출물 및 바라보는 관점이 상이하다는 문제가 발생 될 수 있다는 것이다. 따라서 국내 철도 분야에서 일관된 관점으로 접근하기 위해서 통일된 위험도 평가 기준에 대해 보급이 시급한 실정이다. 또한, 최근 국내 철도 시장은 한국의 철도관련 기술력의 선진화에 따라, 해외 수출로 이어지고 있다는 점에서, 해외 시장에서 요구하는 해외 표준 기준에 부합해야 할 것이다. 이 또한 상당 부분이 인증과 관련하여 연관이 있을 것으로 사료된다.

국내 철도 분야도 국제 표준(International Standards)을 준수하기 위해 2008년 1월부터 국내에서 시행중인 철도차량 안전기준에 관한 지침에서도 위험도 평가에 대한 내용을 담고 있다. 위험도 평가 기준을 정의 및 할당하는데 있어서, 발주자와 공급자간에 일관된 관점을 반영한 결과를 도출하기 위해서는 안전 관리 시스템(Safety Management System)을 기반으로 수행되어야 한다고 언급하고 있다. 하지만 현실적으로 국내의 영세한 업체에서 철도 표준에 대한 이해를 기반으로 수행하기에는 상당한 어려움이 존재한다. 따라서 2008년 시행된 철도차량 안전기준에 관한 지침은 국제 표준을 준수하며 위험도 평가 기준이 통일되어 국내에서 엄격히 준수할 수 있도록 요구해야 할 것이다.

위험도 평가 기준은 안전성 분석에서 요구하는 여러 기법과 매우 밀접하게 연관되어 있다. 이는 안전성 분석 기법이 정량적 평가를 요구하거나 혹은 정성적 평가를 요구한다는 점에서 평가 기준 자체가 달라지기 때문이다.

일반적으로 안전 공학(Safety Engineering)에서 널리 알려진 단순 공식인 위험도와 관련하여 “위험도(Risk) = 발생빈도 x 심각도”로 규정하고 있다. 공식에서도 살펴볼 수 있듯이 위험과 관련하여 보다 정량적인 정보를 통해 객관적으로 평가하고자 함을 알 수 있다. 위험도 평가에 대한 기준은 위험원의 연간 발생 빈도 즉, 다시 말해 한 해 동안 얼마나 자주 일어나는지에 대한 기준을 말한다. 또는 체계적으로 수집된 정량적 지표를 가지고 사고가 발생 가능한 확률을 계산하고 추정하는 과정이라고 볼 수 있다. 반면 정성적 평가는 위험원의 발생빈도와 심각도를 일정한 카테고리별로 등급화하고 위험원의 발생빈도와 심각도를 조합한 위험도 등급을 구한 후 이를 허용 등급과 비교하여 평가하는 방법이다.

정량적 평가의 경우 방대한 양의 통계적 산출물과 수학적 접근법을 요구하기 때문에 객관화된 접근으로 적용 시 국내 현실에 많은 어려움이 따를 것이라 예측된다. 따라서 국내 철도 분야의 특성상 개발되는 철도 시스템에서 모든 위험원에 대해 적용하기에는 한계성을 지니고 있다. 반면, 정성적 평가의 경우 실제 수행하는 엔지니어의 전문성을 바탕으로 개략적으로 예측 가능하다는 용이함이 있다. 다만, 정성적 평가는 수집된 자료를 기반으로 안전성 분석을 수행하기 때문에 약간의 부정확성이 존재하는 것이 단점이다.

이로 인해, 대부분의 철도 분야의 국외 추세에서는 철도 시스템에 대한 위험도를 평가하고 진단하는데 있어서 정성적 방법 또는 준 정량적 접근에 기반하여 적용 하고 있는 실정이다. 다만 보다 안전성이 중요 및 판단이 요구되는 화재와 탈선과 같은 인적/물적 상당한 손실을 초래할 수 있는 사고에 한해 필요 시 정량적 평가 방법이 적용되고 있다. 국내 철도 분야에서는 그동안 사고에 대하여 체계적인 통계 산출에 필요한 누적 데이터의 관리에 있어서 한계성을 띄고 있다는 점에서 정성적 또는 준 정량적인 평가 기법을 적용하는 것이 대안이 될 수 있다.

안전성 분석을 수행하는데 있어서, 위험도 매트릭스는 정성적 평가의 수행을 위한 필수적 도구이다. 물론, 위험도 매트릭스의 형태는 다양한 형태의 기준 표를 기반으로 위험원의 발생빈도와 심각도 카테고리의 두 요소를 정성적, 준 정량적, 정량적으로 표현하여 조합하는 방안이 가장 타당한 방안이 될 수 있다.

시스템 안전성 분석 가이드에서 적용하고 있는 표준은 다음과 같이 정성적 위험도 평가 매트릭스가 제시되고 있다. 철도 분야에서의 기준이 되는 위험원의 위험도는 [표 34]로부터 [표 36]까지의 위험도 평가를 위한 요소(발생빈도, 심각도, 허용수준)를 기준으로 식별된 위험원은 위험도 매트릭스의 허용(Tolerable) 및 무시(Negligible) 수준으로 제어되어야 하며 관련 기준은 아래와 같다. (위험원으로 인해 발생할 수 있는 잠재적인 사고에 대한 심각도 수준은 5개의 등급으로 분류되며 관련한 심각도 기준은 아래와 같다.)

표 34 위험도 허용수준의 정성적 심각도(Severity) 등급 [7]

심 각 도	등급	정성적 기준	정량적 기준 예시
치명적인 위험 (Catastrophic)	A	인명의 사망, 시스템의 손실 또는 심각한 환경상의 피해를 유발하는 위험	3인 이상 사망
중대한 위험 (Critical)	B	심각한 인명의 상해, 직업상의 질병 및 중요한 시스템 또는 환경상의 피해를 초래하는 위험	1인 이상 사망, 3인 미만 사망
중요하지 않은 위험(Marginal)	C	최소한의 상해, 작업상의 질병 및 최소한의 시스템 또는 환경상의 피해를 초래하는 위험	1인 이상 중상, 10인 미만 중상
사소한 위험 (Insignificant)	D	최소한의 상해, 직업상의 질병보다 작고, 최소한의 시스템 및 환경상의 피해보다 작은 영향을 초래하는 위험	1인 이상 경상, 20인 미만 중상
신뢰성관련	R	인명이나 환경 상에 피해를 발생하지 않으나 경제적 손실을 동반하는 위험	유지보수필요

표 35 위험도 허용수준의 정성적 발생빈도(Frequency) 등급 [7]

발생 빈도	등급	설명	정량적 기준 (위험측 고장률, 단위시간당 발생확률)
빈번한 발생 (Frequent)	1	생명주기 동안 빈번하게 발생할 가능성이 있음	10^{-3} 미만
가능성 있는 발생(Probable)	2	생명주기 동안 여러 번 발생할 가능성이 있음	$10^{-4} < to \leq 10^{-3}$
종종 발생 가능(Occasional)	3	생명주기 동안 가끔 발생할 가능성이 있음	$10^{-6} < to \leq 10^{-4}$
발생가능성이 미약함(Remote)	4	생명주기 동안 한두 차례 발생할 가능성이 있음	$10^{-8} < to \leq 10^{-6}$
발생 가능성이 거의 없음(Improbable)	5	생명주기 동안 발생 가능성은 있지만, 발생하지 않음	$10^{-9} < to \leq 10^{-8}$
발생 가능성이 전혀 없음(Incredible)	6	발생가능성도 희박하며, 절대 발생하지 않음	10^{-9} 이하

심각도 분류와 발생빈도 분류에 의해 위험도를 허용할 수 있는 수준을 표준을 근거로 매트릭스 형태로 제시하고 매트릭스는 위험도를 정성적으로 판단하는 경우에 사용된다.

표 36 철도분야 위험도 허용수준 [7]

발생 빈도	심 각 도			
빈번한 발생 (Frequent)	Intolerable	Intolerable	Intolerable	Undesirable
가능성 있는 발생(Probable)	Intolerable	Intolerable	Undesirable	Tolerable
종종 발생 가능(Occasional)	Intolerable	Undesirable	Undesirable	Tolerable
발생가능성이 미약함(Remote)	Undesirable	Undesirable	Tolerable	Negligible
발생 가능성이 거의 없음(Improbable)	Tolerable	Tolerable	Negligible	Negligible
발생 가능성이 전혀 없음(Incredible)	Negligible	Negligible	Negligible	Negligible
	Catastrophic	Critical	Marginal	Insignificant
	Severity levels of hazard consequence			

표 37 위험도 평가 및 허용수준의 정의 [7]

위험도 구간		위험도 평가 조건의 정의
Intolerable (허용 불가능한)		반드시 제거되어야 함.
ALARP 구간	Undesirable (바람직하지 않은)	위험도 경감이 현실적으로 가능하지 않을 경우, 반드시 운영기관의 동의가 있을 경우에만 허용할 수 있음.
	Tolerable (허용 가능한)	적절한 위험도 관리를 조건으로 운영기관의 동의를 얻는 경우 허용 가능함.
Negligible (무시할 수 있는)		운영기관의 동의가 없어도 허용 가능함.

위험도 평가 시 위험도 수준에 따른 위험도 저감 또는 안전 요구사항 및 허용여부에 관한 기준은 ALARP(As Low As Reasonably Practicable)의 원칙을 적용하며 다음과 같다.

표 38 위험원 결과심각도 [7]

결과심각도	의 미
Catastrophic (재난 수준의)	수명의 사망자 그리고 / 또는 수명의 중상자 그리고 / 또는 환경에 대한 대규모 피해
Critical (매우 심각한)	1명의 사망자 그리고 / 또는 중상 그리고 / 또는 환경에 대한 중대한 피해
Marginal (다소 심각한)	경상 발생 그리고 / 또는 환경에 대한 중대한 위협
Insignificant (경미한)	경상 발생 가능

다음은 위험원의 발생빈도와 심각도를 등급으로 표시하고 그 곱을 숫자로 표현하는 형식의 위험도 매트릭스 이다. 이는 상세한 분석이 필요한 위험원을 사전에 구분하기 위해 보통 사용되며, 주로, 영국의 Network Rail 및 철도운영회사(TOC)들이 널리 사용하는 형태이다.

표 39 준-정량적 위험도 매트릭스

잠재적인 인명피해 / 금전손실의 심각도							
			5	4	3	2	1
인명피해			다중사망	1인 사망	중상	중상	경상
영업적 손실			재난수준	매우 심각	심각한	경미	매우 경미
환경적 손실			재난수준/장기적	매우 심각 / 중기적	심각/단기적	경미	매우 경미
빈도	매주- 매월 발생	5	25	20	15	10	5
	매월 또는 연간	4	20	16	12	8	4
	1-5년마다	3	15	12	9	6	3
	5-10년마다	2	10	8	6	4	2
	>10년 이상에 1회	1	5	4	3	2	1

[표 39]는 앞서 제시된 [표 36]과는 다른 형태의 위험도 매트릭스 이다. R1은 허용 가능하지 않은(Unacceptable) 수준을 나타낸다. R4의 경우는, 특별한 조치를 필요로 하지 않는 매우 경미한 위험도(Broadly Acceptance) 수준을 나타낸다. R2와 R3는 ALARP 원칙을 사용하여 추가적인 조치의 필요 여부를 결정해야 하는 ALARP 영역을 나타낸다. 이는 주로 호주, 홍콩에서 주로 사용하는 형태의 매트릭스이다.

국내 철도 분야에서는 철도 사고와 관련해, ‘철도사고 등의 보고에 관한 지침’을 통해, 사고구분, 사상자 구분, 보고기준을 다음과 같이 정의하고 있다. 특히, 운행지연의 기준을 고속열차와 전동차의 경우, 10분 이상, 객차의 경우 20분으로 규정하고 있으며, 사상자와 열차 지연 시간을 통해 사고의 규모를 구분한다.

표 40 기관별 사고보고 기준 [56]

보고 조직	기 준
철도 사고 조사위원회	1) 1인 이상의 승객 또는 승무원 사망사고 2) 2인 이상 사상 또는 3천만 원 이상의 재산 3) 사상자 발생 또는 5천만 원 이상의 재산 피해 4) 1시간 이상의 운행지연이 발생한 경우
국토교통부	3인 이상의 사상 또는 5천만 원 이상의 재산 피해 발생

표 41 상해의 기준 [56]

구 분	기 준
경 상 자	1일 이상 3주 미안의 치료를 요하는 부상을 입은 자.
중 상 자	3주 이상의 입원 치료를 요하는 상해를 입은 자 또는 부분적인 상실 혹은 기능을 영구적으로 상실한 자.
사 망 자	사상자 및 부상 후 그 부상에 기인하여 72시간 이내 사망한 자

표 42 사고의 종류 [56]

기 준	대형사고	중형사고	경미사고
사 상 자	다수	5인 이상	경미
열차지연	24시간 이상	3시간 이상	10분미만
기 타	1) 인명과 재산 피해 규모가 매우 큰 사고일 경우 20 사회적 큰 물의가 예상되는 사고일 경우	1) 사회적 물의가 예상되는 사고	해당없음.

철도 시스템의 안전성 분석을 수행하기 위한 안전성 분석기법은 안전 무결성 등급(SIL 1~4)별로 [표 43]과 같이 복수로 사용될 수 있다. 이러한 안전성 분석 및 위험도 평가를 위한 각 방법들은 안전계획서에 제시된 방법에 따라 안전성 분석이 수행되어야 하며 입증 자료 또한 제시되어야 한다.

표 43 시스템의 안전성 분석 방법 [10, 57]

기법/수단	SIL1	SIL2	SIL3	SIL4
PHA	HR	HR	HR	HR
FTA	R	R	HR	HR
FMECA(FMEA)	R	R	HR	HR
HAZOP	R	R	HR	HR
ETA	R	R	R	R
IHA	R	R	HR	HR

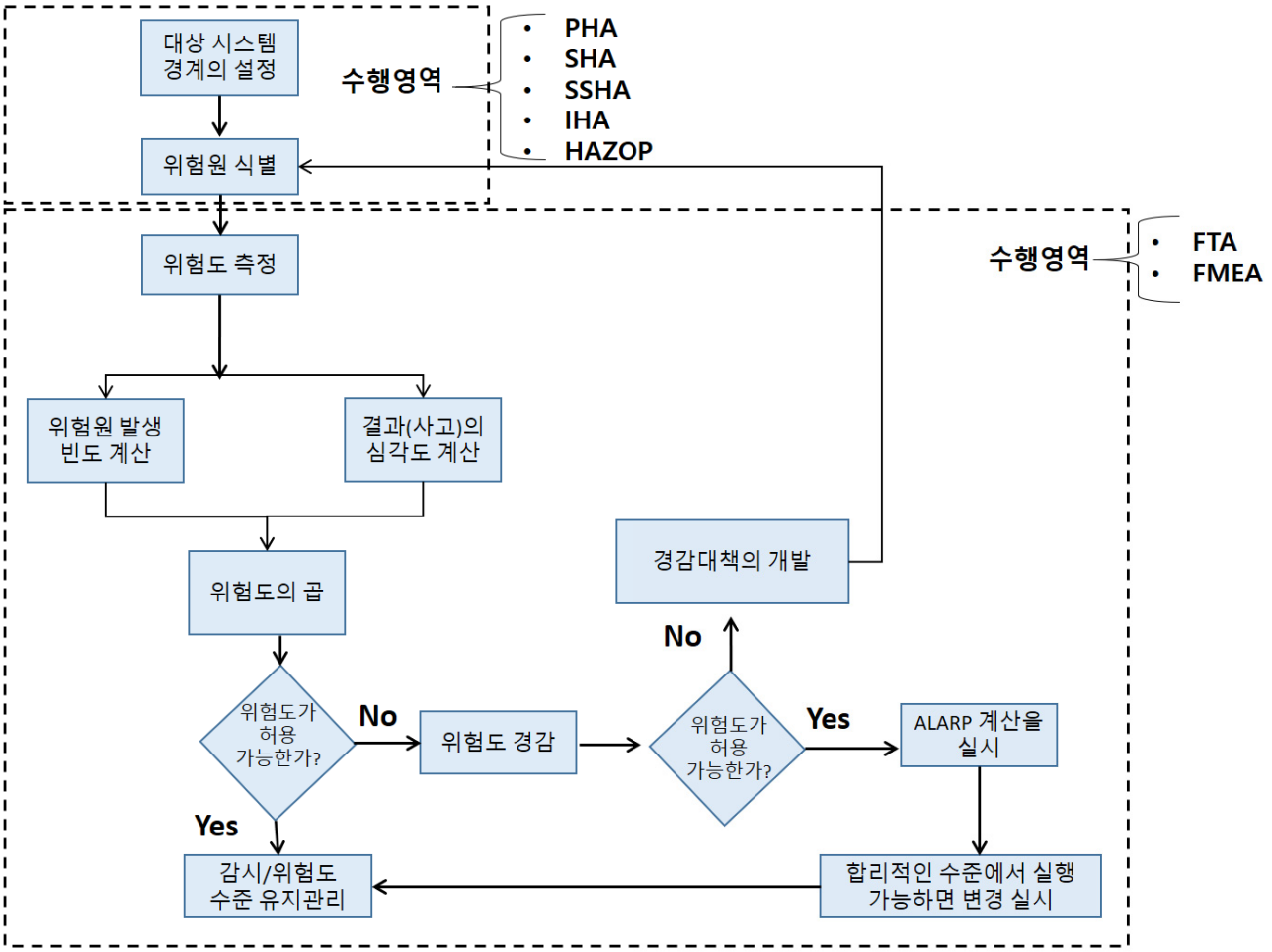
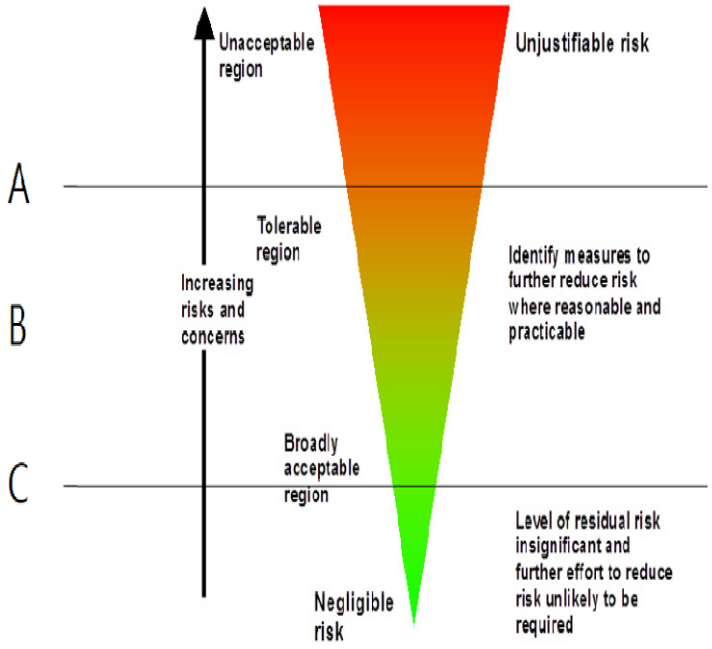


그림 54 위험도 평가 절차

본 시스템 안전성 분석 가이드에서 제공하는 안전성 분석 기법을 기반으로 한 활동은 [그림 54]의 위험도 평가 절차 모식도의 두 가지 영역 중에 하나에 해당이 된다. 좌-상단에 위치한 영역은 위험원을 찾기 위한 분석적 활동으로 위험원 정의 및 분석 활동에 해당되는 영역이다. 아래의 영역은 앞서 분석 및 정의 과정을 거쳐 식별된 위험요소 및 위험 원인을 찾아 이들이 얼마만큼 영향을 미치는지 평가하는 단계이다.

	심각도	사소한 위험요인	중대하지 않은 위험요인	중대한 위험요인	치명적인 위험요인
발생빈도	등급	IV	III	II	I
빈번한 발생	A	U	I	I	I
가능성 있는 발생	B	T	U	I	I
종종 발생	C	T	U	U	I
발생가능성이 미약	D	N	T	U	U
발생가능성이 거의 없음	E	N	N	T	T
발생가능성이 존재할 수 없음	F	N	N	N	N



발생빈도수준	분류	고장 발생 빈도 정의
빈번한 발생	A	수차례 발생 가능 (일 기준)
가능성 있는 발생	B	수차례 발생 가능 (월 기준)
종종 발생	C	수차례 발생 가능 (년 기준)
발생가능성이 미약	D	수차례 발생 가능 (시스템 수명 주기)
발생가능성이 거의 없음	E	거의 발생하지 않음
발생가능성이 존재할 수 없음	F	발생하지 않음

심각도 수준	분류	사람이나 환경에 미치는 결과
치명적인 위험요인	I	환경 재해 혹은 다중의 심각한 손상 혹은 중대한 손상
중대한 위험요인	I	국지적 환경 재해 혹은 심각한 손상 혹은 중대한 손상
중대하지 않은 위험요인	III	환경의 경미한 손상 혹은 심각한 위협
사소한 위험요인	IV	일어날 가능성이 적은 경미한 손상

그림 55 위험도 매트릭스와 ALARP [7]

위험도 평가 절차 과정을 거쳐 개별 위험원에 대해서 위험도가 허용 가능한 위험원에 대해서 ALARP 계산을 실시하게 된다. ALARP 계산은 위험원에 대한 발생빈도와 심각도 평가를 수행해야한다. 심각도와 발생빈도의 평가는 [그림 55]에서 제공하는 바와 같이 참조 기준 표를 활용하여 평가 수행하게 된다. 개별적인 평가 결과를 종합반영하기 위해, [그림 55] 좌-상단에 위치한 복합(빈도-위험요인) 기준표를 참조로 ALARP에 대한 최종 평가를 통해 위험도 평가를 수행하게 된다.

6. 안전성 분석을 위한 주요 위험원 분석기법

6.1. 안전성 분석기법간 절차

철도차량 안전 지침 기준(2008)에 따르면 본 시스템 안전성 분석 가이드에서 다루는 안전성 분석 기법에 대해서 전담하여 수행하는 조직(기관)을 명시하고 있다. [그림 56]과 같이 운영기관, 차량제작기관, 성능시험기관, 제작/감독기관 이렇게 총 4개 조직(기관)을 통해서 안전성 분석을 수행하도록 지침으로 규정하고 있다. 그리고 본 시스템 안전성 분석 가이드는 철도 분야 특성을 반영하기 위해 거시적으로 철도차량 안전 지침 기준(2008)을 따르고 있다. 또한 상세 안전성 분석 기법에 대해서 수행 목적 및 사용자의 편의를 제공하기 위해 [그림 56]의 우측 그림과 같이, 2가지 활용 대안을 제시한다.

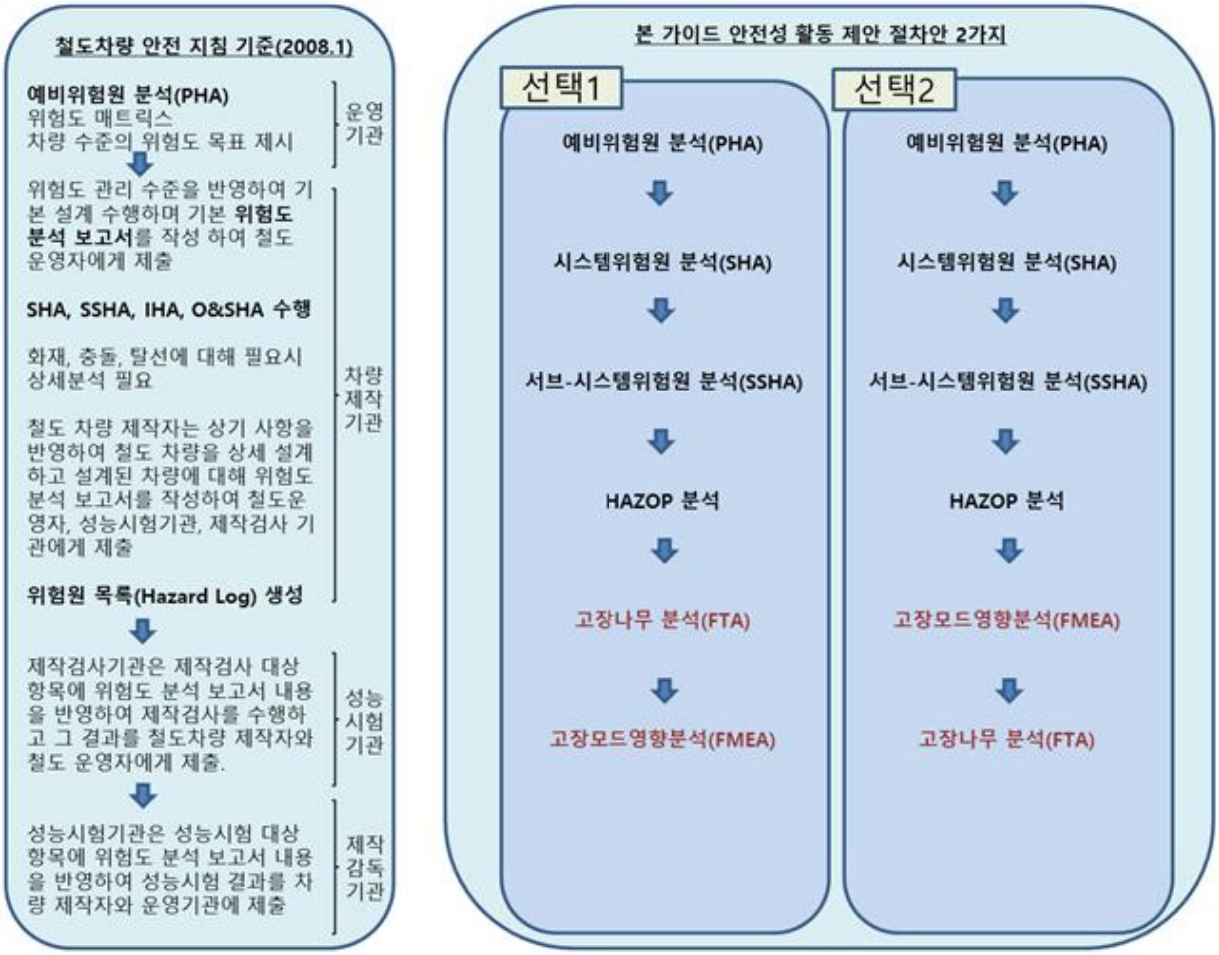


그림 56 철도차량 안전지침 및 안전성 분석 절차

제시된 활용 대안은 FTA 또는 FMEA 중 어느 안전성 분석 기법을 먼저 수행하는지에 대한 차이이다. 본 시스템 안전성 분석 가이드에서 사용자 및 조직에 판단 근거를 제공하고자 한다. [표 44]에서 제공하는 기준을 바탕으로 체크가 많이 된 기법을 중심으로 사용자는 해당 기법을 우선으로 사용하는 방안에 대해서 제시하고자 한다.

표 44 FMEA와 FTA 기법 우선수행 선정표

안전성 활동 기법	식별 번호	점검 항목	점검항목 해당여부 (O,X)	
			FMEA	FTA
1.FMEA	1	부품의 고장모드가 시스템이나 서브시스템에 어떤 영향을 주는지 아는가?		
	1.1	부품의 고장모드를 검토 할 수 있는가? 또는, 부품의 고장모드를 식별한 산출물을 보유 하고 있는가?		
	1.2	고장이 발생하면 시스템이나 서브시스템의 동작에 어떠한 영향을 주는가를 분석할 수 있는가? 또는, 분석된 산출물을 보유하고 있는가?		
	1.3	시스템 및 서브시스템의 구성도 및 기능을 정의 할 수 있는가?		
	1.4	시스템 및 서브시스템이 지니고 있는 고장모드의 원인 및 영향에 대해 분석 및 식별할 수 있는가?		
	1.5	FMEA 수행을 통해, 시스템 또는 서브시스템의 고장에 대한 영향의 중요성이 파악된다면 설계 변경이 필요한 경우, 설계적 반영을 수행할 수 있는가?		
2. FTA	2	시스템이나 기기에 발생하는 고장이나 결함의 원인을 알고 있는가?		
	2.1	정상사상(Top-Event)을 일으키는 원인인 기본사상(Basic-Event)를 파악 할 수 있는가? 또는 관련 산출물을 보유하고 있는가?		
	2.2	시스템 또는 서브시스템에 대한 고장이나 원치 않는 사항을 정의 할 수 있는가?		
	2.3	시스템 또는 서브시스템에 대한 시스템의 작동과 환경을 분석 수행 가능한가?		
	2.4	원치 않는 사상(Event)의 발생 원인과 인과관계를 분석 또는 파악 가능한가?		
	2.5	정상 사상의 원인이 되는 상위사상을 찾아내고 그들의 인과관계를 분석 및 파악 가능한가?		
	2.6	개별 사상(Event)에 대한 발생확률을 정의 할 수 있는가? 또는 관련 산출물을 보유 하고 있는가?		
	2.7	병렬 또는 중복 결함의 선택적인 결함 경로에 대해 다룰 수 있는가?		
		선정된 우선 기법		

이 밖에, 사용자 및 조직은 FMEA와 FTA의 개별 안전성 분석 기법이 지니고 있는 수행 목적 및 의도에 따라, 선택을 위한 근거를 다음과 같이 제시 하고자 한다.

FTA 수행 후에 FMEA를 수행하는 방법은 먼저 FTA를 통해 최상위 사건(Top Event)인 시스템의 고장을 결과로 선정하여 하향식(Top-Down) 방식으로 고장 원인을 파악할 수 있다. 이후 FMEA를 통해 고장 원인(Cause)의 1차와 2차에 걸쳐 저감대책에 따라 안전성을 확보할 수 있다. 즉, 소프트웨어의 쉽게 파악할 수 없는 복합적 고장 및 결함을 파악하기 위한 측면에서 접근 시 해당 절차가 적합하다.

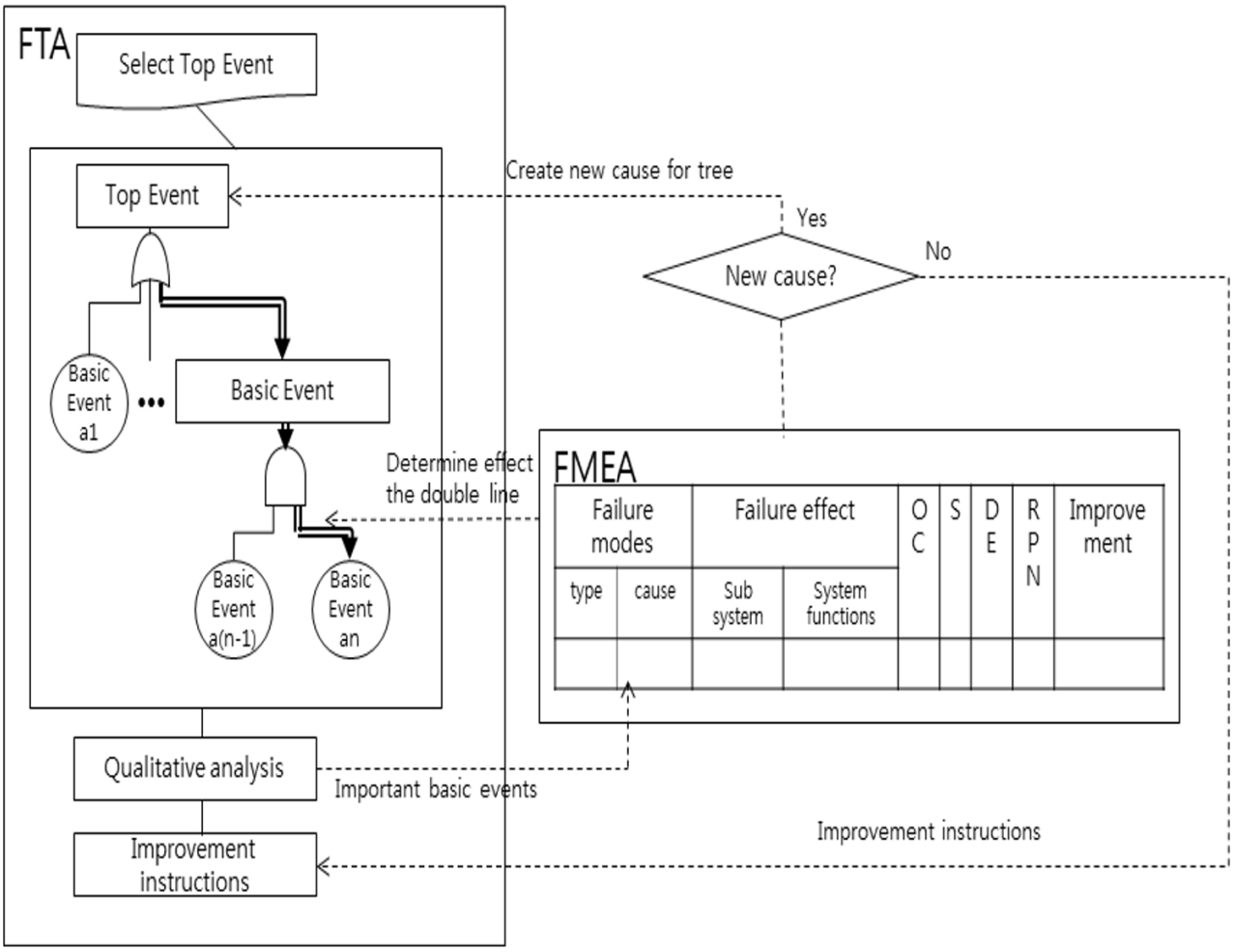


그림 57 FTA 우선 수행의 경우 FMEA 연계 [58]

[그림 57]은 FTA를 중심으로 FMEA의 후속 수행을 통해 보조하는 절차이다. 먼저 Fault Tree(FT)를 만들기 위한 최상위 사건을 선택한다. 만약 FT에서 부정확성이 나타나거나 고장 영향(Failure effect)의 심각도에 따라 최상위 사건이 나타난다면 FT를 수정할 수 있다. 그리고 FMEA에 의해 심각도 분석이 이루어진 후에 최상위 사건의 발생 확률을 계산할 수 있다.

(1) 안전성 분석에 따른 최상위 사건 선택과 FT 생성

(2) Minimal Cutsets와 해당 사건들을 질적인 분석을 통해 확인

(3) 고장 모드에 의한 해당 사건의 FMEA를 형성

(4) FT 및 저감대책을 수정하여 새로운 FT를 만들고 다음 분석을 수행하기 위한 최상위 사건의 결함을 선택

FMEA 수행 후에 FTA를 수행하는 방법은 먼저 FMEA를 통해 시스템의 안전성 수치 판단의 정량적 분석이 주를 이루며 모든 고장 원인들에 대해서 분석을 수행한다. 이후, FTA를 통해 고장 원인 간 관계 파악을 통해 최상위 사건을 도출하여 이에 대한 저감대책에 따라 안전성을 확보할 수 있다. 즉, 하드웨어의 설계단계 혹은 공정단계에 있어서 변경점이나, 기술관리 측면에서 접근 시에 해당 절차가 적합하다.

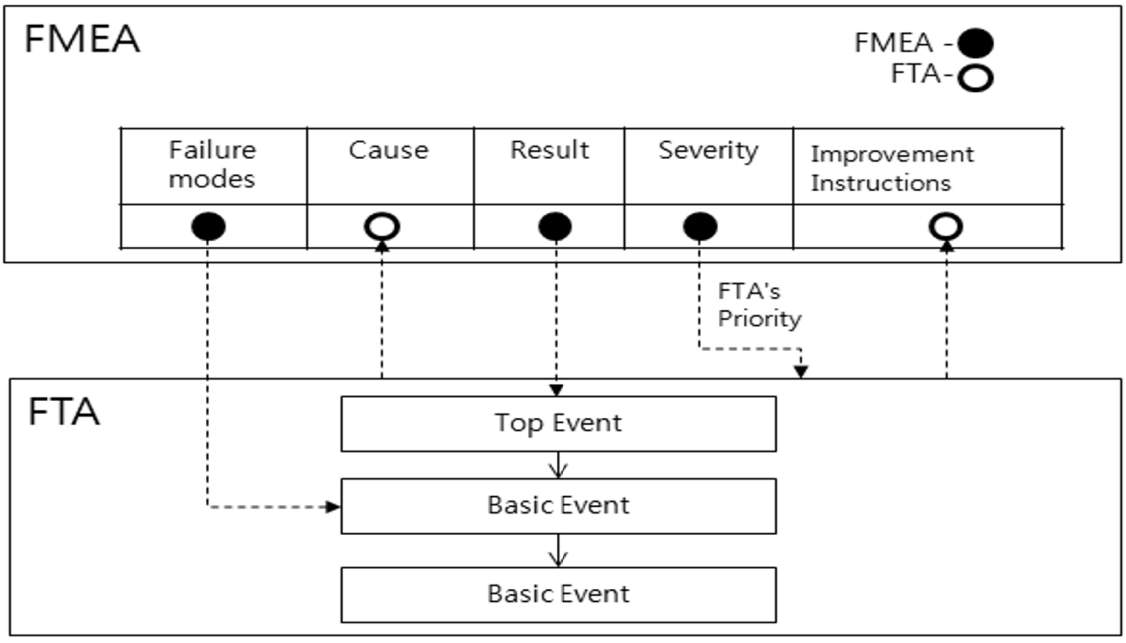


그림 58 FMEA 우선 수행의 경우 FTA 연계 [58]

[그림 58]은 FMEA를 중심으로 FTA의 후속 수행을 통해 보조하는 절차이다. FTA는 FMEA의 결과로부터 고장 영향의 심각도에 따라 형성된다. 이때 고장 영향을 더욱 철저하게 확인하기 위해 높은 심각도의 영향을 최상위 사건으로 사용한다. 고장 모드(Failure modes)는 FMEA로 표현하기 어려운 고장 모드의 원인을 확인하기 위해 FTA의 중간 사건으로 사용된다.

(1) 기능적이거나 구조적인 분석 단계나 사항을 선택

(2) 각 고장에 대한 영향 평가

(3) FTA를 수행하기 위해서 최상위 사건에 해당하는 높은 심각도의 영향 선택 또는 중간 사건에 해당하는 고장 모드를 선택

(4) FTA 결과로부터 고장 모드의 원인들과 논리 관계에 따른 저감대책 선택

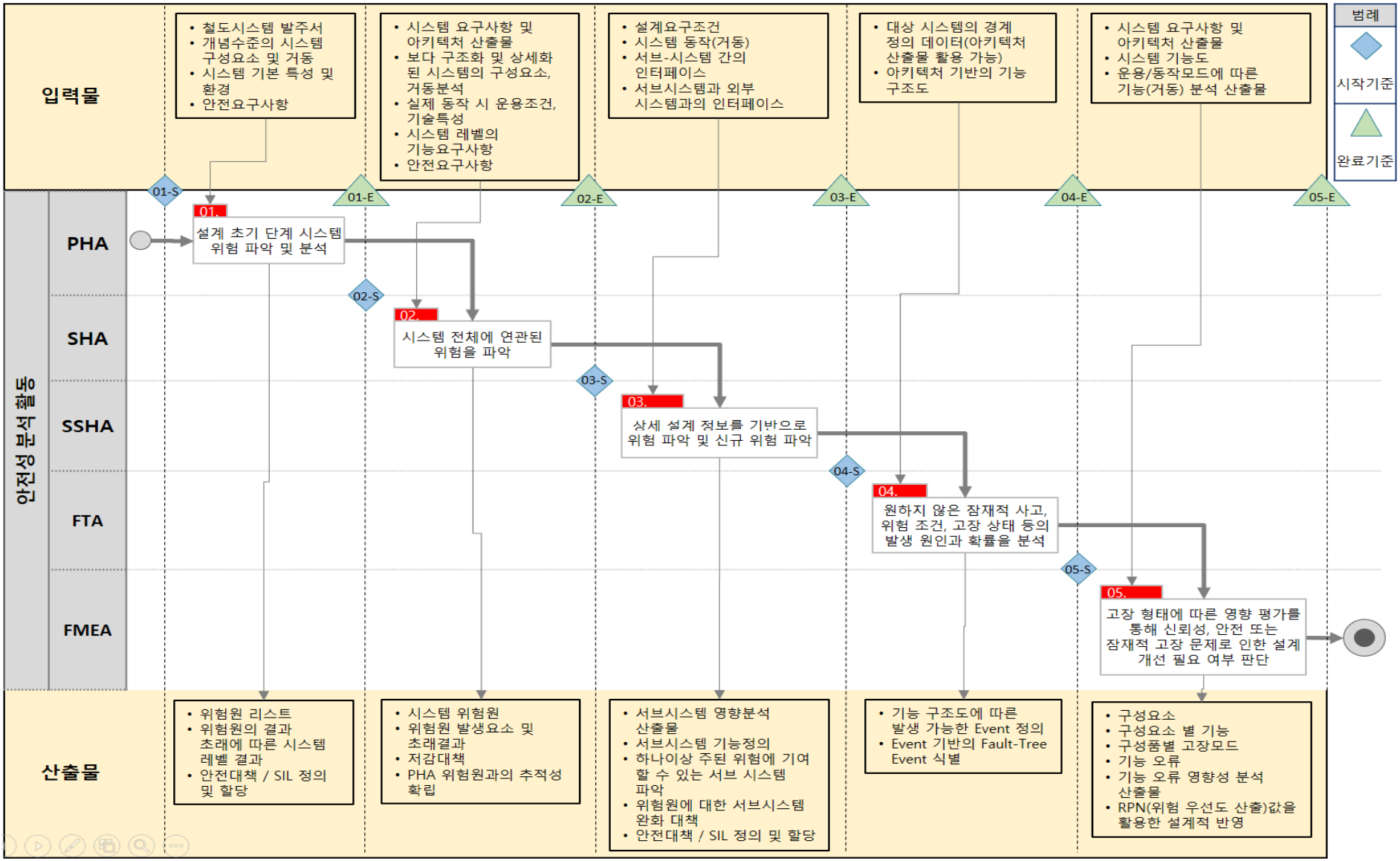


그림 59 안전성 분석 개요

ID	활동	설명	입력물	시작 기준		산출물	완료 기준	
01	PHA	시스템에 관한 추상적인 주요 위험원 및 사고 초래 요인 식별	▪ 철도시스템 발주서 ▪ 개념수준의 시스템 구성요소 및 거동 ▪ 시스템 기본 특성 및 환경 ▪ 안전요구사항	01-S	▪ 철도시스템 발주서 확정 ▪ 초기 개발 시스템의 개념 확립 ▪ 시스템 정의 및 요구조건 단계에서 수행	▪ PHA 수행 결과 보고서	01-E	▪ 초기 위험원 수정조치 방법 수립 ▪ 초기 위험원 안전 대책 수립
02	SHA	시스템에 관한 최상위 기능으로부터 발생 가능한 위험원 식별 및 초래 결과 식별	▪ 시스템 요구사항 및 아키텍처 산출물 ▪ 시스템수준의 시스템 구성요소 및 거동 ▪ 실제 동작 시 운용조건 및 기술특성 ▪ 시스템수준의 기능 요구사항 ▪ 안전요구사항	02-S	▪ 시스템 요구사항 확정 ▪ 시스템 아키텍처 수립 ▪ 시스템 요구사항 및 아키텍처를 통해 시스템수준의 기능 식별	▪ SHA 수행 결과 보고서	02-E	▪ 시스템수준의 최상위 기능 식별 ▪ PHA 위험원과 추적성을 통해 시스템수준의 위험원 수정조치 방법 수립 ▪ PHA 위험원과 추적성을 통해 시스템수준의 위험원 안전 대책 수립
03	SSHA	서브시스템간 시스템 기능 구분 또는 할당을 확인, 분석하여 위험요소를 초래할 수 있는 왜곡 및 잠재적 누락 요건 등을 파악	▪ 설계요구조건 ▪ 시스템 동작(거동) ▪ 서브시스템 간의 인터페이스 ▪ 서브시스템과 외부 시스템과의 인터페이스	03-S	▪ 시스템 요구사항 및 아키텍처를 통해 서브시스템수준의 기능 식별 ▪ 서브시스템 간의 인터페이스 식별 ▪ 서브시스템과 외부 시스템과의 인터페이스 식별	▪ SSHA 수행 결과 보고서	03-E	▪ 서브시스템수준의 기능 식별 ▪ 서브시스템간 인터페이스의 영향을 서브시스템수준에서 분석 ▪ 서브시스템과 외부 시스템과의 인터페이스 영향을 서브시스템수준에서 분석 ▪ SHA 위험원과 추적성을 통해 서브시스템수준의 위험원 수정조치 방법 수립 ▪ SHA 위험원과 추적성을 통해 서브시스템수준의 위험원 안전 대책 수립
04	FTA	원하지 않은 사건이 어떻게 일어 나는지 원인 식별 수행	▪ 대상 시스템의 경계 정의 데이터(아키텍처 산출물 활용 가능) ▪ 아키텍처 기반의 기능 구조도	04-S	▪ 개발 시스템의 기능 식별 범위 및 아키텍처 확정 ▪ 식별된 기능간 상호작용에 관한 잠재적 고장 식별	▪ FTA 수행 결과 보고서	04-E	▪ 시스템의 기능을 식별가능한 수준까지 식별 ▪ SSHA 위험원과 추적성을 통해 Event 기반의 위험원 수정조치 방법 수립 ▪ SSHA 위험원과 추적성을 통해 Event 기반의 위험원 안전 대책 수립
05	FMEA	시스템에 속해 있는 장치의 물리적 구성이 지니고 있는 기능 요구사항을 바탕으로 구성품이 고장 났을 경우, 어떠한 영향이 미치는지에 대해서 분석을 수행	▪ 시스템 요구사항 및 아키텍처 산출물 ▪ 시스템 기능도 ▪ 운용/동작모드에 따른 기능(거동) 분석 산출물	05-S	▪ 시스템 아키텍처를 통해 기능 및 구성요소 구조 식별 ▪ 기능 및 구성요소의 운용/동작모드에 따른 기능 거동 확정	▪ FMEA 수행 결과 보고서	05-E	▪ 기능 및 구성요소 구조 식별 ▪ 기능 및 구성요소의 운용/동작모드에 따른 기능 거동 식별 ▪ Fault-Tree Event와 추적성을 통해 식별된 RPN 값이 높은 요소에 대한 위험원 안전 대책 수립

그림 60 안전성 분석 산출물 및 시작, 완료 기준

6.2. 예비위험원분석(PHA)

6.2.1. 개요

PHA는 본격적인 위험원 분석을 수행하기 위한 준비단계에서 가장 우선적으로 수행되는 위험원 분석을 의미한다. PHA는 시스템 생명주기 중 개념설계 및 RAMS 목표 설정 단계에서 철도 시스템의 기능 요구사항을 바탕으로 간소화된 기능별 위험원을 도출한다. 따라서 PHA는 시스템 설계 안에 내재되어 있거나 관련되어 있는 위험원인, 위험상황, 사고 등을 시스템 설계 초기에 식별 및 제거해 내고자 하는 것이다. 구체적으로 시스템 내의 어디에 어떤 위험원이 존재하는가? 어느 정도의 위험 상태에 있는가? 안전기준 및 시설의 수준은 어떠한가를 정성적으로 평가한다. 그리고 정성적인 위험도 평가를 통해 도출된 예비위험원은 저감대책 수립에 의해 허용할 수 있는 수준으로 위험도를 제어해야한다. 이와 같은 과정을 통하여 철도 시스템의 주요 위험원은 초기 단계부터 제거, 최소화 혹은 통제될 수 있게 된다.

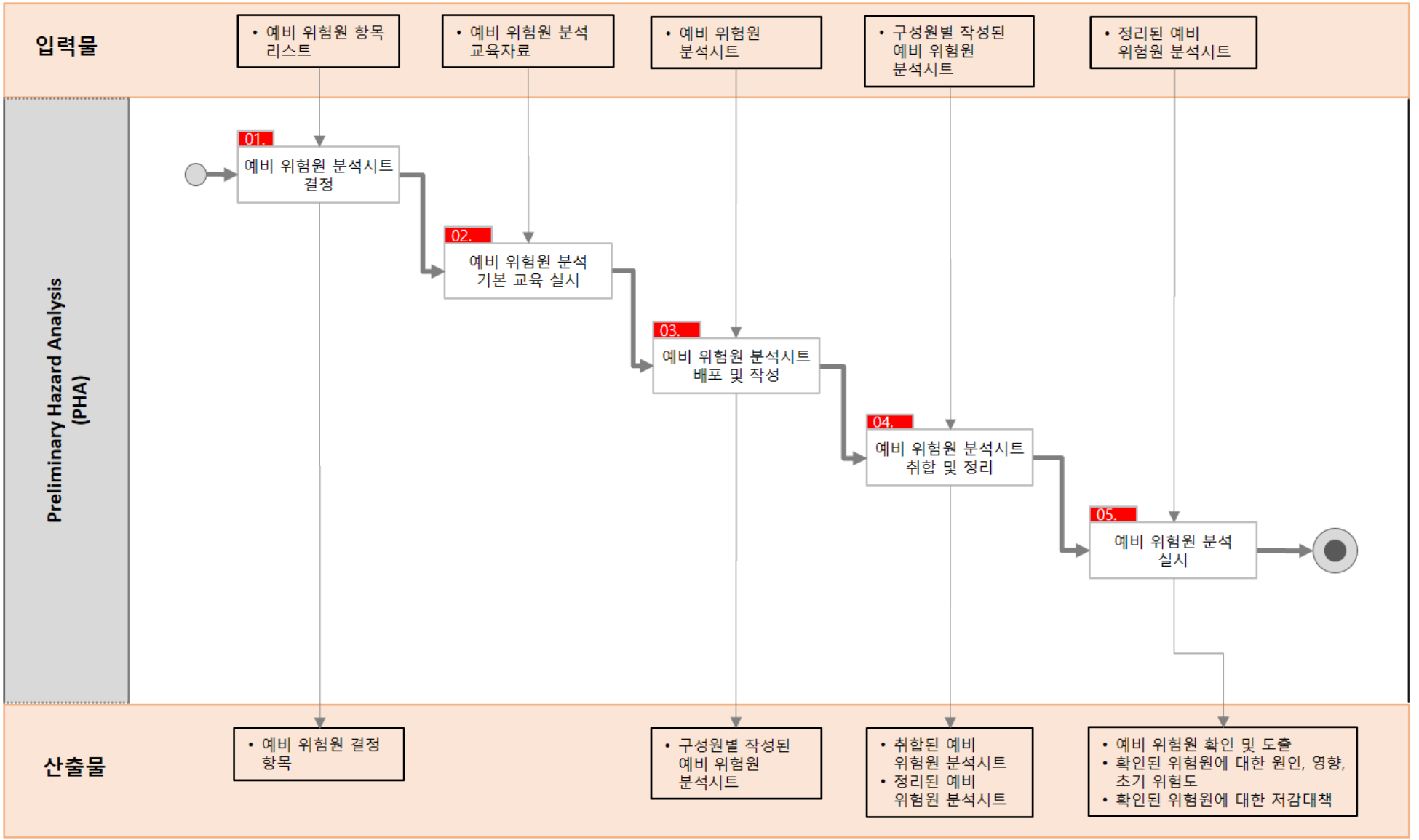


그림 61 PHA 활동 흐름도

6.2.2. 절차

PHA에 대한 활동 흐름도는 [그림 61]과 같다. 활동 흐름도에는 PHA에 대한 수행절차와 입・출력물을 명시하여 표기하였다. PHA는 시스템에 관한 모든 주요 위험원을 식별하여 포괄적으로 기술한다. 예비 위험원은 시스템 요구사항 명세서를 기반으로 생성된 아키텍처 산출물과 특히 기능요구사항을 기반으로 위험원을 식별하게 된다. 예비위험원은 시스템의 다양한 운용 모드에서 수행되어야 하는 기능에서 발생될 수 있는 오류를 기반으로 분석을 통해 식별된다. 이후 시스템의 물리적 장치가 지니고 있는 기능을 기반으로 오작동 또는 동작실패로 인해 초래될 수 있는 사고의 원인을 식별하게 된다. 그리고 PHA에서는 도출된 위험원에 대해 영향을 분석하여 위험원으로부터 초래할 수 있는 문제를 해결하기 위해서 저감대책을 수립하여 설계에 반영을 통해 적용될 수 있도록 한다.

다음은 PHA 과정을 나타낸다. PHA 과정은 크게 5 Steps로 나뉜다. 개별 Step에 대해서 살펴보면 다음과 같은 특성을 지닌다.

(1) 예비위험원분석시트 결정

안전 관리자는 PHA를 수행하기 위해 가장 적합한 예비위험원 항목을 결정해야 한다. 결정해야 할 항목은 아래와 같다.

표 45 예비위험원 결정 항목

- 위험원 No. - 위험원 - 위험원 설명 - 원인 - 영향 - 초기 위험도(심각도, 발생빈도, 위험도) - 대책 - 비고 - 책임

(2) 예비위험원분석 기본 교육 실시

안전 관리자는 해당 개발 구성원들에게 PHA를 수행하기 위한 기본적인 교육을 실시해야 한다. 교육의 내용은 다음과 같다.

표 46 교육 내용

1) 예비위험원분석 목적, 개요, 방법 2) 위험도를 규명하기 위한 위험원 발생빈도 및 심각도 3) 위험도 매트릭스

(3) 예비위험원 분석시트 배포 및 작성

안전 관리자는 예비위험원 분석시트를 구성원에게 배포한다. 구성원은 예비위험원 분석시트를 작성한다. (최소한 위험원과 원인에 대해 도출 가능하다).

(4) 예비위험원 분석시트 취합 및 정리

안전 관리자는 구성원이 작성한 예비위험원 분석시트를 취합한다. 안전 관리자는 취합된 예비위험원 분석시트에 기술된 위험원을 정리(시스템 수준과 동일 위험원별로 정리)한다.

(5) 예비위험원분석 실시

안전 관리자는 예비위험원 분석시트 내용을 최종 결정하기 위한 회의를 소집하고 진행한다. 회의 내용은 다음과 같다.

표 47 회의 내용

1) 정리된 예비위험원 분석시트를 기준으로 위험원 식별 및 도출 2) 식별된 위험원에 대한 원인, 영향, 초기 위험도 규명 3) 식별된 위험원에 대한 저감대책 수립

6.2.3. 체크리스트

PHA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 PHA에 대한 검증은 [표 48]을 통해서 수행된다.

표 48 PHA 기법 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
1. PHA 기법	1.1 개념수준의 시스템 컴포넌트를 포함하고 있는가?		철도 시스템 발주서	
	1.2 시스템 기본 특성 및 환경적 요소를 다루고 있는가?		철도 시스템 발주서	
	1.3 대상 시스템의 최상위 기능은 식별 되었는가?		PHA 수행 결과 보고서	
	1.4 개념적 수준에서 인터페이스 정보를 식별 하였는가?		철도 시스템 발주서	
	1.4 식별된 최상위 기능으로부터 초기 개념적 위험원 리스트를 도출 하였는가?		PHA 수행 결과 보고서	
	1.5 초기 위험원으로부터 유발되는 시스템 영향은 분석이 되었는가?		PHA 수행 결과 보고서	
	1.6 식별된 초기 위험원을 예방하기 위한 저감대책은 반영 되었는가?		PHA 수행 결과 보고서	

6.2.4. 적용 범위 및 제한조건

PHA를 통해 도출된 결과는 시스템 설계 초기에 안전성 확보를 위한 기초자료로 제공된다. 원활한 PHA를 수행하기 위해 관련된 모든 인원은 다음과 같은 사항을 준수해야 한다.

- RAMS 관리자는 PHA를 수행하기 위해 회의를 소집해야 한다.
- 관련된 각 팀의 팀장은 회의를 적극적으로 지원해야 한다.
- RAMS 팀은 예비위험원 분석시트를 준비하여 배포해야 한다.
- 상기 사항은 PHA, SHA, SSHA, IHA, O&SHA와 같은 안전성 분석 기법에 공통적으로 적용된다.

6.3. 시스템 위험원 분석(SHA)

6.3.1. 개요

초기 안전성 분석으로 PHA를 수행하였다면, 이후 시스템에 대한 보다 명확한 안전성 분석을 수행하기 위해서 SHA을 수행한다. SHA의 목적은 시스템에 대한 상세한 분석을 수행하여 시스템에서 위험을 초래할 수 있는 원인을 찾아 이에 대한 저감대책을 수립하는 것이다. 수립한 저감대책을 시스템 설계에 적극 반영하여 시스템의 안전성을 확보해야 한다. 앞서 제시한 [그림 60]을 통해서 제시한 매트릭스를 활용하여 PHA를 기반으로 SHA와 연계성을 확보할 수 있다. SHA에 대한 전체적인 흐름은 다음과 같다.

- 시스템의 시스템 기능을 정의한다.
- 시스템 기능을 기반으로 시스템 위험원을 식별한다.
- 시스템 위험원으로 초래할 수 있는 원인을 파악 한다.
- 시스템 위험원의 원인으로 인한 시스템 영향 분석을 수행한다.
- 시스템 위험원에 대한 저감 대책을 결정한다.

6.3.2. 절차

SHA, SSHA은 시스템 수준에 따른 차이를 지닌 기법이기 때문에 안전성 분석을 수행하는데 있어서 입·출력물에 대한 수준에 대한 차이만 존재한다. 따라서 수행 절차 및 입·출력물 전달에 대한 모식도를 [그림 62]을 통해 하나로 통합하여 표기 하였으며 상세한 내용은 SSHA에서 다룬다.

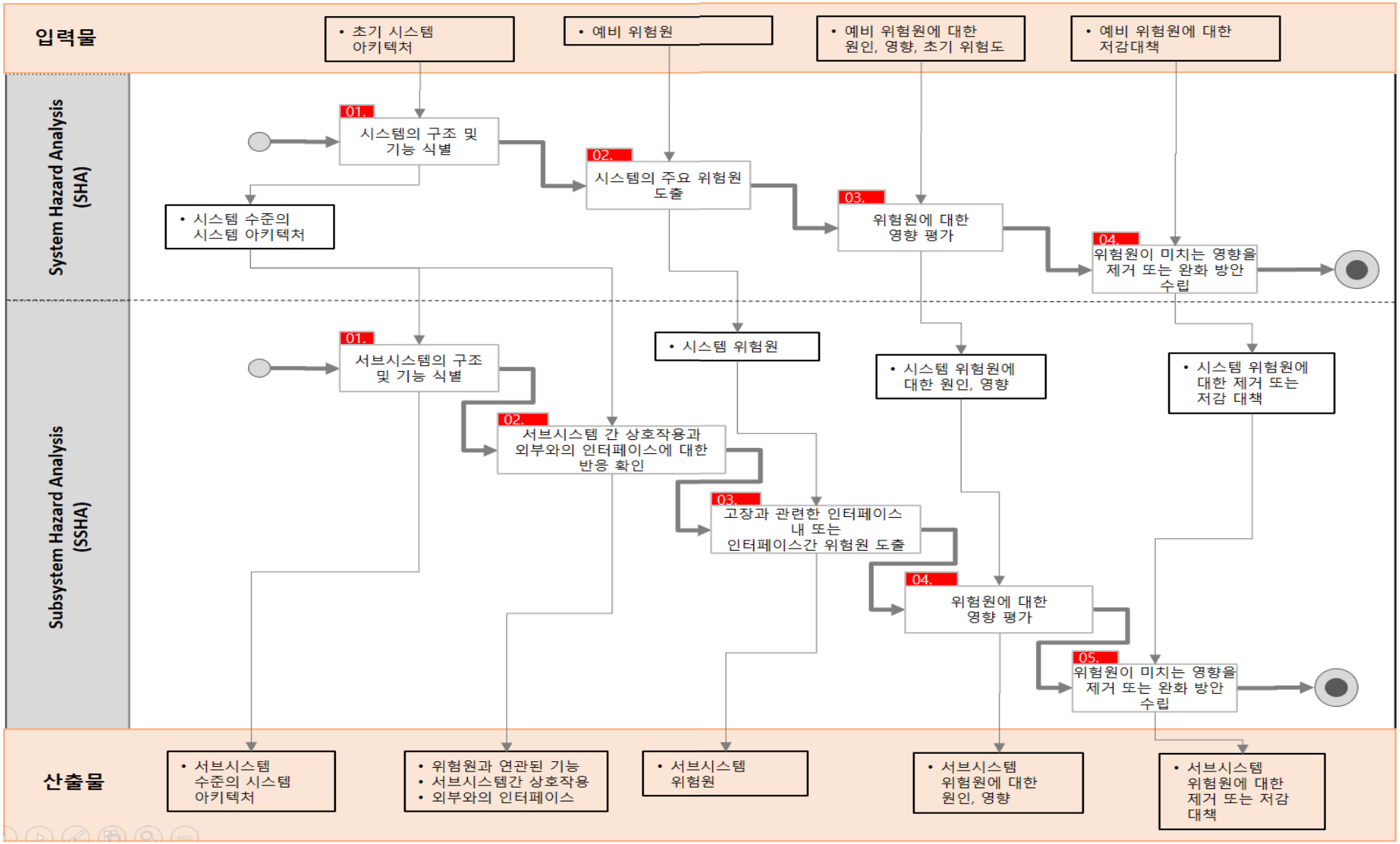


그림 62 SHA & SSHA 활동 흐름도

6.4. 서브시스템 위험원 분석(SSHA)

6.4.1. 절차

SSHA은 SHA의 수행결과의 연장선에서 보다 상세화 된 안전성 분석 기법으로 활용한다. 일반적으로 SSHA는 시스템의 각 서브시스템을 블랙박스로 간주하여 시스템을 구성하는 서브시스템 주변의 위험원들을 식별한다. 이러한 위험원 식별은 주로 서브시스템 간 가능한 모든 상호작용과 서브시스템 기능 책임에 준한 외부 사고에 대한 서브시스템의 반등 등에 초점을 맞춘다. 서브시스템간 기능 구분 또는 배분을 확인, 분석하여 위험원으로부터 초래할 수 있는 왜곡 및 잠재적 누락 요구사항 등을 파악한다. 안전성 분석을 통해 안전 메커니즘을 지정하여 가능한 고장과 관련한 인터페이스 내 또는 인터페이스 간 위험원을 감지, 완화한다. 이를 통해 서브시스템 안전 요구사항을 정의하게 된다. SSHA는 SHA에서 제시하고 있는 시스템 위험원 저감을 위해 시스템 내의 각 서브시스템의 역할과 책임사항을 확인, 결정해야 한다. SSHA를 통해, 서브시스템의 안전 무결성 등급을 안전 기능 구현을 위한 기능 책임에 준해 확인한다. 시스템 설계는 다음의 조건에서 안전한 것으로 판단한다.

- 서브시스템 간 기능 배정의 정확성은 확인 및 검증 가능한 근거를 통해 확인해야 한다.
- 서브시스템 상호작용 및 반응의 예상 동작을 통해 예측 가능한 불확실성이나 서브시스템 고장과 관련하여 파악된 위험원 조건을 완화할 수 있다.
- 서브시스템과의 인터페이스를 통한 외부에서 파악된 사고에 대한 예상 반응 정보를 제공한다.

또한 SSHA는 시스템 설계의 정확성을 확인하고 위험원의 발생 가능한 설계 내에 존재하는 결함을 파악하는데 목적이 있다. 기본적으로 시스템 내부 고장이 없음을 전제로 안전 요구사항을 반드시 정확히 구현해야 한다. 결함 감지를 위한 절차는 다음 항목을 고려해야 한다.

- 설계와 요구사항을 검토하여 설계가 의도하는 시스템 기능을 이해한다. 효과적인 위험원 식별과 시스템 설계 단계의 기술적 구현 실패를 방지하기 위해 안전 원칙을 지침으로 사용한다.
- 서브시스템들 간의 상호작용과 외부와의 인터페이스에 대한 반응을 확인하여 시스템 수준의 안전 요구사항으로 도출한다. 이러한 안전 요구사항을 아키텍처 및 서브시스템간 기능 할당 관련 설계에 정확히 반영하여 갭이나 예측 불가능한 불확실성 없이 안전 기능을 제공해야 한다.

- 제어 기능과 안전 제동 모델은 위험원 분석 시 위치와 속도의 불확실성을 고려해야 한다.
- 시스템에 다수 서브시스템에 따라 일정한 기능을 가지고 있을 경우, 데이터 목록 및 공통 표 내 여러 오류 들을 고려해야 한다.

6.4.2. 체크리스트

SHA & SSHA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 SHA & SSHA에 대한 검증은 [표 49]를 통해서 수행된다.

표 49 SHA & SSHA 기법 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
2. SHA 기법	2.1 시스템 수준의 구조 및 컴포넌트를 식별하였는가?		시스템 구조 아키텍처	
	2.2 시스템 수준의 기능을 식별하였는가?		시스템 기능 아키텍처	
	2.3 시스템 내/외부 요소간 인터페이스 정보를 식별하였는가?		인터페이스 통제 문서(ICD)	
	2.4 시스템 수준의 기능을 바탕으로 주요 위험원을 식별 하였는가?		SHA 수행 결과 보고서	
	2.5 식별된 최상위 기능으로부터 초기 개념적 위험원 리스트를 도출 하였는가?		SHA 수행 결과 보고서	
	2.6 식별된 시스템 수준의 위험원으로부터 유발되는 시스템 영향은 분석이 되었는가?		SHA 수행 결과 보고서	
3. SSHA 기법	3.1 식별되어 평가된 시스템 수준의 위험원을 예방하기 위한 저감 대책이 설계적으로 반영되기 위한 대책이 수립 되었는가?		SHA 수행 결과 보고서	
	3.2 서브시스템 수준의 구조 및 컴포넌트를 식별하였는가?		서브시스템 구조 아키텍처	
	3.3 서브시스템 수준의 기능을 식별하였는가?		서브시스템 기능 아키텍처	
	3.4 서브시스템 수준의 인터페이스 정보를 식별하였는가?		서브시스템 기능 인터페이스 통제문서	

	3.5 서브시스템 수준의 기능을 바탕으로 주요 위험원을 식별 하였는가?		SSHA 수행 결과 보고서	
	3.6 식별된 서브시스템 수준의 기능으로부터 보다 상세화된 서브시스템의 위험원 리스트를 도출 하였는가?		SSHA 수행 결과 보고서	
	3.7 식별된 서브시스템 수준의 위험원으로부터 유발되는 시스템 영향은 분석이 되었는가?		SSHA 수행 결과 보고서	
	3.8 식별되어 평가된 시스템 수준의 위험원에 대한 저감대책을 설계에 반영하기 위한 대책이 수립 되었는가?		SSHA 수행 결과 보고서	

6.4.3. 적용 범위 및 제한조건

시스템의 기능에 영향을 미치는 시스템 및 기능과 관련한 사항에 적용된다. 시스템 위험원 분석에서는 다음의 사항에 의해 야기된 위험은 고려되지 않는 가정 사항으로 정의한다.

- 시스템 경계를 벗어나 위치한 외부 시스템, 서브시스템, 컴포넌트
- 정의된 사양을 벗어나는 환경조건
- 자연적 대재난(홍수, 지진 등)
- 테러 및 의도적인 기물 파손

6.5. 인터페이스 위험원 분석(IHA)

6.5.1. 개요

IHA은 시스템과 관련된 내부 및 외부 인터페이스를 정의하고, 이러한 인터페이스 상에서 발생할 수 있는 위험을 분석하기 위해서 수행된다. 식별된 인터페이스 위험원을 제거 또는 위험도를 저감시키기 위한 내부 인터페이스 위험원에 대한 저감대책은 안전 요구사항으로 도출되어 설계에 적용한다. 외부 인터페이스 위험원은 외부 시스템 및 컴포넌트 설계를 책임지는 조직 및 실무자에게 제공하여 상호 협의 하에 저감대책이 수립되도록 한다. IHA는 다양한 서브시스템과 인터페이스에서 위험원을 유발할 수 있는 시스템영역을 표시하는데 사용된다. 또한 필요하다면 추가적인 인터페이스 조사를 실시해야 할 부분도 표시한다. SSHA와 유사하게 IHA는 각 위험원의 발생 가능성과 각 위험원의 심각도를 정량적으로 예측 가능하다는데 특징이 있다.

6.5.2. 절차

IHA을 위한 절차는 [그림 63]과 같으며 상세한 내용은 다음과 같다.

(1) 시스템 분석을 통해 도출한 시스템 아키텍처와 시스템 요구사항 분석을 통해 도출한 인터페이스 요구사항을 기반으로 시스템의 인터페이스를 분석한다. 인터페이스 분석 방법은 다음과 같다.

- 시스템의 내부/외부간의 인터페이스를 식별하기 위해서는 상호 연동되는 시스템, 서브시스템, 컴포넌트들의 관계를 명확히 한다. 또한 연동 시에 주는 대상과 받는 대상을 구별하여 분석한다. 이는 주거나 받을 때에 연동 정보가 다를 수도 있고, 해당 연동 정보를 다른 요소로 전달할 수도 있기 때문이다.
- 인터페이스 관계를 정리하였다면, 어떠한 정보/입력 데이터를 전달하는지 정의한다. 이때 인터페이스 연동되는 대상 간 어떠한 정보/입력 데이터를 주고받지 않는다면, 앞에서 분석한 연동 대상을 잘못 분석하였거나, 해당 기능이 정상적으로 작동할 수 없는 상태라고 볼 수 있다.
- 인터페이스 관계, 인터페이스 정보를 정리하였다면, 어떻게 정보/입력 데이터들을 전달할 것인지 정의한다. 예를 들자면 통신 시스템에서는 A요소에서 B요소로 정보를 전송할 때, 특정 프로토콜을 사용하여 전달한다. 이때 프로토콜은 인터페이스의 한 형태가 될 수 있다.
- 인터페이스 관계, 인터페이스 정보, 인터페이스 형태를 토대로 해당 인터페이스에 대한 간략한 설명을 기술한다.

(2) 인터페이스 분석 결과를 토대로 인터페이스 위험원을 식별한다. 인터페이스의 위험원은 인터페이스를 대상으로 HAZOP을 통해 식별된다.

(3) 인터페이스 위험원에 대한 원인과 인터페이스 위험원으로부터 발생될 수 있는 영향을 분석한다. 인터페이스 위험원에 대한 원인은 물리적 요소들에 대한 것뿐만 아니라 인터페이스를 구성하는 논리적 요소들에 대한 내용까지 포함한다. 또한 원인 개별마다 이로 인한 영향을 도출한다. 원인과 영향 분석이 완료되었다면, 이를 토대로 초기 발생빈도/심각도 평가를 수행한다. 초기 발생빈도/심각도 평가는 식별된 인터페이스 위험원이 발현되었을 때를 기준으로 수행한다. 위험도 평가에 대한 자세한 내용은 본 시스템 안전성 분석 가이드 ‘위험도 평가 기준’ 항을 참고한다.

(4) 앞서 인터페이스 위험원에 대한 원인과 영향을 분석한 내용을 토대로 원인 해결을 위한 저감대책을 도출한다. 인터페이스 위험원에 대한 저감대책은 인터페이스를 구성하는 대상들 간의 물리적/논리적 연동 시에 설계, 안전성 항목들을 포함시켜 작성한다. 저감대책 작성이 완료되었다면, 이를 토대로 잔여 발생빈도/심각도 평가를 수행한다. 잔여 발생빈도/심각도 평가는 초기 발생빈도/심각도 평가를 토대로 저감대책을 통해 해당 인터페이스 위험원이 발현되었을 때를 기준으로 수행한다.

(5) 잔여 발생빈도/심각도 평가 후 초기 발생빈도/심각도 평가 시와 비교하여 위험도 수준에 대한 평가를 수행한다. 이는 잔여 발생빈도/심각도 평가 후 적절하지 않은 저감대책으로 인해 더 안전한 수준을 유지하도록 저감대책을 마련해야 하지만 그렇지 못한 항목들에 대해서 식별할 수 있다. 이러한 항목들까지 모두 허용 가능한 상태가 되었을 때, 위험도 평가를 종료한다.

(6) 위험도 평가 수행 후 해당 결과를 바탕으로 시스템 인터페이스에 대한 안전 요구사항을 도출한다.

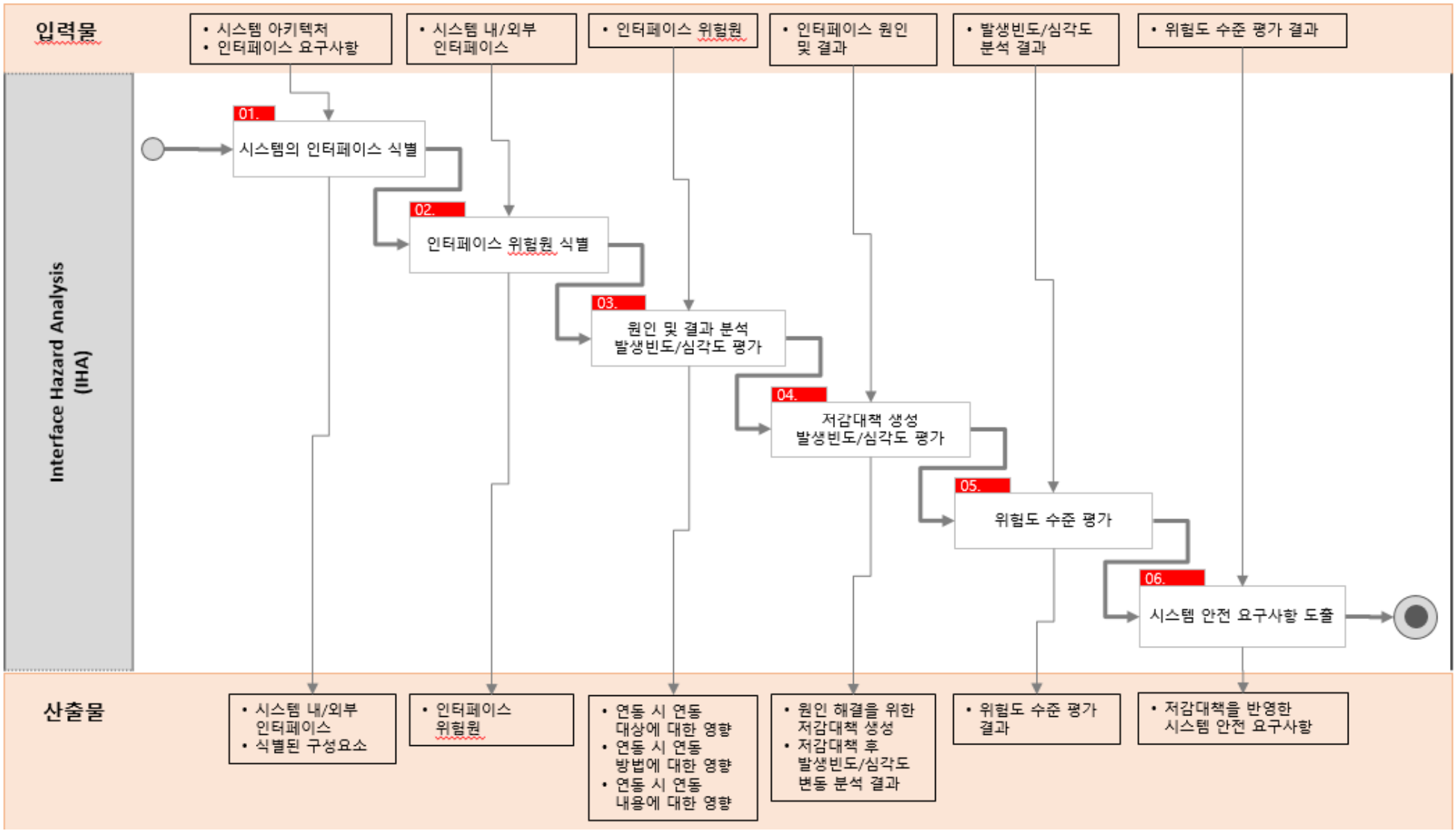


그림 63 IHA 활동 흐름도

6.5.3. 체크리스트

IHA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 대한 평가를 수행해야 한다. 또한 수행결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 IHA에 대한 검증은 [표 50]을 통해서 수행된다.

표 50. IHA 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
4. IHA 기법	4.1 대상 시스템의 물리적 경계는 정의 되었는가?		IHA수행 보고서	
	4.2 정의된 물리적 대상이 지니고 있는 인터페이스는 정의 되었는가?		IHA수행 보고서	
	4.3 시스템의 물리적 경계가 정의 되었다면, 내・외부 인터페이스는 정의 되었는가?		IHA수행 보고서	
	4.4 식별된 인터페이스 정보로부터 위험원은 식별 하였는가?		IHA수행 보고서	
	4.5 식별된 인터페이스 기반의 위험원을 저감할 수 있는 대책은 반영 되었는가?		IHA수행 보고서	

6.5.4. 적용 범위 및 제한조건

일반적으로 IHA는 다음 사항에 대한 안전성 분석을 수행하는 활동이다.

- O 물리적 환경과 인터페이스
- O 다른 기술적인 시스템과의 인터페이스
- O 인간과의 인터페이스
- O 다른 철도 인증기관과의 인터페이스

시스템의 물리적 환경과의 인터페이스와 다른 기술적인 시스템과의 인터페이스를 고려하여 IHA를 수행한다. 반면 인간과의 인터페이스는 운용적인 측면에서 O&SHA에서 수행하며, 다른 철도 인증기관과의 인터페이스는 본 시스템 안전성 분석 가이드에서 제공하는 범주에서는 제외하였다.

6.6. 결함 위험원 분석(FHA)

6.6.1. 개요

FHA의 목적은 시스템의 사고 발생 위험을 수용 가능한 수준까지 낮추기 위해서 위험원을 사전에 파악, 추적, 평가, 제거 또는 통제하기 위한 것이다. FHA 수행을 위해서는 시스템에 대한 운용 중 위험원 발생 시나리오를 정의해야 한다.

시스템 설계 초기 단계에서는 아직 여러 가지 정보들이 안정되어 있지 못하다. 그 중 어떤 산출물들은 시스템 설계 후기 단계에서 정돈되는 것도 있으며, 더욱이 컴포넌트에 대한 설계가 어느 정도 진행되어야 문제들을 파악할 수 있다. 따라서 FHA는 통상 컴포넌트 설계 단계에서 수행된다. FHA을 통해 각 컴포넌트는 한 가지 이상의 고장모드를 가질 수 있으며, 각각의 고장모드는 정상적인 시스템 기능에 위험을 초래할 수 있다. 따라서 시스템의 고장 모드나 위험원인 그리고 시스템이나 컴포넌트에 미치는 영향 등은 발생빈도와 심각도를 이용하여 기술해야 한다.

6.6.2. 체크리스트

FHA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 FHA에 대한 검증은 [표 51]을 통해서 수행된다.

표 51 FHA 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
5. FHA 기법	5.1 대상 시스템이 지니고 있는 기능은 모두 식별 되었는가?		FHA수행 보고서	
	5.2 대상 시스템이 지니고 있는 기능을 모드(Mode)로 구분 할 수 있는가?		FHA수행 보고서	
	5.3 개별 모드에 따른 기능의 오류(오동작)수행 시 발생되는 상황은 식별 되었는가?		FHA수행 보고서	
	5.4 대상 시스템이 지니고 있는 개별 기능의 오류로부터 발생 가능한 위험원은 식별 되었는가?		FHA수행 보고서	
	5.5 식별된 위험원으로부터 저감 대책은 수립되고 반영 되었는가?		FHA수행 보고서	

6.7. 운용 및 지원상의 위험원 분석(O&SHA)

6.7.1. 개요

시스템의 운용과 지원 또는 유지보수 상에 관련된 위험원을 규명하기 위해 O&SHA을 수행한다. 시스템에 대한 운용 및 유지보수 계획, 절차를 기반으로 관련된 잠재적 위험원을 규명한다. 또한 각각의 위험원의 위험도를 저감시키기 위하여 필요로 하는 안전 요구사항, 운용 절차 및 해당 검증 방법을 정의한다. O&SHA는 인간공학, 훈련과 교육, 인간기계 인터페이스(MMI)를 강조하면서 시스템의 운용과 유지보수에 관련된 위험성을 분석하기 위한 것이다.

또한 O&SHA는 운영자, 유지보수 담당자, 승객 등에게 발생할 수 있는 위험원을 식별하는데 목적이 있다. 뿐만 아니라, 식별된 위험원을 최소화 또는 제거하기 위해 본 안전성 분석에서 도출된 저감대책이 운용 및 유지보수 매뉴얼 및 안전 계획서에 반영해야 한다. 시스템이 운용되기 시작하면 예상치도 못한 여러 가지 상황에서 여러 방면으로 시스템을 사용하는 일이 발생하기 때문에 바람직하지 않은 혼란 및 사고가 초래될 수 있다. 시스템 특성과 관련된 정보들이 추가됨에 따라 반복적으로 O&SHA 결과가 지속적으로 수정되어야 한다. 이를 통하여 설계 완료 이전에 제안된 변경사항, 추가사항 그리고 개발 시스템에 대한 기능적 관점에서 운용 모드가 개발되고 평가해야 한다. 따라서 O&SHA는 위험원 발생 가능성을 정량적으로 예측하고 운용/유지보수 절차 및 운용상 위험원을 유발할 수 있는 조건이나 오류 등을 파악할 수 있어야 한다.

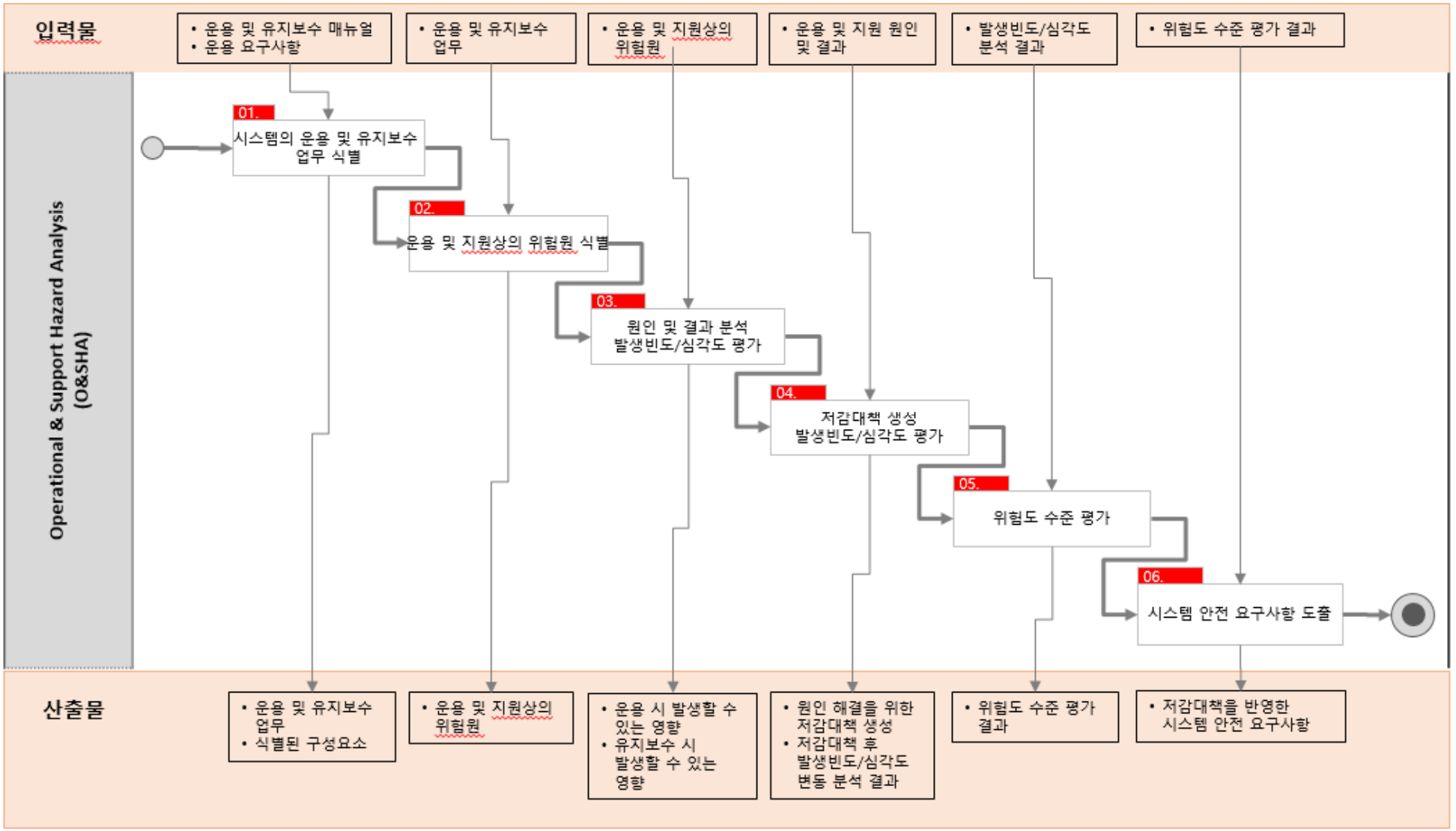


그림 64 O&SHA 활동 흐름도

6.7.2. 절차

O&SHA 수행을 위한 절차는 [그림 64]과 같으며 상세한 내용은 다음과 같다.

(1) 운용 및 유지보수 매뉴얼 분석을 통해 도출한 운용 및 유지보수 업무를 기반으로 시스템의 운용 및 유지보수 업무를 분석한다.

(2) 운용 및 유지보수 분석 결과를 토대로 운용 및 지원상의 위험원을 식별한다. 운용 및 지원상의 위험원은 운용 및 유지보수 업무를 대상으로 HAZOP 분석을 통해 식별된다.

(3) 운용 및 지원상의 위험원에 대한 원인과 운용 및 지원상의 위험원으로부터 발생될 수 있는 영향을 분석한다. 운용 및 지원상의 위험원에 대한 원인은 물리적 요소들에 대한 것뿐만 아니라 물리적 요소를 구성하는 소프트웨어 요소들에 대한 내용까지 포함한다. 또한 원인 개별마다 이로 인한 영향을 도출한다. 원인과 영향 분석이 완료되었다면, 이를 토대로 초기 발생빈도/심각도 평가를 수행한다. 초기 발생빈도/심각도 평가는 식별된 운용 및 지원상의 위험원이 발현되었을 때를 기준으로 수행한다. 위험도 평가에 대한 자세한 내용은 본 시스템 안전성 분석 가이드 ‘위험도 평가 기준’ 항을 참고한다.

(4) 앞서 운용 및 지원상의 위험원에 대한 원인과 결과를 분석한 내용을 토대로 원인 해결을 위한 저감대책을 도출한다. 운용 및 지원상의 위험원에 대한 저감대책은 운용 및 유지보수 시 물리적/논리적 구성 요소들의 설계, 안전성 항목들을 포함시켜 작성한다. 저감대책 작성이 완료되었다면, 이를 토대로 잔여 발생빈도/심각도 평가를 수행한다. 잔여 발생빈도/심각도 평가는 초기 발생빈도/심각도 평가를 토대로 저감대책을 통해 해당 운용 및 지원상의 위험원이 발현되었을 때를 기준으로 수행한다.

(5) 잔여 발생빈도/심각도 평가 후 초기 발생빈도/심각도 평가 시와 비교하여 위험도 수준에 대한 평가를 수행한다. 이는 잔여 발생빈도/심각도 평가 후 적절하지 않은 저감대책으로 인해 더 안전한 수준을 유지하도록 저감대책을 마련해야 하지만 그렇지 못한 항목들에 대해서 식별할 수 있다. 이러한 항목들까지 모두 허용 가능한 상태가 되었을 때, 위험도 평가를 종료한다.

(6) 위험도 평가 수행 후 해당 결과를 바탕으로 시스템 운용 및 지원상에 대한 안전 요구사항을 도출한다.

6.7.3. 체크리스트

O&SHA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 O&SHA에 대한 검증은 [표 52]을 통해서 수행된다.

표 52 O&SHA 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
5. O&SHA 기법	5.1 대상 시스템이 지니고 있는 운용 및 지원상 업무는 모두 식별 되었는가?		O&SHA수행 보고서	
	5.2 대상 시스템이 지니고 있는 운용 및 지원상 업무에 따른 매뉴얼이 존재하는가?		O&SHA수행 보고서	
	5.3 대상 시스템이 기능의 오류(오동작)수행 시 운용 및 유지보수를 하기 위한 상황은 식별 되었는가?		O&SHA수행 보고서	
	5.4 대상 시스템이 지니고 있는 운용 및 지원상의 상황으로부터 발생 가능한 위험원은 식별 되었는가?		O&SHA수행 보고서	
	5.5 식별된 위험원으로부터 저감 대책은 수립되고 반영 되었는가?		O&SHA수행 보고서	

6.8. HAZOP(HAZard and OPerability Analysis)

6.8.1. 개요

HAZOP은 위험원의 식별을 위한 형식화된 시스템적 기법이다. HAZOP은 시스템 기능, 인터페이스, 운용과 관련된 위험원을 식별하고, 분석하기 위한 기술로서 복잡한 분석과정을 보다 단순화 하여 체계적으로 분석하는 방법이다. HAZOP은 계획된 의도부터 가이드 워드의 사용을 통해 일어날 수 있는 시스템의 이상 현상(Deviation)을 확인하는 것이며 다음과 같은 목적을 가진다.

- 의도된 설계기능이나 조건 등을 포함하여, 시스템의 상세한 설명을 제공한다.
- 시스템의 모든 부분을 체계적으로 검토함으로써, 설계 의도로부터 이상현상(사고, 고장, 결함)이 어떻게 발생하는가를 파악한다.
- 이상 현상들이 기능, 인터페이스, 운용상의 문제를 초래할 수 있는지 결정한다.

위험원 식별에서 가장 중요한 기술은 위험원의 원인을 분석하는 것으로, 이러한 단계에서 HAZOP을 이용하면 안전성 분석 초기에 유용하게 위험원을 식별할 수 있다. 또한 시스템의 안전성 분석은 지속적으로 검토되어야 하며, 특히 시스템에 변경사항이 발생했을 경우에 추가적으로 HAZOP을 다시 수행하는 것이 매우 중요하다.

6.8.2. 절차

HAZOP은 다음과 같은 순서로 진행된다.

(1) 설계자의 의도로부터 발생 가능한 차이점을 가정해야 한다.

(2) 그 차이점으로부터 발생 가능한 원인 및 영향을 조사해야 한다.

(3) 시스템을 위해 계획된 또는 실제 현장에서의 제어 및 관리 방법을 설명한다.

(4) 위험원인의 존재 여부를 결정해야 한다.

(5) 회의 결과를 기록해야 한다.

HAZOP은 시스템의 위험원을 도출하기 위해 가이드 워드(Guide word)와 파라미터(Parameter)를 이용한다. 이러한 가이드 워드와 파라미터를 이용하여 설계 의도로부터 벗어나 사고로 이어질 수 있는 상태인 이상 현상이 정의된다. HAZOP이 수행되는 동안 가이드 워드와 파라미터는 시스템의 특성에 따라 전문가의 충분한 협의를 거쳐 수정이 가능하다. 가이드 워드란 HAZOP만이 가지고 있는 독특한 기법으로, 난상토론을 통해 이상 현상을 발견할 수 있도록, 의도된 기능을 정성적으로 또는 정량화하는 데 사용되는 간단한 용어들을 말한다.

가이드 워드를 활용해 시스템의 의도에서 벗어난 잠재적 이탈 상태 파악 및 다양한 레벨의 각종 시스템에 적용 가능(서브시스템, 컴포넌트, 소프트웨어, 절차, 환경, 인적 오류 등)하다. 또한 HAZOP의 수행시점은 생명주기상 모든 단계에서 활용되는 것이 아니며, 개념 설계 단계와 상세 설계 단계에서 활용된다. 그리고 HAZOP에 대한 어려움이나 또는 불가하다면, PHA, SSHA의 산출물을 활용하거나 또는 대체하여 수행할 수 있다.

표 53 가이드워드 종류 및 설명

가이드워드(Guide word)	설 명
No	정의된 파라미터가 없음
Part of	일부만 수행되거나 고려됨
Early	정의된 시간보다 일찍 수행됨
Late	정의된 시간보다 늦게 수행됨
More	파라미터의 초과 및 증가
Less	파라미터의 미달 및 감소
Other	비정상적인 파라미터

표 54 파라미터 종류 및 설명

파라미터(Parameter)	설 명
Interface	철도 시스템/서브시스템 간 인터페이스
Time	시스템의 동작 시간, 처리 시간 또는 일반적으로 정의된 시간
Action	운용자, 기관사, 유지보수자의 조작 또는 시스템 조작
Limit	정보 처리 한계, 시스템의 가용성 및 신뢰성 한계
Procedure	운용, 운전, 유지보수 절차, 비상 시 대응 절차
Outside	자연현상, 승객 행동, 의도적인 외부 장애요건
Data	제어명령, 현장 정보, 데이터베이스 정보

6.8.3. 예시

철도 객실 내 비상방송장치에서 “객실 내 방송” 시에 HAZOP에 대한 수행 예시는 다음과 같다.

표 55 HAZOP 수행 양식

HAZOP Analysis									
(1) 번호	(2) 항목	(3) 기능/파라미터	(4) 가이드워드	(5) 결과	(6) 원인	(7) 위험	(8) 리스크	(9) 조치	(10) 비고

표 56 HAZAOP 수행 결과 (예시)

HAZOP 분석										
(1) 번호	(2) 항목	(3) 기능 /목적	(4) 파라미터	(5) 가이드 워드	(6) 결과	(7) 원인	(8) 위험	(9) 리스크	(10) 조치	(11) 비고
1	비상 방송 장치	객실 내 방송	Action	No	비상 시 객실 내 비상방송 불가	스피커 고장	객실 내 방송 불가	A	인증된 부품 사양서를 통한 신뢰성이 확보된 부품 사용	
			Action	No	비상 시 객실 내 비상방송 불가	내부 부품 고장	객실 내 방송 불가	A	인증된 부품 사양서를 통한 신뢰성이 확보된 부품 사용	
			Interface	No	비상 시 객실 내 비상방송 불가	통신 불량	객실 내 방송 불가	A	3GPP, TTA 통신규격을 통한 신뢰성 확보	
2	비상 방송 장치	객실 내 방송	Action	Part of	비상 시 객실 내 비상방송 끊김	스피커 고장	객실 내 일부의 방송	B	인증된 부품 사양서를 통한 신뢰성이 확보된 부품 사용	
			Interface	Part of	비상 시 객실 내 비상방송 끊김	통신 불량	객실 내 일부의 방송	B	3GPP, TTA 통신규격을 통한 신뢰성 확보	
			Outside	Part of	비상 시 객실 내 비상방송 끊김	전자파 간섭	객실 내 일부의 방송	B	EMC 규격에 부합하여 설계	
3	비상 방송 장치	객실 내 방송	Action	Less	비상 시 객실 내 비상방송 끊김	스피커 고장	객실 내 방송 끊김	B	인증된 부품 사양서를 통한 신뢰성이 확보된 부품 사용	
			Interface	Less	비상 시 객실 내 비상방송 끊김	통신 불량	객실 내 방송 끊김	B	3GPP, TTA 통신규격을 통한 신뢰성 확보	
			Outside	Less	비상 시 객실 내 비상방송 끊김	전자파 간섭	객실 내 방송 끊김	B	EMC 규격에 부합하여 설계	

다음은 철도 신호 시스템 ATP의 ‘임시속도 제한 설정’ 오류에 대한 사례 예시이다. 국내 철도 분야에서 적용되는 ‘HAZOP-KR’에서 제공하는 파라미터 및 가이드워드에 따른 HAZOP을 수행한 사례에 대한 예시이다.

매개변수	가이드 워드	이상현상	원인	설명	결과	저감대책
Interface	No	Interface 불가	선택된 구역에 대한 구역 확인 실패	ATS로부터 선택된 구역 확인 실패로 임시속도를 제한할 수 없음.	열차추돌	ATS로부터 인터페이스 불가 시 제동발생 또는 저속 운행
			선택된 구역에 대한 임시속도 정보 확인 실패	ATS로부터 선택된 구역은 확인하였으나, 임시속도 정보 확인 실패로 임시속도를 제한할 수 없음.	열차추돌	ATS로부터 인터페이스 불가 시 제동발생 또는 저속 운행
			임시속도 정보 설정 및 제거 실패	ATS와 열차 간의 임시속도 정보를 설정 및 제거 확인 실패로 선택된 구역의 확장 및 감소를 지정할 수 없음.	열차충돌 및 추돌	ATS로부터 인터페이스 불가 시 제동발생 또는 저속 운행
Action	No	동작하지 않음	구역 내 임시속도제한 실패	임시속도제한 실패로 인한 열차 과속으로 인한 안전제동 실패	열차충돌 및 추돌	S/W 건전성확보 및 기능시험
Limit	More	정의된 한계를 초과함	열차 임시속도 초과	열차 과속으로 인한 안전제동 실패	열차충돌	열차속도 초과 시, 열차 운행 속도 제어
Data	No	Data 없음	임시속도제한 설정 오류	안전하지 못한 임시속도제한 설정으로 열차 간 접근	열차충돌	ATS로부터 인터페이스 불가 시 제동발생 또는 저속 운행
	Other	기타 비정상적인 조작 및 동작	임시속도제한 설정 오류	잘못된 임시속도제한으로 인한 열차간격 유지 실패	열차충돌 및 탈선	열차와의 인터페이스 불가 시, 열차 비상정지

그림 65 HAZOP 예시 – 위험원이 “임시속도제한 설정 오류”인 경우

6.8.4. 체크리스트

HAZOP에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 HAZOP에 대한 검증은 [표 57]을 통해서 수행된다.

표 57 HAZOP 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
7. HAZOP 기법	7.1 대상 시스템의 시스템 기능, 인터페이스, 운용은 식별 되었는가?		시스템 운용 개념서	
	7.2 시스템 기능, 인터페이스, 운용에 작용하는 매개변수는 식별 되었는가?		HAZOP 수행 결과 보고서	
	7.3 가이드워드는 식별 되었는가?		HAZOP 수행 결과 보고서	
	7.4 파라미터는 식별 되었는가?		HAZOP 수행 결과 보고서	
	7.5 가이드워드에 따른 이상 현상은 식별 되는가?		HAZOP 수행 결과 보고서	
	7.6 이상 현상을 일으키는 원인요소는 식별 되었는가?		HAZOP 수행 결과 보고서	
	7.7 이상 현상에 따른 시스템 또는 하부 요소에 미치는 결과는 어떠한가?		HAZOP 수행 결과 보고서	
	7.8 이상 현상을 제거 또는 최소화하기 위한 저감대책이 설계에 반영되었는가?		HAZOP 수행 결과 보고서	

바로 다음 기술할 안전성 분석 기법 중 FMEA, FTA는 상호 안전성 분석 접근의 차이에 대한 사전 이해를 돕기 위해 [그림 66]에 표현하였다. FMEA는 상향식(Bottom-up) 방식의 접근이며, FTA는 하향식(Top-Down) 방식의 접근을 통해 안전성 분석을 수행한다. 이러한 차이는 FMEA, FTA 각 기법에 대한 설명을 통해 확인할 수 있다.

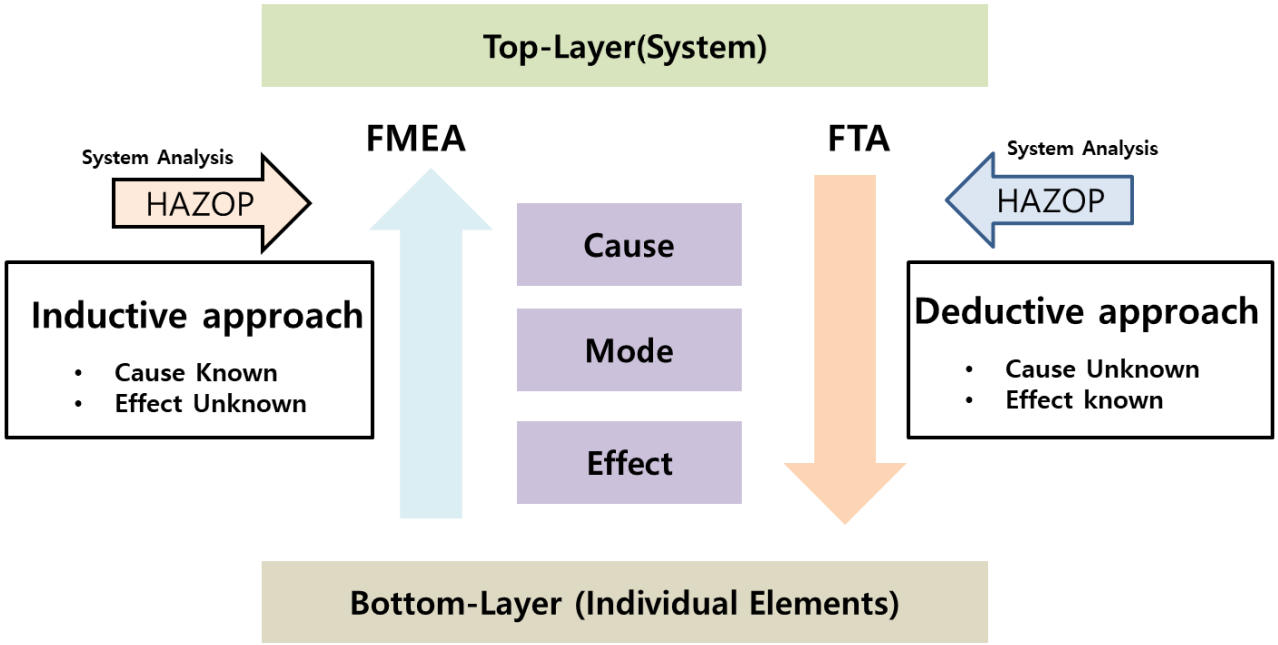


그림 66 안전성 분석 접근의 차이

6.9. FMEA(Failure Mode and Effects Analysis)

6.9.1. 개요

FMEA는 서브시스템, 컴포넌트 혹은 기능들의 잠재적 고장 상태의 영향을 측정하기 위한 분석 기법이다. 이것은 종합적인 시스템 신뢰도에 부정적인 영향을 주는 고장상태를 확인하기 위한 분석 기법이기도 하다. 또한 FMEA는 정량적인 분석을 위해 각각의 고장상태에 대한 고장률을 포함하는 특성을 가지고 있다. 게다가 FMEA는 시스템 위험원과 같이 원치 않은 시스템 상태에 의한 고장상태를 측정하기 위해 확장될 수 있으므로 위험원 분석에도 사용될 수 있다.

FMEA는 컴포넌트 그 자체에 고장 발생 원인이 개입되는 것을 피하기 위한 기법이다. FMEA은 위험과 잠재적 고장상태의 발견을 다룰 때의 자세한 분석과 정보가 포함된 더 많은 정보를 요구한다. FMEA에서 심각성 부분을 확장 기법은 FMECA이다. FMECA에서 C(Criticality)가 의미하는 것은 다양한 고장 영향성의 심각성이 고려되는 정도로 표현한다.

일반적으로 FMEA를 수행을 한 후에 FMECA 활동의 수행을 통해, 시스템 안전 요구사항 분석을 수행하게 된다. FMEA 방식은 시스템 고장이나 프로세스 고장의 위험률을 줄이기 위한 활동의 우선순위를 정하기 위해 시스템과 프로세스의 설계나 기능에 초점을 맞춘 체계적인 상향식 접근의 기법이며 분석된 결과를 기록하고 권고된 설계변경을 반영하는 기법이다. FMEA은 시스템이 고장 난 경우에 고장이 미치는 영향을 파악하는 것을 목적으로 한다.

○ FMEA 생명주기(Vee-Model) 관점

보다 효율적인 FMEA을 위해서는 [그림 67]에서 알 수 있듯이, 시스템 아키텍처 기반의 물리적 컴포넌트 분석을 통해 수행해야 한다. 시스템/하드웨어/소프트웨어에 이르기 까지, 각각의 시스템 수준에서 설계 산출물이 활용될 수 있다. 특히, 구조적 정보, 기능 기반의 고장에 대한 정보 데이터를 활용해야 한다. 본 시스템 안전성 분석 가이드에서는 도구 기반의 FMEA을 통해, 설계 산출물 확보와 개별 산출물 데이터 간 추적성(Traceability)을 확립할 수 있다.

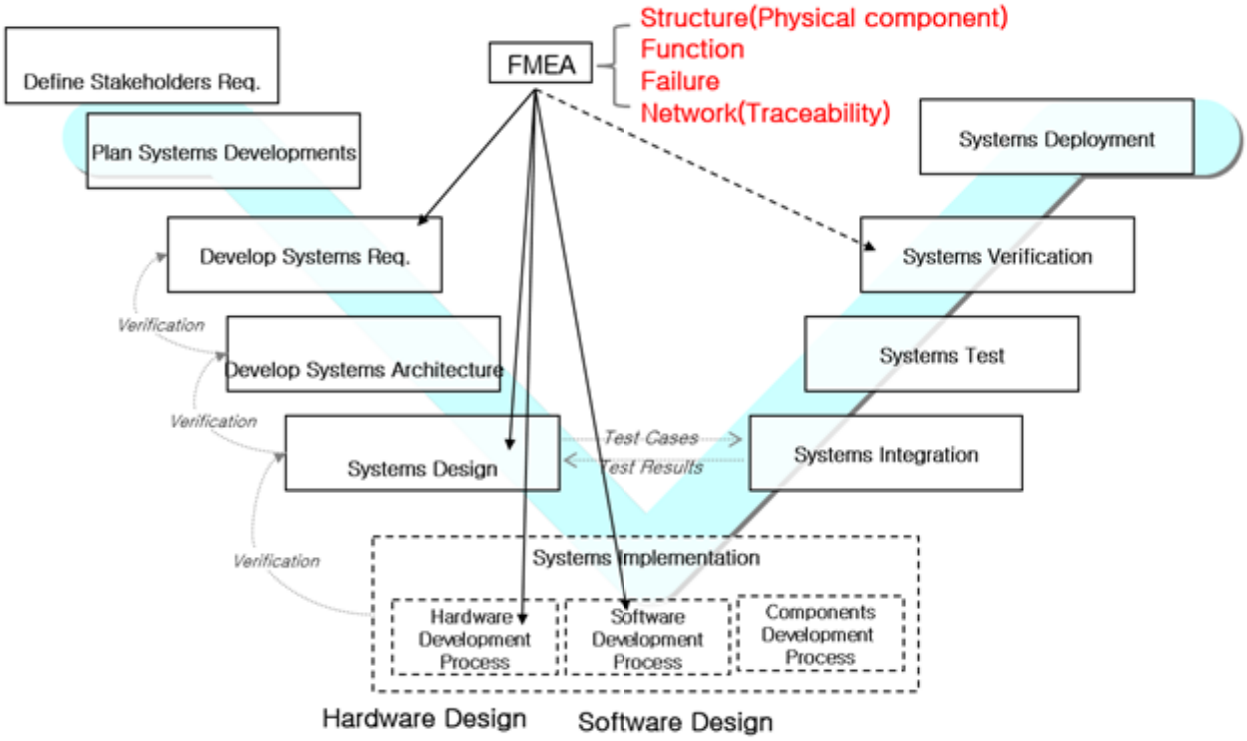


그림 67 설계적 데이터를 활용한 FMEA 수행 [59]

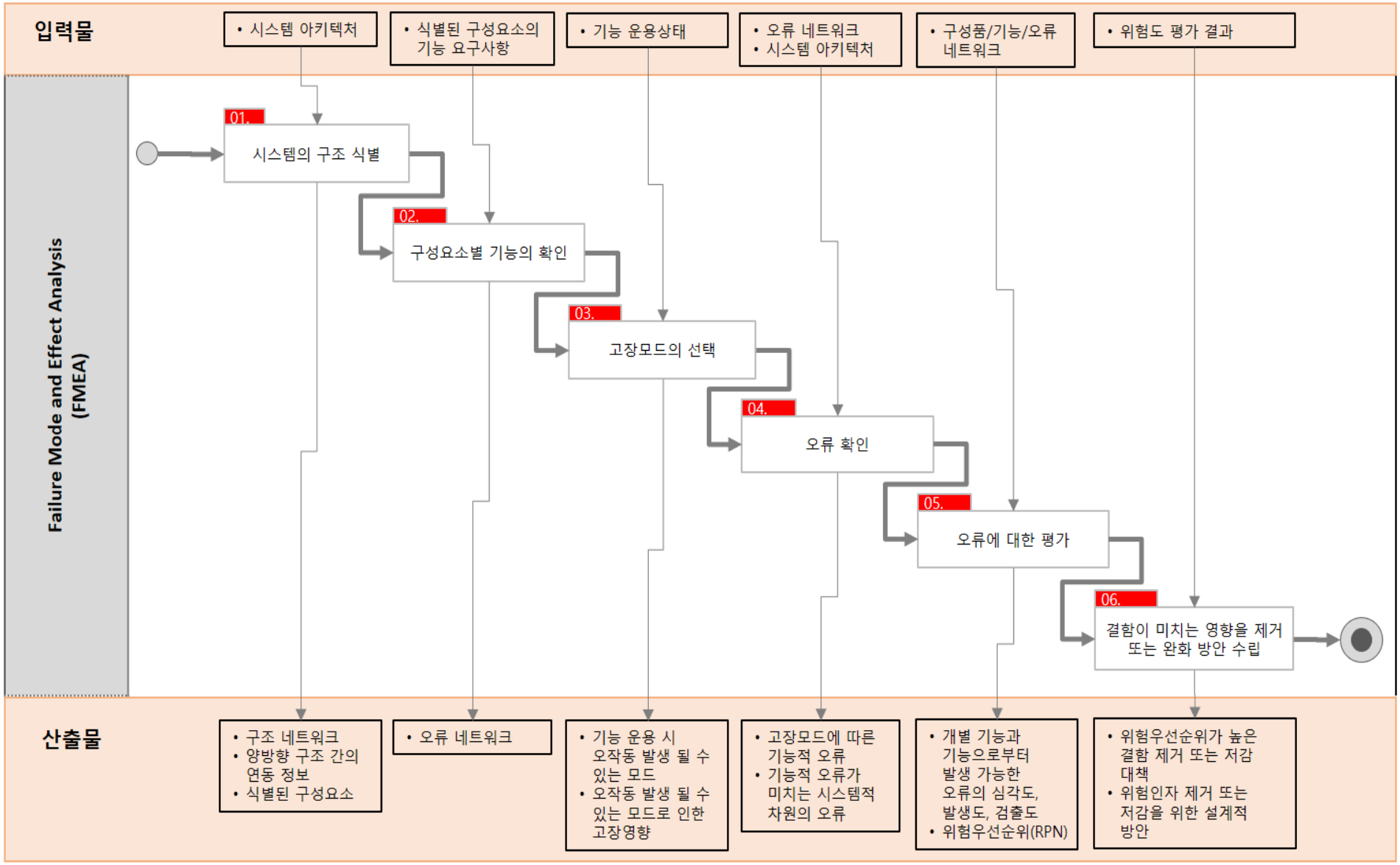


그림 68 FMEA 활동 흐름도

6.9.2. 절차

FMEA 수행을 위한 단계는 다음과 같다.

(1) 컴포넌트의 식별

- 컴포넌트의 식별은 아키텍처 산출물을 활용해 수행 대체 가능하다.

(2) 컴포넌트별 기능의 확인

- 개별 컴포넌트가 지닌 기능을 확인하여 기능 분석된 결과를 나열한다.

(3) 고장모드의 선택

- 개별 컴포넌트가 지닌 기능 작동(운용)시 발생할 수 있는 다양한 모드에서 오작동 발생 될 수 있는 모드를 식별하여 나타낸다.

(4) 지역적인 오류 확인(결함이 시스템의 다른 부분에 즉시 영향을 주는 것)

- 해당 결함모드에 따른 기능적 오류 발생 시 발생하는 연계 오류 된 사항을 기술하게 된다.

(5) 시스템적 오류 확인(시스템 전체에 대한 영향)

- 기능적 오류가 미치는 시스템적 차원의 오류를 식별하여 기술한다. 이때, 아키텍처 산출물을 기반으로 분석 가능하다.

(6) 결함이 영향을 차단하기 위한 방법 확인

- 결함이 미치는 영향을 제거 또는 완화하기 위한 방안을 설계적으로 반영될 수 있는 방안을 기술하게 된다.

(7) 권고안 작성

- 최종적으로 설계적 반영하기 위한 권고안에 기술하게 된다.

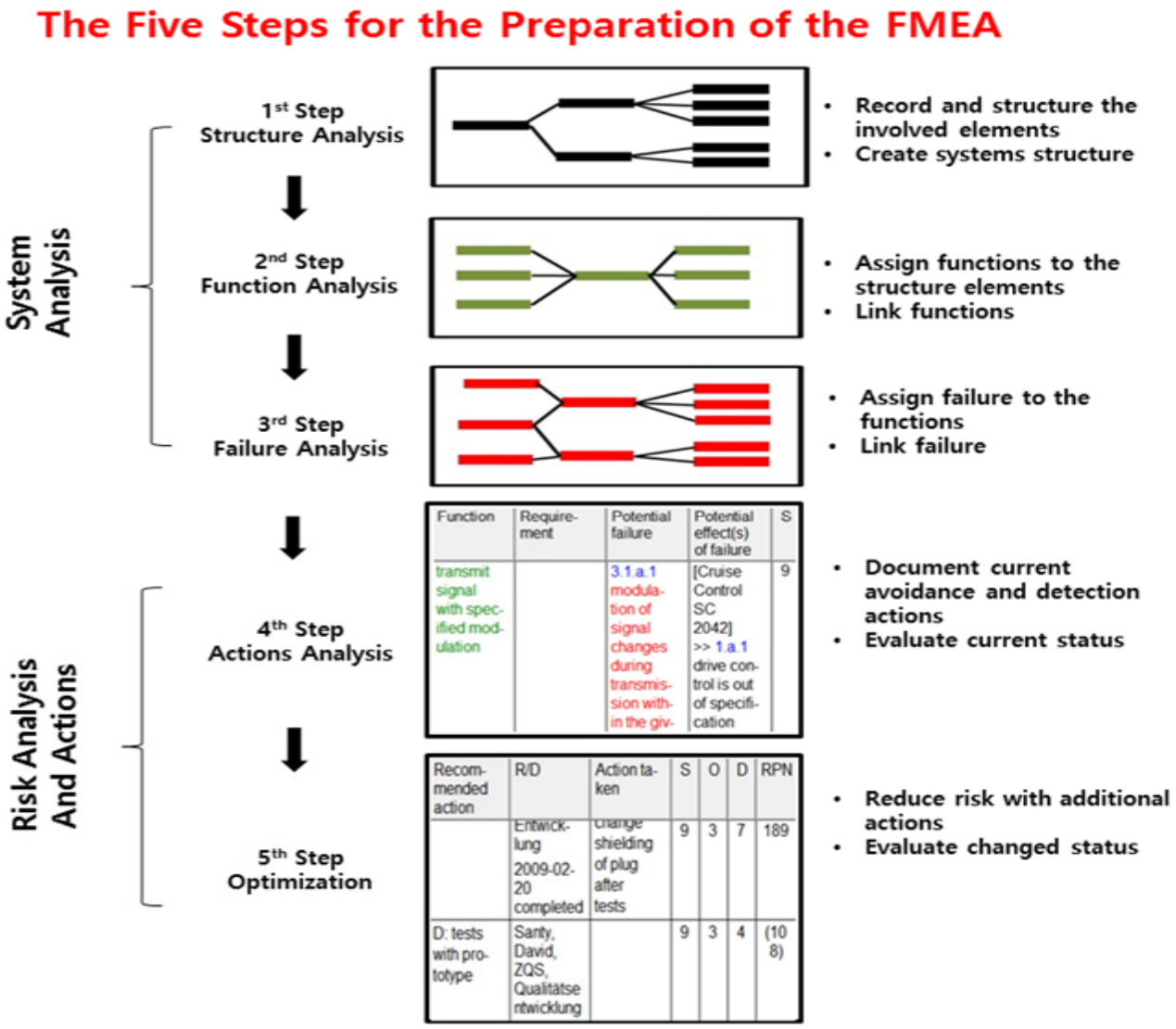


그림 69 FMEA 수행을 위한 단계

FMEA은 [그림 69]와 같이, 크게 2개의 범주 영역으로 구분된다. 하나는 시스템 분석 단계이며, 다른 하나는 위험분석 및 조치 단계이다. 앞에서 언급한 것처럼 FMEA을 수행하기 위해서는 물리적 컴포넌트를 식별해야 한다. 이러한 맥락에서 앞에서 언급한 시스템 분석 과정을 거쳐 해당 시스템이 어떻게 구성되었는지/해당 컴포넌트는 어떠한 기능이 있고, 해당 기능이 고장 시 어떠한 오류로 나타내는지 분석하는 과정을 거치게 된다. 시스템 분석 단계가 마치게 되면, 위험 분석 및 조치 단계 과정으로 넘어간다. 위험 분석 및 조치 단계에는 식별된 물리적 컴포넌트가 지니고 있는 잠재적 오류에 대한 심각도/발생빈도 등에 대한 평가를 통한 결과를 바탕으로 설계적으로 오류에 대한 결함을 보완하기 위한 접근을 수행한다.

6.9.3. 예시

FMEA는 시스템 구조의 하위 수준에서 상위 수준으로 상향식 분석을 실행한다. FMEA 시 컴포넌트 수준의 고장결과를 시스템 수준의 고장 원인으로 기술해야 한다. 이와 같은 과정은 시스템의 원인을 분석하기 위하여 기능적 구조를 따라 상향식으로 반복적으로 이루어진다. 다음은 FMEA를 적용한 원인 분석의 예를 나타낸다.

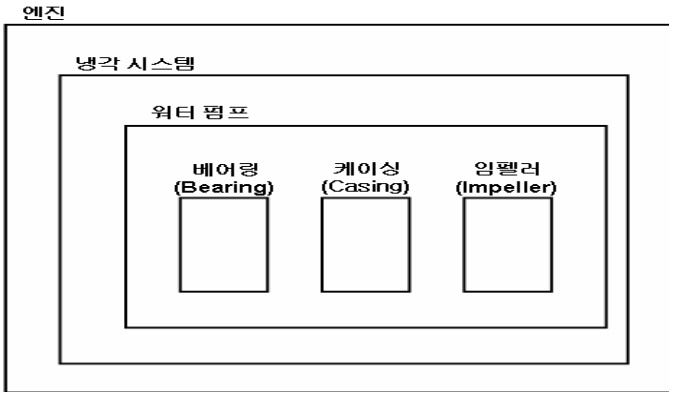


그림 70 FMEA 예시

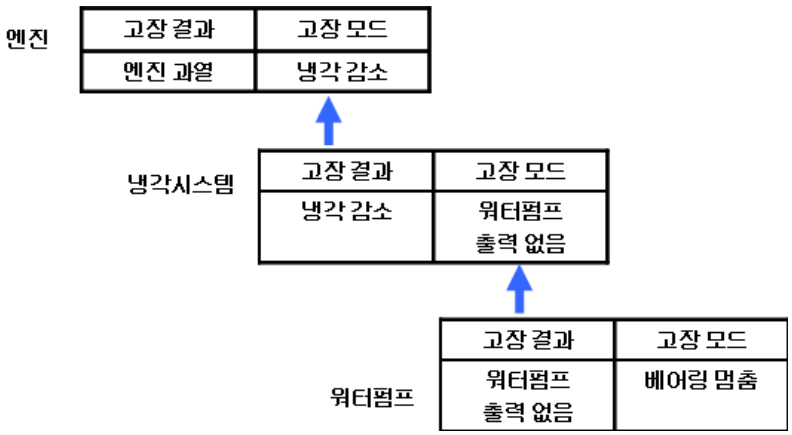


그림 71 FMEA 기반의 원인분석 과정

표 58 FMEA를 이용한 원인분석 예시

항 목	설 명	모 드	원 인	국지적결과	보상 규정	최종 결과
1	베어링	이상 정지	제조 결함	출력 없음	과열 경고	과열
2	케이싱	파손	충돌	냉각제 손실	과열 경고	과열
3	임펠러	파손	피로	출력 감소	과열 경고	과열

6.9.4. 체크리스트

FMEA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 FMEA에 대한 검증은 [표 59]를 통해서 수행된다.

표 59 FMEA 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
8. FMEA 기법	8.1 시스템 구조 분석을 수행할 수 있는 시스템 구조 아키텍처 산출물을 확보하고 있는가?		시스템 구조 아키텍처	
	8.2 시스템 기능 분석을 수행할 수 있는 시스템 기능 아키텍처 산출물을 확보하고 있는가?		시스템 기능 아키텍처	
	8.3 시스템의 구조화 분석이 수행되었는가?		FMEA 수행 결과 보고서	
	8.4 분석된 구조/기능을 중심으로 추적성이 확보 되었는가?		FMEA 수행 결과 보고서	
	8.5 식별된 컴포넌트가 지니고 있는 기능중심의 고장 모드는 식별 되었는가?		FMEA 수행 결과 보고서	
	8.6 식별된 기능의 오류에 대해서 시스템 또는 컴포넌트에 미치는 영향이 분석 되었는가?		FMEA 수행 결과 보고서	
	8.7 개별 기능적 오류 영향에 대한 심각도, 발생도, 검출도 평가 수행이 되었는가?		FMEA 수행 결과 보고서	
	8.8 내부 합의된 일정 수준의 기능적 오류의 심각도가 높은 우선순위 항목에 대한 설계적 조치가 반영 되었는가?		FMEA 수행 결과 보고서	

6.10. FTA(Fault Tree Analysis)

6.10.1. 개요

FTA는 시스템의 고장을 발생시키는 원인들과의 관계를 논리적으로 사용하여 나무 모양의 그림으로 나타낸 Fault Tree(FT)를 만들고 이에 기반 시스템의 고장확률을 구함으로서 시스템이 지닌 취약 부분을 찾아내 시스템의 신뢰도를 개선시키기 위한 정량적 고장해석 및 신뢰성 평가 방법이다. FTA의 주요 특징은 AND와 OR의 두 종류 논리게이트의 조합에 의해 시스템 또는 고장의 위험성을 나무(Tree) 구조에 의해 표현하므로, 가시적으로 파악하는데 우수한 수단이라 할 수 있다. 또한 기본 사건(Basic Event) 발생률로써 중간 및 정상 사건에 대한 확률을 차례로 계산할 수 있게 함으로써 경험기반의 사고로부터 탈피할 수 있으며 논리적이고 확률론적인 정량적 결과를 도출할 수 있다는 특징을 지니고 있다.

FTA는 열거된 원치 않은 사고의 발생확률과 기본 원인을 결정하기 위한 시스템 분석 기술이다. FTA는 잠재적 문제를 이해하고 방지하기 위하여 대규모 복합 기능의 시스템을 평가하는데 사용된다. 정확하고 조직화된 방법을 사용하여, FTA는 시스템 분석가가 원치 않은 사고의 발생 원인이 되는 고장 사건의 특수한 조합을 만들도록 한다. FTA의 목적은 최상위 위험원의 발생확률을 컴포넌트 별 고장률을 사용하여 저감 대책을 수립하는데 있다.

FTA는 반복적인 분석을 통해 수행된다. 시스템의 중요한 양상을 표현하는 각각의 주요 층으로 각 층에서의 FT 전개를 설명한다. 예를 들어, 최상위 FT는 대개 시스템 기능과 단계를 모델링하고, 중간 FT는 서브시스템 고장 흐름을 모델링하며, 컴포넌트의 FT는 컴포넌트 고장 흐름을 모델링 한다. FTA는 서브시스템 수준에서 기능이나 기능 간 상호작용에 잠재적 고장을 일으킬 수 있는 요인들로부터 위험원을 구분하는데 사용된다. 원인분석은 해당 시스템의 단일 고장과 이들이 전체 시스템에 미치는 영향을 파악하고, 위험원 조건을 제거하거나 이를 허용 가능한 수준까지 필요한 제어 방법을 결정하는데 사용된다.

FT는 시스템 수준의 위험원을 컴포넌트 수준의 위험원으로 구분하는데 사용된다. FTA는 연역 기법의 하향식 접근법으로 최상위 사건에서 상위 사건의 원인이 될 수 있는 기본 사건에 이르기까지 점진적으로 분석한다. 기본 사건은 서브시스템 기능의 고장 또는 외부 장비와의 서브시스템 인터페이스 장애이다. FT에서는 단일 혹은 다수의 기본 사건 발생으로 인해 발생될 수 있는 위험원으로 이어지는 경로들을 제시한다. FT Diagram은 “AND”, “OR” 등의 논리게이트를 사용하여 컴포넌트 수준의 위험원과 시스템 위험원 사이의 관계를 정의한다.

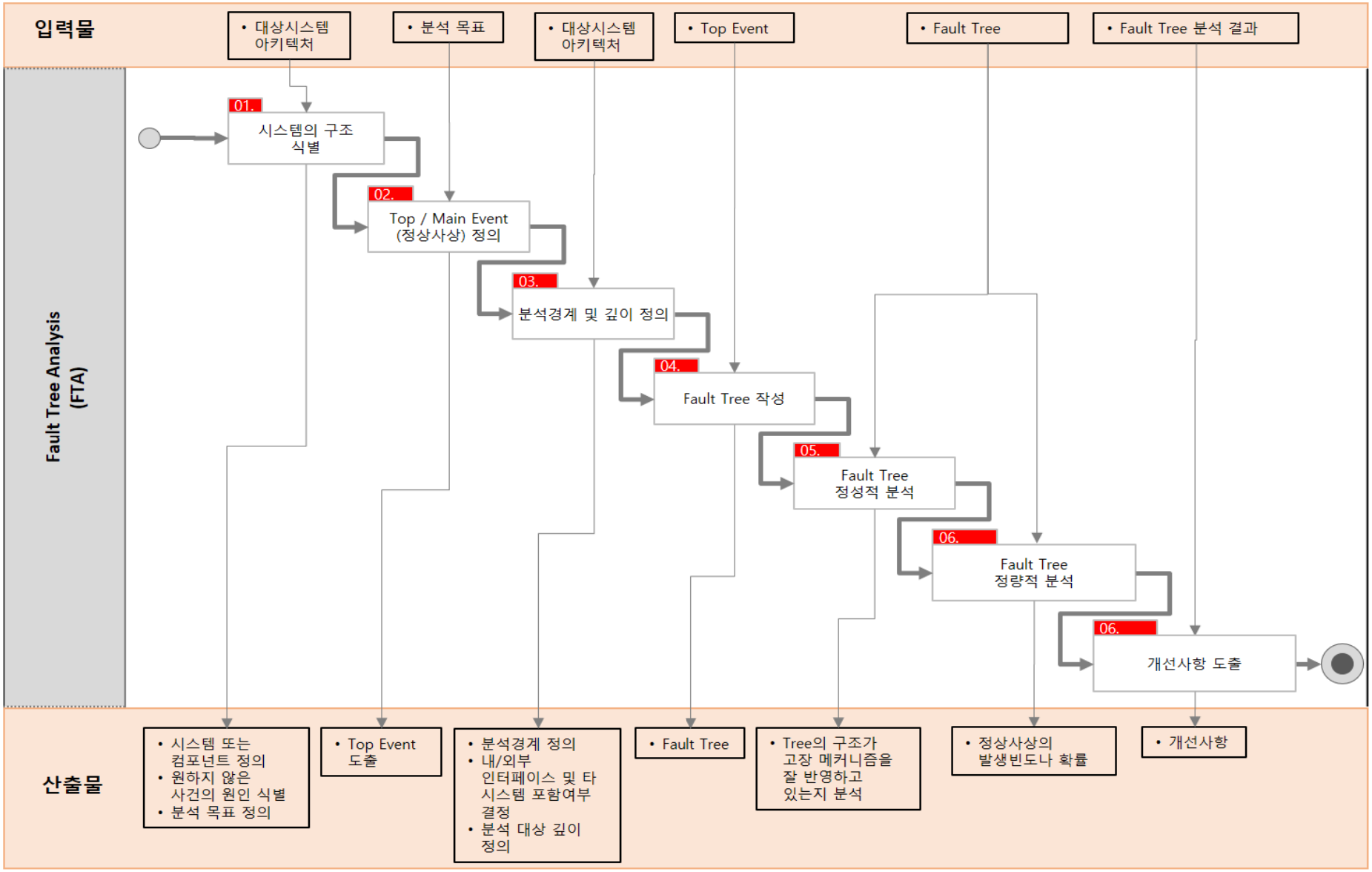


그림 72 FTA 활동 흐름도

6.10.2. 절차

FTA를 수행하기 위해서는 다음과 같은 항목들이 준비 되어져야 한다. FTA를 수행하기 위한 준비사항으로 다음과 같은 입력물이 사전에 준비되어야 한다.

(1) FME(C)A, (2) HAZOP 산출물 정보, (3) 시스템 설계 기준(System Design Criteria), (4) 시스템 설계 산출물, (5)고장률 데이터가 기본적으로 요구되어 진다. 또한 이러한 데이터를 기반으로 수행되는 FTA 수행절차는 적용이 되는 시스템의 특성이나 목적, 수준에 따라 다소 차이가 있을 수 있으나 일반적으로는 다음과 같다.

- ㅇ 시스템의 이해(정의 및 분석)
  - 원하지 않은 사건이 어떻게 일어나는지 그 원인 인지 식별 한다.
  - 시스템 또는 컴포넌트를 정의한다.
  - 분석 목표를 정의한다.
  - 가정(Assumption)을 기술한다.
  - FTA는 구성도/ 기능 구조도/ FMEA / HAZOP을 분석 및 수행하는데 중요한 정보를 제공한다.
- ㅇ Top / Main event(정상사건) 정의.
  - 분석 목표로부터 “Top Event” 를 도출한다.
- ㅇ 분석경계 및 수준 정의
  - 시스템의 분석 경계를 정의한다.
  - 내/외부 인터페이스 및 타 시스템 포함여부를 결정한다.
  - 시스템/컴포넌트 등의 분석 수준을 정의한다.
- ㅇ FT 작성
  - 일반적으로 FT는 사건과 그것을 연결하는 게이트로 구성된다.
  - Top Event 정의하고 그 아래 단에 그 “Top Event” 을 직접적으로 발생시키는 원인이 되는 사건들을 나열 후 논리게이트로 “Top Event” 와 연결
  - 아래 단의 사건과 바로 그 위의 단의 사건을 연결하는 논리게이트는 기본적으로 AND 게이트 그리고 OR 게이트
  - AND 게이트란 아래 단에서의 사건B1과 B2의 양쪽이 일어나면, 즉, [B1이고 더구나 B2]이며 위단의 출력 A가 생긴다는 것으로 논리곱의 관계를 표시 할 수 있다. 다시 말해, AND 게이트는 하위의 사건을 모두 만족하는 경우에 사용하는 논리게이트를 일컫는다.
  - OR게이트는 아래 단에서 사건B1과 B2의 어느 쪽 한쪽에서 일어나면 즉, [B1 또는 B2]면 위단의 출력 A가 생긴다는 것으로서 논리합의 관계를 표시하게 된다. 즉, 하위의 사건 중 하나라도 만족하면 사용하는 논리게이트 이다.

○ FT 정성적 분석

- FT가 작성되면 나무(Tree)의 구조가 정성적으로 고장 메커니즘을 잘 반영하고 있는지를 분석한다.
- 나무(Tree)의 구조는 모든 가능한 고장유형(즉, 정상사건을 일어날 수 있게 하는 사건들의 모든 조합들)들을 표현한다.
- 이 과정은 보통 Minimal Cutset 분석을 통하여 이루어진다.
- 특히, 보호 장치의 유효성, 여러 가지 하위 사건들의 정성적인 중요도, 공통 원인고장의 기여 정도 등을 관찰한다.

○ FT 정량적 분석

- FT를 수식으로 표현하거나 간소화된 표현을 위해 부울대수(Boolean Alegebra)를 사용한다. 부울대수 방법은 논리계산의 한 수단으로서 어느 집합을 구성하는 부분집합들의 논리곱과 논리합을 사용하여 표현하고, 이들의 확률을 계산식을 사용함으로서 FT의 정상사건이나 중간사건, 즉 구하고자하는 재해발생확률을 계산하는 방법을 말한다. 해당과정에서 주의해야할 사항은 FT 내의 2개소 이상에 동일 기본 사건이 포함되는 경우의 확률 계산은 부울대수에 의해 정리를 수행한 후 실시하지 않으면 전혀 다른 결과가 나올 수 있다는 점이다.
- FT가 수많은 기본 사건으로 구성된 경우가 많은데 이때는 그것을 몇 개의 부분 FT를 분해하여 각 부분에 대한 FT가 작성되고 각 기본 사건들의 발생빈도를 조사하고, 이로부터 정상사건의 발생빈도나 확률을 계산한다.
- 보통, 부울 대수식으로 표현되는 minimal Cutset들을 Poincare식을 사용하여 계산되어지며, 또한, 정상사건의 발생빈도가 계산되어지면 심각도, 불확실성과 영향도 분석 또한, 시행되어야 한다.
- FTA을 통해, 다음과 같은 사항들이 개선사항으로 도출되어 질 수 있다. (1) 시험을 통한 개선, (2) 런타임 진단 (3) 설계 변경, (4) 컴포넌트 변경이 해당된다.

FTA 수행에 대한 상세 절차는 다음의 단계와 과정을 거친다.

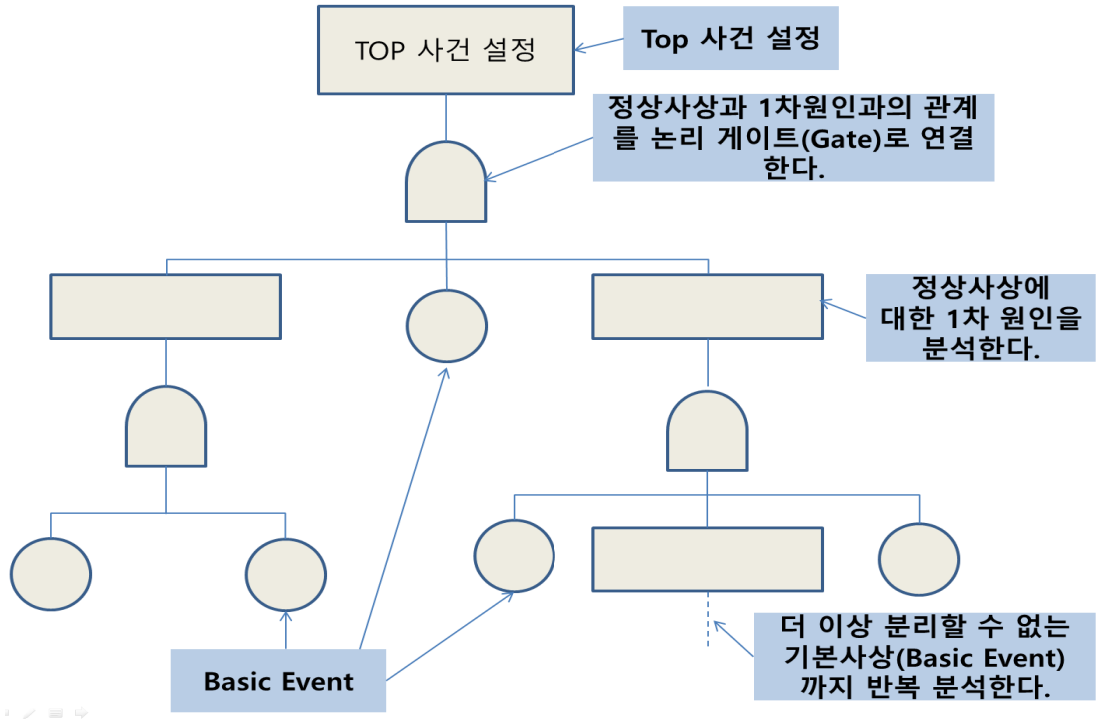


그림 73 FTA 수행의 주요 구성 및 산출물

(1) 정상사건(Top-Event) 설정을 수행해야 한다.

(2) 시스템 및 프로세스의 특성을 파악해야 한다.

(3) FT 작성을 하여야 한다.

- 정상사건에 대한 1차 원인을 분석해야 한다.
- 정상사건과 1차 원인과의 관계를 논리게이트로 연결해야 한다.
- 정상사건과 1차 원인과의 관계를 논리게이트로 연결한다.
- 1차 원인에 대한 2차 원인(결함사건)을 분석한다.
- 1차 원인과 2차 원인에 대한 관계를 논리게이트로 연결해야 한다.
- 4)~5) 항을 더 이상 분해할 수 없는 기본 사건(Basic Event)까지 반복적으로 분석해야 한다.

(4) FT 해석

- 작성된 FT를 수학적(Boolean Algebra)으로 간소화 되어야한다.
- Minimal Cut Sets, Minimal Path Sets를 구한다.
- 정상사건에 영향을 미치는 중요한 중간 및 기본 사건에 대해 파악을 수행한다.
- Critical Path: 가장 높은 확률을 가진 주요 원인의 경로

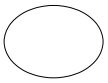
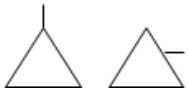
(5) FT 정량화

- 기본 사건의 발생빈도나, 고장률, 오류 데이터 등을 정리하여 중간사건 및 정상사건의 발생확률을 계산한다.
- 재해발생 확률 계산 결과는 과거의 재해 또는 유사한 재해의 발생률과 비교하고 현격한 차이가 날 경우는 작성된 FT에 대하여 재검토를 수행하여야 한다.

(6) 해석결과의 평가

- 재해 발생 확률이 허용할 수 있는 위험도 수준을 초과할 경우 위험수준을 감소시키기 위한 저감 대책이 수립되어야 한다.

표 60 FTA의 기호 및 설명

기 호		내 용
	사상(Event)	개개의 사상 보통 결함사상을 표시 논리기호의 입력 또는 출력
	기본사상(Basic Event)	더 이상 전개하지 않는 기본적 사상 논리기호의 입력 논리기호의 출력은 되지 않음
	전입(In) 전출(Out)	동일한 Fault-Tree 속에서 내용이 같은 타부분과의 사이에 전이를 나타내는 기호로서 삼각형의 상부에 선이 나오는 경우는 타부분으로의 전출을 나타낸다.
	부전개 사상(undeveloppedevent)	정보부족에 의해 분석되지 않거나 또는 분석의 필요가 없는 생략 현상을 나타내는 기호이다.

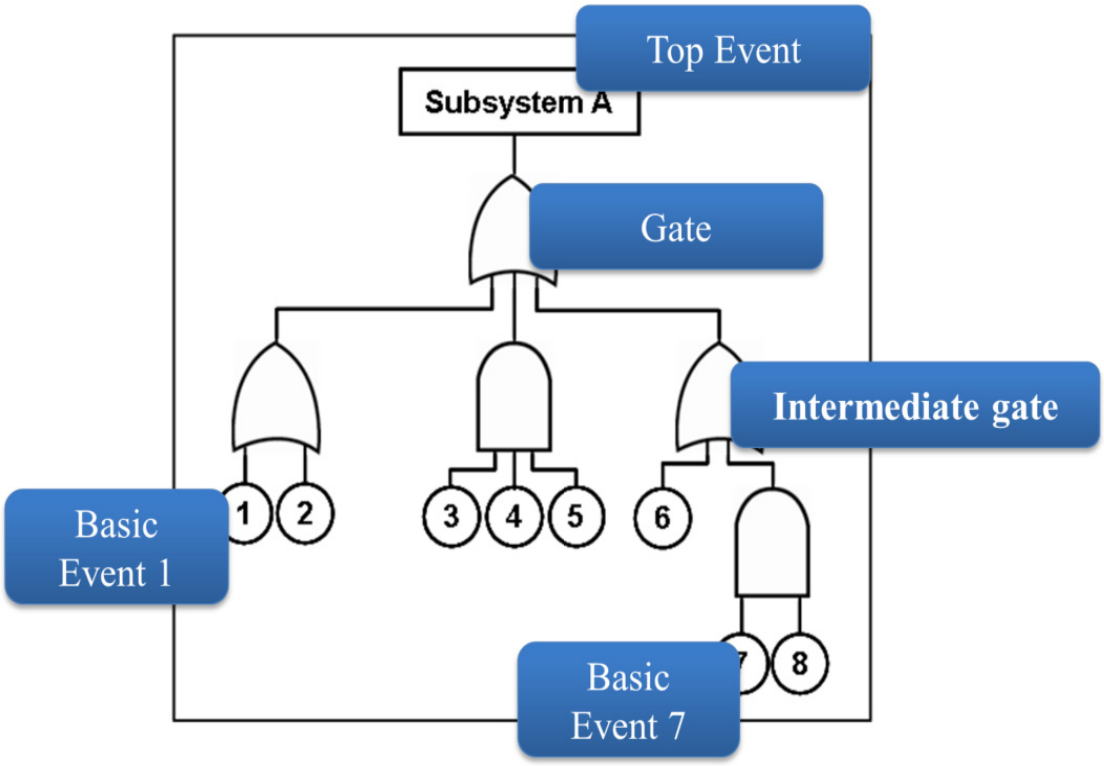


그림 74 Fault Tree 컴포넌트

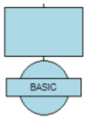
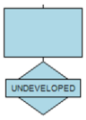
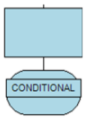
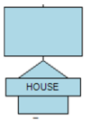
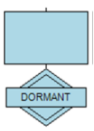
Symbol	Name	Description
	Basic event	더 이상 전개되지 않는 기본적인 사상 또는 발생확률이 단독적으로 얻어는지 기본 사상
	Undeveloped event	정보부족, 분석기술의 불충분 등으로 더 이상 전개 할 수 없는 사상, 분석 진행이 가능할 때 다시 분석
	Conditional event	An Event that only occurs if at least one other event h appened already
	House event	A (usually by the user) controlled event that only can be TRUE or FALSE
	Dormant event	Event shows a sleeping failure, not immediately recog nized.

그림 75 이벤트의 명칭과 설명

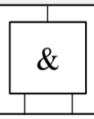
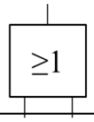
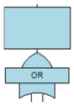
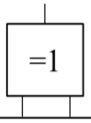
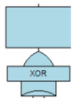
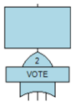
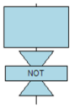
Symbol		Name	Description
		AND-Gate	Output event occurs only if all input events occur.
		OR-Gate	Output event occurs if any input events occur
		Exklusive OR-Gate	Output event occurs if exactly one of the input event occurs
		MooN-Gate (Majority VOTE Gate)	Output event occurs if m out of n input events occur
		NOT-Gate (Inverter)	Output event occurs if the input event does not occur and vice versa

그림 76 기호의 명칭과 설명

6.10.3. 체크리스트

FTA에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 FTA에 대한 검증은 [표 61]을 통해서 수행된다.

표 61 FTA 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
9. FTA 기법	9.1 대상 시스템 분석을 통한 시스템 경계 정의는 되었는가?		시스템 구조 아키텍처	
	9.2 시스템 경계정의 활동을 통해, 대상 시스템 구조 분석 자료를 확보하고 있는가?		시스템 구조 아키텍처	
	9.3 분석 대상 깊이는 정의 되었는가?		시스템 구조 아키텍처	
	9.4 기능 간 인터페이스 정보가 확보 되었는가?		시스템 기능 아키텍처	
	9.5 기능 구조 트리는 생성 하였는가?		시스템 기능 아키텍처	
	9.6 기능 구조도를 기반으로 해당 시스템의 최상위 사건(Top-Event)는 식별 되었는가?		FTA 수행 결과 보고서	
	9.7 최상위 사건 중심으로 사건 트리는 생성 되었는가?		FTA 수행 결과 보고서	
	9.8 Tree 구조가 고장 메커니즘을 잘 반영 하고 있는지 분석서는 보유 하고 있는가?		FTA 수행 결과 보고서	
	9.9 정상 사건의 발생 빈도나 확률이 반영 되었는가?		FTA 수행 결과 보고서	
	9.10 개별 사건의 발생 오류로 인해 미치는 영향을 고려한 FT 사건은 식별되었는가?		FTA 수행 결과 보고서	
	9.11 FTA를 통해서 개선사항이 식별되고 설계적으로 반영 되었는가?		FTA 수행 결과 보고서	

6.11. FRACAS

6.11.1. 개요

FRACAS는 고장 데이터를 수집하고, 고장 원인을 결정하기 위한 절차와 수행된 개선 조치에 대한 문서를 제공하는데 목적을 갖는다. 또한 고장 보고 및 분석은 시스템의 신뢰성 및 유지보수성이 요구사항에 부합되고 유지된다는 것을 보장하기 위하여 필요로 한다. FRACAS의 적용은 신규 시스템 설계 및 생산에 대하여 고장 반복 및 재발을 제어하기 위한 핵심 요소이다.

6.11.2. 절차

FRACAS 절차는 고장이 정확하게 보고되고 철저히 분석되었다는 것과 개선 조치가 고장 재발을 감소하고 예방하기 위하여 적정한 시기에 수행되었음을 보장하기 위한 규정 또는 조항을 포함해야 한다.

- 무엇이 고장을 발생하였는가?
- 언제 고장이 발생하였는가?
- 어떻게 고장이 발생하였는가?
- 왜 고장이 발생하였는가?
- 이후에 발생할 고장을 어떻게 방지 또는 제거 할 것인가?

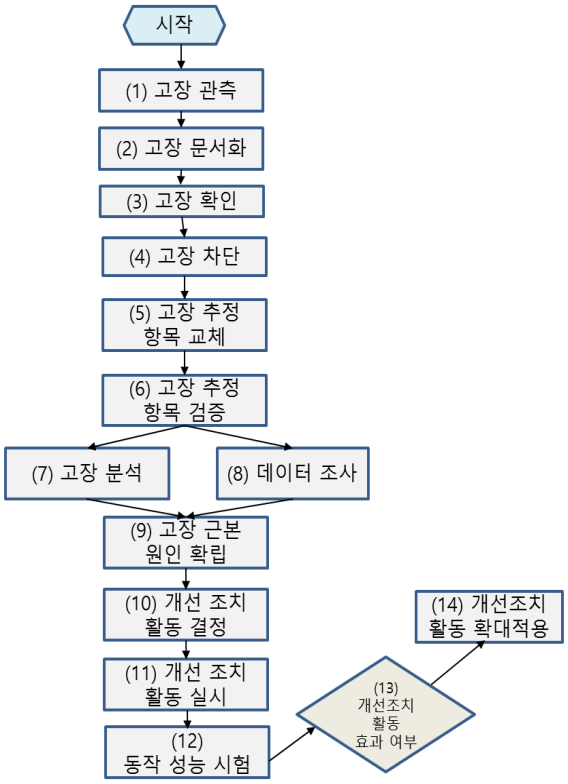


그림 77 FRACAS 흐름

6.11.3. 체크리스트

FRACAS에 대한 절차가 완료 되었다면, 실제로 해당 기법에 대한 입력물은 충분하였는지, 해당 절차는 준수하였는지에 평가를 수행해야 한다. 또한 수행 결과에 대한 산출물 리스트는 적절한지에 대한 평가를 최종적으로 점검하기 위해서 본 시스템 안전성 분석 가이드에서 제시하는 안전성 분석 기법에 대한 체크리스트를 제공한다. 따라서 FRACAS에 대한 검증은 [표 62]을 통해서 수행된다.

표 62 FRACAS 주요 항목 및 산출물 체크 리스트

안전성 분석 기법 및 주요 항목	점검사항			
	체크리스트	O / X	산출물	
			산출물명	산출물 보유여부
10. FRACAS기법	10.1 대상 시스템의 고장 데이터는 모두 수집 되었는가?		고장데이터 수집 보고서	
	10.2 식별된 고장 데이터는 문서화 되었는가?		FRACAS 수행 보고서	
	10.3 고장원인이 식별 되었는가?		FRACAS 수행 보고서	
	10.4 고장을 차단하기 위한 대책은 수립 되었는가?		FRACAS 수행 보고서	
	10.5 고장을 차단하기 위한 개선 조치 활동을 식별하였는가?		FRACAS 수행 보고서	
	10.6 동작 성능 시험 결과는 포함하고 있는가?		FRACAS 수행 보고서	
	10.7 고장을 차단하기 위한 개선 조치 활동이 효과적 이였는지 분석된 결과를 식별 하였는가?		FRACAS 수행 보고서	
	10.8 개선조치를 취하기 위한 활동의 확대 적용 가능성은 식별 되었는가?		FRACAS 수행 보고서	
	10.8 이상 현상을 제거 또는 최소화하기 위한 설계적 저감 대책은 반영되었는가?		FRACAS 수행 보고서	

제 4 절 안전성 분석의 적용 사례

본 시스템 안전성 분석 가이드에서는 표준에 따른 안전성 분석 가이드에 대한 정보를 제공하기 위해서 안전성 분석 절차 및 기법들을 적용할 대상 시스템을 선정하였다. 대상 시스템을 선정하여 본 시스템 안전성 분석 가이드의 적용 및 활용성을 높이고자 하였다.

1. 적용대상 시스템: 열차제어 시스템

열차제어 시스템은 선로의 신호, 운전, 역(Station) 등의 정보를 기관사에게 제공하고, 선행열차와의 간격 제어와 더불어 제한속도 초과 시 열차의 안전을 확보하는 기능을 담당한다[60]. 열차제어를 위한 신호방식으로 기존의 자동열차정지(ATS)방식으로부터 고속, 고밀도 운행에 대비한 자동열차방호(ATP) 방식으로 개량해 나가고 있다[61].

ATS가 지상에서 제한속도를 생성하여 기관사 실수에 대한 열차보호만을 담당하는 반면, ATP는 열차운행에 필요한 각종 정보를 지상 설비를 통해 차량으로 전송하고, 차량의 컴퓨터를 통해 속도프로파일의 생성 및 열차방호가 이루어진다. ATP를 통해 운전시격의 단축, 선로용량 증가 등의 효과를 얻을 수 있으나 이의 운용을 위해서는 발리스(Balise) 및 선로변제어기(LEU: Lineside Electronics Unit) 등의 지상설비가 신규 설치되어야 한다. 또한 기존 궤도회로 기반의 자동폐색장치(ABS: Automatic Block System), 전자연동장치, 건널목 제어장치 등과 연계 운용해야 하므로 일정 수준 이상의 유지보수 인력과 비용이 지속적으로 필요하다[62].

일일 운행횟수가 적은 지선의 경우에 고가의 제어시스템을 운용하는 것은 효율성 측면에서 문제점이 있다. 이러한 점에서 저밀도 구간인 철도지선에 대하여 선로변 시설물을 최소화하여 이로 인한 유지보수 비용을 절감하면서 기존과 동등한 운행 안전성을 보장할 수 있는 차상중심의 열차제어 시스템의 개발이 요구되고 있다[63].

열차제어 시스템의 효과적인 개발을 위해서는 선로변에 지상 설비들의 최소화하되 해당 기능들이 열차제어 시스템에서 제대로 수행될 수 있는지가 가장 중요하다. 따라서 지상 설비 최소화에 따른 차상중심 열차제어 시스템의 효율적인 설계안을 모색하고 이와 더불어 발생할 수 있는 안전 문제들을 사전에 식별하는 것이 중요하다[64].

2. 안전성 분석 수행사례

본 시스템 안전성 분석 가이드의 안전성 분석 기법을 활용하여 열차제어 시스템을 대상으로 안전성 분석 수행 사례를 기술하였다. 앞서, 열차제어 시스템이 안전성 분석이 필요하다는 것을 인지하였다.

위험원 판별은 전문지식 및 이전 경험을 바탕으로 PHA, HAZOP 등의 기법을 통해 수행될 수 있다. 시스템 안전관리의 설계단계에서 수행하는 PHA은 광범위한 정성적 분석으로서, 관리대상이 되는 주요 위험원을 식별하고, 위험원의 초기 평가와 위험원의 관리 및 후속 조치를 판단하기 위하여 수행한다.

PHA 결과를 토대로 열차제어 시스템 주요 위험원을 기반으로 한 시스템 영향 분석을 수행한다. 열차제어 시스템의 시스템 기능을 정의한 후 하나 이상의 주된 위험에 기여할 수 있는 시스템의 원인을 파악 한다. 이후 위험원에 대한 시스템 저감 대책을 결정한다.

열차제어 시스템은 무선통신망을 기반으로 한 상호정보를 교환하는 시스템으로서 통신에 대한 안전성과 신뢰성을 최우선해야 한다. 또한 열차, 지상장비, 차상장비간의 물리적 인터페이스뿐만 아니라 열차 내부에 많은 컴포넌트를 탑재하고 있기 때문에 내부간 인터페이스 식별도 매우 중요하다. 그리고 선후행 열차와의 적절한 간격 유지 및 역사에서 승객의 승하차 시 등에도 상호정보를 실시간으로 처리해야 되기 때문에 외부 전원이나 역사 설비와 같은 외부와의 인터페이스도 같이 고려해야 한다.

그리고 운용 중 위험원 발생 시나리오를 통해 기능 수행 중 위험원이 어떤 상황에서 발행되는지 식별할 수 있다. 식별된 위험원을 기반으로 FHA를 통해 저감 대책을 수립하여 컴포넌트 설계 시 적용을 통해 반영될 수 있다.

또한 FMEA는 열차제어 시스템 개발에 필요로 하는 개별 구성 및 통합체계의 안전성 평가 결과를 제공함으로써, 향후 열차제어 시스템을 구축하는데 있어서 필요한 설계적 활동과 안전성 분석의 중요한 지침으로서 정보 제공이 가능하다.

열차제어 시스템의 운영과 지원 또는 유지보수 상에 관련된 위험원을 규명하기 위해 O&SHA을 수행한다. 열차제어 시스템에 대한 운영 및 유지보수 계획, 절차를 기반으로 관련된 잠재적 위험원을 규명한다. 열차제어 시스템의 경우 열차 실시간 추적정보, 스크린도어 개/차폐 명령 등 고도화된 운영 환경을 갖추고 있다. 따라서 체계적인 운영 및 이에 따른 유지보수도 같이 이루어져야 한다.

이러한 안전성 분석 수행 사례를 통해 본 시스템 안전성 분석 가이드에서 기술한 주요 안전성 분석 기법들에 대해 전반적인 흐름을 파악할 수 있다.

2.1. 예비위험원분석(PHA)

지상 선로변 위에 위치한 열차, 지상장비, 차상장비로 크게 나뉜다는 것을 인지할 수 있다. 이러한 물리적 정보를 바탕으로 열차제어 시스템의 외부 시스템과의 연동(인터페이스) 정보에 대해 식별을 통해, 안전활동을 수행하는데 있어서 중요한 정보를 제공한다. 전체 시스템의 내·외부 구성이 식별되었다면, [그림 78]과 같이, 열차제어 시스템에 대한 물리적 범위를 정의 및 식별할 수 있다. 이후 물리적 컴포넌트에 대한 기능이 명세 되어야 한다.

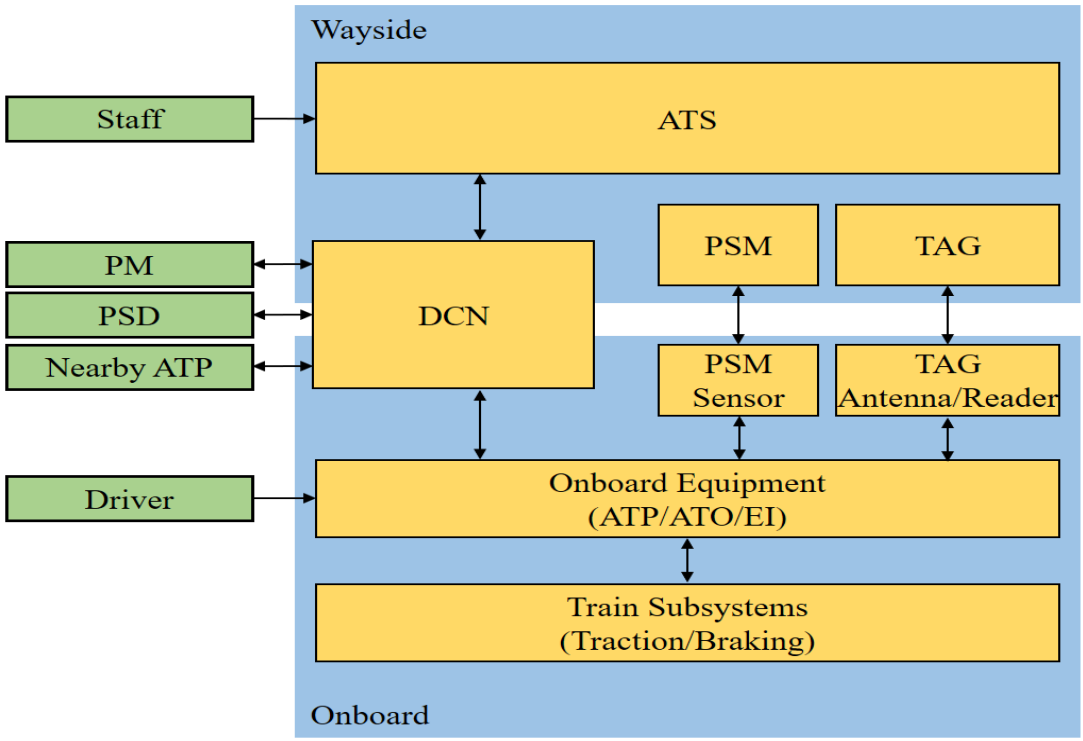


그림 78 열차제어 시스템의 PHA 수행 범위

열차제어 시스템의 핵심 장비인 ATO/ATP에 대한 기능을 아래 [표 63]과 같다. 해당 기능의 식별은 열차제어 시스템이 지니고 있는 물리적 구성, 해당 컴포넌트가 지니고 있는 기능적 구현 모드를 기반으로 식별되어 진다. 식별된 기능을 중심으로 보다 기능 상세화 과정을 거쳐 위험원 식별을 수행하게 된다.

표 63 열차제어 시스템의 ATP/ATO 기능 명세

순번	기능번호(ID)	기 능	구분
	4.3.5	**열차위치 결정**	
1	4.3.5.1	열차위치 초기화	ATP
2	4.3.5.2	열차길이 결정	ATP
3	4.3.5.3	열차위치 계산	ATP
4	4.3.5.4	열차위치 결정	ATP
5	4.3.5.5	분리된 열차 검지	ATP
6	4.3.5.6	열차위치 고장의 대응	ATP
	4.3.7	**ATP 프로파일 결정**	
7	4.3.7.1	최대 허용 속도 결정	ATP
8	4.3.7.2	임시속도제한구역 설정	ATP
9	4.3.7.3	열차 및 노선 속도에 관한 계산	ATP
10	4.3.7.4	경보곡선, 전상용 제동곡선, 비상제동곡선 계산	ATP
	4.3.11	**출입문 제어**	
11	4.3.11.1	올바른 측면에 있는 열차 출입문 개방 제어	ATP
12	4.3.11.2	열차 출입문 닫힘 상태 및 PSD 닫힘 상태 제어	ATP
13	4.3.11.3	역구내 출입문 제어	ATP
14	4.3.11.4	역구내 정위치 정차	ATP
	4.3.14	**열차의 상태 관리**	
15	4.3.14.1	열차의 주요장비 고장상태 확인	ATP
16	4.3.14.2	열차의 화재 및 연기검지 상태 확인	ATP
17	4.3.14.3	인접 열차의 비상 제동 명령 수신시 차량 비상제동	ATP
	4.4.1	**열차의 Sleep 상태 명령/해제**	
18	4.4.1.1	열차의 Sleep 상태 해제	ATO
19	4.4.1.2	열차의 Sleep 상태 명령	ATO
	4.4.2	**ATO 프로파일 결정**	
20	4.4.2.1	ATO 프로파일 결정	ATO
	4.4.3	**열차 자동 제어**	
21	4.4.3.1	ATO 프로파일에 따른 열차이동	ATO
22	4.4.3.2	ATO 인칭 제어	ATO
	4.4.4	**ATO 프로파일에 따라 가속 또는 감속 신호 송출**	
23	4.4.4.1	ATO 프로파일에 따라 가속 또는 감속 신호 송출	ATO
	4.4.5	**차상의 현시 화면에 데이터 전송**	
24	4.4.5.1	차상의 현시 화면에 데이터 전송	ATO
	4.4.6	**출입문 제어**	
25	4.4.6.1	출입문 열림 명령	ATO
26	4.4.6.2	출입문 닫힘 명령	ATO
27	4.4.6.3	특수한 상황에서도 출입문 열림/닫힘 기능 명령	ATO
· · ·	· · ·	· · ·	· · ·

2.1.1. 적용 범위 및 제한사항

열차제어 시스템을 구성하는 ATP/ATO의 기능별 PHA는 ATP/ATO의 안전기능 실패 확률을 정량적으로 확보할 수 없다. 따라서 ATP/ATO에서 발생되는 위험측 제어는 사고발생의 원인으로 되는 것으로 가정하고, 안전측 제어에 의한 원인 및 저감 대책의 분석은 제외하였다.

위험원으로부터 사고발생으로 연결될 확률은 Event Tree Analysis(ETA)를 통해 평가해야 하지만, PHA에서는 발생된 모든 위험원이 사고를 발생시키는 것으로 가정(위험원 발생빈도=사고발생빈도)하며, ETA는 FMECA 또는 HAZOP에서 실시한다. 위험원을 식별할 때는 지금까지 고려되지 않았던 잠재적인 위험원의 상호작용에 대해 항상 고려해야 하며, 정상상태와 운용단계만으로 국한시키지 말고 철도에 설치된 시점부터 유지보수, 업그레이드를 포함하여 최종 사용중지까지의 시스템 전 생명주기에 걸친 모든 측면을 고려해야 한다.

표 64 열차위치 초기화의 PHA (예시)

Code	위험원(Hazard)	원인(Cause)	시스템레벨 결과	최종결과	저감대책
PHA_01	차상장치 시동과정 이후 열차 위치 초기화의 실패	1. I/F H/W 결함 2. S/W 결함 3. 연산부 결함	위치 초기화 실패로 정상운행 시작 불가	운행지연	1. I/F의 결함허용 설계 2. S/W건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증
PHA_02	열차위치 초기화의 실패	1. I/F H/W 결함 2. S/W 결함 3. 연산부 결함	진입하는 열차 위치 초기화 실패로 예측불가	열차충돌	1. I/F의 결함허용 설계 2. S/W건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증
PHA_03	차상장치 고장으로부터 회복 시 열차 위치 초기화의 실패	1. I/F H/W 결함 2. S/W 결함 3. 연산부 결함	위치 초기화 실패로 정상운행 시작 불가	운행지연	1. I/F의 결함허용 설계 2. S/W건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증

2.1.2. PHA 위험원 식별

열차제어 시스템을 대상으로 위험원을 식별하기 위해 설계 산출물을 활용하여 물리적 컴포넌트를 식별하였다. 식별된 물리적 컴포넌트에서 개별 컴포넌트가 지니고 있는 기능을 식별하여 해당 기능이 발생할 수 있는 기능을 보다 상세화 하여 기능에서 발생할 수 있는 다양한 모드를 기반으로 보다 상세화 과정을 거쳐 제시된다. 이러한 과정을 거쳐 시스템 및 컴포넌트로부터 위험원을 식별하게 된다. 시스템이 지닌 특정 기능에 대해 다음과 같은 가정을 기반으로 하여, 위험원을 식별하였으며, 주된 위험원 이외의 위험원은 안전하다고 가정한다.

- 물리적 궤도 레이아웃을 나타내는 가이드웨이는 선로전환기에 의해 상호 연결되고 가이드웨이 말단에 의해 종료된 궤도로 구성된다.
- 열차는 함께 연결된 하나 이상의 차량으로 구성된다.
- 차상 승객들이 열차 밖으로 떨어지는 것을 방지하기 위해 출입문이 장착되어 있다.
- 이동권한을 발하여 열차 이동에 대한 안전 조정이 달성된다. 특정 열차에 대한 이동권한은 다음과 같은 세 가지 요소로 구성된다.
  - 인가된 이동방향
  - 인가된 위치 한계
  - 인가된 제한속도

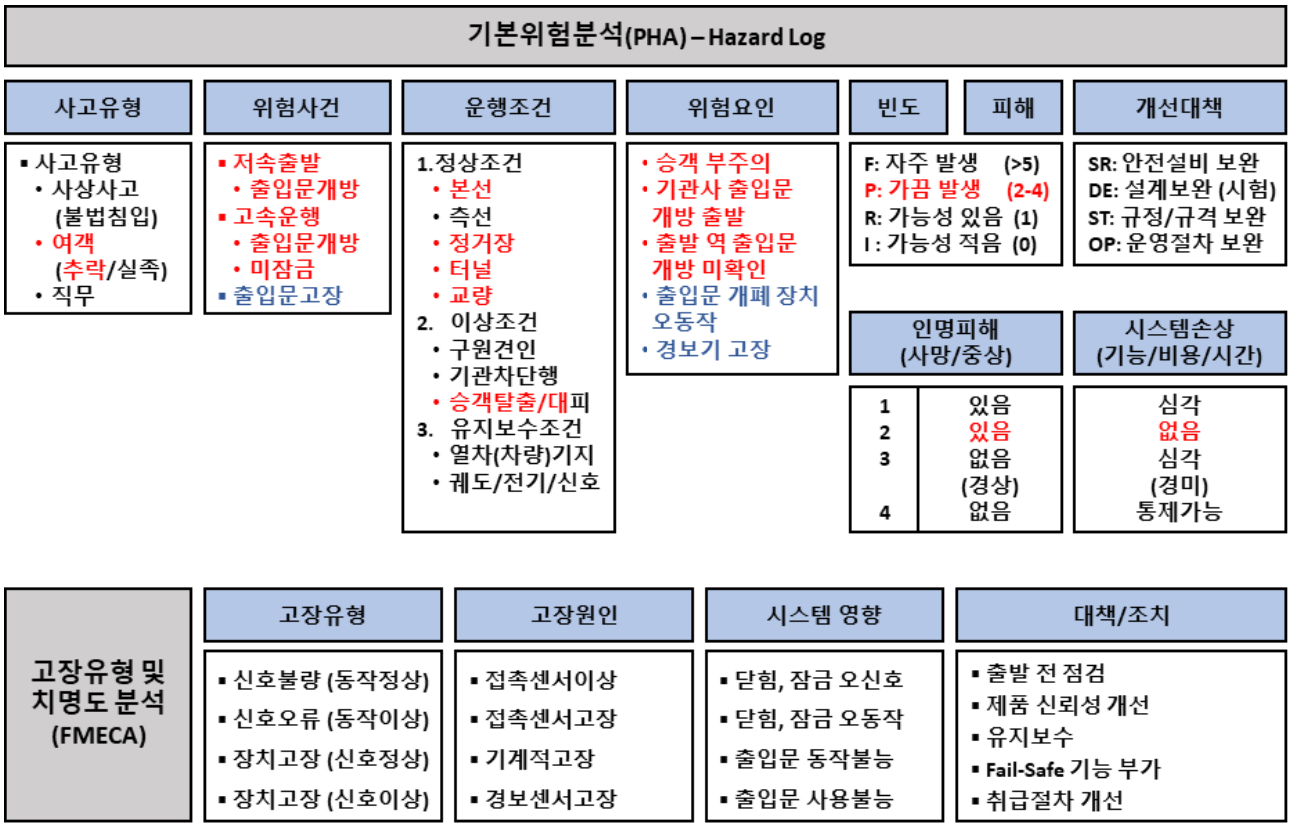


그림 79 철도 인명사상 사고에 적용 된 예비위험원분석(PHA) 사례

2.1.3. PHA 수행 결과

열차제어 시스템에 대한 PHA를 수행하여, PHA를 통해 식별된 위험원은 [표 65]를 통해서 알 수 있다. 각 위험원을 유발 시키는 최상위 원인인 기여 인자를 도출하였다. 이러한 과정을 거쳐 위험도를 저감시키기 위한 저감대책을 결정, 최소한 허용 가능 수준으로 감소시키고 잔여 위험도를 결정하기 위해 시스템 원인 분석을 기록하여야 한다. 분석 결과 관련된 모든 위험원과 원인에 대한 대책 수립으로 위험도 수준이 '무시 가능한' 수준으로 정의 되었다. 이는 안전계획서의 위험도 허용 지침에 따라 허용 가능한 범위에 포함된다. 또한, PHA을 통해 수립된 저감대책은 반드시 설계에 적용되고 관리해야 할 것이다. 추후 PHA에 따라 결정된 위험도를 근거로 안전 무결성 등급(SIL) 할당이 이루어 져야 한다.

따라서 열차제어 시스템의 ATP/ATO 장치는 제시된 RAMS 요구사항을 만족하도록 설계, 제작, 설치되어야 한다. 본 시스템 안전성 분석 가이드에서 제시된 RAMS 목표 및 PHA 안전성 분석 기법은 향후 프로젝트 수행과정에서 발생하는 변경 또는 제작설치 업체와 ISA와의 협의를 통해 갱신 될 수 있다.

다음은 본 장에서 수행한 열차제어 시스템의 ATP/ATO의 PHA 결과, 위험원이 도출되었으며, 열차충돌, 열차 탈선 및 인명 사상과 관련된 위험원은 [표 65]와 같이 37개 위험원이 도출 되었다. 앞선 활동을 통해, 시스템 및 컴포넌트가 지니고 있는 기능을 보다 상세화 하여 해당 기능이 오류 및 오작동 일어날 시에 대한 사항을 위험원이라고 규정하여야 한다. 해당 개별적 위험으로부터 최종적으로 초래할 수 있는 사고결과를 분석하게 된다. 철도 신호 시스템의 오작동 및 고장으로 발생할 수 있는 사고의 결과에 대해서 열차충돌, 인명 사상 운행지연 등의 카테고리로 식별하여 평가를 수행하였다.

표 65 열차제어 시스템 ATP/ATO의 PHA 결과

위험원코드	위 험 원	사고결과
PHA_01	운행중인 열차에 대해 열차위치 초기화의 실패	열차충돌
PHA_02	인접 차상장치에게 열차의 길이 전송 실패	열차충돌
PHA_03	열차의 위치 계산 실패	열차충돌
PHA_04	이동거리의 정해진 상대오차 초과	열차충돌
PHA_05	차상장치가 열차위치를 ATS에 전송 실패	열차충돌
PHA_06	운행 중인 분리된 열차 검지 실패	열차충돌
PHA_07	열차 무결성 장치에 의해 열차 무결성 확인 실패	열차충돌
PHA_08	열차 MMI와의 인터페이스를 통해 열차 무결성 손실 정보 제공 실패	열차충돌
PHA_09	열차 위치추적 실패를 검지했을 때, 차상장치의 비상제동명령의 실패	열차충돌
PHA_10	차상장치의 열차위치 실패에 대한 필요한 정보 제공 실패	열차충돌
PHA_11	열차위치 보고가 없는 경우 차상장치의 방호구역 결정 실패	열차충돌
PHA_12	차상장치의 선로 위치 별 최대 허용 속도 결정 실패	열차충돌
PHA_13	차상장치의 철도차량의 각 형식별 최대 허용 속도 결정 실패	열차충돌
PHA_14	차상장치의 ATS로부터 운행명령에 따라 임시속도제한구역 설정 실패	열차충돌
PHA_15	새로운 임시속도제한설정이 동일위치에서 시행 중인 기존 속도제한을 제거	열차충돌
PHA_16	수동으로 설정된 임시속도제한이 차상장치가 제공한 명령에 의해 설정 및 해제의 실패	열차충돌
PHA_17	차상장치의 열차 및 노선 속도에 관한 정보 수집 실패	열차충돌
PHA_18	차상장치의 노선상의 모든 위치별로 열차에 대해 허용된 속도를 계산 실패	열차충돌
PHA_19	차상장치의 경보곡선, 전상용 제동곡선 및 비상제동곡선의 계산 실패	열차충돌
PHA_20	생성된 동적 속도 프로파일을 ATO로 전송 실패	열차충돌
PHA_21	차상장치는 열차방위에 따라 열차 출입문만 개방되도록 허가의 실패	인명사상
PHA_22	차상장치는 올바른 쪽의 출입문만 개방되도록 허가의 실패	인명사상

위험원코드	위 험 원	사고결과
PHA_23	역에 정차한 열차가 없을 때 플랫폼 출입문이 개방되어 있다고 검지되면 차상장치의 방호영역 설정 실패	인명사상
PHA_24	출입문 닫힘 상태가 손실되고 열차가 역 사이에 정차한 경우, 차상장치의 열차의 부동화 명령 실패	인명사상
PHA_25	출입문 폐쇄상태가 손실되고 열차가 운행 중인 경우, 차상장치의 비상제동명령 발령 실패	인명사상
PHA_26	화재/연기 검지의 경우, 차상장치의 관련 정보가 현시되고 기록될 수 있도록 ATS와 연계된 정보 제공 실패	인명사상
PHA_27	차상장치의 화재/연기 검지 정보를 열차 MMI에 정보 제공 실패	인명사상
PHA_28	차상장치는 인접 차상장치로부터 비상 제동 명령을 수신 시 차량 비상제동 투입 실패	열차충돌
PHA_29	차상장치의 EB/FSB 투입 시 ATO로 EB/FSB 상태 전송 실패	열차충돌
PHA_30	ATO는 다음 정차 지점에서 정차할 수 있도록 하기 위한 운행속도 프로파일 결정 실패	열차충돌
PHA_31	ATO는 ATS의 명령에 의해 열차의 정차 및 통과가 가능하도록 운행 속도 프로파일 결정 실패	열차충돌
PHA_32	운행 속도 프로파일에 따라 철도차량에게 가속 또는 감속 신호 송출 실패	열차충돌
PHA_33	출입문이 자동열림인 경우 ATO의 열차 출입문 및 플랫폼 출입문 열림 신호 송출 실패	인명사상
PHA_34	출입문이 수동열림인 경우 ATO의 열차 출입문 및 플랫폼 출입문 열림 신호 송출 실패	인명사상
PHA_35	출입문이 자동닫힘이면서 ATS로부터 닫힘 명령을 받은 경우, ATO의 열차 출입문 및 플랫폼 출입문 닫힘 신호 송출 실패	인명사상
PHA_36	출입문이 수동닫힘인 경우 ATO의 열차 출입문 및 플랫폼 출입문 닫힘 신호 송출 실패	인명사상
PHA_37	특정 출입문 세트의 고장에 의해 수동으로 사용 중지시킨 경우, ATO는 전체 출입문 세트의 자동열림/닫힘 기능 작동 실패	인명사상

2.1.4. 열차제어 시스템의 PHA 위험원과 기여 인자와의 관계 정리

시스템 수준에 따른 안전성 분석 산출물 간의 추적성을 확보하기 위해서 아래[표 66]과 같이, 매트릭스 테이블을 확보하는 것이 안전성 분석에 유용한 정보를 제공할 수 있다. 따라서 본 단계에서는 열차제어 시스템의 위험원과 기여 인자와의 상관관계를 확인한다. 위험원과 기여 인자와의 관계는 [표 66]의 매트릭스 구조의 활용을 통해 연동 정보를 식별 할 수 있게 된다.

표 66 위험원 매트릭스

원인 제공 요인			주요 위험원						
PHA_ID	설명	차상 신호 시스템에 의해 완화 되거나 제어됨	A	B	C	D	E	F	G
01	선로전환기 오작동	YES	O	O		O	O		O
02	건널목 차단기 오작동	YES						O	
03	진로 요청에 대한 승인 없이 열차가 이동함	YES	O	O		O		O	O
04	열차가 의도와는 달리 연결되지 않음		O					O	
05	의도하지 않은 열차 움직임		O	O	O		O	O	O
06	선로상의 미확인 열차	YES	O	O	O	O	O	O	
07	비상정지 장치 고정		O	O	O	O	O	O	
08	열차이동에 방해 받음								O
09	안전하지 않은 상태로의 모드전환		O	O	O	O	O	O	O

2.2. 시스템 위험원 분석(SHA)

2.2.1. 적용 범위 및 제한사항

열차제어 시스템의 기능에 영향을 미치는 장비 및 기능과 관련한 사항에 적용된다. 다음의 사항에 의해 야기된 위험은 포함되지 않는다.

- 열차제어 시스템 경계를 벗어나 위치의 외부장비
- 정의된 사양을 벗어나는 환경조건
- 자연적 대재난(홍수, 지진 등)
- 테러 및 의도적인 기물 파손

2.2.2. ATP 기능 요구사항 분석

ATO 차상 장치는 최소한 ATP에서 설정한 안전 조건 내에서 아래와 같은 ATO 기능을 지원해야 한다. 이때, 열차제어 시스템의 차상간 양방향통신은 ATO 기능을 지원하는데 충분해야 한다. 차상 ATO 장치의 기능 요구사항은 아래 [표 67]과 같다.

표 67 ATO 장치의 기능 요구사항

ID	기능 요구사항
REQ_01	ATO 프로파일결정
REQ_02	자동속도제어
REQ_03	열차 역 통과제어
REQ_04	열차 역 정차제어
REQ_05	인칭제어
REQ_06	출입문/PSD제어

2.2.3. SHA 수행 결과

ATO 장치의 기능 요구사항을 명시한 [표 67]을 중심으로 시스템 수준에서 요구되는 기능과 그로인해 발생될 수 있는 위험원과 그에 따른 결과를 바탕으로 [표 68]과 같은 결과를 식별하였다.

표 68 차상 ATO 시스템 위험원

위험원 ID	위 험 원	결 과
SHA_11	ATO 프로파일의 위험측 생성 및 결정	서비스 지연
SHA_12	ATO 프로파일의 안전측 생성 및 결정	서비스 지연
SHA_13	차상ATP 출력차단으로 인한 열차위치 보고중단	충돌
SHA_14	운행 중 ATO 프로파일 대비 과속운행	충돌
SHA_15	운행 중 열차분리 검지실패	서비스 지연
SHA_16	운행 중 ATO 프로파일 대비 저속운행	서비스 지연
SHA_17	정차역 정보 미수신 또는 수신 오류로 인해 정차역 을 오류로 통과	승객의 불편함
SHA_18	통과역 정보 미수신 or 수신 오류로 인해 통과역을 오류로 정차	서비스 지연
SHA_19	자동인칭제어의 기능실패	서비스 지연
SHA_20	수동인칭제어의 수행불가	서비스 지연

2.3. 인터페이스 위험원 분석(IHA)

2.3.1. 적용 범위 및 제한사항

열차제어 시스템의 인터페이스에 영향을 미치는 장비 및 기능과 관련한 사항에 적용되며 [그림 78]과 같다. 외부 인터페이스 중 인간과 관련된 부분은 O&SHA에서 진행하기로 하며, IHA에서는 물리적인 연결과 네트워크 상 신호를 주고 받는 장비에 한해 분석 대상으로 삼는다. 다음의 사항에 의해 야기된 위험은 포함되지 않는다.

- 외부 장비의 인터페이스를 위해 필요한 데이터 송수신 정보 없음
- 외부 장비의 인터페이스를 위해 필요한 인터페이스 형태 지정(예, 프로토콜, Network 등) 정보 없음
- 외부 장비의 인터페이스를 위해 필요한 물리적 환경 구축 및 조건 정보 없음

2.3.2. 인터페이스 분석

열차제어 시스템 분석을 통해 도출한 시스템 아키텍처와 시스템 요구사항 분석을 통해 도출한 인터페이스 요구사항을 기반으로 열차제어 시스템의 인터페이스를 분석한다. 열차제어 시스템의 인터페이스 분석 시 주요 분석 대상은 ATS/ATP/ATO로 해당 장비에 많은 컴포넌트들이 연결되어 수행되도록 설계되었기 때문에 상당부분의 인터페이스가 존재할 수 있을 것이라는 예측을 할 수 있다.

또한 해당 장비들이 가지고 있는 기능별로 필요한 타 장비와의 연계성을 고려해본다면 연동 정보들 또한 다양할 것으로 추측할 수 있다. 따라서 우선 시스템 내부/외부간의 인터페이스를 먼저 식별한다. 내부/외부간의 인터페이스가 식별된 후로는 장비와 장비마다 연결된 흐름이 단방향인지 양방향인지를 식별한다. 흐름을 파악하는 이유는 어떠한 장비가 타 장비와 인터페이스 시 필요한 정보를 자체적으로 생성해야하는지 아니면 입력을 받아 수행해야 하는지를 파악하는데 용이하다. 또한 이러한 흐름은 인터페이스 분석 대상이 맞는지 아닌지 판단하는데도 도움이 된다.

인터페이스 방향을 정리한 후에는, 각 방향마다 어떠한 정보를 주거나 받는지를 식별하고, 어떻게 전달할 것인지를 파악한다. 열차제어 시스템의 경우 열차, 지상장비, 차상장비간의 네트워크나 신호를 통해 주고 받거나 하는 것이 대부분이다. 또한 이러한 네트워크나 신호 같은 경우, 다양한 정보를 전달할 수 있으므로 이러한 내용을 정확히 파악할 수 있도록 인터페이스 설명을 통해 기술하였다. 열차제어 시스템의 인터페이스 분석 결과를 [표 69]에 기술하였다.

표 69 열차제어 시스템의 인터페이스 분석 결과

인터페이스 ID	인터페이스				
	인터페이스 설명	인터페이스 메인	인터페이스 대상	데이터/정보	인터페이스 형태
IN_001	ATS는 열차운행계획 관리 정보, 열차운행상태 감시 정보, 원격제어 정보를 차상 신호장치로 전송	ATS	차상 신호장치	열차운행계획관리정보 열차운행상태감시정보 원격제어정보	ATS안테나
IN_002	ATP는 열차위치 실시간 추적정보, 열차이동권한을 선/후행 ATP로 전송	ATP	선/후행 ATP	열차위치실시간 추적정보 열차이동권한	발리스
IN_003	ATP는 속도프로파일을 제동장치로 전송	ATP	제동장치	속도프로파일	무선통신
IN_004	ATP는 열차속도, 주행할거리를 MMI로 전송	ATP	MMI	열차속도 주행할거리	무선통신
IN_005	ATP는 출입문 개폐명령을 PSD에 전송	ATP	PSD	출입문 개폐명령	DCN
IN_006	ATO는 열차속도를 ATP에 전송	ATO	ATP	열차속도	Onboard Equipment
IN_007	ATO는 정위치정차 정보를 ATP에 전송	ATO	ATP	정위치정차 정보	Onboard Equipment
IN_008	ATO는 출입문 개방명령을 PSD에 전송	ATO	PSD	출입문 개방명령	무선통신
IN_009	EI는 열차진로 신호를 열차관제 설비에 전송	EI	열차관제 설비	열차진로 신호	분기기

2.3.3. IHA 수행 결과

인터페이스 분석 결과를 토대로 인터페이스 위험원을 식별한다. 인터페이스의 위험원은 인터페이스를 대상으로 HAZOP을 통해 식별되며 전문가와의 브레인스토밍을 통해 적절한 가이드워드를 선정하였다. 또한 선정된 가이드워드를 통해 인터페이스의 위험원마다 발생할 수 있는 영향을 분석하였다. 또한 식별된 위험원에 대한 원인을 파악하여 이에 대한 저감 대책을 기술하였으며, 해당 저감 대책은 설계 시 구현할 수 있으며, 안전 요구사항에 반영되어 설계에 반영되어야 한다. 그리고 식별된 위험원으로부터 발생할 수 있는 영향 분석을 토대로 초기 위험도를 평가하고, 저감 대책을 통해 해당 위험원에 대한 위험도가 얼마나 감소하였는지를 평가한다. 이러한 열차제어 시스템의 인터페이스 위험원 분석 결과를 [표 70]에 기술하였다.

표 70 인터페이스 위험원 분석 결과 (예시)

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
ATS는 열차운행계획 관리 정보, 열차운행상태 감시 정보, 원격제어 정보를 차상 신호장치로 전송	No	열차운행계획 관리 정보, 열차운행상태 감시 정보, 원격제어 정보 전송 불가	열차운행 계획, 상태 감시 및 원격제어 불가로 인하여 곡선구간 또는 하구배 구간에서 열차탈선사고 발생 가능(사상자 발생 가능)	4	4	수용불가	2	4	수용	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATS 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										차상 신호장치에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
	Part of	열차운행계획 관리 정보, 열차운행상태 감시 정보, 원격제어 정보의 일부 전송	열차운행 계획, 상태 감시 및 원격제어 불가로 인하여 곡선구간	4	4	수용불가	2	4	수용	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
			또는 하구배 구간에서 열차탈선사고 발생 가능(사상자 발생 가능)							ATS 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										차상 신호장치에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
ATP는 열차위치 실시간 추적정보, 열차이동권한을 선/후행 ATP로 전송	No	열차위치 실시간 추적정보, 열차이동권한 전송 불가	열차위치 실시간 추적정보, 열차이동권한 한 전송 불가로 인해 열차위치정보, 열차정지위치 파악 불가능으로 인한 추돌 및 충돌 사고 발생 가능(사상자 발생 가능)	3	4	수용(H)	2	4	수용	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										선/후행 ATP에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
	Part of	열차위치 실시간 추적정보, 열차이동권한의 일부 전송	열차위치 실시간 추적정보, 열차이동권한 전송 지연으로 인해 열차위치정보, 열차정지위치 파악 불가능으로 인한 추돌 및 충돌 사고 발생 가능(사상자 발생 가능)	3	4	수용(H)	2	4	수용	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										선/후행 ATP에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
	Wrong	잘못된 열차위치 실시간 추적정보, 열차이동권한 전송	열차위치 실시간 추적정보, 열차이동권한 전송 불가로 인해 열차위치정보, 열차정지위치 파악	3	4	수용(H)	2	4	수용	ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										선/후행 ATP에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
			불가능으로 인한 추돌 및 충돌 사고 발생 가능(사상자 발생 가능)								1.2EMC시험을통해부합성확인
ATP는 속도프로파일을 제동장치로 전송	No	속도프로파일 전송 불가	속도프로파일 전송 불가로 인해 기관사의 열차정지위치 파악 불가로 인한 비상제동 가능	2	3	수용	1	3	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										제동장치에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
	Part of	속도프로파일의 일부 전송	속도프로파일 전송 지연으로 인해 기관사의	2	3	수용	1	3	허용가능	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
			열차정지위치 파악 불가로 인한 비상제동 가능							케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										제동장치에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
ATP는 열차속도, 주행할거리를 MMI로 전송	No	열차속도, 주행할거리 전송 불가	관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										MMI에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
ATP는 출입문 개폐명령을 PSD에 전송	No	출입문 개폐명령 전송 불가	관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
											1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										PSD에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
ATO는 열차속도를 ATP에 전송	No	열차속도 전송 불가	열차속도 전송 불가로 인해 열차위치정보, 열차정지위치 파악 불가로 인한 추돌 및 충돌 사고 발생 가능(사상자 발생 가능)	3	4	수용(H)	2	4	수용	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										내부 부품 소손	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										ATO 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
ATO는 정위치정차 정보를 ATP에 전송	No	정위치정차 정보 전송 불가	관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										내부 부품 소손	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										ATO 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										ATP 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
ATO는 출입문 개방명령을 PSD에 전송	No	출입문 개방명령 전송 불가	관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										ATO 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										PSD에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
EI는 열차진로 신호를 열차관제 설비에 전송	No	열차진로 신호 전송 불가	열차진로 신호 전송 불가로 인해 열차진로 파악 불가로 인한 비상제동 가능	4	2	수용	2	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										EI 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										열차관제 설비에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인

인터페이스	가이드 워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
	Part of	열차진로 신호의 일부 전송	열차진로 신호 전송 지연으로 인해 열차진로 파악 불가로 인한 비상제동 가능	4	2	수용	2	2	허용가능	전원없음	1.1상용전원을평시에장비에공급및공급중단시UPS를통한전원공급설계 1.2케이블은내부식성,방수,방습,염해방지 특성이있는제품적용 1.3전동차내비상전원장치와연결
										통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										EI 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블사용 1.2케이블에태그부착으로오결선예방
										열차관제 설비에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인

2.3.4. IHA 결론

열차제어 시스템에 대한 IHA 결과 총 14개의 위험원이 식별되었다. 인터페이스에 대한 모든 위험원이 식별되었으며, 외부 인터페이스와 관련된 인터페이스에 대한 원인도 식별하였다. 식별된 위험원에 대해 초기 위험도 평가를 통해 추가적인 위험도 저감대책이 요구되는 위험원에 대해서는 해당 위험도를 최소 허용 가능한 수준으로 저감시키기 위한 안전 요구사항을 수립함으로써 최종적인 잔여 위험도 평가를 통해서는 해당 위험원이 허용 가능한 수준이하로 위험도가 저감되었음을 확인하였다.

2.4. 결함 위험원 분석(FHA)

2.4.1. 적용 범위 및 제약사항

열차가 정상적으로 주행 중인 경우 ATS는 열차 및 기타 설비의 운행상태를 지속적으로 감시하며, 감시결과에 맞추어 필요한 경우 열차속도변경을 요구하거나 열차 내 전장품 등을 원격 제어한다. 이 같은 활동은 열차가 출발하여 다음 역에서 정차할 때까지 반복된다. 열차가 정차역의 정차지점에 정차하기 위해서 ATS에서 담당한다.

열차 진로 제어를 담당하는 EI는 영업주행이 결정된 열차가 역사에 정차하고 있는 상태에서 열차임무와 열차에 필요한 진로를 설정한다. ATP의 기능은 크게 2가지로 나눌 수 있다. 첫 번째는 선행열차와 후속 열차 간 충돌사고를 방지하기 위해서 열차 간 안전간격을 제어하는 것이며, 두 번째는 불특정 발생하는 이벤트에 대하여 승객과 열차의 안전을 보장하는 것이다. 열차제어 시스템의 운용에 따른 기능 흐름도는 [그림 80]과 같다.

FHA는 통상 FMEA나 FMECA의 분석 양식과 비슷하지만, 대상요소가 고장발생이 직접적으로 재해발생에 연결되는 것에 한정된다. 또한 컴포넌트의 고장이 발전하여 다른 컴포넌트의 고장이나 기능저하 같은 2차 고장을 발생시킬 수 있는 운용상의 요인이나 환경적인 요인까지 포함해야 한다.

시스템 설계 초기에 수행된 FHA 결과를 과신하여서는 안 되며, 시스템에 관한 중요한 정보들이 추가될 때마다 시스템의 안전성이 여전히 확보되고 있는지를 확인할 수 있도록, 가급적 자주 반복적으로 수정되어야 한다.

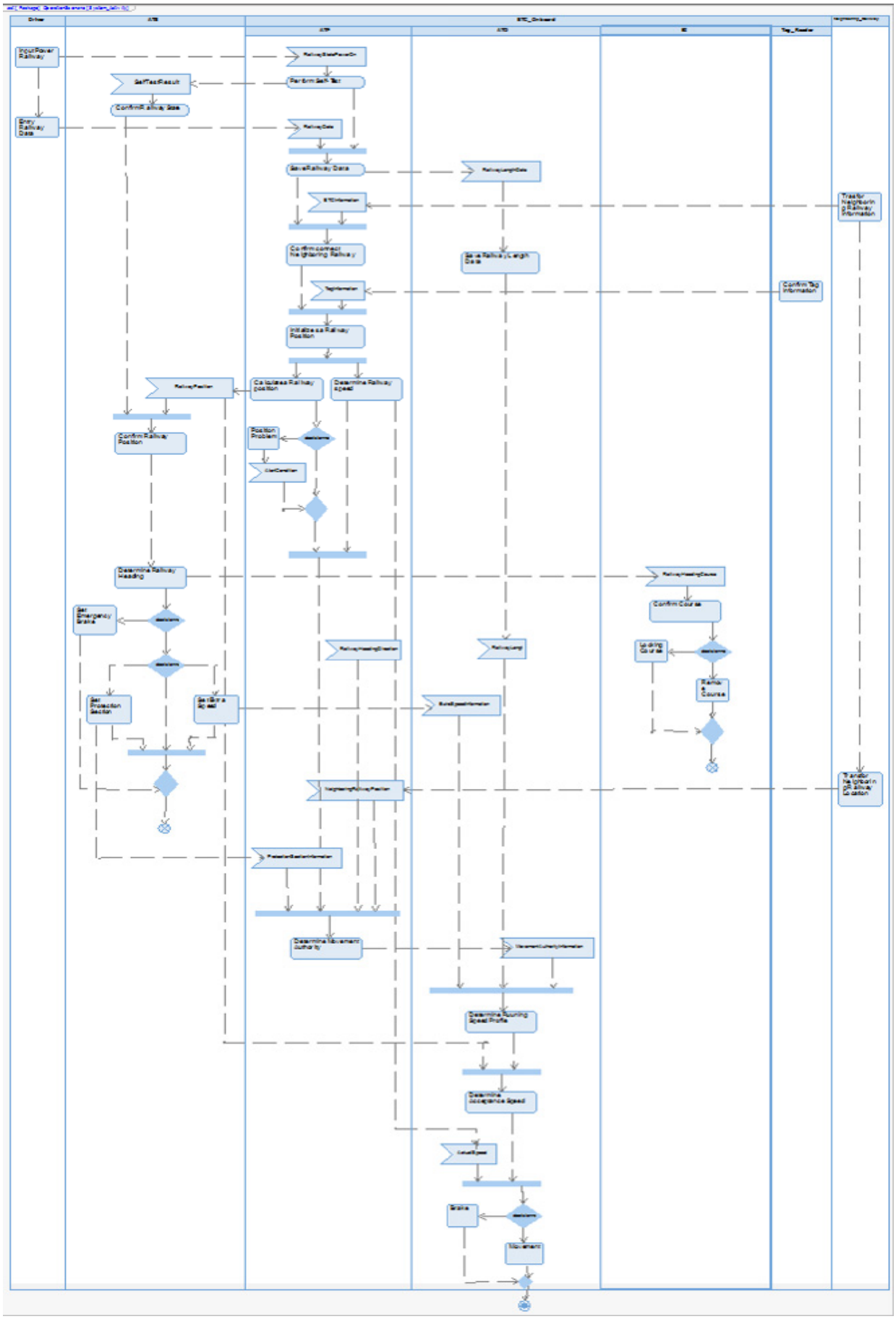


그림 80 열차제어 시스템의 운용시나리오 분석을 위한 모델기반 거동 분석

2.4.2. FHA 수행 결과

FHA 수행은 기능 흐름도를 통해 각 단계별 생성문서들과 최종적으로 생성될 열차제어 시스템의 안전성 입증 문서인 종합저감 대책기술서의 문서 확인을 토대로 위험이 제거 또는 저감되었는지 확인해야 한다. FHA을 통해서, 열차제어 시스템의 기능 흐름도에서 발생할 수 있는 위험원들을 식별하여 고장 모드 및 비상 운용 시나리오를 생성하는데 도움을 줄 수 있다.

(1) Hazard Identification 접근

시스템의 시스템수준 기능 모델을 토대로 위험원 리스트(Hazard List)를 모든 고장 모드로부터 수집한다. 열차제어 시스템의 위험원 분류는 기능적 고장에 대한 부분으로서 주로 다음과 같이 구분할 수 있다.

- 손실(Loss)
- 오작동(Error)
- 의도적 벗어남(Intentional Deviation)
- 정상보다 빠름(Too early)
- 정상보다 느림(Too late)
- 기타(Other)

아래의 식별된 위험원은 열차제어 시스템의 기능 흐름도 및 고장모드로부터 식별된 위험원들을 토대로 상기 기준에 맞춰 아래 [표 71]와 같이 식별되었다.

표 71 위험원 식별

#1 정보의 손실	
HAZ005	인접 차상장치에게 열차의 길이 전송 실패
HAZ008	차상장치가 열차위치를 ATS에 전송 실패
HAZ011	열차 MMI와의 인터페이스를 통해 열차 무결성 손실 정보 제공 실패
HAZ012	열차 위치추적 실패를 검지했을 때, 차상장치의 비상제동명령의 실패
HAZ013	차상장치의 열차위치 실패에 대한 필요한 정보 제공 실패
HAZ014	열차위치 보고가 없는 경우 차상장치의 방호구역 결정 실패
HAZ020	차상장치의 열차 및 노선 속도에 관한 정보 수집 실패
HAZ023	생성된 동적속도 프로파일을 ATO로 전송 실패
HAZ024	차상장치는 열차방위에 따라 열차 출입문만 개방되도록 허가의 실패
HAZ025	차상장치는 올바른 쪽의 출입문만 개방되도록 허가의 실패

HAZ030	출입문 닫힘 상태가 손실되고 열차가 역 사이에 정차한 경우, 차상장치의 열차의 부동화 명령 실패
HAZ031	출입문 폐쇄상태가 손실되고 열차가 운행 중인 경우, 차상장치의 비상제동명령 발령 실패
HAZ036	화재/연기 검지의 경우, 차상장치의 관련 정보가 현시되고 기록될 수 있도록 ATS와 연계된 정보 제공 실패
HAZ037	차상장치의 화재/연기 검지 정보를 열차 MMI에 정보 제공 실패
HAZ039	차상장치의 EB/FSB 투입 시 ATO로 EB/FSB 상태 전송 실패
#2 기능 오작동	
HAZ002	열차위치 초기화의 실패
HAZ006	차상장치의 열차의 위치 계산 실패
HAZ007	이동거리의 정해진 상대오차 초과
HAZ010	열차 무결성 장치에 의해 열차 무결성 확인 실패
HAZ015	차상장치의 선로 위치별 최대 허용 속도 결정 실패
HAZ016	차상장치의 철도차량의 각 형식별 최대 허용 속도 결정 실패
HAZ017	차상장치의 ATS로부터 운행명령에 따라 임시속도제한구역 설정 실패
HAZ018	새로운 임시속도제한설정이 동일위치에서 시행 중인 기존 속도제한을 제거
HAZ019	수동으로 설정된 임시속도제한이 차상장치가 제공한 명령에 의해 설정 및 해제의 실패
HAZ021	차상장치의 노선상의 모든 위치별로 열차에 대해 허용된 속도를 계산 실패
HAZ022	차상장치의 경보곡선, 전상용 제동곡선 및 비상제동곡선의 계산 실패
HAZ038	차상장치는 인접 차상장치로부터 비상 제동 명령을 수신 시 차량 비상제동 투입 실패
HAZ042	ATO는 다음 정차 지점에서 정차할 수 있도록 하기 위한 운행속도 프로파일 결정 실패
HAZ043	ATO는 ATS의 명령에 의해 열차의 정차 및 통과가 가능하도록 운행 속도 프로파일 결정 실패
HAZ048	운행 속도 프로파일에 따라 철도차량에게 가속 또는 감속 신호 송출 실패
HAZ050	출입문이 자동열림인 경우 ATO의 열차 출입문 및 플랫폼 출입문 열림 신호 송출 실패
HAZ053	출입문이 자동닫힘이면서 ATS로부터 닫힘 명령을 받은 경우, ATO의 열차 출입문 및 플랫폼 출입문 닫힘 신호 송출 실패
HAZ056	특정 출입문 세트의 고장에 의해 수동으로 사용 중지시킨 경우, ATO는 전체 출입문 세트의 자동열림/닫힘 기능 작동 실패
#3 의도적 벗어남	
HAZ028	역에 정차한 열차가 없을 때 플랫폼 출입문이 개방되어 있다고 검지되면 차상장치의 방호영역 설정 실패
HAZ051	출입문이 수동열림인 경우 ATO의 열차 출입문 및 플랫폼 출입문 열림 신호 송출 실패
HAZ054	출입문이 수동닫힘인 경우 ATO의 열차 출입문 및 플랫폼 출입문 닫힘 신호 송출 실패

[표 71]에서 식별한 위험원을 토대로 위험원들이 운용 중 위험원 발생 시나리오 상에서 어떻게 발현되는지 정리한다.

(2) 운용 중 위험원 발생 시나리오 도출

○ HAZ002: 열차위치 초기화의 실패

열차는 열차위치 초기화를 통해 인접 차상장치에게 이를 전송하여 상호간의 위치 정보를 파악해야 한다. 운영 시나리오 상에서는 Driver가 인접 차상장치 ATP와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않을 시 열차위치 초기화의 실패로 이어진다.

○ HAZ005: 인접 차상장치에게 열차의 길이 전송 실패

인접 차상장치에게 열차의 길이 전송을 통해 상호간의 위치 정보 파악 및 방호구역 설정을 위한 정보를 제공한다. 운영 시나리오 상에서는 ATP가 인접 차상장치와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 방호구역 설정의 실패로 이어진다.

○ HAZ006: 차상장치의 열차의 위치 계산 실패

차상장치의 ATP가 열차의 위치 계산을 통해 인접 차상장치와 방호구역 간격을 유지하고 다음 역 정차시의 정보를 관제탑과 공유한다. 운영 시나리오 상에서는 ATP가 ATO 및 ATS와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차의 위치 계산 실패로 이어진다.

○ HAZ007: 이동거리의 정해진 상대오차 초과

차상장치의 미리 계산된 이동거리에 따라 운행 중에 앞선 차상장치 및 정차역의 간격 오차를 유지하는 것을 제공한다. 운영 시나리오 상에서는 ATP가 ATO 및 ATS와의 계산된 이동거리 및 열차의 길이, 위치를 종합하여 수행한다. 하지만 일부 계산의 오차로 인해 열차의 이동거리가 정해진 상대오차를 초과로 이어진다.

○ HAZ008: 차상장치가 열차위치를 ATS에 전송 실패

차상장치의 ATP가 열차위치를 ATS에 전송하여 그 다음 운행 프로파일 및 다음 역의 정보를 제공받을 수 있다. 운영 시나리오 상에서는 ATP와 ATS가 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차위치를 파악하지 못한다.

○ HAZ009: 운행 중인 분리된 열차 검지 실패

운행 중인 차상장치가 다른 분리된 열차 검지를 통해 운행 속도 및 거리를 조절할 수 있다. 운영 시나리오 상에서는 ATP가 다른 차상장치의 ATP와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차 검지 실패로 이어진다.

○ HAZ010: 열차 무결성 장치에 의해 열차 무결성 확인 실패

열차 무결성 장치에 의해 열차 무결성 확인을 통해 해당 차상장치의 열차 길이 및 위치를 제대로 파악할 수 있다. 운영 시나리오 상에서는 ATP가 열차 무결성 장치에 의해 차상장치를 인지하고 상태를 확인하는 과정을 거친다. 하지만 상태를 확인하는 과정이 제대로 수행되지 않으면 다른 인접 차상 장차의 열차 길이, 위치 등 계산과 방호구역 설정에 영향을 미친다.

○ HAZ011: 열차 MMI와의 인터페이스를 통해 열차 무결성 손실 정보 제공 실패

열차 무결성 장치에 의해 열차 무결성 확인 내용이 MMI와의 인터페이스를 통해 정보를 제공하여 운영자가 파악할 수 있다. 운영 시나리오 상에서는 ATP가 MMI와의 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차 무결성 손실 정보 제공 실패로 이어진다.

○ HAZ012: 열차 위치추적 실패를 검지했을 때, 차상장치의 비상제동명령의 실패

차상장치의 ATP가 다른 인접 차상장치 ATP와의 연결을 통해 위치추적을 수행한다. 하지만 이러한 위치추적의 실패를 검지하였을 때, ATP는 비상제동명령을 통해 열차를 제어할 수 있다. 운영 시나리오 상에서는 ATP는 ATS와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차 위치추적 실패로 인해 비상제동명령 시 실패로 이어진다.

○ HAZ013: 차상장치의 열차위치 실패에 대한 필요한 정보 제공 실패

차상장치의 열차위치 파악 및 계산 실패에 대한 정보를 ATO, ATS, Driver, 관제소에 정보를 제공하여 추가 발생되는 사고를 막을 수 있다. ATP는 이러한 열차위치 에러에 대한 정보를 제공하는 역할을 한다. 하지만 이러한 정보 제공에 대한 인터페이스가 수행되지 않으면 더 큰 위험으로 이어진다.

○ HAZ014: 열차위치 보고가 없는 경우 차상장치의 방호구역 결정 실패

열차위치에 대한 보고가 없는 경우 차상장치의 ATP는 방호구역 설정을 통해 인접 차상장치에 정보를 전달한다. 운용 중 위험원 발생 시나리오 상에서는 ATP가 인접 차상장치 ATP에 열차위치 정보 전달을 받지 못하는 경우 운행 속도 프로파일 조정을 통해 방호구역을 결정하고 수행한다. 하지만 이러한 정보 전달이 없는 경우 방호구역 결정 실패로 이어진다.

(3) 피해도 분석

피해도 분석은 운용 중 위험원 발생 시나리오 상에서 위험원이 발현되었을 때 얼마나 영향을 줄 것인가에 대하여 정리한다.

○ HAZ002의 감소

표 72 HAZ001의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
열차위치 초기화	진입하는 열차 위치 초기화 실패로 예측이 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	열차위치 초기화를 통해 인접 차상장치에게 이를 전송하여 상호간의 위치 정보를 파악해야 한다. 운영 시나리오 상에서는 Driver가 인접 차상장치 ATP와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않을 시 열차위치 초기화의 실패로 이어진다.

○ HAZ005의 감소

표 73 HAZ005의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
열차길이 결정	인접 차상장치에게 열차의 길이 전송 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	인접 차상장치에게 열차의 길이 전송을 통해 상호간의 위치 정보 파악 및 방호구역 설정을 위한 정보를 제공한다. 운영 시나리오 상에서는 ATP가 인접 차상장치와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 방호구역 설정의 실패로 이어진다.

○ HAZ006의 감소

표 74 HAZ006의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
열차위치 계산	열차 위치 계산 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	차상장치의 ATP가 열차의 위치 계산을 통해 인접 차상장치와 방호구역 간격을 유지하고 다음 역 정차시의 정보를 관제탑과 공유한다. 운영 시나리오 상에서는 ATP가 ATO 및 ATS와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차의 위치 계산 실패로 이어진다.

○ HAZ007의 감소

표 75 HAZ007의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
열차위치 계산	정해진 상대오차 초과로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	차상장치의 미리 계산된 이동거리에 따라 운행 중에 앞선 차상장치 및 정차역의 간격 오차를 유지하는 것을 제공한다. 운영 시나리오 상에서는 ATP가 ATO 및 ATS와의 계산된 이동거리 및 열차의 길이, 위치를 종합하여 수행한다. 하지만 일부 계산의 오차로 인해 열차의 이동거리가 정해진 상대오차를 초과로 이어진다.

○ HAZ008의 감소

표 76 HAZ008의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
열차위치 결정	열차위치 전송 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	차상장치의 ATP가 열차위치를 ATS에 전송하여 그 다음 운행 프로파일 및 다음 역의 정보를 제공받을 수 있다. 운영 시나리오 상에서는 ATP와 ATS가 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차위치를 파악하지 못한다.

○ HAZ009의 감소

표 77 HAZ009의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
분리된 열차 검지	분리된 열차 검지 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	운행 중인 차상장치가 다른 분리된 열차 검지를 통해 운행 속도 및 거리를 조절할 수 있다. 운영 시나리오 상에서는 ATP가 다른 차상장치의 ATP와 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차 검지 실패로 이어진다.

○ HAZ010의 감소

표 78 HAZ010의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
분리된 열차 검지	열차 무결성 확인 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	열차 무결성 장치에 의해 열차 무결성 확인을 통해 해당 차상장치의 열차 길이 및 위치를 제대로 파악할 수 있다. 운영 시나리오 상에서는 ATP가 열차 무결성 장치에 의해 차상장치를 인지하고 상태를 확인하는 과정을 거친다. 하지만 상태를 확인하는 과정이 제대로 수행되지 않으면 다른 인접 차상장차의 열차 길이, 위치 등 계산과 방호구역 설정에 영향을 미친다.

○ HAZ011의 감소

표 79 HAZ011의 저감대책

Event Tree Mitigation	Description	#1 운영시나리오
분리된 열차 검지	열차 무결성 손실 정보 제공 실패로 인한 예측 불가하다. 하지만 I/F의 결함허용 설계, S/W건전성확보 및 기능시험, 처리결과의 비교 또는 다수결 검증을 통해 위험감소가 가능하다.	열차 무결성 장치에 의해 열차 무결성 확인 내용이 MMI와의 인터페이스를 통해 정보를 제공하여 운영자가 파악할 수 있다. 운영 시나리오 상에서는 ATP가 MMI와의 인터페이스를 통해 수행한다. 하지만 인터페이스가 제대로 수행되지 않으면 열차 무결성 손실 정보 제공 실패로 이어진다.

2.4.3. FHA 결론

FHA 결과는 안전성 분석의 목적에 부합되도록 열차제어 시스템의 위험원을 중심으로 절차와 도출되는 데이터에 초점을 맞추어 가이드 된다. 열차제어 시스템의 개념 및 기본 설계에 대한 사항을 토대로 안전성 분석을 수행하였다. 얻어진 결과물들이 제대로 수행된 정보인지는 추가 분석이 필요하다. 맹목적으로 얻어진 결과물이 최적화된 것이라고 판단하여 사용한다면 문제가 발생할 수 있다. 또한 이러한 문제를 알고 난 후 변경하고자 한다면 추가적인 비용과 일정을 감수해야 할 것이다.

열차제어 시스템의 FHA을 통한 위험원을 분석하는 방법을 기술하였으며, 위험도를 허용수준으로 제어하고 열차충돌관련 기능에 대한 SIL4를 확보하는 것을 열차제어 시스템의 위험 감소 활동의 목표로 제시하였다. 따라서 시스템의 FHA 위험원을 고려하여 설계, 제작, 설치되어야 하며, FHA 위험원은 향후 프로젝트 수행과정에서 발생하는 변경 또는 갱신될 수 있다.

2.5. Fault Tree Analysis(FTA)

2.5.1. 위험원 기여 인자

FTA을 이용하여 열차제어 시스템 수준에서 발생 가능한 기능관점에서의 기여 인자를 확인한다. 주된 위험은 열차제어 시스템의 시스템 수준의 기능에서 보다 상세화 과정의 하위수준 사건으로 분류된다. 이런 주된 위험에 대한 사건 기여 인자들은 시스템 수준 위험으로 추가로 분류하여야 한다.

표 80 주요 위험인자

위험원 ID	설 명
Haz_01	선로전환기 오작동
Haz_02	건널목 차단기 오작동
Haz_03	진로 요청에 대해 승인 없이 열차가 이동함
Haz_04	열차가 의도와는 달리 연결되지 않음
Haz_05	의도하지 않은 열차 움직임
Haz_06	선로상의 미확인 열차
Haz_07	비상정지 장치 고장
Haz_08	열차 이동에 방해 받음
Haz_09	안전하지 않은 상태로의 모드 전환

개별 위험원을 기반으로 FTA 수행을 통해, 해당 위험원으로부터 발생 가능한 위험과 위험 시나리오가 식별되고 전개될 수 있다. 각 기여 인자에 대한 FTA의 수행은 아래와 같다. 위험 분석은 FTA가 사용된 경우 많은 열차제어 시스템 기능 고장에 대한 분석과 단일 시스템 기능 고장에 대한 인과분석인 두 가지 주요 방법으로 구성된다.

첫 번째 단계에서는 주된 위험인자가 더 하위 위험군로 분류되는 경우의 FTA 이다. 이러한 위험은 예상된 열차제어 시스템 ATP 안전필수 기능성 및 인간-기계 인터페이스(HMI) 그리고 차상 시스템-지상 시스템 인터페이스와 관련된 잠재적 고장 사건이다.

두 번째 단계는 연관된 위험원을 결정하기 위해 확인된 위험군을 평가하고 위험원을 허용 가능한 수준으로 감소시키기 위한 저감대책을 제안하며, 잔여 위험원을 결정하는 경우의 인과 분석이다. FT는 시스템과 시스템 간 인터페이스에서 주된 위험 기여 인자를 하위 위험으로 분해하기 위해 사용된다. FTA는 상위 사건으로부터 상위 사건이 발생할 수 있도록 하는 기본사건까지 하향식으로 진행하는 연역적 기법을 이용한 하향식(Top-down) 접근이다. 기본 사건은 열차제어 시스템 ATP 안전필수 기능 고장이다.

FT는 착안할 수 있는 위험에 대한 경로를 확인하며, 이는 단일 또는 다수의 기본 사건이 발생했기 때문에 발생할 수 있다. FT 나무(Tree)은 하위 위험과 상위 위험 사건과의 관계를 정의하기 위해 표준 “AND”, “OR” 논리게이트 기호를 사용한다.

FTA는 서브시스템 수준에서 ATP 기능이나 기능 간 상호작용에 잠재적 고장을 일으킬 수 있는 요인들로부터 위험원을 구분하는데 사용된다. 원인 분석은 ATP 기능의 단일 고장과 이들이 열차제어 시스템에 미치는 영향을 파악하고, 위험원 조건을 제거하거나 이를 허용 가능한 수준까지 줄이는데 필요한 제어 방법을 결정하는데 사용된다. 기본 사건은 서브시스템 기능의 고장(하자) 또는 외부 장비와의 서브시스템 인터페이스 장애이다. FT 나무에서는 단일 혹은 다수의 기본사건 발생으로 인해 발생될 수 있는 위험원으로 이어지는 경로들을 제시한다.

2.6. FMEA

2.6.1. 적용 범위 및 제약사항

아키텍처 산출물을 통해 FMEA을 수행하는데 필요한 구조적정보를 식별하고 시스템이 지니고 있는 물리적 컴포넌트를 정리한다.

표 81 식별 된 컴포넌트와 해당 기능 정보

컴포넌트	기능 ID	기능설명
ATP	4.3.5	열차위치 결정
	4.3.5.1	열차위치 초기화
	4.3.5.2	열차길이 결정
	4.3.5.3	열차위치 계산
	4.3.5.4	열차위치 결정
	4.3.5.5	분리된 열차 검지
	4.3.5.6	열차위치 고장의 대응
ATP	4.3.7	ATP 프로파일 결정
	4.3.7.1	최대 허용 속도 결정
	4.3.7.2	임시속도제한구역 설정
	4.3.7.3	열차 및 노선 속도에 관한 계산
	4.3.7.4	경보곡선, 전상용 제동곡선, 비상제동곡선 계산
ATP	4.3.11	출입문 제어
	4.3.11.1	올바른 측면에 있는 열차 출입문 개방 제어
	4.3.11.2	열차 출입문 닫힘 상태 및 PSD 닫힘 상태 제어
	4.3.11.3	역구내 출입문 제어
	4.3.11.4	역구내 정위치 정차
ATP	4.3.14	열차의 상태 관리
	4.3.14.1	열차의 주요장비 고장상태 확인
	4.3.14.2	열차의 화재 및 연기검지 상태 확인
	4.3.14.3	인접 열차의 비상 제동 명령 수신시 차량 비상제동
ATO	4.4.1	열차의 Sleep 상태 명령/해제
	4.4.1.1	열차의 Sleep 상태 해제
	4.4.1.2	열차의 Sleep 상태 명령
ATO	4.4.2	ATO 프로파일 결정
	4.4.2.1	ATO 프로파일 결정
ATO	4.4.3	열차 자동 제어
	4.4.3.1	ATO 프로파일에 따른 열차이동
	4.4.3.2	ATO 인칭 제어
ATO	4.4.4	ATO 프로파일에 따라 가속 또는 감속 신호 송출
	4.4.4.1	ATO 프로파일에 따라 가속 또는 감속 신호 송출
ATO	4.4.5	차상의 현시 화면에 데이터 전송
	4.4.5.1	차상의 현시 화면에 데이터 전송
ATO	4.4.6	출입문 제어
	4.4.6.1	출입문 열림 명령

컴포넌트	기능 ID	기능설명
	4.4.6.2	출입문 닫힘 명령
	4.4.6.3	특수한 상황에서도 출입문 열림/닫힘 기능 명령

FMEA 수행을 위해서는 순차적 프로세스에 따른 수행과 그것을 기록화 할 수 있는 FMEA 작성 양식 시트의 역할이 중요시 된다.

2.6.2. FMEA 수행 결과

[그림 81]과 같이, FMEA을 전문 지원하는 도구를 활용하여 VDA-FMEA(유럽표준) 접근 기반의 안전성 분석을 수행하였다. 뿐만 아니라, 도구기반의 수행이기 때문에 연동정보(Interface/Traceability)에 대해 명확한 연계성 확보가 가능하다.

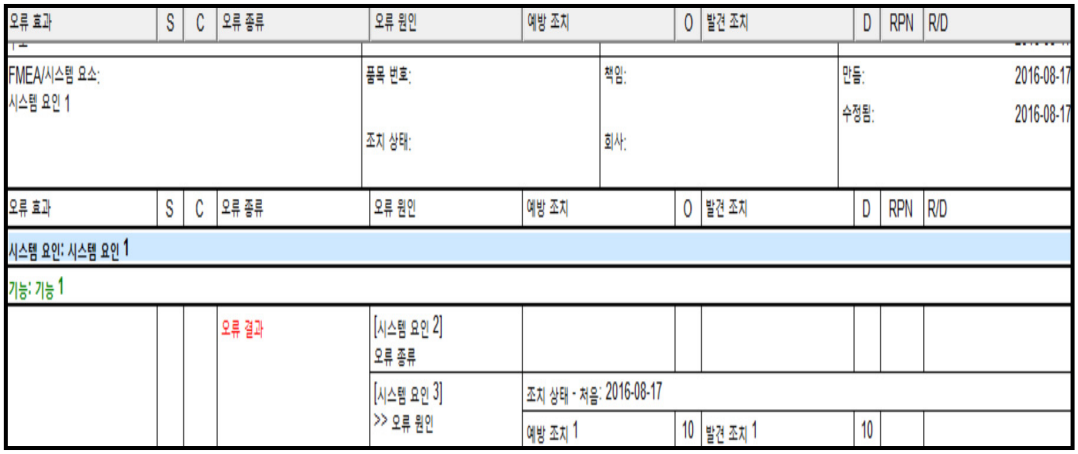


그림 81 FMEA 지원도구의 FMEA 양식 시트

FMEA 수행을 통한 FMEA 작성 시트에 작성되어야 하는 해당 정보로는 다음과 같은 사항들이 포함된다.

(1) 컴포넌트 or 기능

(2) 1번과 관련한 요구사항

(3) 2번의 잠재적 고장 모드

(4) 3번의 잠재적 고장영향

(5) 고장의 심각도(Severity)

(6) 고장 발생빈도(Occurrence)

(7) 고장 검출도(Detection)

(8) RPN 값(5,6,7번 값의 총체적 평가에 따른 우선위험도 분석을 바탕으로 위험원에 대한 저감 조치) 등이 반영되어야 한다.

이러한 활동을 위해서는 순차적 활동 및 수행을 위한 입력 산출물의 중요성을 인지하게 된다. 설계 산출물인 [그림 78]의 아키텍처 산출물을 바탕으로 열차제어 시스템의 구조 데이터를 기반으로 구조 네트워크(Structure Network)를 구축하였다. 구조 네트워크 확립을 통해 양방향 구조(Structure, Component)간의 연동 정보를 파악 할 수 있는 자료를 산출할 수 있다. 최상위 수준의 시스템을 열차제어 시스템이라고 두고, 계층적 관점에서 하부의 컴포넌트에 관해 구조적 연계성 확립을 수행할 수 있도록 주요 컴포넌트인 ATP, ATP, EI, 및 외부연동 장치의 관점에서 열차제어 시스템의 구성을 접근하여 수행되어야 한다.

구조분석을 기반으로 컴포넌트가 식별 되었다면, 식별된 컴포넌트가 지니고 있는 기능 요구사항을 기반으로 기능적 관점에서의 추적성을 확립되어야 한다. 이러한 기능들이 오류가 발생 시 발생될 수 있는 기능적 오류에 대해 연관관계를 확립한 산출물이 오류 네트워크이다. 오류 네트워크를 기반으로 시스템을 중심으로 상하 관계적 요소간의 기능적 오류 네트워크를 구축하게 된다. 이러한 개별 기능과 기능으로부터 발생 가능한 오류의 심각도(Severity), 발생도(Occurrence), 검출도(Detection)에 대한 평가가 수행되어야 한다.

IQ-FMEA 도구에서 제공하는 개별 항목에 대한 기준은 다음과 같다. 다음의 항목의 개별사항에 대해서 평가를 반영하여 FMEA 분석 시트 작성이 수행되어야 할 것이다.

○ 심각도(Severity: S)

심각도 평가 기준은 크게 10단계로 나누어진다.

표 82 심각도 평가 기준

Effect	Criteria: Severity of Effect on Product	Severity
안전과 법규 불만족	잠재적 고장형태가 사전 경고 없이 안전한 차량운행에 영향을 미치거나 정부법규에 대해 불일치 사항을 포함할 때	10
	잠재적 고장형태가 사전경고 후 차량운행에 영향을 미치거나 정부법규에 대해 불일치 사항을 포함할 때	9
주기능 상실 또는 열화	주요 기능 상실	8
	주요기능 성능 저하	7
보조기능 상실 또는 열화	보조기능 상실	6
	보조기능 성능하락	5
불쾌감	외관 또는 들리는 소음, 차량운행 기능, 부품이 불일치하거나 대부분의 고객이 인지(75%이상)	4
	외관 또는 들리는 소음, 차량운행기능, 부품이 불일치하거나 많은 고객이 인지(50% 이상)	3
	외관 또는 들리는 소음, 차량운행기능, 부품이 불일치하거나 일부 고객이 인지(25% 이하)	2
영향 없음	인지할 수 있는 영향 없음	1

○ 발생도(Occurrence: O)

표 83 발생도 평가기준

고장 가능성	Criteria: Occurrence of Cause	발생도
매우 높음	새로운 기술/전혀 없던 새로운 설계	10
높음	신규설계, 새로운 적용 또는 작동조건/부과사이클의 변화에 있어서 고장은 필연적이다.	9
	신규설계, 새로운 활용 또는 작동조건/부과사이클의 변화 있어서 고장 가능성이 있다.	8
	신규설계, 새로운 활용 또는 작동조건/부과사이클의 변화에 있어서 고장이 불확실하다.	7
중간	유사한 설계 또는 시뮬레이션/시험에서의 빈번한 고장	6
	유사한 설계 또는 시뮬레이션/시험에서의 가끔 발생하는 고장	5
	유사한 설계 또는 시뮬레이션/시험에서의 산발적인 고장	4
낮음	거의 동일한 설계 또는 시뮬레이션/시험과 관련된 단절된 고장	3
	거의 동일한 설계 또는 시뮬레이션/시험에서 관찰되지 않음	2
	예방관리에 의해 고장 제거됨	1

○ 검출도(Detection: D)

표 84 검출도 평가기준

검 출 도	Criteria: Likelihood of Detection by Design Control	발생도
검출기회 없음	새로운 기술/전혀 없던 새로운 설계	10
여타단계에서도 거의 검출불가	설계분석/검출관리가 약함. 시뮬레이션은 기대되는 실제작동조건과 부합되지 않음	9
런치 이전과 설계 확정 후	설계확정 후 제품검증/제품검증/논증 그리고 론치 이전단계에서의 합부시험 (조향, 승차 시스템시험과 출허평가 등).	8
	설계 확정 후 제품검증/논증 그리고 론치 이전단계에서의 고장시험(시스템 고장분석시험 등)	7
	설계 확정 후 제품검증/논증 그리고 론치 이전단계에서의 열화시험(내구후 기능평가 시스템 시험 등)	6
설계 확정 전	설계확정이전 단계에서 합부시험(성능, 기능)에 의한 제품논증	5
	설계확정이전 단계에서의 고장시험에 의한 제품 논증(신뢰성 시험, 개발 논증시험)	4
	설계확정 이전 단계에서 열화시험에 의한 제품논증	3
정합되는 시뮬레이션	시뮬레이션은 설계 확정 전에 실제 또는 기대되는 작동조건과 확실하게 정합함.	2
검출활용 없음; 고장예방	설계해법(검증된 설계표준, 상용재료, 또는 활용사례)을 통해 완벽히 예방되고 있음	1

앞서 활동된 FMEA 초기 과정을 거쳐 컴포넌트/기능/오류 네트워크를 구축할 수 있었다. 구축된 다양한 관점에서 발생된 산출물을 기반으로 위험원에 대한 중요도 평가를 수행해야 한다. 평가는 심각도/발생빈도/제어가능성(=검출도)을 기반으로 위험성을 평가하여 최종적으로 RPN 값으로 도출된다.

도출된 RPN이 높을수록 고 위험원이기 때문에 위험원 제거 또는 저감을 위한 방안이 설계에 반영되어야 한다. 이와 같이 FMEA 수행 후 도출 된 결과는 설계에 반영되어야 한다.

열차제어 시스템은 시스템을 구성하는 하부 장치에 ATP, ATO, EI 등 주요 하부 장치들이 존재한다. 특히, 이번 열차제어 시스템의 FMEA 수행에서는 열차제어 시스템의 주축 장비인 ATP에 상대적으로 많은 기능과 오류 네트워크에 대한 정보들이 존재한다는 것을 알 수 있다.

Item	오류 종류 (Failure Mode)	오류 효과 (Failure Effect)	오류 원인 (Failure Cause)	Risk Evaluation			예방 조치(Mitigation Measure)	Residual Risk Evaluation		
				F	S	R		F	S	R
시스템 요소: ATP(Automatic Train Protection)										
요소 기능										
시동 및 자가 테스트	열차 시동 시 자동으로 차상장치 가동 실패	열차 운행불능	1. 하드웨어 인터페이스 결함 2. 소프트웨어 오류 3. 연산부 결함	3	D	Tolerable	1. 인터페이스의 결함허용 설계 2. 소프트웨어 건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증	3	D	Tolerable
	열차 시동 시 차상장치 자가 테스트 수행 실패 또는 위험측 오류 수행	열차 운행불능, 열차 제동성능 및 무결성 확보 불가능 열차 추돌 및 충돌발생		3	A	InTolerable		5	A	Tolerable
	열차 시동 후 노선에 운행중인 다른 차상장치에 등록 수행 실패	등록 실패로 인한 정상운행 시작 불가		3	D	Tolerable		3	D	Tolerable
시스템 요소: EI										
요소 기능										
안전한 경로 설정의 제한	오브젝트 컨트롤러로부터의 선로전환기 상태 정보 송수신 오류	선로전환기 고장 및 전환중 상태를 정상으로 인식하거나 선로전환기 정상상태를 고장 및 전환 중으로 인식 등으로 인한 열차 탈선상황 발생	1. 소프트웨어 오류 2. 연산 오류 3. 하드웨어 결함	2	B	InTolerable	1. 소프트웨어 건전성확보 및 기능시험 2. 처리결과의 비교 3. 출력 하드웨어 결함허용 설계			Tolerable
	차상장치의 제동 거리 계산 오류 또는 기관사의 정보 수신 실패로 인한 해정 및 선로전환기 강제 전환	열차 주행중 선로전환기 강제 취급	1. 소프트웨어 오류 2. 연산 오류 3. 통신 오류	2	B	InTolerable	1. S/W 건전성확보 및 기능시험 2. 처리결과의 비교 3. 출력 H/W 결함허용 설계			Tolerable
시스템 요소: ATO										
요소 기능										
열차 제한 속도	제어된 추진 속도보다 차상장치의 속도가 더 높거나 낮은 값으로 출력되는 경우	과속 방지 기능이 방호 및 저속으로 인한 운행지연 발생	1. S/W 오류 2. H/W 고장 3. 연산부 결함	4	C	Tolerable	1. Overspeed protection 기능 필요	4	C	Tolerable
	제어된 제동 속도보다 차상장치의 속도가 더 높거나 낮은 값으로 출력되는 경우	과속 방지 기능이 방호 및 저속으로 인한 운행지연 발생		4	C	Tolerable	1. Overspeed protection 기능 필요 2. 정위치 정차를 위한 인칭제어 필요	4	C	Tolerable

그림 82 열차제어 시스템의 FMEA 수행 결과

2.7. 운용 및 지원상의 위험원 분석(O&SHA)

2.7.1. 적용 범위 및 제한사항

본 문서에서 정의하는 안전성 분석의 범위는 열차제어 시스템의 기능범위로 제한한다. 열차제어 시스템의 기능 및 수행범위는 앞에서 언급한 시스템 아키텍처를 참조한다. 추가로 다음 사항으로 인한 고장 및 위험원은 안전성 분석 범위에서 제외하도록 한다.

- 정의된 운영 조건 이외의 운영을 통해 발생 가능한 위험원
- 요구되는 사양을 초과하는 환경 조건
- 자연적인 대 재난(번개, 홍수, 지진 등)
- 테러 또는 반달리즘

2.7.2. 운용 및 지원상의 업무 분석

운용 및 유지보수 매뉴얼 분석을 통해 도출한 운용 및 유지보수 업무를 기반으로 시스템의 운용 및 유지보수 업무를 분석한다. 열차제어 시스템의 운용 및 유지보수 업무의 경우 열차, 차상장치, 지상장치 운용 및 유지보수 시 필요한 활동들이 토대로 분석을 수행한다. 열차제어 시스템의 운용 및 유지보수 업무 분석 결과는 다음 [표 85]와 같다.

표 85 열차제어 시스템의 운용 및 유지보수 업무 분석 결과

O&S ID	출처 내용	Task	Mitigation Measure
O&SHA_Task_001	ATS제어모드 또는 ATP제어모드일 때는 현장요원이 현장에서 설비를 제어할 수 있는 현장제어모드로 운영된다. 현장제어모드는 축소된 기능을 제공하고 현장요원이 수동으로 제어할 수 있다. 현장통제구역을 통과하는 열차주행모드는 수동모드로 운행된다.	현장제어 모드 운영	현장제어모드 운영 시 관제, 열차 운행요원 등에게 알림
O&SHA_Task_002	사전에 정의된 운행절차에 맞추어 선로상에 있는 열차간의 안전간격을 확보한다.	운행절차 확인	안전간격 미확보 시 보조방호시스템 가동
O&SHA_Task_003	관제실 운영요원은 열차의 상태, 위치 및 목적지 정보를 모니터링한다.	열차 정보 확인	열차의 상태, 위치 및 목적지 정보 통신 무결성 확보
O&SHA_Task_004	현장요원이 현장의 선로전환기를 조작하면 관제실 운영요원에게 선로전환기 레이아웃과 선로전환기 전환 제어가 표시된다.	선로전환기 조작	관제조작반에 선로전환기 조작 시 알림음 및 LED 표시
O&SHA_Task_005	열차제어 시스템 차상장치의 적절한 작동을 확인하기 위해, 점검은 열차제어 시스템제어영역으로 진입하기 전에 충분히 수행된다.	상용 운행 전 확인 및 점검	열차운행요원에게열차제어 시스템제어영역에서벗어나기전까 지시간및거리를시각적으로표시 확인점검미완료시고장표시제공

2.7.3. 운용 및 지원상의 위험원 분석

운용 및 지원상의 업무 분석 결과를 토대로 운용 및 지원상의 위험원을 식별한다. 운용 및 지원상의 위험원은 업무를 대상으로 HAZOP을 통해 식별되며 전문가와의 브레인스토밍을 통해 적절한 가이드워드를 선정하였다. 열차제어 시스템의 운용 및 지원상의 업무는 열차제어 시스템의 특성 상 열차를 제어하기 위한 열차와 지상장치, 차상장치와 지상장치간의 효율적인 운용 및 유지보수에 초점이 맞춰져 있다. 열차제어 시스템의 O&SHA 결과를 [표 86]에 기술하였다.

표 86 열차제어 시스템의 운용 및 지원상의 위험원 분석 결과

운용 및 유지보수 Task	가이드워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
현장제어 모드 운용	No	현장제어모드 운용 불가	현장제어모드 운용 불가로 인하여 관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										제어모드 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
	Wrong	잘못된 현장제어모드 운용	현장제어모드 운용 불가로 인하여 관리자/운영자 혼란 초래	2	2	허용가능	1	2	허용가능	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										제어모드 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
운행절차	No	운행절차 확인 불가	운행절차 확인	4	4	수	2	4	수	통신 불량	1.1 3GPP, TTA 통신 규격을

운용 및 유지보수 Task	가이드워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
확인			불가로 인하여 선로상에 있는 열차간 추돌 및 충돌로 인한 사상자 발생 가능			용불가			용		통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										관제조작반 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										차상장치 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										선/후행 ATP에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
열차 정보 확인	No	열차 정보 확인 불가	열차정보 확인 불가로 인하여 열차간 안전간격 유지를 위한 비상 제동 가능	3	4	수용(H)	2	4	수용	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										ATS 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										차상장치	1.1 인증된 부품사양서 기반

운용 및 유지보수 Task	가이드워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
										장애	신뢰성이 확보된 부품사용
										선/후행 ATP에 전송 불가	3GPP, TTA 표준 연동규격식별 및 사전 검증을 통한 무결성 확보
	Part of	열차 정보 일부의 확인	열차정보 확인 불가로 인하여 열차간 안전간격 유지를 위한 비상 제동 가능	3	4	수용(H)	2	4	수용	케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										ATS 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										차상장치 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
선로전환기 조작	No	선로전환기 조작 불가	선로전환기 조작 불가로 인하여 정해진 선로 이탈 방지를 위해 비상 제동 가능	2	3	수용	1	3	허용가능	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										선로전환기 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
	Wrong	잘못된 선로전환기 조작	선로전환기 조작 불가로 인하여 정해진 선로	2	3	수용	1	3	허용가	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및	1.1내구성이확보된인증된케이블

운용 및 유지보수 Task	가이드워드	위험원	영향	초기			잔여			원인	저감대책
				F	S	R	F	S	R		
			이탈 방지를 위해 비상 제동 가능						능	커넥터연결 오류	사용 1.2케이블에태그부착으로오결선 예방
										전파 간섭	1.1EMC규격에부합하도록설계 1.2EMC시험을통해부합성확인
										선로전환기 장애	1.1 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
상용 운행 전 확인 및 점검	No	상용 운행 확인 및 점검 불가	상용 운행 확인 및 점검 미 완료로 인한 열차 운행 정지로 인한 승객 대기	2	2	허용가능	1	2	허용가능	통신 불량	1.1 3GPP, TTA 통신 규격을 통한 통신 신뢰성 확보
										케이블 및 커넥터연결 오류	1.1내구성이확보된인증된케이블 사용 1.2케이블에태그부착으로오결선 예방
										소프트웨어 오류	1. 정적, 동적 분석을 통한 소프트웨어 품질 확보

2.7.4. O&SHA 결론

열차제어 시스템에 대한 운용 및 지원상의 위험분석 결과 총 28개의 위험원이 식별되었다. 운용 및 지원상의 대한 모든 위험원이 식별되었으며, 인터페이스와 관련된 원인도 식별하였다. 식별된 위험원에 대해 초기 위험도 평가를 통해 추가적인 위험도 저감 대책이 요구되는 위험원에 대해서는 해당 위험도를 최소 허용 가능한 수준으로 저감시키기 위한 안전 요구사항을 수립함으로써 최종적인 잔여 위험도 평가를 통해서는 해당 위험원이 허용 가능한 수준이하로 위험도가 저감되었음을 확인하였다.

제 5 절 시스템 안전성 분석과 소프트웨어 개발 연계활동

국제표준 IEC 62278을 바탕으로 철도 시스템의 안전성 분석을 통해, 상위수준에 대한 안전성 분석을 수행할 수 있도록 본 시스템 안전성 분석 가이드에서 정보를 제공하고 있다. 시스템 수준의 시스템 및 하드웨어에서의 안전성 분석은 해당 단계에서 마치는 것이 아니라, 이후 보다 서브시스템 수준으로 상세화 과정을 거쳐 소프트웨어 안전성 분석에 근간이 되는 정보를 제공한다.

따라서 시스템의 안전성 확보란 시스템, 하드웨어, 소프트웨어에 이르기까지 설계 정보를 바탕으로 연속적인 안전활동 결과를 발생 시키는데 목적이 있다. 본 시스템 안전성 분석 가이드에서는 시스템 수준의 안전성 분석 통해 식별된 위험원을 바탕으로 소프트웨어에서의 안전성 분석에 연계될 수 있는 접근 방안을 제시 하고자 한다. 특히, 본 안전 가이드의 5장. 가이드 적용 사례에서 시스템과 소프트웨어간의 연계 활동에 대해 사례를 통해 설명하고 있다.

제 4 장 철도 소프트웨어 개발 가이드

제 1 절 소프트웨어 요구사항

1. 개요

시스템 요구사항과 시스템 안전 요구사항을 준수하는 소프트웨어 요구사항을 기술하여 소프트웨어 아키텍처 및 설계 단계의 정보를 제공한다. 그리고 종합 소프트웨어 테스트 계획을 수립하고 소프트웨어 요구사항 검증을 수행한다.

1.1. 목표

- 소프트웨어 요구사항 명세서를 작성하여 시스템 요구사항과 시스템 안전성 요구사항에서 소프트웨어에 해당하는 항목을 식별하여 기술한다.
- 소프트웨어 안전 무결성 등급은 소프트웨어 개발 생명주기의 각 개발 단계별로 준수한다.
- 소프트웨어 개발 생명주기에 대해 단계별 추적성을 제공한다.
- 소프트웨어 요구사항은 “시스템 요구사항 명세서” 와 “시스템 설계 기술서” 를 준수하고 소프트웨어-하드웨어 인터페이스 명세와 일관되게 기술되었는지 검토한다.

1.2. 범위

소프트웨어 개발 생명주기에서 IEC 62279 7.2절에 해당하는 소프트웨어 요구사항 단계에 대해 설명한다.

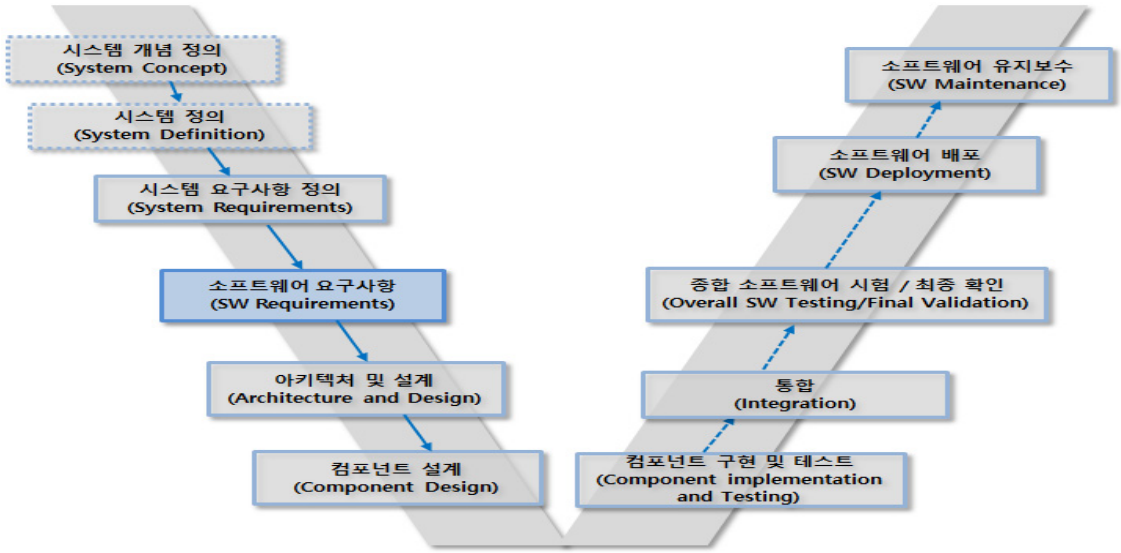


그림 83 소프트웨어 개발 생명주기 - 소프트웨어 요구사항 단계

1.3. 시작 기준

- 시스템 요구사항 확정
- 시스템 안전 요구사항 확정
- 시스템 요구사항, 시스템 안전 요구사항, 시스템 아키텍처를 통해 시스템 위험원 분석이 이루어지고 시스템 안전 무결성 등급 결정
- 소프트웨어가 탑재될 하드웨어 사양이 확정되어 소프트웨어-하드웨어 인터페이스 식별
- 소프트웨어 품질 보증 계획 수립
- 소프트웨어 확인 계획 수립

1.4. 완료 기준

- 시스템의 소프트웨어 구성요소에 할당되는 소프트웨어 요구사항 식별 및 도출 완료
- 소프트웨어 요구사항의 분석과 분류 완료
- 소프트웨어 안전 요구사항의 안전 무결성 등급 결정
- 소프트웨어 요구사항 구현에 필요한 우선순위 결정
- 소프트웨어 확인 계획을 준수하는 ‘종합 소프트웨어 테스트 명세서’ 작성 완료
- 소프트웨어 품질 보증 계획에 의한 ‘소프트웨어 요구사항 검증 보고서’ 작성 완료

1.5. 입력물

- 시스템 요구사항 명세서
- 시스템 안전 요구사항 명세서
- 시스템 아키텍처 기술서
- 외부 인터페이스 명세서 (소프트웨어/소프트웨어 인터페이스 명세서, 소프트웨어/하드웨어 인터페이스 명세서)
- 소프트웨어 품질 보증 계획서
- 소프트웨어 확인 계획서

1.6. 출력물

표 87 소프트웨어 요구사항 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 요구사항 명세서	HR	HR	HR	HR	HR
종합 소프트웨어 테스트 명세서	HR	HR	HR	HR	HR
소프트웨어 요구사항 검증 보고서	HR	HR	HR	HR	HR

1.7. 역할 및 책임

표 88 소프트웨어 요구사항 단계 역할 및 책임

단 계	문서	작성자	1차 검토	2차 검토
소프트웨어 요구사항	6. 소프트웨어 요구사항 명세서	RQM	VER	VAL
	7. 종합 소프트웨어 테스트 명세서	TST	VER	VAL
	8. 소프트웨어 요구사항 검증 보고서	VER		VAL
RQM (Requirement Manager) 요구사항 관리자 TST (Tester) 테스터 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 소프트웨어 요구사항 주요 활동

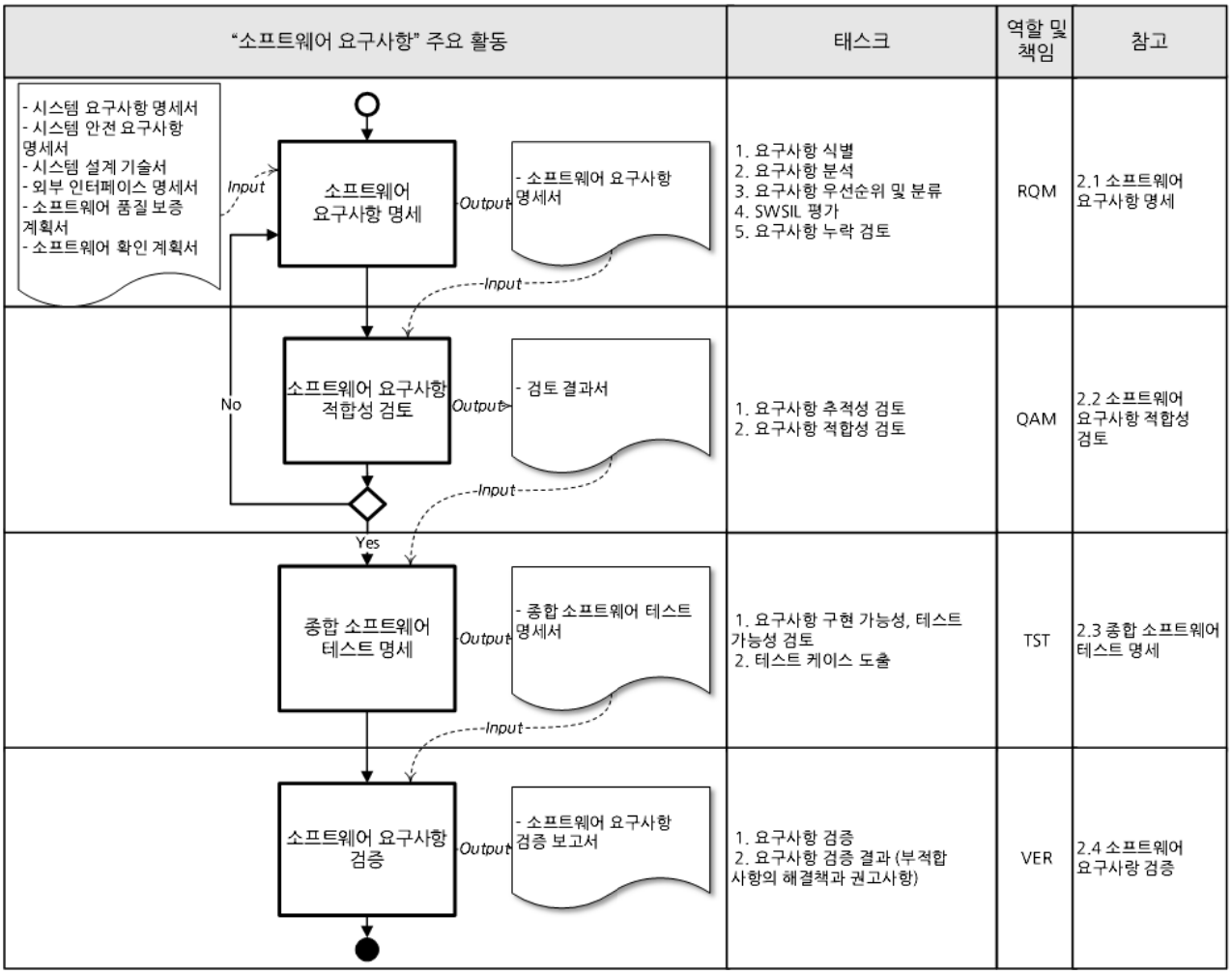


그림 84 소프트웨어 요구사항 주요 활동

표 89 소프트웨어 요구사항 단계

활 동 ID	활 동 명	설 명
REQ.01	소프트웨어 요구사항 명세	소프트웨어로 할당된 시스템 요구사항을 구현하기 위한 소프트웨어 요구사항을 시스템 설계 제약사항을 고려하여 기술한다. 소프트웨어 요구사항이 누락 없이 명세 되었는지 검토한다. 소프트웨어 안전 요구사항을 포함하여 소프트웨어 요구사항을 기술하고 소프트웨어 안전 무결성 등급을 평가한다.
REQ.02	소프트웨어 요구사항 적합성 검토	'소프트웨어 요구사항 추적표'를 작성하여 소프트웨어 요구사항이 시스템 요구사항에서 도출되었는지 추적성을 검토한다. 소프트웨어 안전 요구사항의 소프트웨어 안전 무결성 등급에 따라 선택된 기법 및 대책을 통해 적합하게 명세가 되었는지 검토한다. '소프트웨어 요구사항 체크리스트'를 통해 소프트웨어 요구사항의 명세 품질을 검토한다.

활 동 ID	활 동 명	설 명
REQ.03	종합 소프트웨어 테스트 명세	요구사항의 구현 가능성, 테스트 가능성을 검토한다. ‘소프트웨어 요구사항 명세서’를 기반으로 테스트 케이스를 도출하여 ‘종합 소프트웨어 테스트 명세서’를 기술한다. 안전관련 시스템인 경우, 테스터는 품질보증 관리자의 검토 결과 및 관련 산출물의 보완 결과를 받아 해당 산출물에 반영한다.
REQ.04	소프트웨어 요구사항 검증	‘소프트웨어 요구사항 명세서’와 ‘종합 소프트웨어 테스트 명세서’를 기반으로 소프트웨어 요구사항을 검증한다. 부적합 사유가 있으면 이에 대한 권고사항과 해결책을 기술한다.

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 소프트웨어 요구사항 명세

- ○ 시스템 요구사항과 시스템 안전 요구사항을 기반으로 요구사항의 식별, 분석, 우선순위 및 분류를 통해 ‘소프트웨어 요구사항 명세서’를 기술한다.
- ○ 소프트웨어 안전 요구사항에 대한 소프트웨어 안전 무결성 등급을 결정한다.

2.1.1. 소프트웨어 요구사항 명세 절차

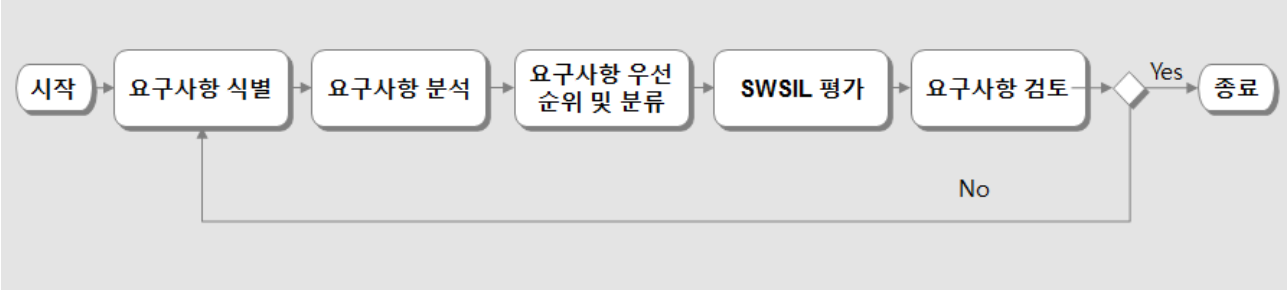


그림 85 소프트웨어 요구사항 명세 흐름도

- ○ 소프트웨어 요구사항 명세 절차

표 90 소프트웨어 요구사항 명세 절차 설명

항 목	설 명
요구사항 식별	시스템 요구사항과 시스템 아키텍처 설계를 준수하도록 소프트웨어의 기능, 비기능, 안전 요구사항을 식별하고 소프트웨어 요구사항을 기술한다. 소프트웨어만 개발하는 경우, 시스템 요구사항과 시스템 아키텍처 설계는 사용되는 운영 환경을 참조한다.
요구사항 분석	식별된 소프트웨어 요구사항을 구현가능성, 위험성, 테스트 용이성의 기술적인 관점에서 분석한다.
요구사항 우선순위 및 분류	식별되고 분석된 소프트웨어 요구사항에 대해 우선순위를 정하고 이를 구조적으로 분류한다. 시스템 위험원이나 소프트웨어 위험에 관련된 소프트웨어 일반 요구사항을 소프트웨어 안전 요구사항으로 분류한다.
소프트웨어SIL 평가	소프트웨어 안전 요구사항에 대해 소프트웨어 안전 무결성 등급(소프트웨어SIL)을 평가한다.

항 목	설 명
요구사항 검토	시스템 요구사항이 소프트웨어 요구사항으로 전환 시 누락되지 않도록 검토한다. 소프트웨어 요구사항을 비용, 일정, 기술적 영향 관점에서 검토하여 필요시 시스템 요구사항과 시스템 아키텍처 설계에 반영한다. 소프트웨어 요구사항을 승인하고, '소프트웨어 요구사항 명세서'를 갱신한다.

○ 소프트웨어 무결성은 안전 무결성을 위해 소프트웨어 SIL 0 (최저)에서 소프트웨어 SIL 1 ~ 4의 5 단계 중 하나로 결정되어야 한다.

○ 소프트웨어 안전 무결성 등급은 시스템 안전 무결성 등급과 시스템에서의 소프트웨어 관련된 위험 등급에 기반하여 시스템 수준에서 결정되고 평가되어야 한다.

○ 초기의 소프트웨어 안전 무결성 등급은 시스템 안전 무결성 등급 상속한다.

○ 시스템 안전 요구사항에서 도출된 소프트웨어 안전 요구사항의 소프트웨어 안전 무결성 등급은 시스템 안전 무결성 등급을 따른다.

○ 시스템 위험원 또는 소프트웨어 위험(소프트웨어 자체의 구조적인 사항)과 관련된 소프트웨어 일반 요구사항은 소프트웨어 안전 요구사항으로 식별하고, 소프트웨어 안전 무결성 등급은 소프트웨어 제어 유형과 시스템 안전 무결성 등급의 조합으로 결정한다.

○ 소프트웨어에서 시스템 위험원에 대해 발생 가능성을 완화하거나 방지를 하게 되면 소프트웨어 안전 무결성 등급을 낮출 수 있다.

○ 소프트웨어 안전 무결성 등급을 결정하기 위해서는 다음과 같은 조건이 만족되어야 한다.

- 시스템의 안전 무결성 등급이 결정되어 있어야 한다.
- 상세한 시스템 구조적 특징이 정의되어 있어야 한다.
- 시스템 안전성 분석을 통해 위험원 목록과 각 위험원에 대한 위험 인자 및 위험도가 결정되어 있어야 한다.

○ 시스템의 기능을 제어하는 소프트웨어 제어특성은 다음 사항을 고려한다.

- 시스템 기능을 제어하는가?
- 시스템 위험원을 발생시키는가?
- 소프트웨어 고장을 방지하는 다른 하위 시스템의 기능이 있는가?
- 시스템 위험원을 검지하고, 완화하기 위한 사용자의 조치를 요구하는가?

○ 소프트웨어 제어 유형과 특성에 따라 분류한다. (ISO 15026, MIL-STD-882(E), IEC 61508 참조)

표 91 소프트웨어 제어 유형 및 특성 (예시)

소프트웨어 제어 유형	소프트웨어 제어 특성				시스템 위험원 발생 가능성
	시스템 기능제어	시스템 위험도 수준	하위 시스템에서 고장 방지 및 완화	시스템 위험원 검지 및 완화 사용자 조치 요구	
Ⅰ	O	H	X	X	매우 높음
Ⅱ	O	H	X	O	높음
Ⅲ	O	L	O	X	중간
Ⅳ	X	L	X	△ (시스템 위험원 검지)	낮음
Ⅴ	X	X	X	X	매우 낮거나 없음

※범례: O (있음), X (없음), H(높음), L(낮음), △ (일부)

○ 소프트웨어 안전 무결성 등급 결정 매트릭스를 통해 소프트웨어 안전 무결성 등급을 평가한다.

표 92 소프트웨어 안전 무결성 등급 결정 매트릭스 (예시)

소프트웨어 제어 유형	시스템 안전 무결성 등급			
	4	3	2	1
Ⅰ	4	3	2	1
Ⅱ	3	2	1	0
Ⅲ	2	1	0	0
Ⅳ	1	0	0	0
Ⅴ	0	0	0	0

○ 소프트웨어 제어 특성과 설명을 고려하더라도 소프트웨어 제어 유형 할당과 소프트웨어 안전 무결성 등급의 평가는 쉽지 않다. 보통 이러한 경우 시스템 차원의 설계 세분화를 통해 소프트웨어에 요구하는 사항을 명확하게 함으로써 유형을 할당할 수 있도록 하거나 시스템 안전 무결성 등급에 준하여 소프트웨어 안전 무결성 등급으로 평가한다.

2.1.2. 소프트웨어 요구사항 명세 지침

○ 기능성, 강건성과 유지보수성, 안전성, 효율성, 사용편의성, 이식성 같은 품질 특성을 고려해야 한다. (7.2.4.2)

표 93 요구사항 단계에서 고려되는 소프트웨어 품질 특성 및 평가 내용 (예시)

품질 특성	설 명	부특성	설 명	평가 내용
기능성	요구하는 기능을 만족하는 능력	기능성숙도	명시된 요구사항 구현 정도	기능/성능 요건에 대한 준수 여부
		기능정확도	정의된 정밀도에 따라 정확하게 결과를 제공하는 정도	
		기능타당성	사용자의 목적 달성에 소프트웨어가 도움을 주는 정도	
강건성 및 유지보수성	소프트웨어 수정 및 변경의 용이성	모듈성	최소의 영향을 가진 컴포넌트로 구성된 정도	특정 조건에서 성능 수준의 유지 여부
		재사용성	자산이 하나 이상의 시스템에서 사용될 수 있고, 기타 자산을 구축할 수 있는 정도	
		분석성	시스템 변화에 대해 어떠한 영향을 받는지 평가할 수 있는 보고서를 제공하는 정도	
		수정가능성	제품 혹은 시스템이 장애 없이 효과적이고 효율적으로 수정될 수 있는 정도	
		시험가능성	제품 사용 전, 사용에 필요한 검증 기능 제공 여부	
안전성	시스템 위험원을 제거 또는 완화하기 위해 필요한 속성			안전 무결성 준수 여부
효율성	적절한 자원의 사용 및 적정한 반응시간 정도	시간반응성	기능 수행 시 응답, 처리 시간과 처리율이 요구사항을 충족시키는 정도	소프트웨어 운전 시 메모리, 저장 장치 등 자원을 적절하고 효율적으로 운전하는지 여부
		요소활용	기능 수행 시, 사용되는 자원의 유형 및 양이 요구사항을 만족시키는 정도	
		기억용량	제품 혹은 시스템 파라미터(최근 사용자 수, 통신 대역폭, 데이터베이스가 저장할 수 있는 데이터 양 등)의 최대 한계가 요구사항을 만족시키는 정도	

품질 특성	설 명	부특성	설 명	평가 내용
사용성	사용자가 이해하고 배우기 쉬운 정도	타당성 식별력	사용자의 요구에 적절한 기능인지 식별할 수 있는 정도	사용자가 쉽게 이해하고 학습할 수 있는지 여부
		학습성	사용자가 소프트웨어의 사용법을 배워 명시된 목적을 달성할 수 있는 정도	
		운용성	제품 혹은 시스템이 작동 및 제어를 쉽게 할 수 있는 정도	
		사용자 오류보호	소프트웨어가 발생한 오류로부터 사용자를 보호하는 정도	
		사용자 인터페이스	사용자 인터페이스가 사용자에게 만족하는 정도	
		접근성	연령과 장애에 관계없이 사용될 수 있는 정도	
이식성	지원하는 다양한 환경에서 운영될 수 있는 능력	적용성	제품 혹은 시스템이 다른 하드웨어, 소프트웨어 혹은 기타 사용 환경에 효과적이고 효율적으로 적용될 수 있는 정도	다양한 사용 환경(하드웨어, 운영체제 등)에서 사용 가능 여부
		설치성	제품 또는 시스템이 성공적으로 설치 및 제거 될 수 있는 정도	
		대치성	제품이 동일한 환경에서 동일한 목적을 위해 다른 지정 소프트웨어 제품으로 대체될 수 있는 정도	

- 소프트웨어 품질 특성을 고려하여 비-기능 요구사항을 기술한다. (5장 가이드 적용 사례 요구사항 명세, [표 239 MMI 소프트웨어 비-기능 요구사항 (예시)] 참조)
- 요구사항 명세 단계의 체크리스트 작성 시 소프트웨어 품질 속성을 고려하며, 품질 특성은 과제 특성에 따라 테일러링 될 수 있다.

○ 소프트웨어 아키텍처 및 설계 단계의 산출물과 추적성을 확보하고 ‘시스템 요구사항 명세서’, ‘시스템 안전 요구사항 명세서’, ‘시스템 아키텍처 기술서’를 준수하여 기술해야 한다. (7.2.4.3)

- 일관성을 유지해야만 추적이 가능하기 때문에 입력물에서 명시하는 사항을 반드시 준수해야 한다.

○ 필요한 기능으로 요구사항을 완전하게 기술해야 하고 소프트웨어 개발 생명주기 동안 소프트웨어 요구사항이 추적 가능하도록 기술해야 한다. (7.2.4.4)

- 추적성은 시스템 생명주기에서 가능하도록 하기 위해서는 모호성을 제거하고, 중복된 내용이 없어야 가능하다. 본 단계에서는 입력물과 출력물이 서로 추적이 가능하도록 양방향 추적성을 권장한다. (5장 가이드 적용 사례 요구사항 명세, [표 238 ~ 240] 참조)
- 소프트웨어 개발 생명주기에서 각 단계의 책임자에게 이해 가능하도록 상세하게 설명해야 한다. (7.2.4.5) (5장 가이드 적용 사례 요구사항 명세, [표 238 MMI 소프트웨어 기능 요구사항 (예시)] 참조)
- 소프트웨어 개발 생명주기에서 역할은 조직별로 구분된다. 한 조직에서 작성한 문서는 다른 조직에서 명확하게 파악하도록 객관화 시켜야 한다.
- 애매한 표현이나 서로 다른 의미로 해석할 수 있는 사항을 제거한다.

○ 타 시스템과의 인터페이스를 식별하고 기술해야 한다. (7.2.4.6)

- 타 시스템이란 개발범위가 아닌 다른 시스템이나 연동 대상 시스템을 의미한다. 일례로 차상 MMI 시스템이 개발 대상이면 차상 ATP 시스템과의 인터페이스를 의미한다. (5장 가이드 적용 사례 요구사항 명세, [그림 178 차상 MMI 시스템 개념도 (예시)] 참조)

○ 관련된 운영 모드가 기술되어야 한다. (7.2.4.7)

- 5장 가이드 적용 사례 요구사항 명세, [표 234 시스템 요구사항 (예시)] SRS_MMI_REQ1.1 의 “운영 모드” 참조

○ 프로그램 가능한 전자 장치의 모든 행위에 관련된 모드(특히 고장 행위)를 기술되어야 한다. (7.2.4.8)

- 다른 모드와 구별되는 요구사항과 모드별 운영이 필요하면, 각 모드를 식별하고 정의해야 한다. 모드의 예로써는 유휴(idle), 준비(ready), 활동(active), 사용 후 분석, 기능 저하(degrade), 비상사태(emergency), 백업(backup), 자동(auto), 수동(manual), 전시, 평상 시 등이 있다.

○ 하드웨어와 소프트웨어 간의 제약사항을 기술해야 한다. (7.2.4.9)

- 소프트웨어에서 하드웨어를 제어 또는 접근하기 위해 유의해야할 사항을 기술한다.
- 예시: 펌웨어의 저장 공간은 8M이고 업그레이드의 안정을 위해 듀얼로 구성한다.

- 소프트웨어 자체적인 고장과 오류에 대해 검지하고 보고하는 기능과 하드웨어 상태 확인 기능을 기술해야 한다. (7.2.4.10)
  - 필요 시 데이터보호 혹은 시스템 무결성에 대한 감시(Watchdog) 기능을 기술해야 한다.
  - 소프트웨어 자체적으로 발생하는 고장과 오류를 검지하고 이에 대응하며, 하드웨어 제어나 접근 시 반드시 하드웨어 정상 동작 유무를 확인해야 한다.
- 시스템 안전 요구사항과 관련된 사항을 주기적으로 테스트하는 기능을 기술해야 한다. (7.2.4.11)
- 시스템 안전 요구사항과 관련해서 시스템 운영 중에 안전 기능을 테스트할 수 있도록 기술해야 한다. (7.2.4.12)
- 시스템 운영 중 시스템 안전 요구사항에 관련된 기능이 정확하게 동작하는지 확인하기 위해 주기적으로 테스트해야 한다.
- 시스템 안전 무결성 등급을 준수하기 위한 기능은 명확하게 식별해야 한다. (7.2.4.13)
- 비안전 기능(일반 기능)은 명확하게 식별해야 한다. (7.2.4.14)
- 소프트웨어 안전 요구사항은 시스템의 기능저하 모드 혹은 안전 상태로 진입 및 유지 상태에 대한 방안을 기술해야 한다.
- 소프트웨어 요구사항 명세에서 가장 중요한 사항은 “무엇(what)”을 할 것인가를 정의하는 것이다. 따라서 “어떻게(how to)”에 해당 하는 부분에 대해서는 요구사항 단계 이후에서 기술하게 된다.
- 본 단계에서는 “정형 기법”을 사용하여 기술하는 것이 중요하다. 하지만 모든 것을 “정형 기법”으로 기술하는 것은 쉽지 않기 때문에 “준정형(semi-formal) 기법”을 사용한다. 무엇에 해당하는 부분에 대해 논리적인 “모델링”을 통해 의미를 제시하고 부연 설명을 통해 의미를 명확하게 표현한다.
  - IEC 62279 표준의 “Table A.17 Modeling” (4장 9.7의 [표 220] 참조) 에 기술된 11가지 기법 및 대책에 대해 소프트웨어 안전 무결성 등급에 따라 적어도 하나 이상의 HR로 표기된 기법을 사용하여 요구사항을 명세하도록 권장한다.
  - 시범적용 예시에서는 처리 절차의 흐름을 나타내는 순차 다이어그램(Sequence Diagram)을 사용하였다. (5장 가이드 적용 사례 요구사항 명세 [그림 177 차상 MMI 시스템의 시퀀스 다이어그램 (예시)] 참조)
  - 순차 다이어그램의 경우 처리 순서를 도식화해서 이해하기 쉽기 때문에 이후 “아키텍처 및 설계” 단계에서 명세할 때 보다 편리해 지는 장점이 있다.

○ 철도 시스템의 소프트웨어 요구사항 단계에서 소프트웨어 안전 무결성 등급에 따라 선택되는 적용 기법 및 대책은 다음과 같다. (7.2.4.15)

표 94 소프트웨어 요구사항 단계 적용 기법 및 대책 설명 (예시)

기법 및 대책	설 명
정형 기법 (Formal Method)	- 수리, 논리에 기반하여 하드웨어 시스템 또는 소프트웨어 시스템을 명세, 개발, 검증하는 방법이다. - 안전, 임무, 보안 필수 시스템에 적용 가능하다. - 기능적 요구사항 이외에도 비기능적 요구사항 적용 가능하다. - 철도 시스템 같은 종료되는 시점이 없이 무한히 동작하는 환경에서 사용된다. - 시스템 전체뿐만 아니라 일부 중요한 모듈에 대해서도 적용할 수 있다. - 정형 명세와 정형 검증의 두 가지 측면으로 설명된다. - 부록 B-28 참조
정형 명세 (Formal Specification)	- 수리, 논리를 이용하여 시스템의 기능, 행위, 동작 환경을 기술하는 것이다. - 사용 목적은 모호함을 없애기 위함이다. - 무엇(what)을 명세 하는지에 해당한다. 이를 통해 수많은 요구사항 간 불일치를 없앨 수 있다. - 이를 통해 요구사항을 명세하면 수학적으로 증명할 수 있다. 즉, 설계가 요구사항을 만족시키는지 증명할 수 있다. (1) 수리, 논리에 기반하여 시스템을 명세 - 시스템의 상태를 나타내는 변수의 타입을 명세하기 용이하다. - 기능(function)의 전후 상태를 명세한다. - Z notation, B method (TGV 철도 신호 시스템에서 사용), 이벤트 기반의 Event-B, Galina 등 (2) 대수(algebra)에 기반하여 시스템을 명세 - 병렬성(parallelism)과 동시성(concurrency(동시성)을 설명하기 용이하다. - CCS (Pi-calculus, 모바일), CSP (보안), ACSR (펜실베니아 대학교, 선형 논리의 리소스를 표현) (3) 시각적 명세언어에 기반하여 시스템을 명세하는 오토마타 기반 - 상태(state)와 천이(transition)으로 시스템의 행위를 기술한다. - 대상 시스템의 행위 표현이 용이하다. - 의미(semantics)가 명확하다. - Finite State Machine, State chart (FSM + 계층), UPPAAL(FSM + 시간) - 부록 B-28 참조
정형 검증 (Formal Verification)	- 수리, 논리를 이용하여 시스템의 명세, 설계, 요구사항을 검증하는 것 (1) 의미론적(semantics) 증명 - 시스템 모델이 특성 속성을 만족하는지 상태를 추적하여 확인한다. - 오토마타 기반의 명세를 대상으로 한다. - 자동화가 가능하다. - 모델 체킹 (model checking) (2) 구문적(syntactic) 증명 - 시스템 모델이 특정 속성을 만족하는지 구문적 규칙을 적용하여 중명한다. - 논리 기반의 명세를 대상으로 한다. - 사람이 직접 단계별로 증명해야 하며, 이 과정에서 도구의 도움을 받을 수 있다. - 추론 시스템 (deductive system)을 이용하여 증명한다. - 정리 증명 (theorem proving)

기법 및 대책	설 명
	(3) SAT, SMT (SAT + Modeling Theories) - 부록 B-28 참조
모델링 (Modeling)	- 개발할 소프트웨어를 가시적으로 분석이 가능하도록 하는 기법 - 소프트웨어에서 UML 다이어그램, Z 언어를 이용하여 요구사항을 정형적으로 표기하는 기법 (예시) - [표 220 모델링 참조
데이터 모델링	- 데이터 모델은 데이터, 데이터 관계, 데이터 의미 및 데이터 제약 조건을 기술하기 위한 개념적 도구의 집단이다. 데이터 관계에 따른 데이터가 논리적으로 조직될 수 있는 양식이며, 데이터 양식에는 엔티티형과 엔티티 관계에 의한 추상적 개념의 데이터 조직 규칙, 연산의 정의, 여러 제약 조건이 포함된다. - 사용자의 요구사항으로부터 데이터 엔티티(구성요소)를 정의 - 부록 B-65 참조
구조적 방법론 (Structured Methodology)	- 정형화된 분석 절차 및 도형 중심의 도구를 이용하여 사용자 요구사항 파악 및 문서화하는 기법 - 구조적 프로그래밍에서 출발하여 설계의 원칙을 정리한 구조적 설계, 시스템 복잡성 해결을 위한 구조적 분석으로 발전 - 정보와 정보의 구조를 중심으로 분석/설계 및 구현을 실행 - 정형화된 분석 절차에 따라 사용자 요구사항을 파악하고 도형 중심의 다이어그램을 이용하여 문서화 작업 진행 - 부록 B-52 참조
결정 테이블 (Decision Tables)	- 입력 조건의 모든 조합에 대한 시스템의 행동을 고려하여 테스트 케이스를 도출하는 방법 - 시스템 행동에 영향을 주는 조건 분석 - 조건을 먼저, 행동을 후에 위치하는 테이블 작성 - 모든 조건의 조합을 나열하고 그에 해당하는 행동을 기입 - 일어날 수 없는 조건의 조합을 테이블에서 삭제 - 동일한 행동을 유발하는 조건의 조합을 찾아 테스트 케이스를 줄임 - 결정 테이블 테스트 방식은 복잡한 논리적 관계를 표현하기가 좋고, 누락된 요구사항 검사 시 용이함. - 부록 B-13 참조

2.1.3. 소프트웨어 요구사항 명세서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 요구사항 및 소프트웨어 안전 요구사항을 정의한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문서를 나열한다.
- 예시) '시스템 요구사항 명세서', '시스템 안전 요구사항 명세서', '시스템 아키텍처 기술서', '소프트웨어/하드웨어 인터페이스 명세서'

1.4 약어표

- 약어 설명을 한다.
- 예시 1) ATP - Automatic Train Protection
- 예시 2) MMI - Man Machine Interface

2. 소프트웨어 요구사항

2.1 기능 요구사항

- '시스템 요구사항 명세서'와 '시스템 설계 명세서', '소프트웨어-하드웨어 인터페이스 명세서'에서 식별되어 소프트웨어 요구사항으로 할당된 것을 기술한다.
- 시스템이 사용자에게 제공해야 하는 기능을 기술한다.
- 예시) MMI 소프트웨어는 차상 ATP/ATO 에서 수신된 운행 및 상태 정보를 저장해야 한다.

2.2 비기능 요구사항

- 성능 요구사항을 기술한다. (특정 조건에서 기능을 수행할 때 필요한 시간이나 처리량 또는 자원의 최대 사용치를 기술한다.)
- 기능 테스트가 되지 않는 소프트웨어 품질 속성 (강건성과 유지보수성, 효율성, 사용편의성, 이식성)을 기술한다.
- 제약 조건을 기술한다.
- 안전 무결성 등급에 따라 복잡도를 제시한다.
- 예시) MMI 소프트웨어는 차량 운행 정보의 속도 표시는 0~140km/h까지 표시해야 한다.
  근거 - ATP/ATO에서 수신된 차량 운행 정보의 속도는 140Km/h를 넘지 않는다.

2.3 안전 요구사항

- 시스템 위험원과 관련이 있거나 안전한 상태를 유지하기 위해 필요한 사항을 기술한다.
- 소프트웨어 안전 무결성 등급을 표시한다.

- 예시 1) 운영모드가 Active 이고 차량이 주행 중 일 때는 설정모드로 전환되어서는 안 된다.
- 예시 2) 차량 전원 차단이 검지된 경우 로그를 기록해서는 안 된다.

A. 부록

- 추가적인 사항을 기술한다.
- 예시) 승인 내역

그림 86 소프트웨어 요구사항 명세서 템플릿 (예시)

2.1.4. 소프트웨어 요구사항 명세서 체크리스트

체크리스트로 ‘소프트웨어 요구사항 명세서’의 품질을 검토한다.

표 95 소프트웨어 요구사항 명세서 체크리스트 (예시)

구 분	점검사항
정확성/완전성	모든 기능이 기술되었는가?
	적용 가능한 요구사항이 모두 식별되었는가?
	적용 가능한 시간자원 사용과 관련된 시스템 용량 요구사항이 모두 식별되었는가?
	성능요구사항과 자원예산 안에 운영시스템 기존 소프트웨어의 효과가 분석되어 포함되어 있는가?
	적용 가능한 안전 요구사항이 식별되었는가?
	적용 가능한 보안 요구사항이 식별되었는가?
	적용 가능한 설계 제약사항 요구사항이 식별되었는가?
	적용 가능한 소프트웨어 품질 특성(기능성, 강건성과 유지보수성, 안전성, 효율성, 사용편의성, 이식성)은 식별되었는가?
	각 할당된 요구사항은 추적 가능한가?
	프로세스를 처리하는데 요구되는 수학적 방정식 등이 기술되어 있거나 참조되어 있는가? (정형 기법 사용 시)
	요구사항이 애매모호하지 않고 정확하게 정의되어 있는가?
	모든 소프트웨어 기능이 고려되었는가?
	모든 테스트 요구사항이 정의 되었는가? (요구사항 테스트 명세서 검토 사항)
일관성	소프트웨어 요구사항은 시스템 요구사항으로부터 할당된 것인가?
	요구사항은 논리적 상충관계가 없는가?
	요구사항은 시간적 상충관계가 없는가?
	요구사항은 한번만 명세 되었는가?
	요구사항은 다른 요구사항과 상충관계가 없는가?
	약어와 용어가 일관성 있게 사용되고 있는가?

구 분	점검사항
	수학적 방정식이 일관되게 정의되고 있는가? (정형 기법 사용 시)
	다이어그램에 사용된 기능명이 요구사항 문장과 일관성이 있는가?
구현 가능성	요구사항은 가용한 기술로 달성 가능한가?
	요구되는 도구가 가용한가?
	요구사항의 범위가 현실적인가? (소프트웨어 예상/예측, 일정과 지원 설비 계획을 고려했는가?)
	가용한 사실 또는 모델링 정보에 기초하여 성능 요구사항이 현실적인가?
	자원 예산이 현실적인가?
	중요 소프트웨어 기능이 시스템 운영과 연결되어 기술되어 있는가?
표준 준수성	요구사항이 명확하고 애매함 없이 기술되어 있는가?
	사용 용어가 이해 가능하고 일관적인가?
	모든 표기와 명명규칙이 정의 되었는가?
	용어사전이 적절한가?
	요구사항 명세서가 요구되는 양식에 의해 기술되었는가?
	요구사항이 명확하게 일련번호화 되거나 표시되는가?
	사용 용어가 이해 가능하고 일관적인가?
	철자나 문법 오류에 대해 수정/편집 되었는가?
	요구사항 표현 용어가 정확하게 사용되었는가?
테스트 용이성	소프트웨어에 대한 모든 요구사항이 명세 되었는가?
	요구사항이 여러 수단으로 검증 가능한가?
	현재 또는 계획된 자원을 이용하여 모든 요구사항들에 대한 테스트 절차가 기술될 수 있는가?
	사전에 정의된 완료 기준에 따라 테스트 결과가 평가될 수 있는가?

2.2. 소프트웨어 요구사항 적합성 검토

- 소프트웨어 요구사항 적합성은 소프트웨어 품질 보증 계획에 따라 작성자와 별도의 조직에서 검토해야 한다.
- 요구사항 추적표를 작성하여 소프트웨어 요구사항이 시스템 요구사항으로부터 도출되었는지 추적성을 검토한다. (별도로 요구사항 추적표를 작성하지 않고 ‘소프트웨어 요구사항 명세서’ 내에서 시스템 요구사항과의 연관성을 표시하기도 한다. (4장 가이드 적용 사례 요구사항 명세 참조)
- ‘소프트웨어 요구사항 명세서 체크리스트’를 통해 정확성/완전성, 일관성, 구현 가능성, 표준 준수성, 테스트 용이성을 검토한다.
- 소프트웨어 개발 생명주기에서 요구사항 단계와 설계 단계는 단계별 적합성을 반드시 검토해야 한다. 그 이유는 안전관련 시스템 구축 시 소프트웨어 개발 생명주기에서 요구사항 단계와 설계 단계가 시간적 요소가 많이 할애되기 때문이다.

2.3. 종합 소프트웨어 테스트 명세

소프트웨어 요구사항의 테스트하기 위해 요구사항을 검토하고 테스트 케이스를 기술한다.

2.3.1. 종합 소프트웨어 테스트 명세 절차

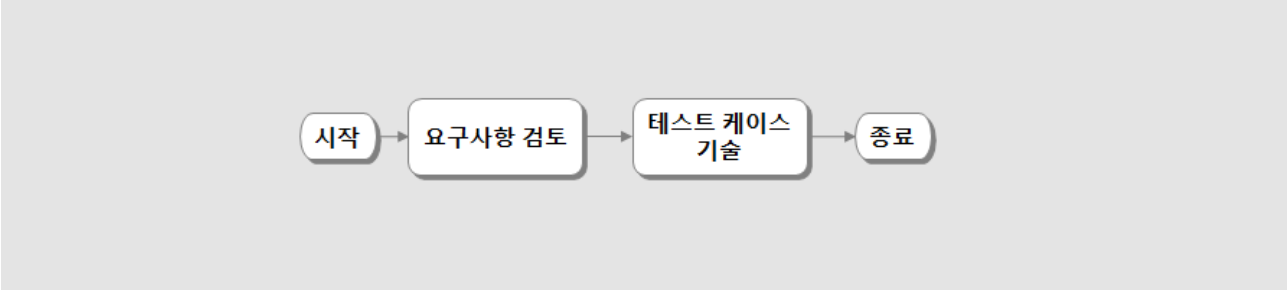


그림 87 종합 소프트웨어 테스트 명세 흐름도

○ 종합 소프트웨어 테스트 명세 절차

표 96 종합 소프트웨어 테스트 명세 절차 설명

항 목	설 명
요구사항 검토	- 구현 가능성 및 테스트 가능성을 검토한다. - 안전 무결성 등급을 준수하도록 기법 및 대책을 선택한다.
테스트 케이스 기술	- 요구사항을 테스트 할 수 있는 테스트 케이스를 도출한다. - 입력값, 예상 출력값, 성공 기준을 기술한다.

2.3.2. 종합 소프트웨어 테스트 명세 지침

○ 개발 완료된 소프트웨어에서 수행할 테스트에 대한 설명을 다음과 같은 사항을 통해 기술해야 한다. (7.2.4.17)

- 테스트 목적
- 테스트 케이스, 테스트 데이터, 예상 결과값
- 테스트 환경, 도구, 형상 및 프로그램
- 테스트 완료 기준
- 테스트 절차에 소속되어 있는 구성원의 역할과 책임
- 테스트 케이스가 어떤 요구사항을 테스트 하는지 명시
- 테스트 장비의 선택 기준과 활용 방안

○ 소프트웨어 안전 무결성 등급에 따라 대책 및 기법을 선택한다. (7.2.4.18)

표 97 소프트웨어 안전 무결성 등급에 따른 기법 선택 (예시)

항 목	설 명
소프트웨어SIL 3, 4	- 성능 테스트 (Avalanche/스트레스 테스트, 반응 시간 및 메모리 제약사항, 성능 요구사항) 필수 - 기능 테스트 (경계값 분석, 동치 클래스 및 입력 파티션 테스트) 필수
소프트웨어SIL 1, 2	- 성능 테스트 (Avalanche/스트레스 테스트, 반응 시간 및 메모리 제약사항, 성능 요구사항) 와 기능 테스트 (경계값 분석, 동치 클래스 및 입력 파티션 테스트) 의 조합

○ 테스트 케이스에 대한 요구사항 기술 시 다음과 같은 사항을 포함하여 기술한다. (7.2.4.19)

- 실행 순서와 입력값
- 실행 순서와 출력값
- 성능과 품질을 포함하는 테스트 완료 기준

2.3.3. 종합 소프트웨어 테스트 명세서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 요구사항을 정의한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘시스템 요구사항 명세서’, ‘시스템 안전 요구사항 명세서’, ‘시스템 아키텍처 기술서’, ‘소프트웨어/하드웨어 인터페이스 명세서’, ‘소프트웨어 요구사항 명세서’

1.4 약어표

- 약어에 대한 설명을 한다.
- 예시 1) ATP – Automatic Train Protection
- 예시 2) MMI – Man Machine Interface

2. 종합 요구사항 테스트 개요

- IEC 62279에서 요구하는 안전 수준에 따른 테스트 기법을 선택하여 기술한다.
- 제약사항을 기술한다.

3. 테스트 케이스

- 테스트 케이스 요구사항을 기술한다.
- 예시)

테스트 ID (Test ID)	목적 (Objective)	관련 요구사항 ID (Requirement ID)	입력값 (Input)	예상 출력값 (Expected Output)	완료 기준
TCID-01	차량 전원 공급 상태를 확인한다.	소프트웨어 SRS_MMI_REQ1.2 MMI 시스템은 차량 전원 차단을 검지해야 한다.	CPU 레지스터 (0x80)	(POWER_ON) (POWER_OFF)	예상 출력값 확인

A. 부록

그림 88 종합 소프트웨어 테스트 명세서 템플릿 (예시)

2.3.4. 종합 소프트웨어 테스트 명세서 체크리스트

표 98 종합 소프트웨어 테스트 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
품질 요구사항	테스트 목적, 테스트 케이스, 테스트 데이터, 예상 결과값, 테스트 환경, 도구, 형상 및 프로그램, 테스트 완료 기준을 기술하였는가?
	테스트 절차에 소속되어 있는 구성원의 역할과 책임이 명시되어 있는가?
	테스트 케이스가 어떤 요구사항을 테스트 하는지 명시하는가?
	테스트 장비의 선택 기준과 활용 방안이 기술되어 있는가?
(기타)	기타항목

2.4. 소프트웨어 요구사항 검증

소프트웨어 요구사항 검증 보고서는 ‘소프트웨어 요구사항 명세서’, ‘종합 소프트웨어 테스트 명세서’가 시스템 요구사항, 시스템 안전 요구사항 및 ‘시스템 아키텍처 기술서’를 준수하는지 검증한다.

2.4.1. 소프트웨어 요구사항 검증 보고 절차

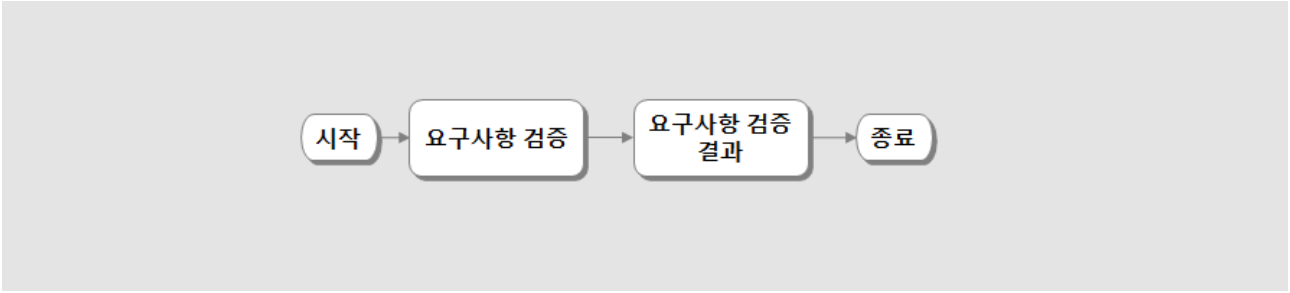


그림 89 소프트웨어 요구사항 검증 보고 흐름도

표 99 소프트웨어 요구사항 검증 보고 절차 설명

항 목	설 명
요구사항 검증	- 시스템 요구사항에 소프트웨어 요구사항의 전환이 적절하게 되었는지 검증한다. - 소프트웨어 요구사항의 타당성, 일관성, 완전성, 현실성을 검증한다.
요구사항 검증 결과	- 부적합사항의 해결책과 권고사항 - 요구사항 검증 결과 요약

2.4.2. 소프트웨어 요구사항 검증 지침

○ 소프트웨어 요구사항 검증 보고서는 다음과 같은 사항을 기술해야 한다. (7.2.4.21)

- 검증자 이름, 검증 항목의 식별과 형상
- ‘소프트웨어 요구사항 명세서’를 준수하지 않는 항목
- 컴포넌트, 데이터, 자료구조, 알고리즘이 적당하지 않아 문제를 발생시키는 것
- 오류 또는 불완전한 사항의 검지 여부
- 소프트웨어 검증 계획의 준수 여부 (준수하지 못한 경우의 중요도 기술)
- 검증 결과 요약

○ 하드웨어와 소프트웨어 간의 제약사항을 반드시 고려하고, ‘소프트웨어 요구사항 명세서’와 ‘시스템 요구사항 명세서’, ‘시스템 안전 요구사항 명세서’, ‘소프트웨어 품질 보증 계획서’의 적합성을 검증해야 한다. (7.2.4.22)

- ‘소프트웨어 요구사항 명세서’의 요구사항이 가독성과 추적성을 고려하여 기술했는지 검증해야 한다. (7.2.4.22)
- 요구사항 검증은 다음 사항을 고려해야 한다. (7.2.4.22)
  - 검증 방법은 리뷰, 워크스루를 사용한다.
  - 필요한 사항을 명확하고, 확실하게 구분해야 한다.
- 테스트할 수 없는 요구사항의 정확한 커버리지를 증명하기 위한 추가 활동을 정의해야 한다. (7.2.4.22)
- 소프트웨어 요구사항이 개발하고자 하는 시스템의 모습을 정확히 반영하고 있는지 검증해야 한다. (7.2.4.22)
  - ‘소프트웨어 요구사항 명세서’가 내부적으로 일관성을 유지하고 있는지 검증한다.
  - 개발자(요구사항 관리자, 설계자, 구현자)와 별도의 조직에서 명세서를 검증한다.
  - ‘소프트웨어 요구사항 명세서’가 확정되면 요구사항이 형식에 맞게 기술되었는지 검증한다.
- 요구사항이 안전성 관점에서 적합한지 여부를 검증해야 한다. (7.2.4.22)
  - 타당성: 소프트웨어 요구사항이 시스템 요구사항을 준수해야 한다.
  - 일관성: 어떠한 요구사항이라도 다른 요구사항과 충돌을 일으켜서는 안 된다.
  - 완전성: 모든 기능과 제약사항을 포함해야 한다.
  - 현실성: 하드웨어/소프트웨어 기술을 사용하여 해결 가능한 요구사항인지 예측해야 한다.
- 소프트웨어 요구사항이 시스템 요구사항으로부터 도출되었으며, 제대로 작성되었는지 여부를 명시해야 한다.
- ‘소프트웨어 요구사항 명세서’에 대한 검증 결과 및 부적합 사항에 대한 해결책과 권고사항을 명시해야 한다.

2.4.3. 소프트웨어 요구사항 검증 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 요구사항을 정의한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘소프트웨어 요구사항 명세서’, ‘종합 소프트웨어 테스트 명세서’

1.4 약어표

- 예시 1) ATP – Automatic Train Protection
- 예시 2) MMI – Man Machine Interface

2. 요구사항 검증 개요

- 요구사항 검증 활동에 필요한 조직, 소프트웨어 안전 무결성 등급, 사용 도구, 검증 기법, 제약 사항 등을 기록한다.

3. 검증 결과

- 검증 결과를 기술한다.

4. 검증 요약

- 검증 결과를 요약한다.
- 부적합사항을 기술하고 부적합사항의 해결책과 권고사항을 기술한다.

A. 부록

그림 90 소프트웨어 요구사항 검증 보고서 템플릿 (예시)

2.4.4. 소프트웨어 요구사항 검증 보고서 체크리스트

표 100 소프트웨어 요구사항 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	‘소프트웨어 요구사항 명세서 체크리스트’, ‘종합 소프트웨어 테스트 명세서 체크리스트’를 활용하여 검증한 결과를 기술하였는가?
	검증 결과에 종합적인 분석과 검증자, 수행일시를 포함하여 기술하였는가?
	검증 결과에 확인된 문제들과 적용된 조치사항을 기술하였는가?
품질 요구사항	‘소프트웨어 요구사항 명세서’를 준수하지 않는 항목이 기술되어 있는가?
	컴포넌트, 데이터, 자료구조, 알고리즘이 적당하지 않아 문제를 발생시키는 것이 있는가?
	오류 또는 불완전한 사항을 검지했는가?
	소프트웨어 검증 계획의 준수 여부 (준수하지 못한 경우의 중요도 기술)
	하드웨어와 소프트웨어 간의 제약사항이 기술되어 있는가?
(기타)	기타항목

제 2 절 아키텍처 및 설계

1. 개요

소프트웨어 요구사항 명세를 바탕으로 소프트웨어 아키텍처 구조 및 인터페이스를 정의하고, 설계하여 소프트웨어 컴포넌트 설계 및 구현에 필요한 정보를 제공 한다. 소프트웨어 컴포넌트의 통합 테스트와 소프트웨어/하드웨어 통합 테스트 명세를 작성하고 소프트웨어 아키텍처 및 설계를 검증한다.

1.1. 목표

- 소프트웨어 설계 기법을 정의하고 선정한다.
- 소프트웨어 요구사항을 충족시키는 소프트웨어 아키텍처를 개발한다.
- 하드웨어/소프트웨어 상호 작용의 안전 중요성을 식별하고 평가한다.
- 소프트웨어 요구사항 명세의 안전 무결성 등급으로 소프트웨어를 설계한다.
- 검증 및 테스트 요구사항을 고려하여 시스템과 소프트웨어의 테스트가 용이하도록 설계한다.

1.2. 범위

소프트웨어 개발 생명 주기에서 IEC 62279 7.3절에 해당하는 소프트웨어 아키텍처 및 설계 단계에 대해 설명한다.

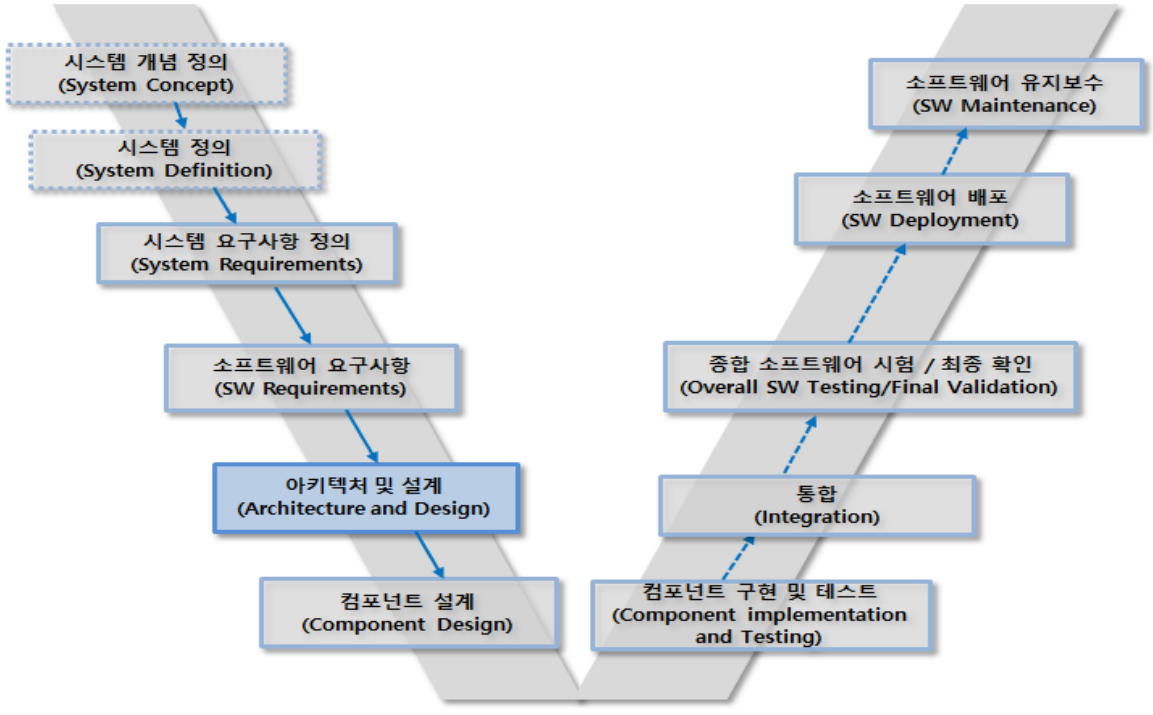


그림 91 소프트웨어 개발 생명주기 - 아키텍처 및 설계 단계

1.3. 시작 기준

- ○ 소프트웨어 요구사항 도출
- ○ 소프트웨어 테스트 명세 작성
- ○ 소프트웨어 요구사항 검증

1.4. 완료 기준

- ○ 소프트웨어 아키텍처 설계 완료
- ○ 기존 소프트웨어 재사용 검토 완료
- ○ 소프트웨어 아키텍처 안전 분석 완료
- ○ 소프트웨어 인터페이스 명세 작성
- ○ 소프트웨어 설계 및 안전 분석 완료
- ○ 소프트웨어 통합 테스트 계획 수립
- ○ 소프트웨어 / 하드웨어 통합 테스트 계획 수립
- ○ 소프트웨어 아키텍처 및 설계 검증

1.5. 입력물

- ○ 소프트웨어 요구사항 명세서

1.6. 출력물

표 101 아키텍처 및 설계 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 아키텍처 명세서	HR	HR	HR	HR	HR
소프트웨어 설계 명세서	HR	HR	HR	HR	HR
소프트웨어 인터페이스 명세서	HR	HR	HR	HR	HR
소프트웨어 통합테스트 명세서	HR	HR	HR	HR	HR
소프트웨어/하드웨어 통합테스트 명세서	HR	HR	HR	HR	HR
소프트웨어 아키텍처 및 설계 검증 보고서	HR	HR	HR	HR	HR

1.7. 역할 및 책임

표 102 아키텍처 및 설계 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
아키텍처 및 설계	9. 소프트웨어 아키텍처 명세서	DES	VER	VAL
	10. 소프트웨어 설계 명세서	DES	VER	VAL
	11. 소프트웨어 인터페이스 명세서	DES	VER	VAL
	12. 소프트웨어 통합테스트 명세서	INT	VER	VAL
	13. 소프트웨어/하드웨어 통합테스트 명세서	INT	VER	VAL
	14. 소프트웨어 아키텍처 및 설계 검증 보고서	VER		VAL
DES (Designer) 설계자 INT (Integrator) 통합자 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 아키텍처 및 설계 주요 활동

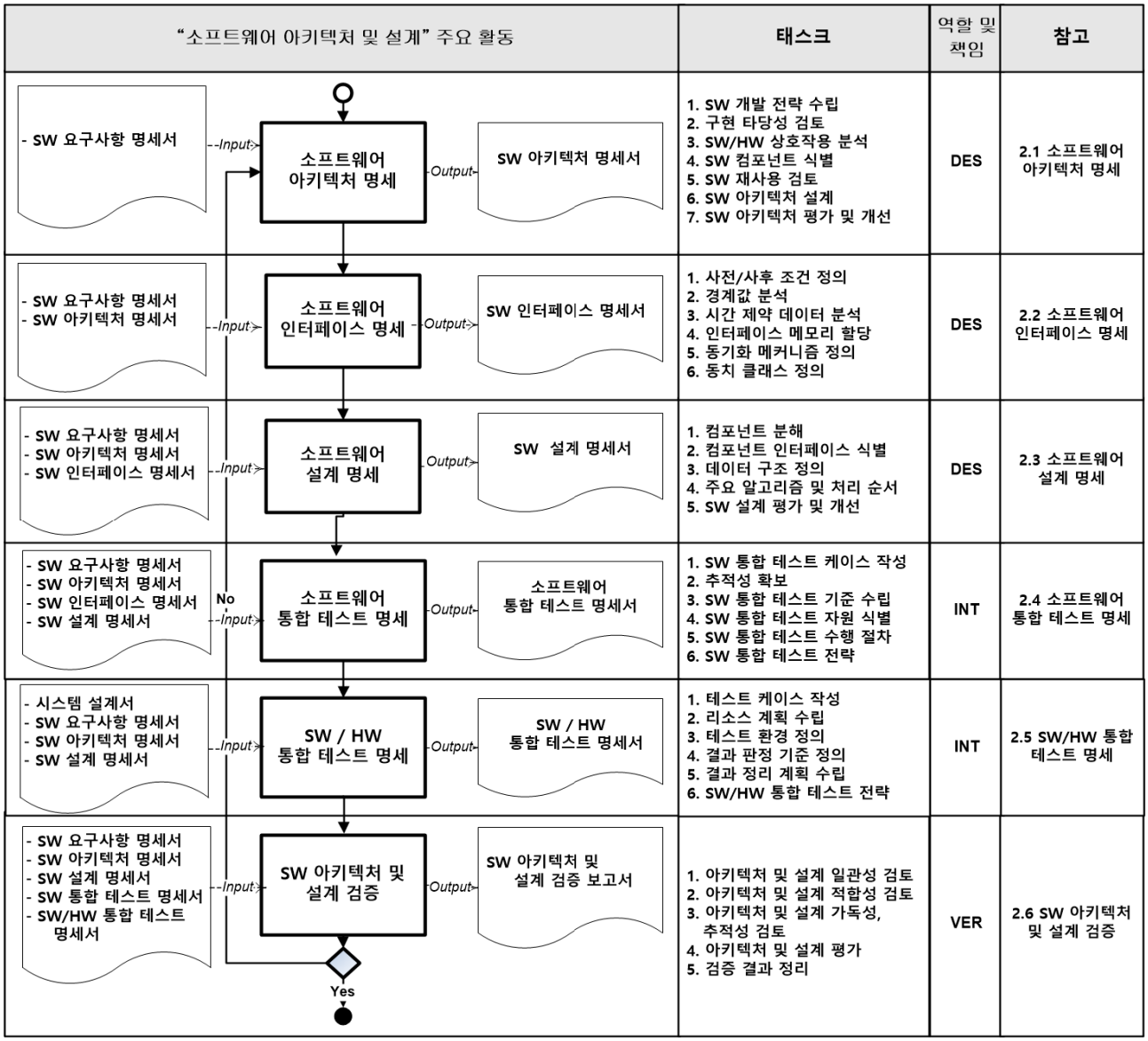


그림 92 소프트웨어 아키텍처 및 설계 주요 활동

표 103 소프트웨어 아키텍처 및 설계 단계

활 동 ID	활 동 명	설 명
SADS.01	소프트웨어 아키텍처 명세	• 소프트웨어 요구사항을 기반으로 구현 타당성을 검토한다. • 소프트웨어 / 하드웨어 상호작용을 분석한다. • 소프트웨어 컴포넌트를 식별한다, • 기존 소프트웨어 재사용을 검토 한다 • 컴포넌트의 의존성과 독립성을 분석한다, • 소프트웨어 개발전략을 수립 한다, • 소프트웨어 아키텍처 평가 및 개선을 수행한다.

활 동 ID	활 동 명	설 명
SADS.02	소프트웨어 인터페이스 명세	• 소프트웨어 컴포넌트와 전체 소프트웨어 간의 인터페이스에 대한 사전 / 사후 조전을 정의한다. • 인터페이스의 경계값을 분석한다. • 시간 제약 입출력 데이터를 분석한다. • 인터페이스 버퍼에 대한 메모리를 할당한다. • 기능간의 동기화 방식을 확인한다. • 동치 클래스를 정의한다.
SADS.03	소프트웨어 설계 명세	• 소프트웨어 컴포넌트를 컴포넌트 설계와 테스트 단위로 분해한다. • 소프트웨어 컴포넌트와 환경과의 인터페이스를 정의한다. • 소프트웨어 컴포넌트간의 인터페이스를 정의한다. • 데이터 구조를 식별한다. • 요구사항을 컴포넌트에 할당하고 추적한다. • 주요 알고리즘과 처리 순서를 정의한다. • 오류 보고 메커니즘을 구성한다.
SADS.04	소프트웨어 통합 테스트 명세	• 소프트웨어 통합 테스트 전략을 수립한다. • 테스트 케이스 추출 방법을 선정하고 작성한다. • 추적 메트릭스를 작성한다, • 결과 판정 기준을 수립한다. • 테스트 수행 환경을 식별한다, • 문제 발견 및 해결 절차를 작성한다.
SADS.05	소프트웨어/하드웨어 통합 테스트 명세	• 소프트웨어 / 하드웨어 통합 테스트 전략을 수립한다. • 테스트 케이스 추출 방법을 선정하고 작성한다, • 추적 메트릭스를 작성한다, • 결과 판정 기준을 수립한다. • 테스트 수행 리소스를 식별한다. • 문제 발견 및 해결 절차를 작성한다.
SADS.06	소프트웨어 아키텍처 및 설계 검증	• 아키텍처 및 설계의 일관성을 검증한다. • 아키텍처 및 설계의 적합성을 검증한다. • 아키텍처 및 설계의 가독성을 검증한다. • 아키텍처 및 설계의 추적성을 검증한다. • 아키텍처 및 설계 검증 결과를 정리하고 판정한다.

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 소프트웨어 아키텍처 명세

소프트웨어 요구사항을 기반으로 구현 타당성 검토, 소프트웨어 / 하드웨어 상호작용 분석, 소프트웨어 컴포넌트 식별, 기존 소프트웨어 재사용 검토, 컴포넌트 의존성 분석, 소프트웨어 개발전략 수립, 아키텍처 평가 및 개선을 수행한다.

2.1.1. 소프트웨어 아키텍처 명세 수행 절차

○ 소프트웨어 아키텍처 명세 수행 흐름

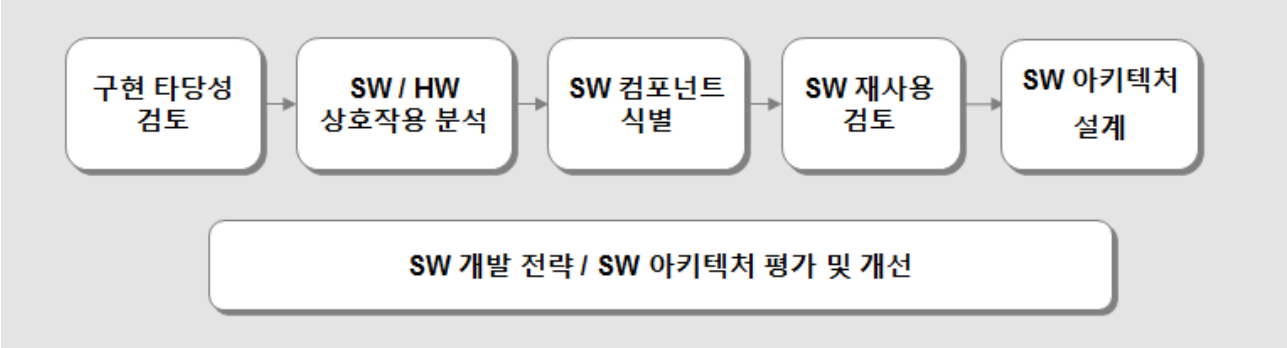


그림 93 소프트웨어 아키텍처 명세 수행 흐름도

○ 소프트웨어 아키텍처 명세 절차

표 104 소프트웨어 아키텍처 명세 절차

항 목	설 명
구현 타당성 검토	- 소프트웨어 요구사항의 설계와 구현 가능성을 검토한다. - 소프트웨어 아키텍처의 평가 가능성 및 유지보수성을 고려한다.
소프트웨어/하드웨어 상호작용 분석	- 하드웨어 / 소프트웨어의 상호 작용을 모두 식별한다. - 식별된 하드웨어 / 소프트웨어 상호 작용을 상세하게 분석한다. ・소프트웨어와 하드웨어간의 인터페이스와 데이터 경로 ・타이밍 동작 및 처리 순서

항 목	설 명
소프트웨어 컴포넌트 식별	- 소프트웨어 요구사항의 하위 집합을 모두 포함하도록 모든 소프트웨어 컴포넌트를 식별한다. - 신규 또는 기 존재 컴포넌트 여부를 확인한다. - 기 존재 컴포넌트 검증 여부와 검증 조건을 확인한다. - 컴포넌트의 소프트웨어 안전 무결성 등급을 확인한다. - 소프트웨어 컴포넌트를 형상 관리 시스템으로 명확하게 식별하고 개별적으로 버전 관리한다.
소프트웨어 재사용 검토	- 기존 소프트웨어 재사용으로 충족하려는 요구사항을 기술한다, - 기존 소프트웨어의 환경에 대한 가정을 기술한다. - 다른 소프트웨어 부분과의 인터페이스를 기술한다. - 기존 소프트웨어의 실패 가능성 및 전체 소프트웨어에 대한 영향 분석한다. - 기존 소프트웨어의 고장 탐지 및 이러한 고장으로부터 시스템을 보호하기 위한 전략 정의한다. - 기존 소프트웨어가 할당된 요구사항을 충족함을 검증 및 확인한다. - 기존 소프트웨어의 오류가 감지되고 기존 소프트웨어가 통합된 시스템이 이러한 오류로부터 보호됨을 검증 및 확인한다. - 기존 소프트웨어의 환경에 대한 가정이 성취되었음을 검증 및 확인한다.
소프트웨어 아키텍처 설계	- 소프트웨어 아키텍처를 구상하고 상세하게 작성한다. - 소프트웨어와 응용 데이터 / 알고리즘 간의 상세한 인터페이스를 명시한다. - 소프트웨어 아키텍처의 크기와 복잡성이 균형을 이루도록 구성한다.
소프트웨어 개발 전략	- 요구 소프트웨어 안전 무결성 등급에 따른 소프트웨어 개발 전략을 수립한다. - 선정한 기법, 수단 및 도구가 요구 소프트웨어 안전 무결성 등급의 소프트웨어 요구사항 명세를 만족함을 실증한다. - 소프트웨어 아키텍처는 애플리케이션의 안전 부분의 크기와 복잡성을 최소화하는 것을 목표로 한다.
소프트웨어 평가 및 개선	- 컴포넌트가 간섭으로부터 독립적임의 증거는 소프트웨어 아키텍처 명세에 기록한다. - 장애 회피와 장애 대응 전략 간의 균형을 이루어 장애 처리 수단을 수립한다.

2.1.2. 소프트웨어 아키텍처 명세 지침

○ 소프트웨어 요구사항 명세를 기반으로 소프트웨어 아키텍처 명세를 작성한다. (7.3.4.1)

○ 제안 소프트웨어 아키텍처를 수립하고 상세화 한다. (7.3.4.2)

표 105 소프트웨어 아키텍처 뷰 (예시)

항 목	설 명
분할 뷰	컴포넌트간의 분할 관계에 중점
사용 뷰	컴포넌트간의 기능 의존에 중점
계층 뷰	컴포넌트간의 서비스 사용과 제공에 중점
컴포넌트와 커넥터 뷰	실행 시간에 나타나는 컴포넌트간의 연결에 중점
파이프와 필터 뷰	입력과 출력 스트림의 변환에 중점

○ 요구 소프트웨어 안전 무결성 등급의 요구사항이 구현 가능한지 분석한다. (7.3.4.3)

표 106 구현 가능성 분석 주요 기법 (예시)

항 목	설 명
난이도 분석	- 요구사항 달성의 어려움 정도 식별 - 개발 조직 역량을 감안하여 조정
업무량 산정	- 소프트웨어 규모 산정에 따른 필요 공수 계산 - 개발 조직 감안하여 개발 기간 산정

○ 소프트웨어 아키텍처는 애플리케이션의 안전 부분의 크기와 복잡성을 최소화하는 것을 목표로 한다. (7.3.4.3)

표 107 소프트웨어 아키텍처 복잡성 최소화 설계 속성 (예시)

항 목	설 명
단순화	- 소프트웨어 요구사항을 실현할 수 있는 가장 단순한 설계 개발
모듈화	- 컴포넌트 간은 낮은 결합도를 갖도록 설계 - 컴포넌트 내부는 높은 응집도를 갖도록 설계
은닉화	- 복잡한 세부의 사항들을 자세히 설명하지 않고 통칭함 - 주요 사항이 드러나지 않도록 감춤 - 인터페이스를 명확히 하고 낮은 결합도를 갖도록 은닉화 설계

○ 모든 하드웨어/소프트웨어 상호 작용을 식별하고 분석하여 상세하게 작성한다.(7.3.4.4)

- 5장 가이드 적용 사례 아키텍처 및 설계 명세, [그림 178 차상 MMI 시스템 개념도 (예시)] 참조

※ 하드웨어/소프트웨어 상호 작용 분석 및 식별 방안 예시

하드웨어와 소프트웨어 간의 데이터 흐름을 분석하여 적절한 단위로 구분

・정적 측면과 동적 측면을 고려하여 식별

표 108 하드웨어/소프트웨어 상호 작용 분석 및 식별 (예시)

항 목	설 명
정적 측면	- 인터페이스 및 데이터 경로 ・요구사항을 분석하여 식별한 데이터에 대한 하드웨어 / 소프트웨어 간의 인터페이스 및 데이터 경로 분석
동적 측면	- 동기화 및 순차 처리 ・데이터 흐름을 수행하기 위한 동기화 및 순차 처리의 제어 흐름을 분석

○ 모든 소프트웨어 컴포넌트를 식별하고 각 컴포넌트에 대하여 다음을 확인한다.(7.3.4.5)

- 신규 또는 기 존재 컴포넌트 여부
- 기 존재 컴포넌트 검증 여부와 검증 조건
- 컴포넌트의 소프트웨어 안전 무결성 등급

※ 소프트웨어 컴포넌트 식별 방안 예시

표 109 소프트웨어 컴포넌트 식별 (예시)

항 목	설 명
유즈케이스 분석	- 컴포넌트간의 분할 관계에 중점
데이터 흐름 분석	- 요구사항의 각 항목들을 분석하여 데이터를 식별 - 데이터에 대한 활동 및 흐름을 식별
제어 흐름 분석	- 동기화, 메시지 전송, 데이터 교환, 시작, 중단 등의 제어 흐름을 분석
인터페이스 상호작용 분석	- 초기 구조의 상호 작용을 모두 식별 - 식별된 상호 작용을 상세하게 분석 - 인터페이스 데이터 경로와 타이밍 동작 및 처리 순서 분석
계층 구조와 스테레오 타입	- 계층 간의 요소와 속성을 분석하고 관계를 스테레오 타입으로 정의

- 5장 가이드 적용 사례 아키텍처 및 설계 명세, [표 242 차상 MMI 소프트웨어 컴포넌트 (예시)]

○ 소프트웨어 컴포넌트에 다음의 사항을 포함해야 한다. (7.3.4.6)
- 소프트웨어 요구사항의 하위 집합을 모두 포함하도록 소프트웨어 컴포넌트를 식별
- 소프트웨어 컴포넌트를 형상 관리 시스템 내에서 명확하게 식별하고 독립적으로 버전 관리

○ 기존 소프트웨어를 재사용하기 위해서는 모든 소프트웨어 안전 무결성 등급에 대하여 다음 정보를 명확하게 식별하고 문서화해야 한다. (7.3.4.7)
- 기존 소프트웨어 재사용으로 충족하려는 요구사항
- 기존 소프트웨어의 환경에 대한 가정
- 다른 소프트웨어 부분과의 인터페이스

○ 모든 소프트웨어 안전 무결성 등급에 대해 전체 소프트웨어 검증 프로세스에 기존 소프트웨어를 포함하여 수행한다. (7.3.4.7)

○ 소프트웨어 안전 무결성 등급이 SIL 3 또는 SIL 4의 경우 다음의 안전활동을 수행해야 한다. (7.3.4.7)
- 기존 소프트웨어의 예상 가능한 실패 및 전체 소프트웨어에 대한 영향 분석
- 기존 소프트웨어의 고장 탐지 및 이러한 고장으로부터 시스템을 보호하기 위한 전략 정의
- 기존 소프트웨어가 할당된 요구사항을 충족함을 검증 및 확인
- 기존 소프트웨어의 오류가 감지되고 기존 소프트웨어가 통합된 시스템이 이

러한 오류로부터 보호됨을 검증 및 확인

- 기존 소프트웨어의 환경에 대한 가정이 성취되었음을 검증 및 확인

○ 기존 소프트웨어에 대하여 다음을 포함하여 충분히 자세하고 완전한 설명을 기술해야 한다. (7.3.4.7)

- 사용된 기능으로의 제한
- 기능, 제약 및 증거
- 하드웨어 및 소프트웨어 제약 사항
- 소프트웨어 설계 기능, 특성, 작동 및 특징

※ 기존 소프트웨어 재사용 방안 예시

사용의 정의나 조건이 이미 출시되어 작동 중인 제품과 동일하거나, 많은 공통점을 가진 모든 유형의 제품에 적용된 소프트웨어

○ 기존 소프트웨어의 검증에 통계적 증거를 사용할 수 있다. (7.3.4.7)

※ 기존 소프트웨어의 통계적 증거 검증 방안 예시

· 다른 목표에 부분 또는 전체적으로 적용되어 수행중인 철도 어플리케이션
· 철도 산업에서 관찰 가능한 사고율 이하로 사용 중임
· 다른 안전관련 산업에서 관찰 가능한 사고율 이하로 사용 중임
· 널리 사용되고 있는 COTS

○ 가능하면 표준에 따라 개발하고 검증한 기존 소프트웨어 컴포넌트를 재사용한다. (7.3.4.8)

○ 소프트웨어가 여러 소프트웨어 안전 무결성 등급의 컴포넌트로 구성되는 경우 상위 소프트웨어 안전 무결성 등급 컴포넌트와 하위 소프트웨어 간에 독립성이 있다는 증거가 없는 한 모든 소프트웨어 컴포넌트는 최상위 수준에 속하는 것으로 취급한다. (7.3.4.9)

※ 소프트웨어 컴포넌트 안전 무결성 등급 할당 예시

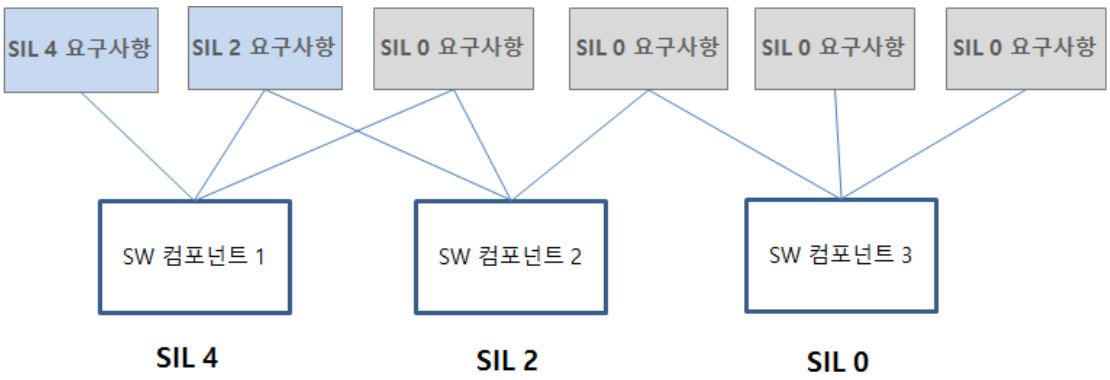


그림 94 소프트웨어 컴포넌트 안전 무결성 등급 할당 (예시)

○ 컴포넌트 독립성의 증거는 소프트웨어 아키텍처 명세에 기록한다. (7.3.4.9)

※ 소프트웨어 컴포넌트 독립성 증거 예시

- 안전 무결성 등급별로 소프트웨어를 파티션 하여 각 등급별 요건에 맞도록 컴포넌트를 개발
- 파티션은 연계 고장을 방지하기 위한 목적으로 결함을 격리하기 위해 사용
- 파티션 간에 종속 고장이 전파되지 않도록 아키텍처를 구성하여 종속고장 또는 의존장애가 발생하지 않음을 보장

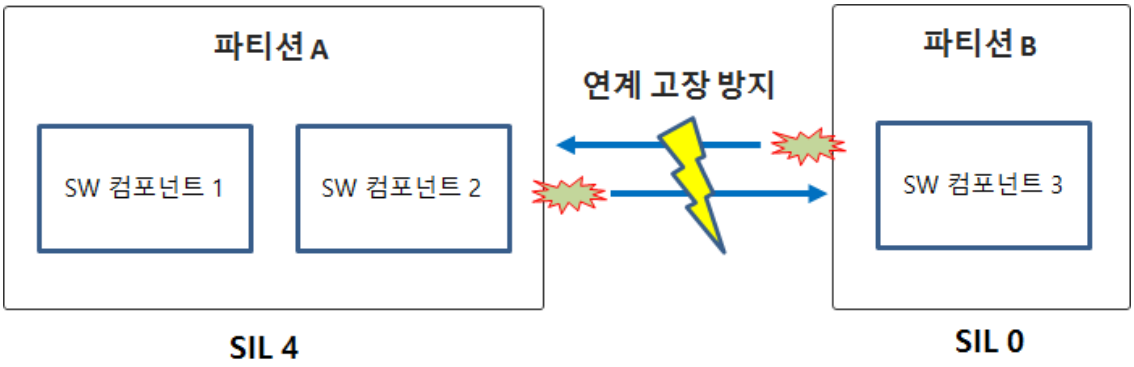


그림 95 소프트웨어 컴포넌트 독립성 증거 (예시)

○ 요구 소프트웨어 안전 무결성 등급에 따른 소프트웨어 개발 전략을 수립한다. (7.3.4.10)

○ 소프트웨어 개발 전략은 다음과 같이 정제하고 구조화하여 소프트웨어 아키텍처 명세에 수립한다. (7.3.4.10)

- 완전성, 일관성, 명확성, 정확성, 명료성
- 검증 가능성, 시험 가능성, 실행 가능성, 유지 보수 가능성
- 소프트웨어 요구사항 명세 추적성

※ 소프트웨어 아키텍처 개발 전략 예시

표 110 소프트웨어 아키텍처 주요 품질 속성 (예시)

항 목	설 명
완전성	- 제공되는 기능들이 요구사항들을 얼마나 커버하는지를 의미 - 아키텍처는 시스템의 요구사항과 실행 시간 자원에 대한 제약사항 모두를 만족해야함
일관성	- 아키텍처는 유사한 방법으로 유사한 일들을 수행하고, 일괄된 방식으로 적용될 수 있어야 함 - 모든 수준의 시스템 설계를 통합하는데 근간이 되는 개념

항 목	설 명
검증용이성	- 아키텍처가 얼마나 검증을 수행하기에 용이한지를 정의 ・요구사항에 대한 명확한 추적이 가능 ・적절한 길이로 구분되어 있음 ・명칭들이 직관적으로 이해하기 쉬움
시험가능성	- 아키텍처가 얼마나 테스트 수행이 용이한지를 정의 - 적은 테스트 케이스로 커버될 수 있는 단순한 아키텍처가 테스트 수행이 용이
실행가능성	- 주어진 시간 내에 주어진 자원을 가지고 해당 아키텍처를 구현할 수 있는지를 의미 - 아키텍처 설계 구현의 실행이 가능한지 나타냄
유지보수가능성	- 아키텍처가 얼마나 수정 및 변경 적용이 용이한지 정의 ・유지보수 변경을 반영하기 위한 수정 범위가 작음 ・유지보수 변경을 반영하기 위한 수정이 특정 영역에 집중

○ 장애 회피와 장애 대응 전략 간의 균형을 이루어 장애 처리 수단을 수립한다. (7.3.4.11)

※ 소프트웨어 장애 처리 주요 수단 예시

표 111 장애 회피와 장애 대응 (예시)

항 목	설 명
장애 회피	- 소프트웨어 다중화 ・처리를 서로 다르게 중복 구현하여 결과를 비교하여 결정 - 감시기 ・주어진 시간 내에 기능 수행이 완료되는지 감지 - 메모리 값 검사 ・메모리 값의 변조 여부 검사 - 영역 복제 ・메모리를 복제하고 비교하여 오류를 감지
장애 대응	- 점진적 기능 감소 ・고장 시 덜 중요한 기능을 제한하고 중요 기능을 유지 - 복구 블록 ・첫 섹션 실행 결과의 오류 판단 시 복구 섹션 실행 - 후방 복구 ・복구 지점 설정 후 오류 시 복구 지점부터 재실행 - 전방 복구 ・오류 유형별 오류 처리 로직 실행 - 재시도 복구 ・오류 시에 반복 재실행하여 복구 시도

○ 선정한 기법, 대책 및 도구가 요구 소프트웨어 안전 무결성 등급의 소프트웨어 요구사항 명세를 만족함을 증명한다. (7.3.4.12)

※ 소프트웨어 도구 식별 종류 및 도구 선정 항목 예시

표 112 소프트웨어 도구 종류 및 선정 (예시)

항 목	설 명
도구 종류	- 프로젝트 관리 - 형상 관리 - 요구사항 관리 - 소프트웨어 설계 - 소프트웨어 컴파일러 - 정적 분석 - 안전 분석
도구 선정	- 공급업체 - 버전 - 사용 목적 및 기능 - 알려진 버그 및 문제점 - 도구 설정 - 사용 환경 - 요구되는 안전 무결성 등급

○ 소프트웨어가 응용 프로그램 데이터 또는 알고리즘으로 구성된 경우, 소프트웨어와 응용 데이터 / 알고리즘 간의 상세한 인터페이스를 명시한다. (7.3.4.13)

○ 표 A.3의 기법과 대책을 선택하여 소프트웨어 아키텍처를 작성한다. (7.3.4.14)

○ 소프트웨어 아키텍처의 크기와 복잡성은 균형을 이루도록 개선한다. (7.3.4.15)

○ 프로토타이핑을 사용하여 요구사항과 결과에 대한보다 상세한 뷰를 얻을 수 있다. (7.3.4.16)

○ 프로토타입의 코드는 소스 코드와 개발 프로세스 및 산출 문서가 이 표준을 충족한다는 것을 입증한 경우에만 대상 시스템에 사용할 수 있다. (7.3.4.17)

2.1.3. 소프트웨어 아키텍처 명세서 템플릿

1. 개요

1.1 범위

- ‘소프트웨어 아키텍처 명세서’에는 소프트웨어 인터페이스 명세와 소프트웨어 설계 명세에 필요한 소프트웨어 아키텍처를 기술한다.

1.2 목적

- ‘소프트웨어 아키텍처 명세서’를 작성하는 설계자와 참조하는 통합자, 검증자의 목적을 작성한다. ‘소프트웨어 아키텍처 명세서‘를 기준으로 소프트웨어 인터페이스 명세, 소프트웨어 설계 명세 및 개발을 수행하고 검증된 소프트웨어 통합 및 통합 검증을 수행한다.

1.3 시스템 개요

- 소프트웨어 아키텍처 명세 대상 시스템의 목적에 대해 간략한 설명을 기술한다.

1.4 형상 식별

- 소프트웨어 아키텍처 명세 항목의 식별을 위한 항목 식별자를 설명한다.

1.5 관련 표준

- ‘소프트웨어 아키텍처 명세서’와 관련된 표준을 설명한다.

2. 아키텍처 배경

2.1 아키텍처 소개

- 소프트웨어 아키텍처 명세 이유를 설명한다. ‘소프트웨어 아키텍처 명세서’를 참조하는 관련자들에게 아키텍처 선정 내역을 설명한다.

2.2 가정 사항

- 소프트웨어 아키텍처 명세 시에 수립한 가정 사항에 대하여 기술한다.

3. 소프트웨어/HW 상호작용

3.1 요소, 관계, 속성

- 소프트웨어 / 하드웨어 상호 작용의 요소, 관계, 속성을 기술한다.

3.2 상세 내용

- 소프트웨어 / 하드웨어 상호 작용의 상세 내용을 기술한다.

4. 컴포넌트 식별

4.1 OOO 컴포넌트

- OOO 컴포넌트의 역할 및 기능에 대해 상세히 기술한다.

4.2 OOO 컴포넌트

- OOO 컴포넌트의 역할 및 기능에 대해 상세히 기술한다.

4.3 OOO 컴포넌트

- OOO 컴포넌트의 역할 및 기능에 대해 상세히 기술한다.

... (반복 작성)

5. 재사용 컴포넌트

5.1 OOO 컴포넌트

- OOO 재사용 컴포넌트의 속성을 상세히 기술한다.

5.2 OOO 컴포넌트

- OOO 재사용 컴포넌트의 속성을 상세히 기술한다.

- ... (반복 작성)

5. 아키텍처 명세

5.1 컴포넌트 아키텍처

- 소프트웨어 컴포넌트 아키텍처를 구성하고 기능을 할당한다.

5.2 인터페이스

- 소프트웨어와 응용 데이터 / 알고리즘 간의 인터페이스를 기술한다.

6. 요구사항 추적 매트릭스

- 소프트웨어 요구사항과 소프트웨어 아키텍처 명세 간의 추적성을 기술한다.

7. 용어

- '아키텍처 명세서'에 사용된 약어, 용어에 대해 기술한다.

그림 96 소프트웨어 아키텍처 명세서 템플릿 (예시)

2.1.4. 소프트웨어 아키텍처 명세서 체크리스트

○ '소프트웨어 아키텍처 명세서 체크리스트'를 통해 명세서의 품질을 검토한다.

표 113 소프트웨어 아키텍처 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
문서 구성	문서에서 사용하는 정의가 명확한가?
	문서에 근거, 제약사항, 배경, 아키텍처의 후보에 관한 정보가 있는가?

구 분	점검 사항
	소프트웨어 아키텍처 명세의 요소와 관련된 소프트웨어 요구사항을 추적할 수 있는가?
	문서에 불필요한 정보가 있지 않은가?
완전성	모든 아키텍처 항목이 표준을 준수하고 추적되고 있는가?
	‘아키텍처 명세서’ 내에 일관성이 있는가?
	해석하기에 애매한 표현이 있는가?
	누락된 항목은 있는가?
정확성	가능하지 않은 부정확한 가정 사항이 있는가?
	사실에 기반하지 않은 항목이나 사실이 틀린 항목이 있는가?
	이해당사자가 명확하게 식별되고, 이해당사자의 요구사항을 만족하도록 기술되어 있는가?
	모호하거나, 해당 주제에 대한 해석이 다양하게 보여 질 수 있는가?
일치성	구현 타당성 검토가 완료 되었는가?
	소프트웨어 / 하드웨어 상호작용이 모두 식별 되었는가?
	모든 소프트웨어 컴포넌트가 식별 되었는가?
	재사용 소프트웨어가 식별되고 기술되었는가?
	컴포넌트의 의존성과 독립성이 분석되어 안전 무결성 등급이 할당되었는가?
(기타)	기타항목

2.2. 소프트웨어 인터페이스 명세

소프트웨어 인터페이스 명세는 소프트웨어 요구사항 명세와 소프트웨어 아키텍처 명세의 기반 하에 소프트웨어 컴포넌트와 전체 소프트웨어 간의 인터페이스에 대한 기술적 사항을 작성한다.

2.2.1. 소프트웨어 인터페이스 명세 수행 절차

○ 소프트웨어 인터페이스 명세 수행 흐름

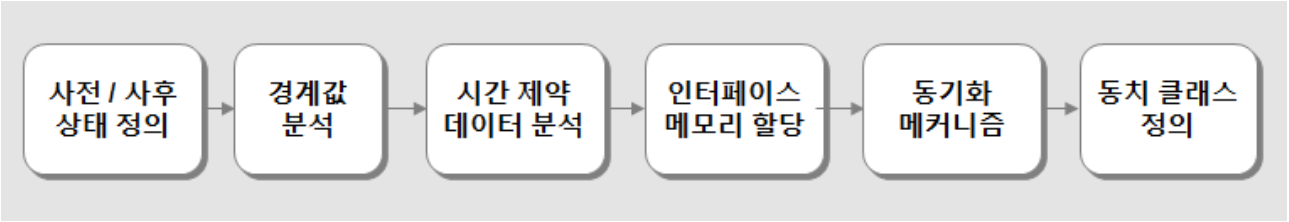


그림 97 소프트웨어 인터페이스 명세 수행 흐름도

○ 소프트웨어 인터페이스 명세 절차

표 114 소프트웨어 인터페이스 명세 절차 설명

항 목	설 명
사전 / 사후 상태 정의	- 인터페이스 수행 전의 상태를 식별하고 정의한다. - 인터페이스 수행 후의 생태를 식별하고 정의한다.
경계값 분석	- 인터페이스의 특정 데이터에 대한 경계값을 정의하고 기술한다. - 인터페이스에서경계값을 초과했을 때의 동작을 기술한다. - 인터페이스에서 값이 경계에 있을 때의 동작을 기술한다.
시간제약 데이터 분석	- 시간 제약 입력과 출력 데이터의 조건과 정상 동작에 대한 요구사항을 식별하고 기술한다. - 시간 제약 입력과 출력 데이터의 예외 관리 방안을 기술한다.
인터페이스 메모리 할당	- 인터페이스 버퍼에 메모리를 할당한다. - 메모리를 할당할 수 없거나 모든 버퍼가 가득 찬 경우 감지하는 메커니즘을 기술한다.
동기화 메커니즘	- 기능 간의 동기화 로직을 분석한다. - 기능 간의 동기화 메커니즘을 개발한다.
동치 클래스 정의	- 특정 데이터와 소프트웨어 기능에 대한 동치 클래스를 정의하고 기술한다. - 사용하지 않거나 금지된 동치 클래스를 정의한다.

2.2.2. 소프트웨어 인터페이스 명세 지침

○ 소프트웨어 인터페이스 명세에 ‘소프트웨어 요구사항 명세서’와 ‘소프트웨어 아키텍처 명세서’를 기반으로 소프트웨어 컴포넌트와 전체 소프트웨어의 경계에 대한 모든 인터페이스를 기술한다. (7.3.4.18)

○ 인터페이스 명세에 다음의 사항을 포함해야 한다. (7.3.4.19)

- 사전/사후 조건
- 모든 특정 데이터에 대한 경계값 정의 및 기술
- 경계값을 초과했을 때의 동작
- 값이 경계에 있을 때의 동작
- 시간-필수적인 입력과 출력 데이터
  - 시간 제약 및 올바른 동작을 위한 요구사항
  - 예외 처리
- 인터페이스 버퍼에 할당된 메모리, 메모리를 할당할 수 없거나 모든 버퍼가 가득 찬 경우 감지하는 메커니즘 (해당할 경우)

표 115 버퍼 오버플로 검지 및 회피 기법 (예시)

항 목	설 명
안전 라이브러리	- 버퍼 Overflow를 일으킬 여지가 있는 표준 라이브러리 함수를 사용하지 말고, 버퍼 경계 검사를 수행하는 안전 라이브러리 사용
경계값 검사	- 입력 값 검사 : 모든 입력을 받아들이고 특정 입력을 차단하는 방식에서 특정 입력만 받아들이고 나머지를 차단하는 방식 적용 - 버퍼 경계 검사 : 매개변수 사용 전에 경계 내에 존재하는지 확인
정적 코드 분석	- 주로 도구를 사용하여 소스 코드상의 오버플로, 언더플로 등을 테스트

※ 버퍼 오버플로 감지 및 회피 주요 방안 예시

- 기능 간의 동기화 메커니즘
- 모든 특정 데이터와 그것을 사용하는 각 소프트웨어 기능에 대한 모든 동치 클래스(equivalence classes)의 정의 및 기술
- 사용하지 않거나 금지된 동치 클래스 정의

※ 동치 클래스(equivalence classes) 식별 및 정의 방안 예시

・동등 관계에 속할 수 있는 데이터의 집합

・정적 분석을 통해 식별

・대표원(representative) : 동치 클래스에서 대표되는 하나의 값을 선택

・예) 동일 경로로 실행되는 데이터 집합

○ 기능 수행 시 사용되지 않는 범위를 포함하여, 데이터 유형에 따른 전체 범위 값의 모든 인터페이스 입출력 데이터를 정의한다. (7.3.4.19)

○ 데이터 유형은 다음과 같은 사항을 포함한다. (7.3.4.19)

- 기능 및 프로시저의 입력 변수와 출력 결과
- 전보 또는 통신 패킷에 지정된 데이터
- 하드웨어로 부터의 데이터

○ 5장 가이드 적용 사례 아키텍처 및 설계 명세, [표 244 소프트웨어 인터페이스 (예시)]

2.2.3. 소프트웨어 인터페이스 명세서 템플릿

1. 개요

- ‘소프트웨어 인터페이스 명세서’ 의 전반적인 소개와 내용을 기술한다.

1.1 목적

- ‘소프트웨어 인터페이스 명세서 ‘의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- ‘소프트웨어 인터페이스 명세서 ‘에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 나열한다.

1.3.1 참조 문서

- 참조 문서를 나열한다.

1.4 약어

- ‘소프트웨어 인터페이스 명세서’ 에 포함되어 있는 약어를 설명한다.

2. 인터페이스 식별

2.1 OOO 인터페이스

- 인터페이스 설명 및 관련 기능 기술
- 속성 정의

・사전 / 사후 조건

・경계값 정의 및 동작

· 시간 관련 입력과 출력 데이터 관련 사항

· 메모리 정보

· 통치 클래스 정보

2.2 OOO 인터페이스

- 인터페이스 설명 및 관련 기능 기술

- 속성 정의

· 사전 / 사후 조건

· 경계값 정의 및 동작

· 시간 관련 입력과 출력 데이터 관련 사항

· 메모리 정보

· 통치 클래스 정보

2.3 ... (반복 작성)

3. 아키텍처 추적 매트릭스

- 소프트웨어 아키텍처 명세와 소프트웨어 인터페이스 명세 간의 추적성을 기술한다.

그림 98 소프트웨어 인터페이스 명세서 템플릿 (예시)

2.2.4. 소프트웨어 인터페이스 명세서 체크리스트

○ ‘소프트웨어 인터페이스 명세서‘ 체크리스트를 통해 명세서의 품질을 검토한다.

표 116 소프트웨어 인터페이스 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
문서 구성	채택한 템플릿이나 표준을 잘 준수하는가?
	각 인터페이스의 정의(요소, 관계, 속성)를 설명하고 있는가?
	문서에서 사용하는 정의가 명확한가?
	문서에 근거, 제약사항, 배경에 관한 정보가 있는가?

구 분	점검 사항
	소프트웨어 인터페이스 명세의 요소와 소프트웨어 요구사항을 추적할 수 있는가?
	문서에 불필요한 정보가 있지 않은가?
	모든 인터페이스 항목이 표준을 준수하고 추적되고 있는가?
완전성	‘소프트웨어 인터페이스 명세서’ 내에 일관성이 있는가?
	해석하기에 애매한 표현이 있는가?
	누락된 항목은 있는가?
정확성	가능하지 않은 부정확한 가정 사항이 있는가?
	사실에 기반하지 않은 항목이나 사실이 틀린 항목이 있는가?
	모호하거나, 해당 주제에 대한 해석이 다양하게 보여 질 수 있는가?
일치성	인터페이스의 사전 / 사후 조건이 식별 되었는가?
	인터페이스의 경계값이 분석되고 정의 되었나?
	입출력 데이터의 시간 제약 사항이 분석되었는가?
	인터페이스 버퍼에 메모리가 할당 되었나?
	기능간의 동기화 방식이 정의 되었는가?
	동치 클래스를 식별하고 정의하였나?
(기타)	기타항목

2.3. 소프트웨어 설계 명세

소프트웨어 컴포넌트를 분해하여, 소프트웨어 컴포넌트와 그들의 상호작용을 계층 구조로 나타내고, 소프트웨어 컴포넌트 사이의 인터페이스와 데이터 흐름과 같은 정적 측면과 프로세스 시퀀스와 타이밍 동작과 같은 동적 측면을 기술한다. 또한 소프트웨어 안전 요구사항은 물론 안전과 관련 없는 요구사항을 모두 설계한다. 소프트웨어 요구사항을 할당하고 소프트웨어 개발 복잡도 관리와 소프트웨어 설계 평가 및 개선을 수행한다.

2.3.1. 소프트웨어 설계 명세 수행 절차

○ 소프트웨어 설계 명세 수행 흐름

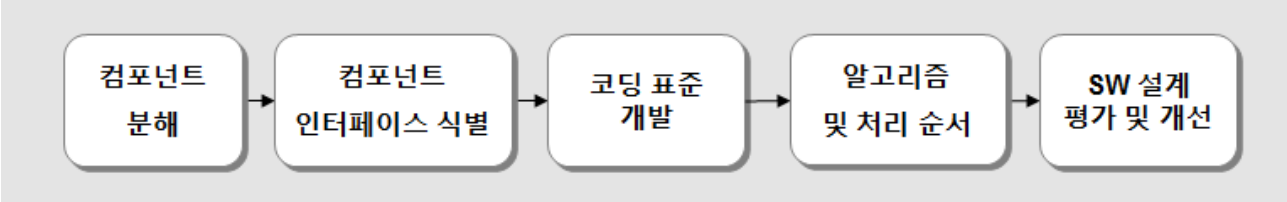


그림 99 소프트웨어 설계 명세 수행 흐름도

○ 소프트웨어 설계 명세 절차

표 117 소프트웨어 설계 명세서 작성 흐름

항 목	설 명
컴포넌트 분해	- 소프트웨어 컴포넌트를 분해하여 소프트웨어 컴포넌트간의 상호 작용을 식별한다. - 요구사항을 컴포넌트에 할당하고 추적성을 확보한다.
컴포넌트 인터페이스 식별	- 소프트웨어 컴포넌트와 외부 환경과의 인터페이스를 식별한다. - 소프트웨어 컴포넌트 간의 인터페이스를 식별한다.
코딩 표준	- 코딩 표준을 개발하고 기술한다. - 프로그래밍 언어 오류를 회피하거나 탐지하기 위한 대책을 기술한다. - 소스 코드 문서화 절차를 기술한다. - 코딩 표준의 선택 근거를 제시한다.
알고리즘 및 처리 순서	- 주요 알고리즘을 기술한다. - 데이터 구조를 식별하고 기술한다. - 처리 순서를 기술한다. - 오류 보고 메커니즘을 정의한다.
소프트웨어 설계 평가 및 개선	- 추상화, 모듈화 및 복잡성 제어 특색을 평가한다. - 기능성, 컴포넌트 간의 정보 흐름 용이성을 평가한다. - 처리 순서 및 시간 관련 정보의 명확성을 평가한다. - 동시성 및 데이터 구조 및 속성이 명확한지 평가한다. - 소프트웨어 유지보수성이 확보 되었는지 평가한다.

2.3.2. 소프트웨어 설계 명세 지침

○ ‘소프트웨어 요구사항 명세서’, ‘소프트웨어 아키텍처 명세서’ 및 ‘소프트웨어 인터페이스 명세서’를 기반으로 소프트웨어 설계 명세를 작성한다. (7.3.4.20)

※ 소프트웨어 설계 주요 품질 속성 예시

표 118 소프트웨어 설계 품질 속성 (예시)

항 목	설 명
검증용이성	- 소프트웨어 설계가 얼마나 검증하기 용이한지를 나타냄 ・정적 검증 : 설계 리뷰, 설계 인스펙션 ・동적 검증 : 테스팅
테스트용이성	- 소프트웨어 설계가 얼마나 테스트 용이한지를 나타냄 - 적은 테스트 케이스로 커버될 수 있는 단순한 설계
유지보수성	- 소프트웨어 설계가 얼마나 수정이 용이한지를 나타냄 - 특정 변경 적용을 위해 수정해야 하는 범위가 작음 - 변경에 따른 설계 수정의 영향이 작음 - 설계 수정 반영의 부작용이 없음
적합성	- 소프트웨어 설계의 요구사항 준수를 나타냄 - 제공되는 기능들이 요구사항들을 모두 커버 - 결과의 정확도가 요구되는 오차 내에 포함

○ 입력 문서는 최종 완성본이 아니더라도, 설계가 시작되기 전에 이용 가능해야 한다. (7.3.4.21)

○ 컴포넌트 분해하여 소프트웨어 설계를 작성하고, 각 컴포넌트는 ‘소프트웨어 컴포넌트 명세서’ 및 ‘소프트웨어 컴포넌트 테스트 명세서’로 작성된다. (7.3.4.22)

표 119 소프트웨어 컴포넌트 분해 기법 (예시)

항 목	설 명
모 듈 화	- 단일의 잘 정의된 작업 또는 기능을 수행하도록 컴포넌트 분해
은 닉 화	- 컴포넌트의 내부는 다른 컴포넌트로부터 보이지 않게 감춤 - 불필요한 정보는 감추고, 컴포넌트간의 인터페이스를 명확화 - 컴포넌트 변경 시 다른 컴포넌트의 변경을 최소화할 수 있음
캡 슐 화	- 컴포넌트의 데이터와 행위를 하나로 묶고, 구현을 외부에 감춤 - 메시지 전달을 통해 간접적으로 데이터에 접근
변수개수 제한	- 컴포넌트 인터페이스의 매개변수 개수 제한 - 매개변수 개수를 5 이하로 제한
전체 인터페이스 정의	- 컴포넌트간의 인터페이스를 완전하게 정의
단일 진입 단일 종료	- 가독성을 높이고 구현 실수 가능성을 줄일 수 있음 - 프로그램 분석과 테스트를 보다 효과적으로 수행 가능

※ 소프트웨어 컴포넌트 분해 주요 방안 예시

○ 소프트웨어 설계 명세에 다음의 사항을 포함해야 한다. (7.3.4.23)

- 소프트웨어 아키텍처와 그 안전 무결성 등급에 대한 소프트웨어 컴포넌트의 추적성
- 소프트웨어 컴포넌트와 외부 환경과의 인터페이스
- 소프트웨어 컴포넌트 간의 인터페이스
- 데이터 구조
- 컴포넌트에 대한 요구사항 할당 및 추적
- 주요 알고리즘과 처리 순서
- 오류 보고 메커니즘

○ [표 207]의 기법과 대책을 소프트웨어 안전 무결성 등급에 따라 선택 적용하여 소프트웨어 설계 명세를 작성한다. (7.3.4.24)

※ 소프트웨어 설계 모델링 주요 방안 예시

표 120 소프트웨어 설계 모델링 주요 기법 및 대책 (예시)

항 목	설 명
상태 전이 다이어그램	- 입력의 변화에 따라 출력이 변화해 가는 상황을 나타냄 - 소프트웨어의 동적 특징 - 컴포넌트 내부 동작이나 컴포넌트 간 상호작용
시퀀스 다이어그램	- 프로세스 또는 컴포넌트가 다른 프로세스와 어떻게 그리고 어떤 순서로 동작하는지 보여줌

※ 소프트웨어 설계 주요 방안 예시

표 121 소프트웨어 설계 주요 기법 및 대책 (예시)

항 목	설 명
구조적 방법론	- 시스템을 기능에 따라 분할하여 개발하고 통합 - 프로세스 중심의 하향식 방법론
분석 가능한 프로그램	- 프로그램 분석이 쉽도록 프로그램을 설계 - 정적 분석 기법으로 분석하기 쉬운 프로그램
엄격한 형식의 프로그래밍 언어	- 컴파일러에서 고 수준의 검사를 허용하는 언어
구조적 프로그래밍	- 프로그램 작성에 있어서 프로그램 논리 흐름을 TOP DOWN 방식으로 구조화를 하여 설계

항 목	설 명
절차적 프로그래밍	- 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍

○ 코딩 표준을 개발하고 명시한다. (7.3.4.25)

- [표 226] 기법 및 대책에 정의 된 바와 같은 모범 프로그래밍 사례
- 검증 과정에서는 탐지 할 수 없는 프로그래밍 언어 적용 중에 발생할 수 있는 오류를 회피하거나 탐지하기 위한 수단. 이러한 실패는 프로그래밍 언어의 모든 특색에 대한 분석을 통해 도출
- 소스 코드 문서화 절차

※ 코딩 표준 개발 주요 방안 예시

표 122 코딩 표준 주요 기법 및 대책 (예시)

항 목	설 명
코딩 표준	- 구현 코드의 레이아웃 일관성을 보장하고 일관된 프로그래밍을 시행하여 오류를 회피 - 해당 언어를 사용할 때 발생할 수 있는 잠재적인 결함을 피하기 위해 주어진 프로그래밍 언어에 대한 규칙 및 제한 사항
코딩 스타일 가이드	- 코딩 시 요구되는 일반적인 스타일 - 개발과정에서의 오류 유입을 방지하고 철도 소프트웨어의 전체적인 품질을 향상시킴 - 철도 차량 제어의 안전 기능에 코딩 스타일을 적용하여 신뢰성 있는 소프트웨어를 개발
크기와 복잡도 제한	- 함수, 서브루틴, 메소드의 크기와 복잡도를 제한
점프 사용 제한	- 조건이 없는 점프문을 사용하지 말 것 - 제어 흐름을 복잡하게 하여 다양한 오류가 발생할 수 있음
전역변수 사용 제한	- 가능한 전역 변수를 사용하지 말 것 - 전역 변수는 여로 곳에서 동시에 접근이 가능하여 결함의 주요 원인이 될 수 있음 - 전역 변수가 필요한 경우 접근 관리 수행

○ 코딩 표준의 선택에 대하여 소프트웨어 안전 무결성 등급 요구 범위에서 충분한 근거를 제시한다. (7.3.4.26)

○ 모든 소프트웨어의 개발에 코딩 표준을 사용하고, 소프트웨어 품질 보증 계획에서 참조된다. (7.3.4.27)

○ 선택된 설계 방법은 요구 소프트웨어 안전 무결성 등급에 따라 다음의 특성을

포함해야 한다. (7.3.4.28)

- 추상화, 모듈화 및 복잡성 제어 특색
- 다음에 대한 명확하고 정확한 표현
  - 기능성
  - 컴포넌트 간의 정보 흐름
  - 순차 실행 및 시간 관련 정보
  - 동시성
  - 데이터 구조 및 속성
- 사람의 이해
- 검증 및 확인
- 소프트웨어 유지 보수

○ 5장 가이드 적용 사례 아키텍처 및 설계 명세, [그림 179 차상 MMI 시스템의 소프트웨어 컴포넌트 및 인터페이스 (예시)] 참조

2.3.3. 소프트웨어 설계 명세서 템플릿

1. 개요

1.1 범위

- ‘소프트웨어 설계 명세서’에는 소프트웨어 컴포넌트 명세와 테스트에 필요한 소프트웨어 설계를 기술한다.

1.2 목적

- ‘소프트웨어 설계 명세서’를 작성하는 설계자와 참조하는 통합자, 검증자의 목적을 작성한다. ‘소프트웨어 설계 명세서‘를 기준으로 소프트웨어 컴포넌트 명세, 컴포넌트 테스트 명세 및 구현을 수행하고 검증된 소프트웨어 컴포넌트의 통합 및 통합 검증을 수행한다.

1.3 시스템 개요

- 소프트웨어 설계 대상 시스템의 목적에 대해 간략한 설명을 기술한다.

1.4 형상 식별

- 소프트웨어 설계 항목의 식별을 위한 항목 식별자를 설명한다.

1.5 관련 표준

- ‘소프트웨어 설계 명세서’와 관련된 표준을 설명한다.

2. 설계 배경

2.1 설계 정의

- ‘소프트웨어 설계 명세서’를 참조하는 개발자, 통합자, 테스터, 검증자에게 설계의 배경과 의사 결정 내역 및 설계 안전성에 대해 설명한다.

2.2 가정 사항

- 소프트웨어 설계 명세 시에 수립한 가정 사항에 대하여 기술한다.

3. 컴포넌트 분해

3.1 정의

- 컴포넌트의 이름, 책임, 기능 등의 상세 정보를 기술한다.

3.2 외부 환경 인터페이스

- 소프트웨어 컴포넌트와 외부 환경과의 인터페이스를 기술한다.

3.3 컴포넌트 간 인터페이스

- 소프트웨어 컴포넌트 간의 인터페이스 관계와 속성을 기술한다.

3.4 데이터 구조

- 컴포넌트의 데이터 처리 단위와 저장 공간 및 데이터 구조를 기술한다.

3.5 주요 알고리즘 및 처리 순서

- 주요 알고리즘과 처리 순서를 기술한다.

3.6 오류 보고 메커니즘

- 오류 보고 메커니즘을 기술한다.

4. 요구사항 추적 매트릭스

- 소프트웨어 요구사항 및 아키텍처 명세와 소프트웨어 설계 명세 간의 추적성을 기술한다.

그림 100 소프트웨어 설계 명세서 템플릿 (예시)

2.3.4. 소프트웨어 설계 명세서 체크리스트

표 123 소프트웨어 설계 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
문서 구성	채택한 템플릿이나 표준을 잘 준수하는가?
	각 설계 뷰에서 뷰의 정의(요소, 관계, 속성)를 설명하고 있는가?
	문서에서 사용하는 정의가 명확한가?
	문서에 근거, 제약사항, 배경, 설계의 후보에 관한 정보가 있는가?
	소프트웨어 설계 명세의 요소와 관련된 소프트웨어 요구사항을 추적할 수 있는가?
	문서에 불필요한 정보가 있지 않은가?
정확성/완전성	모든 소프트웨어 컴포넌트는 분해되었는가?
	컴포넌트간의 기능과 데이터 흐름은 정의되었는가?
	주요한 알고리즘은 정의되었는가?
	모든 운영상의 트랜잭션에 대한 위협은 정의되었는가?
	모든 이벤트와 타이밍 순서는 정의되었는가?
	모든 입출력 기술이 정의되었는가?
	운영시스템과 예정된 컴포넌트 간의 모든 인터페이스는 정의되었는가?
	데이터를 공유하기 위한 동시 접근이 올바로 처리되었는가?
	설계가 모든 필요한 통제적인 측면들을 표현하고 있는가?
	설계가 비정상/정상 상태를 표현하고 있는가?
	설계가 테스트될 수 있는 시점이 식별되어 있는가?

구 분	점검 사항
	설계에 요구되는 모든 필요한 정보를 설계가 담고 있는가?
	기존 소프트웨어 산출물이 식별되었는가?
	‘소프트웨어 요구사항 명세서‘ 상의 모든 소프트웨어 요구사항을 ’ 소프트웨어 설계 명세서‘에 나타내고 있는가?
	소프트웨어에 대한 용량 및 시간 추정 예측이 소프트웨어 요구사항과 부합되는가?
일관성	설계가 내부적으로 일관성이 있는가?
	소프트웨어 인터페이스가 일관성이 있는가?
	데이터 정의와 처리에 대해 일관성이 있는가?
	설계가 다른 문서와 일관성이 있는가?
실현 가능성	설계가 현재 프로젝트의 제약사항 아래에서 정해진 비용과 일정 안에서 수행될 수 있는가?
	설계는 알려진 또는 증명된 것을 기초로 했는가?
	설계는 선택된 프로그래밍 언어에 의해 선택된 플랫폼에서 구현되는가?
	설계는 필요한 도구에 의해 지원 받는가?
	기술적 위험 들을 포함한 모든 설계 제약사항들은 식별되고 표현되었는가?
표준 준수성	‘소프트웨어 설계 명세서‘를 활용하여 구현할 수 있는가?
	설계는 이해가 용이하고, 다양하게 해석되는 경우는 없는가?
	설계가 소프트웨어 개발 계획서 상에 기술된 방법론 및 지원 도구에 따라 구현되는가?
	모든 설계 정보가 적절한 분류에 의해 표시되고 있는가?
	문서는 요구되는 표준에 의해 기술되었는가?
(기타)	기타항목

2.4. 소프트웨어 통합 테스트 명세

소프트웨어 아키텍처 및 설계에 대한 통합 테스트 명세를 작성한다. 소프트웨어 통합 테스트 전략을 수립하고, 테스트 케이스 추출 방법 선정 및 수행, 추적 매트릭스 작성, 결과 판정 기준 상세화, 테스트 필요 환경 식별, 문제 발견 및 해결 절차와 회귀 테스트 수행 기준을 작성한다.

2.4.1. 소프트웨어 통합 테스트 명세 수행 절차

○ 소프트웨어 통합 테스트 명세 수행 흐름

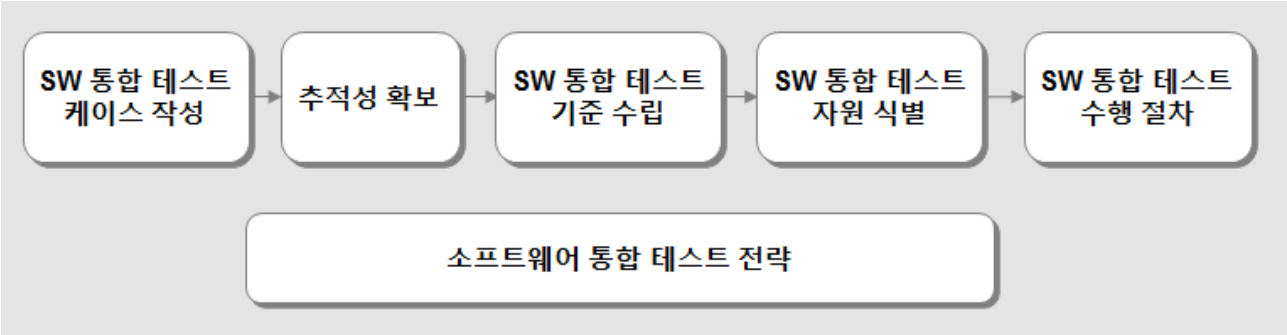


그림 101 소프트웨어 통합 테스트 명세 수행 흐름도

○ 소프트웨어 통합 테스트 명세 절차

표 124 소프트웨어 통합 명세서 작성 절차

항 목	설 명
소프트웨어 통합 테스트 케이스 작성	- 요구 입력 데이터, 데이터 순서, 데이터 값을 테스트 케이스 기반으로 테스트 케이스를 작성한다. - 예상 출력 데이터, 데이터 순서, 데이터 값을 테스트 케이스 기반으로 테스트 케이스를 작성한다.
추적성 확보	- 소프트웨어 요구사항과 소프트웨어 아키텍처 및 설계에 대한 추적성을 확보한다.
소프트웨어 통합 테스트 기준 수립	- 소프트웨어 검증 계획을 기반으로 소프트웨어 통합 테스트 기준을 상세화한다. - 각 테스트 별로 통과 기준을 수립한다. - 소프트웨어 통합 테스트 방법을 선정하고 상세 계획을 수립한다.
소프트웨어 통합 테스트 자원 식별	- 소프트웨어 통합 테스트 환경을 식별한다. - 테스트 시 측정해야 할 지표 및 장비를 식별한다.
소프트웨어 통합 테스트 수행 절차	- 소프트웨어 통합 테스트 관련 의존성을 분석한다. - 소프트웨어 통합 테스트 수행 절차를 수립한다.

항 목	설 명
	- 문제 발견 및 해결 절차를 수립한다. - 회귀 테스트 수행 기준 및 절차를 수립한다.
소프트웨어 통합 테스트 전략	- 컴포넌트 테스트 결과를 소프트웨어 통합 테스트에 재사용할 경우 이를 명시한다. - 각 소프트웨어 컴포넌트의 인터페이스를 제공받는 컴포넌트와 함께 실행하여 특정 인터페이스를 제공함을 확인한다. - 규격을 벗어난 인터페이스 입력에 소프트웨어가 적절한 방식으로 동작함을 확인한다.

2.4.2. 소프트웨어 통합 테스트 명세 지침

○ 소프트웨어 요구사항 명세, 소프트웨어 아키텍처 명세, 소프트웨어 설계 명세 및 소프트웨어 인터페이스 명세에 기초하여, 소프트웨어 통합 테스트 명세를 작성한다. (7.3.4.29)

※ 소프트웨어 통합 테스트 주요 방안 예시

표 125 소프트웨어 통합 테스트 기법 (예시)

항 목	설 명
요구사항 기반 기능 테스트	- 요구 소프트웨어 기능 구현 일치 확인
자원 사용 테스트	- 자원 사용의 적절성 확인 - 프로세서 사용량, 메모리 사용량, 네트워크 사용량
인터페이스 테스트	- 컴포넌트 인터페이스 구현 정확성 확인
결합 주입 테스트	- 오류에 대한 소프트웨어 대비 확인

○ 소프트웨어 테스트 명세 일반 지침에 따라서 소프트웨어 통합 테스트 명세를 작성한다. (7.3.4.30)

- 소프트웨어 테스트 명세 일반 지침
  - · 테스트 목적
  - · 테스트 케이스, 테스트 데이터 및 예상 결과 값
  - · 수행할 테스트 유형
  - · 테스트 환경, 도구, 구성 및 프로그램
  - · 테스트 완료 판정 기준

· 테스트 커버리지 기준 및 수준
· 역할과 책임
· 요구사항의 테스트 명세 커버리지
· 테스트 장비 선택 및 사용

○ 소프트웨어 통합 테스트 명세는 다음의 사항을 다루어야 한다. (7.3.4.31)

- 각 소프트웨어 컴포넌트를 인터페이스를 제공받는 컴포넌트와 함께 실행하여 특정 인터페이스를 제공함을 확인
- 규격을 벗어난 인터페이스 입력에 소프트웨어가 적절한 방식으로 동작함을 확인
- 요구 입력 데이터, 데이터 순서, 데이터 값을 기반으로 테스트 케이스 작성
- 예상 출력 데이터, 데이터 순서, 데이터 값을 기반으로 테스트 케이스 작성
- 컴포넌트 테스트 결과가 소프트웨어 통합 테스트에 재사용되었을 경우 이를 명시

※ 소프트웨어 통합 테스트 케이스 설계 주요 방안 예시

표 126 소프트웨어 통합 테스트 케이스 설계 기법 (예시)

항 목	설 명
요구사항 분석	- 소프트웨어 통합 테스트 명세의 입력 문서를 분석하여 테스트 케이스 설계
동치 클래스 분석	- 입력 구간의 최소, 최대값을 기준으로 유효한 값과 무효한 값을 선택하여 테스트 케이스 설계 - 안전 요구사항에 명시적으로 기술된 사항 및 일반 고려 사항 반영
경계값 분석	- 각 구간의 경계값을 선택하여 테스트 케이스 설계
유즈케이스 분석	- 엑터의 시스템 사용에 따른 동작 시나리오를 분석
고장 사례 분석	- 과거 발생 결함이나 축적된 안전 로그를 이용하여 결합을 검증할 테스트 케이스 설계
인터페이스 분석	- 유효한 인터페이스 입력값을 기준으로 테스트 케이스 설계

○ [표 208] 검증 및 시험 단계 기법 및 대책을 소프트웨어 안전 무결성 등급에 따라 선택 적용한다. (7.3.4.32)

※ 소프트웨어 통합 테스트 주요 기법 및 대책 예시

표 127 소프트웨어 통합 테스트 기법 및 대책 (예시)

항 목	설 명
동적 분석 및 테스트	- 경계값 테스트 ・각 구간의 경계값을 입력하여 테스트 수행 - 오류 추측 테스트 ・발생 가능한 오류를 예측하여 검증할 테스트 케이스 설계 - 결함 주입 테스트 ・특별한 테스트 인터페이스를 통해 결함을 발생시켜 테스트 - 성능 모델링 - 동치 클래스와 입력 분할 테스트 - 구조 기반 테스트
기능 / 블랙박스 테스트	- 원인 결과 다이어그램 테스트 - 프로토타입 / 애니메이션 - 경계값 분석 - 동치 클래스와 입력 분할 테스트 - 프로세스 시뮬레이션
성능 테스트	- 과부하 / 스트레스 테스트 ・각 인자별로 과부하를 발생시켜 테스트 수행 - 응답 시간과 메모리 제약 ・결함 허용 가능 시간 간격과 메모리 사용량 측정 - 성능 요구사항

2.4.3. 소프트웨어 통합 테스트 명세서 템플릿

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 통합 테스트 전략

2.1 테스트 케이스 추출 방법

- 소프트웨어 통합 테스트 케이스 추출 방법을 기술한다.

2.2 소프트웨어 통합 테스트 수행 시점

- 소프트웨어 통합 테스트 수행 시점을 기술한다.

2.3 테스트 결과 재사용 내역

- 컴포넌트 테스트 결과 재사용 내역을 상세하게 기술한다.

3. 테스트 케이스

3.1 식별자

- 테스트 케이스를 구분한다.

3.2 설명

- 테스트 케이스를 설명한다.

3.3 입력값

- 테스트 입력값을 기술한다.

3.4 예상 결과

- 테스트 예상 결과를 기술한다.

3.5 출력값

- 테스트 출력값으로 테스트 수행 후 결과 보고서 작성 시에 기술된다.

3.6 판정 결과

- 테스트 판정 결과는 테스트 수행 후 결과 보고서 작성 시에 기술된다.

4. 추적성 확보

- 요구사항 / 아키텍처 / 설계 / 테스트 케이스 간의 추적 메트릭스를 작성한다.

5. 소프트웨어 통합 테스트 기준

5.1 소프트웨어 통합 테스트 방법

- 소프트웨어 통합 테스트 방법을 기술한다.

5.2 통과 기준

- 각 테스트 별로 통과 기준을 기술한다.

6. 소프트웨어 통합 테스트 자원

6.1 소프트웨어 통합 테스트 수행 환경

- 소프트웨어 통합 테스트 수행 환경을 기술한다.

6.2 소프트웨어 통합 테스트 수행 자원

- 소프트웨어 통합 테스트 측정 지표 및 장비에 대하여 상세히 기술한다.

7. 소프트웨어 통합 테스트 절차

7.1 소프트웨어 통합 테스트 수행 절차

- 소프트웨어 통합 테스트 수행 절차를 기술한다.

7.2 문제 처리 절차

- 문제 발견 및 해결 절차를 기술한다.

- 회귀 테스트 수행 기준 및 절차를 기술한다.

A. 부록

- 추가적인 내용을 기술한다.

그림 102 소프트웨어 통합 테스트 명세서 템플릿 (예시)

2.4.4. 소프트웨어 통합 테스트 명세서 체크리스트

표 128 소프트웨어 통합 테스트 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	테스트 목적을 기술하고 있는가?
	테스트 케이스, 테스트 데이터 및 예상 결과값이 기술되어 있는가?
	수행할 테스트 유형을 기술하고 있는가?
	테스트 환경, 도구, 구성 및 프로그램을 기술하고 있는가?
	테스트 완료 판정 기준이 기술되어 있는가?
	테스트 커버리지 기준 및 수준이 기술되어 있는가?
	테스트 관련 역할 및 책임이 기술되어 있는가?
	요구사항의 테스트 명세 커버리지가 식별되어 기술되었는가?
	테스트 장비 선택 및 사용 내역이 기술되어 있는가?
품질 요구사항	소프트웨어 요구사항과 소프트웨어 아키텍처 및 설계에 대한 추적성을 확보하고 있는가?

구 분	점검 사항
	컴포넌트 테스트 결과 재사용 사항을 기술하고 있는가?
	동적 분석 및 테스트 기법이 적절한가?
	기능 테스트 기법이 적절한가?
	블랙박스 테스트 기법이 적절한가?
	적절한 성능 테스트 방안이 수립되었는가?
(기타)	기타항목

2.5. 소프트웨어 / 하드웨어 통합 테스트 명세

시스템 설계와 소프트웨어 설계에 대한 통합 테스트 명세를 작성한다. 소프트웨어 / 하드웨어 통합 테스트 전략을 수립하고, 테스트 케이스 추출 방법 선정 및 수행, 추적 메트릭스 작성, 결과 판정 기준 상세화, 테스트 수행 리소스 식별, 문제 발견 및 해결 절차와 회귀 테스트 수행 기준을 작성한다.

2.5.1. 소프트웨어 / 하드웨어 통합 테스트 명세 수행 절차

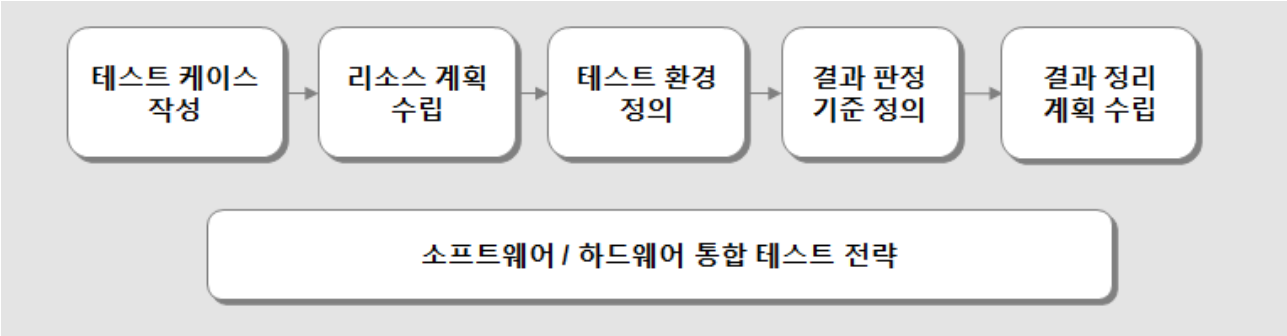


그림 103 소프트웨어 / 하드웨어 통합 테스트 명세 수행 흐름도

표 129 소프트웨어/하드웨어 통합 테스트 명세서 작성 절차

항 목	설 명
테스트 케이스 작성	- 요구 입력 데이터, 데이터 순서, 데이터 값을 테스트 케이스 기반으로 테스트 케이스를 작성한다. - 예상 출력 데이터, 데이터 순서, 데이터 값을 테스트 케이스 기반으로 테스트 케이스를 작성한다.
리소스 계획 수립	- 하드웨어 특성에 따라 세부적인 접근 방법이 달라 질 수 있으므로 하드웨어 특성을 고려한 테스트 케이스를 기술한다. - 입력값과 예상되는 출력값을 기술한다.
테스트 환경 정의	- 공급 업체가 자신의 설비에서 자체적으로 수행할 수 있는 활동과 고객 사이트에서 수행해야 하는 활동을 구분하여 작성한다. - 도구, 지원 소프트웨어, 설정을 포함한 테스트 환경을 정의한다.
결과 판정 기준 정의	- 소프트웨어 검증 계획을 기반으로 소프트웨어 / 하드웨어 통합 테스트 기준을 상세화한다. - 테스트 완료를 판정할 테스트 기준을 정의한다.
결과 정리 계획 수립	- 소프트웨어 / 하드웨어 통합 테스트 수행 절차를 수립한다. - 문제 발견 및 해결 절차를 수립한다. - 회귀 테스트 수행 기준 및 절차를 수립한다.
통합 테스트 전략	- 컴포넌트 테스트와 소프트웨어 통합 테스트 결과를 소프트웨어/하드웨

항 목	설 명
	어 통합 테스트에 재사용할 경우 이를 명시한다. - 하드웨어에서 지정된 하드웨어 인터페이스를 통해 소프트웨어가 적절한 방법으로 실행되는지 확인한다. - 하드웨어에서 지정된 하드웨어 인터페이스를 통해 소프트웨어가 적절한 방법으로 실행했는지 확인한다. - 소프트웨어가 하드웨어 오류를 요구에 맞도록 처리할 수 있는지 확인한다. - 요구 타이밍과 성능에 대하여 시연한다.

2.5.2. 소프트웨어 / 하드웨어 통합 테스트 명세 지침

○ 시스템 설계, 소프트웨어 요구사항 명세, 소프트웨어 아키텍처 명세 및 소프트웨어 설계 명세에 기초하여 ‘소프트웨어/하드웨어 통합 테스트 명세서’를 작성한다. (7.3.4.33)

○ 통합 테스트의 적절한 방향설정과 특정 설계 또는 기타 통합 요구가 적절하게 제공 될 수 있도록 개발 생명주기 초기에 ‘소프트웨어/하드웨어 통합 테스트 명세서’를 생성한다. 시스템의 크기에 따라 ‘소프트웨어/하드웨어 통합 테스트 명세서’는 개발 과정에서 여러 개의 하위 문서로 세분화 될 수 있으며, 하드웨어 및 소프트웨어 설계가 진행되고 상세한 통합 요구가 명확해짐에 따라서 점차적으로 수정할 수 있다. (7.3.4.34)

○ 공급 업체가 자신의 설비에서 자체적으로 수행할 수 있는 활동과 고객 사이트에서 수행해야 하는 활동을 구분하여 작성한다. (7.3.4.35)

○ 소프트웨어/하드웨어 통합 테스트 명세는 다음의 사항을 보장해야 한다. (7.3.4.36)

- 하드웨어에서 지정된 하드웨어 인터페이스를 통해 소프트웨어가 적절한 방법으로 실행됨
- 소프트웨어는 하드웨어 오류를 요구에 맞도록 처리할 수 있음
- 요구 타이밍과 성능에 대한 시연
- 요구 입력 데이터, 데이터 순서, 데이터 값을 기반으로 테스트 케이스 작성
- 예상 출력 데이터, 데이터 순서, 데이터 값을 기반으로 테스트 케이스 작성
- 컴포넌트 테스트와 소프트웨어 통합 테스트 결과를 소프트웨어/하드웨어 통합 테스트에 재사용할 경우 이를 명시

※하드웨어 / 소프트웨어 통합 테스트 케이스 추출 참고 사항 예시

・하드웨어 장치의 동작모드 관련 매개변수 설정

・소프트웨어 독립성을 지원하는 하드웨어 특성 고려

- 하드웨어 자원 공유 및 사용 검증
- 하드웨어 장치 엑세스 메커니즘 검증
- 각 서비스의 타이밍 제약 사항 검증
- 하드웨어 진단, 소프트웨어 진단 기능 검증
- 하드웨어와 소프트웨어 간 상호 안전 메커니즘 고려

○ 소프트웨어/하드웨어 통합 테스트 명세에 다음의 사항을 포함해야 한다. (7.3.4.37)

- 테스트 케이스와 테스트 데이터
- 수행할 테스트 유형
- 도구, 지원 소프트웨어, 설정을 포함한 테스트 환경
- 테스트 완료 판정 기준

○ 소프트웨어 테스트 명세 일반 지침에 따라서 소프트웨어/하드웨어 통합 테스트 명세를 작성한다. (7.3.4.38)

- 소프트웨어 테스트 명세 일반 지침은 다음 사항을 기술
  - 테스트 목적
  - 테스트 케이스, 테스트 데이터 및 예상 결과 값
  - 수행할 테스트 유형
  - 테스트 환경, 도구, 구성 및 프로그램
  - 테스트 완료 판정 기준
  - 테스트 커버리지 기준 및 수준
  - 역할과 책임
  - 요구사항의 테스트 명세 커버리지
  - 테스트 장비 선택 및 사용

○ [표 208] 검증 및 시험 단계 기법 및 대책을 소프트웨어 안전 무결성 등급에 따라 선택 적용한다. (7.3.4.39)

2.5.3. 소프트웨어 / 하드웨어 통합 테스트 명세서 템플릿

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 / 하드웨어 통합 테스트 전략

2.1 테스트 케이스 추출 방법

- 소프트웨어 통합 테스트 케이스 추출 방법을 기술한다.

2.2 소프트웨어 / 하드웨어 통합 테스트 수행 시점

- 소프트웨어 / 하드웨어 통합 테스트 수행 시점을 기술한다.

2.3 테스트 결과 재사용 내역

- 컴포넌트 테스트와 소프트웨어 통합 테스트 결과 재사용 내역을 상세하게 기술한다.

3. 테스트 케이스

3.1 식별자

- 테스트 케이스를 구분한다.

3.2 설명

- 테스트 케이스를 설명한다.

3.3 입력값

- 테스트 입력값을 기술한다.

3.4 예상 결과

- 테스트 예상 결과를 기술한다.

3.5 출력값

- 테스트 출력값으로 테스트 수행 후 결과 보고서 작성 시에 기술된다.

3.6 판정 결과

- 테스트 판정 결과는 테스트 수행 후 결과 보고서 작성 시에 기술된다.

4. 리소스 계획

4.1 인원 계획

- 담당자 이름 / 역할 / 설명 / 일정을 기술한다.

4.2 도구 계획

- 테스트 측정 지표 및 장비에 대하여 상세히 기술한다.

5. 테스트 환경 정의

5.1 소프트웨어 / 하드웨어 통합 테스트 환경

- 공급 업체 설비에서의 테스트 항목을 기술한다.
- 고객 설비에서의 테스트 항목을 기술한다.

5.2 소프트웨어 / 하드웨어 통합 테스트 자원

- 도구, 지원 소프트웨어, 설정을 포함한 테스트 환경을 기술한다.

6. 소프트웨어 / 하드웨어 통합 테스트 결과 판정 기준

6.1 소프트웨어 / 하드웨어 통합 테스트 기준

- 소프트웨어 / 하드웨어 통합 테스트 기준을 상세히 기술한다.

6.2 테스트 완료 판정 기준

- 문제 발견 및 해결 절차를 기술한다.
- 회귀 테스트 수행 기준 및 절차를 기술한다.

7. 통합 테스트 계획

7.1 소프트웨어 / 하드웨어 통합 테스트 수행 절차

- 소프트웨어 통합 테스트 수행 절차를 기술한다.

7.2 문제 처리 절차

- 문제 발견 및 해결 절차를 기술한다.
- 회귀 테스트 수행 기준 및 절차를 기술한다.

A. 부록

- 추가적인 내용을 기술한다.

그림 104 소프트웨어 / 하드웨어 통합 테스트 명세서 템플릿 (예시)

2.5.4. 소프트웨어 / 하드웨어 통합 테스트 명세서 체크리스트

표 130 소프트웨어 / 하드웨어 통합 테스트 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	테스트 목적을 기술하고 있는가?
	테스트 케이스, 테스트 데이터 및 예상 결과값이 기술되어 있는가?
	수행할 테스트 유형을 기술하고 있는가?
	테스트 환경, 도구, 구성 및 프로그램을 기술하고 있는가?
	테스트 완료 판정 기준이 기술되어 있는가?
	테스트 커버리지 기준 및 수준이 기술되어 있는가?
	테스트 관련 역할 및 책임이 기술되어 있는가?
	요구사항의 테스트 명세 커버리지가 식별되어 기술되었는가?
	테스트 장비 선택 및 사용 내역이 기술되어 있는가?
품질 요구사항	시스템 설계, 소프트웨어 요구사항 명세, 소프트웨어 아키텍처 및 설계에 대한 추적성을 확보하고 있는가?
	컴포넌트 테스트와 소프트웨어 통합 테스트 결과 재사용 사항을 기술하고 있는가?
	공급 업체 설비 테스트와 고객사 설비 테스트가 식별되었는가?
	하드웨어 인터페이스를 통한 소프트웨어 실행 테스트가 명시되었는가?
	하드웨어 오류 처리에 대한 소프트웨어 테스트가 식별되었는가?
	요구 타이밍과 성능에 대한 테스트 방안이 수립되었는가?
	하드웨어 자원 공유 및 사용 검증이 테스트 가능한가?
(기타)	기타항목

2.6. 소프트웨어 아키텍처 및 설계 검증

소프트웨어 아키텍처 및 설계 가이드라인을 사용하여 소프트웨어 아키텍처 및 설계를 검증한다. 소프트웨어 아키텍처, 소프트웨어 인터페이스, 소프트웨어 설계, 소프트웨어 통합 테스트 및 소프트웨어/하드웨어 통합 테스트 명세의 일관성, 적합성, 가독성 및 추적성에 대한 일반 요구사항과 각 명세별 특정 요구사항의 충족 여부를 검증한다.

2.6.1. 소프트웨어 아키텍처 및 설계 검증 수행 절차

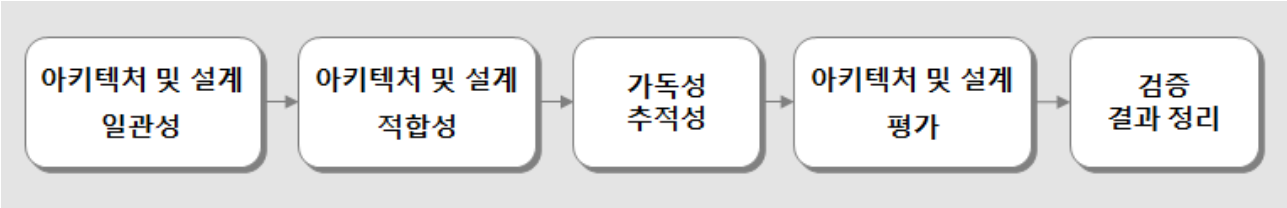


그림 105 소프트웨어 아키텍처 및 설계 검증 수행 흐름도

표 131 소프트웨어 아키텍처 및 설계 검증 보고서 작성 절차

항 목	설 명
아키텍처 및 설계 일관성	- 소프트웨어 요구사항 명세 충족에 대한 일관성, 완전성을 검토한다. - 소프트웨어 아키텍처, 인터페이스 및 설계 명세의 내부적 일관성을 검토한다.
아키텍처 및 설계 적합성	- 소프트웨어 아키텍처, 인터페이스 및 설계 명세의 적합성을 검토한다. - 하드웨어 및 소프트웨어 제약 조건을 고려한 소프트웨어 아키텍처 명세 및 소프트웨어 설계 명세의 적합성을 검토한다.
아키텍처 및 설계 가독성, 추적성	- 소프트웨어 아키텍처 명세의 가독성 및 추적성을 검토한다. - 소프트웨어 인터페이스 명세의 가독성 및 추적성을 검토한다. - 소프트웨어 설계 명세의 가독성 및 추적성을 검토한다. - 소프트웨어 통합 테스트 명세의 가독성 및 추적성을 검토한다. - 소프트웨어 / 하드웨어 통합 테스트 명세의 가독성 및 추적성을 검토한다.
아키텍처 및 설계 평가	- 규격 불일치 아이템을 평가한다. - 불완전하게 적용된 컴포넌트, 데이터 구조 및 알고리즘을 평가한다. - 발견 오류 또는 결함을 평가한다.
검증 결과 정리 및 판정	- 가정 사항을 기술한다. - 검증 결과를 요약한다. - 소프트웨어 검증 계획의 충족 또는 부족 여부를 판정한다.

2.6.2. 소프트웨어 아키텍처 및 설계 검증 보고서 작성 지침

○ 소프트웨어 요구사항 명세, 소프트웨어 아키텍처 명세, 소프트웨어 설계 명세, 소프트웨어 통합 테스트 명세 및 소프트웨어/하드웨어 통합 테스트 명세에 기초하여 소프트웨어 아키텍처 및 설계 검증 보고서를 작성한다. (7.3.4.40)

※ 소프트웨어 아키텍처 및 설계 검증을 위한 정적분석 방안 예시

표 132 소프트웨어 아키텍처 및 설계 검증 정적 분석 기법 및 대책 (예시)

항 목	설 명
경계값 분석	- 매개변수의 한계 또는 경계에서 발생하는 소프트웨어 오류 검증
체크리스트	- 검증 대상 전반에 대한 평가 질의 및 달성 정도 확인
제어 흐름 분석	- 프로그램 실행 각 구문, 명령, 함수가 호출되는 흐름을 분석 - 결함이 있거나 잠재적으로 정확하지 않은 프로그램 구조를 탐색
데이터 흐름 분석	- 요구사항의 각 항목들을 분석하여 생성되고 활용되는 데이터를 정의 - 데이터에 대한 활동 및 흐름을 식별 - 변수에 대한 값의 사용과 갱신 시점에 따른 잠재적 결함 탐색
오류 추측	- 발생 가능한 오류를 예측하여 설계 대책 반영 여부 검증
워크스루 / 디자인 리뷰	- 설계 산출물에 대한 공식 / 비공식 동료 검토

○ 소프트웨어 검증 보고서 일반 지침에 따라서 소프트웨어 아키텍처 및 설계 검증 보고서를 작성한다. (7.3.4.41)

- 소프트웨어 검증 보고서 일반 지침은 다음 사항을 기술
  - 검증 아이템의 특성 정보 및 구성, 검증자 이름
  - 규격 불일치 아이템
  - 불완전하게 적용된 컴포넌트, 데이터 구조 및 알고리즘
  - 발견 오류 또는 결함
  - 소프트웨어 검증 계획의 충족 또는 부족 여부
    (검증 보고서에 편차가 있는 경우 편차가 중대한 것인지 아닌지를 기술)
  - 가정 사항 (존재 시)
  - 검증 결과 요약

○ 소프트웨어 아키텍처, 인터페이스, 설계 명세 확정 후 다음의 사항을 보장하여야

한다. (7.3.4.42)

- 소프트웨어 아키텍처, 인터페이스 및 설계 명세의 내부적 일관성
- 일관성 및 완전성과 관련된 소프트웨어 요구사항 명세를 충족하는 소프트웨어 아키텍처, 인터페이스 및 설계 명세의 적합성
- 소프트웨어 아키텍처 명세는 가독성 및 추적성에 대한 일반 요구사항과 2.1.2의 특정 요구사항을 충족
- 소프트웨어 인터페이스 명세는 가독성 및 추적성에 대한 일반 요구사항과 2.2.2의 특정 요구사항을 충족
- 소프트웨어 설계 명세는 가독성 및 추적성에 대한 일반 요구사항과 2.3.2의 특정 요구사항을 충족
- 하드웨어 및 소프트웨어 제약 조건을 고려한 소프트웨어 아키텍처 명세 및 소프트웨어 설계 명세의 적합성

○ 소프트웨어 통합 및 소프트웨어/하드웨어 통합 테스트 명세 확정 후 다음의 사항을 보장하여야 한다.(7.3.4.43)

- 소프트웨어 통합 테스트 명세는 가독성 및 추적성에 대한 일반 요구사항과 2.4.2의 특정 요구사항을 충족
- 소프트웨어/하드웨어 통합 테스트 명세는 가독성 및 추적성에 대한 일반 요구사항과 2.5.2의 특정 요구사항을 충족
- 소프트웨어 가독성 및 추적성 일반 요구사항
  - 고유 참조 번호 및 관계로 각 문서간의 추적성을 제공
  - 용어, 두문자어 또는 약어는 모든 문서에서 동일한 의미
  - 선행 문서의 모든 적용 가능한 조건과 요구사항을 포함하거나 구현
  - 이전 문서와 모순되지 않아야 함
  - 각 항목 또는 개념은 모든 문서에서 동일한 이름 또는 설명으로 참조
  - 생명주기 전 단계에 대한 요구사항 추적성 확인 수단 제공
  - 소프트웨어 안전 무결성 등급에 적합하도록 다음의 추적성 제공
    - 요구사항을 충족시키는 설계 또는 다른 객체에 대한 추적성
    - 설계 객체를 인스턴스화하는 구현 객체에 대한 추적성
    - 요구사항 및 설계 객체의 테스트 및 검증 분석에 대한 추적성
  - 추적성은 형상 관리의 대상

2.6.3. 소프트웨어 아키텍처 및 설계 검증 보고서 템플릿

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 개요

2.1 검증 아이템

- 검증 아이템의 특성 정보 및 구성을 기술한다.

2.2 검증자

- 검증자 이름을 기술한다.

2.3 규칙 불일치 아이템

- 규칙 불일치 아이템을 기술한다.

2.4 불안전 사항

- 불완전하게 적용된 컴포넌트, 데이터 구조 및 알고리즘을 기술한다.

2.5 발견 오류 및 결함

- 발견 오류 및 결함을 기술한다.

3. 결과 판정

3.1 판정 결과

- 소프트웨어 검증 계획의 충족 또는 부족 여부를 기술한다.

3.2 편차 사항

- 검증 보고서에 편차가 있는 경우 편차가 중대한 것인지 아닌지를 기술한다.

4. 가정 사항

가정 사항이 존재할 경우 기술한다.

5. 검증 결과 요약

검증 결과를 요약 기술한다.

A. 부록

- 추가적인 내용을 기술한다.

그림 106 소프트웨어 아키텍처 및 설계 검증 결과 보고서 템플릿 (예시)

2.6.4. 소프트웨어 아키텍처 및 설계 검증 보고서 체크리스트

표 133 소프트웨어 아키텍처 및 설계 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	검증 아이템의 특성 정보 및 구성, 검증자 이름을 기술하고 있는가?
	규격 불일치 아이템을 기술하고 있는가?
	불완전하게 적용된 컴포넌트, 데이터 구조 및 알고리즘을 기술하고 있는가?
	발견 오류 또는 결함을 기술하고 있는가?
	소프트웨어 검증 계획의 충족 또는 부족 여부를 기술하고 있는가?
	가정 사항을 기술하고 있는가?
	검증 결과 요약이 작성 되었는가?
설계 명세 품질 요구사항	소프트웨어 아키텍처, 인터페이스 및 설계 명세의 내부적 일관성
	일관성 및 완전성과 관련된 소프트웨어 요구사항 명세를 충족하는 소프트웨어 아키텍처, 인터페이스 및 설계 명세의 적합성
	소프트웨어 아키텍처 명세의 가독성 및 추적성
	소프트웨어 인터페이스 명세의 가독성 및 추적성
	소프트웨어 설계 명세의 가독성 및 추적성
	하드웨어 및 소프트웨어 제약 조건을 고려한 소프트웨어 아키텍처 명세 및 소프트웨어 설계 명세의 적합성
테스트 명세 품질 요구사항	소프트웨어 통합 테스트 명세의 가독성 및 추적성
	소프트웨어/하드웨어 통합 테스트 명세의 가독성 및 추적성
(기타)	기타항목

제 3 절 소프트웨어 컴포넌트 설계

1. 개요

소프트웨어 컴포넌트의 설계는 ‘소프트웨어 설계 명세서’를 준수하고 실행 가능한 컴포넌트를 설계한다.

1.1. 목표

- ‘소프트웨어 설계 명세서’를 준수하는 ‘소프트웨어 컴포넌트 설계 명세서’를 기술한다.
- ‘소프트웨어 컴포넌트 설계 명세서’를 테스트하는 ‘ 소프트웨어 컴포넌트 테스트 명세서’를 기술한다.
- ‘소프트웨어 컴포넌트 설계 명세서’를 검증하는 ‘소프트웨어 컴포넌트 설계 검증 보고서’를 기술한다.

1.2. 범위

소프트웨어 개발 생명주기에서 IEC 62279 7.4절에 해당하는 소프트웨어 컴포넌트 설계 단계에 대해 설명한다.

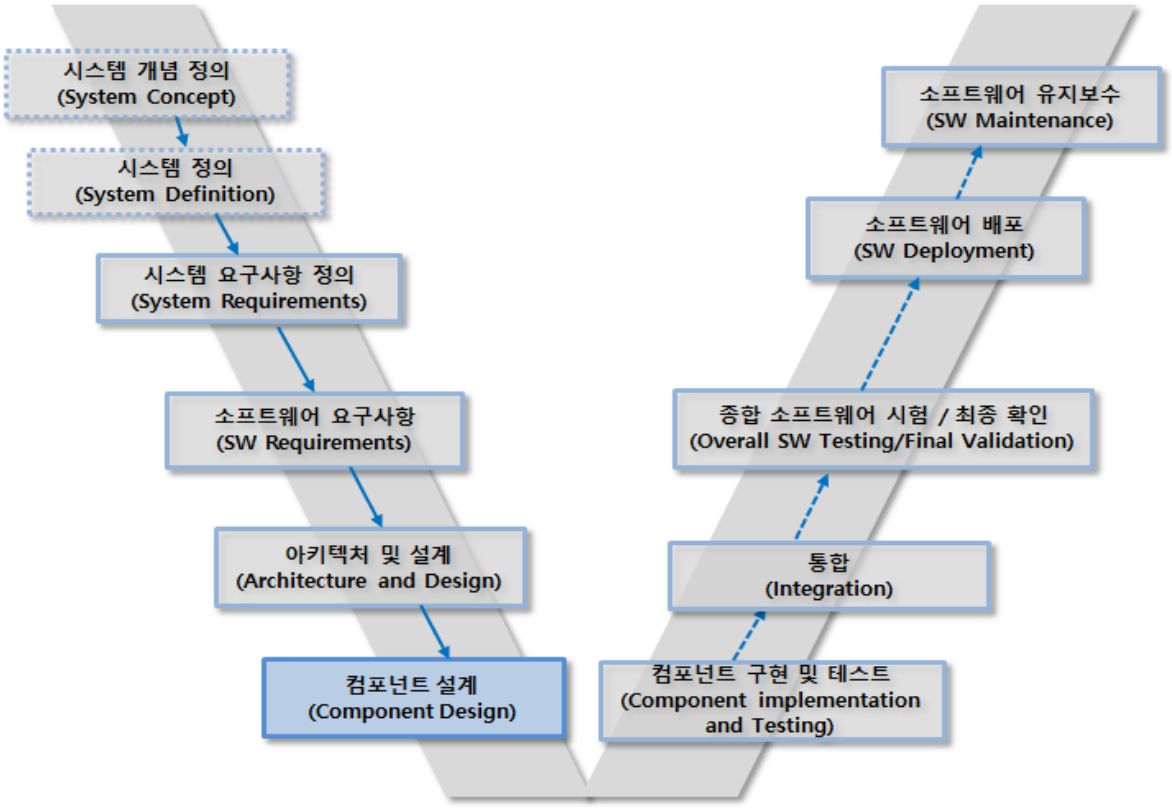


그림 107 소프트웨어 개발 생명주기 - 소프트웨어 컴포넌트 설계 단계

1.3. 시작 기준

- ‘소프트웨어 설계 명세서’ 작성 완료
- ‘소프트웨어 통합테스트 명세서 ‘ 작성 완료
- ‘소프트웨어 / 하드웨어 통합테스트 명세서’ 작성 완료
- ‘소프트웨어 아키텍처 및 설계 검증 보고서 ‘ 작성 완료

1.4. 완료 기준

- 소프트웨어 컴포넌트 내부 인터페이스 및 외부 인터페이스 기술 완료
- 소프트웨어 컴포넌트 설계 완료
- 소프트웨어 컴포넌트 내부 알고리즘 및 데이터 구조 정의 완료
- 소프트웨어 설계와 소프트웨어 컴포넌트 설계의 일관성과 양방향 추적성 수립
- 소프트웨어 컴포넌트 테스트 절차 및 테스트 케이스 도출 완료
- ‘소프트웨어 컴포넌트 설계 명세서’ 검증 완료

1.5. 입력물

- 소프트웨어 설계 명세서

1.6. 산출물

표 134 소프트웨어 컴포넌트 설계 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 컴포넌트 설계 명세서	R	HR	HR	HR	HR
소프트웨어 컴포넌트 테스트 명세서	R	HR	HR	HR	HR
소프트웨어 컴포넌트 설계 검증 보고서	R	HR	HR	HR	HR

1.7. 역할 및 책임

표 135 소프트웨어 컴포넌트 설계 단계 역할 및 책임

단 계	문 서	작성자	1차 검토	2차 검토
컴포넌트 설계	15. 소프트웨어 컴포넌트 설계 명세서	DES	VER	VAL
	16. 소프트웨어 컴포넌트 테스트 명세서	TST	VER	VAL
	17. 소프트웨어 컴포넌트 설계 검증 보고서	VER		
DES (Designer) 설계자 TST (Tester) 테스터 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 소프트웨어 컴포넌트 설계 주요 활동

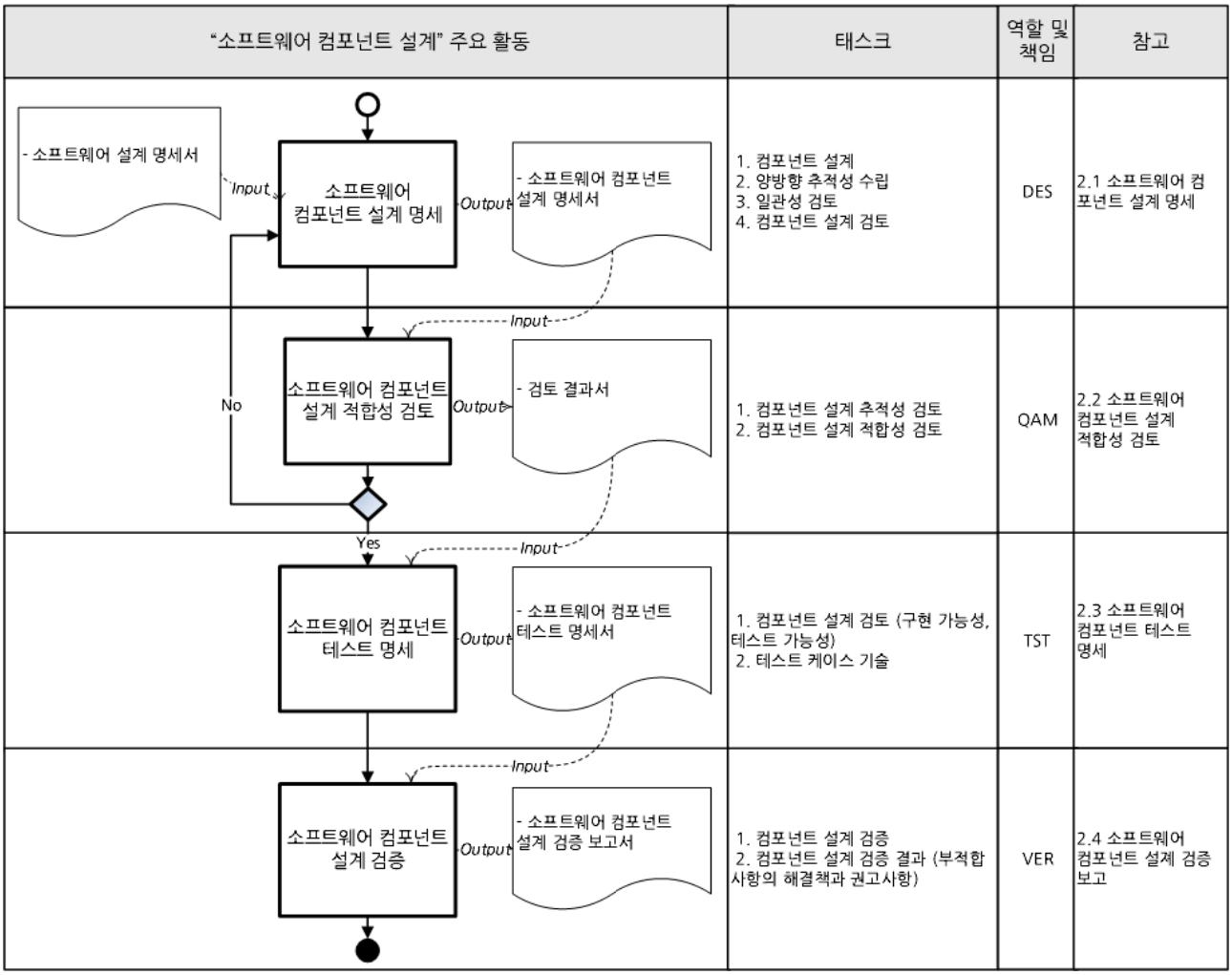


그림 108 소프트웨어 컴포넌트 설계 주요 활동

표 136 소프트웨어 컴포넌트 설계 단계

활 동 ID	활 동 명	설 명
CDES.01	소프트웨어 컴포넌트 설계 명세	• ‘소프트웨어 설계 명세서’를 준수하는 컴포넌트의 상세 설계를 위해 컴포넌트의 내적/외적 요소 및 설계 제약사항을 고려하여 기술한다. • 상세한 알고리즘 및 데이터 구조를 정의한다.
CDES.02	소프트웨어 컴포넌트 설계 적합성 검토	• ‘소프트웨어 컴포넌트 설계 추적표’를 작성하여 소프트웨어 컴포넌트 설계가 소프트웨어 설계를 준수하였는지 추적성을 검토한다. • ‘소프트웨어 컴포넌트 설계 체크리스트’를 통해 ‘ 소프트웨어 컴포넌트 설계 명세서’ 품질을 검토한다.
CDES.03	소프트웨어 컴포넌트 테스트 명세	• 컴포넌트 설계의 구현 가능성, 테스트 가능성을 검토한다. • ‘소프트웨어 컴포넌트 설계 명세서’를 기반으로 테스트 케이스를 도출하여 ‘소프트웨어 컴포넌트 테스트 명세서’를 기술한다. • 테스터는 품질보증 관리자의 검토 결과 및 관련 산출물의 보완 결과를 받아 해당 산출물에 반영한다.
CDES.04	소프트웨어 컴포넌트 설계 검증	• ‘소프트웨어 컴포넌트 설계 명세서’와 ‘소프트웨어 컴포넌트 테스트 명세서’를 기반으로 컴포넌트 설계를 검증한다. • 부적합 사유가 있으면 사유를 기술하고 이에 대한 해결책을 제시한다.

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 소프트웨어 컴포넌트 설계 명세

○ 컴포넌트 설계는 알고리즘의 상세화 및 데이터 구조를 정의함으로써 컴포넌트의 구현 단계에서 다른 문서 참조 없이 소스 코드를 구현할 수 있도록 컴포넌트 별로 상세하게 설계하는 것이다.

○ ‘소프트웨어 컴포넌트 설계 명세서’는 ‘소프트웨어 설계 명세서’를 준수하고 일관성 있게 기술한다.

2.1.1. 소프트웨어 컴포넌트 설계 명세 절차

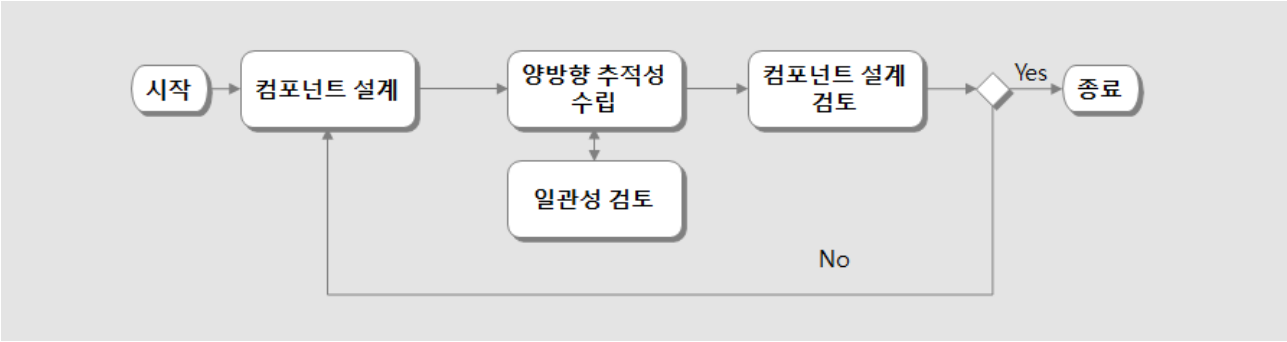


그림 109 소프트웨어 컴포넌트 설계 명세 흐름도

○ 소프트웨어 컴포넌트 설계 명세 절차

표 137 소프트웨어 컴포넌트 설계 명세 절차 설명

항 목	설 명
컴포넌트 설계	- ‘소프트웨어 설계 명세서’를 준수하여 컴포넌트 설계를 구체화 한다. - 컴포넌트 설계 시 컴포넌트의 개별 단위로 상세하게 기술한다. - 설계 명세의 상위 개념 알고리즘을 상세한 알고리즘 (코딩 구현 가능 수준)으로 구체화한다. - 정의된 컴포넌트의 데이터 구조를 정의한다.
양방향 추적성 수립	- 소프트웨어 컴포넌트 설계는 소프트웨어 설계를 준수하였다는 것을 검토한다.
일관성 검토	- 소프트웨어 설계의 의미나 목적이 소프트웨어 컴포넌트 설계와 같은지 검토한다.
컴포넌트 설계 검토	- 소프트웨어 컴포넌트 설계는 소프트웨어 설계를 준수하였는지 검토한다.

2.1.2. 소프트웨어 컴포넌트 설계 명세 지침

○ 소프트웨어 설계를 소프트웨어 컴포넌트 설계를 위해 상세하게 분해하고 컴포넌트를 개별적으로 기술한다. (5장 시범 적용 [그림 181 COMM 컴포넌트 모델 인터페이스 (예시)] 참조)

○ ‘소프트웨어 컴포넌트 설계 명세서’는 해당하는 컴포넌트의 구현 시 다른 명세서의 참조 없이 구현이 가능해야 한다.(5장 시범 적용 [그림 181 COMM 클래스 모델 (예시)] 참조)

○ 소프트웨어 컴포넌트는 다음과 같은 항목을 반드시 기술해야 한다. (7.4.4.2)

- 작성자를 기술해야 한다.
- 형상 이력은 현재 및 이전 버전을 정확하게 식별하고 변경된 내용을 기술해야 한다.
- 간략한 설명을 기술해야 한다.

○ ‘소프트웨어 컴포넌트 설계 명세서’에는 다음과 같은 내용을 반드시 포함해야 한다. (7.4.4.3)

- 상위 레벨과 역추적되는 최하위 소프트웨어 유닛 (예: 서브루틴, 메소드, 프로시저)의 식별
- 외부 요인(환경)과 상세한 인터페이스, 다른 컴포넌트와의 상세한 입력 및 출력
- 컴포넌트에 할당된 안전 무결성 등급
  - 안전 무결성 등급은 컴포넌트 단위로 할당이 되기 때문에 안전 무결성 등급이 할당되기 위해서는 더 이상 분해되지 않는 컴포넌트 이어야 한다.
- 상세한 알고리즘 및 데이터 구조

표 138 상세한 알고리즘 및 데이터 구조 설명

항 목	설 명
상 세 한 알고리즘	- 아키텍처 및 설계 단계에서는 명시한 알고리즘에 대해 코딩이 가능한 수준으로 수학적 또는 논리적으로 기술 - 잠재적인 언더플로 및 오버플로의 상태를 식별할 수 있는 수준 - 서브루틴, 메소드, 프로시저를 빠져 나왔을 때 상태가 점검되고, 오류 상태가 지시되었을 때 적절한 행위의 수행 - 재귀 사용에 대한 타당성 제시
데이터 구조	- 잠재된 자료처리 문제(부정확한 자료 초기화, 저장 자료의 부정확한 평가, 부정확한 자료 측정 등)에 대해 평가 가능 수준 - 상태변수의 일치성 및 일관성 유지

항 목	설 명
	- 입력 및 시간변수의 전체 범위에 대해 안전성 입증 가능 - 각 유닛의 입력값에 대한 유효성 확인 처리 - 안전에 중요한 데이터는 반드시 초기화 후 사용 - 부동소수점 연산 사용에 대한 타당성

○ ‘소프트웨어 컴포넌트 설계 명세서’는 가독성과 테스트 용이성을 고려해야 한다. (7.4.4.4)

○ 소프트웨어 컴포넌트 설계 시 크기와 복잡도에 대한 균형이 고려되어 컴포넌트가 구현될 때 반영될 수 있도록 기술되어야 한다. (7.4.4.5)

○ 소프트웨어 컴포넌트 설계 단계의 기법과 대책

표 139 소프트웨어 컴포넌트 설계 단계 적용 기법 및 대책 설명 (예시)

기법 및 대책	설 명
정형 명세 (Formal Specification)	- 설계 단계에서는 어떻게(How to)에 대한 구체적인 내용을 명세 - 설계가 요구사항을 만족하는지 증명 가능 - 제 3절 소프트웨어 요구사항 단계 2.1.2의 기법 및 대책 참조
데이터 흐름 다이어그램	- 시스템으로부터 정보가 어떻게 투입, 처리, 저장, 출력되는지를 보여주기 위한 것으로 외부 개체, 데이터 저장소, 데이터 프로세스, 데이터 흐름을 표현 - 부록 B-11 참조
모듈 방식	- 독립적으로 재활용 될 수 있는 소프트웨어 - 잘 정의된 인터페이스 이외에는 자신의 내부구조에 대한 직접적인 접근을 차단 - 독립적으로 테스트 될 수 있으며, 다른 모듈에 대한 의존성 최소화 - 제공되는 인터페이스를 지원하는 다른 모듈로 대체되는 것도 가능 - 계층화를 통해 제공해야 할 인터페이스와 자신이 의존하고 있는 인터페이스를 명확하게 분리 - 부록 B-38 참조
컴포넌트	- 소프트웨어 개발을 블록 쌓듯이 쉽게 할 수 있도록 하는 기법 - 기존의 코딩 방식에 의한 개발에서 벗어나 소프트웨어 모듈을 미리 만든 뒤 필요한 응용 기술을 개발할 때 이 모듈을 조립하는 기법 - 재사용성이 가능한 장점 보유
정보 은닉화	- 한 모듈에서 인터페이스와 구현을 명확하게 분리하여 각 모듈의 내부 항목에 대한 정보는 감추고, 인터페이스를 통해서만 메시지를 전달하는 기법 - 다른 모듈을 변경하지 못하게 한다. - 모듈 안에 있는 자료 구조와 메소드에 사용된 알고리즘은 외부에서 그 값을 직접 변경할 수 없고, 공개 인터페이스로 정의된 메소드를 통해서만 접근 가능

기법 및 대책	설 명
정보 캡슐화	- 객체 내부의 관련된 데이터와 메소드를 함께 포장하는 방식 - 캡슐 내부와 외부를 구별
매개 변수 개수 제한	- 컴포넌트의 인터페이스에서 매개변수와 리턴값으로 메시지 확인 - 매개 변수의 개수를 제한하는 것은 소프트웨어의 복잡도와 실행 환경의 메모리(스택) 사용에 밀접한 관련 - IEC 62279 표준에서 제시하는 것은 5개 이하
설계 및 코딩 표준	- 소스코드 구현 시 일관성을 유지하기 위해 코딩 규칙 및 코딩 스타일을 사용하여 소스코드 작성 시 에러를 줄이고 소스코드의 가독성 향성
분석 가능한 프로그램	- 정적 분석이 가능한 프로그램을 제작함. - 정적 프로그램 분석은 실제 실행 없이 소프트웨어를 분석 - 프로그램의 이해를 통해 인간에 의한 자동화된 도구를 사용한 분석, 또는 코드 검토에 적용
엄격한 형식의 프로그래밍 언어	- 기본 데이터 타입에 대해 엄격한 형식을 정의하는 언어이다. - 예시로 엄격한 형식의 프로그래밍 언어는 정수(INTEGER)와 실수(REAL)를 엄격하게 구분하여 서로 혼용하여 사용하지 못하게 한다. 하지만 엄격하지 못한 또는 유연성을 가진 언어는 정수와 실수의 혼용을 사용할 수 있다. 즉 실수에 정수를 대입하여 사용할 수 있다.
구조적 프로그래밍	- 저수준 관점에서 간단하고, 계층적인 프로그램 제어 구조로 구성 - 순차, 선택, 반복 이라는 3가지 형태로 수행 - 대표적인 예로 GOTO를 사용하지 않거나 GOTO 사용을 축소 가능 - 큰 조각의 코드를 이해하기 쉬운 크기의 작은 하부 프로그램 (함수, 프로시저, 메소드, 블록 등)으로 세분화 - 전역 변수는 거의 사용하지 않아야 하고, 하부 프로그램은 지역 변수를 사용하거나, 값이나 참조에 의한 인자 (매개변수)로 받음. - 전체 프로그램을 한 번에 이해하지 않고, 분리된 작은 코드 조각을 쉽게 이해 가능하게 함.
프로그래밍 언어	- 안전 무결성 등급에 따라 적절한 언어를 선택 - 철도 시스템에서는 전통적으로 C언어를 가장 많이 사용-
언어 하위 집합	- 언어의 기본 제공 기능에서 시스템 고장이나 위험원을 유발 시킬 수 있는 기능을 제외 혹은 변형된 형태의 기능으로 제공 - 대표적인 예가 C언어의 printf() 함수이다. printf()의 오용은 전체 시스템을 셧다운 또는 홀딩 시킬 수 있거나 보안에 심각한 취약점을 보임.
객체 지향 언어	- 객체 중심 프로그래밍을 위해 사용하는 언어 - 연산문의 집합 - 객체는 자료와 프로그램의 추상화로써 구현

기법 및 대책	설 명
	- 연산하고자 하는 여러 가지 객체 속에서 그 연산의 정의가 나타남 - 객체에 대한 정의는 그 연산의 여러 가지 측면을 나타냄. - 절차적 프로그래밍 언어에 대응
절차적 프로그래밍	- 절차 (순서) 대로 프로그램을 처리하는 방식 - C언어가 대표적으로 함수 위주의 구조화된 방식 - 절차식 프로그래밍은 함수를 중심으로 프로그램을 설계한 후 필요한 데이터를 정의
메타 프로그래밍	- 자기 자신 혹은 다른 컴퓨터에서 컴퓨터 프로그램을 데이터로 처리

2.1.3. 소프트웨어 컴포넌트 설계 명세서 템플릿

1. **서론**

- 컴포넌트 설계를 간략하게 기술한다.

1.1 개요

- 컴포넌트 설계의 개요를 기술한다.

1.2 문서의 목적

- 문서의 목적을 기술한다.

1.3 관련 문서

1.3.1 프로젝트 문서

- 프로젝트에서 산출되는 문서를 나열한다.

1.3.2 참고 문서

- 참고 문서를 나열한다.

2. 컴포넌트 설계 개요

2.1 설계 목표 및 개념

- 컴포넌트 설계의 목표와 개념을 설명한다.

2.2 설계 도구 및 환경

- 설계에 사용된 도구와 그 사용 환경에 대해 기술한다.
- 안전 무결성 등급과 적합한 도구를 사용해야 한다.

3. 컴포넌트 상세 설계

- 각각의 컴포넌트에 대해 상세 설계를 기술한다.

A. 부록

- 추가적인 사항을 기술한다.

그림 110 소프트웨어 컴포넌트 설계 명세서 템플릿 (예시)

2.1.4. 소프트웨어 컴포넌트 설계 명세서 체크리스트

표 140 소프트웨어 컴포넌트 설계 명세서 체크리스트 (예시)

구 분	점검 사항
정확성/완전성	모든 소프트웨어 컴포넌트는 식별되었는가?
	로컬 데이터는 정확하게 정의되었는가?
	상세한 알고리즘은 정의되었는가?
	모든 데이터의 입출력이 정의되었는가?
	설계가 테스트될 수 있는 시점이 식별되어 있는가?
	설계의 모든 요소가 소프트웨어 요구사항으로 추적 가능한가?
	쓰레드 응답 시간 요구사항 한계 내에 있는가?
일관성	설계가 내부적으로 일관성이 있는가?
	소프트웨어 인터페이스가 일관성이 있는가?
	데이터 정의와 처리가 일관성이 있는가?
	설계가 다른 문서와 일관성이 있는가?
구현 가능성	설계가 현재 프로젝트의 제약사항 아래에서 정해진 비용과 일정 안에서 수행될 수 있는가?
	설계는 알려진 또는 증명된 것을 기초로 했는가?
표준 준수성	'설계 명세서'를 활용하여 구현할 수 있는가?
	설계는 이해가 용이하고, 다양하게 해석되는 경우는 없는가?
	모든 설계 정보가 적절한 분류에 의해 표시되고 있는가?
	문서는 요구되는 표준에 의해 작성되는가?
(기타)	기타항목

2.2. 소프트웨어 컴포넌트 설계 적합성 검토

- 소프트웨어 컴포넌트 설계 적합성은 소프트웨어 품질 보증 계획에 따라 작성자와 별도의 조직에서 검토해야 한다.
- 추적표를 작성하여 소프트웨어 컴포넌트 설계가 소프트웨어 설계 명세에서 도출되었는지 추적성을 검토한다. (별도로 추적표를 작성하지 않고 ‘소프트웨어 컴포넌트 설계 명세서’ 내에서 소프트웨어 설계 명세와의 연관성을 표시하기도 한다.)
- ‘소프트웨어 컴포넌트 설계 명세서 체크리스트’를 통해 정확성/완전성, 일관성, 구현 가능성, 표준 준수성을 검토한다.
- 소프트웨어 개발 생명주기에서 소스코드의 구현으로 연계되는 단계이므로 소스코드의 구현 시 고려되는 사항(예: 소스코드의 크기와 복잡도를 고려)을 기술해야 하고 설계 적합성을 반드시 검토해야 한다.

2.3. 소프트웨어 컴포넌트 테스트 명세

○ 소프트웨어 컴포넌트 테스트는 소프트웨어 컴포넌트 설계에 대한 구현 가능성 및 테스트 가능성을 검토하고 테스트 케이스를 기술한다.

2.3.1. 소프트웨어 컴포넌트 테스트 명세 절차

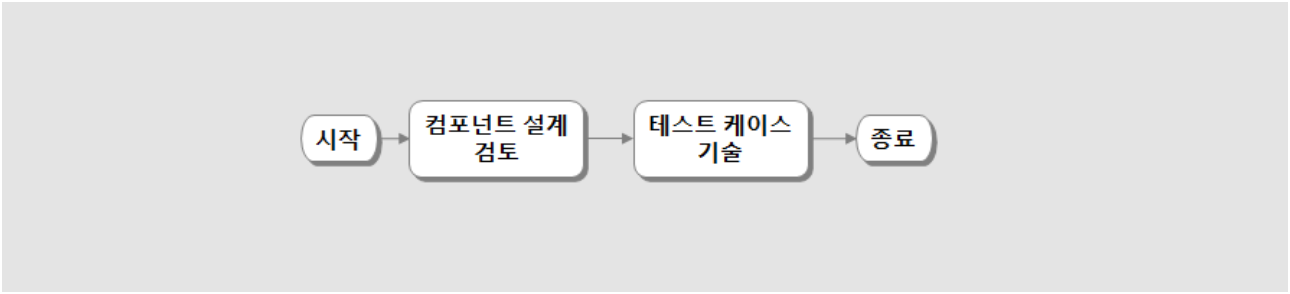


그림 111 소프트웨어 컴포넌트 테스트 명세 흐름도

○ 소프트웨어 컴포넌트 테스트 명세 절차

표 141 소프트웨어 컴포넌트 테스트 명세 절차 설명

항 목	설 명
컴포넌트 설계 검토	- 안전 무결성 등급에 따라 기법을 선택하고 검토한다. - 구현 가능성을 검토한다. - 테스트 용이성을 검토한다.
테스트 케이스 기술	- 테스트 케이스가 요구사항과 관련되어 있다는 것을 기술한다. (관련 요구사항 번호 명시) - 입력값을 기술한다. - 예상 출력값을 기술한다. - 완료기준을 기술한다.

2.3.2. 소프트웨어 컴포넌트 테스트 명세 지침

○ 소프트웨어 컴포넌트 테스트 명세는 다음과 같은 사항을 기술해야 한다. (7.4.4.8)

- 테스트 목적
- 테스트 케이스, 테스트 데이터, 예상 결과값
- 테스트 환경, 도구, 형상 및 프로그램
- 테스트 완료 기준
- 테스트 절차에 소속되어 있는 구성원의 역할과 책임
- 테스트 케이스가 어떤 컴포넌트 설계를 테스트하는지 기술해야 한다.
- 소프트웨어 테스트 장비의 선택과 활용을 기술해야 한다.

○ 테스트는 다음 3가지 목적을 준수 가능하도록 설계해야 한다. (7.4.4.9)

- 컴포넌트가 의도된 기능을 수행한다는 것을 증명해야 한다. (블랙박스 테스팅)
- 의도된 기능을 수행하기 위해 컴포넌트 내부의 상호작용을 검토한다. (블랙/화이트 박스 테스팅)

표 142 블랙박스 및 화이트박스 테스트 설명

테스트 종류	설 명
블랙박스	- 입력과 예상 출력을 확인 - 내부적인 상호작용 확인 불가
화이트박스	- 입력과 예상 출력 확인 - 내부적인 상호작용 확인 - 컴포넌트 커버리지 확인 가능

- 컴포넌트의 모든 부분은 테스트가 된다는 것을 증명해야 한다.
  - 테스트 되지 않는 컴포넌트는 컴포넌트 설계에 오류가 있는 것이므로 오류사항을 수정해야 한다. 필요 시 이전 단계 (아키텍처 및 설계 단계 또는 요구사항 단계)로 피드백 한다.

○ 각 소프트웨어 컴포넌트를 테스트하기 위한 ‘소프트웨어 컴포넌트 설계 명세서’와 소프트웨어 컴포넌트 테스트 케이스를 다음 사항을 고려하여 기술한다.

- 테스트 결과에 대한 설명과 각 컴포넌트가 ‘소프트웨어 컴포넌트 설계 명세서’의 지침을 준수하는지 여부를 기술해야 한다.
- 테스트 실패 항목을 기술해야 한다.
- 테스트 케이스와 결과는 후속 분석을 위해 프로그램이 읽을 수 있는 언어로 기록하는 것이 좋다.
- 테스트는 반복 가능해야 하며, 실행 가능한 경우 자동화된 방식으로 수행되어야 한다.
- 자동 테스트 실행을 위해 작성된 테스트 스크립트가 검증되어야 한다.

2.3.3. 소프트웨어 컴포넌트 테스트 명세서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 컴포넌트 테스트를 정의한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘시스템 요구사항 명세서’, ‘시스템 안전 요구사항 명세서’, ‘시스템 아키텍처 기술서’, ‘소프트웨어/하드웨어 인터페이스 명세서’, ‘소프트웨어 요구사항 명세서’

1.4 약어표

- 약어에 대한 설명을 한다.
- 예시 1) ATP – Automatic Train Protection
- 예시 2) MMI – Man Machine Interface

2. 소프트웨어 컴포넌트 테스트 개요

- IEC 62279에서 요구하는 안전 수준에 따른 테스트 기법을 선택하여 기술한다.
- 제약사항을 기술한다.

3. 테스트 환경 및 절차

- 하드웨어의 특성과 구성을 기술한다.
- 예시) 차상 MMI 시스템은 정보를 표현하기 위해 800x600의 컬러 LCD가 필요하다.
- 테스트 수행에 필요한 시스템과 응용 소프트웨어를 기술한다. 운영체제, 컴파일러, 시뮬레이터 등이 있다.
- 테스트 수행에 필요한 절차 및 제약사항을 기술한다. 제약사항에는 특정 셋업, 운영자 개입 여부, 출력 확인 절차 등이 포함될 수 있다.

4. 테스트 세트

- 테스트할 대상의 컴포넌트를 모두 나열한다.

5. 테스트 케이스

- 테스트 세트에서 나열한 컴포넌트를 테스트하기 위한 테스트 케이스를 기술한다.
- 예시)

테스트 케이스 ID	관련 요구사항 ID	대상	커버리지 (Coverage)	범위 (Range)	메소드 (Method)	비고
TC_HLRQ_11	소프트웨어 SRS_MMI_REQ1.2	OBJ_1	P	N	I	요구사항에 따라 필요한 행위 충족

* OBJ_1: 헬스 모니터링에 관련된 객체 (데이터타입 및 심볼정의, 시스템 헬스 모니터 테이블, 헬스 모니터 테이블 컴포넌트, 헬스 모니터 모드, 에러 처리, 트랩 코드 매핑, 초기 조건, 헬스 모니터링 진입시점)
* OBJ_2: 파티션 간 메시지 통신을 위한 객체 (큐잉포트 개방, 샘플링포트 개방, 큐잉포트 채널 생성, 샘플링포트 채널 생성)
* 커버리지, 범위, 메소드 관련:
F: 요구사항이 완전하게 테스트 됨.
P: 요구사항이 부분적으로 테스트 됨.
N: 정상 범위 테스트 케이스
R: 강인성 테스트 케이스
N+R: 정상 범위 및 강인성 테스트 케이스
T: 요구사항을 테스트로 검증
I: 요구사항을 인스펙션(inspection)으로 검증
A: 요구사항을 분석으로 검증

A. 부록

- 추가적인 사항을 기술한다.

그림 112 소프트웨어 컴포넌트 테스트 명세서 템플릿 (예시)

2.3.4. 소프트웨어 컴포넌트 테스트 명세서 체크리스트

표 143 소프트웨어 컴포넌트 테스트 명세서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
품질 요구사항	테스트 목적, 테스트 케이스, 테스트 데이터, 예상 결과값, 테스트 환경, 도구, 형상 및 프로그램, 테스트 완료 기준을 기술하였는가?
	테스트 절차에 소속되어 있는 구성원의 역할과 책임이 명시되어 있는가?
	테스트 케이스가 어떤 요구사항을 테스트 하는지 명시하는가?
	테스트 장비의 선택 기준과 활용 방안이 기술되어 있는가?
(기타)	기타항목

2.4. 소프트웨어 컴포넌트 설계 검증 보고

○ ‘소프트웨어 컴포넌트 설계 명세서’와 ‘소프트웨어 컴포넌트 테스트 명세서’를 기반으로 적합하게 기술되었는지 검증한다.

2.4.1. 소프트웨어 컴포넌트 설계 검증 보고 절차

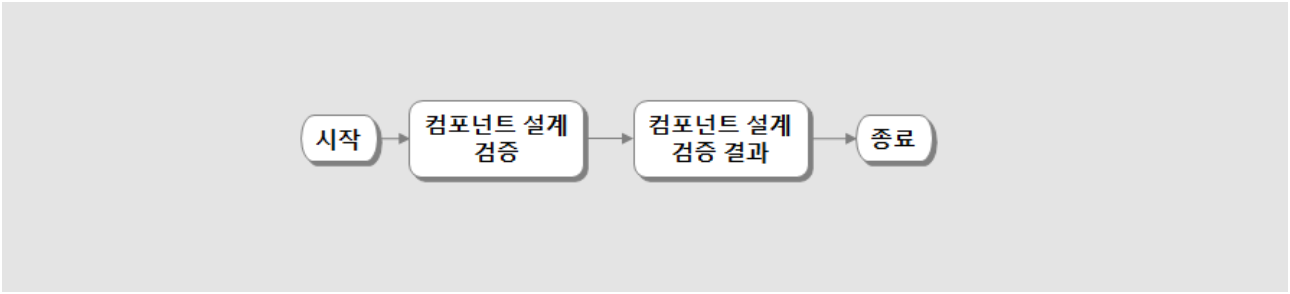


그림 113 소프트웨어 컴포넌트 설계 검증 보고 흐름도

○ 소프트웨어 컴포넌트 설계 검증 보고 절차

표 144 소프트웨어 컴포넌트 설계 검증 보고 절차 설명

항 목	설 명
컴포넌트 설계 검증	- 컴포넌트 설계가 의도된 데로 기술되었는지 검토한다. - 타당성: 소프트웨어 컴포넌트 설계는 ‘소프트웨어 설계 명세서’를 적절히 반영해야 한다. - 일관성: 소프트웨어 설계에서 정의한 기능이 일관성 있게 기술되어야 한다. - 완전성: 모든 기능과 제약사항을 기술해야 한다. - 현실성: 현재 시점의 하드웨어/소프트웨어 기술을 사용하여 해결할 수 있는 컴포넌트 설계인지 예측해야 한다.
컴포넌트 설계 검증 결과	- 컴포넌트 설계 검증 결과를 기술한다. - 부적합 사항의 해결책과 권고사항을 기술한다.

2.4.2. 소프트웨어 컴포넌트 설계 검증 보고서 지침

○ 소프트웨어 컴포넌트 설계 검증 보고서는 다음과 같은 사항을 기술해야 한다. (7.4.4.12)

- 검증자 이름과 검증 항목의 식별과 형상을 기술해야 한다.
- ‘소프트웨어 컴포넌트 설계 명세서’를 준수하지 않는 항목을 기술해야 한다.
- 컴포넌트, 데이터, 구조, 알고리즘이 적당하지 않아 문제를 발생시키는 것을 기술해야 한다.
  - 불명확한 사항에 대해 해결책 및 권고 사항을 기술한다.
- 오류 또는 불완전한 사항의 검지 여부를 기술해야 한다.
- 소프트웨어 검증 계획의 준수 여부를 기술해야 한다. 계획 미준수에 대한 중요도를 명시한다.
- 컴포넌트 설계 시 가정한 사항이 있다면 기술해야 한다.
- 검증 결과를 요약 한다.

○ 소프트웨어 컴포넌트 설계가 확정된 후 소프트웨어 설계 명세를 준수하고 있음을 검증해야 한다. (7.4.4.13)

○ ‘소프트웨어 컴포넌트 설계 명세서’는 가독성과 추적성을 가지는지 검증해야 한다. (7.4.4.13)

○ ‘소프트웨어 컴포넌트 테스트 명세서’는 ‘소프트웨어 컴포넌트 설계 명세서’의 모든 테스트 케이스를 기술했는지 검증해야 한다. (7.4.4.13)

○ ‘소프트웨어 컴포넌트 테스트 명세서’는 가독성과 추적성을 가지는지 검증해야 한다. (7.4.4.13)

○ ‘소프트웨어 컴포넌트 설계 명세서’는 ‘소프트웨어 설계 명세서’를 소프트웨어 컴포넌트로 세분화하는 과정에서 다음과 같은 사항을 고려했는지 검증한다. (7.4.4.13)

- 요구되는 성능의 구현 가능성
- 검증에 대한 테스트 가능성
- 유지보수성

2.4.3. 소프트웨어 컴포넌트 설계 검증 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 컴포넌트 설계를 검증한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘소프트웨어 컴포넌트 명세서’, ‘소프트웨어 컴포넌트 테스트 명세서’

1.4 약어표

- 예시 1) ATP – Automatic Train Protection
- 예시 2) MMI – Man Machine Interface

2. 소프트웨어 컴포넌트 설계 검증 개요

- 컴포넌트 설계 검증 활동에 필요한 조직, 소프트웨어 안전 무결성 등급, 사용 도구, 검증 기법, 제약 사항 등을 기록한다.

3. 검증 결과

- 검증 결과를 기술한다.

4. 검증 요약

- 검증 결과를 요약한다.
- 부적합사항을 기술하고 부적합사항의 해결책과 권고사항을 기술한다.

A. 부록

그림 114 소프트웨어 컴포넌트 설계 검증 보고서 템플릿 (예시)

2.4.4. 소프트웨어 컴포넌트 설계 검증 보고서 체크리스트

표 145 소프트웨어 컴포넌트 설계 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?

구 분	점검 사항
일반 요구사항	‘소프트웨어 컴포넌트 설계 명세서 체크리스트’, ‘소프트웨어 컴포넌트 테스트 명세서 체크리스트’를 활용하여 검증한 결과를 기술하였는가?
	검증 결과에 종합적인 분석과 검증자, 수행일시를 포함하여 기술하였는가?
	검증 결과에 확인된 문제들과 적용된 조치사항을 기술하였는가?
품질 요구사항	‘소프트웨어 컴포넌트 설계 명세서’를 준수하지 않는 항목을 기술되어 있는가?
	컴포넌트, 데이터, 구조, 알고리즘이 적당하지 않아 문제를 발생시키는 것을 식별되어 있는가?
	불명확한 사항에 대해 해결책 및 권고 사항을 기술되어 있는가?
	오류 또는 불완전한 사항의 검지 여부가 기술되어 있는가?
	컴포넌트 설계 시 가정한 사항이 있다면 기술되어 있는가?
(기타)	기타항목

제 4 절 소프트웨어 컴포넌트 구현 및 테스팅

1. 개요

분석, 테스팅, 검증 가능하고 유지보수가 용이한 소프트웨어 컴포넌트 구현과 구현된 소프트웨어 컴포넌트 테스팅을 위해 필요한 정보를 기술한다.

1.1. 목표

- ○ 분석, 테스팅, 검증 가능하고 유지보수가 용이한 소프트웨어 컴포넌트 구현
- ○ 소프트웨어 컴포넌트 테스팅

1.2. 범위

- ○ 소프트웨어 개발 생명주기에서 IEC 62279 7.5에 해당하는 소프트웨어 컴포넌트 구현 및 테스팅 단계에 대해 설명한다.

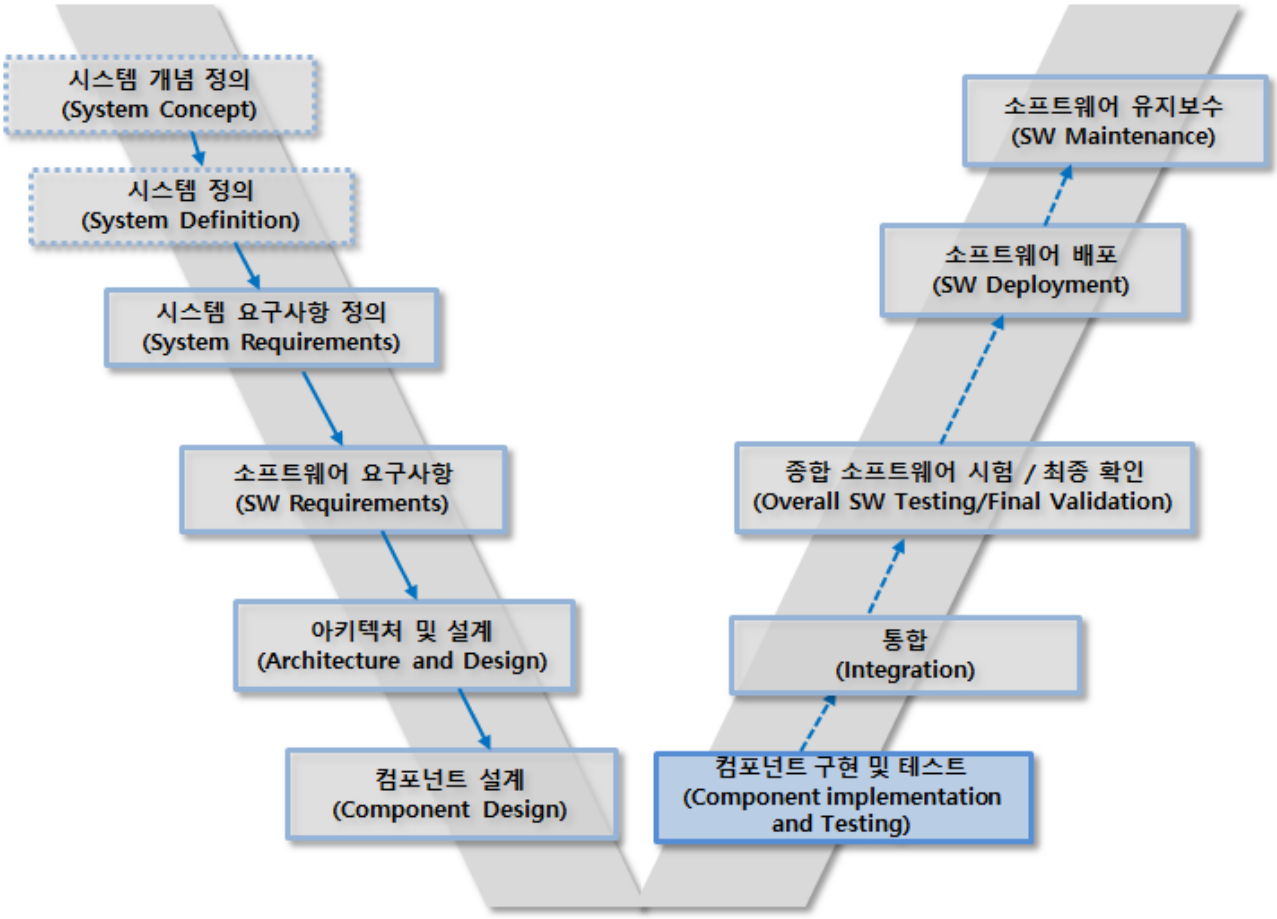


그림 115 소프트웨어 개발 생명주기 - 소프트웨어 컴포넌트 구현 및 테스팅 단계

1.3. 시작 기준

○ ‘소프트웨어 컴포넌트 설계 명세서’ 작성 완료

○ ‘소프트웨어 컴포넌트 설계 명세 검증 보고서’ 완료

1.4. 완료 기준

○ 소스코드 구현 완료

○ ‘소프트웨어 컴포넌트 테스트 보고서’ 작성 완료

○ ‘소프트웨어 소스코드 검증 보고서’ 작성 완료

○ ‘소프트웨어 검증 보고서’에 소프트웨어 구현 및 테스팅 단계 작성 완료

1.5. 입력물

○ 소프트웨어 컴포넌트 설계 명세서

○ 소프트웨어 컴포넌트 테스트 명세서

1.6. 출력물

표 146 소프트웨어 컴포넌트 구현 및 테스팅 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 소스코드 및 지원 문서	HR	HR	HR	HR	HR
소프트웨어 컴포넌트 테스트 보고서	R	HR	HR	HR	HR
소프트웨어 소스코드 검증 보고서	HR	HR	HR	HR	HR

1.7. 역할 및 책임

표 147 소프트웨어 컴포넌트 구현 및 테스팅 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
컴포넌트 구현 및 테스팅	18. 소프트웨어 소스코드 및 지원 문서	IMP	VER	VAL
	19. 소프트웨어 컴포넌트 테스트 보고서	TST	VER	VAL
	20. 소프트웨어 소스코드 검증 보고서	VER		VAL
IMP (Implementer) 개발자 TST (Tester) 테스터 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 소프트웨어 컴포넌트 구현 및 테스팅 주요 활동

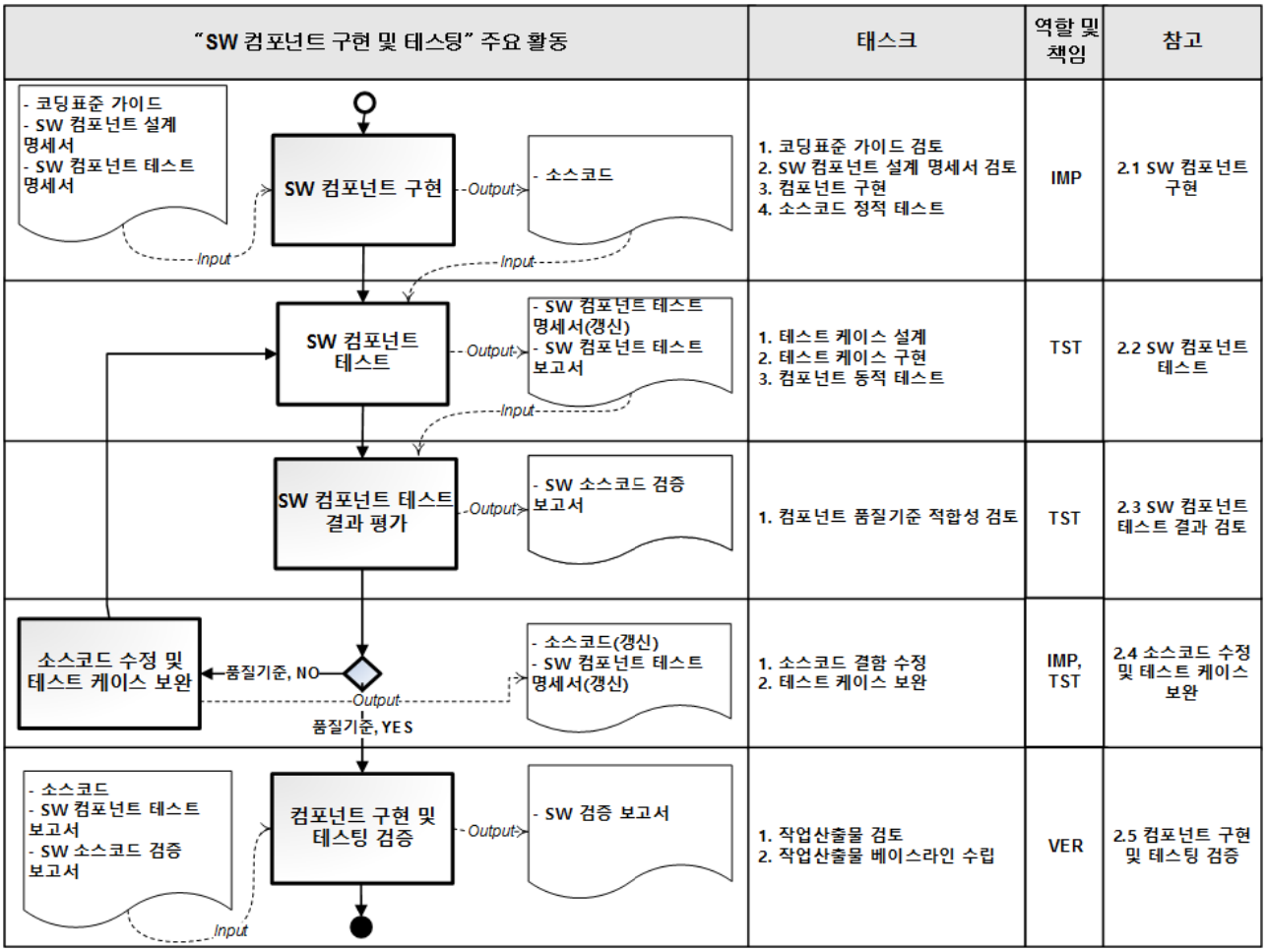


그림 116 소프트웨어 컴포넌트 구현 및 테스팅 단계 주요 활동

표 148 소프트웨어 컴포넌트 구현 및 테스팅 단계 주요 활동 설명

활 동 ID	활 동 명	설 명
CIMPnT.01	소프트웨어 컴포넌트 구현	• 소프트웨어 컴포넌트 테스트 계획을 수립하고 컴포넌트 구현 우선순위를 결정한다. • '소프트웨어 컴포넌트 설계 명세서'의 내역을 코딩 가이드에 따라 정의된 프로그래밍 언어로 컴포넌트의 세부 알고리즘을 구현한다. • '소프트웨어 컴포넌트 설계 명세서'의 모든 항목이 컴포넌트 소스코드에 반영되도록 일관성 있게 구현한다. • 소프트웨어 컴포넌트 설계 명세의 모든 항목이 구현되었음을 추적표를 작성하여 점검한다. • 코드 리뷰 또는 자동화된 도구를 사용한 정적 분석을 수행하여 구현된 소스코드 품질을 검증하고 결과를 '소프트웨어 컴포넌트 테스트 결과 보고서'에 기술한다. • 소스코드에서 발견된 문제점을 수정한다.

활 동 ID	활 동 명	설 명
CIMPnT.02	소프트웨어 컴포넌트 테스트	• 수립된 컴포넌트 테스트 계획 및 절차를 기반으로 컴포넌트를 테스트하기 위한 테스트 케이스를 설계하고 구현한다. • 테스트 자동화 도구를 활용하여 테스트 커버리지를 만족하기 위한 테스트 케이스를 작성한다. • 추적표를 작성하여 ‘소프트웨어 컴포넌트 설계 명세서’와 컴포넌트 테스트 케이스 간의 추적성을 확인한다. • ‘소프트웨어 컴포넌트 테스트 명세서’의 테스트 수행 절차에 따라 테스트 수행환경과 데이터를 준비하고 각 테스트 케이스별로 테스트를 실시하고 결과를 ‘소프트웨어 컴포넌트 테스트 보고서’에 기술한다. • 발견한 문제점을 기록하고 해결 방안을 검토한다.
CIMPnT.03	소프트웨어 컴포넌트 테스트 결과 평가	• 컴포넌트 테스트 결과가 ‘소프트웨어 컴포넌트 테스트 명세서’에 기술된 목표한 품질기준을 만족하는지 확인한다. • 컴포넌트 소스코드 수정 여부, 회귀 테스트 수행 필요 여부를 결정하고 그 내역을 기록한다.
CIMPnT.04	소스코드 수정 및 테스트 케이스 보완	• 테스트 결과가 계획된 품질기준을 만족하지 못하면 소스코드를 수정하거나 테스트 케이스를 보완하여 품질기준에 미달된 컴포넌트를 대상으로 테스트를 수행한다.
CIMPnT.05	소프트웨어 컴포넌트 구현 및 테스팅 검증	• 소프트웨어 컴포넌트 구현 및 컴포넌트 테스트 결과를 기반으로 소프트웨어 소스코드 검증 절차에 따라 검증을 수행한다. • 소프트웨어 요구사항과 컴포넌트 설계 사항이 소스코드에 반영되었는지 여부를 추적표를 구성하여 검토한다.

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 소프트웨어 컴포넌트 구현

2.1.1. 소프트웨어 컴포넌트 구현 절차

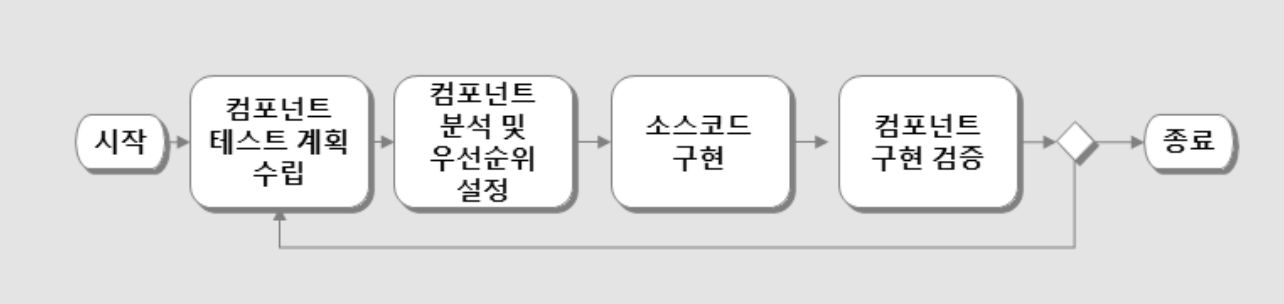


그림 117 소프트웨어 컴포넌트 구현 흐름도

표 149 소프트웨어 컴포넌트 구현 절차 설명

항 목	설 명
컴포넌트 테스트 계획 수립	- 컴포넌트 테스트 방법(정적, 동적 테스트) 정의 - 컴포넌트 소스코드 품질기준(코딩규칙 위배율, 코드 커버리지) 정의
컴포넌트 분석 및 우선순위 설정	- 작성된 소프트웨어 설계 명세서의 내용을 검토한다. - 컴포넌트 구현 우선순위를 결정한다.
소스코드 구현	- 작성된 코딩표준가이드 내용을 검토한다. - 소프트웨어 컴포넌트 설계와 코딩 표준을 준수하여 정의된 프로그래밍 언어로 컴포넌트를 구현한다.
컴포넌트 구현 검증	- 소스코드가 컴포넌트 설계 및 코딩표준을 준수하여 구현되었는지 여부를 검토한다. - 구현된 컴포넌트 소스코드와 소프트웨어 요구사항 간의 추적성을 확인한다. - 구현된 컴포넌트 소스코드와 소프트웨어 컴포넌트 설계 간의 추적성을 확인한다.

2.1.2. 소프트웨어 컴포넌트 구현 지침

○ 소스코드는 ‘소프트웨어 컴포넌트 설계 명세서’를 기반으로 개발자(Implementer)의 책임 하에 구현되어야 하며, 소스코드 구현 관련 요구사항들은 다음과 같다. (7.5.4.1)

○ 소스코드는 크기(size)와 복잡도(complexity)는 균형을 이루어야 한다. 다음 표는 소프트웨어 SIL 등급별로 권장하는 소스코드 복잡도(Cyclomatic Complexity) 기준 예시이다. (7.5.4.2)

표 150 소프트웨어 SIL별 복잡도 기준 (예시)

소프트웨어SIL	복잡도 기준
SIL 1	- 복잡도 50 이하 허용
SIL 2	- 복잡도 20 이하 허용
SIL 3	- 복잡도 10 이하 허용
SIL 4	

○ 소스코드는 읽기 쉬워야 하고, 이해하기 쉬워야 하며, 테스트 가능해야 한다. (7.5.4.3)

- 소스코드 품질을 높은 수준으로 유지하기 위해 설계 단계에서 수립된 코딩규칙 표준을 준수하여 소스코드 구현 시 적용하는 것이 좋다.
- 정적 분석(Static Analysis)은 자동화된 도구에 의해서 수행되는 정적 분석 과정으로, 정의된 코딩규칙 표준을 기반으로 테스트 대상이 되는 소스코드를 분석하여 결과를 내는 원리로 동작한다.

표 151 주요 코딩규칙 (예시)

코딩규칙 항목	설 명
소스코드 파일의 코드 라인수(LOC) 제한	- 소스코드 분석이 용이하도록 하나의 소스코드 파일은 코드 라인수(LOC)를 제한하는 것이 좋다.
함수의 최대 중첩 깊이(nesting depth) 제한	- 최대 중첩 깊이(nesting depth)가 높다는 것은 제어문이 중복되어 조합되어 있어 내부 제어흐름 구조가 복잡하여 소스코드 가독성을 저해한다. - 함수가 과다한 기능을 수행하는 경우 기능 분해를 통해 함수를 재구성해야 한다. - 최대 nesting depth를 5 이하로 제한하는 것이 좋다.
0으로 나눗셈 연산 금지	- 0으로 나누게 되면 소프트웨어 실행 시 에러가 발생하므로 제수가 0인지 여부를 검증하고 사용해야 한다.
수행되지 않는 코드 작성 금지	- 예외를 처리하기 위한 방어 코드를 제외하고 진입점이 없어 수행될 수 없는 코드가 없어야 한다.
문장이 있는 switch 절은 break문으로 끝냄	- 방어적(Defensive) 프로그래밍의 일환으로 모든 switch 조건절 또는 복합문을 구성하는 마지막 문장은 break 여야 한다. - 이는 의도하지 않는 미완료(Fall through) 오류를 방지한다.
비교 조건식의 연산결과가	- 조건식의 연산결과가 항상 참 또는 거짓인 경우는 대부분의 경우

코딩규칙 항목	설 명
항상 동일하게 나오는 관계연산자 조합사용 금지	에러이다. - 이는 의도하지 않는 연산 결과나 실행되지 않는 코드(Unreachable code)를 유발한다.
변수 사용 전에 값 할당이 되어야함	- 모든 변수는 사용하기 전에 값이 할당되어야 한다. - 값이 할당되지 않는 변수 사용 시 쓰레기 값으로 인해 에러가 발생할 수 있다.
미사용 변수 검사	- 사용되지 않는 변수는 가독성을 저해하고 논리적 오류를 유발할 수 있다. - 미사용 변수는 소스코드에서 제거해야 한다.
함수의 매개변수(Parameter)를 사용하기 전에 검사	- 방어적(Defensive) 프로그래밍의 일환으로 함수에 사용된 매개변수(Parameter)는 사용 전에 유효한 범위의 값인지 확인하고 사용해야 한다.
함수의 매개변수 수 제한	- 함수의 매개변수 수는 5 이하로 제한하는 것이 좋다.
함수의 반환 값(return value) 검증	- 방어적(Defensive) 프로그래밍의 일환으로 함수의 반환 값은 검증해야 한다.
재귀 호출(Recursion Call) 지양	- 증명할 수 없고 검사할 수 없는 호출을 피하기 위해서 함수의 재귀호출은 지양해야 한다. - 재귀 호출을 사용해야 한다면, 재귀의 깊이를 예측할 수 있는 판단 기준이 명확해야 한다.
주석(Comment) 비율 검사	- 가독성 향상을 위해 소스코드 주석 비율은 50% 이상을 유지하는 것이 좋다.
함수 복잡도 제한	- 소스코드 구조의 복잡도를 최소화하기 위해 함수 복잡도(Cyclormatic complexity number)를 50 이하로 제한하는 것이 좋다. - 수행 가능한 경로의 수가 많으면 내부 제어흐름 구조가 복잡해져 소스코드 가독성을 저해한다. - 또한 복잡도가 높으면 테스트 커버리지 품질기준을 달성하기 위한 테스트 케이스 설계 및 구현이 어려워진다.
함수의 진입점과 출구점 수 제한	- 구조의 복잡도를 낮추기 위해 함수는 하나의 진입점(Entry Point)와 출구점(Exit Point)를 가져야 한다.
동적 메모리 할당 금지	- 메모리 리소스 용량의 부족으로 발생하는 에러를 방지하기 위해서 동적 변수나 객체 사용을 금지한다.

- 코드 리뷰는 컴포넌트 구현 과정에서 적용할 수 있는 소스코드 검증 방법으로 ‘코드를 실행하지 않고 사람이 검토하는 과정을 통하여 코드에 숨어 있는 잠재적인 결함을 찾아내고 이를 개선하는 일련의 과정’으로 정의될 수 있다.

표 152 주요 코드 리뷰 기법

기 법	설 명
코드 인스펙션	- 가장 정형화된 패턴의 기법으로 전문화된 코드 리뷰 팀이 구현된 소스코드를 대상으로 일정한 패턴을 가지고 코드를 분석한다. - 코드 인스펙션은 계획, 오버뷰, 준비, 인스펙션, 재작업, 후처리 프로세스로 진행된다.
팀 리뷰	- 코드 인스펙션보다 덜 정형화된 기법으로 계획, 오버뷰, 준비 단계와 같은 사전 준비 단계를 생략한다. - 리뷰 시간에는 발표자(코드 작성자)가 코드에 대해 설명하고 팀원은 결함이나 개선안을 찾는다.
워크스루(Walkthrough)	- 가장 비정형적인 기법으로 발표자가 리뷰의 주제와 시간을 정해서 발표를 하고 동료로부터 의견을 수렴한다. 주로 사례에 대한 정보 공유나, 아이디어 수집을 위해서 사용될 수 있다. - 발표자만이 리뷰를 주관하고 발표하는 역할을 수행하며, 다른 참여자들은 아무런 역할과 책임을 가지지 않고 자유롭게 의견을 개진한다.
피어 리뷰	- 2~3명이 진행하는 코드 리뷰 기법으로 코드의 작성자가 모니터를 보면서 코드를 설명하고 다른 사람이 설명을 들으면서 의견을 주거나 결함을 발견한다.

○ 소스코드는 테스트가 시작되기 전에 형상관리 시스템에 등록하여 변경 이력을 관리해야 한다. (7.5.4.4)

2.2. 소프트웨어 컴포넌트 테스트

2.2.1. 소프트웨어 컴포넌트 테스트 절차

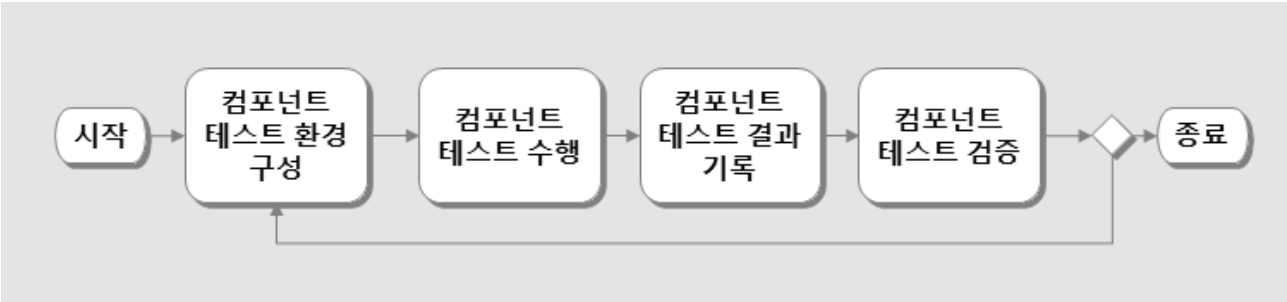


그림 118 소프트웨어 컴포넌트 테스트 흐름도

표 153 소프트웨어 컴포넌트 테스트 절차 설명

항 목	설 명
컴포넌트 테스트 환경 구성	- 소프트웨어 컴포넌트 테스트를 위한 환경을 점검하고 테스트 시나리오를 점검한다. - 테스트 케이스 정의 및 우선순위를 설정한다. - 테스트 데이터를 준비한다. - 테스트 환경 및 도구를 준비한다. - 구성된 컴포넌트 테스트 환경, 시나리오, 테스트 케이스, 테스트 데이터를 반영하여 ‘소프트웨어 컴포넌트 테스트 명세서’를 갱신한다.
컴포넌트 테스트 수행	- 정의된 테스트 케이스를 구현하고 도구에서 사용되는 테스트용 스크립트를 구현한다. - 테스트 환경에서 테스트 케이스를 검증한다. - 테스트 데이터를 사용하여 테스트 케이스를 실행하고 결과를 기록한다. - 테스트 과정에서 결함이 발견되면, 결함의 내용과 결함의 발생절차를 기록한다.
컴포넌트 테스트 결과 기록	- 테스트 결과와 결함 보고서를 기반으로 ‘소프트웨어 컴포넌트 테스트 보고서’를 작성한다.
컴포넌트 테스트 검증	- 계획된 컴포넌트 테스트가 빠짐없이 완료되었는지 여부를 확인한다. - ‘소프트웨어 테스트 명세서’와 ‘소프트웨어 컴포넌트 테스트 보고서’ 간의 추적성을 확인한다.

2.2.2. 소프트웨어 컴포넌트 테스트 지침

○ 컴포넌트 테스트 수행 환경은 실제 동작환경에서 수행되거나 최대한 근접한 환경하에서 수행되어야 한다. 실제 동작환경이 아닌 경우 소스코드와 적용된 오브젝트 코드의 차이, 테스트 환경과 실제 타겟 환경과의 차이를 분석하여 추후에 실시하는 통합 또는 시스템 테스트에서 추가적으로 수행할 테스트를 명세한다.

○ ‘소프트웨어 컴포넌트 테스트 보고서’는 기 작성된 보고서와 소소코드를 기반

으로 테스터의 책임 하에 작성되어야 하며, 보고서 작성 관련 요구사항들은 다음과 같다. (7.5.4.5)

- ‘소프트웨어 컴포넌트 테스트 보고서’는 테스트 보고서 작성에 필요한 일반 요구사항을 준수하여 작성되어야 한다. (7.5.4.6)
- 테스트 결과, 소프트웨어 컴포넌트 설계명세에 기술된 요구사항을 만족하는지 여부와 같은 항목들이 포함되어야 한다. (7.5.4.7)
- 각 컴포넌트별로 테스트 커버리지 지표가 제공되고 요구되는 커버리지 수준 대비 달성률이 기술되어야 한다. 할당된 SIL에 해당하는 테스트 커버리지 수준(테이블 A.21 참조)을 만족해야 한다. (7.5.4.7)
  - 테스트 커버리지는 소스코드에 대한 테스트 수행결과를 정량지표로 나타내는 방법으로 컴포넌트에 할당된 SIL 별로 권장하는 커버리지 유형과 설명은 다음과 같다.

표 154 소프트웨어 SIL별 컴포넌트 테스트 커버리지 기준 (예시)

소프트웨어SIL	테스트 커버리지 유형
SIL 0	- 구문(Statement) 커버리지 : 필수 - 분기(Branch) 커버리지 : 선택 - MC/DC(Modified Condition/Decision Coverage) 커버리지 : 선택
SIL 1	
SIL 2	
SIL 3	- 분기(Branch) 커버리지 : 필수 - MC/DC(Modified Condition/Decision Coverage) 커버리지 : 선택
SIL 4	

- ‘소프트웨어 컴포넌트 테스트 보고서’는 다음과 같은 ‘소프트웨어 테스트 보고서’ 일반 요구사항에 따라 적성되어야 한다.
  - 테스트 보고서는 테스터 이름 및 테스트 결과를 기술하고 테스트 목적과 테스트 기준이 준수되었는지를 기술해야 한다.
  - 테스트 실패 항목을 기술해야 한다.
  - 테스트 케이스와 결과는 후속 분석을 위해 프로그램이 읽을 수 있는 언어로 기록하는 것이 좋다.
  - 테스트는 반복 가능해야 하며, 실행 가능한 경우 자동화된 방식으로 수행되어야 한다.
  - 자동 테스트 실행을 위해 작성된 테스트 스크립트가 검증되어야 한다.
  - 관련된 모든 항목 (사용된 하드웨어 및 소프트웨어, 사용된 장비, 장비 교정 데이터, 소프트웨어의 버전 정보, 테스트 규격의 버전 정보)을 식별하여 형상 항

목을 기술해야 한다.

- 테스트 범위 및 테스트 완료에 대한 평가가 제공되어야 하고 수립된 계획과의 모든 편차가 기록되어야 한다.

○ ‘소프트웨어 컴포넌트 테스트 보고서’는 다음의 사항을 기술해야 한다.

- 테스트 결과를 기록하고 각 컴포넌트가 소프트웨어 컴포넌트 설계 명세의 요구사항을 준수해야 한다.
- 테스트 커버리지에 대한 설명이 모든 구성 요소에 대해 제공되어야 하며, 필요한 테스트 기준에 적합한 테스트 커버리지가 달성되었음을 보여야 한다.
- 컴포넌트에 할당된 SIL에 적합한 테스트 커버리지를 선정하고 기술해야 한다.

2.2.3. 소프트웨어 컴포넌트 테스트 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 차상 MMI 시스템의 사용자 어플리케이션에 대한 소프트웨어 컴포넌트 테스트 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘소프트웨어 요구사항 명세서’, ‘소프트웨어 아키텍처 설계 명세서’, ‘소프트웨어 컴포넌트 설계 명세서’, ‘소프트웨어 테스트 명세서

1.4 약어표

- 문서에 사용된 약어에 대한 설명을 한다.

2. 소프트웨어 통합 결과

2.1 통합 일정

- 소프트웨어 통합 일정을 기술한다.

2.2 통합 절차 및 내역

- 소프트웨어 통합 절차 및 내역을 기술한다.

2.3 통합 결과

- 소프트웨어 통합 결과를 기술한다.

3. 테스트 개요

- 테스트 개요를 기술한다.
- 예시) 테스트명, 목적, 범위, 장소, 일정, 조직, 담당자, 방법

4. 테스트 결과 요약

- 테스트 결과를 요약하여 기술한다.
- 예시) 총괄분석, 테스트 환경, 보완 요구사항 및 대책

5. 세부 테스트 결과

- 소프트웨어 컴포넌트 테스트 상세 결과를 기술한다.
- 예시) 컴포넌트별 기능, 인터페이스, 비기능 테스트 결과

6. 추적성

- 소프트웨어 테스트 명세와의 추적성을 기술한다.

A. 부록

그림 119 소프트웨어 컴포넌트 테스트 보고서 템플릿 (예시)

2.2.4. 소프트웨어 컴포넌트 테스트 보고서 체크리스트

표 155 소프트웨어 컴포넌트 테스트 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	테스트 결과에 합격/불합격 여부를 기술하고 있는가?
	테스트 구성(소프트웨어 버전, 사용된 하드웨어 및 소프트웨어 정보 포함)이 기술되어 있는가?
	'소프트웨어 컴포넌트 테스트 명세서'에 기술된 테스트 절차가 모두 수행되었는지 여부를 기술하고 있는가?
	모든 테스트 케이스가 수행되었는지 여부를 기술하고 있는가?
	테스트 결과에 테스트 담당자, 수행일시, 시험항목, 발견된 모든 문제들과 조치사항이 기술되어 있는가?
품질 요구사항	코딩규칙 기반의 정적 테스트 결과와 계획한 품질기준을 기술하고 있는가?
	런타임 오류 발생 위험을 확인하는 정적 테스트 결과와 계획한 품질기준을 기술하고 있는가?
	컴포넌트 소스코드를 대상으로 테스트 커버리지 측정 결과와 계획된 품질 기준을 기술하고 있는가?
(기타)	기타항목

2.3. 소프트웨어 컴포넌트 테스트 결과 평가

2.3.1. 소프트웨어 컴포넌트 테스트 결과 검토 절차

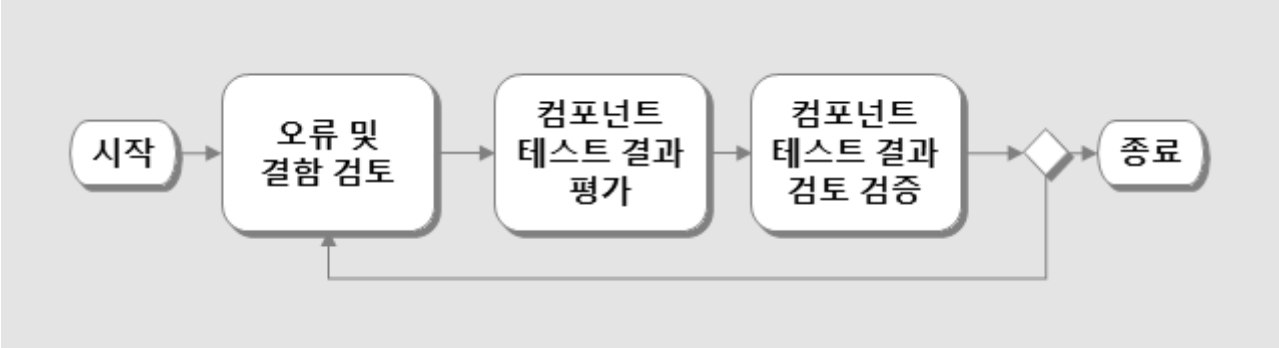


그림 120 소프트웨어 컴포넌트 테스트 결과 검토 흐름도

표 156 소프트웨어 컴포넌트 테스트 결과 검토 설명

항 목	설 명
오류 및 결함 검토	- 소프트웨어 컴포넌트 테스트에서 검출된 오류 및 결함을 검토할 계획을 수립한다. - 오류 및 결함이 발생한 소스코드를 리뷰하고 결과를 기술한다.
컴포넌트 테스트 결과 평가	- 테스트 결과를 반영하여 소프트웨어 소스코드 검증 보고서를 작성한다. - 컴포넌트 테스트 결과가 목표한 품질기준을 만족하는지 평가한다. - 후속 조치 계획을 수립한다.
컴포넌트 테스트 결과 평가 검증	- ‘소프트웨어 컴포넌트 테스트 보고서’를 검토한다. - ‘소프트웨어 소스코드 검증 보고서’를 검토한다. - ‘소프트웨어 검증 보고서’를 작성한다. - ‘소프트웨어 컴포넌트 테스트 명세서’와 ‘소프트웨어 컴포넌트 테스트 보고서’ 간의 추적성을 확인한다. - ‘소프트웨어 요구사항 명세서’와 ‘소프트웨어 컴포넌트 테스트 보고서’ 간의 추적성을 확인한다.

2.3.2. 소프트웨어 컴포넌트 테스트 결과 평가 지침

○ ‘소프트웨어 소스코드 검증 보고서’는 ‘소프트웨어 검증 보고서’ 작성에 필요한 일반 요구사항들을 준수하여 작성되어야 한다. (7.5.4.9)

- 검증 아이템의 특성 정보 및 구성, 검증자 이름
- 규격 불일치 아이템
- 불완전하게 적용된 컴포넌트, 데이터 구조 및 알고리즘
- 발견 오류 또는 결함
- 소프트웨어 검증 계획의 충족 또는 부족 여부

(검증 보고서에 편차가 있는 경우 편차가 중대한 것인지 아닌지를 기술)

· 가정 사항 (존재 시)

· 검증 결과 요약

○ 소스코드와 ‘소프트웨어 컴포넌트 테스트 보고서’ 작성 완료시 검증해야 할 항목들은 다음과 같다. (7.5.4.10)

- 소스코드가 ‘소프트웨어 컴포넌트 설계 명세서’에 기반 하에 적절하게 구현되었는지 여부
- 소스코드가 가독성 및 추적성 요구사항을 만족하는지 여부
- ‘소프트웨어 컴포넌트 테스트 명세서’에 기술된 테스트들의 수행 기록이 ‘소프트웨어 컴포넌트 테스트 보고서’에 기술되었는지 여부와 테스트 결과가 ‘소프트웨어 소스코드 검증 보고서’에 기록되었는지 여부

○ 소프트웨어 소스코드 구현과 ‘소프트웨어 컴포넌트 테스트 보고서’ 작성이 완료된 이후 ‘소프트웨어 소스코드 검증 보고서’는 다음과 같은 사항을 포함해서 기술되어야 한다.

- 소프트웨어 컴포넌트 설계 명세에 따라 소스코드가 적절하게 구현되었는지 여부
- SIL 등급에 따른 기법 및 대책의 적절한 선택 및 사용
- 코딩규칙을 준수하여 소스코드가 작성되었는지 여부
- 가독성 및 추적성에 대한 일반 요구사항과 소프트웨어 소스코드 구현과 관련된 특정 요구사항의 충족 여부
- 소프트웨어 컴포넌트 테스트 명세를 준수하여 테스트를 수행했는지 여부

2.3.3. 소프트웨어 소스코드 검증 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 ooo 시스템에 OOO 소프트웨어의 소스코드를 대상으로 검증한 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘소프트웨어 요구사항 명세서’, ‘소프트웨어 아키텍처 설계 명세서’, ‘소프트웨어 컴포넌트 설계 명세서’, ‘소프트웨어 컴포넌트 테스트 명세서’, ‘소프트웨어 컴포넌트 테스트 결과서’

1.4 약어표

- 약어에 대한 설명을 한다.
- 예시) MMI - Man Machine Interface

2. 수행 내역

- 검증 항목 리스트, 담당자, 수행 일시를 기술한다.

3. 상세 결과

- 컴포넌트 소스코드 검증 결과를 요약하여 기술한다.
- 예시) 컴포넌트 설계 명세의 적합성, 코딩표준 적용, 소스코드 요구사항 충족, SIL 등급에 따른 기법 및 대책 적용, ‘소프트웨어 컴포넌트 테스트 보고서’ 품질

4. 결과 정리

- 소프트웨어 소스코드 검증을 통과하기 위한 품질기준을 만족하는지 여부를 판정하여 검증결과를 요약하여 기술한다.

5. 추적성

- ‘소프트웨어 컴포넌트 설계 명세서’와 소스코드의 추적성을 기술한다.
- ‘소프트웨어 컴포넌트 테스트 명세서’와 ‘소프트웨어 컴포넌트 테스트 보고서’의 추적성을 기술한다.
- ‘소프트웨어 검증 명세서’와 ‘소프트웨어 검증 보고서’의 추적성을 기술한다.

A. 부록

그림 121 소프트웨어 소스코드 검증 보고서 템플릿 (예시)

2.3.4. 소프트웨어 소스코드 검증 보고서 체크리스트

표 157 소프트웨어 소스코드 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
일반 요구사항	검증자 이름, 검증 항목의 식별자과 소스코드 형상버전을 기술하고 있는가?
	구현된 소스코드에서 ‘소프트웨어 컴포넌트 설계 명세서’에 기술된 내용을 준수하지 않는 항목들을 기술하고 있는가?
	컴포넌트, 데이터, 자료구조, 알고리즘이 적당하지 않아 문제를 발생시키는 항목들을 기술하고 있는가?
	소스코드의 오류 또는 구현되지 않는 항목들을 기술하고 있는가?
	‘소프트웨어 컴포넌트 테스트 명세서’에 기술된 테스트 계획을 준수하고 있는가?
소스코드 품질 요구사항	소스코드 컴포넌트 설명에 컴포넌트명, 식별자, 기능 설명, 컴포넌트와 관련된 소프트웨어 요구사항과 컴포넌트 설계 참조, 개발자명, 작성날짜, 변경이력을 기술하고 있는가?
	소스코드가 코딩표준을 준수하고 있는지 여부를 기술하고 있는가?
	구현된 소스코드는 계획한 품질기준을 만족하고 있는지 여부를 기술하고 있는가?
	소프트웨어 컴포넌트 설계 시 결정된 사항들이 소스코드에서 식별할 수 있도록 되어 있는지 여부를 기술하고 있는가?
	소프트웨어 컴포넌트 설계 항목을 구현한 소스코드가 식별되어 추적할 수 있는가?
(기타)	기타항목

2.4. 소스코드 수정 및 테스트 케이스 보완

2.4.1. 소스코드 수정 및 테스트 케이스 절차

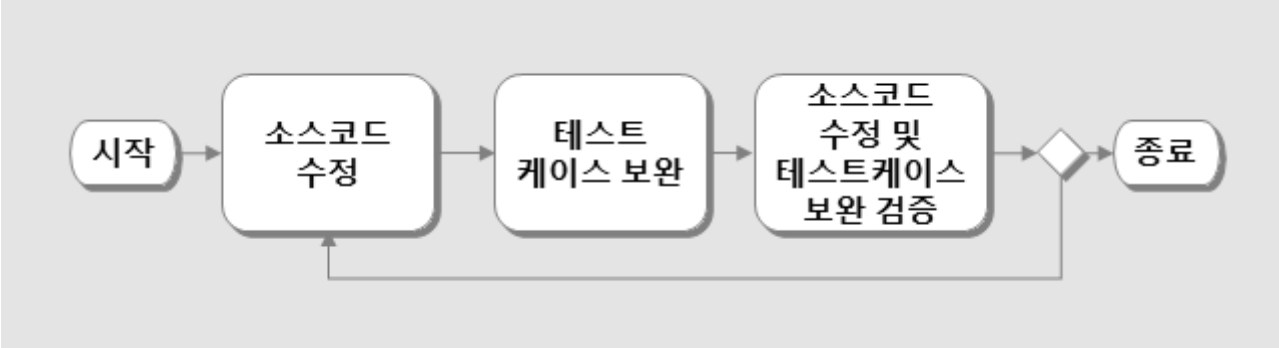


그림 122 소스코드 수정 및 테스트 케이스 보완 흐름도

표 158 소스코드 수정 및 테스트 케이스 보완 설명

항 목	설 명
소스코드 수정	- 테스트로 검출된 오류 및 결함에 대한 후속조치 계획에 따라 소스코드를 수정한다.
테스트 케이스 보완	- 수정된 소소코드를 반영하여 테스트 케이스를 보완한다.
소스코드 수정 및 테스트 케이스 보완 검증	- 소소코드 수정 및 테스트 케이스 보완 검증 기준을 만족하는지 평가한다. - 추적성(컴포넌트 소스코드와 결함보고서)을 확인한다.

2.4.2. 지침

○ 소프트웨어 소스코드 수정 및 테스트 케이스 보완 시 다음과 같은 사항을 고려해야 한다.

- 소스코드 변경 시 영향 분석을 수행하고 회귀 테스트 계획을 수립한다.
- 회귀 테스트는 “수정(Modification)이 기대하지 않은 결과를 발생하지 않는다는 것을 증명하기 위한 시스템이나 컴포넌트에 대한 선택적 재테스트”로 소스코드 수정되었을 때 의도치 않게 포함된 버그를 찾아낼 수 있도록 테스트를 수행하는 기법이다.
- 많은 비용이 필요한 회귀 테스트는 자동화 테스팅 도구를 활용하여 비용을 감소시키는 노력이 필요하다.

2.4.3. 소프트웨어 컴포넌트 테스트 명세서 템플릿

○ 본 문서 “소프트웨어 컴포넌트 테스트 명세서 템플릿” 참조

2.4.4. 소프트웨어 컴포넌트 테스트 명세서 체크리스트

○ 본 문서 “소프트웨어 컴포넌트 테스트 명세서 체크리스트” 참조

2.5. 컴포넌트 구현 및 테스팅 검증

2.5.1. 컴포넌트 구현 및 테스팅 검증 절차

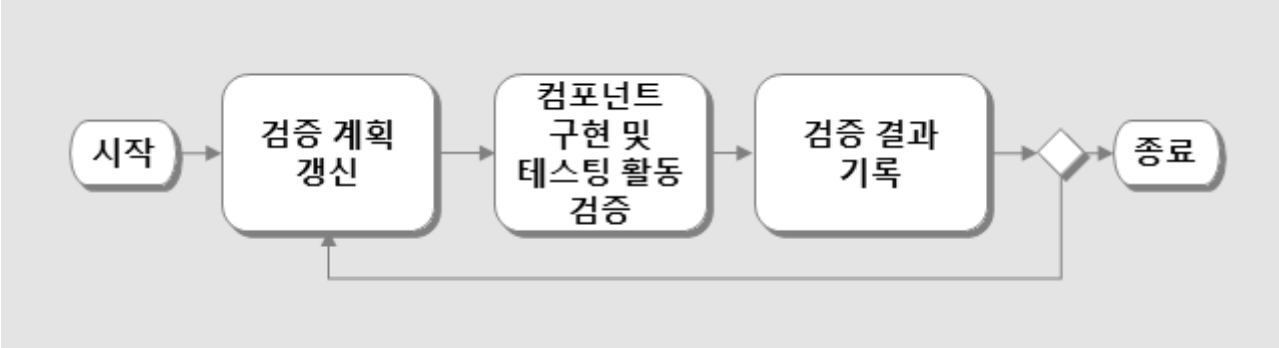


그림 123 컴포넌트 구현 및 테스팅 검증 흐름도

표 159 컴포넌트 구현 및 테스팅 검증 설명

항 목	설 명
검증 계획 갱신	- 소프트웨어 컴포넌트 구현 및 테스팅 단계의 검증 계획을 갱신한다.
컴포넌트 구현 및 테스팅 활동 검증	- 소프트웨어 테스트 보고서, 소프트웨어 소스코드 검증 보고서를 체크리스트를 활용하여 검증한다.
검증 결과 기록	- ‘소프트웨어 검증 명세서’에 기술된 검증 계획 및 절차에 따라 수행된 검증 결과를 ‘소프트웨어 검증 보고서’에 기술한다.

2.5.2. 컴포넌트 구현 및 테스팅 검증 지침

○ ‘소프트웨어 검증 보고서’는 컴포넌트 구현 및 테스팅 단계의 산출물을 대상으로 검증자(Verifier)의 책임 하에 작성되어야 한다.

2.5.3. 소프트웨어 검증 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 ooo 시스템에 OOO 소프트웨어의 컴포넌트 구현 및 테스팅 활동이 올바르게 수행되었는지 여부를 검증한 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.
- 예시) ‘소프트웨어 컴포넌트 설계 명세서’, ‘소프트웨어 컴포넌트 테스트 명세서’, ‘소프트웨어 컴포넌트 테스트 결과서’, ‘소프트웨어 소스코드 검증 보고서’

1.4 약어표

- 약어에 대한 설명을 한다.
- 예시) MMI - Man Machine Interface

2. 수행 내역

- 검증 항목 리스트, 담당자, 수행 일시를 기술한다.

3. 검증 결과의 개요

- 소프트웨어 검증 결과를 요약하여 기술한다.

4. 세부 검증 결과

- 컴포넌트 구현 및 테스팅 단계에서 발견된 문제점, 제약사항들을 기술하고 개선사항과 영향을 기술한다.

5. 추적성

- ‘소프트웨어 검증 계획서’와의 추적성을 기술한다.

A. 부록

그림 124 소프트웨어 검증 보고서 템플릿 (예시)

2.5.4. 소프트웨어 검증 보고서 체크리스트

표 160 소프트웨어 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
컴포넌트 구현 및 테스팅 단계 일반 요구사항	‘소프트웨어 컴포넌트 테스트 보고서’, ‘소프트웨어 소스코드 검증 보고서’를 체크리스트를 활용하여 검증한 결과를 기술하였는가?
	검증 결과에 종합적인 분석과 검증자, 수행일시를 포함하여 기술하였는가?
	검증 결과에 확인된 문제들과 적용된 조치사항을 기술하였는가?
(기타)	기타항목

제 5 절 통합

1. 개요

소프트웨어 컴포넌트 통합, 소프트웨어와 하드웨어 통합을 위해 필요한 정보를 기술한다.

1.1. 목표

○ 소프트웨어 컴포넌트 통합, 소프트웨어와 하드웨어 통합 수행

○ 소프트웨어와 하드웨어가 상호 작용하여 의도한 기능들이 올바르게 동작한다는 입증

1.2. 범위

○ 소프트웨어 개발 생명주기에서 IEC 62279 7.6에 해당하는 통합 단계에 대해 설명한다.

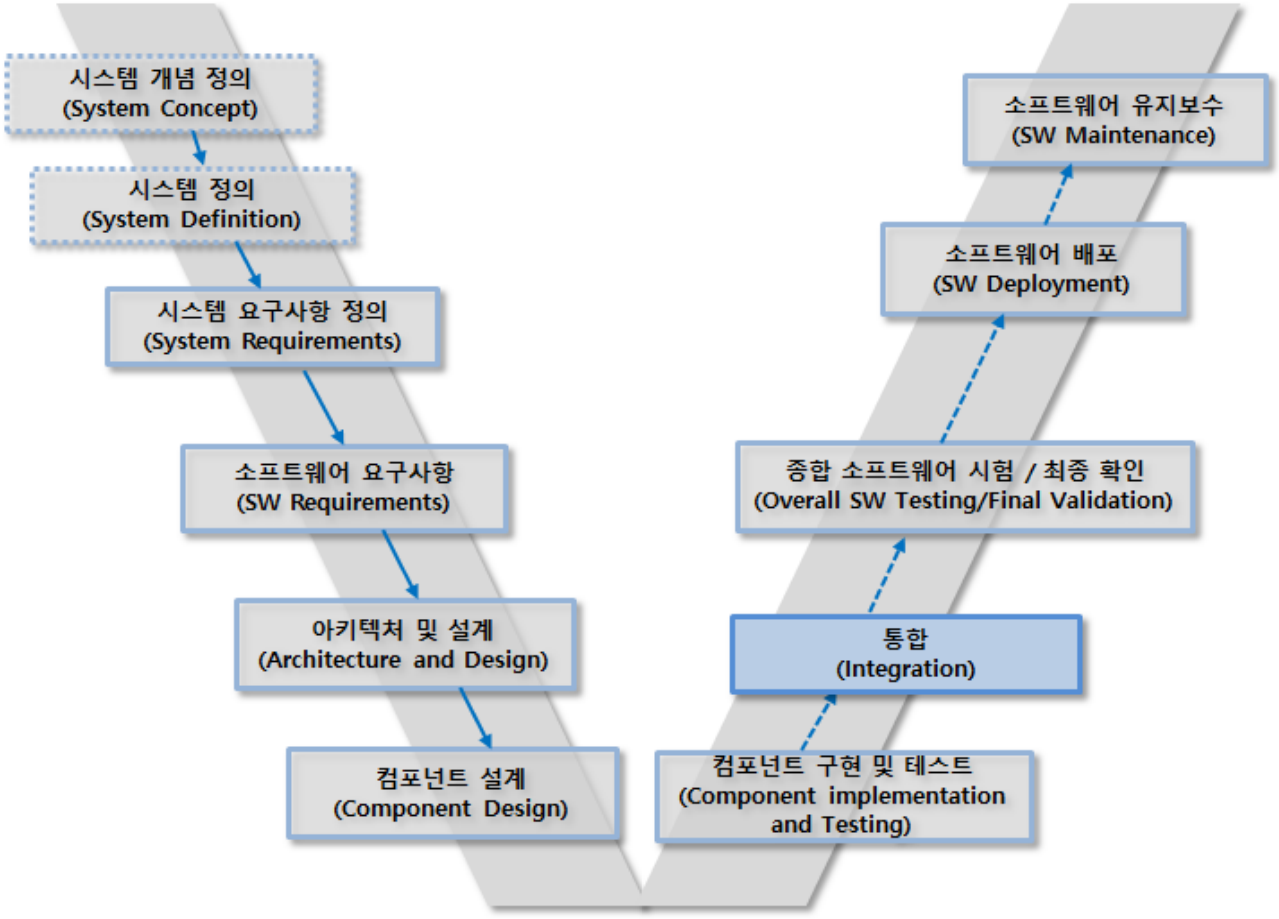


그림 125 소프트웨어 개발 생명주기 - 통합 단계

1.3. 시작 기준

- ‘소프트웨어/하드웨어 통합 테스트 명세서’ 작성 완료
- ‘소프트웨어 통합 테스트 명세서’ 작성 완료

1.4. 완료 기준

- ‘소프트웨어 통합 테스트 보고서’ 작성 완료
- ‘소프트웨어/하드웨어 통합 테스트 보고서’ 작성 완료
- ‘소프트웨어 통합 검증 보고서’ 작성 완료

1.5. 입력물

- 소프트웨어/하드웨어 통합 테스트 명세서
- 소프트웨어 통합 테스트 명세서

1.6. 출력물

표 161 통합 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 통합 테스트 결과서	HR	HR	HR	HR	HR
소프트웨어/하드웨어 통합 테스트 결과서	HR	HR	HR	HR	HR
소프트웨어 통합 검증 보고서	HR	HR	HR	HR	HR

1.7. 역할 및 책임

표 162 통합 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
통합	21. 소프트웨어 통합 테스트 결과서	INT	VER	VAL
	22. 소프트웨어/하드웨어 통합 테스트 결과서	INT	VER	VAL
	23. 소프트웨어 통합 검증 보고서	VER		

INT (Integrator) 통합자
VER (Verifier) 검증자
VAL (Validator) 확인자

1.8. 통합 주요 활동

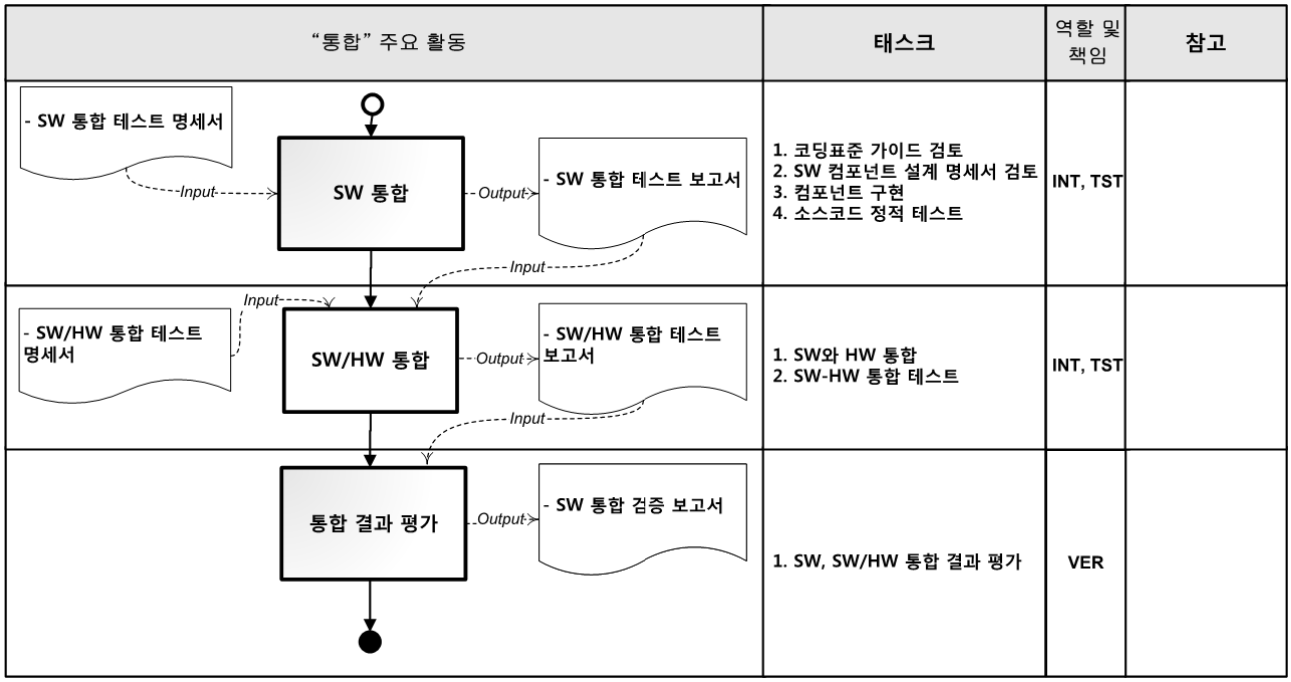


그림 126 통합 단계 주요 활동

표 163 통합 단계 주요 활동 설명

활 동 ID	활 동 명	설 명
INT.01	소프트웨어 통합	• 소프트웨어 컴포넌트 통합 계획에 따라 컴포넌트를 통합 • 소프트웨어 통합 테스트 • 소프트웨어 통합 테스트 보고서 작성
INT.02	소프트웨어/하드웨어 통합	• 소프트웨어/하드웨어 통합 계획에 따라 소프트웨어를 하드웨어에 통합 • 소프트웨어/하드웨어 통합 테스트 • 소프트웨어/하드웨어 통합 테스트 보고서 작성
INT.03	통합 결과 평가	• 통합(소프트웨어, 소프트웨어/하드웨어) 결과를 평가하여 소프트웨어 통합 검증 보고서 작성

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 소프트웨어 통합

2.1.1. 소프트웨어 통합 절차

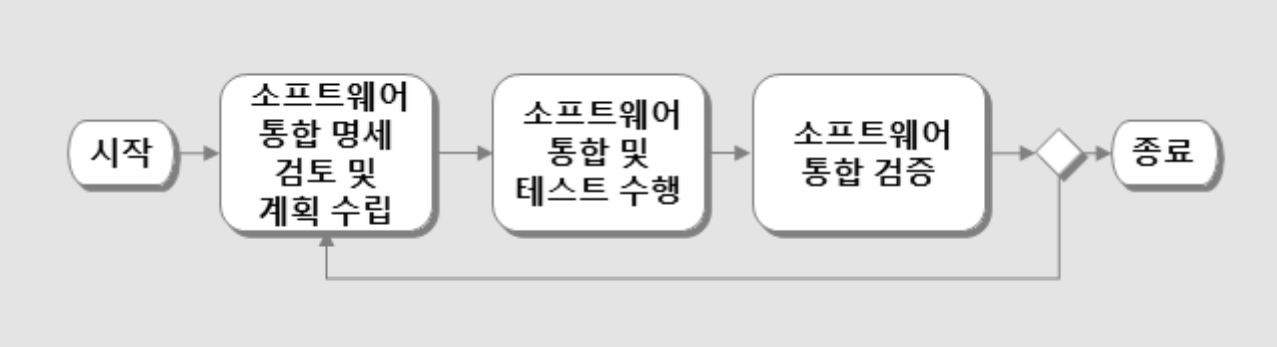


그림 127 소프트웨어 통합 흐름도

표 164 소프트웨어 통합 절차 설명

항 목	설 명
소프트웨어 통합 명세 검토 및 계획 수립	- 소프트웨어 통합 테스트 명세서의 내용을 검토하여 수정 및 추가가 필요한 부분을 갱신 또는 추가한다. - 소프트웨어 통합 테스트를 위한 테스트 명세를 작성한다.
소프트웨어 통합 및 테스트 수행	- 소프트웨어 통합 순서에 따라 통합 작업을 수행한다. - 작성된 소프트웨어 통합 테스트 명세에 따라 테스트를 실행한다. - 소프트웨어 통합 테스트 결과를 작성한다.
소프트웨어 통합 검증	- 작성된 소프트웨어 통합 테스트 결과서를 검토한다.

2.1.2. 소프트웨어 통합 지침

○ 소프트웨어 컴포넌트 통합은 시스템 통합 테스트, 시스템 테스트를 수행하기 위해 컴포넌트 인터페이스, 결합 소프트웨어의 적절한 복합 구성으로 개별 또는 기 테스트된 컴포넌트를 포함하여 점진적으로 수행되어야 한다. (7.6.4.1)

표 165 점진적 소프트웨어 통합 방법

구 분	장 점	단 점	설 명
백본(Backbone)	- 결함 격리 쉬움 - 위험이 높은 순으로 통합 결함	- 테스트 시간이 오래 걸림	- 가장 중요하고 위험(Risk)이 높은 모듈을 우선으로 초기 백본을 형성하고 나머지 모듈들을 통합하는 방식

구 분	장 점	단 점	설 명
	발견		
상향식(Bottom up)	- 결함 격리 쉬움 - 하위 모듈을 충분히 테스트할 수 있음	- 큰 문제(설계상 결함)을 상위에서 늦게 발견할 수 있음 - 비즈니스 로직 반영이 어려움	- 가장 하위의 모듈부터 통합해서 상위 모듈 통합으로 올라가는 방식
하향식(Top down)	- 결함 격리 쉬움 - 설계상의 결함을 조기에 발견할 수 있음	- 수정이 어려운 큰 문제를 하부에서 발견할 수 있음	- 가장 상위의 모듈부터 통합해서 하위의 모듈 통합으로 내려가는 방식

○ 소프트웨어와 하드웨어를 통합하는 중에 수정 또는 변경이 발생하면 시스템 영향분석을 실시하여 관련된 모든 컴포넌트를 식별하고 필요한 재검증 활동들을 수행하여야 한다. (7.6.4.2)

○ 소프트웨어 통합 테스트 보고서는 소프트웨어 통합 테스트 명세서를 기반으로 통합자(Integrator)의 책임 하에 작성되어야 한다. (7.6.4.3)

○ 소프트웨어 통합 보고서는 테스트 보고서에 대한 일반 요구사항(6.1.4.5)들을 준수하여 작성되어야 한다. (7.6.4.4)

○ 소프트웨어 통합 테스트 보고서는 다음 항목들을 준수하여 작성되어야 한다. (7.6.4.5)

- 소프트웨어 통합 테스트 보고서는 소프트웨어 통합 테스트 명세서에 기술된 테스트 목표 및 기준에 부합하는지 여부를 테스트 결과로 작성되어야 한다. 만약 테스트 결과에 실패(Failure) 건이 있다면 실패한 테스트 정보가 기록되어야 한다.
- 소프트웨어 통합 테스트 케이스와 결과들은 후속 분석을 위해 프로그램이 읽을 수 있는 형식으로 기록되어야 한다.
- 테스트는 반복적으로 수행되어야 하며 가능하다면 테스트 수행이 자동화 되어야 한다.
- 소프트웨어 통합 테스트 보고서는 식별자와 테스트 관련 모든 항목들을 포함한 환경이 기술되어야 한다.

○ 소프트웨어 통합 테스트 보고서는 [표* A.6]의 기능 및 블랙박스 테스팅(Functional and Black-box Testing), 성능 테스팅(Performance Testing) T&M을 올바르게 선택하여 사용하였는지 여부를 증명해야 한다. (7.6.4.6)

표 166 적용 T&M 예시 (SIL 2)

T & M		설 명
Functional and Black-box Testing	경계값 분석 (Boundary Value Analysis)	- 프로그램 입력 도메인의 경계영역에 초점을 맞춘 블랙박스 테스팅 기법 - 상세 내용은 “T&M-경계값 분석” 참조
	동등 분할 테스팅 (Equivalence Classes and Input Partition Testing)	- 전체 테스트 케이스 개수를 최소화하기 위해 각 테스트 케이스가 가능한 많은 다양한 입력 조건을 갖게 하는 블랙박스 기법 - 상세 내용은 “T&M-동등분할 테스팅” 참조

2.1.3. 소프트웨어 통합 테스트 결과서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 XXX 시스템의 어플리케이션에 대한 소프트웨어 모듈의 통합 테스트 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.

1.4 약어표

- 문서에 사용된 약어에 대한 설명을 한다.

2. 소프트웨어 통합 테스트 계획

2.1 입력 및 출력 산출물

- 소프트웨어 모듈 통합 시 필요한 입출력 산출물을 기술한다.

2.2 소프트웨어 모듈 통합 절차 및 내역

- 소프트웨어 모듈 통합 방법, 절차, 세부 내역을 기술한다.

2.3 시험 환경

- 시험에 사용할 소프트웨어 도구, 설비를 기술한다.

3. 소프트웨어 통합 테스트 결과

3.1 통합 결과

- 소프트웨어 모듈의 통합 테스트 결과를 요약하여 기술한다.

3.2 소프트웨어 모듈 통합 시험 상세

- 시험 대상, 참조문서, 시험 책임자, 평가기준, 시험 항목 및 결과, 테스트 커버리지를 기술한다.

4. 추적성

- 소프트웨어 통합 테스트 명세와의 추적성을 기술한다.

A. 부록

그림 128 소프트웨어 모듈 통합 테스트 결과서 템플릿 (예시)

2.1.4. 소프트웨어 통합 테스트 보고서 체크리스트

표 167 소프트웨어 통합 테스트 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
소프트웨어 모듈 통합 테스트	소프트웨어 모듈 통합 테스트 결과에 합격/불합격 여부를 기술하였는가?
	테스트 결과에 소프트웨어 모듈 통합 테스트 환경 구성 정보를 기술하였는가?
	모든 소프트웨어 모듈 통합 테스트 절차와 테스트 케이스가 수행되었는지 여부를 기술하였는가?
	소프트웨어 모듈 통합 테스트 결과에 수행일시, 테스트 식별자(ID)를 빠짐없이 기술하였는가?
	소프트웨어 모듈 통합 테스트 결과에서 확인된 문제에 대한 분석과 적용된 조치 결과를 기술하였는가?
(기타)	기타항목

2.2. 소프트웨어/하드웨어 통합

2.2.1. 소프트웨어/하드웨어 통합 절차

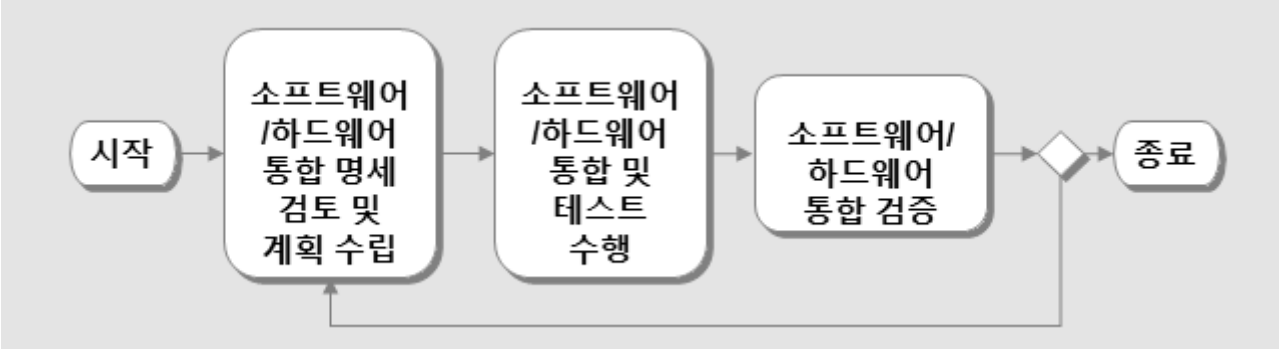


그림 129 소프트웨어/하드웨어 통합 흐름도

표 168 소프트웨어/하드웨어 통합 절차 설명

항 목	설 명
소프트웨어/하드웨어 통합 명세 검토 및 계획 수립	- 소프트웨어/하드웨어 통합 테스트 명세서의 내용을 검토하여 수정 및 추가가 필요한 부분을 갱신 또는 추가한다. - 소프트웨어/하드웨어 통합 테스트를 위한 테스트 명세를 작성한다.
소프트웨어/하드웨어 통합 및 테스트 수행	- 소프트웨어와 하드웨어 통합 순서에 따라 통합 작업을 수행한다. - 작성된 소프트웨어/하드웨어 통합 테스트 명세에 따라 테스트를 실행한다. - 소프트웨어/하드웨어 통합 테스트 결과를 작성한다.
소프트웨어/하드웨어 통합 검증	- 작성된 소프트웨어/하드웨어 통합 테스트 결과서를 검토한다.

2.2.2. 소프트웨어/하드웨어 통합 지침

○ 소프트웨어/하드웨어 통합 테스트 보고서는 소프트웨어/하드웨어 통합 테스트 명세서를 기반으로 통합자(Integrator)의 책임 하에 작성되어야 한다. 7.6.4.8 ~ 7.6.4.10 절의 요구사항들은 소프트웨어/하드웨어 통합 테스트 보고서에 반영되어야 한다. (7.6.4.7)

○ 소프트웨어/하드웨어 통합 테스트 보고서는 테스트 보고서에 대한 일반 요구사항들(6.1.4.5)을 준수하여 작성되어야 한다. (7.6.4.8)

○ 소프트웨어/하드웨어 통합 테스트 보고서는 다음 항목들을 준수하여 작성되어야 한다. (7.6.4.9)

- 소프트웨어/하드웨어 통합 테스트 보고서는 소프트웨어 통합 테스트 명세서에 기술된 테스트 목표 및 기준에 부합하는지 여부를 테스트 결과로 작성되어야

한다. 만약 테스트 결과에 실패(Failure) 건이 있다면 실패한 테스트 정보가 기록되어야 한다.

- 소프트웨어/하드웨어 통합 테스트 케이스와 결과들은 후속 분석을 위해 프로그램이 읽을 수 있는 형식으로 기록되어야 한다.
- 소프트웨어/하드웨어 통합 테스트 보고서는 식별자와 테스트 관련 모든 항목들을 포함한 환경이 기술되어야 한다.

○ 소프트웨어/하드웨어 통합 테스트 보고서는 [표 209]의 기능 및 블랙박스 테스팅(Functional and Black-box Testing), 성능 테스팅(Performance Testing) T&M을 올바르게 선택하여 사용하였는지 여부를 증명한다. (7.6.4.10)

2.2.3. 소프트웨어/하드웨어 통합 테스트 결과서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 XXX 시스템의 어플리케이션에 대한 소프트웨어와 하드웨어 통합 테스트 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.

1.4 약어표

- 문서에 사용된 약어에 대한 설명을 한다.

2. 소프트웨어/하드웨어 통합 테스트 계획

2.1 입력 및 출력 산출물

- 소프트웨어와 하드웨어 통합 시 필요한 입출력 산출물을 기술한다.

2.2 소프트웨어/하드웨어 통합 절차 및 내역

- 소프트웨어와 하드웨어 통합 방법, 절차, 세부 내역을 기술한다.

2.3 시험 환경

- 시험에 사용할 소프트웨어 도구, 설비를 기술한다.

3. 소프트웨어/하드웨어 통합 테스트 결과

3.1 통합 결과

- 소프트웨어/하드웨어 모듈의 통합 테스트 결과를 요약하여 기술한다.

3.2 소프트웨어/하드웨어 통합 시험 상세

- 시험 대상, 참조문서, 시험 책임자, 평가기준, 시험 항목 및 결과, 테스트 커버리지를 기술한다.

4. 추적성

- 소프트웨어 통합 테스트 명세와의 추적성을 기술한다.

A. 부록

그림 130 소프트웨어/하드웨어 통합 테스트 결과서 템플릿 (예시)

2.2.4. 소프트웨어/하드웨어 통합 테스트 결과서 체크리스트

표 169 소프트웨어/하드웨어 통합 테스트 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
소프트웨어/하드웨어 통합 테스트	소프트웨어/하드웨어 통합 테스트 결과에 합격/불합격 여부를 기술하였는가?
	테스트 결과에 소프트웨어/하드웨어 통합 테스트 환경 구성 정보를 기술하였는가?
	모든 소프트웨어와 하드웨어 통합 테스트 절차와 테스트 케이스가 수행되었는지 여부를 기술하였는가?
	소프트웨어/하드웨어 통합 테스트 결과에 수행일시, 테스트 식별자(ID)를 빠짐없이 기술하였는가?
	소프트웨어/하드웨어 통합 테스트 결과에서 확인된 문제에 대한 분석과 적용된 조치 결과를 기술하였는가?
(기타)	기타항목

2.3. 통합 검증

2.3.1. 통합 검증 절차

표 170 통합 검증 설명

항 목	설 명
검증 계획 갱신	- 통합(소프트웨어, 소프트웨어/하드웨어) 단계의 검증 계획을 갱신한다.
통합 활동 검증	- 소프트웨어 통합 테스트 보고서, 소프트웨어/하드웨어 통합 테스트 보고서를 체크리스트를 활용하여 검증한다.
검증 결과 기록	- ‘소프트웨어 통합 검증 명세서’에 기술된 검증 계획 및 절차에 따라 수행된 검증 결과를 ‘소프트웨어 통합 검증 보고서’에 기술한다.

2.3.2. 통합 검증 지침

○ 소프트웨어 통합 검증 보고서는 소프트웨어, 소프트웨어/하드웨어 통합 테스트 명세서 관련 테스트 보고서를 기반으로 검증자(Verifier)의 책임 하에 작성되어야 한다. 7.6.4.12 ~ 7.6.4.13 절의 요구사항들은 소프트웨어 통합 검증 보고서에 반영되어야 한다. (7.6.4.11)

○ 소프트웨어 통합 검증 보고서는 검증 보고서(Verification Report)에 대한 일반 요구사항들(6.1.4.14)을 준수하여 작성되어야 한다. (7.6.4.12)

○ 소프트웨어 통합 테스트 보고서와 소프트웨어/하드웨어 통합 테스트 보고서가 작성되면, 다음 항목들을 검증해야 한다. (7.6.4.13)

- 소프트웨어 통합 테스트 명세서에 따라 테스트 수행기록이 소프트웨어 통합 테스트 보고서에 적절하게 기록되었는지 여부
- 소프트웨어 통합 테스트 보고서가 가독성 및 추적성에 대한 요구사항들 (5.3.2.7 ~ 5.3.2.10, 6.5.4.14 ~ 6.5.4.17, 7.6.4.3 ~ 7.6.4.6)을 만족하는지 여부
- 소프트웨어/하드웨어 통합 테스트 명세서에 따라 테스트 수행기록이 소프트웨어/하드웨어 통합 테스트 보고서에 적절하게 기록되었는지 여부
- 소프트웨어/하드웨어 통합 테스트 보고서가 가독성 및 추적성에 대한 요구사항들(5.3.2.7 ~ 5.3.2.10, 6.5.4.14 ~ 6.5.4.17, 7.6.4.7 ~ 7.6.4.10)을 만족하는지 여부

2.3.3. 소프트웨어 통합 검증 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 ooo 시스템에 OOO 소프트웨어의 통합 활동이 올바르게 수행되었는지 여부를 검증한 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.

1.4 약어표

- 약어에 대한 설명을 한다.

2. 수행 내역

- 검증 항목 리스트, 담당자, 수행 일시를 기술한다.

3. 검증 결과의 개요

- 소프트웨어 통합 검증 결과를 요약하여 기술한다.

4. 세부 검증 결과

- 통합(소프트웨어 컴포넌트, 소프트웨어/하드웨어) 단계에서 발견된 문제점, 제약사항들을 기술하고 개선사항과 영향을 기술한다.

5. 추적성

- ‘소프트웨어 검증 계획서’와의 추적성을 기술한다.

A. 부록

그림 131 소프트웨어 통합 검증 보고서 템플릿 (예시)

2.3.4. 소프트웨어 통합 검증 보고서 체크리스트

표 171 소프트웨어 통합 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
통합 단계 일반 요구사항	‘소프트웨어 통합 보고서’, ‘소프트웨어/하드웨어 통합 보고서’를 체크리스트를 활용하여 검증한 결과를 기술하였는가?

구 분	점검 사항
	검증 결과에 종합적인 분석과 검증자, 수행일시를 포함하여 기술하였는가?
	검증 결과에 확인된 문제들과 적용된 조치사항을 기술하였는가?
(기타)	기타항목

제 6 절 종합 소프트웨어 테스팅/최종 확인

1. 개요

통합된 소프트웨어와 하드웨어가 소프트웨어 요구사항 명세서에 명시된 기능 및 안전 요구사항들을 만족하는지 여부를 확인하기 위해 분석 및 테스트를 수행하고 어플리케이션이 의도대로 만들어졌는지 여부를 확인한다.

1.1. 목표

○ 하드웨어에 통합된 소프트웨어의 요구사항 만족 여부 확인

1.2. 범위

○ 소프트웨어 개발 생명주기에서 IEC 62279 7.7에 해당하는 종합 소프트웨어 테스팅/최종 확인 단계에 대해 설명한다.

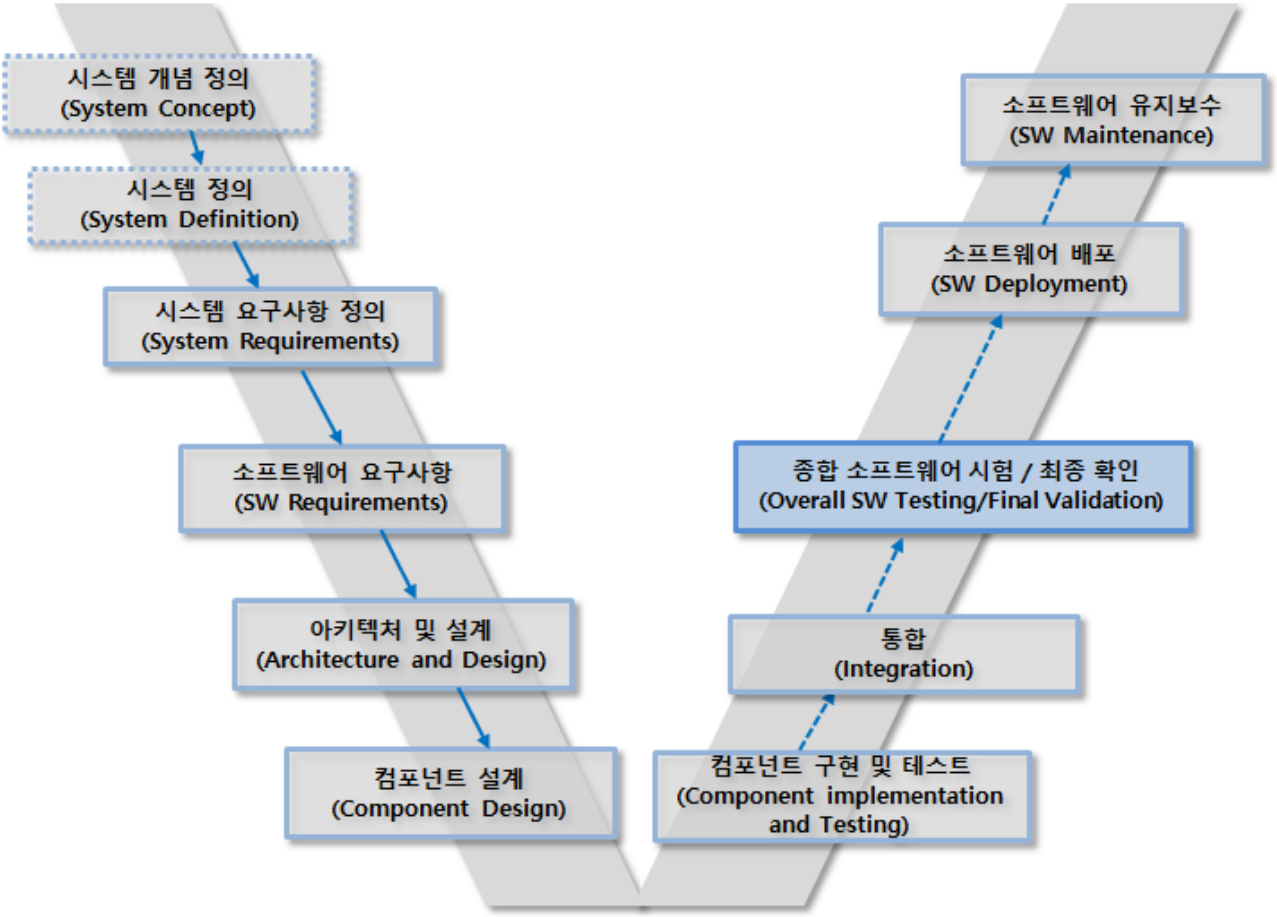


그림 132 소프트웨어 개발 생명주기 - 종합 소프트웨어 테스팅/최종 확인 단계

1.3. 시작 기준

- ‘소프트웨어/하드웨어 통합 테스트 결과서’ 작성 완료
- ‘소프트웨어 통합 테스트 결과서’ 작성 완료

1.4. 완료 기준

- ‘종합 소프트웨어 테스트 보고서’ 작성 완료
- ‘종합 소프트웨어 테스트 검증 보고서’ 작성 완료
- ‘소프트웨어 확인(Validation) 보고서’ 작성 완료
- ‘배포(Release) 노트’ 작성 완료

1.5. 입력물

- 소프트웨어 요구사항 명세서
- 종합 소프트웨어 테스트 명세서
- 소프트웨어 검증 계획서
- 소프트웨어 확인 계획서
- 기 검증 결과들을 포함한 모든 하드웨어 및 소프트웨어 문서
- 시스템 안전 요구사항 명세서

1.6. 출력물

표 172 종합 소프트웨어 시험/최종 확인 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
종합 소프트웨어 테스트 보고서	HR	HR	HR	HR	HR
종합 소프트웨어 테스트 검증 보고서	HR	HR	HR	HR	HR
소프트웨어 확인 보고서	HR	HR	HR	HR	HR
배포(Release) 노트	HR	HR	HR	HR	HR

1.7. 역할 및 책임

표 173 종합 소프트웨어 시험/최종 확인 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
종합 소프트웨어 시험 / 최종	24. 종합 소프트웨어 테스트 보고서	TST	VER	VAL
	25. 종합 소프트웨어 테스트 검증 보고서	VER		VAL

단 계	문 서	작 성 자	1차 검토	2차 검토
확인	26. 소프트웨어 확인 보고서	VAL	VER	
	27. 도구 검증 보고서	a	VER	
	28. 소프트웨어 품질보증 보고서	QAM	VER	
	29. 배포(Release)노트	CGM	VER	VAL

a 역할이 정의되어 있지 않음
TST (Tester) 테스터
VER (Verifier) 검증자
VAL (Validator) 확인자
CGM (Configuration Manager) 형상 관리자
QAM (Quality Assurance Manager) 품질보증 관리자

1.8. 종합 소프트웨어 시험/최종 확인 주요 활동

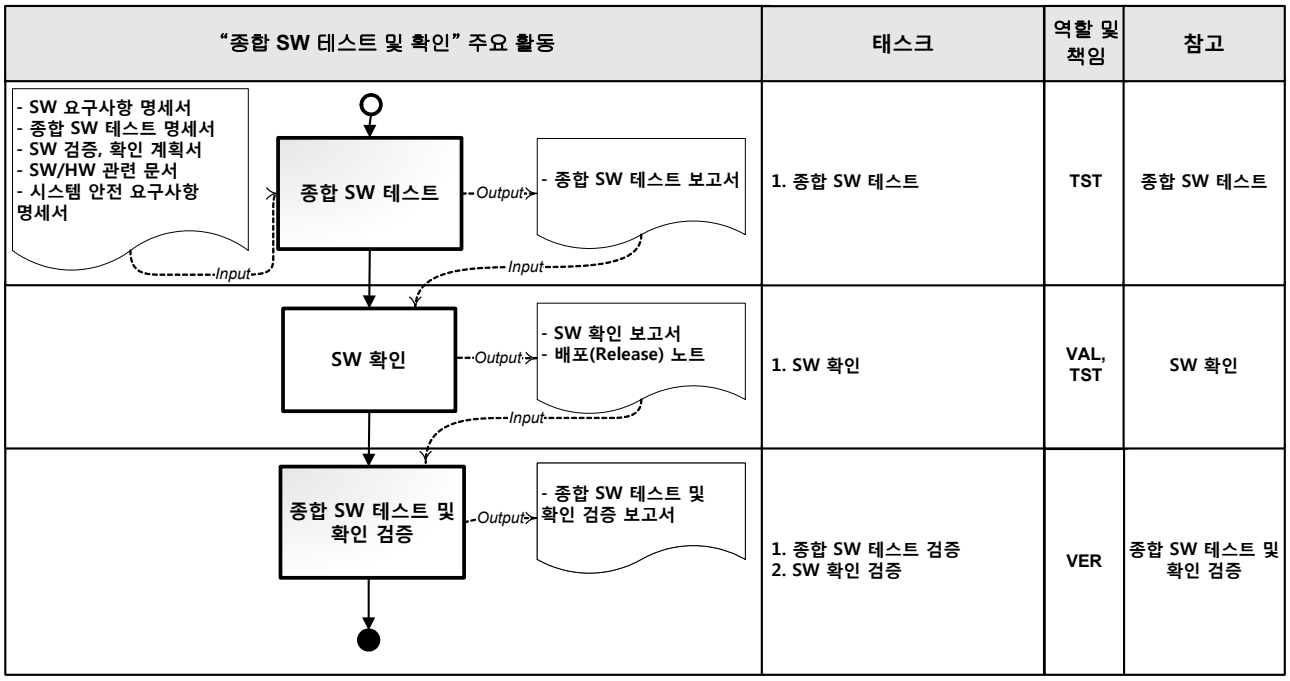


그림 133 종합 소프트웨어 테스트/확인 단계 주요 활동

표 174 종합 소프트웨어 테스팅/최종 확인 단계 주요 활동 설명

활 동 ID	활 동 명	설 명
V&V.01	종합 소프트웨어 테스트	• 개발된 소프트웨어를 타겟 시스템에 설치하고 소프트웨어 테스트를 수행한다. • 종합 소프트웨어 테스팅 결과를 평가하여 종합 소프트웨어 테스트 결과서를 작성한다.
V&V.02	소프트웨어 확인	• 개발된 소프트웨어가 요구사항에 부합하는지 여부를 확인한다. • 요구사항 추적표를 작성하여 요구사항과 테스트 결과 간의 추적성을 확인한다. • 발견한 문제점 및 제약사항들을 배포(Release) 노트에 기록하고 해결 방안을 검토한다.
V&V.03	종합 소프트웨어 테스트 및 확인 검증	• ‘소프트웨어 확인 보고서 체크리스트’를 기반으로 소프트웨어 확인 활동을 검증한다. • 부적합 사유가 있으면 이에 대한 권고사항과 해결책을 기술한다.

2. 세부 수행 활동

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

2.1. 종합 소프트웨어 테스트

2.1.1. 종합 소프트웨어 테스트 절차

그림 134 종합 소프트웨어 테스트 흐름도

표 175 종합 소프트웨어 테스트 절차 설명

항 목	설 명
종합 소프트웨어 테스트 준비	- 소프트웨어 테스트 전략 및 계획을 수립한다. - 회귀 테스트 전략을 수립한다. - 종합 소프트웨어 테스트 명세를 개발한다. 통합 검증 시 사용한 테스트 케이스를 재활용할 수 있다.
종합 소프트웨어 테스트 수행	- 종합 소프트웨어 시험환경을 구성한다. - 작성된 테스트 케이스로 소프트웨어 테스트를 수행한다. - 종합 소프트웨어 테스트 결과를 소프트웨어 테스트 보고서에 기술한다. - 회귀 테스트를 수행한다. (필요시)
종합 소프트웨어 테스트 검증	- 종합 소프트웨어 테스트 보고서 체크리스트를 활용하여 점검한다.

2.1.2. 종합 소프트웨어 테스팅 지침

○ 종합 소프트웨어 테스트 보고서는 종합 소프트웨어 테스트 명세서를 기반으로 테스터(Tester)의 책임 하에 작성되어야 한다. 종합 소프트웨어 테스트 명세서 작성 요구사항은 종합 소프트웨어 테스트 보고서에 반영되어야 한다. (7.7.4.1)

표 176 종합 소프트웨어 테스트 명세서 작성 요구사항 목록

관련 장절	설 명
7.7.4.2	- 테스트 보고서 작성에 필요한 일반 요건 준수
7.7.4.3	- 확인자(Validator)는 재량에 따라 특정 테스트를 지정하고 테스터에 의해 수행될 수 있다. - 테스트 기간 중에 테스트는 주로 소프트웨어 요구사항 명세에 기반을 두고 수행되며, 확인자는 사용자의 실제 요구사항을 반영한 복잡한 시나리오로 구성된 시스템 스트레스 테스트를 추가할 수 있다.
7.7.4.4	- 모든 테스트 결과와 분석 결과는 테스트 보고서에 기록되어야 한다.

- 종합 소프트웨어 테스트 보고서는 테스트 보고서를 작성하기 위한 요구사항 (6.1.4.5)를 준수하여 작성되어야 한다. (7.7.4.2)
- 확인자(Validator)는 재량에 따라 특정한 추가 테스트를 정의하고 테스트에 의해 수행되어야 합니다. 종합 소프트웨어 테스트는 주로 소프트웨어 요구사항 명세서의 구조를 기반으로 하지만, 시스템의 실제 사용자의 필요를 반영한 복잡한 시나리오로 시스템에 부하를 주는 테스트도 같이 수행되어야 한다. (7.7.4.3)
- 모든 테스트와 분석 결과들은 종합 소프트웨어 테스트 보고서에 기록되어야 한다. (7.7.4.4)
- 종합 소프트웨어 테스트 보고서는 검증자의 책임 하에 작성되어야 한다. (7.7.4.5)
- 소프트웨어는 실 하드웨어 연결, 운영 인터페이스로 실 시스템 연결, 입출력 신호의 시뮬레이션으로 테스트 되어야 한다. 시스템에 요구되는 일반 운영모드 및 비정상 조건들에 대해서도 테스트 되어야 한다. 시뮬레이션 입출력 데이터는 사용된 경우 실 입출력 데이터와 다르지 않음을 증명해야 한다. 시뮬레이션 입출력 데이터는 시스템 레벨 또는 장애 모드의 입출력 데이터를 대신하여 사용될 수 있다. (7.7.4.6)

2.2. 소프트웨어 확인

2.2.1. 소프트웨어 확인 절차

표 177 소프트웨어 확인 절차 설명

항 목	설 명
소프트웨어 확인 준비	- 소프트웨어 확인 전략 및 계획을 검토한다. - 소프트웨어 확인 계획서를 갱신한다. (필요시)
소프트웨어 확인 수행	- 확인 시 필요한 계획서, 요구사항명세서 등의 산출물을 준비한다. - 소프트웨어 확인 계획서에 따라 개발된 소프트웨어에 대한 평가를 수행한다.
소프트웨어 확인 검증	- 소프트웨어 확인 보고서를 소프트웨어 확인 보고서 체크리스트를 활용하여 점검한다.

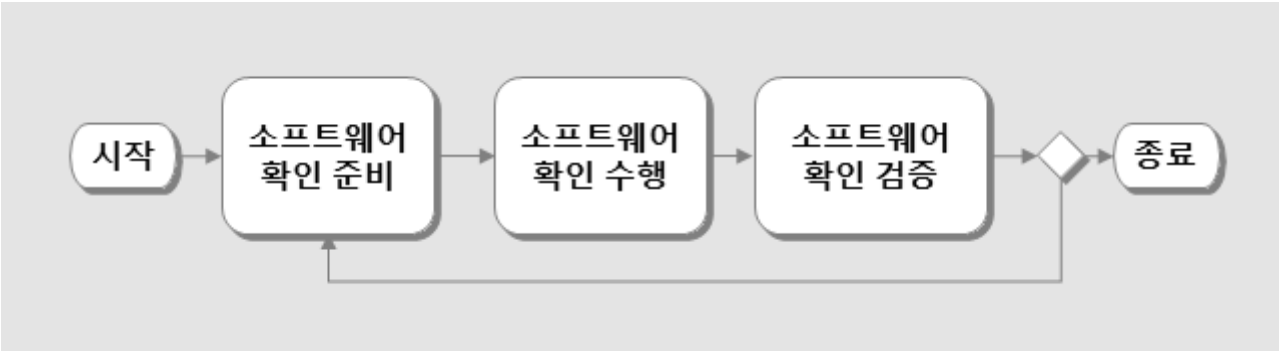


그림 135 소프트웨어 확인 흐름도

2.2.2. 소프트웨어 확인 지침

○ 소프트웨어 확인 보고서는 소프트웨어 확인 계획에 근거하여 확인자(Validator)의 책임 하에 작성되어야 한다. (7.7.4.7)

○ 소프트웨어 확인 보고서는 확인 보고서를 작성하기 위한 일반 요구사항을 준수하여 작성되어야 한다. (7.7.4.8)

○ 통합이 완료되고 종합 소프트웨어 테스팅 및 분석이 완료된 이후에 소프트웨어 확인 보고서는 다음 항목들을 포함하여 작성되어야 한다. (7.7.4.9)

- 소프트웨어 확인 계획서에 기술된 목표 및 기준의 만족하는지 여부와 계획과의 불일치 사항들의 기록 및 검토
- 테스트 결과의 요약 보고서와 타겟 시스템의 소프트웨어 요구사항 명세서의 요건들의 만족 여부
- 소프트웨어 요구사항 명세서의 요건들에 대한 테스트 커버리지 평가
- 소프트웨어 검증 계획에 따른 검증 활동들의 평가와 요구사항 추적성 점검 보

고서

- 확인자(Validator)가 테스트 케이스를 작성하여 테스터(Tester)에 전달하지 않았다면 소프트웨어 확인 보고서는 일반 요구사항을 준수하여 작성되었는지 여부

○ 소프트웨어 확인 보고서는 부속서 A에 따라 선택된 각 T&M 조합에 대해 검토한 내용을 포함해야 한다. 즉, 소프트웨어의 크기 및 복잡도 평가, 테스팅 결과, 검토 및 확인 활동들에 대한 평가를 위해 적용된 T&M에 대한 적절성에 대한 평가가 포함되어야 한다. (7.7.4.10)

○ 소프트웨어 확인 보고서에는 다음 항목들이 기술되어야 한다. (7.7.4.11)

- 소프트웨어 식별 및 구성에 대한 문서
- 지원 소프트웨어 및 장비에 대한 식별 정보
- 사용된 시뮬레이션 모델 정보
- 종합 소프트웨어 테스트 명세서의 적절성에 대한 정보
- 확인된 불일치 사항들에 대한 수집 및 기록
- 위험(영향)에 대한 모든 불일치 사항에 대한 검토 및 평가
- 프로젝트가 변경관리 프로세스 및 절차에 따라 수정 활동을 적절하게 수행했는지 여부, 발견된 불일치 사항의 식별 여부
- 추적 가능한 제약사항 기술
- 어플리케이션 조건 및 제약사항들을 고려하여 소프트웨어가 의도대로 만들어졌는지에 대한 결론

○ 발견된 결함 및 표준, 요구사항, 계획, 제약사항에 대한 모든 불일치 사항들은 소프트웨어 확인 보고서의 별도 장절에서 명확히 식별되어야 하며, 안전 무결성 레벨을 고려해 평가되어 소프트웨어와 함께 제공되는 배포(Release) 노트에 포함되어야 합니다. (7.7.4.12)

○ 소프트웨어와 함께 제공되는 배포(Release) 노트에는 소프트웨어 사용에 대한 모든 제약사항들이 포함되어야 한다. 제약사항들은 다음과 같다.

- 발견된 결함들
- 표준과의 불일치 사항
- 요구사항 만족도
- 계획의 이행정도

2.2.3. 종합 소프트웨어 테스트 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 ㅇㅇㅇ 시스템의 종합 소프트웨어 테스트 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.

1.4 약어표

- 문서에 사용된 약어에 대한 설명을 한다.

2. 종합 소프트웨어 테스트

2.1 종합 테스트 일정

- 종합 소프트웨어 테스트 일정을 기술한다.

2.2 종합 테스트 절차 및 내역

- 종합 소프트웨어 테스트 절차 및 내역을 기술한다.

2.3 종합 테스트 결과

- 종합 소프트웨어 테스트 결과를 기술한다.

3. 추적성

- 소프트웨어 테스트 명세서와 추적성을 기술한다.

A. 부록

그림 136 종합 소프트웨어 테스트 보고서 템플릿 (예시)

2.2.4. 종합 소프트웨어 테스트 보고서 체크리스트

표 178 종합 소프트웨어 테스트 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
종합 테스트	테스트 결과에 합격/불합격 여부를 기술하였는가?

구 분	점검 사항
	테스트 결과에 테스트 환경 구성 정보를 기술하였는가?
	모든 테스트 절차와 테스트 케이스가 수행되었는지 여부를 기술하였는가?
	테스트 결과에 수행일시, 테스트 식별자(ID)를 빠짐없이 기술하였는가?
	테스트 결과에서 확인된 문제에 대한 분석과 적용된 조치 결과를 기술하였는가?
(기타)	기타항목

2.2.5. 소프트웨어 확인 보고서 템플릿

1. 개요

1.1 문서의 목적

- 문서의 목적을 기술한다.
- 예시) 본 문서는 ㅇㅇㅇ 시스템의 소프트웨어 확인(Validation) 결과를 기술한다.

1.2 용어 정의

- 문서에서 사용되는 특정 용어를 설명한다.

1.3 참고 문서

- 참고 문헌을 나열한다.

1.4 약어표

- 문서에 사용된 약어에 대한 설명을 한다.

2. 소프트웨어 확인 결과

2.1 확인 일정

- 소프트웨어 확인 일정을 기술한다.

2.2 확인 절차 및 내역

- 소프트웨어 확인 절차 및 내역을 기술한다.

2.3 확인 결과

- 소프트웨어 확인 결과를 기술한다.

3. 추적성

- 소프트웨어 확인 계획과의 추적성을 기술한다.

A. 부록

그림 137 소프트웨어 확인 보고서 템플릿 (예시)

2.2.6. 소프트웨어 확인 보고서 체크리스트

표 179 소프트웨어 확인 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 장절, 표, 그림의 목차가 최신 버전으로 갱신되어 있는가?
	문서에 사용된 용어 및 약어가 정의되어 있는가?
	참고한 문서목록이 정확히 기술되어 있는가?
추적성 분석	시스템 요구사항과 소프트웨어 요구사항까지의 추적성 분석 결과가 기술되었는가?
	소프트웨어 요구사항과 요구사항을 검증하기 위한 테스트 절차까지의 추적성 분석 결과가 기술되었는가?
	소프트웨어 요구사항과 소프트웨어 설계 요구사항까지의 추적성 분석 결과가 기술되었는가?
	소프트웨어 설계 요구사항과 소프트웨어 컴포넌트 할당까지의 추적성 분석 결과가 기술되었는가?
	소프트웨어 컴포넌트 설계와 소프트웨어 컴포넌트 테스트 절차까지의 추적성 분석 결과가 기술되었는가?
	소프트웨어 컴포넌트 설계와 소스코드까지의 추적성 분석 결과가 기술되었는가?
요구사항 검증 분석	소프트웨어 요구사항이 시스템 요구사항을 만족하는지 여부를 확인하기 위한 분석 결과가 기술되었는가?
	소프트웨어 요구사항이 식별되었고 식별된 요구사항들이 검증가능한지 여부를 확인하기 위한 분석 결과가 기술되었는가?
요구사항 및 구조적 기반 테스트 커버리지 분석	소프트웨어 요구사항 기반의 테스트 커버리지 분석 결과가 기술되었는가?
	소프트웨어 테스트 결과를 기반으로 소프트웨어 설계상의 오류가 있는지 여부를 확인하는 분석 결과가 기술되었는가?
결함 및 불일치 분석	발견된 결함과 표준, 요구사항, 계획, 제약사항에 대한 모든 불일치 사항들이 별도의 장절로 구분되어 기술되었는가?
(기타)	기타항목

제 7 절 소프트웨어 배포

1. 개요

소프트웨어 배포 단계에서 프로젝트 매니저의 책임 하에 목표 버전 소프트웨어의 릴리스 및 배포를 계획, 통제 및 명세하여, 실제 환경에서의 소프트웨어 무결성을 확보할 수 있도록 적합한 형상항목을 승인하여 배포하는 절차에 대하여 기술한다.

1.1. 목표

소프트웨어가 응용 프로그램의 최종 환경에 배포되어 할당된 소프트웨어 안전 무결성 등급과 시스템 종속성을 유지하면서 소프트웨어 요구사항을 충실히 만족하도록 수행됨을 보장한다.

1.2. 범위

소프트웨어 개발 생명 주기에서 IEC 62279 9.1절에 해당하는 소프트웨어 배포 단계에 대해 설명한다.

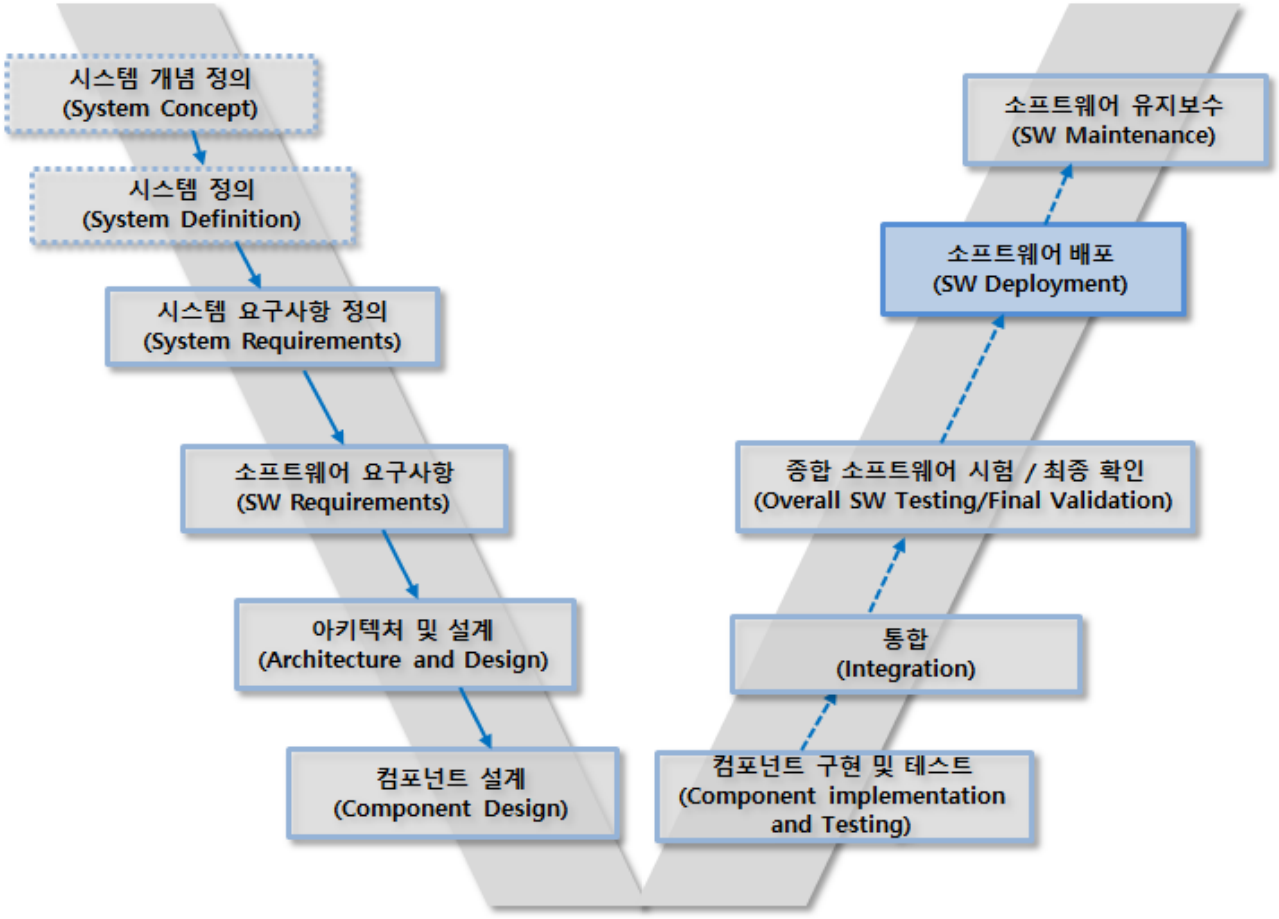


그림 138 소프트웨어 개발 생명주기 - 배포 단계

1.3. 시작 기준

- ○ 종합 소프트웨어 테스트 완료
- ○ 종합 소프트웨어 검증 완료
- ○ 소프트웨어 확인 완료

1.4. 완료 기준

- ○ 소프트웨어 릴리스 및 배포 계획 수립 완료
- ○ 소프트웨어 배포 매뉴얼 작성 완료
- ○ 소프트웨어 릴리스 완료
- ○ 소프트웨어 배포 완료
- ○ 소프트웨어 배포 검증 완료

1.5. 입력물

배포와 관련된 모든 디자인, 개발 및 분석 문서

1.6. 산출물

표 180 소프트웨어 배포 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 릴리스 및 배포 계획서	R	HR	HR	HR	HR
소프트웨어 배포 매뉴얼	R	HR	HR	HR	HR
릴리스 노트	HR	HR	HR	HR	HR
배포 기록	R	HR	HR	HR	HR
배포 검증 보고서	R	HR	HR	HR	HR

1.7. 역할 및 책임

표 181 소프트웨어 배포 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
소프트웨어 배포	38.소프트웨어 릴리스 및 배포 계획서	a	VER	VAL
	39.소프트웨어 배포 매뉴얼	a	VER	VAL
	40.릴리스 노트	DES	VER	VAL
	41.배포 기록	a	VER	VAL
	42.소프트웨어 배포 검증 보고서	VER		
a 역할이 정의되어 있지 않음 DES (Designer) 설계자 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 소프트웨어 배포 주요 활동

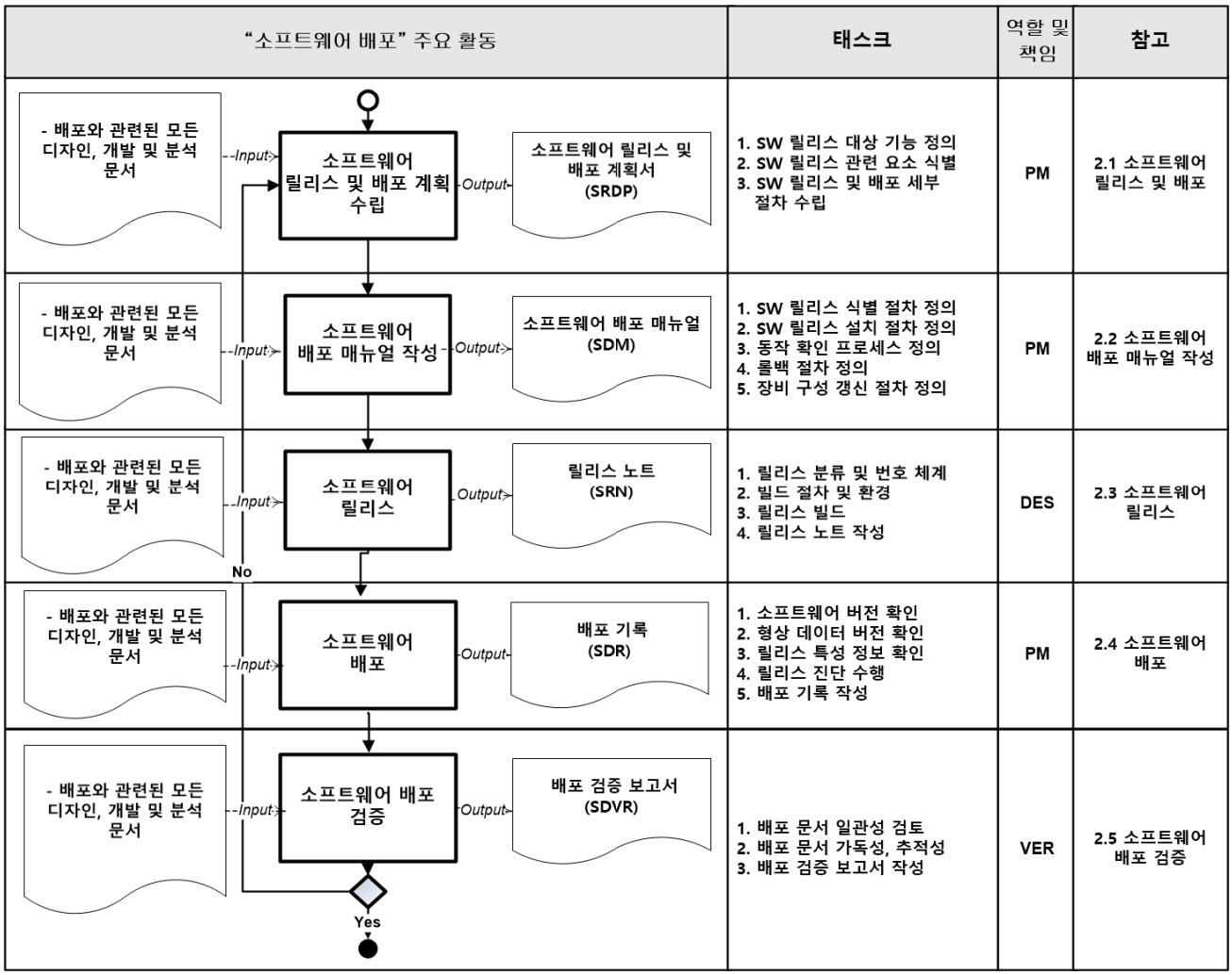


그림 139 소프트웨어 배포 주요 활동

표 182 소프트웨어 배포 단계

활 동 ID	활 동 명	설 명
DEP.01	소프트웨어 릴리스 및 배포 계획 수립	• 각 릴리스에 포함될 기능을 식별한다. • 릴리스 및 배포에 요구되는 관련 요소를 식별한다. • 릴리스 및 배포 세부 절차를 수립한다.
DEP.02	소프트웨어 배포 매뉴얼 작성	• 소프트웨어 릴리스 식별 절차를 정의한다. • 소프트웨어 릴리스 설치 절차를 정의한다. • 소프트웨어 릴리스 롤백 절차를 정의한다.
DEP.03	소프트웨어 릴리스	• 소프트웨어 릴리스 적용 조건을 정의한다. • 소프트웨어 릴리스 호환성 정보를 정의한다. • 소프트웨어 릴리스 제약 사항을 정의한다. • 소프트웨어 릴리스 베이스 라인을 정의한다.
DEP.04	소프트웨어 배포	• 소프트웨어 버전 정보를 기술한다. • 형상 데이터 버전 정보를 기술한다. • 소프트웨어의 특성 정보를 기술한다. • 소프트웨어 진단 정보를 기술한다.
DEP.05	소프트웨어 배포 검증	• 소프트웨어 배포의 일관성, 적합성을 검증한다. • 소프트웨어 배포의 가독성, 추적성을 검증한다. • 소프트웨어 배포를 검증하고 보고서를 작성한다.

2. 세부 수행 활동

소프트웨어 배포 단계에서는 새로운 버전으로 기존의 버전을 교체하는 것을 포함하며, 배포 절차를 정의하고 배포 관리 방법 및 세부 절차 수립, 배포 버전의 개략적인 프로세스 정의, 배포 결과 기록, 배포 검증을 수행한다. 또한 배포에 따른 새로운 형상을 기록하고 관리해야 하며, 운영 절차에는 새롭게 발생한 제약사항을 고려해야 하고, 이전 버전의 제약 조건 중에 변경된 부분은 수정하여 반영한다.

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

- 배포는 프로젝트 매니저의 책임 하에 수행된다. (9.1.4.1)
  ※ 릴리즈 및 배포 관련 프로젝트 매니저의 책임 (IEC 62279 표 B.9)
  - 이해관계자의 안전 요구사항이 충족되고 전달 되도록 보증한다.
  - 개발 과정에서 부분 또는 전체 안전 인도 물을 보증한다.
  - 소프트웨어 인도 및 배포를 책임진다.
- 소프트웨어 릴리스를 제공하기 전에 기존 소프트웨어 및 이 표준의 이전 버전에 따라 개발 된 소프트웨어를 포함하여 소프트웨어 베이스라인을 형상 관리의 통제 하에 기록하고 추적 가능하도록 관리한다. (9.1.4.2)
  ※ 릴리즈 및 배포 관련 형상관리자의 책임 (IEC 62279 표 B.10)
  - 소프트웨어 형상관리 계획을 수립한다.
  - 형상관리 시스템을 구축한다.
  - 형상 관리 시스템 내에서 모든 소프트웨어 컴포넌트를 명확하게 식별하고 독립적으로 버전 화하여 설정한다.
  - 호환되지 않는 버전의 소프트웨어 컴포넌트를 포함하여 릴리스 노트를 준비한다.
- 실행 코드 또는 데이터의 저장, 전송, 전달 또는 복제 중에 발생하는 오류를 예방하거나 탐지하기 위한 수단이 소프트웨어 패키지에 포함되어야 한다. 탑재 프로세스의 코드 무결성 검사의 일환으로 실행 코드를 코드화 [표 206] “오류 검출 코드” 참조)하는 것이 권고된다. (9.1.4.20)

2.1. 소프트웨어 릴리스 및 배포 계획

모든 소프트웨어 버전의 배포는 사전에 계획되고, 최종 승인되어야 한다. 소프트웨어 릴리스 및 배포 계획 단계에서는 소프트웨어 릴리스에 포함되어야 하는 기능을 정의하고, 릴리스에 포함되는 기능들을 정의하며, 배포 관리 방법 및 세부 절차를 수립한다.

2.1.1. 소프트웨어 릴리스 및 배포 계획 절차

○ 소프트웨어 릴리스 및 배포 계획 수행 흐름

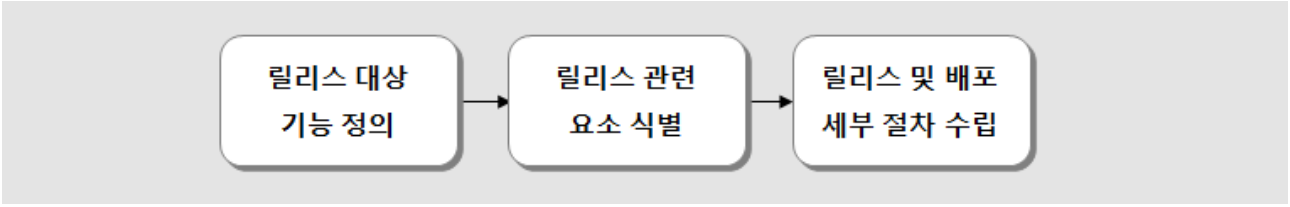


그림 140 소프트웨어 릴리스 및 배포 계획 수행 흐름도

○ 소프트웨어 릴리스 및 배포 계획 절차

표 183 소프트웨어 릴리스 및 배포 계획 절차

항 목	설 명
릴리스 대상 기능 정의	- 각 릴리스에 포함되어야 하는 기능을 정의한다. - 식별된 기능에 영향을 미치는 어플리케이션 파라미터가 어떤 릴리스와 연관되어 있는지 식별한다.
릴리스 관련 요소 식별	- 릴리스와 관련이 있는 요소들을 식별하여 정의한다. - 릴리스 요소에는 프로그래밍 도구들을 포함할 수 있다.
릴리스 및 배포 세부 절차 수립	- 릴리스의 유형, 서비스 수준 및 지원 기간을 식별한다. - 고객의 요구에 맞추어 소프트웨어 인도를 위한 매체 유형을 정의한다.

2.1.2. 소프트웨어 릴리스 및 배포 계획서 작성 지침

※ IEC 62279 표준에는 소프트웨어 릴리스 및 배포 계획에 대한 세부 요구사항이 기술되어 있지 않음

2.1.3. 소프트웨어 릴리스 및 배포 계획서 템플릿

문서 정보

버전	날짜	설명	작성자	승인자

개정 이력

발행 부서	
형상 ID	
문서 상태	

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 릴리스 및 배포 개요

- 릴리스 및 배포 프로세스의 아래의 사항에 대하여 간략하게 설명한다.

2.1 조직

- 릴리스 및 배포조직과 참여 인원별 역할에 대하여 기술한다.

번호	이름	직위/직책	소속

2.2 일정

- 릴리스 및 배포일정 및 업무 우선순위를 기술한다.

업무	계획	완료	우선순위

2.3 자원

- 릴리스 및 배포에 필요한 도구, 기술 및 방법에 대하여 기술한다.

3. 소프트웨어 릴리스 절차

- 릴리스 단계에서 수행해야 할 작업에 대하여 설명한다.

3.1 릴리스 대상 기능 정의

- 소프트웨어 릴리스에 포함되어야 하는 기능의 식별에 대하여 정의한다.

3.2 릴리스 영향 분석

- 식별된 기능에 영향을 미치는 어플리케이션 파라미터와 릴리스와의 연관성 식별을 정의한다.

3.3 릴리스 요소 식별

- 릴리스와 관련이 있는 요소들을 식별에 대하여 정의한다.

3.4 형상 관리

- 릴리스 소프트웨어의 베이스라인 기록 및 형상 통제 계획을 정의한다.

3. 소프트웨어 배포 절차

3.1 배포 유형

- 소프트웨어 배포 유형을 식별한다.

3.2 지원 계획

- 서비스 수준 및 지원 기간을 식별한다.

3.3 매체 정의

- 고객의 요구에 맞추어 소프트웨어 인도를 위한 매체 유형을 정의한다.

그림 141 소프트웨어 릴리즈 및 배포 계획서 템플릿 (예시)

2.1.4. 소프트웨어 릴리스 및 배포 계획서 체크리스트

표 184 소프트웨어 릴리스 및 배포 계획서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	소프트웨어 릴리스 및 배포 수행 조직이 정의되어 기술되었는가?
	소프트웨어 릴리스 및 배포 일정이 수립되어 기술되었는가?
	소프트웨어 릴리스 및 배포 필요 자원이 식별되어 기술되었는가?
	소프트웨어 릴리스 대상 기능이 식별되고 정의되어 있는가?
	소프트웨어 릴리스 대상 기능에의 영향 분석이 정의되어 있는가?
	소프트웨어 릴리스와 관련된 요소들이 식별되어 기술되었는가?
	소프트웨어 릴리스의 베이스라인 및 형상 통제 계획이 정의되어 있는가?
	소프트웨어 배포 유형 식별이 정의되어 있는가?
	소프트웨어 배포 지원 계획이 식별되어 있는가?
	소프트웨어 배포 매체가 정의되어 있는가?

2.2. 소프트웨어 배포 매뉴얼 작성

소프트웨어 배포 매뉴얼에 해당 소프트웨어 릴리스의 버전을 정확하게 식별하는 절차와 롤백을 포함한 설치 및 동작 확인 프로세스를 상세히 기술한다,

2.2.1. 소프트웨어 배포 매뉴얼 작성 수행 절차

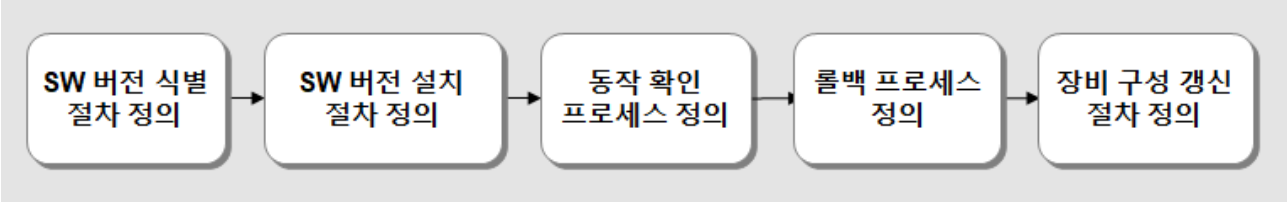


그림 142 소프트웨어 배포 매뉴얼 작성 절차 흐름도

표 185 소프트웨어 배포 매뉴얼 작성 절차

항 목	설 명
소프트웨어 릴리스 식별 절차 정의	- 배포되는 소프트웨어 버전을 확인하는 절차를 기술한다.
소프트웨어 릴리스 설치 절차 정의	- 해당 버전의 로딩 프로세스를 기술한다. - 버전이 올바르게 로딩 되었는지 확인하는 프로세스를 기술한다.
동작 확인 프로세스 정의	- 정상적인 환경에서 로드된 버전이 제대로 작동하는지 확인하는 프로세스를 기술한다. - 여러 가지 기능 테스트를 확인하고 배포 기록에 공식적으로 작성될 테스트 결과를 만든다.
롤백 프로세스 정의	- 문제 발생 시 로딩을 취소하고 이전 상태로 복구하는 프로세스를 기술한다.
장비 구성 업데이트 프로세스 정의	- 장비 구성에 대한 문서 형식(파일) 또는 전자 형식(데이터베이스)의 정보의 업데이트 프로세스를 기술한다. (필요시)

2.2.2. 소프트웨어 배포 매뉴얼 작성 지침

- 소프트웨어 배포 매뉴얼은 9.1.2의 입력 문서에 기초하여 작성되어야한다. (9.1.4.6)
- 소프트웨어 배포 매뉴얼에 소프트웨어 릴리스를 정확하게 식별하고 설치하기 위한 절차를 정의해야한다. (9.1.4.7)
- 증분 배포(즉, 개별 컴포넌트 배포)의 경우, 소프트웨어는 호환되지 않는 버전의 소프트웨어 컴포넌트를 활성화하지 못하도록 하는 기능을 포함하도록 설계되어야 하며, 이는 SIL 3 및 SIL 4에 적극 권장이고 SIL 1 및 SIL 2에 권장 사항이다. (9.1.4.8)
- 형상 관리를 통하여 동일한 소프트웨어 컴포넌트의 서로 다른 버전이 피할 수 없이 공존해야 하는 경우에 이로 인한 아무런 해가 발생하지 않도록 보장해야 한다. (9.1.4.9)
- 새로운 소프트웨어 릴리스를 설치할 때 롤백 절차 (즉, 이전 릴리스로 돌아갈 수 있는 기능)를 사용할 수 있어야 한다. (9.1.4.10)
- 소프트웨어는 탑재 과정에서 그리고 시스템에 적재 후에 식별을 가능하게 하는 내장 된 자체 식별 수단을 가져야한다. 자체 식별 수단은 소프트웨어 및 형상 데이터의 버전 정보와 제품의 특성 정보를 나타내야 한다. (9.1.4.11)

노트: 소프트웨어 릴리스에 대한 정보가 들어있는 코드 내의 데이터는 코드화를 통해 보호 될 수 있다 [표 206] “오류 검출 코드” 참조).

2.2.3. 소프트웨어 배포 매뉴얼 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 설치

2.1 소프트웨어 버전 정보 식별

- 배포되는 소프트웨어 버전을 확인하는 절차를 기술한다.

절차		결과
1		
2		
3		

2.2 소프트웨어 설치 절차

- 해당 버전의 로딩 프로세스를 기술한다.

2.3 소프트웨어 버전 확인

- 버전이 올바르게 로딩 되었는지 확인하는 프로세스를 기술한다.

3. 소프트웨어 확인 /검토 /점검

3.1 설치 확인

- 정상적인 환경에서 로드된 버전이 제대로 작동하는지 확인하는 프로세스를 기술한다.

항목		결과
1		
2		
3		

3.2 설치 테스트

- 설치 결과 확인을 위한 기능 테스트 절차와 및 테스트 로그 수집 절차를 기술한다.

4. 롤백/원상복구 프로세스

- 로딩을 취소하고 이전 상태로 복구하는 절차를 기술한다.

5. 장비 구성 업데이트 절차

- 장비 구성에 대한 문서 형식(파일) 또는 전자 형식(데이터베이스)의 정보의 업데이트 프로세스를 기술한다.

그림 143 소프트웨어 배포 매뉴얼 템플릿 (예시)

2.2.4. 소프트웨어 배포 매뉴얼 체크리스트

표 186 소프트웨어 배포 매뉴얼 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	소프트웨어 버전 정보 식별 절차가 정의되어 있는가?
	소프트웨어 설치 절차가 정의되어 있는가?
	소프트웨어 버전 확인 절차가 정의되어 있는가?
	소프트웨어 설치 확인 절차가 정의되어 있는가?
	소프트웨어 설치 테스트 절차가 정의되어 있는가?
	단일 구성 요소 배포 절차가 정의되어 있는가? (해당 시)
	동일 소프트웨어 컴포넌트의 서로 다른 버전의 공존에 대하여 기술 되어 있는가? (필요 시)
	소프트웨어 설치 롤백 절차가 정의되어 있는가?
	장비 구성 업데이트 절차가 기술되어 있는가?

2.3. 소프트웨어 릴리스

소프트웨어 릴리스를 지원하기 위한 모든 문서를 작성하고, 검토하고, 승인하고, 가용하도록 보장하며, 릴리스 노트를 정의하고 작성한다. 릴리스 노트에는 소프트웨어 릴리스의 주요 특징에 대한 상세 설명을 기술 한다.

2.3.1. 소프트웨어 릴리스 수행 절차

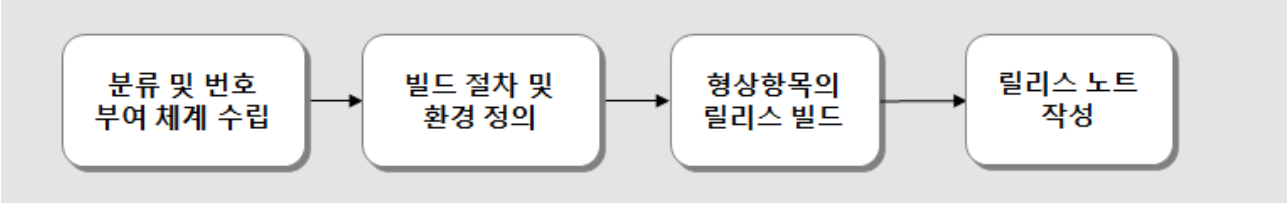


그림 144 소프트웨어 릴리스 수행 흐름도

표 187 소프트웨어 릴리스 절차

항 목	설 명
릴리스 분류 및 번호 부여 체계 수립	- 릴리스의 목적 및 기대 사항에 근거한 제품 릴리스 분류 및 번호 부여 체계를 수립한다.
빌드 절차 및 빌드 환경 정의	- 일관성 있는 빌드 프로세스를 확립하고 유지관리 한다. - 모든 관련자는 구체화되고 일관된 빌드 환경을 사용하여야 한다.
릴리스 빌드	- 무결성을 보장하기 위하여 형상항목으로 부터 릴리스를 빌드 한다. - 소프트웨어 릴리스는 릴리스 이전에 정확한 하드웨어 수정에 맞게 프로그래밍 되어야 한다.
릴리스 노트 작성	- 릴리스의 주요 특징에 대한 상세 설명이 담긴 자세한 정보를 릴리스 노트에 작성한다. - 소프트웨어 릴리스의 개요, 환경적 요구사항, 호환성 정보, 소프트웨어 제약 사항, 신규 기능 식별, 결함 해결 목록, 알려진 결함 및 임시 해결책 등을 포함하여 릴리스 노트를 작성한다.

2.3.2. 소프트웨어 릴리스 지침

○ 소프트웨어 릴리스는 베이스라인 생명주기 전반에 걸쳐 재생성 가능해야한다. (9.1.4.3)

○ 릴리스 노트는 9.1.2의 입력 문서에 기초하여 설계자의 책임 하에 작성되어야한다. (9.1.4.4)

○ 릴리스 노트에는 아래의 사항이 포함되도록 작성되어야 한다. (9.1.4.5)

a) 준수해야하는 적용 조건

b) 소프트웨어 컴포넌트 간 및 소프트웨어와 하드웨어 간의 호환성 정보

c) 소프트웨어 사용에 대한 모든 제약 사항 (7.7.4.13 참조)

2.3.3. 소프트웨어 릴리스 노트 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 릴리스 개요

2.1 릴리스 목적

- 해당 릴리스의 목적을 기술한다.

2.2 릴리스 분류

- 해당 릴리스의 분류를 기술한다.

2.3 릴리스 번호

- 해당 릴리스의 번호를 기술한다.

2.4 베이스 라인

- 릴리스 소프트웨어의 베이스 라인 등록일 설정 정보를 기술한다.

릴리스 목적	문제 수정, 기능 향상, 신규/변경 요구사항 적용 등
릴리스 분류	Major, Minor, Emergency
릴리스 번호	
베이스 라인	Branch, Label, TAG 정보 등
릴리스 일자	

3. 소프트웨어 정보

3.1 환경적 요구사항

- 준수해야하는 적용 조건을 기술한다.

3.2 호환성 정보

- 소프트웨어 컴포넌트 간 및 소프트웨어와 하드웨어 간의 호환성 정보를 기술한다.

3.2 버전 정보

- 소프트웨어 파일 이름, 크기, 날짜 등의 정보를 기술한다.

4. 소프트웨어 제약 사항

- 소프트웨어 사용에 대한 모든 제약 사항을 기술한다.

5. 신규 기능 식별

- 릴리스에 포함된 신규 기능을 식별하여 기술한다.

기능	세부 기능	유형	버전
대분류	소분류	신규/재사용/COTS 등	

6. 결함 해결 목록

- 릴리스에 의해 해결된 결함 목록을 기술한다.

이슈번호	결함내용	결함 근거	해결 요약

7. 알려진 결함 및 임시 해결책

- 미해결 결함 목록과 임시 해결 방안을 기술한다.

이슈번호	결함내용	임시 해결책

8. 릴리스 수행

8.1 릴리스 조직

- 릴리스 수행 부서 정보를 기술한다.

8.2 릴리스 대상 조직

- 릴리스 대상 조직 관련 정보를 기술한다.

8.3 릴리스 담당자

- 릴리스 담당자 사인을 작성한다.

8.4 릴리스 승인자

- 릴리스 담당팀장 사인을 작성한다.

릴리스 부서	
릴리스 대상	
베이스 담당	
릴리스 승인	

그림 145 소프트웨어 릴리스 노트 템플릿 (예시)

2.3.4. 소프트웨어 릴리스 노트 체크리스트

표 188 소프트웨어 배포 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	릴리스의 목적이 기술되어 있는가?
	릴리스의 분류가 기술되어 있는가?
	릴리스 번호가 기술되어 있는가?
	릴리스의 베이스 라인을 기술하고 있는가?
	릴리스 일자를 기술하고 있는가?
	환경적 요구사항을 기술하고 있는가?
	호환성 정보를 기술하고 있는가?
	소프트웨어 버전 정보가 기술되어 있는가?
	소프트웨어 제약 사항이 기술되어 있는가?
	신규 기능이 식별되어 있는가?
	결함 해결 목록이 기술되어 있는가?
	미해결 결함 목록과 임시 해결 방안이 기술되어 있는가?
	릴리스 수행 조직이 식별되어 기술되어 있는가?
	릴리스 대상 조직이 기술되어 있는가?
	릴리스 담당자를 기술하고 사인하였는가?
	릴리스 승인자를 기술되고 사인하였는가?

2.4. 소프트웨어 배포

실 운용 환경에 새로운 버전을 설치하고 진단을 수행하여 배포된 소프트웨어의 동작 상태를 확인하고 소프트웨어 배포 기록을 작성한다,

2.4.1. 소프트웨어 배포 수행 절차

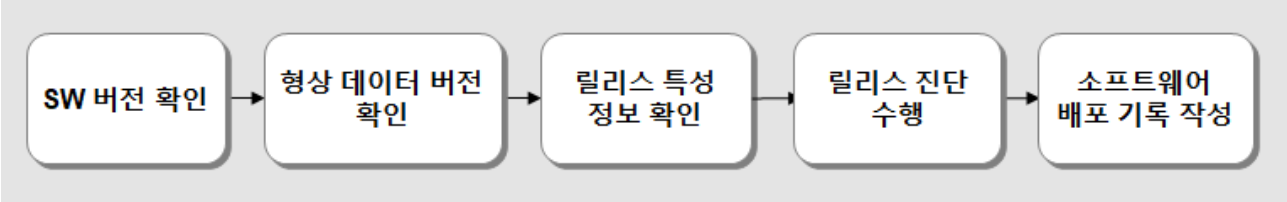


그림 146 소프트웨어 배포 수행 흐름도

표 189 소프트웨어 배포 기록 작성 절차

항 목	설 명
소프트웨어 버전 정보 확인	- 배포된 소프트웨어의 버전을 확인한다. - 버전이 올바르게 설치되었는지 확인한다.
형상 데이터 버전 정보 확인	- 형상 데이터의 버전 정보를 확인한다.
릴리스 특성 정보 확인	- 소프트웨어 릴리스의 특성 정보를 확인한다.
릴리스 진단 정보 확인	- 소프트웨어 릴리스가 정상적으로 배포되었는지 진단을 수행하고, 결과 로그를 수집한다.
소프트웨어 배포 기록 작성	- 소프트웨어 배포 기록을 작성한다.

2.4.2. 소프트웨어 배포 기록 작성 지침

○ 소프트웨어 배포 기록은 9.1.2의 입력 문서에 기초하여 작성되어야한다. (9.1.4.12)

○ 소프트웨어 배포 기록은 내장 된 자체 식별 메커니즘 (9.1.4.11 참조)의 검사를 통하여 의도된 소프트웨어가 탑재되었다는 증거로 제공된다. 이 기록은 다른 검증과 같이 전달된 시스템 관련 문서에 저장되어야하며 시운전 및 승인의 일부이다. (9.1.4.13)

○ 배포된 소프트웨어가 인도된 설치에 대해 추적 가능하도록 관리 되어야 한다. (9.1.4.14)

○ 노트: 이는 치명적인 오류가 발견되어 하나 이상의 설치에서 수정해야 할 때 특히 중요하다.

○ 결함 모니터링의 일부로서 소프트웨어는 진단 정보를 제공해야 한다. (9.1.4.15)

2.4.3. 소프트웨어 배포 기록 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.

- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서

- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 버전 확인

- 배포된 소프트웨어 버전을 확인하고 결과를 기술한다.

3. 형상 데이터 버전 확인

- 형상 데이터의 버전 정보를 확인하고 결과를 기술한다.

4. 릴리스 특성 정보 확인

- 소프트웨어 릴리스의 특성 정보를 확인하고 결과를 기술한다.

5. 소프트웨어 릴리스 진단

- 소프트웨어 릴리스가 올바르게 설치되었는지 확인하고 결과를 기술한다.

6. 소프트웨어 설치 기록

- 소프트웨어 릴리스의 설치 기록을 기술한다.

그림 147 소프트웨어 배포 기록 템플릿 (예시)

2.4.4. 소프트웨어 배포 기록 체크리스트

표 190 소프트웨어 배포 기록 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	배포된 소프트웨어의 버전을 확인하고 결과를 기술하였는가?
	구성 데이터의 버전 정보를 확인하고 결과를 기술하였는가?
	소프트웨어 릴리스의 특성 정보를 확인하고 결과를 기술하였는가?
	소프트웨어 릴리스가 올바르게 설치되었는지 확인하고 결과를 기술하였는가?
	소프트웨어 릴리스의 설치 기록을 기술하였는가?
	배포된 소프트웨어는 인도된 설치에 추적 가능한가?
	진단 정보가 결함 모니터링에 제공 되었는가? (해당 시)

2.5. 소프트웨어 배포 검증

소프트웨어 배포 가이드라인을 사용하여 소프트웨어 배포 매뉴얼, 릴리스 노트, 배포 기록의 일관성, 적합성, 가독성 및 추적성에 대한 일반 요구사항과 각 명세별 특정 요구사항의 충족 여부를 검증한다.

2.5.1. 소프트웨어 배포 검증 수행 절차

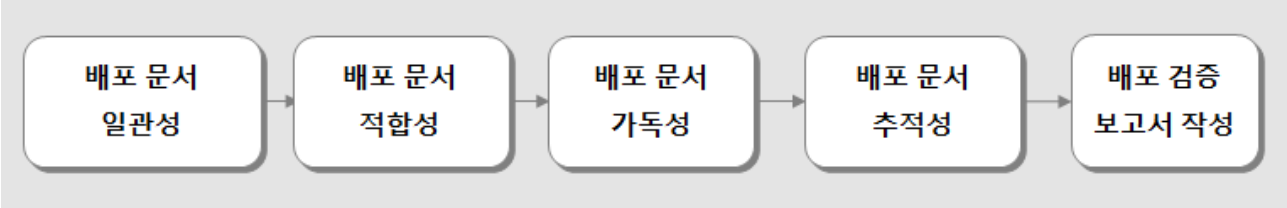


그림 148 소프트웨어 배포 검증 흐름도

표 191 소프트웨어 배포 검증 절차

항 목	설 명
배포 문서 일관성 검토	- 소프트웨어 배포 매뉴얼의 일관성, 완전성을 검토한다. - 소프트웨어 릴리스 노트의 일관성, 완전성을 검토한다. - 소프트웨어 배포 기록의 일관성, 완전성을 검토한다.
배포 문서 가독성 검토	- 소프트웨어 배포 매뉴얼의 가독성을 검토한다. - 소프트웨어 릴리스 노트의 가독성을 검토한다. - 소프트웨어 배포 기록의 가독성을 검토한다.
배포 문서 추적성 검토	- 소프트웨어 배포 매뉴얼의 추적성을 검토한다. - 소프트웨어 릴리스 노트의 추적성을 검토한다. - 소프트웨어 배포 기록의 추적성을 검토한다.
배포 검증 보고서 작성	- 검증 결과를 정리한다. - 소프트웨어 검증 보고서를 작성한다.

2.5.2. 소프트웨어 배포 검증 지침

○ 소프트웨어 배포 검증 보고서는 검증자의 책임아래 9.1.2의 입력 문서를 기반으로 작성되어야한다. (9.1.4.16)

○ 소프트웨어 배포 매뉴얼이 수립되면, 소프트웨어 배포 검증은 다음 사항을 다루어야한다. (9.1.4.17)

a) 소프트웨어 배포 매뉴얼의 가독성 및 추적성에 대한 일반 요구사항과 9.1.4.7의 특정 요구사항 충족 여부

b) 소프트웨어 배포 매뉴얼의 내부 일관성

○ 배포 기록이 확립되면, 소프트웨어 배포검증은 다음 사항을 다루어야한다. (9.1.4.18)

a) 배포 기록의 가독성 및 추적성에 대한 일반 요구사항과 9.1.4.13의 특정 요구사항 충족 여부

b) 배포 기록의 내부 일관성

○ 릴리스 노트가 수립되면, 소프트웨어 배포 검증은 다음 사항을 다루어야한다. (9.1.4.19)

a) 릴리스 노트의 가독성 및 추적성에 대한 일반 요구사항과 9.1.4.5의 특정 요구사항 충족 여부

b) 릴리스 노트의 내부 일관성

결과는 배포 검증 보고서에 기록되어야한다.

2.5.3. 소프트웨어 배포 검증 보고서 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 검증 요약

2.1 배포 개요

- 배포 소프트웨어의 특성 정보 및 구성을 기술한다.

2.2 검증자

- 검증자 이름을 기술한다.

3. 검증 결과 상세

3.1 소프트웨어 배포 매뉴얼

- 소프트웨어 배포 매뉴얼의 가독성 및 추적성에 대한 요구사항 충족 여부
- 소프트웨어 배포 매뉴얼의 내부 일관성을 검증하고 결과를 기술한다.

3.2 배포 기록

- 배포 기록의 가독성 및 추적성에 대한 요구사항 충족 여부
- 배포 기록의 내부 일관성을 검증하고 결과를 기술한다.

3.3 릴리스 노트

- 릴리스 노트의 가독성 및 추적성에 대한 요구사항 충족 여부

- 릴리스 노트의 내부 일관성을 검증하고 결과를 기술한다.

4. 검증 결과

- 검증 결과를 요약하여 기술한다.

그림 149 소프트웨어 배포 검증 보고서 템플릿 (예시)

2.5.4. 소프트웨어 배포 검증 보고서 체크리스트

표 192 소프트웨어 배포 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	배포 소프트웨어의 특성 정보 및 구성을 기술하고 있는가?
	검증자의 이름이 기술되어 있는가?
	소프트웨어 배포 매뉴얼의 가독성 및 추적성에 대한 요구사항 충족여부를 검증하고 결과를 기술하였는가?
	소프트웨어 배포 매뉴얼의 내부 일관성을 검증하고 결과를 기술하였는가?
	배포 기록의 가독성 및 추적성에 대한 요구사항 충족 여부를 검증하고 결과를 기술하였는가?
	배포 기록의 내부 일관성을 검증하고 결과를 기술하였는가?
	릴리스 노트의 가독성 및 추적성에 대한 요구사항 충족 여부를 검증하고 결과를 기술하였는가?
	릴리스 노트의 내부 일관성을 검증하고 결과를 기술하였는가?
	검증 경과를 요약하여 기술되어 있는가?

제 8 절 소프트웨어 유지보수

1. 개요

소프트웨어 유지보수 단계에서 적용할 새로운 버전의 소프트웨어 생성 방안을 포함하여 소프트웨어 유지보수 계획 수립, 소프트웨어 변경의 영향을 판단하여 소프트웨어 수정 수행, 유지보수 결과에 대한 로그 생성 및 유지보수 검증을 수행하는 절차에 대하여 기술한다.

1.1. 목표

소프트웨어 자체에 대한 정정, 개선 또는 수정 작업을 수행할 때 할당된 소프트웨어 안전 무결성 등급과 시스템 종속성을 유지하면서 요구사항을 충실히 만족하도록 소프트웨어가 수행됨을 보장한다.

1.2. 범위

소프트웨어 개발 생명 주기에서 IEC 62279 9.2절에 해당하는 소프트웨어 유지보수단계에 대해 설명한다.

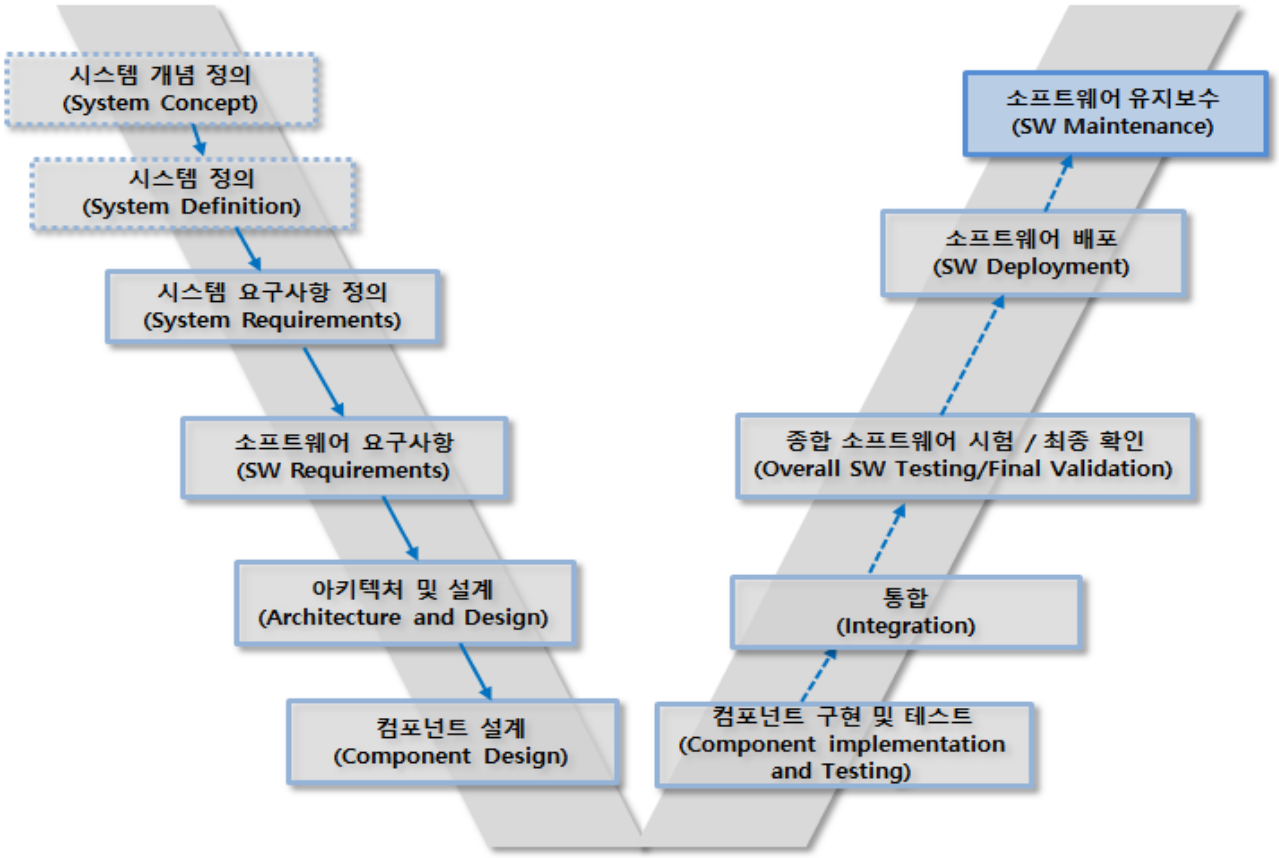


그림 150 소프트웨어 개발 생명주기 - 소프트웨어 유지보수 단계

1.3. 시작 기준

○ 소프트웨어 배포 완료

1.4. 완료 기준

○ 소프트웨어 유지보수 계획 수립 완료
○ 소프트웨어 변경 기록 완료
○ 소프트웨어 유지보수 기록 완료
○ 소프트웨어 유지보수 검증 완료

1.5. 입력물

모든 디자인, 개발 및 분석 문서

1.6. 산출물

표 193 소프트웨어 유지보수 단계 문서

문 서	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
소프트웨어 유지보수 계획서	R	HR	HR	HR	HR
소프트웨어 변경 기록	HR	HR	HR	HR	HR
소프트웨어 유지보수 기록	R	HR	HR	HR	HR
소프트웨어 유지보수 검증 보고서	R	HR	HR	HR	HR

1.7. 역할 및 책임

표 194 소프트웨어 유지보수 단계 역할 및 책임

단 계	문 서	작 성 자	1차 검토	2차 검토
소프트웨어 유지보수	43.소프트웨어 유지보수 계획서	a	VER	VAL
	44.소프트웨어 변경 기록	a	VER	VAL
	45.소프트웨어 유지보수 기록	a	VER	VAL
	46.소프트웨어 유지보수 검증 보고서	VER		VAL
a 역할이 정의되어 있지 않음 VER (Verifier) 검증자 VAL (Validator) 확인자				

1.8. 소프트웨어 유지보수 주요 활동

그림 151 소프트웨어 유지보수 주요 활동

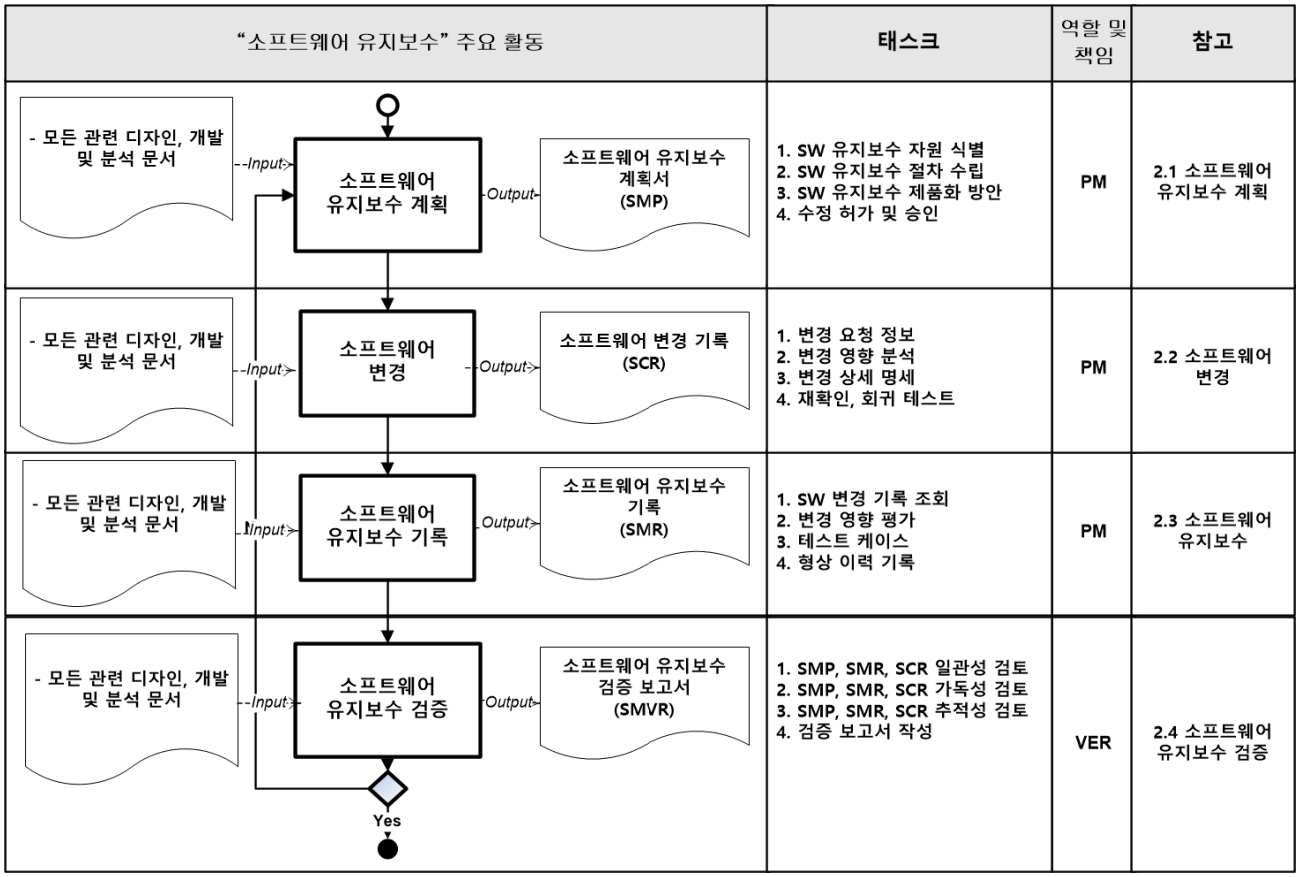


표 195 소프트웨어 유지보수 단계

활 동 ID	활 동 명	설 명
MNT.01	소프트웨어 유지보수 계획	• 소프트웨어 유지보수에 필요한 자원일 식별한다. • 소프트웨어 유지보수 절차를 수립한다. • 소프트웨어 유지보수 제품화 계획을 정의한다. • 소프트웨어 유지보수의 변경 수행 결정 및 결과 승인에 대한 권한에 관하여 기술한다.
MNT.02	소프트웨어 변경	• 변경의 상세 정보를 식별한다. • 변경의 영향을 분석한다. • 변경에 대하여 상세 명세를 수행한다. • 변경에 대한 재확인 및 회귀테스트를 수행한다.
MNT.03	소프트웨어 유지보수	• 소프트웨어 변경 기록을 참조한다. • 소프트웨어 변경 영향을 평가한다. • 소프트웨어 컴포넌트 테스트 케이스를 기술한다. • 소프트웨어 형상 이력을 기록한다.
MNT.04	소프트웨어 유지보수 검증	• 소프트웨어 유지보수의 일관성을 검증한다. • 소프트웨어 유지보수의 가독성을 검증한다. • 소프트웨어 유지보수의 추적성을 검증한다. • 소프트웨어 유지보수 검증 보고서를 작성한다.

2. 세부 수행 활동

소프트웨어 유지보수 단계에서는 소프트웨어 오류에 대한 결함 수정, 소프트웨어 변경 요청에 대한 처리, 변경에 대한 영향 분석 및 평가, 변경 데이터 기록 및 분석, 변경 결과 기록 및 유지보수 기록을 작성하고, 소프트웨어 유지보수에 대한 검증을 수행한다.

본 소프트웨어 개발 가이드에서의 세부 수행 활동 내용 중 IEC 62279에서 제시하는 내용은 항목 번호를 표시하였다.

- 이 표준(62279)은 소급 적용을 목적으로 하지는 않으므로 기본적으로 새로운 개발에 적용하지만, 기존 소프트웨어에 주요한 수정을 가하는 경우에는 전체적으로 적용하며, 9.2 소프트웨어 유지보수에 관한 사항은 사소한 범주를 포함한 모든 변경에도 적용된다. 그럼에도 기존 소프트웨어를 업그레이드하고 유지보수 하는 동안 이 표준(62279) 전체를 적용하는 것이 권고된다. (9.2.4.1)
- 모든 소프트웨어 안전 무결성 등급에 대해 공급자는 변경 작업을 시작하기 전에 유지보수 작업이 주요한 변경 또는 사소한 변경으로 간주해야 할지, 또는 시스템의 기존 유지 관리 방법이 적절한지 여부를 결정해야한다. 공급자는 이러한 결정의 정당성을 입증하고 기록하여, 평가자에게 심사의 근거로 제출하여야한다. (9.2.4.2)

  ※ 유지보수의 주요 변경 판단 예시
  - 함수간의 의존 관계
    - Function Call 10 이상 : 주요 변경
    - Function Call 5 이하 : 사소 변경
  - 변경 모듈의 기능 점수
    - FP 50 이상 : 주요 변경
    - FP 30 이하 : 사소 변경
  - 변경 모듈의 순환 복잡도
    - CC 20 이상 : 주요 변경
    - CC 10 이하 : 사소 변경
  - 변경 요청 결함의 심각성
    - RPN 150 이상 : 주요 변경
    - RPN 100 이하 : 사소 변경
  - 수정 기능의 법규 / 표준 위반 정도
    - 필수 사항 위반 : 주요 변경

- 권고 사항 위반 : 사소 변경

○ 유지보수는 ISO/IEC 90003에 포함된 지침에 따라 수행되어야한다. (9.2.4.3)

○ 유지보수 가능성은 특히 7.3, 7.4 및 7.5의 요구사항을 준수함으로써 소프트웨어의 고유한 측면으로 설계되어야한다. 또한 최소한의 유지보수성을 구현하고 검증하기 위해 ISO/IEC 25010 시리즈를 적용해야한다. (9.2.4.4)

※ 소프트웨어 유지보수 가능성의 부 품질 특성 예시

- 분석가능성: 소프트웨어 결함이나 고장의 원인 혹은 변경될 부분들의 식별에 대한 진단을 가능하게 하는 품질 특성
- 변경가능성 : 변경 병세가 구현될 수 있도록 하는 품질 특성
- 안전성 : 소프트웨어가 변경으로 인한 예상치 않은 결과를 최소화하는 품질 특성
- 테스트 가능성 : 변경된 소프트웨어가 확인될 수 있는 품질 특성

○ 유지보수 활동은 소프트웨어 유지보수 계획에 따라 수행되어야 한다. (9.2.4.15)

○ 표 A.10의 기술과 조치가 선택되어야한다. 선택된 조합은 4.8과 4.9를 만족하는 집합으로 정당화되어야 한다. (9.2.4.16)

○ 유지보수는 최소한 소프트웨어의 초기 개발 단계와 동일한 수준의 전문 지식, 도구, 문서, 계획 및 관리로 수행되어야 한다. 이는 형상 관리, 변경 제어, 문서 제어 및 관련 당사자의 독립성에도 적용된다. (9.2.4.17)

○ 외부 공급자 통제, 문제보고 및 시정 조치는 신규 소프트웨어 개발과 관련하여 소프트웨어 품질 보증 (6.5)의 관련 단락에 명시된 것과 동일한 기준으로 관리되어야 한다. (9.2.4.18)

○ 보고 된 각 문제 또는 개선 사항에 대해 안전 영향 분석을 실시해야 한다. (9.2.4.19)

※ 안전 영향 분석 예시

- 결함 검토 : 생성 시점, 시스템 경계, 결함 원인, 결함 영역, 심각도, 검출도, 발생도, 위험도
- 결함 분석 : 리소스 경합, 시점 불일치, 통신 불량, 통합 오류
- 안전 분석 : FMEA, FTA, HAZOP

○ 유지보수중인 소프트웨어의 경우, 보고 된 문제를 조사하고 수정하는 동안 시스템의 전반적인 무결성을 보장하기 위해 확인된 위험에 비례하는 완화 조치를 취해야 한다. (9.2.4.20)

- 완화 조치 운영 예시
  - 안전 상태 (fail-safe) 운영 : 오류 발생 시 시스템 운영에 최소한의 영향을 주며, 위험을 방지하거나 완화하여 시스템의 본질적인 기능을 수행
  - 기능저하 상태 운영 : 오류 발생 시 제한된 기능을 유지하여 위험을 회피하고 작동 불능 상태가 되지 않도록 성능을 저하하여 운영 유지

2.1. 소프트웨어 유지보수 계획

소프트웨어 유지보수에 필요한 자원의 식별, 오류 보고 체계, 오류 로그 수집, 유지보수 기록 작성, 변경 영향 분석 및 소프트웨어 형상 통제 등 소프트웨어 유지보수에 따른 단계별 절차를 수립하고 변경의 수행, 검증, 확인 및 평가와 수정 허가 및 변경된 소프트웨어의 승인에 대한 권한 정의를 소프트웨어 유지보수 계획서에 작성한다.

2.1.1. 소프트웨어 유지보수 계획 절차

○ 소프트웨어 유지보수 계획 수행 흐름

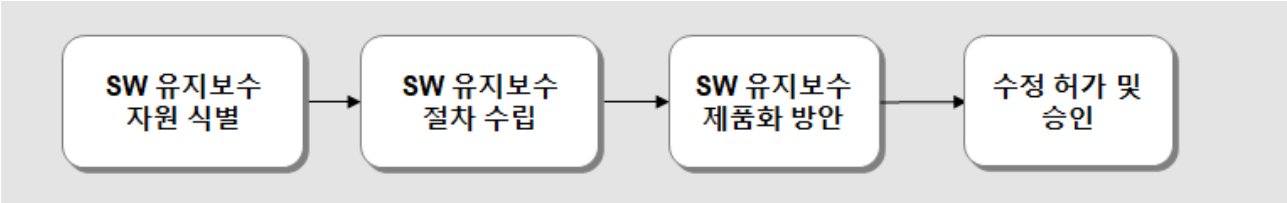


그림 152 소프트웨어 유지보수 계획 흐름도

○ 소프트웨어 유지보수 계획 절차

표 196 소프트웨어 유지보수 계획 절차

항 목	설 명
소프트웨어 유지보수 자원 식별	- 해당 소프트웨어 유지보수에 필요한 자원 분석을 수행한다. - 유지보수에 필요한 공수를 산정한다.
소프트웨어 유지보수 절차 수립	- 오류 처리 절차를 기술한다. - 적용할 유지보수 절차를 식별한다. - 유지보수 기록 절차를 기술한다. - 소프트웨어 형상 통제 방안을 기술한다.
소프트웨어 유지보수 제품화 방안	- 새로운 버전의 생성에 적용할 방법론을 기술한다. - 변경에 대한 검증, 확인 및 평가 절차를 기술한다.
수정 허가 및 승인	- 소프트웨어 수정 및 변경 허가 권한에 대하여 기술한다. - 변경된 소프트웨어 승인 권한에 대하여 정의한다.

2.1.2. 소프트웨어 유지보수 계획 지침

- 소프트웨어 유지보수 계획은 9.2.2의 입력 문서에 기초하여 작성되어야 한다. (9.2.4.5)
- 소프트웨어 유지보수 절차를 수립하여 소프트웨어 유지보수 계획에 기록해야 한다. (9.2.4.6)
- 이 절차는 또한 아래의 사항을 다루어야 한다.
  - 오류보고, 오류 로그, 유지보수 기록, 변경 허가 및 소프트웨어/시스템 형상에 대한 통제 및 [표 A.10]의 기법 및 대책
  - 모든 변경에 대한 검증, 확인 및 평가
  - 변경된 소프트웨어를 승인하는 권한의 정의 및
  - 수정 허가

2.1.3. 소프트웨어 유지보수 계획서 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 유지보수 개요

- 유지보수 프로세스의 아래의 사항에 대하여 간략하게 설명한다.

2.1 조직

- 유지보수 조직과 참여 인원별 역할에 대하여 기술한다.

2.2 일정

- 유지보수 일정 및 업무 우선순위를 기술한다.

2.3 자원

- 유지보수에 필요한 도구, 기술 및 방법에 대하여 기술한다.

3. 소프트웨어 유지보수 수행

- 유지보수 프로세스의 각 단계에서 수행 할 작업을 설명한다.

3.1 변경 요청 분석

- 변경 요청 사항에 대한 상세 내역의 분석 및 식별 방안을 기술한다.

3.2 안전 영향 분석

- 변경이 시스템 전반에 미치는 안전 측면의 영향에 대한 분석 방안을 기술한다.

3.3 변경 설계

- 수정 또는 추가되는 소프트웨어 컴포넌트와 유닛의 설계 방안을 기술한다.

3.4 변경 구현

- 수정 또는 추가되는 소프트웨어 유닛의 구현 방안을 기술한다.

3.5 시스템 테스트

- 시스템의 모든 요소들이 적합하게 통합되어 변경된 소프트웨어가 정확하게 수행되는지 확인하는 시스템 테스트 절차를 기술한다.

3.6 유지보수 배포

- 소프트웨어 유지보수 제품 생성 및 배포 계획을 정의한다.

3.7 형상 관리

- 소프트웨어 변경 과정에서 수정되거나 신규 작성된 산출물을 체계적이고 효율적으로 관리하기 위한 형상 관리 방안을 기술한다.

4. 소프트웨어 유지보수 관리

4.1 승인 권한

- 소프트웨어 유지보수 수행의 결정 권한과 유지보수 결과의 승인 권한에 대하여 기술한다.

4.2 기록 및 검증

- 유지보수 프로세스의 결과를 기록하고, 검증을 수행하는 절차를 기술한다.

그림 153 소프트웨어 유지보수 계획서 템플릿 (예시)

2.1.4. 소프트웨어 유지보수 계획서 체크리스트

표 197 소프트웨어 유지보수 계획서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	유지보수 조직과 참여 인원별 역할이 정의 되었는가?
	유지보수 일정 및 업무 우선순위가 기술되어 있는가?
	유지보수에 필요한 도구, 기술 및 방법 등 자원에 대하여 기술되어 있는가?
	변경 요청 분석 및 식별 방안을 기술하고 있는가?
	안전 영향 분석 방안이 기술되어 있는가?
	수정 또는 추가되는 소프트웨어 컴포넌트와 유닛의 설계 방안이 기술되어 있는가?
	수정 또는 추가되는 소프트웨어 유닛의 구현 방안이 기술되어 있는가?
	시스템 테스트 절차가 기술되어 있는가?
	소프트웨어 유지보수 제품 생성 및 배포 계획이 정의되어 있는가?
	소프트웨어 유지보수 형상관리 방안이 정의되어 있는가?
	소프트웨어 유지보수 수행의 결정 권한과 유지보수 결과의 승인 권한에 대하여 기술하고 있는가?
	유지보수 프로세스의 결과를 기록하고, 검증을 수행하는 절차를 기술하고 있는가?

2.2. 소프트웨어 변경

유지보수 담당자는 변경 요청을 검토하고 변경 영향을 분석하여, 소프트웨어 컴포넌트를 설계 및 구현하고 적절한 테스트를 수행한다.

2.2.1. 소프트웨어 변경 수행 절차

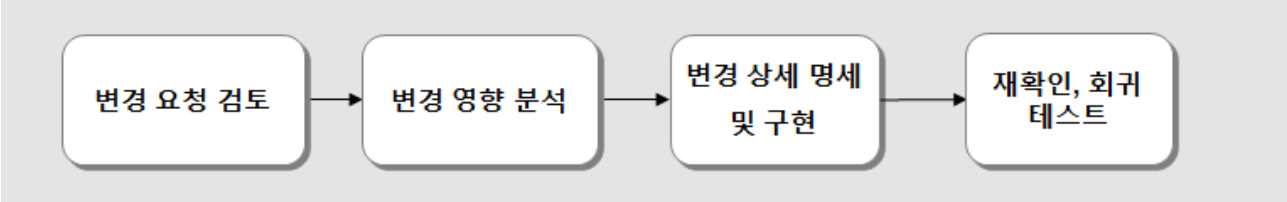


그림 154 소프트웨어 변경 수행 흐름도

표 198 소프트웨어 변경 절차

항 목	설 명
변경 요청 검토	- 변경 요청을 분석하여 검토 필요한 문서와 소프트웨어 컴포넌트, 버전 정보를 파악한다. - 기존 소프트웨어에서 수정되어야할 요소를 식별한다. - 소프트웨어의 유지보수 가능성 평가를 수행한다.
변경 영향 분석	- 변경이 전체 시스템에 미칠 영향을 분석한다. - 변경에 의해 영향 받을 인터페이스를 식별한다. - 파급 효과를 확인한다. - 위험 분석을 수행한다.
변경 상세 명세 및 구현	- 기존 시스템에서 수정되어야할 소프트웨어 컴포넌트를 식별한다. - 소프트웨어 문서를 수정한다. - 변경을 구현한다. - 소프트웨어 매뉴얼을 업데이트 한다.
재확인, 회귀 테스트	- 수정된 부분과 수정되지 않은 부분을 시험하고 평가한다. - 추가 또는 수정 요구사항의 완전하고 정확한 구현을 보장한다. - 변경되지 않은 요구사항이 영향을 받지 않음을 보장한다.

2.2.2. 소프트웨어 변경 지침

○ 소프트웨어 변경 기록은 9.2.2의 입력 문서에 기초하여 작성되어야 한다. (9.2.4.9)

○ 소프트웨어 변경 기록은 반드시 각각의 유지보수 활동에 대하여 작성되어야 한다. 소프트웨어 변경 기록은 다음을 반드시 포함해야 한다. (9.2.4.10)

- 수정 또는 변경 요청, 버전, 결함의 분류, 변경 요청사항 및 변경의 근원
- 하드웨어, 소프트웨어, 인간 상호 작용 그리고 환경과 가능한 상호 작용을 포

함한 전체 시스템에 대한 유지보수 활동의 영향 분석

- 수정 또는 변경 수행에 대한 상세 명세
- 소프트웨어 안전 무결성 등급에서 요구되는 범위의 수정 또는 변경에 대한 재확인, 회귀 테스트 및 재평가.

재확인에 대한 책임은 소프트웨어 안전 무결성 레벨에 따라 프로젝트마다 다를 수 있다. 또한 수정 또는 변경이 재확인 프로세스에 미치는 영향은 서로 다른 시스템 레벨 (변경된 컴포넌트에만 영향, 모든 확인된 컴포넌트에 영향, 전체 시스템에 영향)에 의해 제한 받을 수 있다. 그러므로 소프트웨어 검증 계획은 소프트웨어 안전 무결성 레벨에 따라 두 가지 문제를 모두 해결해야 한다. 유지보수 재확인의 독립 수준은 반드시 이전 확인의 독립 수준과 동일해야 한다.

2.2.3. 소프트웨어 변경 기록 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.

- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 변경 요청 검토

2.1 변경 요청 / 분제 보고 분석

- 수정 또는 변경 요청, 버전, 결함의 분류, 변경 요청사항 및 변경의 근본 사유를 기술한다.

2.2 문제 실증 / 재현

- 문제 현상을 실증하거나 재현하여 확인한다.

2.3 수정 대안 수립

- 수정 또는 변경을 구현하기 위한 옵션을 개발하고 문서화 한다.

2.4 수립 방안 승인

- 수립한 수정 또는 변경 방안에 대하여 승인을 획득한다.

3. 변경 영향 분석

3.1 하드웨어 영향 분석

- 수정 또는 변경의 하드웨어 영향을 분석한다.

3.2 소프트웨어 영향 분석

- 수정 또는 변경의 소프트웨어 영향을 분석한다.

3.3 사용자 영향 분석

- 수정 또는 변경의 사용자 영향을 분석한다.

3.4 환경 및 상호 작용 영향 분석

- 수정 또는 변경의 환경과 그 상호 작용에 대한 영향을 분석한다.

4. 변경 상세 설계 및 구현

4.1 설계 문서 수정

- 수정 또는 변경 수행에 대한 상세 명세를 수정하여 기술한다.

4.2 테스트 문서 수정

- 수정 또는 변경 수행의 테스트에 대한 명세를 수정하여 기술한다.

4.3 소스 코드 수정 및 확인

- 수정 또는 변경 해당 소프트웨어 유닛을 수정하여 작성한다.
- 수정 코드의 테스트 가능성을 확인한다.
- 수정 코드의 코딩 표준 준수여부를 확인한다.

5. 재확인 및 회귀 테스트

5.1 재확인

- 수정 또는 변경에 대한 재확인을 수행하고 결과를 작성한다.

5.2 회귀 테스트

- 수정 또는 변경에 대한 회귀 테스트를 수행하고 결과를 작성한다.

그림 155 소프트웨어 변경 템플릿 (예시)

2.2.4. 소프트웨어 변경 기록 체크리스트

표 199 소프트웨어 / 하드웨어 유지보수 테스트 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	수정 또는 변경 요청 버전의 결함을 분류하여 기술하고 있는가?
	변경 요청사항 및 변경의 근본 사유가 기술되어 있는가?
	문제 현상을 실증하거나 재현하여 확인하고 결과를 기술하였는가?
	수정 또는 변경을 구현하기 위한 옵션을 개발하여 기술하고 있는가?

구 분	점검 사항
	수립한 수정 또는 변경 방안에 대하여 승인을 획득하였는가?
	수정 또는 변경의 하드웨어 영향을 분석하였는가? (해당 시)
	수정 또는 변경의 소프트웨어 영향을 분석하였는가?
	수정 또는 변경의 사용자 영향을 분석하였는가?
	수정 또는 변경의 환경과 그 상호 작용에 대한 영향을 분석하였는가?
	수정 또는 변경 수행에 대한 상세 명세를 수정하여 기술하였는가?
	수정 또는 변경 수행의 테스트에 대한 명세를 수정하여 기술하였는가?
	수정 또는 변경 해당 소프트웨어 유닛을 수정하여 작성하였는가?
	수정 코드의 테스트 가능성을 확인하였는가?
	수정 코드의 코딩 표준 준수 여부를 확인하였는가?
	수정 또는 변경에 대한 재확인을 수행하고 결과를 작성하였는가?
	수정 또는 변경에 대한 회귀 테스트를 수행하고 결과를 작성하였는가?

2.3. 소프트웨어 유지보수 기록

소프트웨어 변경 요청, 수정 요청 또는 문제 보고서 목록 및 변경 영향 평가 결과, 유지보수 활동 시 수집한 측정 데이터와 테스트 결과 및 소프트웨어 형상 이력 등을 소프트웨어 유지보수 기록에 작성한다.

2.3.1. 소프트웨어 유지보수 기록 수행 절차

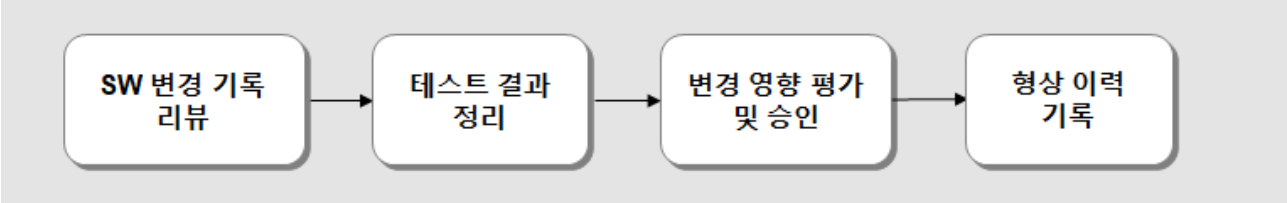


그림 156 소프트웨어 유지보수 기록 수행 절차 흐름도

표 200 소프트웨어 유지보수 기록 절차

항 목	설 명
소프트웨어 변경 기록 리뷰	- 변경 요청, 수정 요청 또는 문제 보고서 목록 등 소프트웨어 변경 기록을 분석한다. - 수정된 시스템의 무결성을 확인한다. - 변경 요구사항, 설계, 코드의 추적성을 검토한다.
소프트웨어 변경 테스트 결과 정리	- 재확인 및 회귀 테스트 데이터를 포함한 컴포넌트에 대한 테스트 케이스를 기술한다.
소프트웨어 변경 영향 평가 및 승인	- 전체 시스템에 대한 소프트웨어 변경의 영향을 평가한다. - 변경 결과 충족에 대한 승인을 획득한다.
소프트웨어 형상 이력 기록	- 소프트웨어 형상 이력을 기록한다.

2.3.2. 소프트웨어 유지보수 기록 지침

○ 소프트웨어 유지보수 기록은 9.2.2의 입력 문서에 기초하여 작성되어야 한다. (9.2.4.7)

○ 소프트웨어 유지보수 기록은 반드시 각각의 소프트웨어 아이템에 대하여 처음 출시되기 전에 수립되어야하고, 유지되어야 한다. 또한 “유지보수 기록 및 보고서 “(ISO / IEC 90003 : 2014 유지보수 참조)에 대한 ISO/IEC 90003:2014의 요구사항을

만족해야 하며, 소프트웨어 유지보수 기록은 반드시 다음의 내용을 포함해야 한다. (9.2.4.8)

- 해당 소프트웨어 아이템에 대한 모든 소프트웨어 변경 기록을 참조
- 변경 영향 평가
- 재확인 및 회귀 테스트 데이터를 포함한 컴포넌트에 대한 테스트 케이스
- 소프트웨어 형상 이력

2.3.3. 소프트웨어 유지보수 기록 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 소프트웨어 변경 수행 결과 정리

2.1 변경 요청 내역 확인

- 수정 요청, 변경 요구사항 또는 문제 보고서 목록 등의 소프트웨어 변경에 대한 근원을 기술한다.

2.2 변경 수행 내역 확인

- 필수 소프트웨어 컴포넌트만 수정되었는지 확인한다.

- 변경 소프트웨어 컴포넌트가 적절히 통합되었는지 확인한다.

2.3 변경 추적성 확인

- 변경 요구사항, 설계, 코드의 추적성을 확인한다.

3. 소프트웨어 변경 테스트 결과 정리

3.1 재확인 테스트 결과 확인

- 수정 또는 변경에 대한 재확인 테스트 케이스를 기술하고 결과를 확인한다.

3.2 회귀 테스트 결과 확인

- 수정 또는 변경에 대한 회귀 테스트 케이스를 기술하고 결과를 확인한다.

4. 소프트웨어 변경 영향 평가 및 승인

4.1 변경 영향 평가

- 전체 시스템에 대한 소프트웨어 변경의 영향을 평가하여 기술한다.

4.2 변경 결과 승인

- 변경 결과 충족에 대한 승인 내역을 한다.

5. 소프트웨어 변경 형상 기록

- 소프트웨어 변경 형상에 대한 이력을 기술한다.

그림 157 소프트웨어 유지보수 기록 템플릿 (예시)

2.3.4. 소프트웨어 유지보수 기록 체크리스트

표 201 소프트웨어 유지보수 기록 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	변경 기록에서 변경 요청 내역을 확인하여 기술하고 있는가?
	필수 소프트웨어 컴포넌트만 수정되었는지 확인하였는가?
	변경 소프트웨어 컴포넌트가 적절히 통합되었는지 확인하였는가?
	변경 요구사항, 설계, 코드의 추적성이 확인되었는가?
	수정 또는 변경에 대한 재확인 테스트 케이스와 결과를 기술하였는가?
	수정 또는 변경에 대한 회귀 테스트 케이스와 결과를 기술하였는가?
	전체 시스템에 대한 소프트웨어 변경의 영향을 평가하여 기술하였는가?
	변경 결과 충족에 대한 승인을 획득 하였는가?
	소프트웨어 변경 형상에 대한 이력이 기술되어 있는가?

2.4. 소프트웨어 유지보수 검증

소프트웨어 유지보수의 일관성, 적합성, 가독성 및 추적성에 대한 일반 요구사항과 각 명세별 특정 요구사항의 충족 여부를 검증한다.

2.4.1. 소프트웨어 유지보수 검증 수행 절차

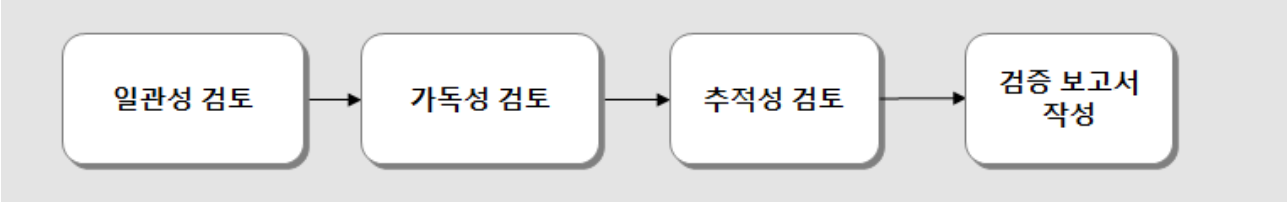


그림 158 소프트웨어 유지보수 검증 흐름도

표 202 소프트웨어 유지보수 검증 보고서 작성 절차

항 목	설 명
유지보수의 일관성	- 소프트웨어 유지보수 계획, 변경 기록, 유지보수 기록의 일관성, 완전성을 검토한다.
유지보수의 가독성	- 소프트웨어 유지보수 계획, 변경 기록, 유지보수 기록의 가독성을 검토한다.
유지보수의 추적성	- 소프트웨어 유지보수 계획, 변경 기록, 유지보수 기록의 추적성을 검토한다.
검증 보고서 작성	- 소프트웨어 유지보수 검증의 충족 여부를 평가하고 결과를 정리한다.

2.4.2. 소프트웨어 유지보수 검증 보고서 작성 지침

○ 소프트웨어 유지보수 검증 보고서는 검증자의 책임 하에 9.2.2의 입력 문서에 근거하여 작성되어야 한다.(9.2.4.11)

○ 소프트웨어 유지보수 계획이 수립되면, 검증은 다음 사항을 다루어야 한다. (9.2.4.12)

- 소프트웨어 유지보수 계획은 가독성 및 추적성에 대한 일반 요구사항과 9.2.4.6의 특정 요구사항을 충족하며,
- 소프트웨어 유지보수 계획서의 내부 일관성.

○ 소프트웨어 유지보수 기록이 수립되면 검증은 다음 사항을 다루어야한다. (9.2.4.13)

- 소프트웨어 유지보수 기록의 가독성 및 추적성에 대한 일반 요구사항과 9.2.4.8의 특정 요구사항 충족 여부
- 소프트웨어 유지보수 기록의 내부 일관성

○ 소프트웨어 변경 기록이 수립되면 검증은 다음 사항을 다루어야한다. (9.2.4.14)

- 소프트웨어 변경 기록의 가독성 및 추적성에 대한 일반 요구사항과 9.2.4.10의 특정 요구사항 충족 여부
- 소프트웨어 변경 기록의 내부 일관성

2.4.3. 소프트웨어 유지보수 검증 보고서 템플릿

문서 정보

발행 부서	
형상 ID	
문서 상태	

개정 이력

버전	날짜	설명	작성자	승인자

1. 개요

- 문서의 전반적인 소개와 내용을 기술한다.

1.1 목적

- 문서의 목적에 대해 간략하게 기술한다.

1.2 용어 설명

- 문서에 사용되는 용어에 대해 기술한다.

1.3 적용 문서

- 적용되는 문서에 대해 기술한다.

1.3.1 인증 표준

- 인증 표준 문서에 대해 기술한다.
- (예시: IEC 62279_2015, EN50128_2001)

1.3.2 프로젝트 문서

- 프로젝트에서 사용하는 인증 표준 이외의 문서
- (예시: 코딩 규칙, 코딩 스타일 가이드)

1.4 참조 문헌

- 참조 문헌을 기술한다.

1.5 약어

- 문서에 포함되어 있는 약어에 대한 설명을 기술한다.

2. 검증 요약

2.1 유지보수 검증 개요

- 유지보수 소프트웨어의 특성 정보 및 구성을 기술한다.

2.2 검증자

- 유지보수 검증자의 이름을 기술한다.

3. 검증 결과 상세

3.1 소프트웨어 유지보수 계획

- 소프트웨어 유지보수 계획의 가독성 및 추적성에 대한 요구사항 충족 여부
- 소프트웨어 유지보수 계획의 내부 일관성을 검증하고 결과를 기술한다.

3.2 소프트웨어 유지보수 기록

- 소프트웨어 유지보수 기록의 가독성 및 추적성에 대한 요구사항 충족 여부
- 소프트웨어 유지보수 기록의 내부 일관성을 검증하고 결과를 기술한다.

3.3 소프트웨어 변경 기록

- 소프트웨어 변경 기록의 가독성 및 추적성에 대한 요구사항 충족 여부
- 소프트웨어 변경 기록의 내부 일관성을 검증하고 결과를 기술한다.

4. 검증 결과

- 소프트웨어 유지보수 검증 결과를 요약하여 기술한다.

그림 159 소프트웨어 유지보수 검증 결과 보고서 템플릿 (예시)

2.4.4. 소프트웨어 유지보수 검증 보고서 체크리스트

표 203 소프트웨어 유지보수 검증 보고서 체크리스트 (예시)

구 분	점검 사항
문서 형식	문서의 버전 및 이력을 기술하였는가?
	문서의 개요와 목적을 기술하고 있는가?
	문서에 사용된 용어 설명 및 약어가 정의되어 있는가?
	적용 문서와 참조한 문서목록이 모두 기술되어 있는가?
요구사항	유지보수 소프트웨어의 특성 정보 및 구성을 기술하고 있는가?
	유지보수 검증자의 이름이 기술되어 있는가?
	소프트웨어 유지보수 계획의 가독성 및 추적성에 대한 요구사항 충족여부를 검증하고 결과를 기술하였는가?
	소프트웨어 유지보수 계획의 내부 일관성을 검증하고 결과를 기술하였는가?
	소프트웨어 유지보수기록의 가독성 및 추적성에 대한 요구사항 충족 여부를 검증하고 결과를 기술하였는가?
	소프트웨어 유지보수기록의 내부 일관성을 검증하고 결과를 기술하였는가?
	소프트웨어 변경 기록의 가독성 및 추적성에 대한 요구사항 충족 여부를 검증하고 결과를 기술하였는가?
	소프트웨어 변경 기록의 내부 일관성을 검증하고 결과를 기술하였는가?
	소프트웨어 유지보수 검증 경과를 요약하여 기술되어 있는가?

제 9 절 기법 및 대책(T&M) 활용 방안

IEC 62279 표준에서는 개발 생명주기 단계에 대하여 적용해야 하는 기법 및 대책을 설명하고 있다. 또한 각 단계에서 소프트웨어 안전 무결성 등급(이하 SIL)에 따라 선택할 수 있는 승인된 조합에 대해서도 설명하고 있다. 따라서 본 절에서는 표준에서 설명하고 있는 개발 생명주기의 6 단계에 대한 기법 및 대책 적용 가이드를 제시한다. 아래의 테이블에서 사용하는 기호들에 대한 설명은 다음과 같다.

표 204 기법 및 대책 기호

기 호	설 명
‘M’	‘M’ 은 기법 및 대책의 사용이 필수임을 의미한다.
‘HR’	‘HR’ 은 기법 및 대책이 해당 안전 무결성 등급에서 많이 권장됨을 의미한다. 해당 기법 또는 대책이 사용되지 않으면 대체 기법 사용에 대한 사유가 소프트웨어 품질 보증 계획 또는 소프트웨어 품질 보증 계획에서 언급된 다른 문서에서 설명되어야 한다.
‘R’	‘R’ 는 이 기술 또는 측정이 해당 안전 무결성 등급에서 권장됨을 의미한다. 이는 ‘HR’ 보다 낮은 수준의 권장 사항이며 이러한 기술을 결합하여 패키지의 일부를 형성할 수 있다.
‘-’	이 기호는 이 기술 또는 측정의 사용을 권장하지 않거나 사용되는 것을 반대한다.
‘NR’	이 기호는 이 기술 또는 측정이 해당 안전 무결성 등급에서 권장되지 않음을 의미한다. 이 기술 또는 측정이 사용되는 경우 근본적인 이유가 소프트웨어 품질 보증 계획 또는 소프트웨어 품질 보증 계획에서 참조하는 다른 문서에서 설명되어야 한다.

1. 소프트웨어 요구사항 명세

표 205 소프트웨어 요구사항 명세 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 정형기법(수학적인 접근법에 기반)	부록.B-28	-	R	R	HR	HR
2. 모델링	7.7 참조	R	R	R	HR	HR
3. 구조적 방법론	부록.B-52	R	R	R	HR	HR
4. 결정 테이블	부록.B-13	R	R	R	HR	HR
요구사항: a) 소프트웨어 요구사항 명세는 자연어로 된 문제에 대한 설명과 필요한 정형 또는 준정형 표기법을 포함해야 한다. b) 이 표는 명세를 명확하고 정확하게 정의하기 위한 추가 요구사항들이 반영되어 있다. 사용 중인 소프트웨어의 안전 무결성 등급을 충족하기 위하여 이러한 기술들 중 하나 이상을 선택해야 한다.						

2. 소프트웨어 아키텍처

표 206 소프트웨어 아키텍처 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 방어적 프로그래밍	부록.B-14	-	HR	HR	HR	HR
2. 결함 검출 & 진단	부록.B-26	-	R	R	HR	HR
3. 오류 정정 코드	부록.B-19	-	-	-	-	-
4. 오류 검출 코드	부록.B-19	-	R	R	HR	HR
5. 고장 단정 프로그래밍	부록.B-24	-	R	R	HR	HR
6. 안전성 백 기법	부록.B-47	-	R	R	R	R
7. 다양화 프로그래밍	부록.B-16	-	R	R	HR	HR
8. 복구 블록	부록.B-44	-	R	R	R	R
9. 역방향 복구	부록.B-5	-	NR	NR	NR	NR
10. 전방향 복구	부록.B-30	-	NR	NR	NR	NR
11. 고장 복구 재시도 방법	부록.B-46	-	R	R	R	R
12. 실행된 사례 기억	부록.B-36	-	R	R	HR	HR
13. 인공지능-결함 정정	부록.B-1	-	NR	NR	NR	NR
14. 소프트웨어의 동적 재구성	부록.B-17	-	NR	NR	NR	NR
15. 소프트웨어 오류 영향 분석	부록.B-25	-	R	R	HR	HR
16. 우아한 저하	부록.B-31	-	R	R	HR	HR
17. 정보 은닉	부록.B-33	-	-	-	-	-
18. 정보 캡슐화	부록.B-33	R	HR	HR	HR	HR
19. 완전하게 정의된 인터페이스	부록.B-38	HR	HR	HR	M	M
20. 정형기법	부록.B-28	-	R	R	HR	HR
21. 모델링	7.7참조	R	R	R	HR	HR
22. 구조적 방법론	부록.B-52	R	HR	HR	HR	HR
23. 컴퓨터 지원 설계 및 명세도구를 통한 모델링	7.7참조	R	R	R	HR	HR

요구사항:

a) 소프트웨어 안전 무결성 등급 3, 4에 대한 기술의 승인된 조합은 다음과 같다.
   1) 1, 7, 19, 22와 4, 5, 12, 21 중 한 가지;
   2) 1, 4, 19, 22와 2, 5, 12, 15, 21 중 한 가지

b) 소프트웨어 안전 무결성 등급 1, 2에 대한 기술의 승인된 조합은 1, 19, 22와 2, 4, 5, 7, 12, 15, 21 중 한 가지이다.

c) 이러한 문제 중 일부는 시스템 수준에서 정의될 수 있다.

d) IEC 62279 요구사항에 따라 오류 검출 코드를 사용할 수 있다.

참고 기법 및 대책 19는 외부 인터페이스 용도로 사용된다.

소프트웨어 아키텍처 단계에서는 다음과 같은 승인된 기술 조합이 있다. SIL 등급에 따라서 선택하여 적용할 수 있다.

○ SIL 3, 4 등급

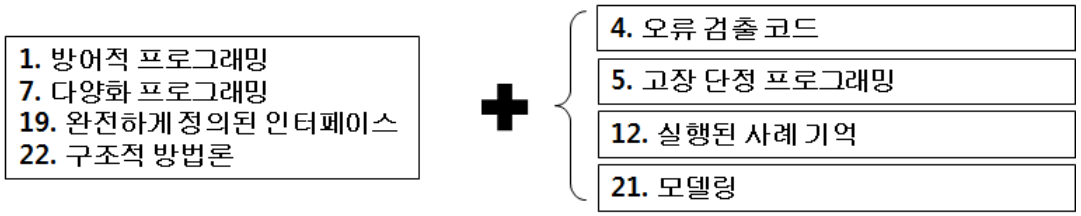


그림 160 소프트웨어 아키텍처 단계 기법 및 대책 SIL 3, 4 등급 적용 방법 1

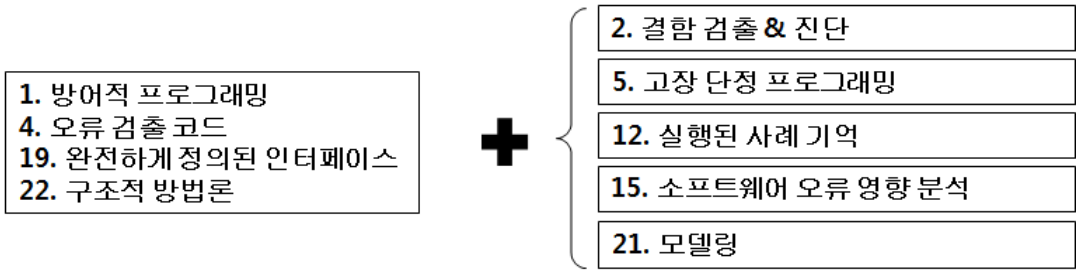


그림 161 소프트웨어 아키텍처 단계 기법 및 대책 SIL 3, 4 등급 적용 방법 2

○ SIL 1, 2 등급

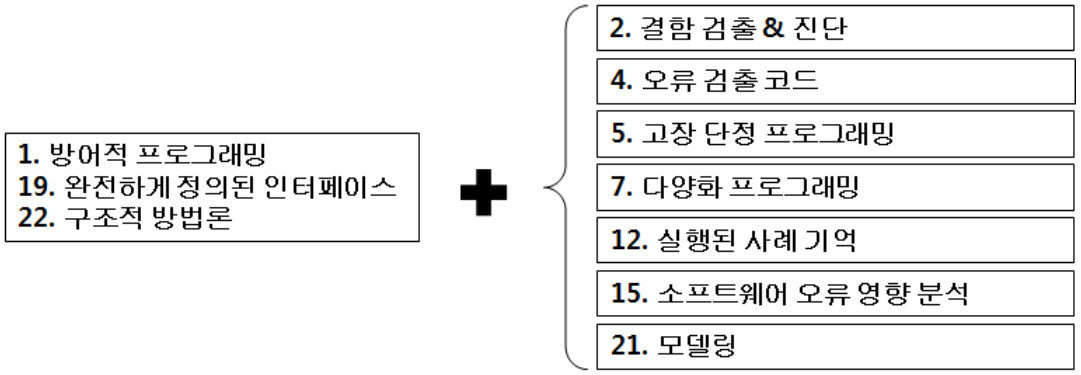


그림 162 소프트웨어 아키텍처 단계 기법 및 대책 SIL 1, 2 등급 적용 방법

3. 소프트웨어 설계 및 구현

표 207 소프트웨어 설계 및 구현 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 정형 기법	부록.B-28	-	R	R	HR	HR
2. 모델링	표 A.17	R	HR	HR	HR	HR
3. 구조적 방법론	부록.B-52	R	HR	HR	HR	HR
4. 모듈 방식	부록.B-38	HR	M	M	M	M
5. 컴포넌트	7.7참조	HR	HR	HR	HR	HR
6. 설계 및 코딩 표준	7.7참조	HR	HR	HR	M	M
7. 분석 가능한 프로그램	부록.B-2	HR	HR	HR	HR	HR

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
8. 엄격한 형식의 프로그래밍 언어	부록.B-49	R	HR	HR	HR	HR
9. 구조적 프로그래밍	부록.B-53	R	HR	HR	HR	HR
10. 프로그래밍 언어	7.7참조	R	HR	HR	HR	HR
11. 언어 하위집합	부록.B-35	-	-	-	HR	HR
12. 객체지향 프로그래밍	7.7참조, 부록.B-57	R	R	R	R	R
13. 절차적 프로그래밍	부록.B-60	R	HR	HR	HR	HR
14. 메타프로그래밍	부록.B-59	R	R	R	R	R
요구사항: a) 소프트웨어 안전 무결성 등급 3, 4에 대해 승인된 기술 조합은 4, 5, 6, 8과 1, 2 중 한 가지이다. b) 소프트웨어 안전 무결성 등급 1, 2에 대한 승인된 기술 조합은 3, 4, 5, 6과 8, 9, 10 중 한 가지이다. c) 메타프로그래밍은 컴파일하기 전 소프트웨어 소스 코드의 생성으로 제한되어야 한다.						

소프트웨어 설계 및 구현 단계에서는 다음과 같은 승인된 기술 조합이 있다. SIL 등급에 따라서 선택하여 적용할 수 있다.

○ SIL 3, 4 등급

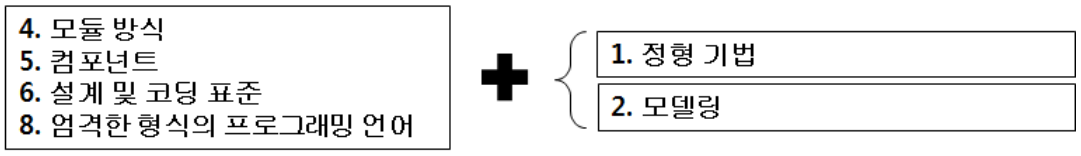


그림 163 소프트웨어 설계 및 구현 단계 기법 및 대책 SIL 3, 4 적용 방법

○ SIL 1, 2 등급

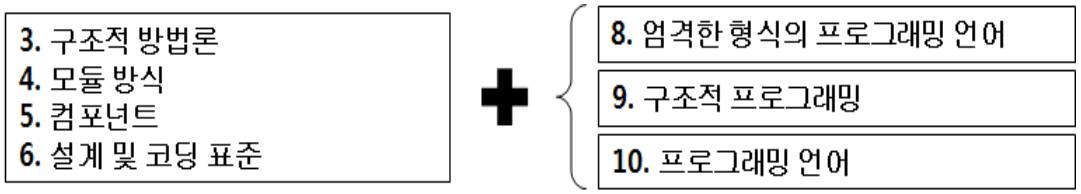


그림 164 소프트웨어 설계 및 구현 단계 기법 및 대책 SIL 1, 2 적용 방법

4. 검증 및 시험

표 208 검증 및 시험 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 정형 증명	부록.B-29	-	R	R	HR	HR
2. 정적 분석	7.7참조	-	HR	HR	HR	HR
3. 동적 분석 및 시험	7.7참조	-	HR	HR	HR	HR
4. 측정 기준	부록.B-37	-	R	R	R	R
5. 추적성	부록.B-58	R	HR	HR	M	M
6. 소프트웨어 오류 영향 분석	부록.B-25	-	R	R	HR	HR
7. 코드 시험 적용범위	7.7참조	R	HR	HR	HR	HR
8. 기능적/블랙박스 시험	7.7참조	HR	HR	HR	M	M
9. 성능 시험	7.7참조	-	HR	HR	HR	HR
10. 인터페이스 시험	부록.B-34	HR	HR	HR	HR	HR
요구사항: a) 소프트웨어 안전 무결성 등급 3, 4의 경우 승인된 기법 조합은 3, 5, 7, 8과 1, 2, 6 중 한 가지이다. b) 소프트웨어 안전 무결성 등급 1, 2의 경우 승인된 기법 조합은 5와 2, 3, 8 중 한 가지이다. 참고1: 기법 및 대책 1, 2, 4, 5, 6, 7은 검증 활동을 위한 것이다. 참고2: 기법 및 대책 3, 8, 9, 10은 시험 활동을 위한 것이다.						

검증 및 시험 단계에서는 다음과 같은 승인된 기술 조합이 있다. SIL 등급에 따라서 선택하여 적용할 수 있다.

○ SIL 3, 4 등급

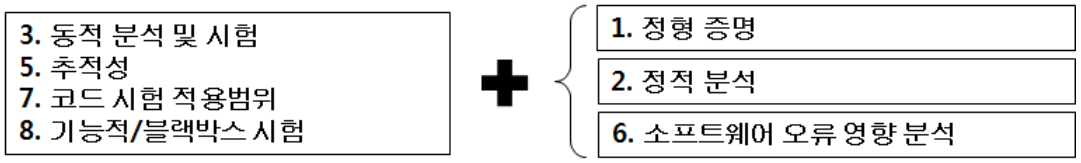


그림 165 검증 및 시험 단계 기법 및 대책 SIL 3, 4 적용 방법

○ SIL 1, 2 등급

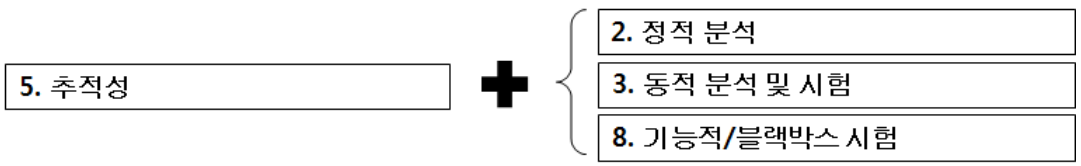


그림 166 검증 및 시험 단계 기법 및 대책 SIL 1, 2 적용 방법

5. 통합

표 209 통합 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 기능 및 블랙박스 시험	7.7참조	HR	HR	HR	HR	HR
2. 성능 시험	7.7참조	-	R	R	HR	HR

6. 종합 소프트웨어 시험

표 210 종합 소프트웨어 시험 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 성능 시험	7.7참조	-	HR	HR	M	M
2. 기능 및 블랙박스 시험	7.7참조	HR	HR	HR	M	M
3. 모델링	7.7참조	-	R	R	R	R
요구사항: a) 소프트웨어 안전 무결성 등급 1, 2의 경우 승인된 기술조합은 1, 2 이다.						

7. 그 외 기법 및 대책

표 211 소프트웨어 분석 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 정적 소프트웨어 분석	부록.B-13 부록.B-37 7.7참조	R	HR	HR	HR	HR
2. 동적 소프트웨어 분석	7.7참조	-	R	R	HR	HR
3. 원인 결과 다이어그램	부록.B-6	R	R	R	R	R
4. 이벤트 트리 분석	부록.B-22	-	R	R	R	R
5. 소프트웨어 오류 영향 분석	부록.B-25	-	R	R	HR	HR
요구사항: a) 하나 이상의 기법들은 사용되는 소프트웨어의 SIL 등급을 만족시키기 위해 선택되어야 한다.						

표 212 소프트웨어 품질 보증 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. ISO 9001 인증		R	HR	HR	HR	HR
2. ISO 9001 준수		M	M	M	M	M
3. ISO/IEC 90003 준수		R	R	R	R	R
4. 회사 품질 시스템		M	M	M	M	M
5. 소프트웨어 형상 관리	부록.B-48	M	M	M	M	M
6. 체크리스트	부록.B-7	R	HR	HR	HR	HR
7. 추적성	부록.B-58	R	HR	HR	M	M
8. 데이터 기록 및 분석	부록.B-12	HR	HR	HR	M	M
요구사항: a) 테이블은 모든 단계의 담당자들에게 적용된다.						

표 213 소프트웨어 유지보수 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 영향 분석	부록.B-32	R	HR	HR	M	M
2. 데이터 기록 및 분석	부록.B-12	HR	HR	HR	M	M

표 214 데이터 준비 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 테이블 명세 기법	부록.B-68	R	R	R	R	R
2. 응용 특화 언어	부록.B-69	R	R	R	R	R
3. 시뮬레이션	부록.B-42	R	HR	HR	HR	HR
4. 기능 테스트	부록.B-42	M	M	M	M	M
5. 체크리스트	부록.B-7	R	HR	HR	M	M
6. 페이건 정밀 검사	부록.B-23	-	R	R	R	R
7. 정형 디자인 리뷰	부록.B-56	R	HR	HR	HR	HR
8. 정형 증명 (데이타)	부록.B-29	-	-	-	HR	HR
9. 워크스루	부록.B-56	R	R	R	HR	HR
요구사항: a) 소프트웨어 SIL 1및 2에서 승인된 기법의 조합은 1과 4이다. b) 소프트웨어 SIL 3및 4에서 승인된 기법의 조합은 1, 4, 5와 7 또는 2, 3, 6이다. 참고: B-29에 대한 설명은 프로그램에 관한 것이고, 이 맥락에서 기법 8은 데이터의 정확성에 대해서 정형 증명을 적용한다는 것이다.						

표 215 코딩 표준 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 코딩 표준	부록.B-15	HR	HR	HR	HR	HR
2. 코딩 스타일 가이드	부록.B-15	HR	HR	HR	HR	HR
3. 동적 객체 사용 금지	부록.B-15	-	R	R	HR	HR
4. 동적 변수 사용 금지	부록.B-15	-	R	R	HR	HR
5. 포인터 사용 제한	부록.B-15	-	R	R	R	R
6. 재귀호출 사용 제한	부록.B-15	-	R	R	HR	HR
7. 조건 없는 점프 사용 금지	부록.B-15	-	HR	HR	HR	HR
8. 함수, 서브루틴과 메소드의 크기와 복잡도 제한	부록.B-38	HR	HR	HR	HR	HR
9. 함수, 서브루틴에 대한 진입/종료 시점 전략 및 방법	부록.B-38	R	HR	HR	HR	HR
10. 서브루틴 인자 수 제한	부록.B-38	R	R	R	R	R
11. 전역 변수 사용 제한	부록.B-38	HR	HR	HR	M	M
요구사항: a) 기법 3, 4 그리고 5번은 확인 된 컴파일러 또는 변환기의 일부로 제공 될 수 있다.						

표 216 동적 분석 및 시험 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 경계값 분석으로부터 테스트 케이스 수행	부록.B-4	-	HR	HR	HR	HR
2. 오류 추축으로부터 테스트 케이스 수행	부록.B-20	R	R	R	R	R
3. 오류 삽입으로부터 테스트 케이스 수행	부록.B-21	-	R	R	R	R
4. 성능 모델링	부록.B-39	-	R	R	HR	HR
5. 동등 클래스 및 입력 분할 테스팅	부록.B-18	R	R	R	HR	HR
6. 구조 기반 시험	부록.B-50	-	R	R	HR	HR
요구사항: a) 테스트 케이스에 대한 분석은 하위 시스템 수준에서 이루어지며 사양 및/또는 코드를 기반으로 한다.						

표 217 기능/블랙박스 테스트 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 원인 결과 다이어그램으로부터 테스트 케이스 수행	부록.B-6	-	-	-	R	R
2. 프로토타입/애니메이션	부록.B-43	-	-	-	R	R
3. 경계값 분석	부록.B-4	R	HR	HR	HR	HR
4. 동등 클래스 및 입력 분할 테스팅	부록.B-18	R	HR	HR	HR	HR
5. 프로세스 시뮬레이션	부록.B-42	R	R	R	R	R
요구사항: a) 시뮬레이션의 완성도는 소프트웨어 안전 무결성 등급, 복잡성 및 응용 프로그램의 범위에 의존한다.						

표 218 프로그래밍 언어 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. ADA	부록.B-54	R	HR	HR	HR	HR
2. MODULA-2	부록.B-54	R	HR	HR	HR	HR
3. PASCAL	부록.B-54	R	HR	HR	HR	HR
4. C or C++	부록.B-54 부록.B-35	R	R	R	R	R
5. PL/M	부록.B-54	R	R	R	NR	NR
6. BASIC	부록.B-54	R	NR	NR	NR	NR
7. Assembler	부록.B-54	R	R	R	R	R
8. C#	부록.B-54 부록.B-35	R	R	R	R	R
9. JAVA	부록.B-54 부록.B-35	R	R	R	R	R
10. Statement List	부록.B-54	R	R	R	R	R

요구사항:
a) 언어의 선택은 표준 6.7과 7.3의 요구사항을 기반으로 한다.
b) 특정 프로그래밍 언어를 배제하기 위해 취한 결정을 정당화해야 할 요구사항은 없다.
참고1: 프로그래밍 언어의 적합성 평가에 대한 자세한 내용은 부록.B-54 적합한 프로그래밍 언어를 참조하십시오.
참고2: 만약에 특정 프로그래밍 언어가 표에 없다고 해서 자동으로 제외되는 것이 아니고 부록.B-54를 따른다.
참조3: 응용 프로그램을 실행하는 데 필요한 선택된 언어와 관련된 런-타임 시스템은 소프트웨어 안전 무결성 등급에 따라 여전히 사용에 대한 정당성이 확보되어야 한다.

표 219 어플리케이션 알고리즘을 위한 다이어그램 언어 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 기능적 블록 다이어그램	부록.B-63	R	R	R	R	R
2. 순차적 함수 도표(차트)	부록.B-61	-	HR	HR	HR	HR
3. 래더 다이어그램	부록.B-62	R	R	R	R	R
4. 상태 차트	부록.B-64	R	HR	HR	HR	HR

표 220 모델링 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 데이터 모델링	부록.B-65	R	R	R	HR	HR
2. 데이터 흐름 다이어그램	부록.B-11	-	R	R	HR	HR
3. 제어 흐름 다이어그램	부록.B-66	R	R	R	HR	HR
4. 유한 상태 기계 / 상태 전이 다이어그램	부록.B-27	-	HR	HR	HR	HR
5. 시간 패트리넷	부록.B-55	-	R	R	HR	HR
6. 결정/진리 테이블	부록.B-13	R	R	R	HR	HR
7. 정형 기법들	부록.B-28	-	R	R	HR	HR
8. 성능 모델링	부록.B-39	-	R	R	HR	HR
9. 프로토타입/애니메이션	부록.B-43	-	R	R	R	R
10. 구조 다이어그램	부록.B-51	-	R	R	HR	HR
11. 순차 다이어그램	부록.B-67	R	HR	HR	HR	HR
요구사항: a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나의 HR 기법을 선택해야 한다.						

표 221 성능 시험 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 과부하/스트레스 시험	부록.B-3	-	R	R	HR	HR
2. 응답 시기 및 메모리 제약	부록.B-45	-	HR	HR	HR	HR
3. 성능 요구사항	부록.B-40	-	HR	HR	HR	HR

표 222 정정 분석 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 경계값 분석	부록.B-4	-	R	R	HR	HR
2. 체크리스트	부록.B-7	-	R	R	R	R
3. 제어 흐름 분석	부록.B-8	-	HR	HR	HR	HR
4. 데이터 흐름 분석	부록.B-10	-	HR	HR	HR	HR
5. 오류 추측	부록.B-20	-	R	R	R	R
6. 워크스루/설계 검토	부록.B-56	HR	HR	HR	HR	HR

표 223 컴포넌트 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 정보 은닉	부록.B-4	-	-	-	-	-
2. 정보 캡슐화	부록.B-7	R	HR	HR	HR	HR
3. 파라미터 수 제한	부록.B-8	R	R	R	R	R
4. 모든 인터페이스 정의	부록.B-10	R	HR	HR	M	M
요구사항: a) 정보 은닉과 캡슐화는 데이터 접근에 대해 일반적인 전략이 없다면 오직 HR이다. 참고 4번은 내부 인터페이스들을 위한 것이다.						

표 224 코드 테스트 커버리지 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 구문	부록.B-50	R	HR	HR	HR	HR
2. 분기	부록.B-50	-	R	R	HR	HR
3. 복합 조건	부록.B-50	-	R	R	HR	HR
4. 데이터 흐름	부록.B-50	-	R	R	HR	HR
5. 경로	부록.B-50	-	R	R	HR	HR
요구사항: a) 모든 SIL에 대해 정량화된 평가 척도는 수행된 시험에 대해 개발되어야 한다. 이것은 시험에서 얻은 신뢰성과 추가 기술에 대한 필요성을 입증할 수 있다. b) 컴포넌트 단계에서 SIL 3 또는 4에 대한 테스트 커버리지는 다음과 같이 측정되어야 한다. - 기법 2와 3; 또는 다른 기법 - 기법 2와 4; 또는 다른 기법 - 기법 5 또는 통합 단계에서 테스트 커버리지는 2, 3, 4 또는 5 중 하나 이상에 따라 측정해야 한다. c) 다른 테스트 커버리지 기준은 타당함이 입증되면 사용할 수 있다. 이들 기준은 소프트웨어 아키텍처와 프로그래밍 언어에 의존한다. d) 테스트할 수 없는 코드는 적합한 방법으로 시연해야 합니다. (예. 정적 분석 기법 및 대책) 참고 1: 구문 커버리지는 기법 2 부터 기법5를 달성하면 자동적으로 만족한다. 참고 2: 표의 테스트 커버리지 기준은 구조-기반(코드기반, 화이트 박스)의 테스트를 위해 사용된다. 기능(명세기반, 블랙박스) 테스트를 위한 기술/측정은 표 A.14 에서 다룬다. 참고 3: 높은 퍼센트의 커버리지는 일반적으로 달성하기 어렵다. 경계값(부록.B-4)과 등가 클래스 및 입력 파티션 테스트(부록.B-18)의 테스트 케이스 사용은 적은 수의 테스트로 충분한 커버리지를 달성할 수 있도록 한다. 참고 4: 기법 2와 3의 차이점은 실제 프로그래밍 언어의 수준과 다중 조건의 사용에 따라 결정된다. 예를 들어 단일 조건만을 사용 할 경우 기법 2와 3의 결과는 같은 것으로 고려한다.						

표 225 객체지향 소프트웨어 아키텍처 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 아키텍처 클래스들에 대한 어플리케이션 도메인 컨셉의 추적성	-	R	R	R	HR	HR
2. 일반적으로 사용되는 클래스들과 디자인패턴의 조합과 같은 적절한 프레임 사용	-	R	R	R	HR	HR
3. 객체지향 상세 설계	7.7참조	R	R	R	HR	HR
요구사항: a) 기존 프레임 및 디자인 패턴을 사용할 때 기존(Pre-existing) 소프트웨어의 요구사항은 이러한 프레임과 패턴을 적용한다. 참고1: 객체 지향 접근법은 절차적 접근법과는 다르게 정보를 제공하며, 다음 목록에는 특정한 고려 사항이 필요한 권고안이 수록되어 있다. - 클래스 계층 구조, 지정된 메소드를 사용할 때 호출되는 메소드의 호출에 대한 소프트웨어 기능을 식별합니다(기존 클래스 라이브러리를 사용할 때). - 구조 기반 시험 (7.7 참조) 애플리케이션 도메인에서부터 클래스 아키텍처로의 추적성은 덜 중요합니다. 참고2: 계획 된 소프트웨어의 일부에 대해서 유사한 작업을 성공적으로 해결하고 개발자들에게 잘 알려져 있는 기존의 소프트웨어의 프레임이 존재한다면, 해당 프레임을 사용하는 것은 좋은 관행이 될 수 있다.						

표 226 객체지향 상세 설계 기법 및 대책

기법 및 대책	참조	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. 클래스들은 오직 하나의 목표만 가진다.	-	R	R	R	HR	HR
2. 파생 클래스가 기본 클래스에 의해 구체화 되는 경우에만 상속이 사용된다.	-	R	HR	HR	HR	HR
3. 상속의 깊이는 코딩 표준에 의해 제한된다.	-	R	R	R	HR	HR
4. 메소드의 상속은 엄격한 제어 하에 수행	-	R	R	R	HR	HR
5. 다중 상속은 오직 인터페이스 클래스들에만 사용된다.	-	R	HR	HR	HR	HR
6. 알 수 없는 클래스로 부터의 상속	-	-	-	-	NR	NR
요구사항: a) 하나의 클래스는 하나의 책임을 가진다. 즉, 긴밀하게 연결된 데이터와 이러한 데이터에 대한 작업을 관리하는 것입니다. b) 객체간의 선형 참조를 방지하기 위해 주의해야 한다.						

제 5 장 가이드 적용 사례

제 1 절 가이드 실무 적용 사례

1. 실무 적용 개요

1.1. 가이드 실무 적용 목표

○ 개발 된 시스템 안전성 분석 가이드와 철도 소프트웨어 개발가이드를 현장에 실무적용 하여 사용성 및 내용을 검증한다.

○ 철도분야 소프트웨어 개발업체의 표준 기반 소프트웨어 엔지니어링 역량을 제고한다.

1.2. 적용 대상 및 수행 방법

○ 가이드 적용 대상 시스템

- 차상표시장치(MMI) 및 비상방송시스템

○ 적용 가이드

- 시스템 안전성 분석 가이드
- 소프트웨어 개발가이드
- 기법 및 대책(T&M) 적용 가이드

2. 적용대상 시스템: MMI 시스템

2.1. 시스템 개요

MMI 시스템은 차상 신호장치의 상태를 표시하는 장치이다. MMI 시스템에 표시되는 화면의 종류는 장치 부팅 후 바로 표시되는 상태 표시 화면과 사용자의 요구에 의하여 표시되는 설정화면, 수신 데이터 및 경보 메시지 화면 등으로 구분되어 진다. MMI 시스템은 차상 ATP장치(편성 2중계)에 LAN으로 접속하여, 지정된 주기로 통신하며, [그림 167]의 시스템 구성도와 같이 차상신호장치, 외부 입출력장치 및 차량용 전원장치와 인터페이스 한다. 차상장치와 MMI 시스템간의 통신 사양의 세부 규격은 인터페이스 명세서를 준수 하여야 한다.

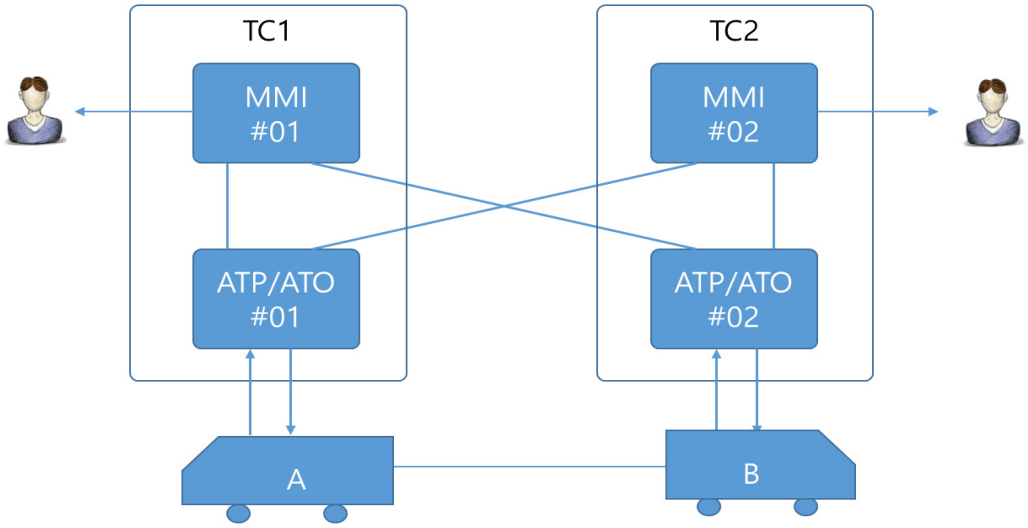


그림 167 MMI 시스템 개요

2.2. MMI 시스템의 안전성 분석 수행을 위한 관련 산출물의 식별

MMI 시스템의 FMEA를 수행하기 위해서는 크게 두 가지 관점에서 산출물을 필요로 한다. 하나는 물리적 구성, 다른 하나는 해당 컴포넌트가 지니고 있는 기능적 정보를 필요로 한다. FMEA 수행에 있어서 우선, 대상 시스템의 물리적 구성도를 식별해야 한다. 식별된 물리적 구성도 정보는 업체가 보유한 아키텍처 산출물을 기반으로 활용될 수 있다.

하지만, 많은 업체들이 아키텍처 산출물에 대해서 가지고 있지 않을 시에는 대상 시스템에 대한 운용개념을 바탕으로 큰 틀의 구성도를 기반으로 활용할 수 있을 것이다. 기능적 관점에서는 해당 물리적 구성이 지니고 있는 기능을 식별해야 한다. 이는 시스템 사양서가 있을 시 이를 활용 가능하다. 대상시스템이 지니고 있는 운용개념도 및 운용개념이 있다면, 이를 기반으로 대상 시스템의 내-외부 인터페이스 정보를 식별할 수 있다. 인터페이스 정보는 FMEA 수행 시 해당 물리적 컴포넌트의 오류 발생 시 타 컴포넌트에 미치는 간섭 영향에 대한 정보를 제공한다.

2.3. MMI 시스템의 운용요구사항 식별

운용 요구사항을 식별할 때는 대상 시스템의 운용 모드에 따라 식별되어야 하며, 운용 요구사항을 기반으로 상위수준에서의 물리적 구성 및 기능, 인터페이스 정보를 식별할 수 있다. 이를 통해, 업체가 해당 정보를 보유하지 않았다면 이러한 방법으로 초기 요구사항을 만들어 안전성 분석에 활용될 수 있다. 운용요구사항을 식별하기 위해서는 MMI 시스템의 동작 모드 별 분석을 수행하였다. MMI 시스템의 동작모드는 아래와 같이, 3가지 동작 모드가 식별되었다. 다음은 운용 모드 별 요구사항을 기술하였다.

○ Active Mode [주행 중(Speed >0)
- MMI 시스템은 OATP/ATO로부터 속도 정보를 수신할 수 있어야 한다.
- MMI 시스템은 Active 모드일 때, 속도가 0보다 크면 주행모드로 인식 하여야 한다.
- MMI 시스템은 주행 중에는 현시모드만 지원되고 조작이 불가능해야 한다.
- MMI 시스템은 주행 중에 다른 모드(슬립모드)로 변환되어서는 안 된다.
- MMI 시스템은 주행 중에 정지중 모드로 변환되어서는 안 된다.

○ 정차 중(Speed=0)
- MMI 시스템은 OATP/ATO로부터 속도 정보를 수신할 수 있어야 한다.
- MMI 시스템은 OATP/ATO로부터 속도 정보를 수신하고 모드 변환을 할 수 있어야 한다.
- MMI 시스템은 차량 정지 상태에서만 시스템 관리 및 설정을 제어할 수 있다.
- MMI 시스템은 차량 정차 중에는 주행모드로 전환되어서는 안 된다.

○ Sleep Mode
- MMI 시스템은 ATP로부터 슬립모드로 변환되도록 상태변화 신호를 받아야 한다.
- MMI 시스템은 슬립 모드 시 설정 기능이 차단되어야 한다.
- MMI 시스템은 슬립 모드 시 화면 터치 기능을 지원해야 한다.
- MMI 시스템은 슬립 모드 시 일정 시간이 지난 후에는 화면이 꺼져야 한다.

2.4. MMI 시스템의 시스템 안전성 분석 수행

2.4.1. FMEA Step. 1_MMI 시스템의 구조 분석

FMEA 수행에 있어서 우선 MMI 시스템의 구조적 정보를 식별하였다. 식별된 정보는 위에서 언급한 운용개념을 기반으로 보완되었다. MMI 시스템의 구조적 정보를 식별하기 위해 시범 적용업체가 가지고 있는 물리적 구성도를 기반으로 시스템 수준(Level)에 따라 분할하려고 했지만, FMEA를 정확하게 수행하기 위해서는 대상 시스템에 대한 계층적 구조도가 필요하므로 시범 적용업체가 보유한 제작 사양서를 기반으로 내용을 보완하여 [그림 168]의 형태의 시스템 계층 구조도를 식별하였다.

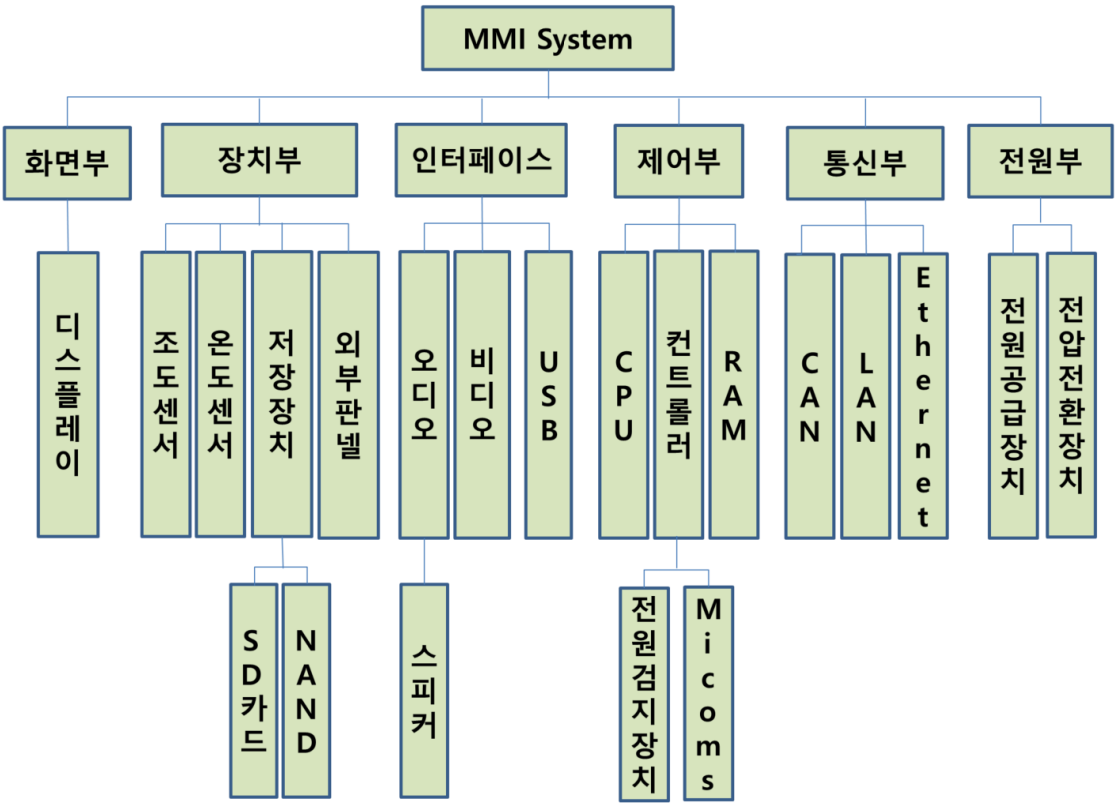


그림 168 MMI 시스템 계층 구조도

위 그림과 같이, 식별된 MMI 시스템 계층 구조도를 기반으로 도구 기반의 FMEA 수행까지 병행한 결과를 아래 그림과 같이, 도구기반의 FMEA 구조분석 결과를 작성하였다.

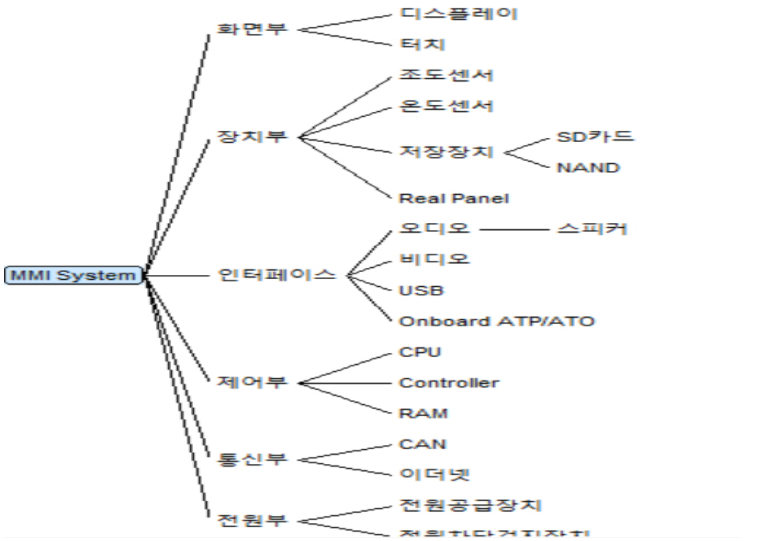


그림 169 구조네트워크 구축

2.4.2. FMEA Step. 2_MMI 시스템의 기능적 정보 식별(컴포넌트/모드별 기능 식별)

선행 활동을 통해, 컴포넌트가 식별되었다면, 해당 컴포넌트가 지니고 있는 고장모드가 식별되어야 한다. 이러한 해당 물리적 컴포넌트가 지니고 있는 고장 모드가 식별되기 위해서는 컴포넌트가 지니고 있는 동작모드 식별이 선행되어야 한다. 일반적인 설계 관점에서 시스템을 구성하는 물리적 컴포넌트의 동작모드의 식별은 운용개념 및 기능을 중심으로 식별된다. 운용요구사항을 기반으로 분석된 기능을 중심으로 상위 요구사항을 분석하여 도출하고, 해당 상위 요구사항으로부터 MMI 시스템이 지닌 하위 세부 요구사항을 명세하였다. 명세 된 요구사항은 위의 구조분석 결과의 범주에 맞게 컴포넌트별 지니고 있는 기능을 나눠 구분되어야 한다. 도출된 요구사항을 기반으로 요구사항의 모드를 구분하였다. 최종적으로 MMI 시스템이 지닌 하부 컴포넌트 별 기능 정보를 기반으로 모드를 나누게 된다.

초기요구사항은 MMI 시스템의 운용개념을 바탕으로 식별된 최상위 요구사항이며, 이를 기반으로 보다 요구사항이 상세 동작환경에서의 운용개념을 바탕으로 보다 상세히 구체화 된다. 따라서 [표 227]과 같이, 최상위 요구사항 5가지가 식별되어 이후 식별된 요구사항을 바탕으로 요구사항 상세 명세화가 진행되었다.

표 227 초기 요구사항(Initial Requirements)

요구사항 ID	요구사항
Initial_Req1	MMI장치는 차량의 운행상태 및 통신 상태를 표시할 수 있어야 한다.
Initial_Req2	MMI장치는 차량의 열차정보 표시 및 시스템 설정을 수행할 수 있는 기능을 가져야 한다.
Initial_Req3	MMI장치는 차량의 통신 데이터 조회를 할 수 있는 기능을 가져야 한다.
Initial_Req4	MMI장치는 시스템 환경 설정 및 소프트웨어 유지/관리를 위한 소프트웨어 업로드 기능을 수행해야 한다.
Initial_Req5	MMI 시스템은 자체 전원공급 또는 전원 차단 시 일정시간 전원공급이 유지되어야 한다.

○ Top Level_Function Requirements

- Initial_Req1_MMI로부터 장치는 차량의 운행상태 및 통신 상태를 표시할 수 있어야 한다.

표 228 초기 요구사항 Initial_Req1 기반 요구사항

요구사항 ID	요구사항
TFRS_01	MMI 시스템은 Target Distance 정보, 운행속도, 운전모드, 출입문 방향, 역정보, 운행방향에 대한 정보가 표시 되어야 한다.
TFRS01_01	MMI 시스템은 운행속도 정보를 표시 할 수 있어야 한다.
TFRS01_02	MMI 시스템은 다음 정차역의 출입문 열림 방향에 대한 정보를 표시하여야 한다.

- Initial_Req2_MMI 시스템은 차량의 열차정보 표시 및 시스템 설정을 수행할 수 있는 기능을 가져야 한다.

표 229 초기 요구사항 Initial_Req2 기반 요구사항

요구사항 ID	요구사항
TFRS_01	MMI 시스템은 운행기록, 시각표시, 설정 취급 메뉴 버튼이 위치하여야 한다.

요구사항 ID	요구사항
TFRS_02	MMI 시스템은 자동밝기 조절 기능이 수행되어야 한다.
TFRS_03	MMI 시스템은 휘도 조절이 가능한 화면 조절 기능을 수행해야 한다.
TFRS03_01	MMI 시스템은 취급자에 의한 자동 휘도 조절 모드를 지원해야 한다.
TFRS03_02	MMI 시스템은 취급자에 의한 수동 휘도 조절 모드를 지원해야 한다.
TFRS_04	MMI 시스템은 표시 상태 조절 기능을 수행해야 한다.
TFRS_05	MMI 시스템은 차량에서 이벤트 발생에 대한 경고음을 표출해야 한다.
TFRS_06	MMI 시스템은 장치 내 전원을 활용해 시스템 시간이 관리되어야 한다.
TFRS_07	MMI 시스템은 장치의 시간과 표준시각과 시간의 동기화가 수행되어야 한다.
TFRS_07_01	MMI 시스템은 차량의 운영 중의 표준 시각에 대한 정보수신은 장치의 시간과 표준시각과 시간의 동기화를 통해 일치 되어야 한다.
TFRS_08	MMI 시스템은 전원공급 중단 시 일정시간 전원을 유지 할 수 있어야 한다.
TFRS_09	MMI 시스템은 기록 메모리를 보유 하여야 한다.
TFRS_10	MMI 시스템의 기록 메모리는 메모리만 교체할 수 있도록 편의성을 갖추어야 한다.
TFRS_11	MMI 시스템은 화면에 표시되는 항목별 색상을 달리 표현할 수 있어야 한다.
TFRS_11_01	MMI 시스템의 화면에 표시되는 심볼은 차상장치로부터 수신한 데이터의 상태에 따라 색상을 달리 표현할 수 있어야 한다.
TFRS_12	MMI 시스템은 Target speed pointer는 목표속도 변경 시 현재속도를 목표속도로 유도하기 위해 준수해야 할 속도를 화살표로 표시할 수 있어야 한다.
TFRS_13	MMI 시스템의 제한속도(Limit Speed)는 현재 운행 중인 구간의 제한속도를 나타내는 모형으로 표시 되어야 한다.
TFRS_14	MMI 시스템은 차량의 운영모드를 표시할 수 있어야 한다.
TFRS_15	MMI 시스템은 차량의 열차스케줄에 따른 운행방향을 표시하여야 한다.
TFRS_16	MMI 시스템은 Position Active와 Proximity Stop 정보를 표시하여야 한다.
TFRS_17	MMI 시스템은 ATP의 상태정보를 수신해야 한다.
TFRS_18	MMI 시스템은 제동상태에 정보를 수신해야 한다.
TFRS_19	MMI 시스템은 차상장치에서 추진/제동 정보를 수신해야 한다.

요구사항 ID	요구사항
TFRS_20	MMI 시스템은 차상장치에서 수신한 정보를 표시해야 한다.
TFRS_21	MMI 시스템은 차상장치로부터 현재시각 및 송수신 상태를 수신 가능해야 한다.

- Initial_Req3_MMI 시스템은 차량의 통신 데이터 조회를 할 수 있는 기능을 가져야 한다.

표 230 초기 요구사항 Initial_Req3 기반 요구사항

요구사항 ID	요구사항
TFRS_22	MMI 시스템은 AP의 연결 상태 정보를 아이콘으로 표시하여야 한다.

- Initial_Req4_MMI시스템은 시스템 환경 설정 및 소프트웨어 유지/관리를 위한 소프트웨어 업로드 기능을 수행해야 한다.

표 231 초기 요구사항 Initial_Req4 기반 요구사항

요구사항 ID	요구사항
TFRS_23	MMI 시스템은 시스템 관리 및 설정 기능을 지원해야 한다.
TFRS_23_1	MMI 시스템은 Driver ID 설정 기능을 지원해야 한다.
TFRS_23_2	MMI 시스템은 Train Number 설정 기능을 지원해야 한다.
TFRS_23_3	MMI 시스템은 WATP IP 설정 기능을 지원해야 한다.
TFRS_23_4	MMI 시스템은 Train Data 설정 기능을 지원해야 한다.
TFRS_23_5	MMI 시스템은 Wheel Size 설정 기능을 지원해야 한다.
TFRS_24	MMI 시스템은 취급(SOM, EOM, PDT Start/stop)기능을 지원해야 한다.
TFRS_25	MMI 시스템은 정보 표시(수신 데이터 스트림 및 수신 이벤트 메시지 표시) 기능을 지원해야 한다.
TFRS_26	MMI 시스템은 MMI(휘도, 음량, 시각, CAB ID) 설정 기능을 지원해야 한다.
TFRS_27	MMI 시스템은 MMI 시스템의 기능 개선 또는 변경을 위하여 프로그램의 업데이트 되도록 구현되어야 한다.

본 단계에서는 동작 모드를 도출하기 위해서 최상위 요구사항 및 보다 상세화 된 요구사항의 분석을 통해서 MMI 시스템이 지니고 있는 동작모드를 아래와 같이 식별하였다.

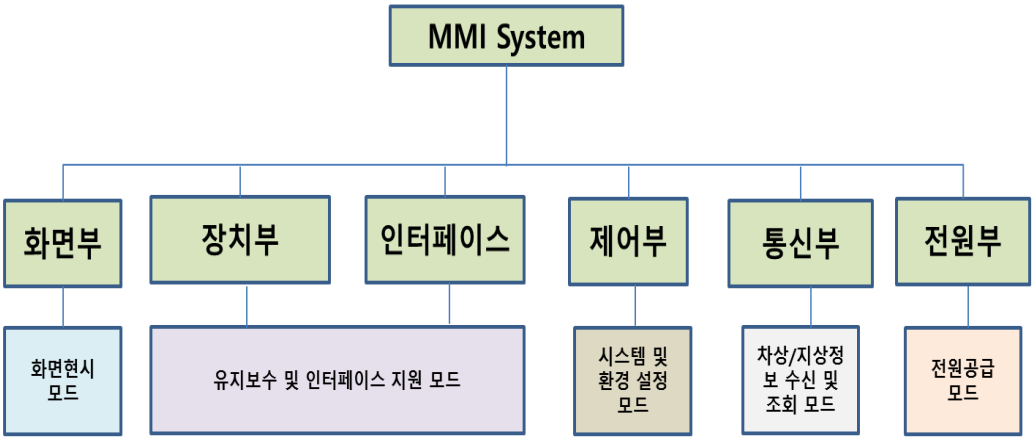


그림 170 MMI 시스템 구성에 따른 동작모드 식별

개별 요구사항에 대해서는 [표 228~231]을 통해서 식별되듯이, 요구사항을 [그림 170]에서 식별된 동작 모드에 따라 유형을 분류하여 개별 요구사항들이 해당 동작모드에 어디에 해당 되는지에 대해서 인지 가능하도록 하였다. 따라서 [그림 171]과 같이, 식별하였다.

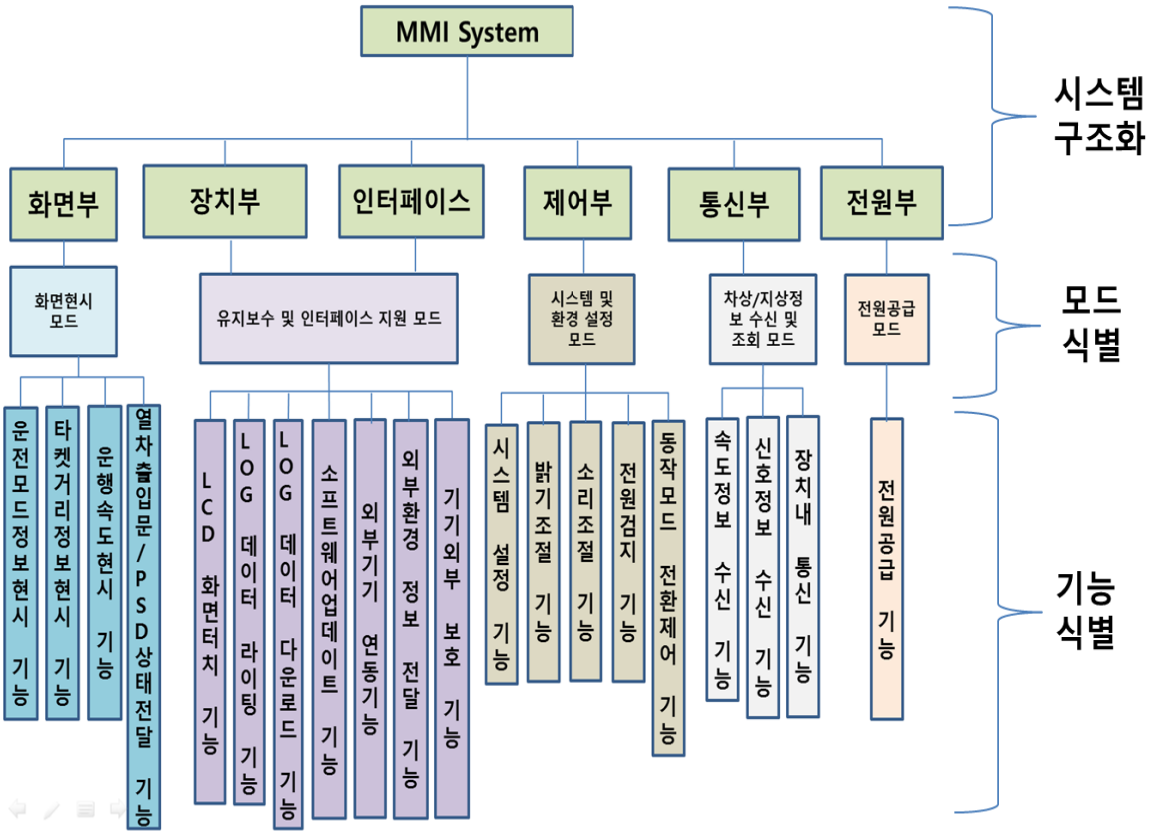


그림 171 컴포넌트/동작모드에 따른 지원기능 식별

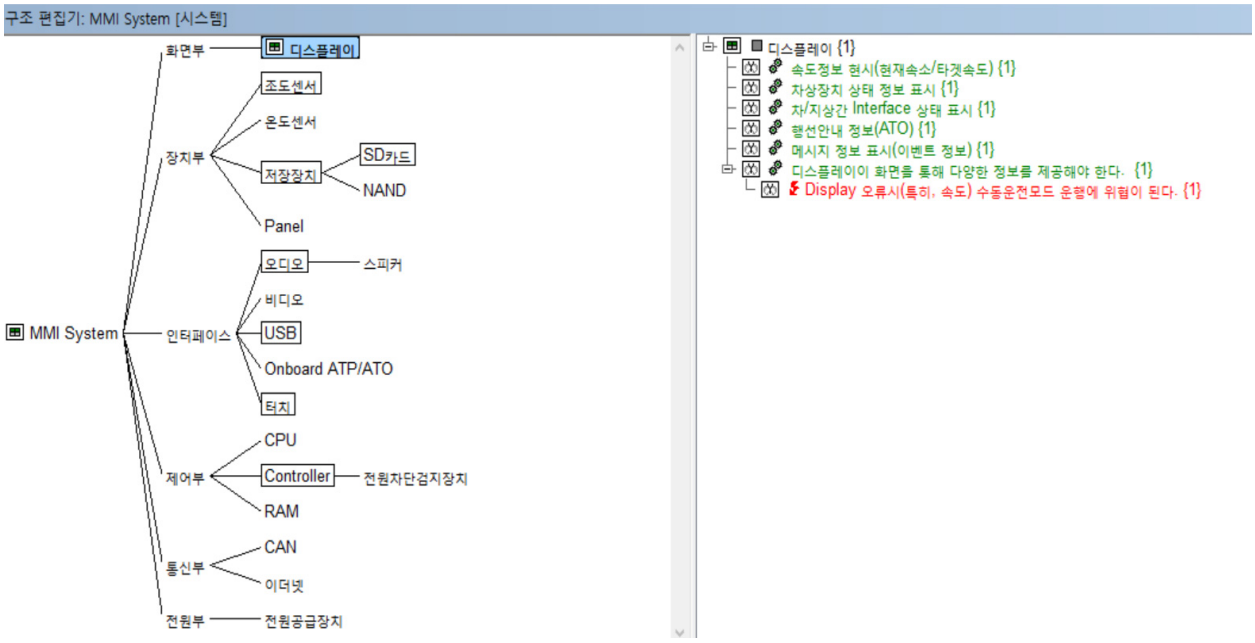


그림 172 도구기반의 FMEA 구조/기능 추적성 확립

2.4.3. FMEA Step. 3_MMI 시스템의 고장모드 및 오류기능 식별

MMI 시스템이 지닌 고장모드 및 오류 기능을 식별하기 위해서, 앞선 대상 시스템의 구조분석 및 기능분석이 수행 되었다. 해당 컴포넌트 및 기능이 지닌 오작동 및 오류 모드를 식별, 그리고 하나의 컴포넌트 및 기능이 다른 기능 및 컴포넌트에 미치는 간섭을 파악하기 위해서, 인터페이스 요구사항 및 성능 요구사항을 활용하여 보다, 간섭으로 미치는 영향 및 제약조건에서 발생할 수 있는 고장모드 및 오류 기능을 보다 식별할 수 있었다.

○ 인터페이스 요구사항(Interface Requirements)

표 232 인터페이스 요구사항

요구사항 ID	인터페이스 요구사항
IR_01	MMI 시스템은 외부장비인 ATP 장치에서 수신한 정보를 표시해야 한다.
IR_02	MMI 시스템은 외부장비 ATP/ATO로부터 운행정보 및 신호정보를 수신해야 한다.
IR_03	MMI 시스템은 외부 기기와 정보 연동될 수 있는 구성이어야 한다.

○ 성능요구사항(Performance Requirements)

표 233 성능 요구사항

요구사항 ID	성능 요구사항
PR_01	MMI 시스템은 목표지점까지 이동 간 100M 단위의 작은 눈금과 500m단위의 큰 눈금으로 최대 1000m까지 표시하여야 한다.
PR_02	MMI 시스템은 수신된 정보가 1000m 이상인 경우 그래프는 1000m까지 표시하고, 디지털 숫자로 실제 수신 거리를 표시해야 한다.
PR_03	MMI 시스템은 메인보드에 장착한 배터리를 사용하여 전원공급 중단 시 1~3초 전원을 유지할 수 있어야 한다.
PR_04	MMI 시스템은 속도계 정보에 대해 140km/h까지 표시할 수 있어야 한다.

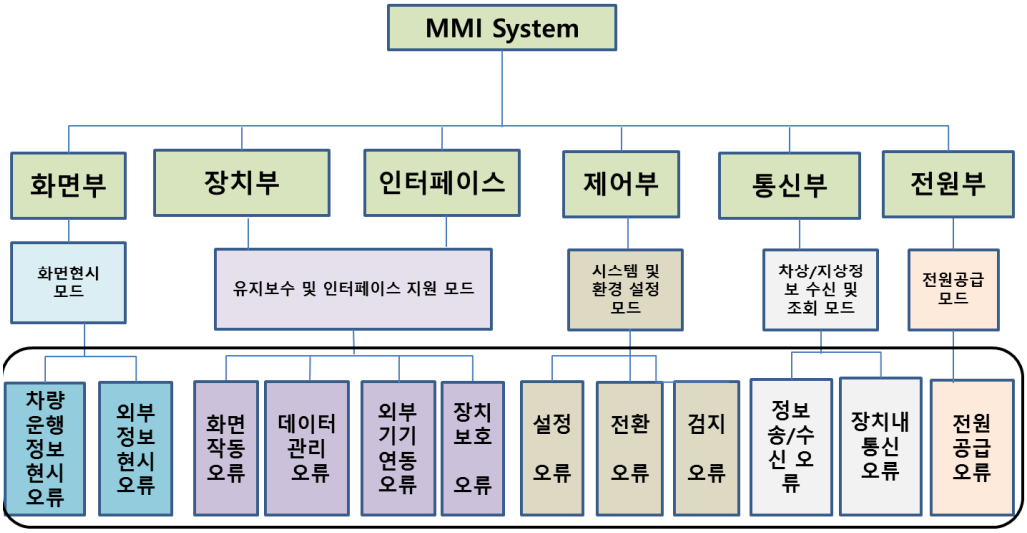


그림 173 MMI 시스템의 오류모드 식별

해당 컴포넌트의 구성도와 해당 컴포넌트가 지닌 모드, 그리고 해당 모드가 지닌 오류 모드는 [그림 173]과 같이 식별하였다. 해당 컴포넌트가 지니고 있는 오류 모드를 식별하였다면, 개별 오류모드가 지닌 기능적 오류를 식별하였다. 식별된 기능적 오류는 앞서 식별된 기능을 중심으로 식별되었고, 개별 기능의 오류로부터 발생할 수 있는 타 오류에 대해서는 인터페이스 요구사항을 기반으로 보다 보완할 수 있었다.

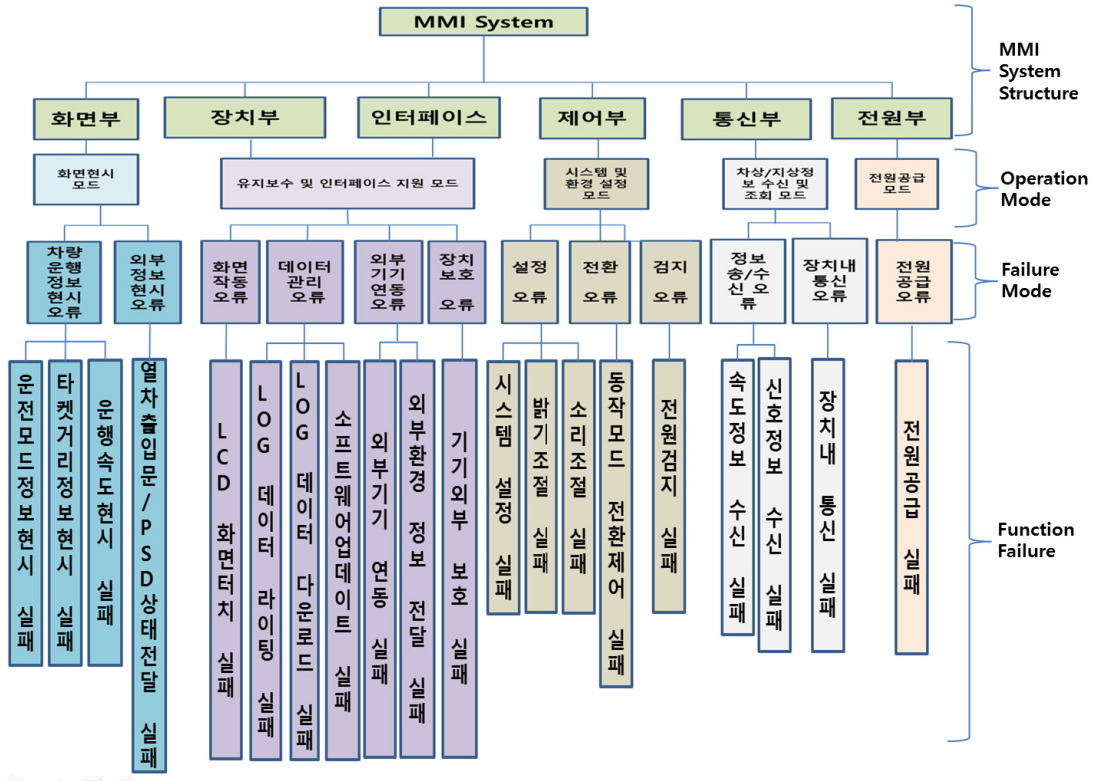


그림 174 MMI 시스템의 실패모드 및 기능식별

2.4.4. FMEA Step. 4 _MMI 시스템의 오류 심각도 평가 및 설계 대안 제시

시스템 명칭	시스템 구조			운용 모드	기능	기능 추적성 (하나의 오작동으로 다른 영향에 미치는 경우)	고장모드(Failure Mode)	고장 영향(Failure Effects)	RPN 평가			RPN	우선순위	오류원인 (Failure Cause	예방조치	안전요구사항	(대책안) 설계반영 방안
									심각도	발생빈도	허용수준						
MMI System	Layer-1	Layer-2	Layer-3														
	화면부	디스플레이	디스플레이 (a)	화면현시 모드	운전모드 정보 현시 기능(a-1)		차량운행정보 현시오류	운전모드 정보 현시 불가능, (운행지연 및 중돌발생)	3	2	5	30	5		1. I/F의 결함허용 설계 2. S/W 건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증		
					타겟거리 정보 현시 기능(a-2)	g-1-1		거리정보 현시 불가능 (운행지연)	2	2	4	16	8				
					운행속도 정보 현시 기능(a-3)	g-1-1		운행속도 정보 현시 불가능 (운행지연)	2	2	4	16	8				
					열차출입문/PSD 상태 현시 기능(a-4)	g-1-1	외부정보 현시오류	열차출입문/PSD상태 전달실패, 승객인명사고 발생	4	1	5	20	6				
	장치부	센서	조도센서 (b)	유지보수 모드	조도상태 감지 기능(b-1)	j-1	외부기기연동 오류	조도감지 오작동	2	2	2	8	10				
			온도센서 (c)		온도상태 감지 기능(c-1)			온도감지 오작동	1	2	2	4	12				
		저장장치	NAND (d)		Application & Configration Data(d-1)	k-1, k-2	어플리케이션 동작오류	어플리케이션 오동작	2	1	3	6	11				
					소프트웨어 업데이트 기능(d-2)	e	데이터관리 오류	소프트웨어 업데이트 실패	2	2	3	12	9				
			SD카드 (e)		Log 데이터 Writing 기능(e-1)	P-1, L		Log 데이터 라이팅 실패	2	3	3	18	7				
					Log 데이터 Download 기능(e-2)	L		Log 데이터 다운로드 실패	2	3	3	18	7				
		외부보호	외부판넬 (f)	기기외부 보호모드	기기 외부 보호 기능(f-1)		기기외부 보호실패	장치보호 실패	2	1	2	4	12				
	인터페이스	오디오 (g)	스피커 (g-1)	인터페이스 지원 모드	외부환경 정보 전달 기능(g-1-1)	a-2, a-3, a-4	외부기기 연동오류	음향정보 전달 실패	2	2	2	8	10				
		USB (h)			외부기기 연동기능(h-1)	d-2		외부 기기 연동 실패	2	2	3	12	9				
		비디오 (i)			LCD 화면 터치 기능(i-1)			비디오 정보 제공 실패	3	2	2	12	9				
	제어부	Controller (J)	Micoms (J-1)	시스템 및 환경설정 모드	시스템 설정 기능(J-1-1)		설정오류	시스템 환경 설정 실패	3	2	3	18	7				
					LCD 밝기 조절 기능(J-1-2)	b-1		LCD 밝기 조절	2	2	2	8	10				
					MMI 소리 조절 기능(J-1-3)			MMI 장치 소리 조절 실패	2	2	2	8	10				
				동작감지 모드	동작검지 기능(J-1-4)		전환오류	동작모드 전환 제어 실패	4	2	4	32	4		1. I/F의 결함허용 설계 2. S/W 건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증		
			전원검지장치 (J-2)	전원감지 모드	전원검지 기능(J-2-1)	P-1,	검지오류	전원검지 기능 실패	5	2	3	30	5		1. I/F의 결함허용 설계 2. S/W 건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증		
		CPU (K)		OS 동작 모드	OS 제어(k-1)	d-1	OS이행오류	OS 오동작	3	2	2	12	9				
				Application 동작 모드	Application 제어(k-2)	d-1	어플리케이션 동작오류	어플리케이션 오동작	2	2	2	8	10				
		RAM (L)		시간 동기화 모드	실시간 정보 전달 기능(L-1)	J-1-4	시간정보 오류	실시간 시전 정보 동기화 오류	2	2	2	8	10				
				데이터 R/W	데이터 읽기 기능(L-2)	e-2	데이터 오류	데이터 읽기 오류	2	2	3	12	9				
					데이터 쓰기 기능(L-3)	e-2		데이터 쓰기 오류	2	2	3	12	9				
	통신부	CAN (M)		차상/지상 정보 수신 및 조회 모드	속도정보 수신 기능(M-1)		정보 송수신 오류	속도정보 수신 실패	2	2	3	12	9				
		LAN (N)			신호정보 수신 기능(N-1)			신호정보 수신 실패	2	2	3	12	9				
		Ethernet (O)			장치내 통신 기능(O-1)			장치내 통신 오류 및 실패	4	3	4	48	3				
	전원부	전원공급장치 (P)		전원공급 모드	전원공급 기능(P-1)	a, e-1, J-2-1	전원공급오류	전원공급 차단 오류, 충/추돌 발생	5	3	6	90	1				
		전압전환 장치 (Q)			전압전환 기능(Q-1)	j-2		DC/DC 전압 전환 차단 오류	4	3	5	60	2		1. I/F의 결함허용 설계 2. S/W 건전성확보 및 기능시험 3. 처리결과의 비교 또는 다수결 검증		

그림 175 시범 적용업체 FMEA 수행 산출물

2.4.5. FMEA Step. 5_MMI 시스템의 안전 요구사항

앞선, MMI 시스템에 대해서 FMEA 안전성 분석활동 수행을 통해 MMI 시스템을 구성하는 서브시스템 컴포넌트를 중심으로 해당 컴포넌트가 지닌 기능적 오류를 기반으로 시스템에 미치는 종합적인 영향에 대한 우선순위를 평가하였다. 개별 오류의 영향이 미치는 영향의 중요도를 RPN(Risk Priority Number) 값을 통해, 상대적으로 평가 하고자 하였다. 평가된 결과를 바탕으로 우선순위를 평가하였다. 평가된 결과를 바탕으로 상위 순위에 해당하는 평가된 요소의 기능 및 컴포넌트를 중심으로 안전 요구사항 생성을 수행하였다. 상위 수준에 해당하는 요소는 [그림 176]을 통해서 식별할 수 있듯이, 전원 및 전압과 관련된 요소가 해당 되었다. 따라서 이에 해당하는 요소를 중심으로 안전 요소를 식별하여 요구사항으로 도출하였다.

- MMI 시스템 안전 요구사항(Safety Requirements)
  - 운영모드가 Active이고 주행 중 일 때는 설정모드로 전환되면 안 된다.
  - 전원이 차단되면 MMI 시스템은 1~3초 후에 종료 되어야 한다.
  - ATP/ATO로부터의 PSD 정보 등 잘못된 정보를 수신 또는 잘못된 MMI 정보 표현은 승객의 안전에 치명적인 피해를 초래할 수 있다.
  - 현시가 필요한 차상장치 데이터는 차상의 현시 화면(MMI)에 현시되도록 데이터를 전송하여야 한다.
  - 차상장치는 차량의 현시 화면(MMI화면)을 통해 운전자의 입력정보를 입력 받아야 한다.
  - MMI 시스템은 운영모드가 Active이고 주행 중 일 때는 설정모드로 전환되어서는 안 된다.
  - MMI 시스템은 전원 차단이 검지된 경우 MMI 시스템은 1~3초 후에 종료 되어야 한다.

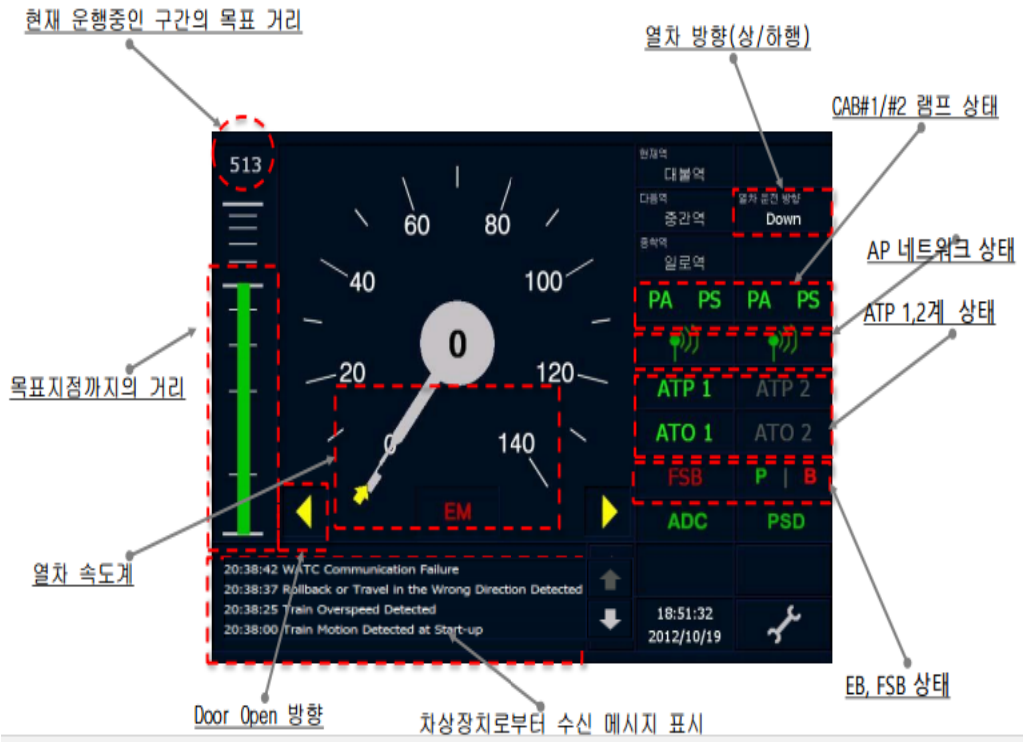


그림 176 현장적용 대상업체 MMI 시스템 화면 구성

MMI 시스템은 철도차량의 내-외부 상태 정보를 현시하는 것이 주목적인 시스템이다. 따라서 주행 중 발생할 수 있는 위 사항(5~6)에 대해서 기능적으로 지켜지는 사항이 시스템 안전성 확보에 기본의 목표가 되겠다. 따라서 해당 안전 요구사항을 구현할 수 있도록 소프트웨어 차원에서 하드웨어적 결합을 통한 구현이 뒷받침 되어야 할 것이다. 따라서 향후의 소프트웨어의 구현은 [그림 176]의 현재 개발 산출물에 대한 개선은 위 1~5 사항을 반영한다면, 안전성 측면에서 현행의 개발 산출물 [그림 176]보다 확고하며, 사전에 미리 안전성을 확보 가능하다는 점에서 본 현장 적용의 큰 공헌이라고 볼 수 있을 것이다. 또한, 본 시스템 안전성 분석 가이드 적용을 통해 도출한 시스템 안전 요구사항을 중심으로 IEC 62279 기반의 소프트웨어 안전성 분석을 수행할 수 있는 정보를 제공할 수 있다.

○ 시스템 요구사항

표 234 시스템 요구사항 (예시)

요구사항 ID	요구사항
차량 운행상태 및 통신 상태 표시	
SRS_MMI_REQ1.1	MMI 시스템은 Target Distance 정보, 운행 속도, 운전 모드, 출입문 방향, 역 정보, 운행 방향에 대한 정보가 표시되어야 한다.
SRS_MMI_REQ1.1-1	MMI 시스템은 다음 정차 역까지의 남은 거리(Target Distance)를 표시할 수 있어야 한다.
SRS_MMI_REQ1.1-2	MMI 시스템은 운행 속도 정보를 표시할 수 있어야 한다.
SRS_MMI_REQ1.1-3	MMI 시스템은 운전 모드를 표시할 수 있어야 한다.
SRS_MMI_REQ1.1-4	MMI 시스템은 다음 정차역의 출입문 열림 방향에 대한 정보를 표시 할 수 있어야 한다.
SRS_MMI_REQ1.1-5	MMI 시스템은 역 정보를 표시할 수 있어야 한다.
SRS_MMI_REQ1.1-6	MMI 시스템은 차량의 열차 스케줄에 따른 운행 방향을 표시하여야 한다.
SRS_MMI_REQ2.1	MMI 시스템은 차상장치(ATP/ATO)의 차량의 운행 및 상태 정보를 수신해야 한다.
SRS_MMI_REQ2.2	MMI 시스템은 차상장치(ATP/ATO)에서 수신한 정보를 표시해야 한다.
SRS_MMI_REQ2.3	MMI 시스템은 차상장치(ATP/ATO)에서 수신한 정보를 저장해야 한다.
SRS_MMI_REQ2.4	MMI 시스템은 차량 설정 정보를 차상장치(ATP/ATO)에 송신해야 한다.
차량 열차정보 표시 및 시스템 설정	
SRS_MMI_REQ3.1	MMI 시스템은 운행기록, 시각표시, 설정 취급 메뉴 버튼이 위치해야 한다.
SRS_MMI_REQ3.2	MMI 시스템은 자동밝기 조절 기능이 수행되어야 한다.
SRS_MMI_REQ3.3	MMI 시스템은 휘도 조절이 가능한 화면 조절 기능을 수행해야 한다.
SRS_MMI_REQ3.3-1	MMI 시스템은 취급자에 의한 자동 휘도 조절 모드를 지원해야 한다.
SRS_MMI_REQ3.3-2	MMI 시스템은 취급자에 의한 수동 휘도 조절 모드를 지원해야 한다.

SRS_MMI_REQ3.4	MMI 시스템은 표시 상태 조절 기능을 수행해야 한다.
SRS_MMI_REQ3.5	MMI 시스템은 차량에서 이벤트 발생에 대한 경고음을 표출해야 한다.
SRS_MMI_REQ3.6	MMI 시스템은 장치 내 전원을 활용해 시스템 시간이 관리되어야 한다.
SRS_MMI_REQ3.7	MMI 시스템은 장치의 시간과 표준 시각과의 시간 동기화가 수행되어야 한다.
SRS_MMI_REQ3.7-1	MMI 시스템은 차량의 운영 중의 표준 시각에 대한 정보 수신은 장치의 시간과 표준 시각의 동기화를 통해 일치해야 한다.
SRS_MMI_REQ3.8	MMI 시스템은 기록 메모리를 보유 하여야 한다.
SRS_MMI_REQ3.9	MMI 시스템의 기록 메모리는 메모리만 교체 할 수 있도록 편의성을 갖추어야 한다.
SRS_MMI_REQ3.10	MMI 시스템은 화면에 표시되는 항목별 색상을 달리 표현할 수 있어야 한다.
SRS_MMI_REQ3.10-1	MMI 시스템의 화면에 표시되는 심볼은 차상장치로부터 수신한 데이터의 상태에 따라 색상을 달리 표현할 수 있어야 한다.
SRS_MMI_REQ3.11	MMI 시스템의 Target Speed Pointer는 목표속도 변경 시 현재속도를 목표속도로 유지하기 위해 준수해야 할 속도를 화살표로 표시할 수 있어야 한다.
SRS_MMI_REQ3.12	MMI 시스템의 제한속도(Limited Speed)는 현재 운행 중인 구간의 제한속도를 나타내는 모형으로 표시되어야 한다.
SRS_MMI_REQ3.13	MMI 시스템은 Position Active와 Proximity Stop 정보를 표시해야 한다.
차량 통신 데이터 조회	
SRS_MMI_REQ4.1	MMI 시스템은 AP(Access Point)의 연결 상태 정보를 아이콘으로 표시하여야 한다.
시스템 환경 설정 및 소프트웨어 유지/관리	
SRS_MMI_REQ5.1	MMI 시스템은 시스템 관리 및 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.1-1	MMI는 Driver ID 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.1-2	MMI는 Train Number 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.1-3	MMI는 WATP IP 설정 기능을 지원해야 한다.

SRS_MMI_REQ5.1-4	MMI 시스템은 Train Data 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.1-5	MMI 시스템은 Wheel Size 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.2	MMI 시스템은 취급(SOM, EOM, PDT Start/Stop) 기능을 지원해야 한다.
SRS_MMI_REQ5.3	MMI 시스템은 CAB ID, 시각, 휘도, 음량 설정 기능을 지원해야 한다.
SRS_MMI_REQ5.4	MMI 시스템은 MMI 시스템의 기능 개선 또는 변경을 위해 프로그램의 업데이트가 되도록 구현해야 한다.
외부장치 인터페이스 (External Device Interface)	
SRS_MMI_REQ6.1	MMI 시스템은 외부 장치인 ATP/ATO 장치에서 운행 및 상태 정보를 수신한다.
SRS_MMI_REQ6.2	MMI 시스템은 차량/지상장치/차상장치 설정 기능을 통해 저장된 데이터(Driver ID, Train Number, WATP IP, Train Data, Wheel Size)를 외부 장치인 ATP/ATO로 송신해야 한다.
SRS_MMI_REQ6.3	MMI 시스템은 데이터 송수신 시 데이터 유효성을 확인할 수 있어야 한다.
성능 (Performance)	
SRS_MMI_REQ7.1	MMI 시스템은 매 300ms 마다 ATP/ATO 장치로부터의 데이터를 수신해서 처리해야 한다.
SRS_MMI_REQ7.2	MMI 시스템은 목표지점까지 이동 간 100m 단위의 작은 눈금과 500m 단위의 큰 눈금으로 최대 1000m까지 표시되어야 한다.
SRS_MMI_REQ7.3	MMI 시스템은 수신된 정보가 1000m 이상인 경우 그래프는 1000m까지 표시하고 디지털 숫자로 실제 수신 거리를 표시해야 한다.
SRS_MMI_REQ7.4	MMI 시스템은 속도계 정보에 대해 140km/h까지 표시할 수 있어야 한다.
시스템 안전 요구사항 (System Safety Requirement)	
SSRS_MMI_REQ1.1	MMI 시스템은 차량이 주행 중 일 때 설정모드로 전환되어서는 안 된다.
SSRS_MMI_REQ2.1	MMI 시스템은 차량 전원 차단이 검지된 경우 1~3초 후에 종료되어야 한다.
(기타)	기타항목

2.5. MMI 시스템 소프트웨어 개발 활동 수행

2.5.1. 소프트웨어 요구사항 식별 및 명세

본 안전가이드 적용 시에는 소프트웨어 안전성 분석을 위해 차상 MMI시스템을 시스템 안전 무결성 등급 2 (SIL 2)를 목표로 요구사항을 식별, 명세하였다. [표 235] 소프트웨어 요구사항 단계 기법 및 대책에서 해당하는 SIL 2에 해당하는 기법 및 대책은 다음과 같은 표로 정리된다.

표 235 소프트웨어 요구사항 단계 기법 및 대책 (SIL 2)

기법 및 대책	참조	SIL 2
1. 정형기법(수학적인 접근법에 기반)	부록.B-28	R
2. 모델링	7.7 참조	R
3. 구조적 방법론	부록.B-52	R
4. 결정 테이블	부록.B-13	R

SIL 2에서 선택하는 “기법 및 대책”은 “2. 모델링”([표 155] 참조)을 선택하였다.

표 236 모델링 기법 및 대책에 대한 상세 표 (SIL 2)

기법 및 대책	참조	SIL 2
1. 데이터 모델링	부록.B-65	R
2. 데이터 흐름 다이어그램	부록.B-11	R
3. 제어 흐름 다이어그램	부록.B-66	R
4. 유한 상태 기계 / 상태 전이 다이어그램	부록.B-27	HR
5. 시간 패트리넷	부록.B-55	R
6. 결정/진리 테이블	부록.B-13	R
7. 정형 기법들	부록.B-28	R
8. 성능 모델링	부록.B-39	R
9. 프로토타입/애니메이션	부록.B-43	R
10. 구조 다이어그램	부록.B-51	R
11. 순차 다이어그램	부록.B-67	HR

[표 155] 모델링 기법 및 대책에서 “4. 유한 상태 기계 / 상태 전이 다이어그램”과 11. 순차 다이어그램”의 선택이 권고사항이다. 이 중에서 “11. 순차 다이어그램”을 선택하였다.

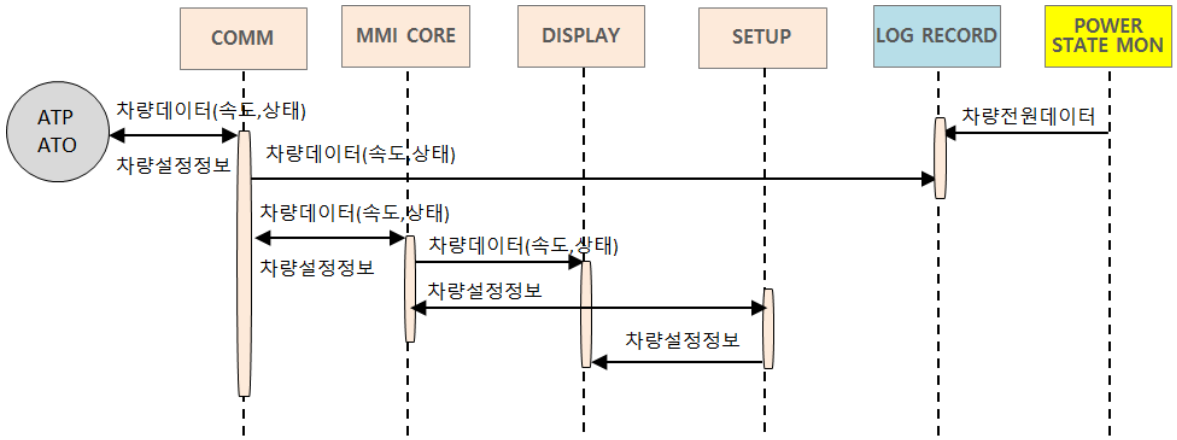


그림 177 차상 MMI 시스템의 시퀀스 다이어그램 (예시)

순차 다이어그램을 기반으로 유즈케이스를 사용하여 소프트웨어 요구사항의 기능 요구사항과 안전 요구사항을 도출하였다.

표 237 MMI 기능 및 안전 요구사항 (예시)

요구사항 ID (관련 사항)	요구사항 (근거)
열차 통신 관련	
SWRS_MMI_REQ1.1 (관련 사항: SRS_MMI_REQ2.1)	MMI 소프트웨어는 LAN으로 차상 ATO/ATP와 데이터를 송/수신한다.
SWRS_MMI_REQ1.2 (관련 사항: SRS_MMI_REQ2.1)	MMI 소프트웨어는 차상 ATP/ATO 에서 매 300ms마다 운행 및 상태 정보를 수신한다.
SWRS_MMI_REQ1.2-1 (관련 사항: SRS_MMI_REQ2.3)	MMI 소프트웨어는 차상 ATP/ATO 에서 수신된 운행 및 상태 정보를 저장해야 한다.
SWRS_MMI_REQ1.2-2 (관련 사항: SRS_MMI_REQ6.3)	MMI 소프트웨어는 차상 ATP/ATO 에서 수신된 운행 및 상태 정보의 데이터 유효성을 확인해야 한다.
SWRS_MMI_REQ1.3 (관련 사항: SRS_MMI_REQ2.4)	MMI 소프트웨어는 차량 설정 정보를 차상 ATP/ATO로 송신해야 한다.
차량 운행 상태 정보	
SWRS_MMI_REQ2.1 (관련 사항: SRS_MMI_REQ1.1-2)	MMI 소프트웨어는 차량 속도를 화면에 표시해야 한다.
SWRS_MMI_REQ2.2 (관련 사항: SRS_MMI_REQ1.1-1)	MMI 소프트웨어는 다음 정차역까지의 거리를 화면에 표시해야 한다.
SWRS_MMI_REQ2.3 (관련 사항: SRS_MMI_REQ1.1-4)	MMI 소프트웨어는 다음 정차역의 출입문 열림 방향에 대한 정보를 화면에 표시해야 한다.
SWRS_MMI_REQ2.4 (관련 사항: SRS_MMI_REQ1.1-3	MMI 소프트웨어는 운전모드를 화면에 표시해야 한다.
소프트웨어 환경 설정	

SWRS_MMI_REQ3.1 (관련 사항: SRS_MMI_REQ5.1)	MMI 소프트웨어는 MMI 상의 운영 환경 설정 기능이 있어야 한다.
SWRS_MMI_REQ3.1-1 (관련 사항: SRS_MMI_REQ5.3	MMI 소프트웨어는 MMI 상의 안내 음성의 음량을 설정할 수 있는 기능이 있어야 한다.
SWRS_MMI_REQ3.1-2 (관련 사항: SRS_MMI_REQ3.3)	MMI 소프트웨어는 화면의 밝기를 조절하는 기능이 있어야 한다.
SWRS_MMI_REQ3.2 (관련 사항: SRS_MMI_REQ5.1)	MMI 소프트웨어는 차량 정보 설정 기능이 있어야 한다.
SWRS_MMI_REQ3.2-1 (관련 사항: SRS_MMI_REQ5.1-1)	MMI 소프트웨어는 DRIVER ID를 설정하는 기능이 있어야 한다.
SWRS_MMI_REQ3.2-2 (관련 사항: SRS_MMI_REQ5.1-5)	MMI 소프트웨어는 Wheel Size를 설정하는 기능이 있어야 한다.
성능 (Performance Requirements)	
SWRS_MMI_REQ4.1 (관련 사항: SRS_MMI_REQ7.1)	MMI 소프트웨어는 목표지점까지 이동 간 100m 단위의 작은 눈금과 500m 단위의 큰 눈금으로 최대 1000m까지 표시해야 한다.
SWRS_MMI_REQ4.2 (관련 사항: SRS_MMI_REQ7.2)	MMI 소프트웨어는 수신된 정보가 1000m 이상인 경우 그래프는 1000m까지 표시하고 디지털 숫자로 실제 수신 거리를 표시해야 한다.
SWRS_MMI_REQ4.3 (관련 사항: SRS_MMI_REQ7.3)	MMI 소프트웨어는 차량 정보에 대해 140km/h까지 표시해야 한다.
소프트웨어 안전 요구사항 (Software Safety Requirement)	
SWSRS_MMI_REQ1.1 (관련 사항: SSRS_MMI_REQ1.1)	MMI 소프트웨어는 운영모드가 Active 이고 차량이 주행 중 일 때는 설정모드로 전환되어서는 안 된다. 근거(Rationale): 주행 중 MMI 화면이 설정모드로 전환되면 차량 운행 및 상태 정보를 육안으로 확인해야 한다.
SWSRS_MMI_REQ1.2 (관련 사항: SSRS_MMI_REQ1.2)	MMI 시스템은 차량 전원 차단이 검지된 경우 로그를 기록해서는 안 된다. 근거(Rationale): 로그 기록 도중 차량 전원이 차단되면 데이터 무결성을 보장할 수 없다.
(기타)	기타항목

이를 소프트웨어 기능, 비-기능 및 안전성 기능 요구사항으로 분류하여 세분화 하였다.

표 238 MMI 소프트웨어 기능 요구사항 (예시)

Use Case Name	수신된 데이터 유효성 검증		ID	SWRS_MMI_REQ1.2-2
요구사항	차상 ATP/ATO 에서 수신된 운행 및 상태 정보의 데이터 유효성을 확인해야 한다.			
Actor	통신 모듈			
Trigger	ATP/ATO의 데이터가 통신 큐로 수신된 경우			
Flow of Events				
Basic Flow	Step	Action		
	1	통신 모듈은 ATP/ATO의 통신 큐(Queue)를 주기적으로 폴링(Polling)한다.		
	2	통신 큐(Queue)에서 데이터를 수신한다.		
	3	통신 큐(Queue)를 비운다.		
	4	수신된 데이터의 유효성(CRC32)을 검증한다.		
	5	MMI 제어 모듈로 데이터를 전달한다.		
	6	Use Case 종료		
Alternative Flow 1	Step	2a - 통신 큐(Queue)에 수신할 데이터가 없는 경우		
	1	통신 모듈은 ATO/ATP의 통신 큐(Queue)를 주기적으로 폴링(Polling)한다.		
	2	Use case 종료		
Alternative Flow 2	Step	3a - 통신 큐(Queue)를 비울 수 없는 경우		
	1	시스템은 경고 메시지를 보여준다. - 메시지: 통신에 장애가 발생하였습니다.		
	2	통신 모듈은 ATO/ATP의 통신 큐(Queue)를 주기적으로 폴링(Polling)한다.		
	3	Use case 종료		
Alternative Flow 3	Step	4a - 수신된 데이터의 유효성(CRC32) 검증이 실패한 경우		
	1	시스템은 경고 메시지를 보여준다. - 메시지: 비정상적인 데이터가 수신되었습니다.		
	2	통신 모듈은 ATO/ATP의 메시지 큐를 주기적으로 폴링(Polling)한다.		
	3	Use case 종료		
Alternative Flow 4	Setp	5a - MMI 제어 모듈로 데이터를 전달하지 못하는 경우		
	1	시스템은 경고 메시지를 보여준다. - 메시지: 수신된 데이터를 처리하지 못하였습니다.		
	2	통신 모듈은 ATO/ATP의 메시지 큐를 주기적으로 폴링한다.		
	3	Use case 종료		
Pre-conditions	해당사항 없음			
Post-conditions	1. 수신된 데이터가 유효성 검증이 된 경우 - MMI 제어 모듈로 데이터를 전달한다.			
관련 요구사항	해당사항 없음			

상세 요구사항	해당사항 없음

표 239 MMI 소프트웨어 비-기능 요구사항 (예시)

비기능 요구사항	
강건성 (Robustness)과 유지보수성 (Maintainability)	시스템의 소프트웨어 업그레이드가 가능해야 하며, 로그 기록 데이터의 백업이 가능해야 한다.
안전성 (Safety)	소프트웨어 안전 무결성 등급 (SIL)의 2를 만족해야 한다.
효율성 (Efficiency)	제한된 크기의 화면에서도 보다 많은 정보를 표현하기 위해 그래프와 숫자의 크기 및 위치는 적절해야 한다.
사용성 (Usability)	차상 ATP/ATO의 데이터를 표시하는 그래프와 정보는 직관적으로 인지 가능해야 한다. 단 특정적인 표현에 대해서는 매뉴얼에 그 사용법을 설명해야 한다.
이식성 (Portability)	특정 운영 환경에서만 사용 가능한 소프트웨어를 배제하고 범용 소프트웨어를 사용하여 타 시스템 환경에서도 운영이 가능해야 한다.
성능(Performance)	매 300ms 주기 마다 수신되는 ATP/ATO 데이터를 처리해야 한다.
제약사항 (Constraints)	차상 ATP/ATO 장치가 설치되어 있는 차량에서만 사용 가능해야 한다.
(기타)	기타항목

표 240 MMI 소프트웨어 안전 요구사항 (예시)

Use Case Name	차량 운행 상태 검지 후 설정 모드 진입 방지	ID	SWSRS_MMI_REQ1.1
안전 요구사항	운영모드가 Active 이고 차량이 주행 중 일 때는 설정모드로 전환되어서는 안 된다.		
Actor	MMI 제어 모듈		
Tigger	운영모드가 주행모드일 때 (차량 운행 중) 통신 모듈로부터 차량 속도 데이터가 0이 들어오는 경우		
Flow of Events			
Basic Flow	Step	Action	
	1	MMI 제어 모듈은 이전전(n-2)과 이전(n-1)에 수신된 차량 속도 데이터가 모두 0이 아님을 확인한다.	
	2	가상 차량 속도는 이전전(n-2)값과 이전(n-1)의 평균값으로 정의한다.	
	3	화면 표시 모듈에 차량 가상 속도값을 전달한다.	
	4	시스템은 경고 메시지를 보여준다. - 메시지: 주행 중 비정상 속도가 감지되었습니다.	
	5	운영모드를 주행모드로 설정한다.	
	6	설정메뉴 버튼을 비활성화 한다.	
	7	Use Case를 종료한다.	
Alternative Flow 1	Step	1a - 이전전(n-2)과 이전(n-1)에 수신된 차량 속도 데이터가 둘 중 하나	

		가 0이 아닌 경우
	1	가상 차량 속도는 이전전(n-2)값과 이전(n-1)의 평균값으로 정의한다.
	2	화면 표시 모듈에 차량 가상 속도값을 전달한다.
	3	시스템은 경고 메시지를 보여준다. - 메시지: 주행 중 비정상 속도가 감지되었습니다.
	4	운영모드를 주행모드로 설정한다.
	5	설정메뉴 버튼을 비활성화 한다.
	6	Use Case를 종료한다.
Alternative Flow 2	Step	1b - 이전전(n-2)과 이전(n-1)에 수신된 차량 속도 데이터가 모두 0인 경우
	1	가상 차량 속도는 이전전(n-2)값과 이전(n-1)의 평균값으로 정의한다.
	2	화면 표시 모듈에 차량 가상 속도값을 전달한다.
	3	운영모드를 정지모드로 설정한다.
	4	설정메뉴 버튼을 활성화 한다.
	5	Use Case를 종료한다.
Pre-conditions	MMI 운전모드는 Active로 설정되어 있다. 차량은 운행 중이다. (차량 속도 > 0, 운영모드는 주행모드)	
Post-conditions	1. 주행 중 차량 속도가 0으로 수신되면 경고메시지를 출력하고 이전 상태값의 평균값으로 차량 속도가 표시된다. 2. 주행 중 차량 속도가 0으로 연속 수신되면 경고메시지를 출력하고 이전 상태값의 평균값으로 차량 속도가 표시된다. 3. 주행 중 차량 속도가 3연속으로 수신되면 경고메시지를 출력하지 않고 운행모드를 정지모드로 전환하고 설정메뉴를 활성화 한다.	
관련 요구사항	SWRS_MMI_REQ2.4 - MMI 소프트웨어는 운전모드를 화면에 표시해야 한다.	
상세 요구사항	SWRS_MMI_REQ3.1 - MMI 소프트웨어는 MMI 상의 운영 환경 설정 기능이 있어야 한다.	
안전 조건	1. 수신 차량 속도가 3연속으로 0이 수신되는 경우만 운영모드를 정지모드로 인식한다. 2. 정지모드인 경우만 화면 설정 메뉴를 활성화 한다.	
테스트 조건	통신 모듈로 수신된 차량 속도가 3연속으로 0인지 확인한다.	
(기타)	기타항목	

2.5.2. 소프트웨어 아키텍처 및 설계

소프트웨어 아키텍처 명세 단계에서는 도출된 소프트웨어 요구사항 (안전 요구사항 포함)에 대해 소프트웨어 전체적인 개념도를 정의하였다. 차상 MMI 시스템은 SIL 2를 목표로 하는 시스템이므로 “기법 및 대책”에 대해 필요한 사항은 다음과 같이 정리되었다.

표 241 소프트웨어 아키텍처 기법 및 대책 (SIL 2)

기법 및 대책	참조	SIL 2
1. 방어적 프로그래밍	부록.B-14	HR
2. 결함 검출 & 진단	부록.B-26	R
3. 오류 정정 코드	부록.B-19	-
4. 오류 검출 코드	부록.B-19	R
5. 고장 단정 프로그래밍	부록.B-24	R
6. 안전성 백 기법	부록.B-47	R
7. 다양화 프로그래밍	부록.B-16	R
8. 복구 블록	부록.B-44	R
9. 역방향 복구	부록.B-5	NR
10. 전방향 복구	부록.B-30	NR
11. 고장 복구 재시도 방법	부록.B-46	R
12. 실행된 사례 기억	부록.B-36	R
13. 인공지능-결함 정정	부록.B-1	NR
14. 소프트웨어의 동적 재구성	부록.B-17	NR
15. 소프트웨어 오류 영향 분석	부록.B-25	R
16. 우아한 저하	부록.B-31	R
17. 정보 은닉	부록.B-33	-
18. 정보 캡슐화	부록.B-33	HR
19. 완전하게 정의된 인터페이스	부록.B-38	HR
20. 정형기법	부록.B-28	R
21. 모델링	7.7참조	R
22. 구조적 방법론	부록.B-52	HR
23. 컴퓨터 지원 설계 및 명세도구를 통한 모델링	7.7참조	R

SIL 2에서 권장하는 IEC 62279 소프트웨어 아키텍처 기법 및 대책은 1. 방어적 프로그래밍, 19. 완전하게 정의된 인터페이스, 22. 구조적 방법론 와 2. 결함 검출 & 진단, 4. 오류 검출 코드, 5. 고장 단정 프로그래밍, 7. 다양화 프로그래밍, 12. 실행된 사례 기억, 15. 소프트웨어 오류 영향 분석, 21. 모델링 중 하나를 권고하고 있다. (기법 및 대책에 대한 자세한 내용은 [부록 B]를 참조한다.)

본 단계의 시범적용에는 1. 방어적 프로그래밍, 19, 완전하게 정의된 인터페이스, 22. 구조적 방법론과 4. 오류 검출 코드 기법 및 대책을 사용하였다.

“1. 방어적 프로그래밍과 4. 오류 검출 코드” 기법 및 대책 같은 경우 소프트웨어 안전 요구사항 “SWSRS_MMI_REQ1.1 - 운영모드가 Active 이고 차량이 주행 중 일 때는 설정모드로 전환되어서는 안 된다.” 를 만족하게 되었다. (차량 운행 중 차량 속도가 0로 잘못 입력되는 경우 입력값에 따라 설정 모드로 전환을 해 차량 운행 및 상태 정보 현시를 방해하는 위험원을 방지하도록 설계하였다.)

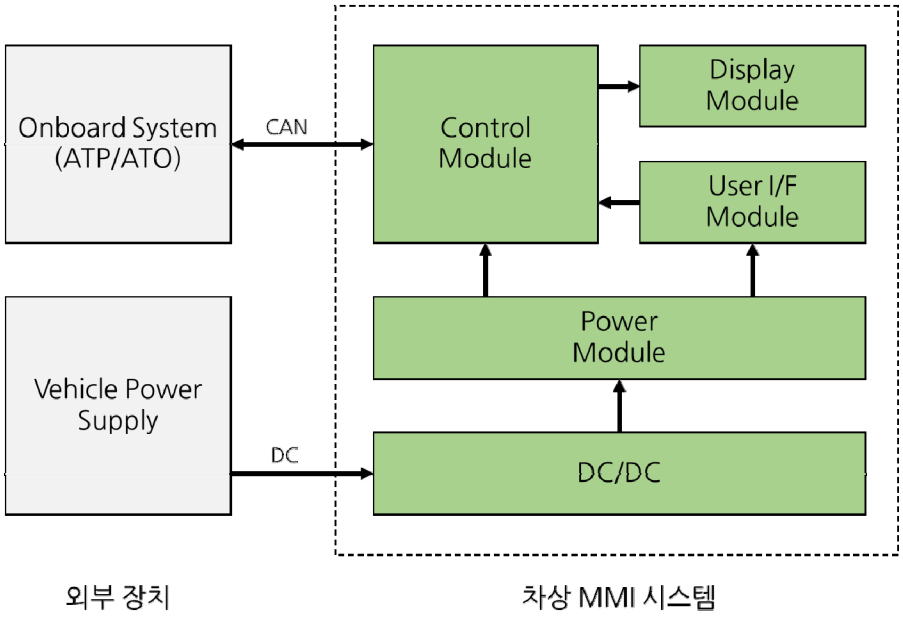


그림 178 차상 MMI 시스템 개념도 (예시)

소프트웨어 컴포넌트를 식별한다.

표 242 차상 MMI 소프트웨어 컴포넌트 (예시)

컴포넌트 명	기능	설명
COMM	통신	차상 ATP/ATO와 통신
LOG_RECORD	로그 저장	COMM의 정보를 로그에 저장
MMI_CORE	MMI 핵심	COMM을 통해 전달된 ATP/ATO 메시지 처리
DISPLAY	화면 표시	MMI_CORE 처리 정보를 화면에 표시
SETUP	차량 정보 설정	MMI_CORE 설정을 입력
POWER_STATE_MON	전원 차단 검지	차량 전원 공급 상태 감시
(기타)	기타항목	기타

소프트웨어 인터페이스는 IEC 62279 표준이 권고하는 “19, 완전하게 정의된 인터페이스” 기법 및 대책을 사용하였다. (자세한 내용은 부록 B 참조) “19. 완전하게 정의된 인터페이스” 기법 및 대책의 가장 중요한 요소는 식별된 컴포넌트간의 내부 인터페이스의 도출과 외부 인터페이스의 도출이다. 컴포넌트 간의 데이터 공유는 반드시 인터페이스를 통해 수행하도록 설계하였다.

표 243 소프트웨어 아키텍처 설계 (예시)

1. 차상 MMI는 ATP/ATO 장치에서 전달된 프로토콜을 처리하기 위해 다음과 같은 컴포넌트들이 식별되었다.
   - COMM – ATP/ATO 장치와 차량 운행 및 상태 정보, 차량 등록 정보의 송수신을 위한 CAN 통신
   - MMI_CORE – 차량 운행 및 상태 정보, 차량 등록 정보화면 처리
   - DISPLAY – 차량 운행 및 상태 정보, 차량 등록 정보화면 표시
   - SETUP – 차량 등록 정보 설정
   - LOG_RECORD – 차량 운행 및 상태 정보의 기록 메모리 저장
   - POWER_STATE_MON – 차량 전원 상태 검지
2. 컴포넌트 공통 사항
   - ATP/ATO 장치의 프로토콜은 폴링 모드(NO_WAIT 모드)로 처리되어야 한다.
   - 컴포넌트 간의 인터페이스는 포트 방식(메시지 중심의 처리)이 가능하도록 구성한다.
3. COMM 컴포넌트
   - ATP/ATO 장치와 프로토콜에 의해 패킷 데이터(Packet Data)를 송수신한다.
   - ATP/ATO 장치에서 받은 패킷 데이터(Packet Data)의 유효성 점검(Check)을 한다.
4. LOG_RECORD 컴포넌트
   - 기록 메모리에 저장한다.
5. POWER_MON 컴포넌트
   - 차량 전원 상태를 검지한다.
6. MMI_CORE 컴포넌트
   - 차상 운행 및 상태 정보를 처리한다,
   - 차량 등록 정보를 처리한다,
7. DISPLAY 컴포넌트
   - 차상 운행 및 상태 정보를 그래픽으로 표시한다.
   - 차량 등록 정보를 그래픽으로 표시한다.
8. SETUP 컴포넌트
   - 차량 등록 정보를 입력 받는다.

소프트웨어 설계는 식별된 컴포넌트가 전체 아키텍처에서 어떤 역할을 수행하게 되는지 명세함으로써 아키텍처를 상세화하고 구체화 하였다.

본 단계의 시범적용에는 “22. 구조적 방법론”를 사용하여 아키텍처를 설계함으로써 식별된 컴포넌트를 기능 위주가 아닌 독립적인 역할을 수행할 수 있도록 설계하였다. 기존은 기능 위주의 설계 방식으로 기능 개선/확장이나 유지보수에 대처하기 어려운 구조로 되어 있었다.

표 244 소프트웨어 인터페이스 (예시)

1 외부 인터페이스

1.1 ATP/ATO 장치와 COMM간의 인터페이스

- ATP/ATO 장치와 프로토콜로 패킷 데이터 (Packet Data)를 송수신한다.
- ATP/ATO 장치에서 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)를 수신한다.
- 차량 등록 정보(ROLLING_STOCK_REG)를 ATP/ATO로 송신한다.

1.2 DISPLAY와 그래픽 API간의 인터페이스

- 그래픽 API 의 표준 인터페이스를 정의한다. (OPENGL의 API 호출 방법 표시)
- 차량 운행 및 상태 정보 (ROLLING_STOCK_INFO)를 그래픽 API로 전달한다.
- 차량 등록 정보(ROLLING_STOCK_REG)를 그래픽 API로 전달한다.

2 내부 인터페이스 (컴포넌트 간 인터페이스)

2.1 COMM 컴포넌트

- 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)를 MMI_CORE로 전달한다.
- 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)를 LOG_RECORD로 전달한다.
- SETUP에서 차량 등록 정보(ROLLING_STOCK_REG)를 전달받는다.

2.2 LOG_RECORD 컴포넌트

- COMM 에서 차량 운행 및 상태 정보(ROLLING_STOCK_INFO를 전달받는다.
- 메모리에서 버퍼를 할당 받지 못했을 경우 DO_NOT_ALLOCATE_BUF 이벤트를 발생한다.
- 버퍼가 가득 찼을 경우 LOG_BUF_IS_FULL 이벤트를 발생한다.

2.3 POWER_STATE_MON 컴포넌트

- 차량 전원 상태를 LOG_RECORD로 전달한다. (POWER_ON: 1, POWER_OFF: 0)

2.4 MMI_CONTROL 컴포넌트

- 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)를 DISPLAY로 전달한다.
- 차량 운행 및 상태 정보(ROLLING_STOCK_INFO를 SETUP으로 전달한다.
- 차량 설정 정보(ROLLING_STOCK_REG)를 COMM으로 전달한다.

2.5 DISPLAY 컴포넌트

- MMI_CORE에서 차량 운행 및 상태 정보 (ROLLING_STOCK_INFO)를 전달받는다.
- SETUP에서 차량 등록 정보 (ROLLING_STOCK_INFO)를 전달받는다.

2.6 SETUP 컴포넌트

- 차량 등록 정보(ROLLING_STOCK_REG)를 DISPLAY로 전달한다.
- 차량 등록 정보(ROLLING_STOCK_REG)를 MMI_CORE로 전달한다.

표 245 소프트웨어 설계 (예시)

1. 차상 MMI 소프트웨어는 MISRA-C 코딩 규칙과 호환되는 C언어를 사용한다.
2. 차상 MMI 소프트웨어는 코딩 스타일을 따라 작성한다.
3. 컴포넌트 공통 사항
   - 각 컴포넌트의 입력 메시지는 유효성을 확인한다.
   - 각 컴포넌트의 출력 메시지는 유효성을 확인한다.
4. COMM 컴포넌트
   - ATP/ATO 장치에서 수신된 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)는 로그 저장을 위해 LOG_RECORD로 전달한다.
   - ATP/ATO 장치에서 수신된 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)의 유효성 확인을 위해 CRC32 기법을 사용한다.
   - ATP/ATO 장치에서 수신된 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)의 유효성 확인이 되지 않는 데이터는 DISPLAY로 전달되지 않는다.
5. LOG_RECORD 컴포넌트
   - COMM에서 수신된 차량 운행 및 상태 정보(ROLLING_STOCK_INFO)는 기록 메모리에 저장한다.
   - POWER_STATE_MON의 상태가 POWER_OFF 인 경우 로그를 저장하지 않는다.
6. POWER_STATE_MON 컴포넌트
   - 매 30ms 주기 이내에 차량 전원 상태를 점검한다.
   - POWER_STATE_MON의 상태를 LOG_RECORD에 전달한다.
7. MMI_CONTROL 컴포넌트
   - COMM에서 전달받은 메시지에서 TACHO_SPEED_INFO를 DISPLAY에 전달하여 속도를 표시한다.
   - TACHO_SPEED_INFO에 따라 운영모드를 주행, 정지 모드로 구분한다.
   - SETUP에서 설정된 차량 등록 정보(ROLLING_STOCK_REG)를 COMM으로 전달한다.
8. DISPLAY 컴포넌트
   - TACHO_SPEED_INFO에 따라 운영 모드를 구분한다. (> 0: 주행 모드, = 0: 정지 모드)
   - DISPLAY의 슬립모드는 COMM에서 전달받는 ACTIVE_MMI의 메시지를 확인한다. (ATIVE_MMI: 1, 1계 활성화, 2계 슬립모드, ACTIVE_MMI: 0, 1계 슬립모드, 2계 활성화)
9. SETUP 컴포넌트
   - TACHO_SPEED_INFO에 따라 운영 모드를 구분한다. (> 0: 주행모드, = 0: 정지모드)
   - DISPLAY의 슬립모드는 COMM에서 전달받는 ACTIVE_MMI의 메시지를 확인한다. (ATIVE_MMI: 1, 1계 활성화, 2계 슬립모드, ACTIVE_MMI: 0, 1계 슬립모드, 2계 활성화)

2.5.3. 소프트웨어 컴포넌트 설계

소프트웨어 컴포넌트 설계 단계에서의 IEC 62279 기법 및 대책 권고사항은 [표 142]를 기준으로 “3. 구조적 방법론, 4. 모듈 방식, 5. 컴포넌트, 6. 설계 및 코딩 표준”과 “8. 엄격한 형식의 프로그래밍 언어, 9. 구조적 프로그래밍, 10. 프로그래밍 언어” 중 하나를 선정하는 것이다.

본 시범적용에서는 “3. 구조적 방법론, 4. 모듈 방식, 5. 컴포넌트, 6. 설계 및 코딩 표준”과 “10. 프로그래밍 언어”를 사용하였다. “6. 설계 및 코딩 표준”은 철도소프트웨어서 자주 사용되는 MISRA-C++ 코딩 표준 사용을 권장하였다.

표 246 컴포넌트 설계 기법 및 대책 (SIL 2)

기법 및 대책	참조	SIL 2
1. 정형 기법	부록.B-28	R
2. 모델링	표 A.17	HR
3. 구조적 방법론	부록.B-52	HR
4. 모듈 방식	부록.B-38	M
5. 컴포넌트	7.7참조	HR
6. 설계 및 코딩 표준	7.7참조	HR
7. 분석 가능한 프로그램	부록.B-2	HR
8. 엄격한 형식의 프로그래밍 언어	부록.B-49	HR
9. 구조적 프로그래밍	부록.B-53	HR
10. 프로그래밍 언어	7.7참조	HR
11. 언어 하위집합	부록.B-35	-
12. 객체지향 프로그래밍	7.7참조, 부록.B-57	R
13. 절차적 프로그래밍	부록.B-60	HR
14. 메타프로그래밍	부록.B-59	R

“10. 프로그래밍 언어” 는 C++언어를 사용하였다. 이는 기존의 MMI 소프트웨어가 C++ 언어로 작성되어 있기 때문에 가이드 시범적용의 이해가 다른 언어의 선택보다 효과적이었다.

소프트웨어 컴포넌트 설계 시 기존 소프트웨어와의 인터페이스를 상세하게 명시하고 컴포넌트간의 내부 인터페이스를 설계하였다. 식별된 컴포넌트에 대해 컴포넌트에 대한 레이어를 정의하고 클래스 다이어그램을 사용하여 상세하게 컴포넌트를 설계하였다.

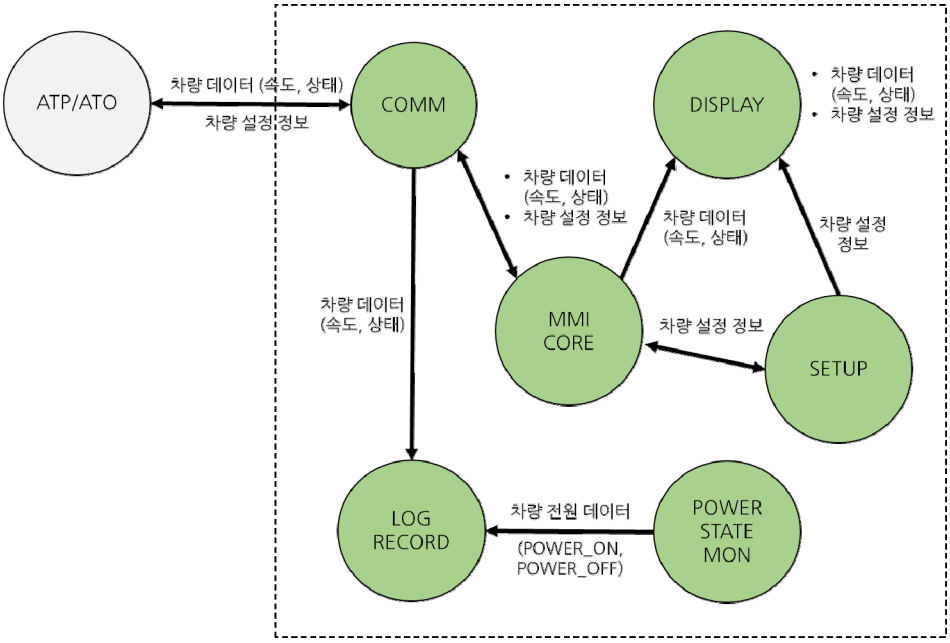


그림 179 차상 MMI 시스템의 소프트웨어 컴포넌트 및 인터페이스 (예시)

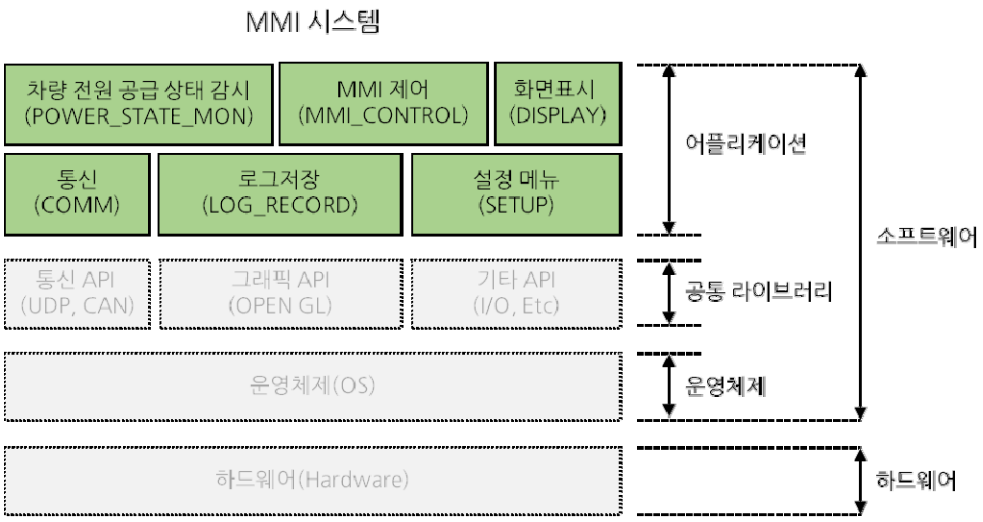


그림 180 차상 MMI 시스템 구성도 (예시)

메시지 중심의 컴포넌트를 설계한다. 차상 MMI는 ATP/ATO 장치에서 전달된 프로토콜을 처리하기 위해 다음과 같은 컴포넌트들이 식별되었다.

- COMM – ATP/ATO 장치와 차량 운행 및 상태 정보, 차량 등록 정보의 송수신을 위한 CAN 통신
- MMI_CORE – 차량 운행 및 상태 정보, 차량 등록 정보화면 처리
- DISPLAY – 차량 운행 및 상태 정보, 차량 등록 정보화면 표시
- SETUP – 차량 등록 정보 설정
- LOG_RECORD – 차량 운행 및 상태 정보의 기록 메모리 저장
- POWER_STATE_MON – 차량 전원 상태 검지

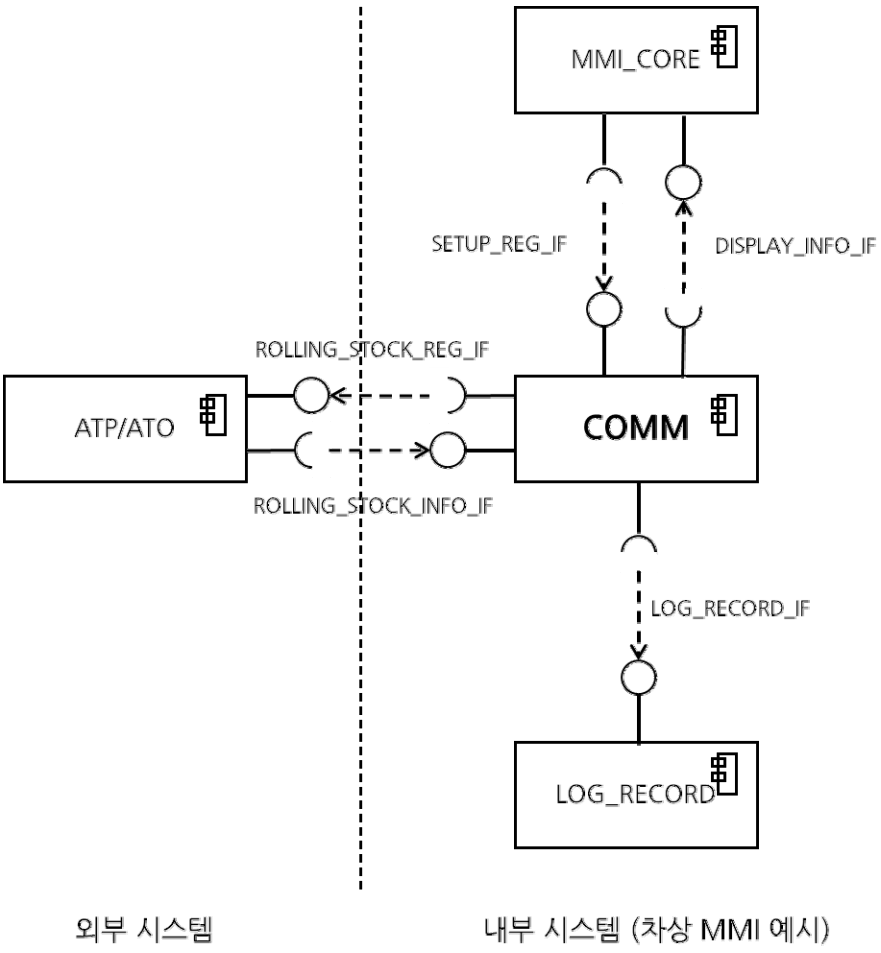


그림 181 COMM 컴포넌트 모델 인터페이스 (예시)

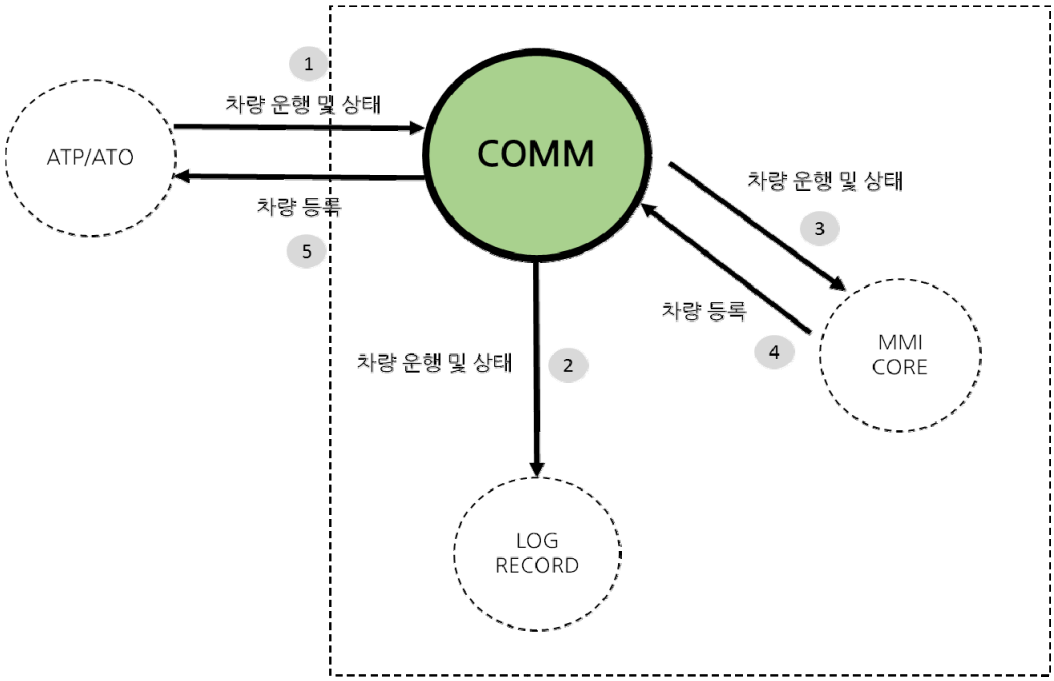


그림 182 COMM 컴포넌트 데이터 흐름 (예시)

COMM
Property train_info_fd train_reg_fd TRAIN_INFO_st TRAIN_REG_st DISPLAY_INFO_st
Method CAN_open() CAN_close() CAN_read() CAN_write() getTrainInfo() setTrainReg() check_CRC32() get_train_setup_info() set_display_info() set_log_record()

그림 183 COMM 클래스 모델 (예시)

표 247 컴포넌트 상세 설계 (예시)

1 컴포넌트 상세 설계

1.1 COMM 컴포넌트 데이터 흐름

- [그림 10 COMM 컴포넌트 데이터 흐름] 참조한다.
- ① ATP/ATO 장치에서 차량 운행 및 상태 정보 (TRAIN_INFO_st) 수신
- ② COMM에서 LOG_RECORD로 차량 운행 및 상태 정보 (TRAIN_INFO_st) 전달
- ③ MMI_CORE에 차량 표시 정보 (DISPLAY_INFO_st) 전달
- ④ MMI_CORE에서 차량 등록 정보 (TRAIN_REG_st) 전달 받음.
- ⑤ ATP/ATO 장치로 차량 등록 정보 (TRAIN_REG_st) 송신

1.2 COMM 컴포넌트 데이터 속성

- 관련 속성(Property)은 [그림 11 COMM 클래스 모델 (예시)] 참조한다.
- train_info_fd : ATP/ATO에서 차량 운행 및 상태 정보 (TRAIN_INFO_st)를 수신하기 위한 파일 디스크립터 (File Descriptor)
- train_reg_fd: 차량 등록 정보 (TRAIN_REG_st)를 ATP/ATO로 송신하는 파일 디스크립터 (File Descriptor)
- TRAIN_INFO_st: 차량 운행 및 상태 정보 구조체 (Structure)
- TRAIN_REG_st: 차량 등록 정보 구조체 (Structure)
- DISPLAY_INFO_st: 차량 운행 및 상태 정보를 화면에 표시하기 위한 구조체 (Structure)

1.3 COMM 컴포넌트 메소드

- 관련 메소드 (Method)는 [그림 11 COMM 클래스 모델 (예시)] 참조한다.
- CAN_open(): 통신을 위한 파일 디스크립터를 생성하고 파일의 상태를 개방한다. (개방 모드에는 RO: Read Only, RW: Read Write, WO: Write Only 등이 있다.)
- CAN_close(): 통신을 위한 파일 디스크립터를 종료한다.
- CAN_read(): 파일 디스크립터가 개방(open)된 상태에서 개방 모드가 RO나 RW일 때 수신된 통신 패킷 데이터를 읽기(read) 동작을 한다.
- CAN_write(): 파일 디스크립터가 개방(open)된 상태에서 개방 모드가 RW나 WO일 때 송신할 통신 패킷 데이터를 쓰기(write) 동작을 한다.
- get_train_info(): ATP/ATO에서 수신된 통신 패킷 데이터를 파싱(Paring)하여 TRAIN_INFO_st에 기록한다.
- set_train_reg(): 차량 등록 정보를 ATP/ATO로 송신하기 위해 TRAIN_REG_st에 조립(Assemble)한다.
- check_CRC32(): 수신된 통신 패킷 데이터의 유효성을 점검한다.
- get_train_setup_info(): 차량 등록 정보를 TRAIN_REG_st 에 저장한다.
- set_display_info(): 차량 운행 및 상태 정보를 화면에 표시하기 위해 TRAIN_INFO_st를 전달한다.
- set_log_record(): 수신된 통신 패킷 데이터를 저장하기 위해 TRAIN_INFO_st에 저장한다.

2.5.4. 소프트웨어 컴포넌트 구현 및 테스팅

컴포넌트의 구현 및 테스팅 단계에서는 설계된 컴포넌트를 실제 구현하는 것이 목적이었으나 시간제약상 수행하는 것은 무리가 있었다. 대안으로 코딩 표준의 중요성을 전달하고 이를 사용하는 것이 소프트웨어 구현 시 발생할 수 있는 잠재적인 결함을 보다 쉽게 발견할 수 있다는 것을 강조하였다.

3. 적용대상 시스템: 비상방송 시스템

3.1. 시스템 개요

전동차량의 분리 등, 열차 방송장치 장애와 전원 공급이 불가능한 상황과 같은 비상상황 발생 시 객실에 비상 안내 방송이 불가능하거나, 관제사령이 객실 비상 상황을 파악하지 못할 수 있다. 또한 지하 구간, 객실에 조명이 끊김에 따라 암흑 상황으로 인해 혼란을 초래할 수 있다. 이때, 객실 내 승객에게 안내 방송을 할 수 있는 비상방송시스템에 대한 필요성이 대두되었으며, 또한 객실 상황을 승객이 관제사령과 비상인터폰으로 통화를 하여 대처해야 하는 점도 같이 부각되고 있다. 또한 최소한의 조명을 제공함으로써 승객들이 적절한 비상 상황 대응을 할 수 있는 가이드역할을 할 수 있다.

비상방송시스템은 전동차량의 분리 등의 열차방송장치 장애와 전원공급 불가능한상황과 같은 비상상황에서 객실 내 승객에게 비상안내방송과 비상조명 제공함을 목적으로 한다. 무선망은 LTE-R망을 기반으로 하고 전동차 객실 비상방송장치를 통해 기관사, 관제사령 및 역무원이 사고 열차 객실 내에 있는 승객에게 안내 방송할 수 있도록 한 비상방송시스템이다.

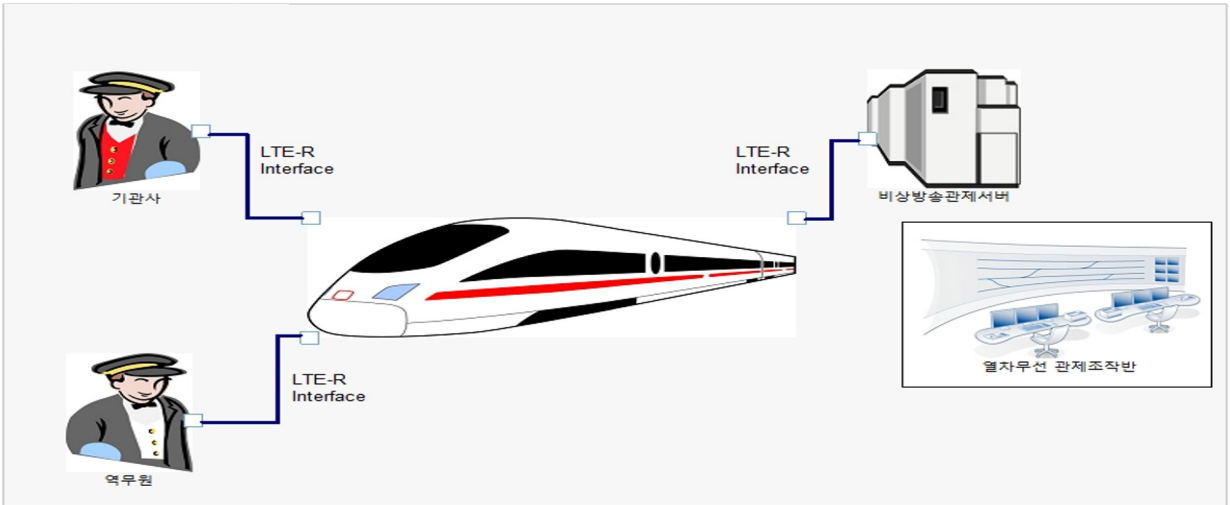


그림 184 비상방송시스템 구성

3.2. 시스템 아키텍처

비상방송 시스템의 시스템 구성은 [그림 185]과 같다. 종합관제센터는 관제조작반의 GUI 화면을 통해 사고 열차 선택 및 사고 열차에 비상방송을 할 수 있으며, 비상인터폰 통화를 통해 효율적으로 승객에게 비상 상황을 대처할 수 있도록 한다. 또한 비상방송 시스템의 영상 서비스를 통해 관제사가 정확하게 사고 현장을 파악하고 대응할 수 있게 도와준다.

비상방송 휴대용 무전기는 기관사가 비상 상황 발생 시 휴대용 무전기를 통해 승객에게 비상 방송을 할 수 있다. 또한 비상방송 시스템의 영상 서비스를 통해 기관사가 정확하게 사고 현장을 파악하고 대응할 수 있게 도와준다. 또한 역무원의 휴대용 무전기도 기관사 휴대용 무전기와 동일한 기능을 통해 비상 상황 발생 시 대응할 수 있다.

비상방송 시스템의 경우 종합관제센터를 통한 객실에 비상방송을 가능하게 하며, 기관사나 역무원을 통해 비상방송 또한 가능하다. 그리고 외부 전원이 공급 불가 시 객실에 조명을 비춤으로써 비상 상황에 승객들이 대처하는데 도움을 준다. 그리고 비상인터폰을 통해 승객이 직접 종합관제센터에 비상 상황을 연락할 수 있다. 또한 종합관제센터는 비상방송 시스템의 카메라를 통한 영상 정보를 토대로 비상 상황을 파악할 수 있다. 비상방송시스템의 각 하부 장치별 구성 및 기능은 [표 248]와 같다.

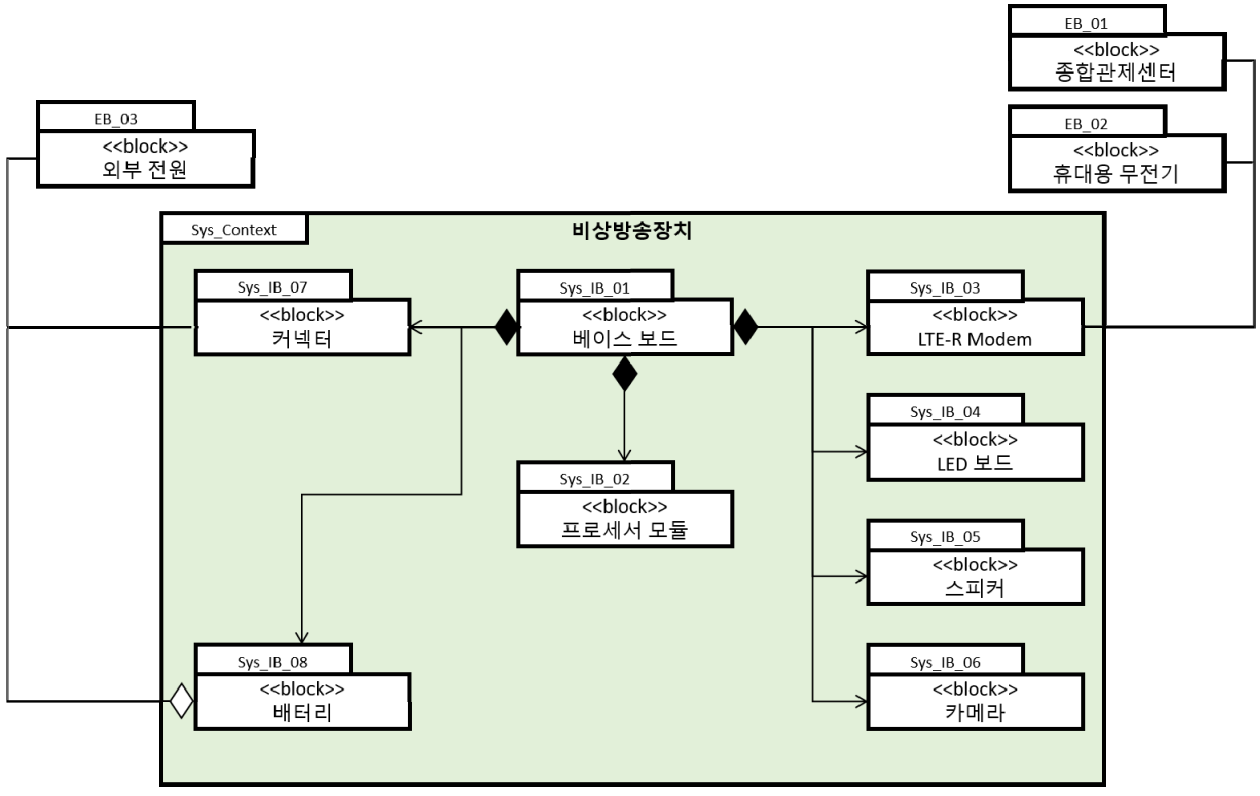


그림 185 비상방송 시스템 아키텍처

표 248 비상방송시스템의 각 하부 장치별 구성 및 기능

장치 ID	장 치 명	기 능
Sys_IB_01	베이스 보드	타 장치에 전원을 공급하며 데이터 전달 수행
Sys_IB_02	프로세서 모듈	운행 편성의 모드 판단과 비상방송장치의 상태 확인 및 비상시 장치 동작 실행 명령
Sys_IB_03	LTE-R 모뎀	종합관제센터, 휴대용 무전기가 비상방송장치와 데이터 송수신
Sys_IB_04	LED 보드	비상시 객실 내를 비추는 LED와 비상방송장치의 상태를 확인할 수 있는 LED
Sys_IB_05	스피커	비상시 객실 내에 비상방송을 송출
Sys_IB_06	카메라	비상시 객실 내를 촬영
Sys_IB_07	커넥터	외부 전원을 공급
Sys_IB_08	배터리	비상시 장치에 전원 공급

3.3. 시스템 안전성 분석 수행 범위

3.3.1. 시스템 구성에 따른 수행범위

비상방송장치의 기능과 하부컴포넌트의 인터페이스를 대상으로 안전성 분석은 비상방송장치 수준에서 안전성에 영향을 미칠 수 있는 컴포넌트를 식별하여 안전성 관리를 수행한다. 비상방송시스템 컴포넌트 수준에서 안전성 분석을 수행하며, 식별된 안전 요구사항의 검증을 통해 최종 안전성보증업무가 수행된다. 여기서 안전성보증업무란 식별된 안전성 요구사항이 비상방송장치 설계에 반영되었는지 확인까지만 해당한다. 시스템 구성 및 상세내용은 시스템 구성을 참고한다.

3.3.2. 시스템 생명주기에 따른 수행범위

비상방송장치 설계에 적용될 시스템 생명주기는 표준에서 정의된 시스템 생명주기를 기반으로 한다. 본 프로젝트에서는 시스템 생명주기 중 시스템 요건 배분부터 설계 및 구현 단계까지로 정의한다. 각 단계별 상세 수행업무는 시스템 생명주기 단계별 업무에서 다룬다.

3.3.3. 가정 및 제한사항

본 문서에서 정의하는 안전성 분석의 범위는 비상방송장치의 기능 및 인터페이스 범위로 제한한다. 또한 비상방송장치의 설계 단계의 안전성을 보증하기 위한 안전성 분석에서 시운전과 관련된 안전성 분석은 제외한다. 추가로 다음 사항으로 인한 고장 및 위험원은 안전성 분석 범위에서 제외하도록 한다.

- 정의된 운영 조건 이외의 운영을 통해 발생 가능한 위험원
- 요구되는 사양을 초과하는 환경 조건
- 자연적인 대 재난(번개, 홍수, 지진 등)
- 테러 또는 반달리즘

3.4. 안전성 분석 수행 전략

3.4.1. 안전성 분석 목표

비상방송장치의 안전성 분석 관점에서의 수행범위 및 목표를 설명한다. 비상방송장치의 안전성 분석은 다음의 목표를 위해 수행한다.

- 비상방송장치 수준의 위험원 식별 및 안전 요구사항 도출
- 비상방송장치 컴포넌트간의 인터페이스 위험원 식별 및 안전 요구사항 도출
- 비상방송장치 수준의 위험원에 대한 위험도 평가 및 검증
- 비상방송장치 컴포넌트간의 인터페이스 위험원에 대한 위험도 평가 및 검증

3.4.2. 안전성 분석 추진전략

- 시스템의 국제적인 안전성 보증을 위하여 국제 규격(IEC 62278)에 부합하는 안전성 분석 수행
- 정의된 위험도 허용 기준에 근거한 위험도 평가 수행
- 안전성 분석을 통해 제시된 안전 요구사항의 만족을 위한 요구사항 관리
- 시스템과 관련하여 안전성 측면에서 치명적인 요소의 안전성 설계(Fail-Safe) 적용
- 안전성 분석을 통해 필요한 경우, 안전성 측면에서 위험원의 발생 확률을 감소시키고 고장확인 가능한 다중계 설계 적용

3.5. 위험도 평가 및 허용 기준

시스템의 위험도 평가 및 허용을 위한 기준은 다음과 같으며, 해당 기준은 해당 프로젝트의 RFP에서 제공한 내용을 기반으로 참조하였다. 본 시스템 안전성 분석 가이드를 위한 적용 예시로서 활용한 사항들은 가정 사항으로 제시한다.

3.5.1. 위험원 발생 빈도 구분

시스템의 위험도 평가를 위한 위험원 발생빈도 구분기준은 다음과 같다. 본 위험원 발생빈도 기준은 위험원으로 발생할 수 있는 결과적인 사고의 발생빈도를 의미한다.

표 249 비상방송시스템의 위험원 발생 빈도 구분

구 분		발생빈도 F	설 명
F6	매주 자주발생	F ≥ 100회/년	특정장소에서 자주 발생하는 경우
F5	빈번함	10회/년 ≤ F < 100회/년	선로구간내에서 자주 발생하는 경우
F4	있음직한	1회/년 ≤ F < 10회/년	선로구간내에서 1회 혹은 2회 발생하는 경우
F3	때때로 발생	0.1회/년 ≤ F < 1회/년	선로구간내뿐만이 아니라 관련 산업분야에서 다수 발생
F2	아주 적은 발생	0.01회/년 ≤ F < 0.1회/년	관련 산업분야내에 1회 혹은 2회 발생하는 경우
F1	희박한 발생	F < 0.01회/년	관련 산업분야내에 발생한 적이 거의 없는 경우

3.5.2. 위험원 심각도 구분

시스템의 위험도 평가를 위한 위험원 발생결과에 따른 심각도 구분기준은 다음과 같다.

표 250 비상방송시스템의 위험원 심각도 구분

심각도 수준		사람 및 환경에 미치는 결과	정량적 기준 예시
C6	재난 수준	인명의 사망, 시스템의 손실 또는 심각한 환경상의 피해를 유발하는 위험	10인 이상 사망
C5	치명적인 수준	심각한 인명의 상해, 직업상의 질병 및 중요한 시스템 또는 환경상의 피해를 초래하는 위험	2인 이상 사망, 10인 미만 사망
C4	중대한 수준	최소한 인명의 상해, 직업상의 질병 및 중요한 시스템 또는 환경상의 피해를 초래하는 위험	1인 이상 사망
C3	주요한 수준	최소한의 상해, 직업상의 질병 및 최소한의 시스템 또는 환경상의 피해를 초래하는 위험	1인 이하 중상
C2	경미한 수준	최소한의 상해, 직업상의 질병보다 작고, 최소한의 시스템 및 환경상의 피해보다 작은 영향을 초래하는 위험	1인 이하 경상
C1	무시할만한 수준	인명이나 환경 상에 피해를 발생하지 않으나 경제적 손실을 동반하는 위험	유지보수 필요

3.5.3. 위험도 허용 기준

시스템의 위험도 평가 결과에 따른 허용 기준은 다음의 위험도 매트릭스에 근거하여 정의된다.

표 251 비상방송시스템의 위험도 허용 기준

발생빈도		심각도					
		C1	C2	C3	C4	C5	C6
		무시할만한	경미한	주요한	중대한	치명적인	재난이 발생가능한
F6	매주 자주발생하는	B	A	A	A	A	A
F5	빈번한	B	B	A	A	A	A
F4	있음직한	B	B	B	A	A	A
F3	때때로	C	B	B	B	A	A
F2	아주 적은	C	C	B	B	B	A
F1	희박한	C	C	C	B	B	B

3.5.4. 위험도 평가 수준별 정의

시스템의 위험도 평가를 통하여 정의된 위험도 수준 및 허용에 대한 정의는 다음과 같다.

표 252 비상방송시스템의 위험도 평가 수준별 정의

리스크 수준	정 의	관리 절차
A	허용불가	본 등급에 해당하는 위험원은 반드시 제거되어야 함
B	허용가능	본 등급에 해당하는 위험원은 운영기관의 동의하에 허용이 가능
C	바람직하지 않음	본 등급에 해당하는 위험원은 운영기관의 동의 없이도 허용이 가능

3.6. 안전성 분석 개요

3.6.1. 시스템 생명주기 단계별 안전성 분석 절차

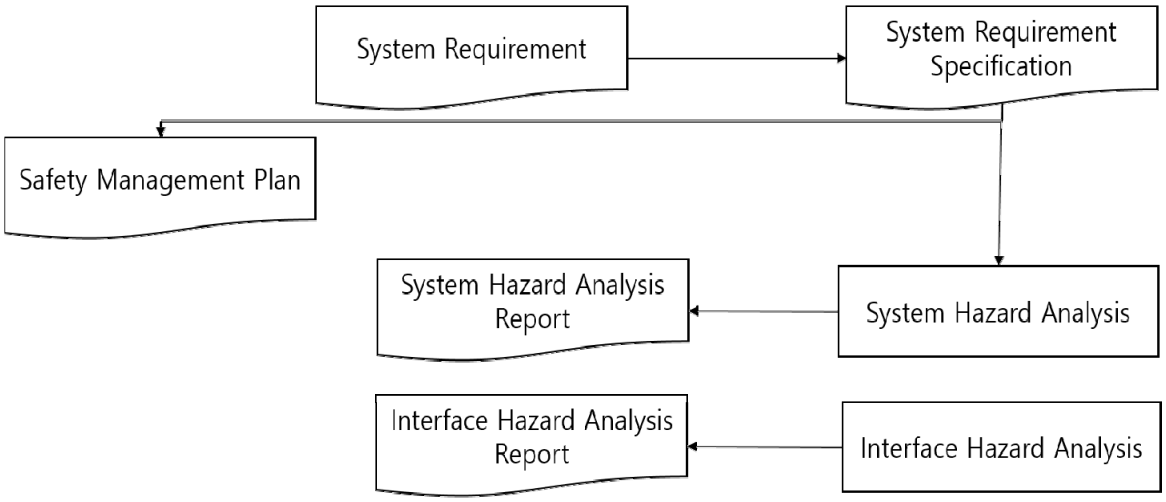


그림 186 비상방송시스템의 안전성 분석 절차

3.6.2. 안전성 분석 방법론

○ 시스템 위험원 분석

시스템의 시스템 위험원 분석은 시스템의 제작설계 완료 이전에 위험상황을 초래할 수 있는 해당 시스템 수준의 위험원을 도출하여 이를 제거하거나 허용 가능한 수준으로 저감시키기 위한 안전성 관련 설계조건을 정의하기 위하여 수행된다.

○ 인터페이스 위험원 분석

시스템의 인터페이스 위험원 분석은 시스템과 관련된 내부 및 외부 인터페이스를 정의하고, 이러한 인터페이스 상에서 발생할 수 있는 위험원을 규명하기 위해서 수행된다. 규명된 인터페이스 위험원을 제거 또는 위험도를 저감시키기 위한 내부 인터페이스는 설계적으로 처리될 것이며, 외부의 타 시스템과의 인터페이스 관련 요구사항은 해당 책임을 갖는 조직에 제공될 것이다.

3.7. 위험원 분석 수행

비상방송장치의 기능을 도출하기 위해 시스템 요구사항 중 기능 요구사항 분석을 통해 도출하였다. 이를 [표 253]에 나타내었다. 비상방송시스템의 내/외부 연동장치는 [표 254]와 같다.

표 253 비상방송장치 기능 요구사항

참조 ID	기능요구사항 ID	요구사항 설명
Sys_RS_10	F_REQ_01	비상 발생 시 종합관제센터로부터 음성 데이터를 비상방송 수신기로 수신한다.
Sys_RS_11	F_REQ_02	비상방송수신기는 LTE-R 휴대용 무전기로부터 수신한 음성 데이터를 비상방송객실장치에 전달한다.
Sys_RS_12	F_REQ_03	비상방송수신기는 종합관제센터로부터 수신한 음성 데이터를 비상방송객실장치에 전달한다.
Sys_RS_13	F_REQ_04	비상방송객실장치는 비상방송수신기로부터 수신한 음성 데이터를 스피커로 출력한다.
Sys_RS_14	F_REQ_05	비상 시 비상방송수신기로부터 신호를 받아 비상조명을 동작시킨다.
Sys_RS_16	F_REQ_06	비상방송장치와 인터페이스장치의 송/수신 정보 확인이 가능해야 한다.
Sys_RS_89	F_REQ_07	비상방송장치의 정상운용 중임을 확인하기 위해 상태 정보를 종합관제센터에 전송해야 한다.
Sys_RS_90	F_REQ_08	비상방송장치는 비상상황 발생 시 비상발생메시지를 종합관제센터에 전송해야 한다.
Sys_RS_91	F_REQ_09	비상방송장치는 비상상황 발생 시 객실 내 영상 정보를 수집하여 종합관제센터에 전송해야 한다.

표 254 비상방송장치 내/외부 연동장치

기능 ID	기 능	장치(내부)	장치(외부)
Sys_FN_01	음성 데이터 수신	LTE-R 모뎀, 베이스 보드, 프로세서 모듈	종합관제센터, 휴대용 무전기
Sys_FN_02	상태정보 전송	LTE-R 모뎀, 베이스 보드, 프로세서 모듈	종합관제센터
Sys_FN_03	비상방송메시지 전송	LTE-R 모뎀, 베이스 보드, 프로세서 모듈	종합관제센터
Sys_FN_04	객실 내 방송	스피커, 베이스 보드, 프로세서 모듈, LTE-R 모뎀	종합관제센터, 휴대용 무전기
Sys_FN_05	비상조명 동작	LED Board, 베이스 보드, 프로세서 모듈	
Sys_FN_06	상태정보 현시	LED Board, 베이스 보드, 프로세서 모듈	
Sys_FN_07	객실 내 영상정보 전송	카메라, 베이스 보드, 프로세서 모듈, LTE-R 모뎀	종합관제센터

비상방송장치 설계 보고서에 근거하여 시스템 요구사항을 분석하여 기능요구사항을 식별하였다. 기능요구사항에 기준한 시스템 기능을 대상으로 시스템 위험 분석이 수행되며 시스템 기능은 아래와 같다. 위험분석의 결과는 이후 항에서 기술할 것이다.

표 255 비상방송장치 시스템 기능

참조 ID (시스템 요구사항)	기능 ID	기능
Sys_RS_10 Sys_RS_11 Sys_RS_12	Sys_FN_01	음성 데이터 수신
Sys_RS_89	Sys_FN_02	상태정보 전송
Sys_RS_90	Sys_FN_03	비상발생메시지 전송
Sys_RS_13	Sys_FN_04	객실내 방송
Sys_RS_14	Sys_FN_05	비상조명 동작
Sys_RS_16	Sys_FN_06	상태정보 현시
Sys_RS_91	Sys_FN_07	객실내 영상정보 전송

비상방송장치 설계보고서에 근거하여 인터페이스 요구사항을 분석하여 시스템의 내부 및 외부 인터페이스를 아래와 같이 식별하였다. 시스템의 내/외부 인터페이스 정보를 대상으로 인터페이스 위험분석이 수행되며, 위험분석의 결과는 이후 항에서 기술할 것이다. [표 256]은 [그림 187]의 각 인터페이스 상세설명을 기술한다.

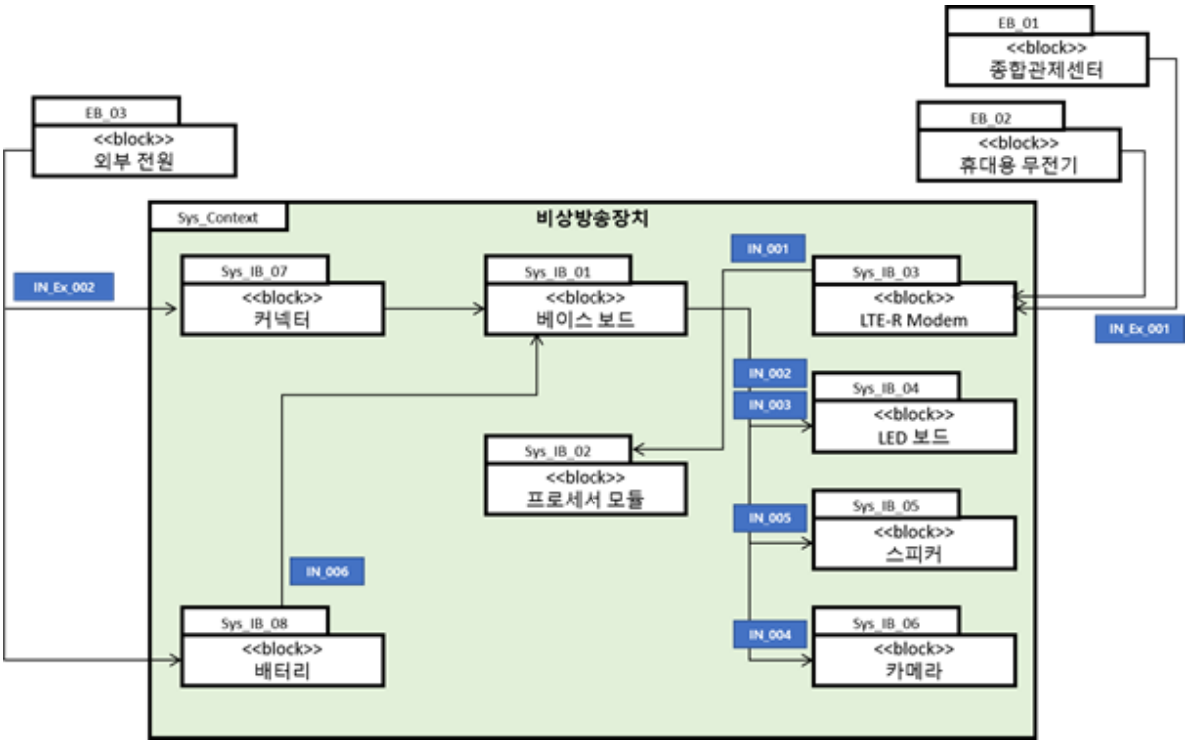


그림 187 비상방송장치 내/외부 인터페이스

표 256 비상방송장치 인터페이스 분석 결과

인터페이스 ID	인터페이스 설명	인터페이스 메인	인터페이스 대상	인터페이스 데이터/정보	인터페이스 형태
IN_Ex_001	종합관제센터로부터 위치정보 또는 본선 진출입 정보를 수신	종합관제센터	비상방송장치	위치정보 또는 본선 진출입 정보	LTE-R
IN_Ex_002	외부로부터 전력 수신	외부전원	비상방송장치	전력	차량전원공급장치 케이블
IN_001	LTE모뎀은 프로세서 모듈로 정보(음성 데이터) 전송	LTE-R 모뎀	프로세서 모듈	음성데이터	PCB Artwork
IN_002	프로세서 모듈은 비상조명 동작 명령을 LED 보드로 전송	프로세서 모듈	LED 보드	비상조명 동작 명령	PCB Artwork
IN_003	프로세서 모듈은 비상방송장치 상태를 LED 보드로 전송	프로세서 모듈	LED 보드	비상방송장치 상태	PCB Artwork
IN_004	프로세서 모듈은 이미지 캡처 명령을 카메라로 전송	프로세서 모듈	카메라	이미지 캡처 명령, 스틸컷 이미지	PCB Artwork
IN_005	프로세서 모듈은 음성데이터를 스피커로 전송	프로세서 모듈	스피커	음성데이터	PCB Artwork

본 문서에서 분석된 결과는 다음과 같은 양식지에 작성되며 각 항목 설명은 다음과 같다.

표 257 시스템 및 인터페이스 위험원 분석 양식

분석 대상	Item ID	Hazard ID	위험원	초기 위험도 평가			잔여 위험도 평가			원인	영향		저감대책
				F	S	R	F	S	R		System	Safty/Service	

표 258 시스템 및 인터페이스 위험원 분석 양식 설명

항 목		설 명
분석대상		기능 또는 인터페이스
Item ID		기능 요구사항 또는 인터페이스 요구사항 ID
Hazard ID		위험원의 식별 ID
위험원		기능 또는 인터페이스의 위험원
초기 위험도 평가	F-Frequency	위험원 발생결과의 발생빈도
	S-Severity	위험원 발생결과의 심각도
	R-Risk	위험원 결과
잔여위험도 평가		초기위험도 평가를 기준하여 저감대책을 수행하였을 때 잔여위험도를 초기위험도 평가와 같은 방식으로 작성
원인		위험원 발생원인
영향	System	시스템 수준의 위험원 발생 결과/영향 정도
	Safety/Service	안전/서비스 관련 위험원 발생 결과/영향 정도
저감대책		해당 위험원의 저감을 위한 대책(설계, 운영 등의 안전장치)

○ 시스템 위험원 분석(SHA)

시스템 위험원 분석을 위해서는 비상방송장치 시스템의 아키텍처와 기능을 분석해야 한다. 앞서 시스템의 아키텍처 분석을 통해 비상방송장치 시스템의 기능과 내/외부 인터페이스에 대해서 파악하였다. 시스템 위험원 분석은 기능 수행 시에 발생되는 세부 장치별 역할을 파악하여 위험원에 대한 원인과 이에 대한 영향을 파악하고자 한다.

비상방송장치의 음성 데이터 수신이라는 기능에 대한 예시를 [그림 188]에 도식화하였다. 음성데이터 수신을 위해서는 종합관제센터, 휴대용 무전기에서 음성 데이터를 LTE-R 모뎀을 통해 수신한다. 수신한 정보를 프로세스 모듈로 전달하여 프로세서 모듈은 최종적으로 음성 데이터 수신을 완료한다.

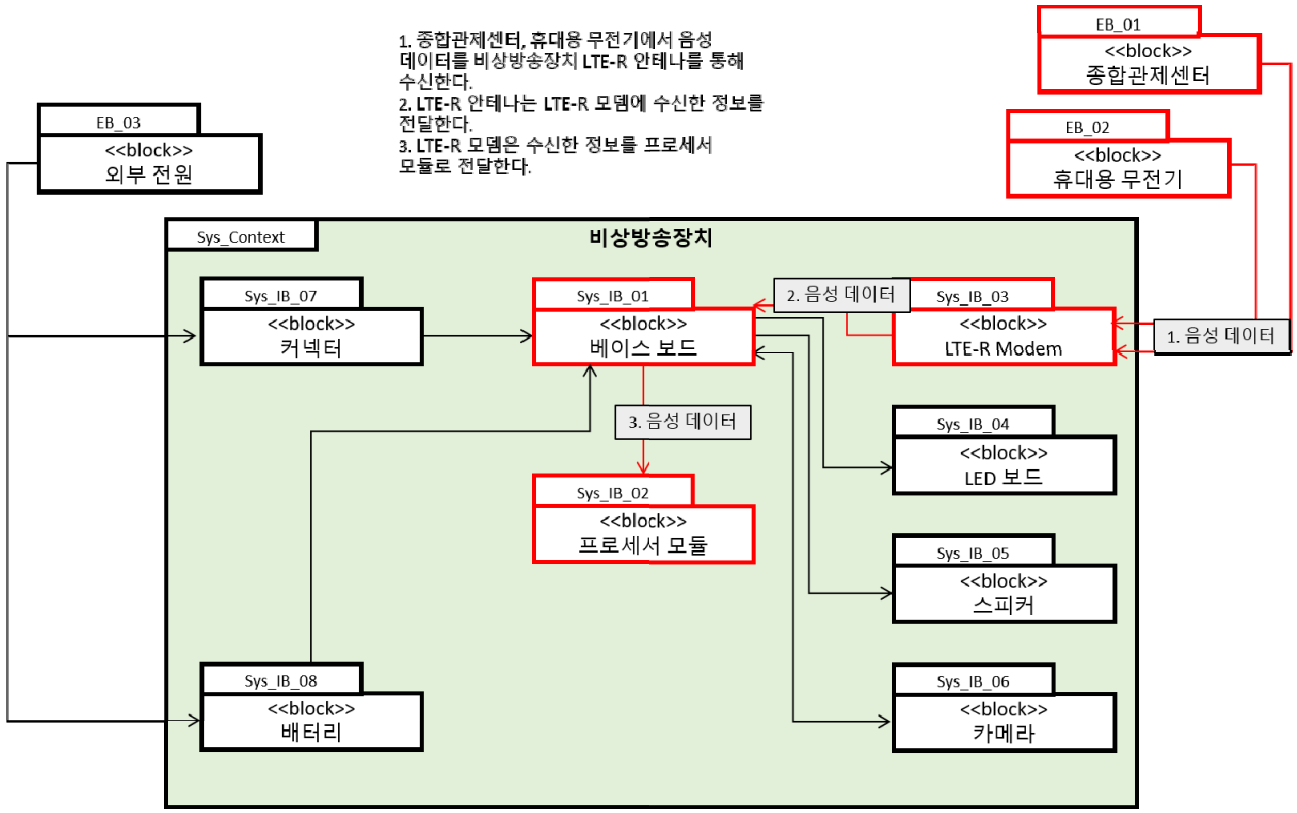


그림 188 비상방송장치의 '음성 데이터 수신' 기능에 대한 기능 흐름 분석(예시)

기능 분석 결과를 토대로 시스템 위험원을 식별한다. 기능의 위험원은 기능을 대상으로 HAZOP 을 통해 식별되며 전문가와의 브레인스토밍을 통해 적절한 가이드워드를 선정하였다. 비상방송장치 시스템의 시스템 위험원 분석 결과를 [표 259]에 기술하였다.

표 259 비상방송장치 시스템 위험원 분석 결과(예시)

기능	가이드워드	System HazardID	위험원	초기위험도 평가			잔여위험도 평가			원인	영향		저감대책
				F	S	R	F	S	R		System	Safty/Service	
509 음성 데이터 수신	No	Sys_Haz_001	음성 데이터 수신 불가	F5	C3	A	F2	C3	B	1. 내부 부품 소손(LTE-R 모뎀/베이스 보드/프로세서 모듈)	방송 불가	비상 발생 인지 불가	1. 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										2. 통신 불량	방송 불가	비상 발생 인지 불가	1.통신규격을통한신뢰성확보(이동통신용무선설비(LTE)기술기준,망사업자) 2.LTE-R수신감도가-94.8dBm이상
										3. 전자파 간섭	방송 불가	비상 발생 인지 불가	1.EMC규격에부합하도록설계 2.EMC시험을통해부합성확인
										4. 소프트웨어 오류	방송 불가	비상 발생 인지 불가	1. 정적, 동적 분석을 통한 SW 품질 확보
										5. 전원 없음	방송 불가	비상 발생 인지 불가	1.외부전원에대한무결점확보 2.외부전원부재시비상방송장치유지를위한배터리설계적용
	Part of	Sys_Haz_002	음성 데이터 일부의 수신	F4	B3	B	F1	C3	C	1. 내부 부품 소손(LTE-R 모뎀/베이스 보드/프로세서 모듈)	방송 끊김	정확한 상황 인지 불가	1. 인증된 부품사양서 기반 신뢰성이 확보된 부품사용
										2. 통신 불량	방송 끊김	정확한 상황 인지 불가	1.통신규격을통한신뢰성확보(이동통신용무선설비(LTE)기술기준,망사업자) 2.LTE-R수신감도가-94.8dBm이상
										3. 전자파 간섭	방송 끊김	정확한 상황 인지 불가	1.EMC규격에부합하도록설계 2.EMC시험을통해부합성확인

○ 시스템 위험원 분석 결론

비상방송장치에 대한 시스템 위험분석 결과 총 16개의 위험원이 식별되었다. 기능에 대한 모든 위험원이 식별되었으며, 컴포넌트 기능에 대한 원인도 식별하였다. 식별된 위험원에 대해 초기 위험도 평가를 통해 추가적인 위험도 저감대책이 요구되는 위험원에 대해서는 해당 위험도를 최소 허용가능한 수준으로 저감시키기 위한 안전 요구사항을 수립함으로써 최종적인 잔여 위험도 평가를 통해서는 해당 위험원이 허용가능한 수준이하로 위험도가 저감되었음을 확인하였다.

○ 인터페이스 위험원 분석

앞서 비상방송장치 시스템에 대한 내/외부 인터페이스 분석을 통해 시스템에서 상호간 정보를 주고받는 장비와 정보들을 식별하였다. 이를 통해 인터페이스 위험원을 식별한다. 인터페이스의 위험원은 인터페이스를 대상으로 HAZOP 분석을 통해 식별되며 전문가와의 브레인스토밍을 통해 적절한 가이드워드를 선정하였다. 비상방송장치 시스템의 인터페이스 위험원 분석 결과를 [표 260]에 기술하였다.

표 260 비상방송장치 시스템의 인터페이스 위험원 분석 결과

인터페이스 ID	인터페이스	위험원	초기 위험도 평가			잔여 위험도 평가			원인	영향		저감대책
			F	S	R	F	S	R		System	Safty/Service	
IN_Ex_002	외부로부터 전력 수신	외부로부터 전력 수신 불가	F4	C1	B	F2	C1	C	1. 전원없음	방송 불가	비상 발생 인지 불가	1.비상방송장치로의 전원공급에 대한 무결성 확보
									2. 연결 케이블 오류	방송 불가	비상 발생 인지 불가	1.내구성이 확보된 인증된 케이블 사용 2.케이블에 태그부착으로 오결선 예방
									3. 전자파 간섭	방송 불가	비상 발생 인지 불가	1.EMC규격에 부합하도록 설계 2.EMC시험을 통해 부합성 확인
		외부로부터 미달 전력 수신	F4	C1	B	F2	C1	C	1. 연결 케이블 오류	방송 불가	비상 발생 인지 불가	1.내구성이 확보된 인증된 케이블 사용
									2. 외부전원장치 고장	방송 불가	비상 발생 인지 불가	1.비상방송장치로의 전원공급에 대한 무결성 확보
									3. 전자파 간섭	방송 불가	비상 발생 인지 불가	1.EMC규격에 부합하도록 설계 2.EMC시험을 해 부합성 확인
		외부로부터 과전력 수신	F4	C1	B	F2	C1	C	1 연결 케이블 오류	방송 불가	비상 발생 인지 불가	1.내구성이 확보된 인증된 케이블 사용
									2. 외부전원장치 고장	방송 불가	비상 발생 인지 불가	1.비상방송장치로의 전원공급에 대한 무결성확보 2.비상방송장치의 과전압 보호회로 설계 적용
									3. 전자파 간섭	방송 불가	비상 발생 인지 불가	1.EMC규격에 부합하도록 설계 2.EMC시험을 통해 부합성 확인
IN_001	LTE 모뎀은 프로세서 모듈로	LTE모뎀은 프로세서 모듈로 정보(음성	F5	C3	A	F2	C3	B	1. PCB 아트웍 고장	음성 데이터 전송 불가		1.EN 50124에 부합하도록 PCB 아트웍 설계
									2. 전자파 간섭	음성 데이터 전송 불가		1.EMC규격에 부합하도록 설계 2.EMC시험을 통해 부합성 확인

인터페이스 ID	인터페이스	위험원	초기 위험도 평가			잔여 위험도 평가			원인	영향		저감대책
			F	S	R	F	S	R		System	Safty/Service	
	정보(음성 데이터) 전송	데이터) 전송 불가										
		LTE모뎀은 프로세서 모듈로 잘못된 정보(음성 데이터) 전송	F4	C1	B	F2	C1	C	1. LTE모뎀 고장	잘못된 방송	정확한 상황 인지 불가	1.내부문서를 통한 신뢰성이 확보된 부품사용(부품 사양서 확인)
									2. PCB 아트웍 고장	잘못된 방송	정확한 상황 인지 불가	1.EN 50124에 부합하도록 PCB 아트웍 설계
									3. 전자파 간섭	잘못된 방송	정확한 상황 인지 불가	1.EMC규격에부합하도록설계 2.EMC시험을통해부합성확인

○ 인터페이스 위험원 분석 결론

비상방송장치 시스템에 대한 인터페이스 위험원 분석 결과 총 8개의 위험원이 식별되었다. 인터페이스에 대한 모든 위험원이 식별되었으며, 외부 인터페이스와 관련된 인터페이스에 대한 원인도 식별하였다. 식별된 위험원에 대해 초기 위험도 평가를 통해 추가적인 위험도 저감대책이 요구되는 위험원에 대해서는 해당 위험도를 최소 허용가능한 수준으로 저감시키기 위한 안전 요구사항을 수립함으로써 최종적인 잔여 위험도 평가를 통해서는 해당 위험원이 허용가능한 수준이하로 위험도가 저감되었음을 확인하였다.

○ 비상방송장치 위험원 분석 결론

비상방송장치 시스템 수행 범위를 토대로 기술된 위험원 분석 방법론에 기반하여 비상방송장치 시스템에 대한 시스템 위험원 분석(SHA), 인터페이스 위험원 분석(IHA)를 수행하였다. 결과적으로 시스템 수준의 기능, 인터페이스를 대상으로 위험원에 대한 분석을 수행하였으며, 시스템 위험원 분석 결과 총 15개의 위험원이 식별되었고, 인터페이스 위험원 분석 결과 총 14개의 위험원이 식별되었다. 시스템 기능 및 인터페이스에 대한 모든 위험원이 식별되었으며, 기능과 관련된 인터페이스에 대한 원인도 식별하였다. 위험도 평가 결과는 다음과 같다.

표 261 시스템 위험원 위험도 평가 결과

위험도 평가 구분	위험도 평가 결과		
	C	B	A
초기 위험도 평가 결과	0	54	11
잔여위험도 평가 결과	54	11	0

표 262 인터페이스 위험원 위험도 평가 결과

위험도 평가 구분	위험도 평가 결과		
	C	B	A
초기 위험도 평가 결과	6	23	7
잔여위험도 평가 결과	29	7	0

식별된 위험원에 대해 초기 위험도 평가를 통해 추가적인 위험도 저감대책이 요구되는 위험원에 대해서는 해당 위험도를 최소 허용 가능한 수준으로 저감시키기 위한 안전 요구사항을 수립함으로써 최종적인 잔여 위험도 평가를 통해서는 해당 위험원이 허용 가능한 수준이하로 위험도가 저감되었음을 확인하였으며 총 21개의 안전 요구사항이 도출되었다. 안전 요구사항 도출 결과는 다음과 같다.

표 263 비상방송시스템의 안전 요구사항 도출 결과

SR ID	안전 요구사항 설명	Hazard ID
SR_001	인증된 부품사양서 기반 신뢰성이 확보된 부품사용	Sys_Haz_001 Sys_Haz_002 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_006 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_010 Sys_Haz_011 Sys_Haz_012 Sys_Haz_013 Sys_Haz_015
SR_002	통신규격을 통한 신뢰성 확보(이동통신용 무선설비(LTE)기술기준,망사업자)	Sys_Haz_001 Sys_Haz_002 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_013 Sys_Haz_014
SR_003	LTE-R 수신감도가 -94.8dBm이상	Sys_Haz_001 Sys_Haz_002 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_013 Sys_Haz_014

SR ID	안전 요구사항 설명	Hazard ID
SR_004	EMC 규격에 부합하도록 설계	Sys_Haz_001 Sys_Haz_002 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_006 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_010 Sys_Haz_011 Sys_Haz_012 Sys_Haz_013 Sys_Haz_014 Sys_Haz_015 In_Haz_001 In_Haz_002 In_Haz_003 In_Haz_004 In_Haz_005 In_Haz_006 In_Haz_007 In_Haz_008 In_Haz_009 In_Haz_010 In_Haz_011 In_Haz_012
SR_005	EMC 시험을 통해 부합성 확인	Sys_Haz_001 Sys_Haz_002 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_006 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_010 Sys_Haz_011 Sys_Haz_012 Sys_Haz_013 Sys_Haz_014 Sys_Haz_015 In_Haz_001 In_Haz_002 In_Haz_003 In_Haz_004 In_Haz_005 In_Haz_006

SR ID	안전 요구사항 설명	Hazard ID
		In_Haz_007 In_Haz_008 In_Haz_009 In_Haz_010 In_Haz_011 In_Haz_012
SR_006	정적, 동적 분석을 통한 소프트웨어 품질 확보	Sys_Haz_001 Sys_Haz_003 Sys_Haz_004 Sys_Haz_005 Sys_Haz_006 Sys_Haz_007 Sys_Haz_008 Sys_Haz_009 Sys_Haz_010 Sys_Haz_011 Sys_Haz_012 Sys_Haz_013 Sys_Haz_015
SR_007	외부 전원에 대한 무결성 확보	Sys_Haz_001 Sys_Haz_003 Sys_Haz_004 Sys_Haz_007 Sys_Haz_010 Sys_Haz_013
SR_008	외부 전원 부재 시 비상방송장치 유지를 위한 배터리 설계적용	Sys_Haz_001 Sys_Haz_003 Sys_Haz_004 Sys_Haz_007 Sys_Haz_010 Sys_Haz_013
SR_009	특정 주기로 상태정보 인식 불가시 소프트웨어 재부팅 기능 설계	Sys_Haz_003
SR_010	같은 편성내 옆객실에 장착된 비상방송장치를 활용하여 비상발생메시지 전송	Sys_Haz_004 Sys_Haz_005
SR_011	내부문서를 통한 신뢰성이 확보된 케이블 및 커넥터 사용(부품사양서 확인)	Sys_Haz_006 In_Haz_005 In_Haz_008 In_Haz_010 In_Haz_012
SR_012	장치 설치 시 화각 조정 및 확인	Sys_Haz_014
SR_013	비상방송장치로의 전원공급에 대한 무결성 확보	In_Haz_001 In_Haz_002 In_Haz_003 In_Haz_013
SR_014	내구성이 확보된 인증된 케이블 사용	In_Haz_001 In_Haz_002 In_Haz_003

SR ID	안전 요구사항 설명	Hazard ID
		In_Haz_013 In_Haz_014
SR_015	케이블에 태그부착으로 오결선 예방	In_Haz_001 In_Haz_013 In_Haz_014
SR_016	비상방송장치의 과전압 보호회로 설계적용	In_Haz_003
SR_017	EN 50124에 부합하도록 PCB 아트웍 설계	In_Haz_004 In_Haz_005 In_Haz_006 In_Haz_007 In_Haz_008 In_Haz_009 In_Haz_010 In_Haz_011 In_Haz_012
SR_018	O&M 매뉴얼을 통한 배터리 상태의 주기적 확인	In_Haz_013
SR_019	배터리 충전상태 표시	In_Haz_014
SR_020	음성 데이터 수신이 불가할 경우, 해당 비상방송장치의 고장을 감시할 수 있는 기능 설계 (종합관제센터 ESR)	Sys_Haz_001 Sys_Haz_002
SR_021	해당 객실내 비상방송장치에서 상태정보 전송이 불가할 경우, 종합관제실에서 상태정보 미전송 비상방송장치에 대한 식별기능에 대한 감시기능 설계 (종합관제센터 ESR)	Sys_Haz_003 Sys_Haz_004

제 2 절 모형 철도 적용 사례

1. 모형 철도 적용 개요

- ○ 개발된 철도 분야 소프트웨어 신뢰성 및 안전성 확보를 위한 가이드의 적용 및 검증
- ○ 개발된 가이드들의 시범 적용 기간 부족에 대한 대안
- ○ 추후 철도 분야 소프트웨어 제품 개발 검증을 위한 모형 철도 프레임워크를 구축

2. 철도 건널목 시스템

- ○ 모형 철도를 활용한 철도 건널목(Level Crossing) 시스템 개발
- ○ 건널목 시스템이 열차를 감지하여 차단기를 동작
- ○ 차단기 동작 시 신호등 상태 전환
- ○ 비상등을 사용하여 비상 연락 시스템을 대체
- ○ 본 시범 적용 시에는 안전성 분석을 위해 시스템 안전 무결성 등급(SIL)을 2 등급으로 적용

2.1. 시스템 개요

- ○ 시스템 기능

표 264 철도 건널목 시스템 기능

항목	시스템 기능	기능 상세 설명
1	차단기 제어	차단기를 상승하는 기능
		차단기를 하강하는 기능
2	신호등 제어	빨간불을 점등하는 기능
		빨간불을 소등하는 기능
		초록불을 점등하는 기능
		초록불을 소등하는 기능
3	비상등 제어	비상등을 점등하는 기능
		비상등을 소등하는 기능
4	아두이노 CPU 제어	열차 접근 여부에 따라 서보 모터 각도를 결정하는 기능
		열차 접근 여부에 따라 신호등 상태를 결정하는 기능
		열차 접근 여부에 따라 비상등 상태를 결정하는 기능
		Wifi 통신 명령을 내리는 기능
5	아두이노 서보 모터 제어	차단기를 상승 및 하강시키기 위해 결정된 서보 모터 각도로 제어하는 기능
6	아두이노 Wifi 제어	Wifi 통신 명령을 수행하는 기능
7	아두이노 적외선 센서 제어	열차 접근을 감지하는 기능

○ 시스템 구성

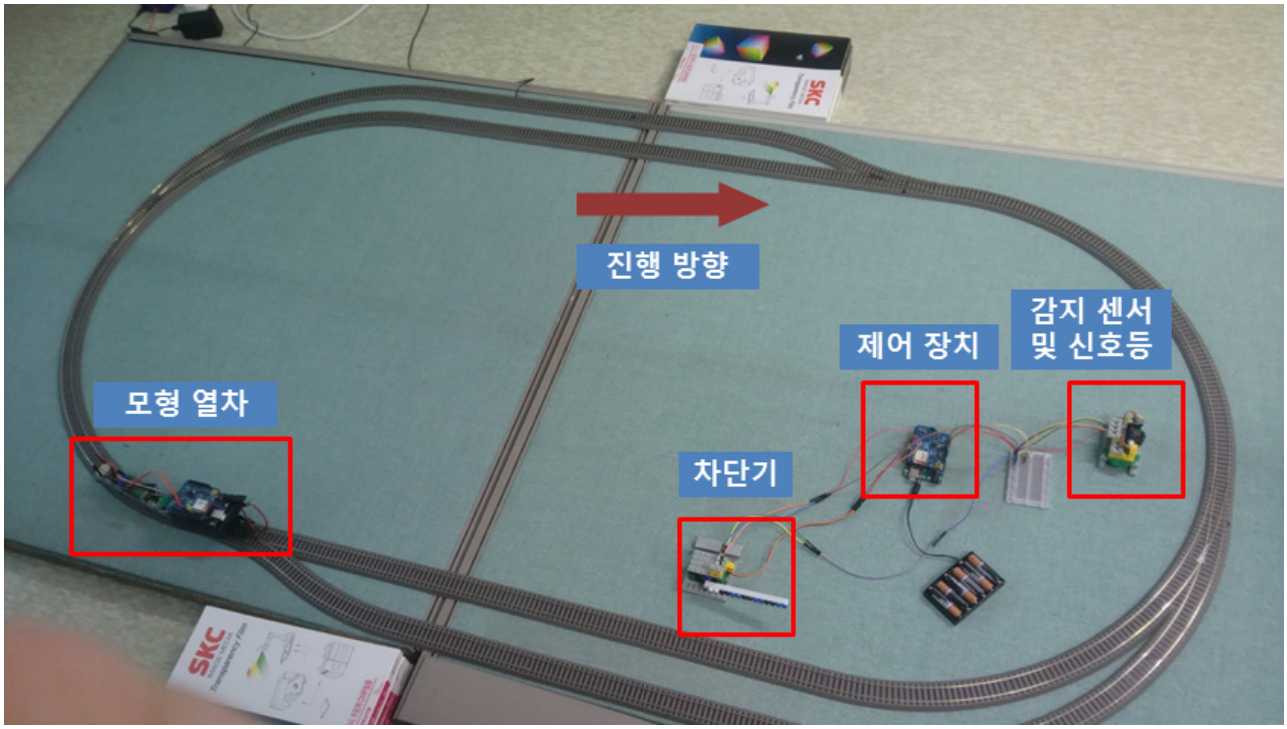


그림 189 철도 건널목 시스템 구성도

2.2. 시범 적용 가이드

○ 시스템 안전성 분석 가이드

○ 소프트웨어 요구사항 명세 가이드

○ 소프트웨어 아키텍처 및 설계 가이드

○ 소프트웨어 컴포넌트 설계 명세 가이드

2.3. 시스템 안전성 분석 가이드 적용

○ 건널목(LC)의 고장 유형(Failure Mode) 및 기능 고장(Function Failure) 식별

- 건널목은 Controlled Process, Controller, Actuator, Sensor의 서브시스템 구성
- 각 서브시스템에 대한 고장 유형 식별
- 식별된 서브시스템에 대한 기능 고장 식별

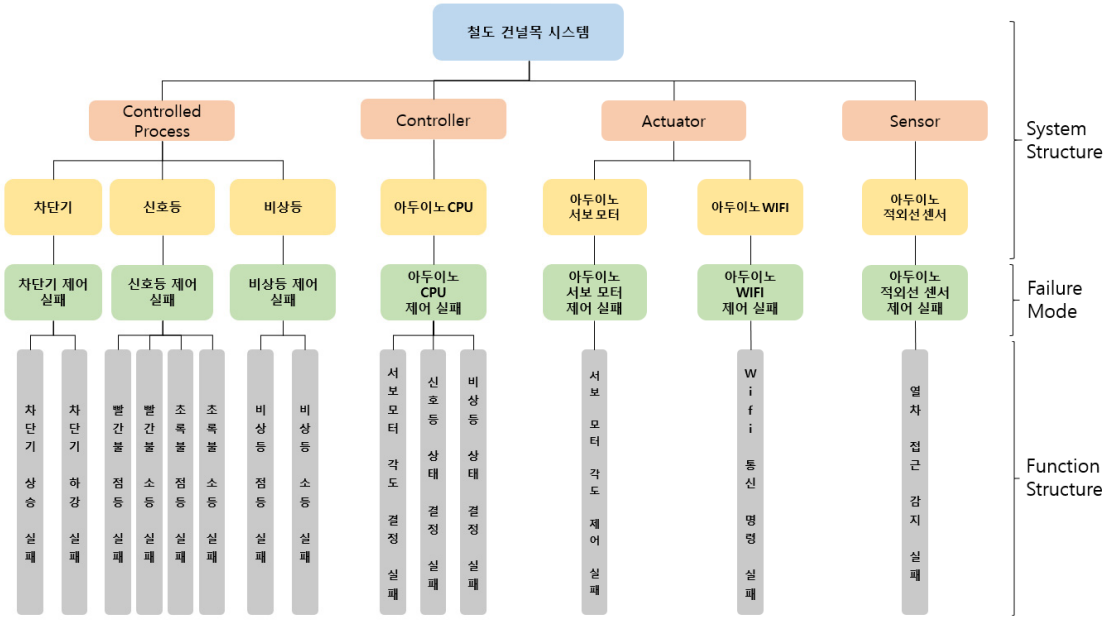


그림 190 건널목(LC) 시스템 고장 유형 및 기능 고장

○ PHA 수행 및 산출물

- 식별된 고장 유형 및 기능 고장을 기반으로 PHA 수행
- 각 기능 고장을 일으키는 고장 원인을 식별
- 각 기능 고장으로 인한 영향을 식별

고장 유형	기능 고장	고장 원인	영향
차단기 제어 실패	차단기 상승 실패	1. 차단기 파손 2. 잘못된 차단기 제어 명령 전달 3. 차단기 제어 명령 미전달	교통 혼란
	차단기 하강 실패		인명 피해
신호등 제어 실패	빨간불 점등 실패	1. 빨간등 파손 2. 파란등 파손 3. 잘못된 신호등 제어 명령 전달 4. 신호등 제어 명령 미전달	인명 피해
	빨간불 소등 실패		교통 혼란
	초록불 점등 실패		교통 혼란
	초록불 소등 실패		인명 피해
비상등 제어 실패	비상등 점등 실패	1. 비상등 파손 2. 잘못된 비상등 제어 명령 전달 3. 비상등 제어 명령 미전달	인명 피해
	비상등 소등 실패		교통 혼란
아두이노 CPU 제어 실패	열차 접근 여부에 따른 서보 모터 각도 결정 실패	1. 아두이노 CPU 파손 2. 정해진 범위 이외의 적외선 센서 값 전달 3. 적외선 센서 값 미전달	교통 혼란 및 인명 피해
	열차 접근 여부에 따른 신호등 상태 결정 실패		교통 혼란 및 인명 피해
	열차 접근 여부에 따른 비상등 상태 결정 실패		교통 혼란 및 인명 피해
	Wifi 통신 명령 실패		교통 혼란 및 인명 피해
아두이노 서보 모터 제어 실패	차단기를 상승 및 하강 시키기 위해 결정된 서보 모터 각도로 제어 실패	1. 서보 모터 파손 2. 잘못된 서보 모터 제어 명령 전달 3. 서보 모터 제어 명령 미전달	교통 혼란 및 인명 피해
아두이노 Wifi 제어 실패	Wifi 통신 실패	1. Wifi 파손 2. 잘못된 Wifi 통신 명령 전달 3. Wifi 통신 명령 미전달	교통 혼란 및 인명 피해
아두이노 적외선 센서 제어 실패	열차 접근 감지 실패	1. 적외선 센서 미작동 2. 적외선 센서 파손	인명 피해

그림 191 PHA 수행 결과 표

○ FMEA 수행 및 산출물

- PHA 수행 산출물을 기반으로 FMEA 수행
- 각 기능 고장 별로 위험도 평가 수행
- 수행된 위험도 평가를 기반으로 예방 조치 및 설계 반영 방안 도출
- 예방 조치 및 설계 반영 방안에 대한 안전 기능 정의

시스템 명칭	시스템 구조		기능	고장 유형 (Failure Mode)	고장 영향 (Failure Effects)	RPN 평가			RPN	고장 원인 (Failure Cause)	예방 조치 및 설계 반영 방안	안전 기능
	Layer-1	Layer-2				심각도	발생빈도	허용수준				
LC	Controlled Process	차단기	차단기 상승 기능	차단목 제어 실패	차단목 상승 실패	1	3	2	6	1. 차단기 파손 2. 잘못된 차단기 제어 명령 전달 3. 차단기 제어 명령 미전달	적외선 센서의 이중화 및 서버를 활용한 비상 상태 모니터링	1. 적외선 센서의 이중화에 따른 적외선 센서 결함 감지 기능 2. 통신 장애를 비롯한 시스템의 비상 상태를 대비한 Heartbeat 기능 3. 통신 장애를 비롯한 시스템의 비상 상태 발생 시 열차로 비상 정지를 전달하는 기능
			차단기 하강 기능		차단목 하강 실패	5	3	4	60			
		신호등	빨간등 점등 기능	신호등 제어 실패	빨간등 점등 실패	5	3	4	60	1. 빨간등 파손 2. 파란등 파손 3. 잘못된 신호등 제어 명령 전달 4. 신호등 제어 명령 미전달		
			빨간등 소등 기능		빨간등 소등 실패	1	3	2	6			
			초록등 점등 기능		초록등 점등 실패	1	3	2	6			
			초록등 소등 기능		초록등 소등 실패	5	3	4	60			
		비상등	비상등 점등 기능	비상등 제어 실패	비상등 점등 실패	5	3	4	60	1. 비상등 파손 2. 잘못된 비상등 제어 명령 전달 3. 비상등 제어 명령 미전달		
			비상등 소등 기능		비상등 소등 실패	1	3	2	6			
	Controller	아두이노 CPU	열차 접근 여부에 따른 서보 모터 각도 결정 기능	아두이노 CPU 제어 실패	열차 접근 여부에 따른 서보 모터 각도 결정 실패	5	3	4	60	1. 아두이노 CPU 파손 2. 정해진 범위 이외의 적외선 센서 값 전달 3. 적외선 센서 값 미전달		
			열차 접근 여부에 따른 신호등 상태 결정 기능		열차 접근 여부에 따른 신호등 상태 결정 실패	5	3	4	60			
			열차 접근 여부에 따른 비상등 상태 결정 기능		열차 접근 여부에 따른 비상등 상태 결정 실패	5	3	4	60			
			Wifi 통신 명령 기능		Wifi 통신 명령 실패	5	3	4	60			
	Actuator	아두이노 서보 모터	차단기를 상승 및 하강 시키기 위해 결정된 서보 모터 각도로 제어 기능	아두이노 서보 모터 제어 실패	차단기를 상승 및 하강 시키기 위해 결정된 서보 모터 각도로 제어 실패	5	3	4	60	1. 서보 모터 파손 2. 잘못된 서보 모터 제어 명령 전달 3. 서보 모터 제어 명령 미전달		
		아두이노 Wifi	Wifi 통신 기능	아두이노 Wifi 제어 실패	Wifi 통신 실패	5	3	4	60	1. Wifi 파손 2. 잘못된 Wifi 통신 명령 전달 3. Wifi 통신 명령 미전달		
	Sensor	아두이노 적외선 센서	열차 접근 감지 기능	아두이노 적외선 센서 제어 실패	열차 접근 감지 실패	5	3	4	60	1. 적외선 센서 미작동 2. 적외선 센서 파손		

그림 192 FMEA 수행 결과 표

○ 시스템 및 안전 요구사항 도출

안전성 분석 결과와 시스템 기능에 대한 자료를 바탕으로 시스템 요구사항과 시스템 안전 요구사항 도출하였다.

표 265 시스템 요구사항

요구사항 ID	요구사항
시스템 요구사항 (System Requirement)	
차단기 제어	
SRS_LC_REQ1.1	차단기는 차단기를 상승 시킬 수 있어야 한다.
SRS_LC_REQ1.2	차단기는 차단기를 하강 시킬 수 있어야 한다.
신호등 제어	
SRS_LC_REQ2.1	신호등은 빨간등을 점등할 수 있어야 한다.
SRS_LC_REQ2.2	신호등은 빨간등을 소등할 수 있어야 한다.
SRS_LC_REQ2.3	신호등은 초록등을 점등할 수 있어야 한다.
SRS_LC_REQ2.4	신호등은 초록등을 소등할 수 있어야 한다.
비상등 제어	
SRS_LC_REQ3.1	비상등은 비상등을 점등할 수 있어야 한다.
SRS_LC_REQ3.2	비상등은 비상등을 소등할 수 있어야 한다.
아두이노 CPU 제어	
SRS_LC_REQ4.1	아두이노 CPU는 열차 접근 여부에 따른 서보 모터 각도를 결정할 수 있어야 한다.
SRS_LC_REQ4.2	아두이노 CPU는 열차 접근 여부에 따른 신호등 상태를 결정할 수 있어야 한다.
SRS_LC_REQ4.3	아두이노 CPU는 열차 접근 여부에 따른 비상등 상태를 결정할 수 있어야 한다.
SRS_LC_REQ4.4	아두이노 CPU는 Wifi 통신 명령을 제어할 수 있어야 한다.
아두이노 서보 모터 제어	
SRS_LC_REQ5.1	아두이노 서보 모터는 차단기를 상승 및 하강시키기 위해 결정된 서보 모터 각도로 제어할 수 있어야 한다.
아두이노 Wifi 제어	
SRS_LC_REQ6.1	아두이노 Wifi는 Wifi 통신할 수 있어야 한다.
아두이노 적외선 센서 제어	
SRS_LC_REQ7.1	아두이노 적외선 센서는 열차 접근을 감지할 수 있어야 한다.

요구사항 ID	요구사항
인터페이스 (Interface Requirement)	
SRS_LC_REQ8.1	아두이노 CPU는 결정된 신호등 상태로 변경할 수 있도록 신호등에게 제어 명령을 내릴 수 있어야 한다.
SRS_LC_REQ8.2	아두이노 CPU는 결정된 비상등 상태로 변경할 수 있도록 비상등에게 제어 명령을 내릴 수 있어야 한다.
SRS_LC_REQ8.3	아두이노 CPU는 결정된 서보 모터 각도 값을 서보 모터가 수행할 수 있도록 아두이노 서보 모터에게 제어 명령을 내릴 수 있어야 한다.
SRS_LC_REQ8.4	아두이노 CPU는 서버와 Wifi 연결을 할 수 있도록 아두이노 Wifi에게 제어 명령을 내릴 수 있어야 한다.
SRS_LC_REQ8.5	아두이노 CPU는 서버와 Wifi 통신을 할 수 있도록 아두이노 Wifi에게 제어 명령을 내릴 수 있어야 한다.
성능 (Performance Requirement)	
SRS_LC_REQ9.1	차단기 상승 시 차단기는 1초 이내에 올라가야 한다.
SRS_LC_REQ9.2	차단기 하강 시 차단기는 1초 이내에 내려가야 한다.
SRS_LC_REQ9.3	아두이노 적외선 센서의 측정 거리가 5cm 이내일 경우, 열차가 접근한 것으로 판단해야 한다.

표 266 시스템 안전 요구사항

요구사항 ID	요구사항
시스템 안전 요구사항 (System Safety Requirement)	
적외선 센서의 이중화	
SSRS_LC_REQ1.1	적외선 센서를 이중화 하고 이에 따라 적외선 센서 결함 발생 시 그 여부를 감지하고 판단해야 한다.
비상 상태 모니터링	
SSRS_LC_REQ2.1	서버를 활용하여 통신 장애를 비롯한 비상 상태를 모니터링하기 위하여 서버와 Heartbeat를 주고받는다.
SSRS_LC_REQ2.2	서버를 활용하여 통신 상태를 비롯한 비상 상태 발생 시 서버는 각 열차로 비상 정지 명령을 전달해야 한다.

2.4. 소프트웨어 요구사항 명세 가이드 적용

앞 단계에서 식별된 시스템 및 안전 요구사항을 기반으로 소프트웨어 요구사항과 안전 요구사항을 도출하고, 상세 요구사항을 명세하였다.

표 267 소프트웨어 요구사항

요구사항 ID (관련 사항)	요구사항
소프트웨어 요구사항 (Software Requirements)	
아두이노 CPU 내장 소프트웨어	
소프트웨어RS_LC_REQ1.1 (관련 사항: SRS_LC_REQ4.1)	LC 소프트웨어는 열차 접근 여부에 따른 서보 모터 각도를 결정할 수 있어야 한다.
소프트웨어RS_LC_REQ1.2 (관련 사항: SRS_LC_REQ4.2)	LC 소프트웨어는 열차 접근 여부에 따른 신호등 상태를 결정할 수 있어야 한다.
소프트웨어RS_LC_REQ1.3 (관련 사항: SRS_LC_REQ4.3)	LC 소프트웨어는 열차 접근 여부에 따른 비상등 상태를 결정할 수 있어야 한다.
소프트웨어RS_LC_REQ1.4 (관련 사항: SRS_LC_REQ4.4)	LC 소프트웨어는 Wifi 통신 명령을 제어할 수 있어야 한다.
인터페이스 (Interface Requirement)	
소프트웨어RS_LC_REQ2.1 (관련 사항: SRS_LC_REQ8.1)	LC 소프트웨어는 결정된 신호등 상태로 변경할 수 있도록 신호등에게 제어 명령을 내릴 수 있어야 한다.
소프트웨어RS_LC_REQ2.2 (관련 사항: SRS_LC_REQ8.2)	LC 소프트웨어는 결정된 비상등 상태로 변경할 수 있도록 비상등에게 제어 명령을 내릴 수 있어야 한다.
소프트웨어RS_LC_REQ2.3 (관련 사항: SRS_LC_REQ8.3)	LC 소프트웨어는 결정된 서보 모터 각도 값을 아두이노 서보 모터에게 제어 명령으로 전달 할 수 있어야 한다.
소프트웨어RS_LC_REQ2.4 (관련 사항: SRS_LC_REQ8.4)	LC 소프트웨어는 서버와 Wifi 연결을 할 수 있도록 아두이노 Wifi에게 제어 명령을 내릴 수 있어야 한다.
소프트웨어RS_LC_REQ2.5 (관련 사항: SRS_LC_REQ8.5)	LC 소프트웨어는 서버와 Wifi 통신을 할 수 있도록 아두이노 Wifi에게 제어 명령을 내릴 수 있어야 한다.
성능 (Performance Requirement)	
소프트웨어RS_LC_REQ3.1 (관련 사항: SRS_LC_REQ9.1)	LC 소프트웨어는 차단기가 차단기 상승을 1초 이내에 수행하도록 제어해야 한다.
소프트웨어RS_LC_REQ3.2 (관련 사항: SRS_LC_REQ9.2)	LC 소프트웨어는 차단기가 차단기 하강을 1초 이내에 수행하도록 제어해야 한다.
소프트웨어RS_LC_REQ3.3 (관련 사항: SRS_LC_REQ9.3)	LC 소프트웨어는 아두이노 적외선 센서의 측정 거리가 5cm 이내일 경우, 열차가 접근한 것으로 판단해야 한다.

표 268 소프트웨어 안전 요구사항

요구사항 ID (관련 사항)	요구사항
소프트웨어 안전 요구사항 (Software Safety Requirement)	
소프트웨어SRS_LC_REQ1.1 (관련 사항: SSRS_LC_REQ1.1)	LC 소프트웨어는 적외선 센서 결함 발생 시 그 여부를 감지하고 판단해야 한다.
소프트웨어SRS_LC_REQ2.1 (관련 사항: SSRS_LC_REQ2.1)	LC 소프트웨어는 통신 장애를 비롯한 비상 상태를 모니터링하기 위하여 서버와 Heartbeat를 주고받는다.
소프트웨어SRS_LC_REQ2.2 (관련 사항: SSRS_LC_REQ2.2)	LC 소프트웨어는 통신 상태를 비롯한 비상 상태 발생 시 서버는 각 열차로 비상 정지 명령을 전달해야 한다.

○ 상세 기능 요구사항

표 269 기능 요구사항 (서보 모터 각도 결정)

Use Case Name	서보 모터 각도 결정		ID	소프트웨어RS_LC_REQ1.1
요구사항	LC 소프트웨어는 열차 접근 여부에 따른 서보 모터 각도를 결정			
Actor	서보 모터			
Trigger	SRS_LC_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 열차 접근에 대한 아날로그 값을 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	열차 접근 여부를 결정한다.		
	4	열차 접근 여부에 따라 서보 모터 각도를 결정한다.		
	5	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 접근 여부에 따른 서보 모터 각도 결정			
관련 요구사항	SRS_LC_REQ1.1, SRS_LC_REQ1.2			
상세 요구사항	해당사항 없음			

표 270 기능 요구사항 (신호등 상태 결정)

Use Case Name	신호등 상태 결정		ID	소프트웨어RS_LC_REQ1.2
요구사항	LC 소프트웨어는 열차 접근 여부에 따른 신호등 상태를 결정할 수 있어야 한다.			
Actor	신호등			
Trigger	SRS_LC_REQ4.2			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 열차 접근에 대한 아날로그 값을 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	열차 접근 여부를 결정한다.		
	4	열차 접근 여부에 따라 신호등 상태를 결정한다.		
	5	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 접근 여부에 따른 신호등 상태 변경			
관련 요구사항	SRS_LC_REQ2.1, SRS_LC_REQ2.2, SRS_LC_REQ2.3, SRS_LC_REQ2.4			
상세 요구사항	해당사항 없음			

표 271 기능 요구사항 (비상등 상태 결정)

Use Case Name	비상등 상태 결정		ID	소프트웨어RS_LC_REQ1.3
요구사항	LC 소프트웨어는 열차 접근 여부에 따른 비상등 상태를 결정할 수 있어야 한다.			
Actor	비상등			
Trigger	SRS_LC_REQ4.3			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 열차 접근에 대한 아날로그 값을 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	열차 접근 여부를 결정한다.		
	4	열차 접근 여부에 따라 비상등 상태를 결정한다.		
	5	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 접근 여부에 따른 비상등 상태 결정			
관련 요구사항	SRS_LC_REQ3.1, SRS_LC_REQ3.2			
상세 요구사항	해당사항 없음			

표 272 기능 요구사항 (Wifi 통신 명령)

Use Case Name	Wifi 통신 명령		ID	소프트웨어RS_LC_REQ1.4
요구사항	LC 소프트웨어는 Wifi 통신 명령을 제어할 수 있어야 한다.			
Actor	Wifi			
Trigger	SRS_LC_REQ4.4			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 열차 접근에 대한 아날로그 값을 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	열차 접근에 따른 명령을 결정한다.		
	4	Wifi 통신을 이용해 명령을 제어한다.		
	5	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 접근 여부에 따른 Wifi 명령 제어			
관련 요구사항	SRS_LC_REQ4.4, SRS_LC_REQ5.1			
상세 요구사항	해당사항 없음			

표 273 인터페이스 요구사항 (신호등 제어 명령)

Use Case Name	신호등 제어 명령		ID	SRS_LC_REQ2.1
요구사항	LC 소프트웨어는 결정된 신호등 상태로 변경할 수 있도록 신호등에게 제어 명령을 내릴 수 있어야 한다.			
Actor	신호등 제어 명령			
Trigger	SRS_LC_REQ1.2			
Flow of Events				
Basic Flow	Step	Action		
	1	열차 접근 여부를 결정한다.		
	2	열차 접근 여부에 따라 신호등 상태를 결정한다.		
	3	신호등에게 신호등 제어 명령을 내린다.		
	4	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 신호등 제어 명령에 따른 신호등 제어			
관련 요구사항	SRS_LC_REQ8.1			
상세 요구사항	해당사항 없음			

표 274 인터페이스 요구사항 (비상등 제어 명령)

Use Case Name	비상등 제어 명령		ID	SRS_LC_REQ3.1
요구사항	LC 소프트웨어는 결정된 비상등 상태로 변경할 수 있도록 비상등에게 제어 명령을 내릴 수 있어야 한다.			
Actor	비상등			
Trigger	소프트웨어RS_LC_REQ1.3			
Flow of Events				
Basic Flow	Step	Action		
	1	열차 접근 여부를 결정한다.		
	2	열차 접근 여부에 따라 비상등 상태를 결정한다.		
	3	비상등에게 비상등 제어 명령을 내린다.		
	4	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 비상등 제어 명령에 따른 비상등 제어			
관련 요구사항	SRS_LC_REQ8.2			
상세 요구사항	해당사항 없음			

표 275 인터페이스 요구사항 (서보 모터 제어 명령)

Use Case Name	서보 모터 제어 명령		ID	SRS_LC_REQ4.1
요구사항	LC 소프트웨어는 결정된 서보 모터 각도 값을 서보 모터가 수행할 수 있도록 아두이노 서보 모터에게 제어 명령을 내릴 수 있어야 한다.			
Actor	서보 모터			
Trigger	소프트웨어RS_LC_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	열차 접근 여부를 결정한다.		
	2	열차 접근 여부에 따라 서보 모터 각도를 결정한다.		
	3	서보 모터에게 서보 모터 제어 명령을 내린다.		
	4	Use Case 종료		
Pre-conditions	1. 열차 접근 여부를 측정한 아날로그 값			
Post-conditions	1. 서보 모터 제어 명령에 따른 각도 제어			
관련 요구사항	SRS_LC_REQ8.3			
상세 요구사항	해당사항 없음			

표 276 인터페이스 요구사항 (Wifi 연결 제어 명령)

Use Case Name	Wifi 연결 제어 명령		ID	SRS_LC_REQ5.1
요구사항	LC 소프트웨어는 서버와 Wifi 연결을 할 수 있도록 아두이노 Wifi에게 제어 명령을 내릴 수 있어야 한다.			
Actor	Wifi 모듈			
Trigger	소프트웨어RS_LC_REQ1.4			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 CPU를 시작한다.		
	2	서버와 Wifi 연결을 명령한다.		
	3	Wifi 연결을 시도한다.		
	4	서버와 Wifi를 연결한다.		
	5	Use Case 종료		
Pre-conditions	해당사항 없음			
Post-conditions	서버와 Wifi 연결			
관련 요구사항	SRS_LC_REQ8.4, SRS_LC_REQ8.5			
상세 요구사항	해당사항 없음			

○ 상세 비기능 요구사항

표 277 비기능 요구사항

비기능 요구사항	
안전성 (Safety)	소프트웨어 안전 무결성 등급 (SIL)이 2 등급을 만족해야 한다.
제약사항 (Constraints)	LC 소프트웨어는 차단기가 차단기 상승을 1초 이내에 수행하도록 제어해야 한다.
	LC 소프트웨어는 차단기가 차단기 하강을 1초 이내에 수행하도록 제어해야 한다.
	LC 소프트웨어는 아두이노 적외선 센서의 측정 거리가 5cm 이내일 경우, 열차가 접근한 것으로 판단해야 한다.

○ 상세 안전 요구사항

표 278 안전 요구사항 (적외선 센서 결함 판단)

Use Case Name	적외선 센서 결함 판단		ID	소프트웨어SRS_TA_REQ1.1
안전 요구사항	LC 소프트웨어는 적외선 센서 결함 발생 시 그 여부를 감지하고 판단해야 한다.			
Actor	LC 소프트웨어			
Trigger	SSRS_TA_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	적외선 센서가 값을 측정한다.		
	2	아날로그 값을 입력받는다.		
	3	입력받은 아날로그 값을 디지털 값으로 변환한다.		
	4	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - 적외선 센서가 값을 측정하지 못한다.		
	1	Fail-safe 기능을 수행한다.		
	2	Use Case를 종료한다.		
Alternative Flow 2	Step	2a - 아날로그 값이 정의된 범위 외의 값이 들어온다.		
	1	입력받은 값이 범위 내에 있는지 확인한다.		
	2	범위 내에 없을 경우 Fail-safe 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 적외선 센서로부터 들어온 아날로그 값			
Post-conditions	해당사항 없음			
관련 요구사항	해당사항 없음			
상세 요구사항	해당사항 없음			
안전 조건	적외선 센서의 값이 정의된 범위 외의 값이 들어온다.			

표 279 안전 요구사항 (통신 장애 판단)

Use Case Name	통신 장애 판단		ID	소프트웨어SRS_TA_REQ2.1
안전 요구사항	LC 소프트웨어는 통신 장애를 비롯한 비상 상태를 모니터링하기 위하여 서버와 Heartbeat를 주고받는다.			
Actor	LC 소프트웨어			
Trigger	SSRS_TA_REQ2.1			
Flow of Events				
Basic Flow	Step	Action		
	1	서버와 Wifi 연결을 한다.		
	2	서버에게 주기적으로 Heartbeat를 전송한다.		
	3	서버로부터 주기적으로 Heartbeat 값을 전달받는다.		
	4	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - 서버로부터 Heartbeat 값을 전달받지 못한다.		
	1	비상 정지 기능을 수행한다.		
	2	Use Case를 종료한다.		
Pre-conditions	1. 서버와 통신이 정상적으로 연결된다.			
Post-conditions	해당사항 없음			
관련 요구사항	소프트웨어SRS_LC_REQ2.2			
상세 요구사항	해당사항 없음			
안전 조건	서버와 주기적으로 Heartbeat 값을 주고받는다.			

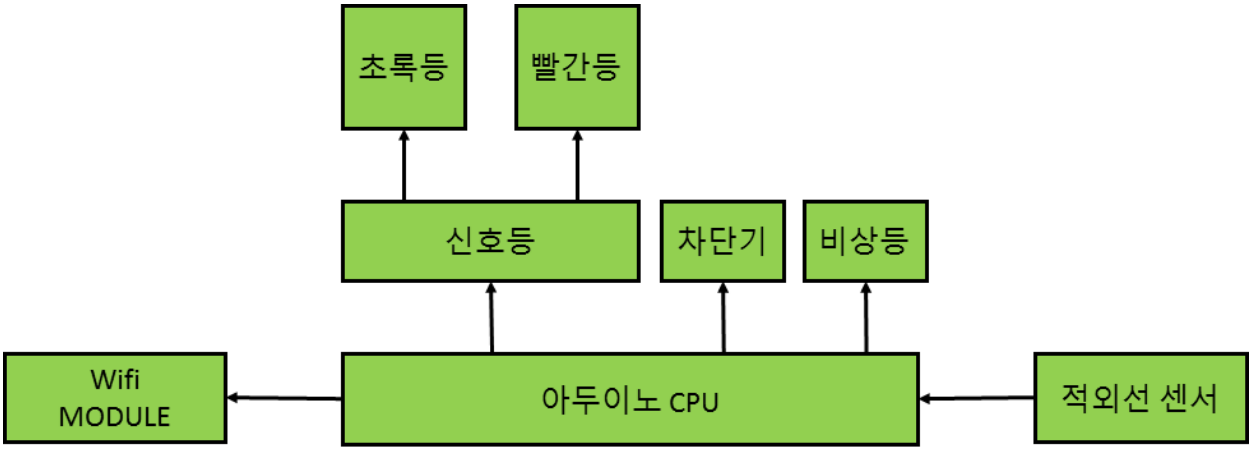
표 280 안전 요구사항 (비상 정지 명령)

Use Case Name	비상 정지 명령		ID	소프트웨어SRS_TA_REQ2.2
안전 요구사항	LC 소프트웨어는 통신 상태를 비롯한 비상 상태 발생 시 서버는 각 열차로 비상 정지 명령을 전달해야 한다.			
Actor	LC 소프트웨어			
Trigger	SSRS_TA_REQ2.2			
Flow of Events				
Basic Flow	Step	Action		
	1	비상 상태가 발생한다.		
	2	서버는 비상 상태를 확인한다.		
	3	서버는 각 열차에게 비상 정지 명령을 전달한다.		
	4	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - 각 열차에게 비상 정지 명령을 전달하지 못한다.		
	1	각 열차는 서버로부터 통신이 오지 않는 것을 확인한다.		
	2	각 열차는 자체적으로 비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 서버와 통신이 정상적으로 연결된다.			
Post-conditions	해당사항 없음			
관련 요구사항	소프트웨어SRS_LC_REQ2.1			
상세 요구사항	해당사항 없음			
안전 조건	비상 상태 발생 시 서버는 각 열차에게 비상 정지 명령을 보낸다.			

2.5. 소프트웨어 아키텍처 및 설계 가이드 적용

○ 건널목 시스템 개념도 도출

그림 193 건널목 시스템 개념도



○ 건널목 소프트웨어 컴포넌트 식별

소프트웨어 요구사항 명세와 시스템 요구사항으로부터 소프트웨어 컴포넌트를 식별하였다.

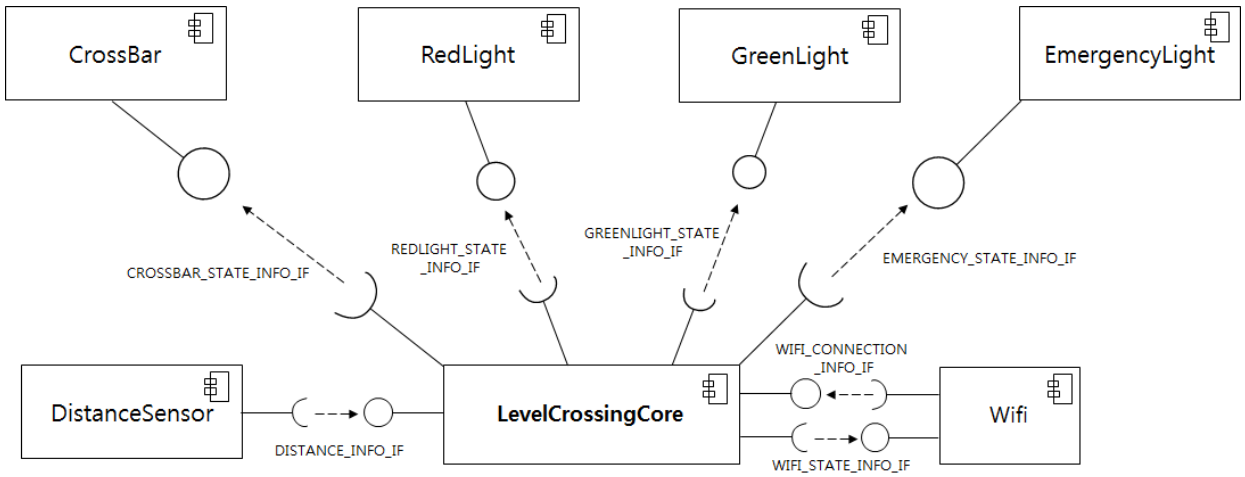
표 281 식별된 건널목 소프트웨어 컴포넌트

컴포넌트	기 능	설 명
DistanceSensor	적외선 거리 계산	감지된 적외선의 거리 계산
LevelCrossingCore	LC 핵심	차단기의 상승과 하강, 빨간등의 점등과 소등, 초록등의 점등과 소등, 비상등의 점등과 소등, 서버와의 연결 여부를 결정
CrossBar	차단기 제어	차단기의 상승과 하강을 제어
RedLight	빨간등 제어	빨간등의 점등과 소등을 제어
GreenLight	초록등 제어	초록등의 점등과 소등을 제어
EmergencyLight	비상등 제어	비상등의 점등과 소등을 제어
Wifi	통신	서버와의 통신 관리

- DistanceSensor: 차량의 접근을 감지
  - 열차가 접근하는 것을 감지한다.
  - 센서가 측정한 거리 값을 LevelCrossingCore 컴포넌트로 전달한다.
- LevelCrossingCore: LC 시스템의 I/O 총괄 제어
  - Distance Sensor 컴포넌트로부터 전달받은 거리 값을 이용해 열차의 접근을 판단한다.
  - 차단기의 상승, 하강을 결정한다.
  - 빨간등의 점등, 소등을 결정한다.
  - 초록등의 점등, 소등을 결정한다.
  - 비상등의 점등, 소등을 결정한다.
  - 중앙 제어부와의 연결 여부를 결정한다.
- CrossBar: 차단기의 상승 및 하강 제어
- RedLight: 빨간등의 점등 및 소등 제어
  - 빨간등을 점등, 소등한다.
  - LevelCrossingCore 로부터 빨간등의 점등, 소등에 대한 명령을 전달받는다.
- GreenLight: 초록등의 점등 및 소등 제어
  - 초록등을 점등, 소등한다.
  - LevelCrossingCore 로부터 초록등의 점등, 소등에 대한 명령을 전달받는다.
- EmergencyLight: 비상등의 점등 및 소등 제어
  - 비상등을 점등, 소등한다.
  - LevelCrossingCore 로부터 비상등의 점등, 소등에 대한 명령을 전달받는다.
- Wifi: 중앙 제어부와의 연결 관리
  - 중앙 제어부로 HeartBeat을 전달한다.
  - 중앙 제어부로부터 HeartBeat을 전달받는다.
  - 중앙 제어부와의 연결을 시작한다.

○ 차단기 시스템 소프트웨어의 컴포넌트간의 인터페이스 식별

그림 194 건널목 시스템의 소프트웨어 컴포넌트 및 인터페이스

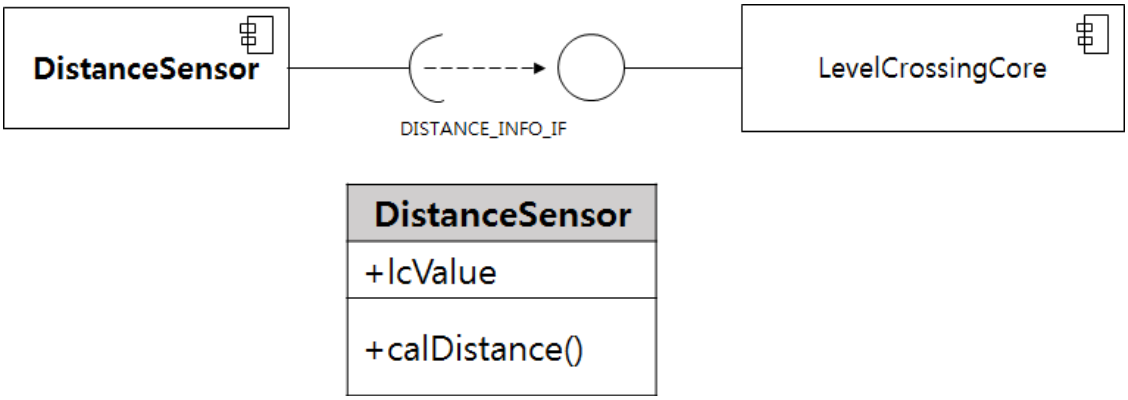


2.6. 소프트웨어 컴포넌트 설계 명세 가이드 적용

○ DistanceSensor 컴포넌트

- ① LevelCrossingCore에 거리 측정 값 전달

그림 195 DistanceSensor 컴포넌트 구성과 클래스



- lcValue : 측정한 거리 값 저장
- calDistance() : 적외선 센서가 거리를 측정한다.

○ LevelCrossingCore 컴포넌트

- [그림 120의 LevelCrossingCore 컴포넌트 데이터 흐름] 참조한다.

① DistanceSensor로부터 측정한 거리 값을 전달 받음

② CrossBar에게 차단기 상승 명령 전달

③ CrossBar에게 차단기 하강 명령 전달

④ RedLight에게 빨간등 점등 명령 전달

⑤ RedLight에게 빨간등 소등 명령 전달

⑥ GreenLight에게 초록등 점등 명령 전달

⑦ GreenLight에게 초록등 소등 명령 전달

⑧ EmergencyLight에게 비상등 점등 명령 전달

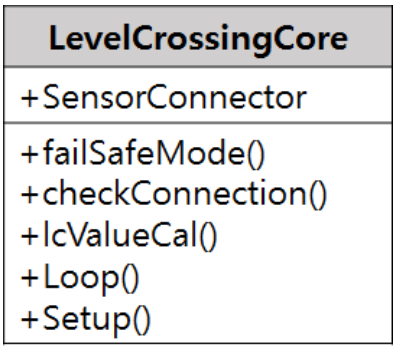
⑨ EmergencyLight에게 비상등 소등 명령 전달

⑩ 중앙 제어부에 Heartbeat 송신

⑪ Wifi에게 중앙 제어부와의 연결 시도 명령 전달

⑫ Wifi에게 중앙 제어부와의 연결 여부를 수신

그림 196 LevelCrossingCore 클래스



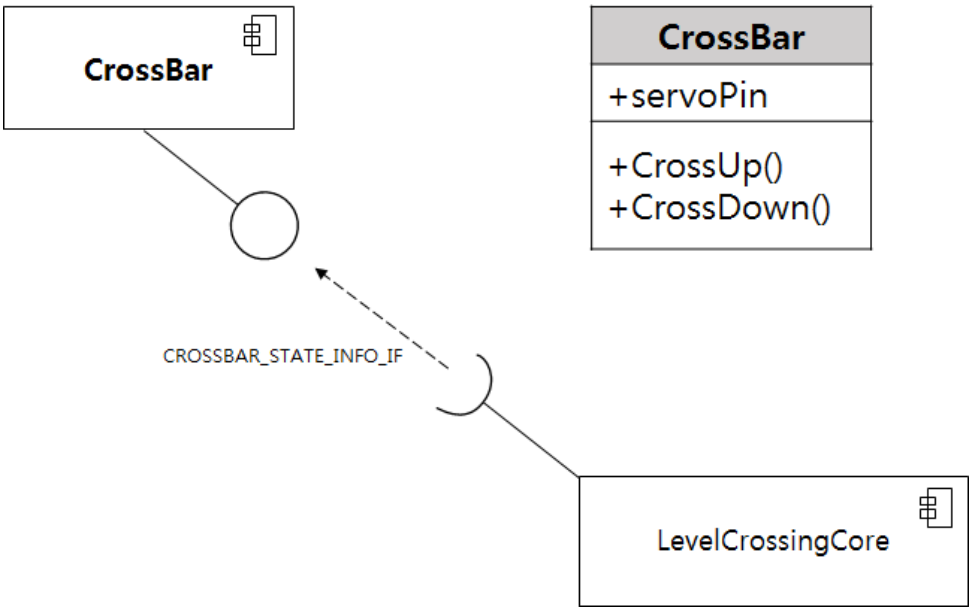
- SensorConnection : 센서의 정상 연결 여부에 대한 값을 가진다.
- Loop() : LC 소프트웨어가 수행해야 할 기능을 관리한다.
- Setup() : LC 소프트웨어가 시작 시 초기 설정을 한다.
- failSafeMode() : Emergency 모드를 작동한다.
- checkConnection() : 센서의 정상 연결 여부를 확인한다.
- lcValueCal() : DistanceSensor로부터 전달 받은 센서 값을 이용해 실제 거리를 계산한다.

○ CrossBar 컴포넌트

① LevelCrossingCore로부터 차단기 상승 명령 받음

② LevelCrossingCore로부터 차단기 하강 명령 받음

그림 197 CrossBar 컴포넌트 구성과 클래스



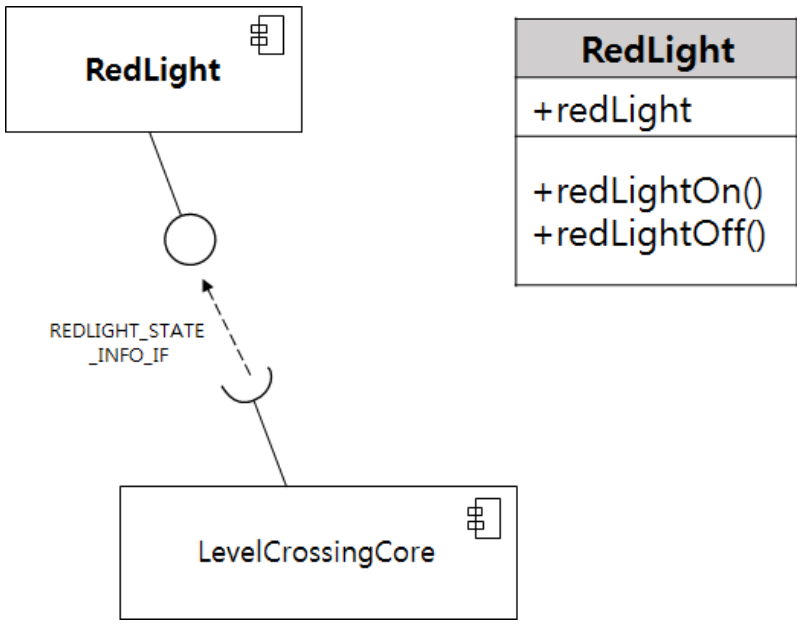
- servoPin : 차단기를 작동하는 서보모터의 핀 번호 저장
- CrossUp() : 차단기를 상승 시킨다.
- CrossDown() : 차단기를 하강 시킨다.

○ RedLight 컴포넌트

① LevelCrossingCore로부터 빨간등 점등 명령 받음

② LevelCrossingCore로부터 빨간등 소등 명령 받음

그림 198 RedLight 컴포넌트 구성 및 클래스



- redLight : 빨간등이 연결된 핀 번호를 저장
- redLightOn() : 빨간등을 점등한다.
- redLightOff() : 빨간등을 소등한다.

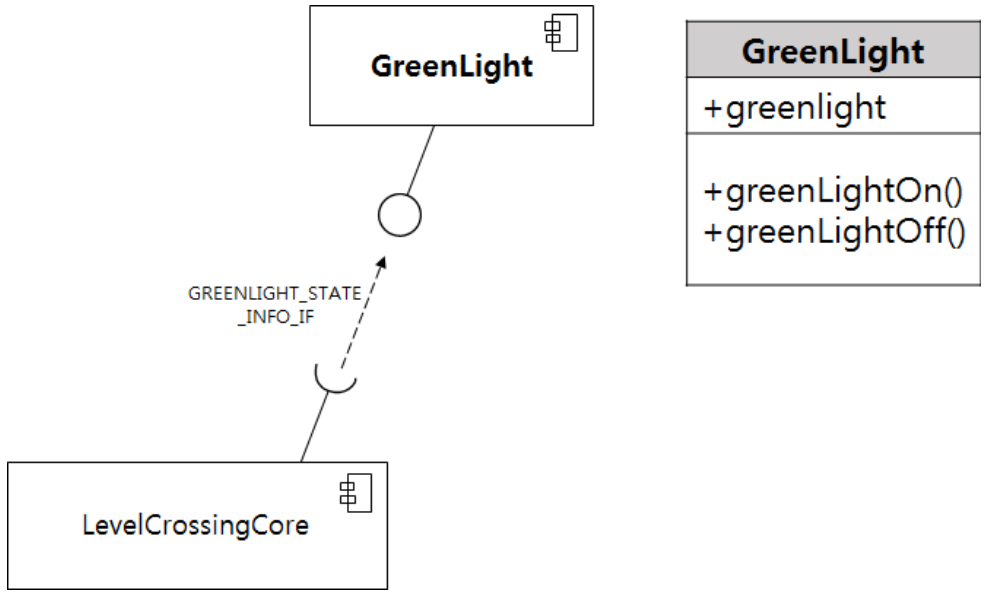
○ GreenLight 컴포넌트

① LevelCrossingCore로부터 초록등 점등 명령 받음

② LevelCrossingCore로부터 초록등 소등 명령 받음

- greenLight : 빨간등이 연결된 핀 번호를 저장
- greenLightOn() : 빨간등을 점등한다.
- greenLightOff() : 빨간등을 소등한다.

그림 199 GreenLight 컴포넌트 구성 및 클래스



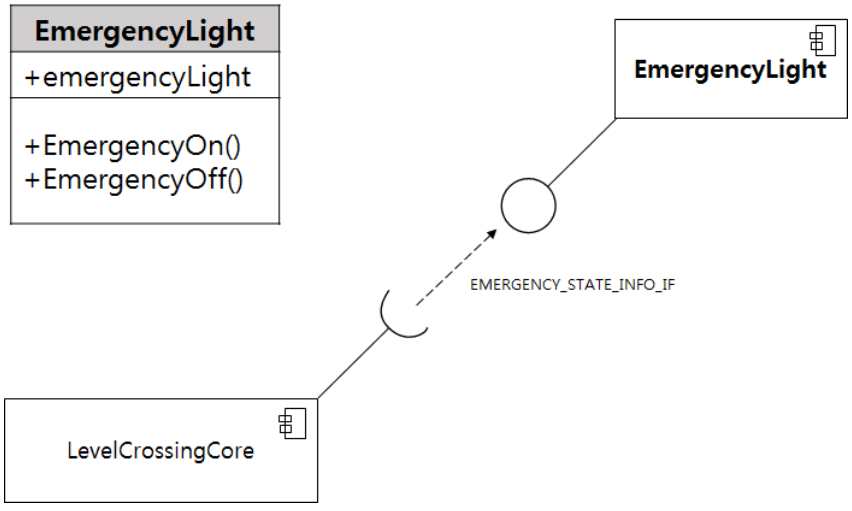
○ EmergencyLight 컴포넌트

① LevelCrossingCore로부터 비상등 점등 명령 받음

② LevelCrossingCore로부터 비상등 소등 명령 받음

- emergencyLight : 비상등이 연결된 핀 번호를 저장
- emergencyLightOn() : 비상등을 점등한다.
- emergencyLightOff() : 비상등을 소등한다.

그림 200 EmergencyLight 컴포넌트 구성 및 클래스



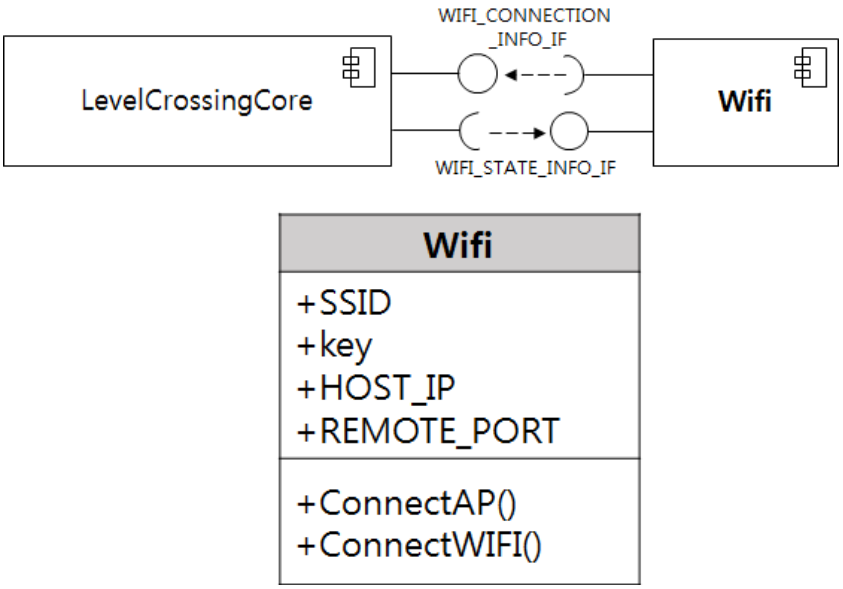
○ Wifi 컴포넌트

① LevelCrossingCore로부터 중앙 제어부에 HeartBeat 전송 명령 받음

② LevelCrossingCore로부터 중앙 제어부와의 연결 시도 명령 받음

③ LevelCrossingCore에게 중앙 제어부와의 연결 여부를 전달함

그림 201 Wifi 컴포넌트 구성 및 클래스



- SSID : 연결하고자 하는 네트워크의 SSID
- key : 연결하고자 하는 네트워크의 보안키
- HOST_IP : 연결하고자 하는 중앙 제어부의 IP
- REMOTE_PORT : 연결하고자 하는 중앙 제어부의 포트
- connectAP() : AP에 연결한다.
- connectWifi() : Wifi에 연결한다.

3. 열차 ACC 시스템

○ 모형 철도를 활용한 열차 ACC(Adaptive Cruise Control) 시스템 개발

○ ACC 시스템이 앞 차와의 거리를 감지하여 자동으로 열차 속도 조절

○ 본 시범 적용 시에는 안전성 분석을 위해 시스템 안전 무결성 등급(SIL)을 2 등급으로 적용

3.1. 시스템 개요

○ 시스템 기능

표 282 열차 ACC 시스템 기능

항 목	시스템 기능	기능 상세 설명
1	모형 열차 제어	속도 제어 명령을 수행하는 기능
2	아두이노 LED 제어	속도가 변경될 시 마다 점등 및 소등하는 기능
3	모형 화물칸 제어	아두이노 부품을 지탱하는 기능
4	아두이노 배터리 팩 제어	아두이노 CPU에 전력을 공급하는 기능
5	아두이노 CPU 제어	앞 차와의 거리에 따라 속도를 결정하는 기능
6	아두이노 모터 제어기 제어	결정된 속도값을 모형 열차가 수행하도록 제어하는 기능
7	아두이노 적외선 센서 제어	앞 차와의 거리를 감지하는 기능

○ 시스템 구성

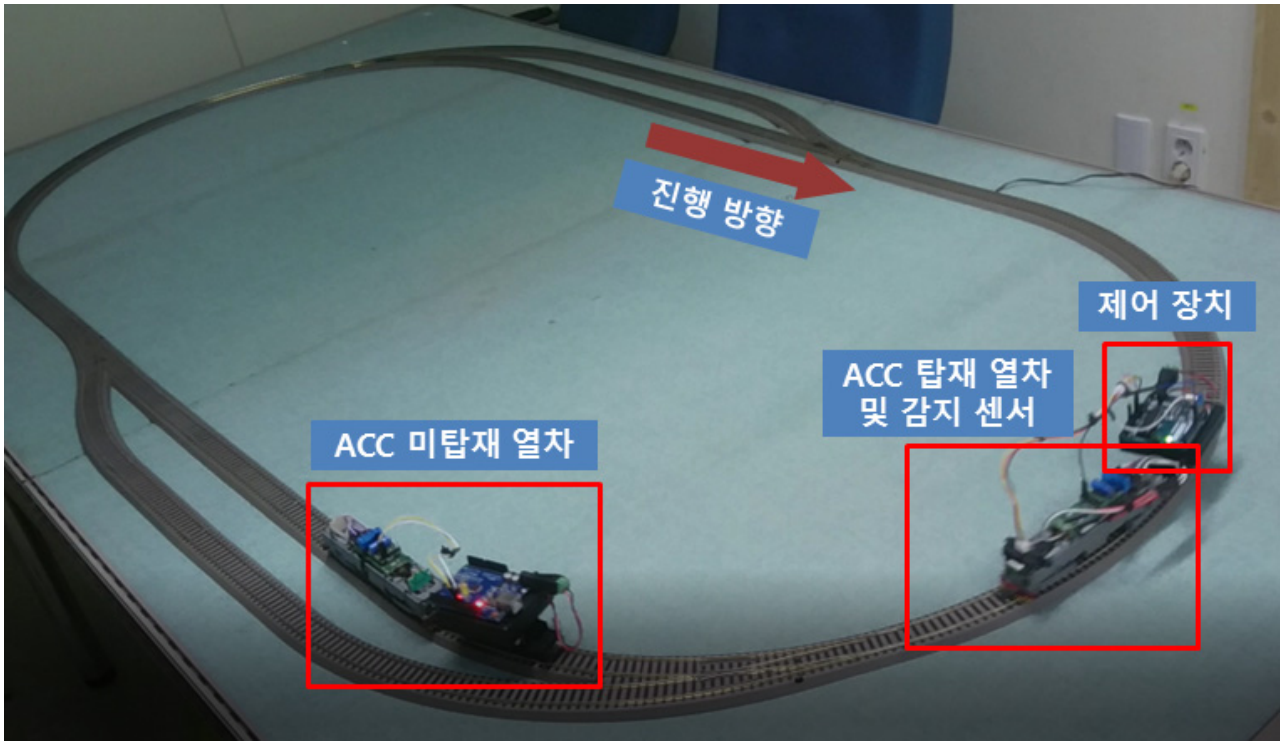


그림 202 열차 ACC 시스템 구성도

3.2. 시범 적용 가이드

○ 시스템 안전성 분석 가이드

○ 소프트웨어 요구사항 명세 가이드

○ 소프트웨어 아키텍처 및 설계 가이드

○ 소프트웨어 컴포넌트 설계 명세 가이드

3.3. 시스템 안전성 분석 가이드 적용

○ ACC(TA)의 고장 유형(Failure Mode) 및 기능 고장(Function Failure) 식별

- ACC는 Controlled Process, Controller, Actuator, Sensor의 서브시스템 구성
- 각 서브시스템에 대한 고장 유형 식별
- 식별된 서브시스템에 대한 기능 고장 식별

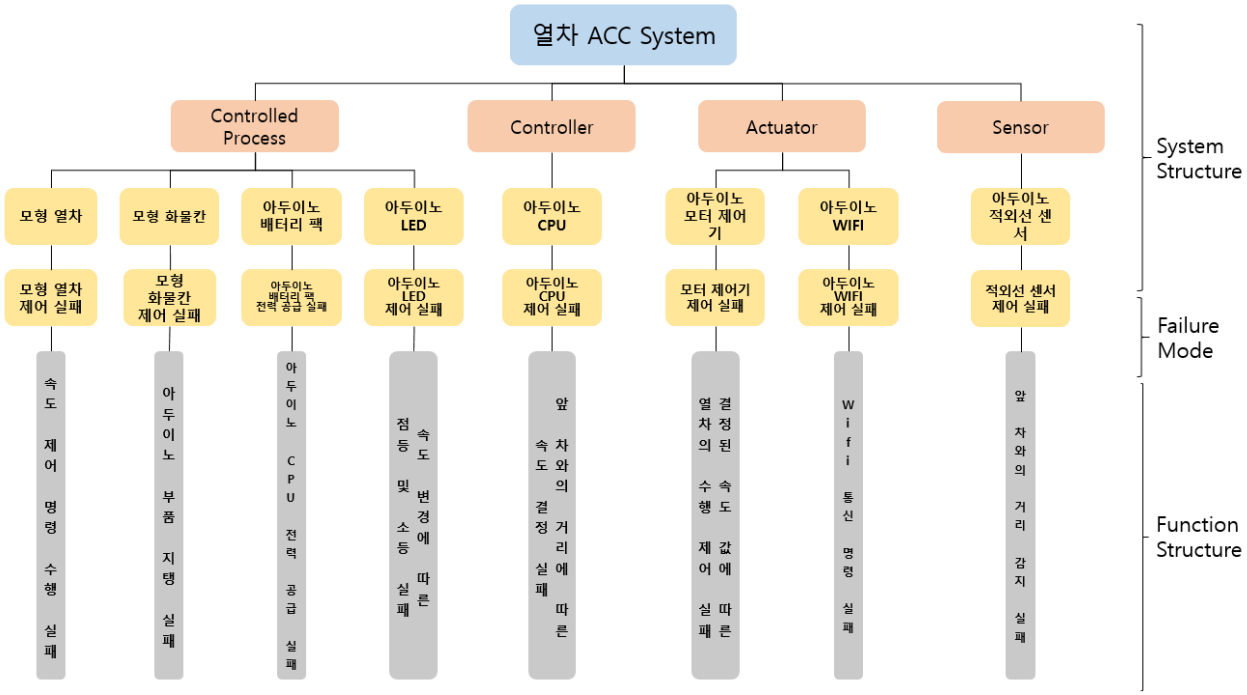


그림 203 ACC 시스템 고장 유형 및 기능 고장

○ PHA 수행 및 산출물

- 식별된 고장 유형 및 기능 고장을 기반으로 PHA 수행
- 각 기능 고장을 일으키는 고장 원인을 식별
- 각 기능 고장으로 인한 영향을 식별

고장 유형	기능 고장		고장 원인	영향
모형 열차 제어 실패	속도 제어 명령 수행 실패		1. 모형 열차 파손 2. 잘못된 속도 제어 명령 전달 3. 속도 제어 명령 미전달	열차 추돌
아두이노 LED 제어 식패	속도가 변경될 시 마다 점등 및 소등 실패		1. 아두이노 LED 미작동 2, 아두이노 LED 파손	열차 상태 정보 숙지 불가
모형 화물칸 제어 실패	아두이노 부품 지령 실패		1. 모형 화물칸 파손	열차 추돌
아두이노 배터리 팩 제어 실패	아두이노 CPU에 전력 공급 실패		1. 잔여 배터리 부족	열차 추돌
아두이노 CPU 제어 실패	앞 차와의 거리에 따른 속도 결정 실패	Distance < 12 but Speed != 0	1. 아두이노 CPU에 전력 미공급 2, 아두이노 CPU에 과전력 공급 3. 아두이노 CPU 파손 4, 정해진 범위 이외의 거리 값 전달 5. 거리 값 미전달	열차 추돌
		Distance >= 12 but Speed = 0		열차 추돌
		Distance < 18 but Speed != 80		열차 추돌
		Distance >= 18 but Speed = 80		열차 추돌
		Distance < 24 but Speed != 100		열차 추돌
		Distance >= 24 but Speed = 100		열차 추돌
		Distance < 30 but Speed != 120		열차 추돌
		Distance >= 30 but Speed = 120		열차 추돌
		Distance >= 30 but Speed != 130		열차 추돌
		Distance < 30 but Speed = 130		열차 추돌
아두이노 모터 제어기 제어 실패	결정된 속도 값을 수행하도록 제어 실패		1. 잘돈 결정된 속도 전달 2, 속도 미결정	열차 추돌
아두이노 적외선 센서 제어 실패	앞 차와의 거리 감지 실패		1. 적외선 센서 미작동 2, 적외선 센서 파손 3. 커브 구간에서의 감지 불가	열차 추돌

그림 204 PHA 수행 결과 표

○ FMEA 수행 및 산출물

- PHA 수행 산출물을 기반으로 FMEA 수행
- 각 기능 고장 별로 위험도 평가 수행
- 수행된 위험도 평가를 기반으로 예방 조치 및 설계 반영 방안 도출
- 예방 조치 및 설계 반영 방안에 대한 안전 기능 정의

시스템 명칭	시스템 구조		기능		고장 유형 (Failure Mode)	고장 영향 (Failure Effects)	RPN 평가			RPN	고장 원인 (Failure Cause)	예방 조치 및 설계 반영 방안	안전 기능
	Layer-1	Layer-2					심각도	발생 빈도	허용 수준				
TA	Controlled Process	모형 열차	속도 제어 명령 수행 기능		모형 열차 제어 실패	속도 제어 명령 실패	5	3	4	60	1. 모형 열차 파손 2. 잘못된 속도 제어 명령 전달 3. 속도 제어 명령 미전달	서버를 활용한 시스템 및 제어의 이중화	1. 열차가 현재 위치 정보 값을 저장하는 기능 2. 서버가 열차들의 현재 위치 정보 값을 저장하는 기능 3. 서버가 각 열차에게 앞 차와의 위치 차이 값을 전달하는 기능 4. 적외선 센서가 감지한 앞 차와의 거리 값과 서버가 보낸 앞 차와의 위치 차이 값을 이용한 다음 속도 결정 기능 5. 통신 장애를 대비한 Heartbeat와 비상 정지 기능
		아두이노 LED	속도가 변경될 시 마다 점등 및 소등하는 기능		아두이노 LED 제어 실패	속도가 변경될 시 마다 점등 및 소등 실패	1	3	2	6	1. 아두이노 LED 미작동 2. 아두이노 LED 파손		
		모형 화물칸	아두이노 부품 지탱 기능		모형 화물칸 제어 실패	아두이노 부품 지탱 실패	5	3	4	60	1. 모형 화물칸 파손		
		아두이노 배터리 팩	아두이노 CPU에 전력 공급 기능		아두이노 배터리 팩 제어 실패	아두이노 CPU에 전력 공급 실패	5	3	4	60	1. 잔여 배터리 부족		
	Controller	아두이노 CPU	앞 차와의 거리에 따라 속도 결정 기능	앞 차와의 거리가 12 미만이면 속도를 0으로 결정하는 기능	아두이노 CPU 제어 실패	Distance < 12 but Speed != 0	5	3	4	60	1. 아두이노 CPU에 전력 미공급 2. 아두이노 CPU에 과전력 공급 3. 아두이노 CPU 파손 4. 정해진 범위 이외의 거리 값 전달 5. 거리 값 미전달		
						Distance >= 12 but Speed = 0	5	3	4	60			
				앞 차와의 거리가 18 미만이면 속도는 80으로 결정하는 기능		Distance < 18 but Speed != 80	5	3	4	60			
						Distance >= 18 but Speed = 80	5	3	4	60			
				앞 차와의 거리가 24 미만이면 속도는 100으로 결정하는 기능		Distance < 24 but Speed != 100	5	3	4	60			
						Distance >= 24 but Speed = 100	5	3	4	60			
				앞 차와의 거리가 30 미만이면 속도는 120으로 결정하는 기능		Distance < 30 but Speed != 120	5	3	4	60			
						Distance >= 30 but Speed = 120	5	3	4	60			
				앞 차와의 거리가 30 이상이면 속도는 130으로 결정하는 기능		Distance >= 30 but Speed != 130	5	3	4	60			
						Distance < 30 but Speed = 130	5	3	4	60			
	Actuator	아두이노 모터 제어기	결정된 속도 값을 모형 열차가 수행하도록 제어하는 기능		아두이노 모터 제어기 제어 실패	결정된 속도 값을 수행하도록 제어 실패	5	3	4	60	1. 잘못 결정된 속도 전달 2. 속도 미결정		
	Sensor	아두이노 적외선 센서	앞 차와의 거리를 감지하는 기능		아두이노 적외선 센서 제어 실패	앞 차와의 거리 감지 실패	5	3	4	60	1. 적외선 센서 미작동 2. 적외선 센서 파손 3. 커브 구간에서의 감지 불가		

그림 205 FMEA 수행 결과 표

○ 시스템 및 안전 요구사항 도출

안전성 분석 결과와 시스템 기능에 대한 자료를 바탕으로 시스템 요구사항과 시스템 안전 요구사항 도출하였다.

표 283 시스템 요구사항

요구사항 ID	요구사항
시스템 요구사항 (System Requirement)	
모형 열차 제어	
SRS_TA_REQ1.1	모형 열차는 속도 제어 명령을 수행할 수 있어야 한다.
아두이노 LED 제어	
SRS_TA_REQ2.1	아두이노 LED는 속도가 변경될 시 마다 점등 및 소등하여 속도의 변경을 알릴 수 있어야 한다.
모형 화물칸 제어	
SRS_TA_REQ3.1	모형 화물칸은 아두이노 부품을 지탱할 수 있어야 한다.
아두이노 배터리 팩 제어	
SRS_TA_REQ4,1	아두이노 배터리 팩은 아두이노 Uno 보드에 전력을 공급할 수 있어야 한다.
아두이노 CPU 제어	
SRS_TA_REQ5.1	아두이노 CPU는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.
SRS_TA_REQ5.1.1	아두이노 CPU는 앞 차와의 거리가 12 미만이면 속도를 0으로 결정할 수 있어야 한다.
SRS_TA_REQ5.1.2	아두이노 CPU는 앞 차와의 거리가 18 미만이면 속도를 80으로 결정할 수 있어야 한다.
SRS_TA_REQ5.1.3	아두이노 CPU는 앞 차와의 거리가 25 미만이면 속도를 100으로 결정할 수 있어야 한다.
SRS_TA_REQ5.1.4	아두이노 CPU는 앞 차와의 거리가 30 미만이면 속도를 120으로 결정할 수 있어야 한다.
SRS_TA_REQ5.1.5	아두이노 CPU는 앞 차와의 거리가 30 이상이면 속도를 130으로 결정할 수 있어야 한다.
아두이노 모터 제어기 제어	
SRS_TA_REQ6.1	아두이노 모터 제어기는 결정된 속도값을 모형 열차가 수행하도록 제어할 수 있어야 한다.
아두이노 적외선 센서 제어	
SRS_TA_REQ7.1	아두이노 적외선 센서는 앞 차와의 거리를 감지할 수 있어야 한다.
인터페이스 (Interface Requirement)	
SRS_TA_REQ8.1	아두이노 CPU는 속도가 변경될 시 마다 아두이노 LED를 작동시켜 속도의 변경을 알릴 수 있어야 한다.
SRS_TA_REQ8.2	아두이노 CPU는 결정된 속도값을 모형 열차가 수행하도록 아두이노 모터 제어기에게 제어 명령을 내릴 수 있어야 한다.
성능 (Performance Requirement)	
SRS_TA_REQ9.1	아두이노 LED의 점등 및 소등은 0.1초 이내에 이루어져야 한다.

표 284 시스템 안전 요구사항

요구사항 ID	요구사항
시스템 안전 요구사항 (System Safety Requirement)	
서버를 활용한 시스템 및 제어의 이중화	
SSRS_TA_REQ1.1	RFID를 이용하여 각 열차는 현재 위치 정보 값을 저장한다.
SSRS_TA_REQ1.2	서버와 Wifi 통신하여 열차들의 현재 위치 정보 값을 저장한다.
SSRS_TA_REQ1.3	서버는 Wifi 통신하여 각 열차에게 앞 차와의 위치 차이 값을 전달한다.
SSRS_TA_REQ1.4	적외선 센서가 감지한 앞 차와의 거리 값과 서버가 보낸 앞 차와의 거리 차이 값을 이용하여 다음 속도를 결정한다.
SSRS_TA_REQ1.5	통신 장애를 대비한 Heartbeat 및 비상 정지 기능을 추가한다.

3.4. 소프트웨어 요구사항 명세 가이드 적용

앞 단계에서 식별된 시스템 및 안전 요구사항을 기반으로 소프트웨어 요구사항과 안전 요구사항을 도출하고, 상세 요구사항을 명세하였다.

표 285 소프트웨어 요구사항

요구사항 ID (관련 사항)	요구사항
소프트웨어 요구사항 (Software Requirements)	
아두이노 CPU 내장 소프트웨어	
소프트웨어RS_TA_REQ1.1 (관련 사항: SRS_TA_REQ5.1)	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.
소프트웨어RS_TA_REQ1.1.1 (관련 사항: SRS_TA_REQ5.1.1)	TA 소프트웨어는 앞 차와의 거리가 12 미만이면 속도를 0으로 결정할 수 있어야 한다.
소프트웨어RS_TA_REQ1.1.2 (관련 사항: SRS_TA_REQ5.1.2)	TA 소프트웨어는 앞 차와의 거리가 18 미만이면 속도를 80으로 결정할 수 있어야 한다.
소프트웨어RS_TA_REQ1.1.3 (관련 사항: SRS_TA_REQ5.1.3)	TA 소프트웨어는 앞 차와의 거리가 24 미만이면 속도를 100으로 결정할 수 있어야 한다.
소프트웨어RS_TA_REQ1.1.4 (관련 사항: SRS_TA_REQ5.1.4)	TA 소프트웨어는 앞 차와의 거리가 30 미만이면 속도를 120으로 결정할 수 있어야 한다.
소프트웨어RS_TA_REQ1.1.5 (관련 사항: SRS_TA_REQ5.1.5)	TA 소프트웨어는 앞 차와의 거리가 30 이상이면 속도를 130으로 결정할 수 있어야 한다.
인터페이스 (Interface Requirement)	
소프트웨어RS_TA_REQ2.1 (관련 사항: SRS_TA_REQ8.1)	TA 소프트웨어는 속도가 변경될 시 마다 아두이노 LED를 점등 및 소등시켜 속도의 변경을 알릴 수 있도록 제어 명령을 내릴 수 있어야 한다.
소프트웨어RS_TA_REQ2.2 (관련 사항: SRS_TA_REQ8.2)	TA 소프트웨어는 결정된 속도값을 모형 열차가 수행하도록 모터 제어기에게 제어 명령을 내릴 수 있어야 한다.
성능 (Performance Requirement)	
소프트웨어RS_TA_REQ3.1 (관련 사항: SRS_TA_REQ9.1)	TA 소프트웨어는 아두이노 LED의 점등 및 소등을 0.1초 이내에 이루어지도록 제어 명령을 내릴 수 있어야 한다.

표 286 소프트웨어 안전 요구사항

요구사항 ID (관련 사항)	요구사항
소프트웨어 안전 요구사항 (Software Safety Requirement)	
소프트웨어SRS_TA_REQ1.1 (관련 사항: SSRS_TA_REQ1.2, SSRS_TA_REQ1.3, SSRS_TA_REQ1.5)	TA 소프트웨어는 서버와 Wifi 통신할 수 있어야 한다.
소프트웨어SRS_TA_REQ1.2 (관련 사항: SSRS_TA_REQ1.1, SSRS_TA_REQ1.2)	TA 소프트웨어는 현재 위치를 서버로 전송해야 한다.
소프트웨어SRS_TA_REQ1.3 (관련 사항: SSRS_TA_REQ1.2, SSRS_TA_REQ1.3)	TA 소프트웨어는 앞 차와의 위치 차이 값을 서버로부터 전송받아야 한다.
소프트웨어SRS_TA_REQ1.4 (관련 사항: SSRS_TA_REQ1.3, SSRS_TA_REQ1.4)	TA 소프트웨어는 다음 속도 결정 시 적외선 센서가 감지한 앞 차와의 거리 값 외에도 서버가 보낸 앞 차와의 위치 차이 값을 고려한다.
소프트웨어SRS_TA_REQ1.4.1 (관련 사항: SSRS_TA_REQ1.3, SSRS_TA_REQ1.4)	TA 소프트웨어는 앞 차와의 위치 값 차이가 2 미만이면 다음 속도 0으로 결정할 수 있어야 한다.
소프트웨어SRS_TA_REQ1.4,2 (관련 사항: SSRS_TA_REQ1.3, SSRS_TA_REQ1.4)	TA 소프트웨어는 앞 차와의 위치 값 차이가 3 미만이면 다음 속도 80으로 결정할 수 있어야 한다.
소프트웨어SRS_TA_REQ1.4.3 (관련 사항: SSRS_TA_REQ1.3, SSRS_TA_REQ1.4)	TA 소프트웨어는 앞 차와의 위치 값 차이가 3 이상이면 다음 속도 120으로 결정할 수 있어야 한다.
소프트웨어SRS_TA_REQ1.5 (관련 사항: SSRS_TA_REQ1.5)	TA 소프트웨어는 서버와 Heartbeat를 주고받을 수 있어야 한다.
소프트웨어SRS_TA_REQ1.6 (관련 사항: SSRS_TA_REQ1.5)	TA 소프트웨어는 통신 장애를 대비하여 비상 정지 기능을 수행할 수 있어야 한다.

○ 상세 기능 요구사항

표 287 기능 요구사항 (속도 결정)

Use Case Name	속도 결정		ID	소프트웨어RS_TA_REQ1.1
요구사항	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.			
Actor	열차			
Trigger	SRS_TA_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 앞 차와의 거리를 측정한 아날로그 값을 전달받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	변환한 값이 12 미만일 경우 속도를 0으로 결정한다.		
	4	Use Case 종료		
Pre-conditions	1. 앞 차와의 거리를 측정한 아날로그 값			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어RS_TA_REQ2.2			
상세 요구사항	소프트웨어RS_TA_REQ1.1.1			

표 288 기능 요구사항 (속도 결정)

Use Case Name	속도 결정		ID	소프트웨어RS_TA_REQ1.1
요구사항	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.			
Actor	열차			
Trigger	SRS_TA_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 앞 차와의 거리를 측정한 아날로그 값을 전달받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	변환한 값이 18 미만일 경우 속도를 80으로 결정한다.		
	4	Use Case 종료		
Pre-conditions	1. 앞 차와의 거리를 측정한 아날로그 값			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어RS_TA_REQ2.2			
상세 요구사항	소프트웨어RS_TA_REQ1.1.2			

표 289 기능 요구사항 (속도 결정)

Use Case Name	속도 결정		ID	소프트웨어RS_TA_REQ1.1
요구사항	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.			
Actor	열차			
Trigger	SRS_TA_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 앞 차와의 거리를 측정한 아날로그 값을 전달 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	변환한 값이 24 미만일 경우 속도를 100으로 결정한다.		
	4	Use Case 종료		
Pre-conditions	1. 앞 차와의 거리를 측정한 아날로그 값			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어RS_TA_REQ2.2			
상세 요구사항	소프트웨어RS_TA_REQ1.1.3			

표 290 기능 요구사항 (속도 결정)

Use Case Name	속도 결정		ID	소프트웨어RS_TA_REQ1.1
요구사항	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.			
Actor	열차			
Trigger	SRS_TA_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 앞 차와의 거리를 측정한 아날로그 값을 전달 받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	변환한 값이 30 미만일 경우 속도를 120으로 결정한다.		
	4	Use Case 종료		
Pre-conditions	1. 앞 차와의 거리를 측정한 아날로그 값			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어RS_TA_REQ2.2			
상세 요구사항	소프트웨어RS_TA_REQ1.1.4			

표 291 기능 요구사항 (속도 결정)

Use Case Name	속도 결정		ID	소프트웨어RS_TA_REQ1.1
요구사항	TA 소프트웨어는 앞 차와의 거리에 따라 속도를 결정할 수 있어야 한다.			
Actor	열차			
Trigger	SRS_TA_REQ7.1			
Flow of Events				
Basic Flow	Step	Action		
	1	아두이노 적외선 센서로부터 앞 차와의 거리를 측정한 아날로그 값을 전달받는다.		
	2	아날로그 값을 디지털 값으로 변환한다.		
	3	변환한 값이 30 이상일 경우 속도를 130으로 결정한다.		
	4	Use Case 종료		
Pre-conditions	1. 앞 차와의 거리를 측정한 아날로그 값			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어RS_TA_REQ2.2			
상세 요구사항	소프트웨어RS_TA_REQ1.1.5			

표 292 인터페이스 요구사항 (LED 작동)

Use Case Name	LED 작동		ID	소프트웨어RS_TA_REQ2.1
요구사항	TA 소프트웨어는 속도가 변경될 시 마다 아두이노 LED를 점등 및 소등시켜 속도의 변경을 알릴 수 있도록 제어할 수 있어야 한다.			
Actor	아두이노 LED			
Trigger	소프트웨어RS_TA_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	측정된 값에 의하여 속도 값이 결정된다.		
	2	속도값을 이전 속도 값과 비교한다.		
	3	속도 값의 변경을 확인한다.		
	4	아두이노 LED 점등 및 소등 제어 명령을 전달한다.		
	5	Use Case 종료		
Pre-conditions	1. 결정된 속도			
Post-conditions	1. 이전 속도 값			
관련 요구사항	해당사항 없음			
상세 요구사항	해당사항 없음			

표 293 인터페이스 요구사항 (모터 제어기 제어)

Use Case Name	모터 제어기 제어		ID	소프트웨어RS_TA_REQ2.2
요구사항	TA 소프트웨어는 결정된 속도값을 모형 열차가 수행하도록 모터 제어기에게 제어 명령을 내릴 수 있어야 한다.			
Actor	모터 제어기			
Trigger	소프트웨어RS_TA_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	측정된 값에 의하여 속도 값이 결정된다.		
	2	속도값을 이전 속도 값과 비교한다.		
	3	속도 값의 변경을 확인한다.		
	4	모터 제어기에 변경된 속도 값으로의 제어 명령을 전달한다.		
	5	Use Case 종료		
Pre-conditions	1. 결정된 속도			
Post-conditions	1. 이전 속도 값			
관련 요구사항	해당사항 없음			
상세 요구사항	해당사항 없음			

표 294 비기능 요구사항

비기능 요구사항	
안전성 (Safety)	소프트웨어 안전 무결성 등급 (SIL)이 2 등급을 만족해야 한다.
제약사항 (Constraints)	TA 소프트웨어는 아두이노 LED의 점등 및 소등을 0.1초 이내에 이루어지도록 제어 명령을 내릴 수 있어야 한다.

○ 상세 안전 요구사항

표 295 안전 요구사항 (Wifi 통신)

Use Case Name	Wifi 통신		ID	소프트웨어SRS_TA_REQ1.1
안전 요구사항	TA 소프트웨어는 서버와 Wifi 통신할 수 있어야 한다.			
Actor	TA 소프트웨어			
Tigger	해당사항 없음			
Flow of Events				
Basic Flow	Step	Action		
	1	AP에 접속한다.		
	2	서버와 Wifi 통신을 연결한다.		
	3	Wifi 통신 명령을 대기한다.		
	4	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	해당사항 없음			
Post-conditions	Heartbeat			
관련 요구사항	소프트웨어SRS_TA_REQ1.5, 소프트웨어SRS_TA_REQ1.6			
상세 요구사항	해당사항 없음			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 296 안전 요구사항 (위치 정보 저장)

Use Case Name	위치 정보 저장		ID	소프트웨어SRS_TA_REQ1.2
안전 요구사항	TA 소프트웨어는 현재 위치를 서버로 전송하여야 한다.			
Actor	TA 소프트웨어			
Trigger	SSRS_TA_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	RFID로부터 RFID 태그 값을 전달받는다.		
	2	RFID 태그 값을 이전 RFID 태그 값과 비교한다.		
	3	RFID 태그 값의 변경을 확인한다.		
	4	서버로 변경된 RFID 태그 값을 전달한다.		
	5	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - RFID로부터 RFID 태그 값을 전달받지 못할 경우		
	1	RFID 태그 값의 변경을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Alternative Flow 2	Step	2a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. RFID로부터 전달된 RFID 태그 값			
Post-conditions	1. 이전 RFID 태그 값			
관련 요구사항	소프트웨어SRS_TA_REQ1.3, 소프트웨어SRS_TA_REQ1.4			
상세 요구사항	해당사항 없음			
안전 조건	일정 시간 이내에 RFID 태그 값의 변경이 확인되지 않을 경우 RFID에 문제가 있는 것으로 판단한다.			
	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 297 안전 요구사항 (위치 차이 값 수신

Use Case Name	위치 차이 값 수신		ID	소프트웨어SRS_TA_REQ1.3
안전 요구사항	TA 소프트웨어는 앞 차와의 위치 차이 값을 서버로부터 전송받아야 한다.			
Actor	TA 소프트웨어			
Trigger	SSRS_TA_REQ1.2			
Flow of Events				
Basic Flow	Step	Action		
	1	서버로 RFID 태그 값을 전달한다.		
	2	서버로부터 앞 차와의 거리 차이 값을 전송받는다.		
	3	Use Case를 종료한다.		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 각 열차로부터 전달된 RFID 태그 값			
Post-conditions	1. 위치 차이 값			
관련 요구사항	소프트웨어SRS_TA_REQ1.4			
상세 요구사항	해당사항 없음			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 298 안전 요구사항 (속도 결정 요소 추가)

Use Case Name	속도 결정 요소 추가		ID	소프트웨어SRS_LC_REQ1.4
안전 요구사항	TA 소프트웨어는 다음 속도 결정 시 적외선 센서가 감지한 앞 차와의 거리 값 외에도 서버가 보낸 앞 차와의 위치 차이 값을 고려한다.			
Actor	TA 소프트웨어			
Trigger	소프트웨어SRS_TA_REQ1.3			
Flow of Events				
Basic Flow	Step	Action		
	1	서버로부터 앞 차와의 위치 값을 전송 받는다.		
	2	서버로부터 전송 받은 위치 값 차가 2 미만인 경우 다음 속도를 0으로 결정한다.		
	3	Use Case 종료		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 앞 차와의 위치 값 차			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어SRS_TA_REQ1.3			
상세 요구사항	소프트웨어SRS_TA_REQ1.4.1			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 299 안전 요구사항 (속도 결정 요소 추가)

Use Case Name	속도 결정 요소 추가		ID	소프트웨어SRS_LC_REQ1.4
안전 요구사항	TA 소프트웨어는 다음 속도 결정 시 적외선 센서가 감지한 앞 차와의 거리 값 외에도 서버가 보낸 앞 차와의 위치 차이 값을 고려한다.			
Actor	TA 소프트웨어			
Trigger	소프트웨어SRS_TA_REQ1.3			
Flow of Events				
Basic Flow	Step	Action		
	1	서버로부터 앞 차와의 위치 값을 전송 받는다.		
	2	서버로부터 전송 받은 위치 값 차가 3 미만인 경우 다음 속도를 80으로 결정한다.		
	3	Use Case 종료		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 앞 차와의 위치 값 차			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어SRS_TA_REQ1.3			
상세 요구사항	소프트웨어SRS_TA_REQ1.4.2			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 300 안전 요구사항 (속도 결정 요소 추가)

Use Case Name	속도 결정 요소 추가		ID	소프트웨어SRS_LC_REQ1.4
안전 요구사항	TA 소프트웨어는 다음 속도 결정 시 적외선 센서가 감지한 앞 차와의 거리 값 외에도 서버가 보낸 앞 차와의 위치 차이 값을 고려한다.			
Actor	TA 소프트웨어			
Trigger	소프트웨어SRS_TA_REQ1.3			
Flow of Events				
Basic Flow	Step	Action		
	1	서버로부터 앞 차와의 위치 값을 전송 받는다.		
	2	서버로부터 전송 받은 위치 값 차가 3 이상인 경우 다음 속도를 120으로 결정한다.		
	3	Use Case 종료		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. 앞 차와의 위치 값 차			
Post-conditions	1. 결정된 속도			
관련 요구사항	소프트웨어SRS_TA_REQ1.3			
상세 요구사항	소프트웨어SRS_TA_REQ1.4.3			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 301 안전 요구사항 (Heartbeat)

Use Case Name	Heartbeat		ID	소프트웨어SRS_TA_REQ1.5
안전 요구사항	TA 소프트웨어는 서버와 Heartbeat를 주고받을 수 있어야 한다.			
Actor	TA 소프트웨어			
Tigger	소프트웨어SRS_TA_REQ1.1			
Flow of Events				
Basic Flow	Step	Action		
	1	서버로 Heartbeat를 보낸다.		
	2	서버로부터 Heartbeat를 수신 받는다.		
	3	Heartbeat 미수신 Count를 초기화 한다.		
	4	Use Case 종료		
Alternative Flow 1	Step	1a - 서버와의 통신에 장애가 발생한 경우		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. Heartbeat			
Post-conditions	1. Heatbeat 미수신 Count			
관련 요구사항	해당사항 없음			
상세 요구사항	해당사항 없음			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

표 302 안전 요구사항 (비상 정지)

Use Case Name	비상 정지		ID	소프트웨어SRS_TA_REQ1.6
안전 요구사항	TA 소프트웨어는 통신 장애를 대비하여 비상 정지 기능을 수행할 수 있어야 한다.			
Actor	TA 소프트웨어			
Trigger	소프트웨어SRS_TA_REQ1.5			
Flow of Events				
Basic Flow	Step	Action		
	1	Heartbeat의 수신을 확인한다.		
	2	비상 정지 기능을 수행한다.		
	3	Use Case를 종료한다.		
Pre-conditions	1. Heartbeat			
Post-conditions	해당사항 없음			
관련 요구사항	해당사항 없음			
상세 요구사항	해당사항 없음			
안전 조건	일정 시간 이내에 Heartbeat에 대한 답변이 수신되지 않으면 통신 장애가 있는 것으로 판단한다.			

3.5. 소프트웨어 아키텍처 및 설계 가이드 적용

○ ACC 시스템 개념도 도출

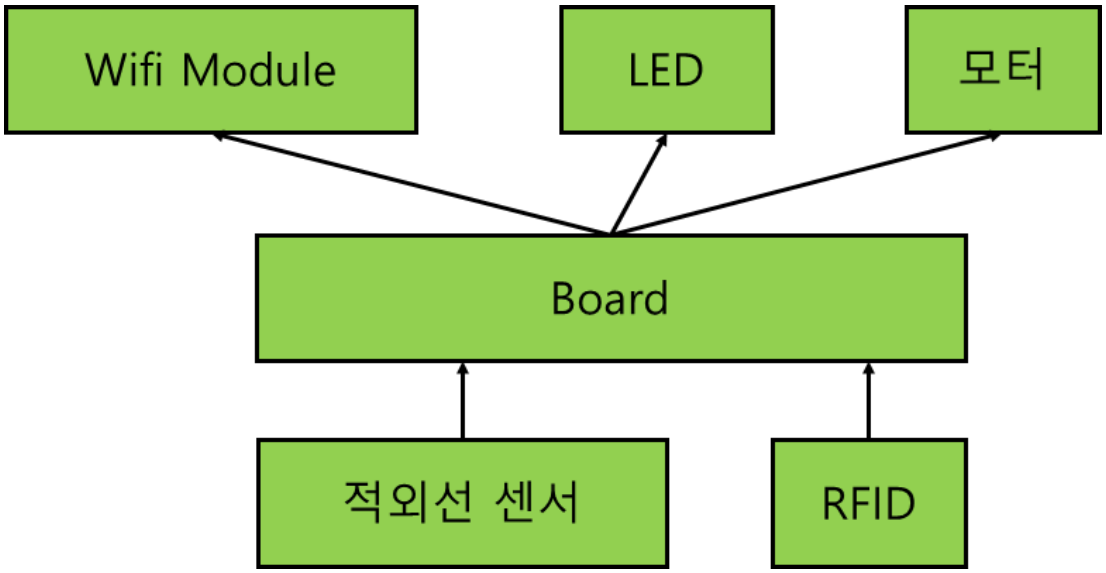


그림 206 ACC 시스템 개념도

○ ACC 소프트웨어 컴포넌트 식별

소프트웨어 요구사항 명세와 시스템 요구사항으로부터 소프트웨어 컴포넌트를 식별하였다.

표 303 식별된 ACC 소프트웨어 컴포넌트

컴포넌트	기 능	설 명
DistanceSensor	적외선 거리 계산	감지된 적외선의 거리 계산
TrainACC	TA 핵심	열차의 가속 및 감속, 정지, LED 의 점등과 소등, RFID 태그 값 인식, 서버와의 연결 여부를 결정
Motor	모터 제어	열차의 속도 제어 명령 수행
LED	LED 제어	LED의 점등과 소등을 제어
RFID	RFID 태그 인식	RFID 태그 인식을 수행
Wifi	통신	서버와의 통신 관리

- DistanceSensor: 앞 차와의 거리를 감지
  - 앞 차와의 거리를 감지한다.
  - 센서가 측정한 거리 값을 TrainACC 컴포넌트로 전달한다.
- TrainACC: ACC 시스템의 I/O 총괄 제어
  - Distance Sensor 컴포넌트로부터 전달받은 거리 값을 이용해 앞 차와의 거리를 판단한다.
  - 열차의 감속 및 가속, 정지를 결정한다.

- LED의 점등, 소등을 결정한다.
- RFID 컴포넌트로부터 전달받은 값을 이용해 현재 위치를 계산한다.
- 서버와의 연결 여부를 결정한다.

• Motor: 열차의 속도 제어
  - 변경된 열차의 속도로 제어한다.
• LED: LED의 점등 및 소등
  - LED를 점등, 소등한다.
• RFID: RFID 태그 값 인식
  - RFID 태가 값을 읽어 들인다.
• Wifi: 서버와의 통신 관리
  - 서버로 HeartBeat을 전달한다.
  - 서버로부터 HeartBeat을 전달받는다.
  - 서버와의 연결을 시작한다.

○ ACC 시스템 소프트웨어의 컴포넌트간의 인터페이스 식별

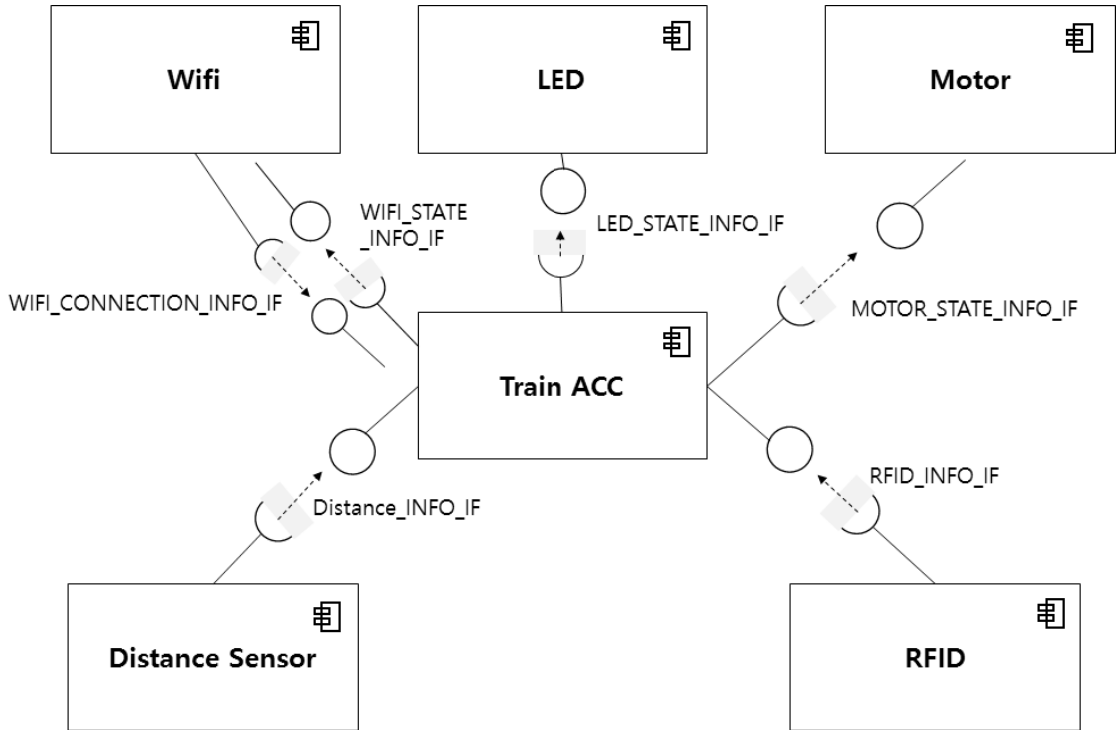


그림 207 ACC 시스템의 소프트웨어 컴포넌트 및 인터페이스

3.6. 소프트웨어 컴포넌트 설계 명세 가이드 적용

○ DistanceSensor 컴포넌트

① TrainACC에게 거리 측정 값 전달

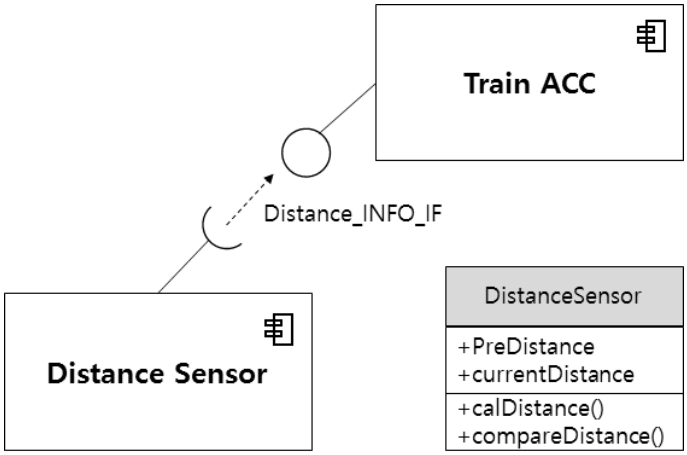


그림 208 DistanceSensor 클래스

- currnetDistance : 측정한 거리 값 저장
- preDistance : 이전 거리 값 저장
- calDistance() : 적외선 센서가 거리를 측정한다.
- compareDistance() : 거리 값이 변경되었는지 확인한다.

○ TrainACC 컴포넌트

[그림 208] 의 TrainACC 컴포넌트 데이터 흐름] 참조한다.

① DistanceSensor로부터 측정한 거리 값을 전달 받음

② Motor에게 속도 명령 전달

③ LED에게 점등 및 소등 명령 전달

④ RFID에게 태그 값 전달 받음

⑤ Wifi에게 서버로 RFID 태그 값 송신 명령 전달

⑥ Wifi로부터 서버에게 수신 받은 앞 차와의 위치 차이 값 전달 받음

⑦ 앞 차와의 거리 값과 위치 차이 값으로 다음 속도 결정

⑧ Wifi에게 서버와의 연결 시도 명령 전달

⑨ Wifi에게 서버로 Heartbeat 전송 명령 전달

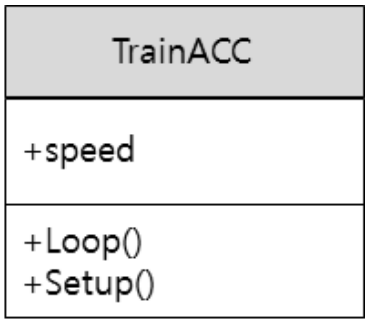


그림 209 TrainACC 컴포넌트

- speed : 현재 속도 값 저장
- Loop() : TA 소프트웨어가 수행해야 할 기능을 관리한다.
- Setup() : TA 소프트웨어가 시작 시 초기 설정을 한다.

○ Motor 컴포넌트

① TrainACC로부터 속도 명령 전달 받음

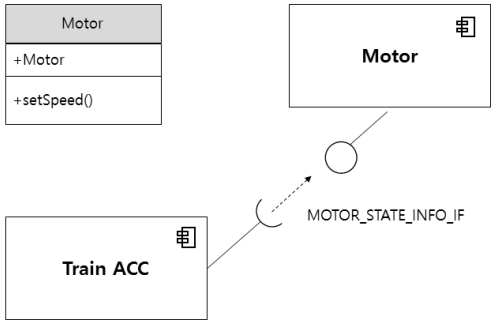


그림 210 Motor 컴포넌트

- Motor : 모터 제어기와 연결된 핀 번호 저장
- setSpeed() : 모터 제어기가 속도 명령을 수행한다.

○ LED 컴포넌트

① TrainACC로부터 LED 점등 및 소등 명령 받음

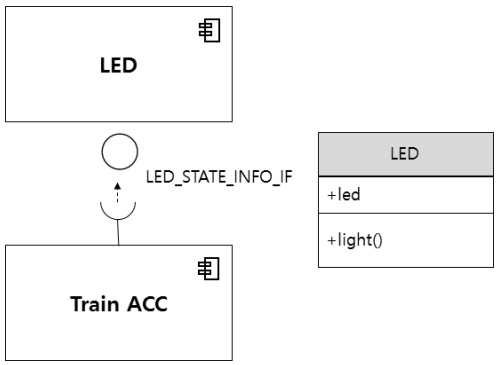


그림 211 LED 클래스

- led : LED가 연결된 핀 번호 저장
- light() : LED를 점등하고 0.1초 후 소등한다.

○ RFID 컴포넌트

① TrainACC에게 RFID 태그 측정값을 전달

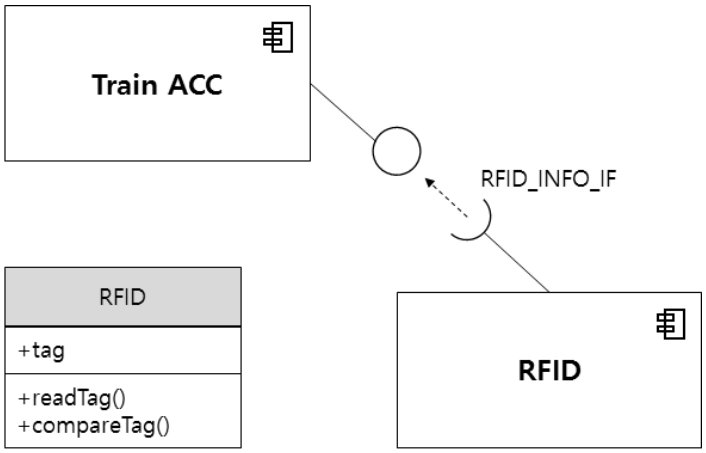


그림 212 RFID 컴포넌트

- tag : 현재 RFID 태그 값 저장
- readTag() : RFID로부터 태그 값을 읽어 들인다.
- compareTag() : RFID 태그 값이 변경되었는지 확인한다.

○ Wifi 컴포넌트

① TrainACC로부터 서버와의 연결 시도 명령 받음

② TrainACC로부터 서버로 데이터 전송 명령 받음

③ TrainACC에게 서버로부터 수신 받은 데이터를 전달

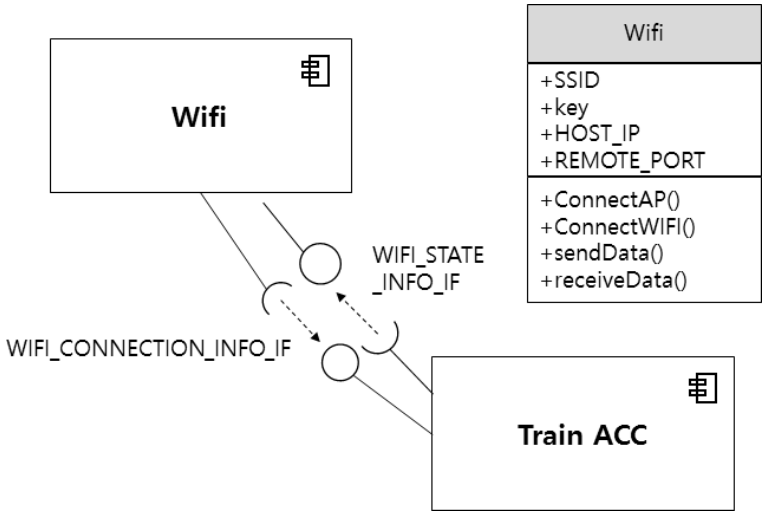


그림 213 Wifi 컴포넌트

- SSID : 연결하고자 하는 네트워크의 SSID
- key : 연결하고자 하는 네트워크의 보안키
- HOST_IP : 연결하고자 하는 중앙 제어부의 IP
- REMOTE_PORT : 연결하고자 하는 중앙 제어부의 포트
- connectAP() : AP에 연결한다.
- connectWifi() : Wifi에 연결한다.
- sendData() : 서버로 데이터를 송신한다.
- receiveData() : 서버로부터 데이터를 수신한다.

제 6 장 결 론

제 1 절 연구요약

철도 시스템은 많은 이점을 제공한다. 지역에서 지역으로 수많은 승객을 한꺼번에 이동시키며, 교통 체증이 없어서 정해놓은 시간을 엄격히 준수한다. 이러한 이점들로 인해서 철도 시스템의 이용자가 해마다 증가하고 있다. 그러나 한편으로, 철도 시스템은 언제든지 사고가 발생할 수 있다. 수많은 승객이 탑승하다 보니, 만약 사고가 발생하면 그 피해 규모는 매우 클 것이다. 그러므로 사고로부터 안전한 철도 시스템을 구축해야 한다. 최근 들어서, 철도 시스템에서 소프트웨어가 차지하는 비중이 높아지고 있다. 우리나라에서도, 운전자 없이 철도 운행을 소프트웨어로 제어하는 노선이 늘고 있다. 따라서 철도 시스템에 사용되는 소프트웨어를 개발할 때에는 무엇보다도 안전성이 확보되어야 한다.

안전한 철도 소프트웨어 개발 프로젝트를 위해서는 철도 안전에 관한 국제 표준을 따라야 한다. 왜냐하면 표준은 수많은 시행착오를 겪으면서 체득한 안전과 관련된 우수 경험을 모아놓은 것이기 때문이다. 그러나 표준에 있는 내용은 추상화 수준이 높아서, 실무 개발자들이 표준을 개발 업무에 바로 적용하기는 쉽지 않다. 이를 위하여 그러므로 본 용역 과제에서는 국제 표준을 기반으로 하여 안전 소프트웨어 개발 가이드를 제작하여 철도 소프트웨어 개발자를 돕고자 하였다. 본 용역 과제의 특장점은 다음과 같다:

- 안전성을 높은 소프트웨어를 개발하기 위해서는 위험원 분석 및 위험도 평가에 대한 이해가 필수적이다. 철도 실무자들의 이해를 돕고자 PHA, SHA, SSHA, IHA, O&SHA, ETA, FTA, FMEA, FHA, HAZOP, FRACAS 등 현업에서 사용하는 거의 모든 기법을 해설하였다.
- 안전 기법과 안전 대책을 의미하는 T&M(Techniques and Measures) 34개를 상세하게 조사하였다. 가능한 철도 관련된 예제를 들어서 T&M을 설명함으로서 철도 종사자들의 이해를 높였다.
- SIL 2 수준에 특화되었다. 국내 철도 소프트웨어 회사의 규모와 수준을 고려해서, 개발하는 소프트웨어가 SIL 2 수준을 달성하도록 상세하게 가이드에 제시하였다.
- 조견표를 제시하였다. 가이드의 각 부분과 국제 표준 및 국내 철도기술기준과의 대응 관계를 제시하였다.
- SW 공학기술 현장적용지원사업에 가이드를 실 개발 업무 적용을 함으로써 가이드 사용성을 제고하기 위한 개선점을 도출하였다.

1. 현황분석

본 과제에서는 철도 산업 분야의 일반적인 현황을 파악하기 위해서 보고서, 간행물, 논문, 신문 기사, 법령 등을 분석하였다. 특히, 안전 소프트웨어 개발 현황을 파악하기 위해서 설문 문항을 제작하여 실무자의 의견을 듣고 이를 분석하였다. 설문 문항은 소프트웨어 공학 및 표준의 이해, 위험 분석, 소프트웨어 개발 방법, 품질 및 형상 관리, 이렇게 네 영역 30문항이다 (부록 D 참조). 설문 조사를 분석한 결과, 소프트웨어 공학의 이해도나 관련된 기술 수준이 낮은 것으로 파악되었다. 설문 조사 뿐만 아니라 현업 실무자도 인터뷰하였다. 그 결과, 소프트웨어 공학의 이해도가 낮은 실무자가 적은 노력으로 사용할 수 있는 국제 표준 기반의 철도 안전 소프트웨어 개발 가이드가 절실히 필요하며, 가이드에는 예제 및 적용 순서가 포함되어야 함을 파악했다.

2. 철도 안전가이드

본 과제에서는 철도 안전 소프트웨어 개발을 위하여 크게 2개 영역으로 구분해 가이드를 구성하였다. 첫 번째 영역은 시스템 안전성 분석 가이드이다. 이것은 표준 IEC 62278 및 IEC 62425를 기반으로 제작하였다. ‘시작이 반이다’라는 속담이 있듯이, 안전 소프트웨어 개발을 위해서는 시작에 해당하는 위험원 식별 및 각 위험원마다 위험도를 추정하고 평가해서, 위험도를 허용 수준 이하로 경감하는 일련의 안전성 분석이 매우 중요하다. 이러한 이유로, 본 과제에서는 시스템 안전성 분석 가이드의 비중을 높게 책정하여 많은 지면을 할애하였다. 두 번째 영역은 소프트웨어 요구사항과 특히 소프트웨어 안전 요구사항을 식별하고 명세하는 요구사항 가이드와, 안전 요구사항을 수행하기에 필요한 서브시스템과 이들 간의 인터페이스를 나타낸 아키텍처 가이드, 아키텍처를 컴포넌트 단위로 더 세분화하여 분할하고 이들을 통합하는 컴포넌트 가이드, 개발된 소프트웨어 제품이 안전 요구사항을 충족하는지를 객관적으로 입증하는 검증 가이드, 구현된 소프트웨어를 배포하고 유지보수하기 위한 가이드를 표준 IEC 62279를 기반으로 제작하였다.

3. 가이드 시범 적용 사례

본 과제에서 제작된 안전가이드를 검증하기 위해서 두 가지 사례에 적용하였다. 첫째는, 차상 표시 장치 MMI (Man Machine Interface)를 개발하는 업체에 적용하였다. 적용 순서는, 실무자를 대상으로 가이드 교육 및 산출물 양식 작성을 먼저 교육하였다. 그 후에, 실무자들이 가이드에 있는 활동을 수행하고 산출물을 작성할 수 있도록 컨설팅을 수행하였다. 마지막으로, 작성된 산출물을 검토하고서, 제작된 가이드 및 산출물 양식을 보완하였다. 시범 적용에 따른 효과는 다음과 같이 정리할 수 있다. 첫째, 이전에

는 파악하지 못했던 안전성 분석에 대한 개념 및 필요성에 대한 인식 개선이 있었다. 둘째, 안전성 분석부터 시스템 및 소프트웨어 안전 요구사항 식별 및 명세, 아키텍처 설계, 컴포넌트 분할 및 통합에 관한 수행 절차 및 수행 기법 등에 관한 이해가 높아졌다. 셋째, 소프트웨어 안전 요구사항에 따른 안전 설계의 중요성을 파악하게 되었다. 짧은 일정에도 불구하고, 시범 적용을 통해서 이러한 정성적인 효과를 얻을 수 있었을 뿐만 아니라 안전 관련 산출물을 확보하게 되어서, 이전 프로젝트 경험을 반복할 수 있는 발판도 확보하였다.

둘째는, 안전가이드를 모형 철도 기반으로 건널목을 통제하는 차단목(level crossing) 시스템 개발에 시범 적용하였다. 차단목 제어에 사용되는 소프트웨어는, 학부에서 소프트웨어 공학 과목을 이수한 학부 학생 둘이서 개발하였다. 이들 학부 학생들에게 가이드를 교육 시킨 이후에, 가이드에 명시된 대로 차단목 시스템의 예비위험원분석, 위험도 분석을 위한 FMEA, 요구사항 활동 등을 수행하게 하였고, 심지어는 가이드에서 요구하고 있는 산출물을 작성하게 하였다. 비록 가이드 작성자의 도움을 받았지만, 큰 어려움 없이 가이드의 활동들을 수행할 수 있었다.

제 2 절 가이드 활용 방안

1. 안전성 분석 절차 수립

본 용역 과제에서 작성된 안전성 분석 가이드는 특정 회사의 업무와는 무관하게 일반적으로 작성되었다. 회사에서 활용하기 위해서는 회사의 규모, 인력, 조직 및 철도 소프트웨어 프로젝트의 성격에 맞는 안전성 분석 절차를 수립해야 한다. 본 가이드 3장에서는 널리 사용되는 안전성 분석 기법인 PHA, HAZOP, FTA, ETA, FMEA 등을 소개하고 있다. 이들을 조합해서 회사 규모에 맞는 안전성 분석 기법을 수립할 수 있다.

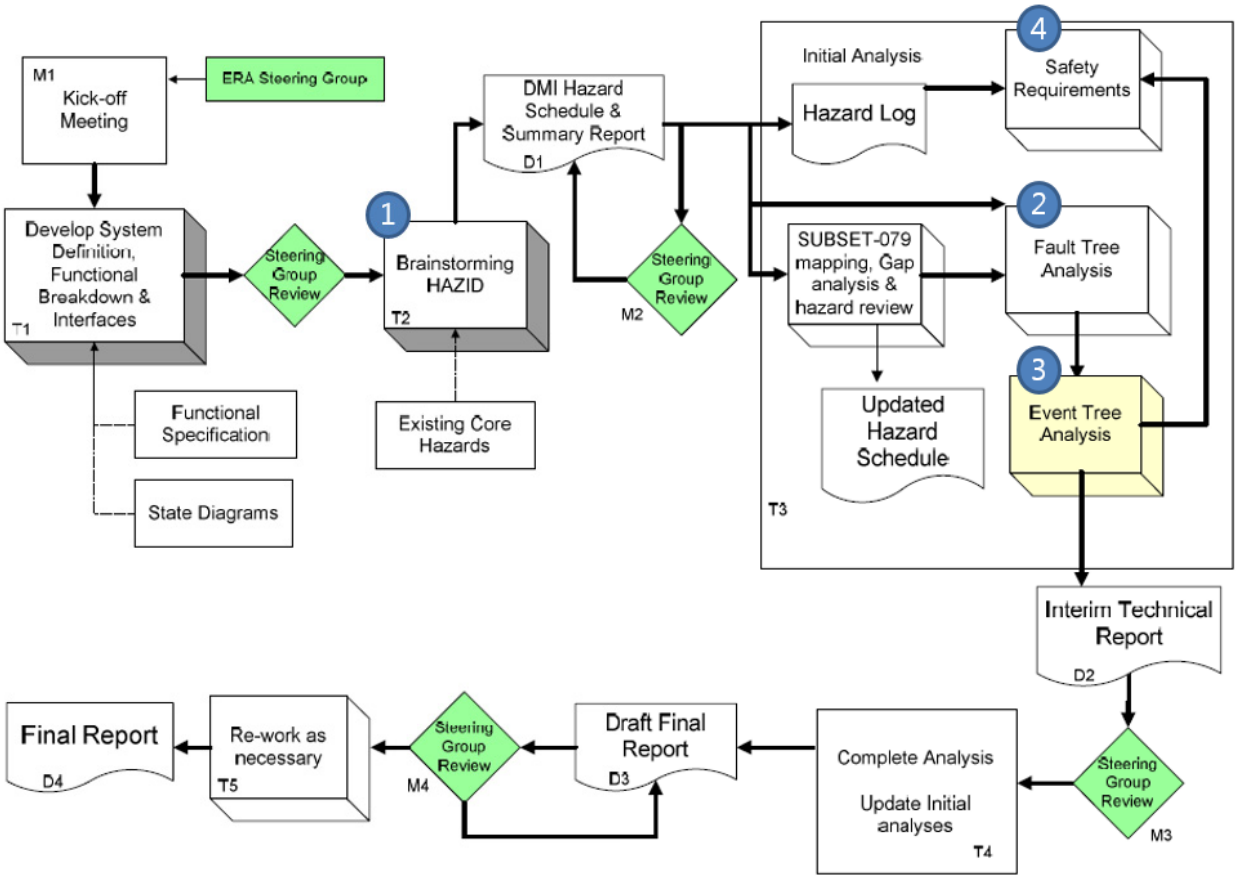


그림 214 열차 DMI (Driver Machine Interface) 안전성 분석 절차

예를 들어 [그림 214]는 로이드레지스터(지금은 리카르도레일)에서 수행한 안전성 분석 절차이다. 운전사에게 열차 정보를 현시하는 DMI의 기능 안전성을 분석하기 위하여 그림에서 보듯이 ① HAZOP과 유사한 방식으로 위험원을 식별하고, ② FTA를 이용하여 각 위험원의 원인 및 그에 따른 발생빈도를 찾아내고, ③ ETA를 이용하여 각 위험원의 결과 및 그에 따른 심각도를 찾아낸 후에, ④ 이들을 토대로 안전 요구사항을 식별한다. 요약하면, DMI 기능 안전성 분석을 위해서 HAZOP, FTA, ETA 기법을 선정하였고, 이러한 기법을 언제 사용해야 하는지에 관한 절차를 수립한 후에 안전성 분석을

수행하였다. “구슬이 서말이라도 꿰여야 보배” 이듯이, 회사 프로젝트 규모에 맞게 가이드에 제시된 안전성 분석 기법을 조합해서 안전성 분석 절차를 수립할 수 있다.

2. 대학 수업 교재로 활용

표준에서도 강조하듯이 소프트웨어 안전성 업무에서는 참여 인력의 역량(competency)이 중요하다. 역량을 높이기 위해서는 교육이 필수적이다. 소프트웨어 공학과 안전성을 겸비한 전문 인력을 양성하기 위해, 대학원 석사과정에서 예를 들어 “Safety Software Engineering” 과목을 개설하여, 본 가이드를 교재로 활용하여 수업할 수 있을 것으로 기대한다. 커리큘럼 예시는 다음과 같다.

주	가이드를 활용한 강의 내용	텀 프로젝트
1주	소프트웨어 안전성 소개 (가이드 1장, 2장 활용)	안전성 프로젝트를 위한 모형 철도 시스템 제안
2주	위험원 식별 및 분석 (가이드 3장 2절 활용)	모형 철도 시스템의 위험원 식별 및 분석
3주	위험도 평가 (가이드 3장 3절 활용)	모형 철도 시스템의 위험도 축소
4주	소프트웨어 안전성 요구사항 파악 (가이드 3장 4절~5절 활용)	모형 철도 시스템의 안전 요구사항 식별
5주	소프트웨어 안전 요구사항 명세 (가이드 4장 1절 활용)	모형 철도 시스템의 안전 요구사항 식별
6주	소프트웨어 아키텍쳐 설계 (가이드 4장 2절 활용)	안전성을 확보하는 아키텍쳐 설계
7주	소프트웨어 컴포넌트 설계 (가이드 4장 3절 활용)	안전성을 확보하는 컴포넌트 설계
8주	텀 프로젝트 중간 보고	
9주	컴포넌트 구현 (가이드 4장 4절 활용)	안전성을 확보하도록 코딩
10주	코드 리뷰 (가이드 부록 B, 활용)	안전성이 확보되었는지 코드를 조사
11주	컴포넌트 테스트 (가이드 4장 5절 활용)	안전성이 확보 여부에 관한 테스팅
12주	소프트웨어 통합 (가이드 4장 4절 활용)	안전성이 확보되도록 통합
13주	소프트웨어 통합 테스트 (가이드 4장 6절 활용)	안전성이 확보 여부를 위해서 통합 시스템 테스트
14주	소프트웨어 확인 (가이드 4장 6절 활용)	전체 소프트웨어가 SRS대로 작성되었는지 확인
15주	산출물 작성 및 리뷰 (가이드 5장 활용)	안전성과 관련된 모든 증거를 수집하여 문서화
16주	텀 프로젝트 최종 보고	

표 304 커리큘럼 예시

수업의 효과를 높이기 위해서, 텀 프로젝트로 사례 시스템을 개발한다. 현실적으로 철도 시스템의 사례를 구하기도 어렵고, 뿐만 아니라 개발 환경에 접근하기도 어려워서,

대학원 수업에서는 모형 철도가 적절하다고 판단한다. 아래 그림은 본 가이드 5장 2절 모형 철도 적용 사례에서 언급한 ACC 시스템이다. 두 열차가 일정한 안전 거리를 유지하야 사고로부터 열차를 보호하는 시스템이다. 이론 수업과 병행해서 모형 철도를 이용하여 안전성 프로젝트를 수행하고 그 결과를 발표하면 수업 효과가 있을 것으로 기대한다.

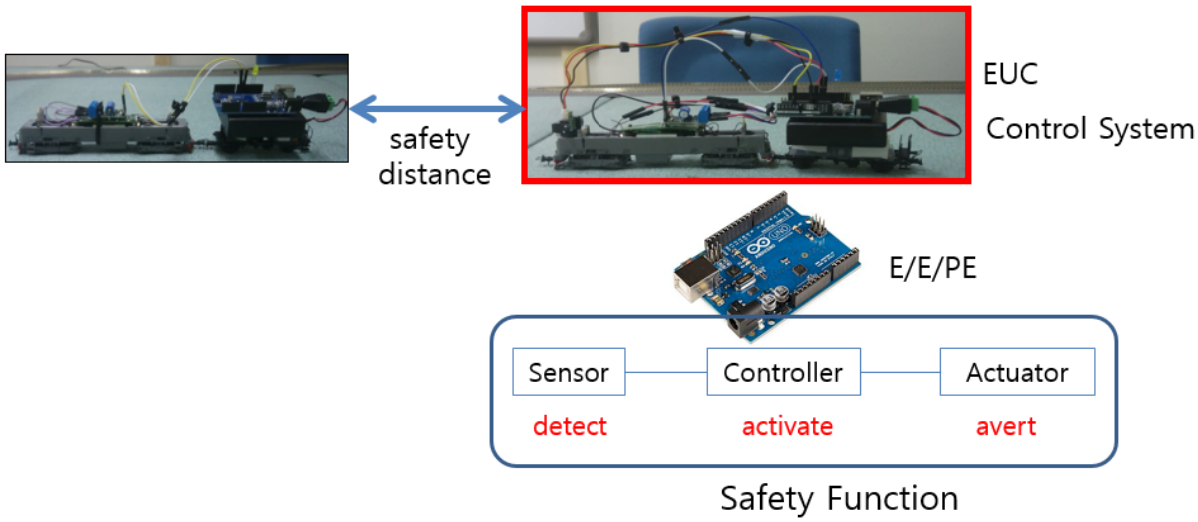


그림 215 모형 철도를 이용하여 개발한 ACC 시스템

3. 기능 안전에 관한 입문서

본 용역 과제는 철도 표준만을 다룬다. 2017년 NIPA 용역 사업에서는 철도 분야뿐만 아니라 IEC 61508 산업 공통, ISO 26262 자동차, IEC 62304 의료 표준도 다루고 있다. 각 가이드마다 차별점이 있다. 철도 가이드는 안전성 분석에 집중하고. 다른 가이드는 테스팅 기법을 강도 높게 다루거나, 또 다른 과제는 안전성 계획 등 지원 활동을 강조할 수 있다. 그래서 네 개의 가이드를 조합하여, 기능 안전에 관한 우수한 입문서로 활용할 수 있을 것이다.

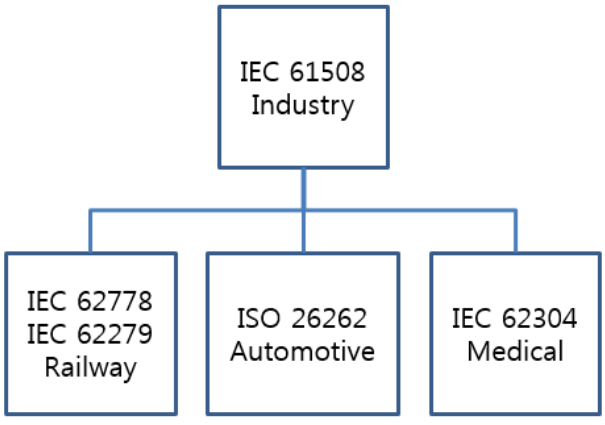


그림 216 NIPA 2017년 안전성 가이드 용역 과제 분야

4. 전문 업체와의 연결성

가이드는 가이드 일 뿐이다! 모든 것을 담고 있는 가이드는 이 세상에 존재하지 않는다. 가이드에 관한 더 상세한 사항은 소프트웨어 공학 및 안전성 전문 컨설팅 업체에 도움을 받는 것이 현명하다. 또한 도구 선정에 관한 문의 역시, 소프트웨어 공학 도구 및 안전성 도구를 공급하고 교육시키는 전문 업체에 의뢰하는 것이 현명하다. 게다가, SIL 인증에 관한 문의 역시, 전문 인증 업체에 의뢰해야 한다. 가이드는 이름대로 안내서 역할 선에서 활용해야 한다.

참고문헌

[1] KRTCS 예비위험원분석(PHA) 보고서, “도시철도용 무선통신기반 열차제어 시스템 표준체계구축 및 성능평가-열차제어시스템 표준체계 및 안전성 평가”

[2] KRTCS 프로젝트 RAMS 관리 계획서, “도시철도용 무선통신기반 열차제어시스템 표준체계구축 및 성능평가-열차제어시스템 표준체계 및 안전성 평가”, (2011)

[3] KRTCS RAMS 관리계획서 작성지침, “도시철도용 무선통신기반 열차제어시스템 표준체계 구축 및 성능평가-열차제어시스템 표준체계 및 안전성 평가”, (2011)

[4] Clifton A. Ericson, ‘Hazard Analysis Techniques for system safety, (2005)

[5] M. Matsumoto, “Changing RAMS for Railways Proposals from Japan”, Special feature article.

[6] 심규돈, 이종우, “차상신호시스템 적용에 다른 위험분석 및 평가활동 연구”, 한국철도학회논문집, 제14권, pp. 121-129. (2011).

[7] IEC 62278 (2002) Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety(RAMS)

[8] IEC 62279 (2015) Railway Applications - Software for Railway Control and Protection Systems.

[9] 황종규, 조현정, 한찬희, 조우식, 안진, 하동명, “열차제어시스템 위험원 분석을 위한 HAZOP-KR에 대한 연구”.한국철도학회논문집, 13권, 제 4호, pp. 396-403, 2010.

[10] IEC 62425, Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signaling.

[11] IEC 62280, Railway applications - Communication, signalling and processing systems - Safety related communication in transmission system.

[12] 소프트웨어 정책 연구소, 소프트웨어 안전성 확보 체계에 대한 연구, (2016)

[13] 한국철도기술연구원, 철도소프트웨어 안전기준 및 체계 구축 연구보고서, (2008)

[14] 한국철도기술연구원, 도시철도용 무선통신기반 열차에서 시스템 표준체계 구축 및 성능평가 최종 보고서, (2014)

[15] 한국철도기술연구원, 열차제어시스템 안전성능 평가 및 사고방지기술 개발 연구보고서, (2011)

[16] 한국철도기술연구원, 철도 안전 통합 매뉴얼, (2012)

[17] 한국교통연구원, 도시철도사고 예방 및 대응체계 개선 연구, (2014)

[18] 국토교통부, “철도차량산업 새 활로 찾기” 공청회 자료, (2016)

[20] 한국교통연구원, 철도서비스 수준 기준 정립 및 개선 연구, (2014)

[21] 한국교통연구원, 고속철도의 안전성 진단과 대응방안, (2014)

[22] 한국철도학회 춘계학술대회 논문집, 국내 철도산업 소재・부품기업 현황 조사, (2015)

[23] 한국교통연구원, 정책밀착형 철도기술의 발굴과 평가에 관한 연구, (2013)

[24] 한국교통연구원, 철도산업 해외진출 활성화 방안, (2011)

[25] 한국철도차량산업협회, 철도차량 제12호, (2016)

[26] 한국교통연구원, 대외경제정책연구원, 중국 철도 발전에 따른 한중 협력 및 대응 방안, (2013)

[27] 국토교통부, 철도사고 분석보고서, (2014)

[28] 철도안전인증센터 구축방안 및 효과분석 연구용역 최종보고서, (2014)

[29] 국토해양부, 철도차량 및 용품 인증제도 개편에 따른 하위법령/기준 마련 및 추진 방안 연구 최종보고서, (2013)

[30] 소프트웨어정책연구소, 소프트웨어 안전(Safety) 산업 동향 조사, (2015)

[31] 소프트웨어정책연구소, 소프트웨어 안전성 확보 체계에 관한 연구, (2016)

[32] 한국교통연구원, 고속철도의 안전성 진단과 대응방안, (2014)

[34] 교통안전공단, 국내외 철도사고 사례, (2011)

[35] “NASA Software Safety Guidebook“, NASA TECHNICAL:NASA-GB-8719.13 (2004)

[36] “SOFTWARE SAFETY STANDARD“, NASA-STD-8719.13B w/Change 1 (2004)

[37] Zaibi Kais(2015) “A UML APPROACH FOR MODELING AND VERIFYING OF RAILWAY SIGNALLING SYSTEMS SPECIFICATIONS“, HAL Id: hal-01166630

[38] J.-L. Boulanger(2006) “From UML to B - a level crossing case study“, WIT Transactions on The Built Environment, Vol 88

[39] “GNU Coding Standards“ https://www.gnu.org/prep/standards/standards.html (2017)

[40] “Google C++ Style Guide“ https://google.github.io/styleguide/cppguide.html (2017)

[41] “C Coding Standard“ https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html (2017)

[42] “C++ Coding Standard “ https://users.ece.cmu.edu/~eno/coding/CppCodingStandard.html (2017)

[43] 위키백과, https://ko.wikipedia.org/wiki/ (2017)

[44] WIKIPEDIA, https://en.wikipedia.org/wiki/ (2017)

[45] 김치수 “쉽게 배우는 소프트웨어 공학“ 한빛아카데미 (2015)

[46] “Encapsulation in C“ https://alastairs-place.net/blog/2013/06/03/encapsulation-in-c/ (2017)

[47] “Conditional and Unconditional Jumps“ https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Mips/jump.html (2017)

[48] “Unconditional Jumps“ http://www.c-jump.com/CIS77/ASM/FlowControl/C77_0030_unconditional_jumps.htmv (2017)

[49] “Unconditional Jumps“ http://books.gigatux.nl/mirror/cinanutshell/0596006977/cinanut-CHP-6-SECT-5.html (2017)

[50] EN 50126 - Railway Application. The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)

[51] 튜브라인란드, YOUNGSANG KIM, ROTECO, 차량기술교육자료, 2014. 10.

[52] Hazard Analysis Techniques for System Safety, 2005, book

[5 3] http://icomod.info/web/ramst/safety/electrical-hv-propulsion-system-of-light-rail-vehicles

[54] IEC61508, Functional Safety of Electrical / Electronic / Programmable Electronic Safety-related Systems, Part 5,IEC, pp. 10- 44, 2010,

[55] 송기태, 이성일, “Risk Graph에 의해 할당된 sil에 따른 철도 승강장 도어 시스템의 정량적 Risk 저감 모델”, pp.141-148, 한국안전학회지, 2011.

[56] 국토해양부고시 제2012-517호, 철도사고 등의 보고에 관한 지침

[57] 황종규, 조현정 외 3명, “열차제어시스템 안전성 활동 기술체계의 분석 및 적용“, 2010.

[58] 진은지, 김명희, 박만곤, “전자 교수학습 시스템의 보안성 평가를 위한 결함트리분석과 고장유형에 대한 영향분석의 통합적 방법“, 한국정보처리학회, 정보처리학회논문지 / 소프트웨어 및 데이터 공학, 제2권, 제1호, 2013, pp. 7-18.

[59] 김주욱, 김영민, “FMEA 안전분석 기법을 활용한 차상중심 열차제어시스템의 아키텍처 무결성 향상을 위한 검증 방법론 구축에 관한 연구“.

[60] 신덕호, 백종현, 이강미, 김용규, “한국형 틸팅열차 차상신호장치 신뢰성관리에 대한 연구,” 한국철도학회논문집, 제12권, 제6호, pp. 825-838. 2009년 12월.

[61] 백종현, 조현정, 이강미, 김건엽, 신덕호, 이재호, “BCT기반 차상중심 지상설비 제어기술 연구,” 2012년도 대한전기학회 하계학술대회 논문집, pp. 1544-1545, 2012년 7월.

[62] 윤용기, 오세찬, 김민수, 김용규, 최준영, 박재영 (2011). 도시철도용 무선통신기반 열차제어시스템 기능배치 연구. 한국철도학회 학술발표대회논문집, 2300-2305.

[63] 빅종현, 조현정, 채은경, 최현영, 김용규 (2013). 선로변 시설물의 지능적 제어를 위한 차상중심 열차제어시스템 시뮬레이션 기반 성능 평가. 한국철도학회 논문집, 16(6), 528-533.

[64] 최명성 외 5명, “차상중심 열차제어시스템 개발에서 모델기반 접근을 통한 안전성 향상에 관한 연구“, (2016).

[65] 김영상, “도시철도차량 적용을 위한 위험도 매트릭스 개발에 관한 연구, 한국안전학회지, pp.111-117, (2011).

[66] 백종현 외 다수, ICT 기반 열차운행 안전성 및 운영효율성 향상 기술개발, 한국찰도기술연구원 보고서, (2014).

[67] 국토교통부, 신호설비 공통 플랫폼 개발 및 철도시스템 인증체계 수립방안 기획. 최종보고서, (2015).

[68] 박승근, “고속선 전자연동장치 국산화 개발에 따른 안전성 확보 방안에 관한 연구”, 우송대학교 철도대학원, (2012).

[69] TTA 표준화 위원회, 소프트웨어 품질 요구사항 작성 지침, 한국정보통신기술협

회,(2011).

[70] 홍연웅, “고장보고분석 및 정비 시스템의 개발에 관한 실증 연구”. 한국데이터정보과학회지, pp.109-119, (2010).

[71] 신덕호 외 6명, “열차제어시스템의 안전입증에 관한 연구”, pp.412-418, (2006).

[72] 이학선, TFM 시스템에서 Fail-Safe 구성 및 최적 시스템의 구현 방안, 서울과학기술대학교, 석사학위, (2009).

[73] 송창석, 모바일 증강현실 소프트웨어의 품질평가 모델에 관한 연구, 숭실대학교 소프트웨어특성화대학원, 석사학위, (2016).

[74] 조우식, 고장률을 이용한 철도 선로변 장치 제어용 PES 차상제어시스템 서비스 가용도 산정에 관한 연구, 서울과학기술대학교, 석사학위, (2015).

[75] 안진, 노희준, 신광호 외 다수, “열차 수동운전시 사고방지를 위한 간이 신호시스템 개발 최종보고서”, 국토교통부 학술정보, (2014).

[76] 한국철도기술연구원, 철도사고방지 및 안전확보를 위한 핵심기술개발 연구, 최종보고서, (2013).

[77] 국토교통부, 철도차량 안전기준에 관한 규칙, 국토교통부령 제1호, (2013)

[78] 국토교통부, 도시철도 차량 기술기준, 국토교통부고시 제 2014-434호, (2013).

[79] 국토해양부, 철도관련 법제 개선 연구, 최종보고서, (2009).

[80] 한국철도기술연구원, 철도차량 안전기준 및 체계 구축 : 차량화재 안전기준 포함, (2008).

[81] 국토해양부, 한국교통연구원, “철도안전법 개정을 위한 연구 최종보고서(안), (2011).

[82] 국토교통부, .국토부 고시, 철도차량 형식승인・제작자 승인・완성검사 시행지침, (2014).

[83] 한국철도기술연구원, 차량/궤도/환경분야 철도용품 인증을 위한 실내/현장 시험규격 정비 및 인증체계 개선방안 연구 최종보고서 Ⅲ (본문7장 3절-7절). (2014).

[84] 이준, 설재훈, 오재학, 이해선, “고속철도 안전성 진단과 대응방안 : 한일 비교를 중심으로”, (2014).

[85] 김영상, “모노레일 차량시스템의 위험도 평가방법에 관한 실증적 연구”, 서울과

학기술대학교, 공학박사, (2013).

[86] 한국철도기술연구원, 철도사고 위험도 분석 및 평가체계 구축 Part Ⅱ : 철도시스템 안전성 검증 및 인증 모델 개발, 보고서, (2011).

[87] 국토교통부 항공・철도 ・철도사고조사위원회, 철도사고 조사보고서, (2014).

[88] 한국철도기술연구원, ICT 기반 열차운행 안전성 및 운영 효율성 향상 기술개발 -ICT 1차년도 보고서, (2011).

[89] 임명재, 위험요인 분석기반 철도 안전관리 개선 방안, 철도안전학과 서울과학기술대학교, 석사학위, (2016).

[90] http://www.sw-eng.kr, 웹진 157호 인사이드 이슈 : 철도분야 SW 품질보증 실현 방안.

[91] 한국철도기술연구원, 도시철도용 무선통신기반 열차제어시스템 표준체계 구축 및 성능평가 보고서, (2015).

[92] https://www.researchgate.net/figure 2. Forward Integrating Method of FMEA to FTA).

[93] 임명재, 위험요인 분석기반 철도 안전관리 개선 방안, 철도안전학과, 서울과학기술대학교, 석사학위, (2016).

부 록

부록 A. 용어집

A-1. 약어

철도 안전 소프트웨어 개발 가이드 작성에서 자주 사용되었던 약어의 전체 이름을 정의한다.

- PHA — Preliminary Hazard Analysis
- SIL — System Integrity Level
- RAMS — Reliability, Availability, Maintainability, and Safety
- HAZOP — Hazard and Operability Study
- FMEA — Failure Mode and Effects Analysis
- FMECA — Failure Mode, Effects and Criticality Analysis
- FTA — Fault Tree Analysis
- SRS — System Requirement Specification
- SSRS — System Safety Requirement Specification
- RAMS — Reliability, Availability, Maintainability and Safety
- IEC — International Electrotechnical Commission
- ETA — Event Tree Analysis
- SHA — System Hazard Analysis
- SSHA — Subsystem Hazard Analysis
- IHA — Interface Hazard Analysis
- O&SHA — Operating and Support Hazard Analysis
- ATO — Automatic Train Operation
- ATP — Automatic Train Protection
- COTS — Commercial off the shelf
- CPU — Central Process Unit
- CRC32 — Cyclic Redundancy Check
- ID — Identification
- IEC — International Electronical Committee
- MMI — Man Machine Interface
- OATP — Onboard ATP
- RFP — Request for Proposals

A-2. 용어 해설

철도 안전 소프트웨어 개발 가이드 작성에서 자주 사용되었던 용어를 해설한다. 가이드 이해에 필요하다고 판단되는 용어들을 국제 표준인 IEC 62278, IEC 6279, IEC 62425에서 대부분 선별하여 풀이하였다. 참조를 용이하도록 용어를 한글 가나다순으로 정렬한 후에, 영어 알파벳순으로 정렬하였다.

용 어	해 설
감사 (audit)	프로젝트가 잘 수행되는지를 살펴보기 위해서, 프로세스에 초점을 맞추어서 조사한다. 보충1) 흔히들 안전성 감사라고 불린다.
거동 (behavior)	철도 시스템엔지니어링 분야에서는 behavior는 거동이라는 의미로 쓰이며, 소프트웨어엔지니어링에서는 행위 등으로 쓰이나 동일한 의미적 차원에서 쓰인다.
검증 (verification)	지정된 명세가 충족되었음을 객관적인 입증으로 하는 것이다. 보충1) 철도 분야에서는 'verification' 용어를 '증명' 으로 사용하고 있다. 반면에, 소프트웨어 공학에서는 이를 '검증' 으로 사용하고 있다.
결함 (fault)	시스템에서 고장을 일으킬 수 있는 비정상적인 조건이다. 보충1) 결함은 랜덤(비 규칙적) 이거나 규칙적일 수 있다.
결함 허용 (fault tolerance)	제한된 수의 하드웨어 또는 소프트웨어 고장이 있는 곳에서 명세된 서비스를 연속적으로 정확하게 제공하기 위한 능력이다.
고객 (customer)	소프트웨어를 포함해서 철도 제어 및 보호 시스템을 발주하는 기관, 부서 또는 개인이다.
고장 (failure)	실제 성능과 기대되는 성능 간의 용인할 수 없는 차이이다. 보충1) 오작동 이라는 용어로도 표현된다.
규칙적 고장 (systematic failure)	오류로 인한 오작동이다. 특히, 어떤 특별한 입력 값의 조합이나 또는 어떤 특별한 환경적 조건하에서 오작동을 일으킨다. 보충1) 소프트웨어 고장이 이 유형에 속한다.
무작위 고장 (random failure)	부품의 노후화 등의 비 규칙적인 원인에 의해서 발생되는 고장이다. 보충1) 하드웨어 고장이 이 유형에 속한다.
발생빈도 (likelihood)	사고가 발생할 가능성이다. 정량적인 확률 또는 빈도 값으로 표현하거나 정성적인 범주로 표현할 수 있다. 보충1) 철도에서는 발생가능성을 여섯 범주로 나타낸다: 빈번한 발생(Frequent), 가능성 있는 발생(Probable), 종종 발생 가능(Occasional), 발생가능성이 미약함(Remote), 발생 가능성이 거의 없음(Improbable), 발생 가능성이 전혀 없음(Incredible).
사고 (accident)	사람에게 피해를 주는 의도하지 않은 사건이나 또는 의도하지 않은 사건들의 연속이다.
상용 소프트웨어 (commercial off-the-shelf software, COTS)	상업적으로 구매하여 바로 이용 가능한 소프트웨어이다.
생명주기 (life cycle)	국내 소프트웨어엔지니어링 분야에서는 life cycle에 대해서 생명주기로 많이 쓰이나, 철도 분야에서는 대체적으로 생명주기라는 용어로도 많이 쓰인다.
소프트웨어 생명 주기 (software life cycle)	소프트웨어가 계획에서 시작하여 더 이상 사용할 수 없어 종료되는 기간 동안에 행해지는 일련의 활동들이다. 보충1) 소프트웨어 생명 주기는 대체적으로 요구사항, 설계, 테스트, 통합, 유지보수 활동들을 포함한다.

용 어	해 설
소프트웨어 안전 무결성 등급 (software safety integrity level)	소프트웨어에 적용되어야 하는 기법과 측정 등을 결정하는 등급 번호이다. 보충1) 안전성 관련 소프트웨어는 0이 가장 낮고 4가 가장 높은 5개의 안전 무결성 등급으로 분류된다.
소프트웨어 유지보수 (software maintenance)	소프트웨어를 현장에 실제 적용한 후에, 소프트웨어의 기능 강화 또는 수정 등을 목적으로 수행하는 활동이다.
시스템 생명 주기 (system life cycle)	개념 정립에서부터 시작하여 시스템을 더 이상 사용할 수 없어서 폐기 처분하기 까지, 그 동안에 행해지는 일련의 활동들이다.
시스템 안전 무결성 등급 (system safety integrity level)	하드웨어와 소프트웨어로 구성되는 통합된 시스템이 명세된 안전성 요구사항에 부합할 것이라는 필요한 확신의 정도를 나타내는 분류 번호이다.
심각도 (severity)	사고 발생에 따른 피해 규모의 양이다. 정량적인 값이나 또는 정성적인 범주로 표현할 수 있다. 보충1) 철도에서는 심각도를 네 범주로 나타낸다: 치명적(Catastrophic), 중대(Critical), 중대하지 않음(Marginal), 사소(Insignificant).
안전 관련 시스템 (safety-related system)	사고를 방지하는 기능을 수행하는 시스템이다.
안전 또는 안전성 (safety)	수용 불가능한 위험으로부터 자유로운 것이다. 다시 말해서, 수용 불가능한 위험이 발생하지 않는 것이다. 보충1) 바꾸어서 얘기하면, 발생하는 위험은 모두 수용할 수 있다는 의미이다.
안전 무결성 (safety integrity)	안전 관련 기능이 지정된 조건하에서 성공적으로 수행될 수 있는 가능성이다. 보충1) 지정된 조건은 지정된 운영 환경 및 시간 이내를 포함한다.
안전 무결성 등급 (safety integrity level)	안전 관련 시스템에 할당된 안전성 기능에 관한 안전 무결성 요구사항을 명시한 미리 정의된 숫자 중의 하나이다. 보충1) 철도 시스템은 0~4 중에서 하나이다. 숫자가 높을수록 안전 무결성 등급이 높다.
안전성 계획 (safety plan)	철도 프로젝트가 안전성 요구사항을 만족하도록 안전성 분석의 절차를 규정하고, 조직을 구성하고, 담당 인력 등을 배치하는 일련의 준비 활동이다.
안전성 규제 기관 (safety regulatory authority)	철도에 대한 안전성 요구사항을 조정하거나 동의하고, 철도가 그 요구사항을 충족함을 보장하는데 책임이 있는 정부 기관이다.
안전성 기능 (safety function)	안전 요구사항의 부분 또는 전체를 구현한 기능이다.
안전성 요구사항 명세서 (safety requirements specification)	시스템이 안전하다고 판정 받기 위해서 반드시 만족해야 하는 안전성 요구사항을 적어 놓은 문서이다.
안전성 활동 (safety activity)	위험원 식별, 위험원 분석, 위험도 평가, 위험도 저감대책 구현 및 이의 확인 등 안전과 관련된 활동들을 지칭하는 일반적인 용어이다.
예비위험원분석 (preliminary hazard analysis)	프로젝트 초기에 위험원을 식별하고 그에 따른 위험도를 추정하기 위한 활동이다.
오류 (error)	시스템이 오작동 되도록 의도한 설계에서 벗어 난 상태이다.
요구사항 관리 (requirements management)	요구사항을 추출, 문서화, 분석, 우선순위 정함, 변경의 통제 및 연관된 이해관계자들과 의사소통하는 활동이다/ 보충1) 요구사항 관리는 프로젝트 내내 반복적으로 수행된다.

용 어	해 설
위험 또는 위험도 (risk)	피해의 발생 가능성 및 그 피해의 심각도, 이 둘의 조합이다. 보충1) 위험도는 위험이 수치로 계량화 되었을 때의 표현이다.
위험원 (hazard)	사고를 일으킬 수 있는 조건이다. 다른 말로는 피해의 잠재적인 원인이다. 보충1) ‘조건’은 상태 또는 상황 등으로도 불린다. 보충2) 위험원은 특정 시스템 문맥 내에서 언급되어야 한다.
위험원 로그 (hazard log)	식별된 위험원의 상세 정보를 기록한 문서이다. 여기에는 각 위험원의 상태, 결정된 사항들, 사용된 해법 및 구현 상태 등을 기록한다. 보충1) 안전성 로그로도 불린다.
저감대책 (mitigating measures)	위험도를 낮추기 위해서 실시하는 조치이다.
종합 안전 대책 기술서 (safety case)	제품이 명시된 안전성 요구사항을 모두 충족한다는 문서화된 입증이다. 보충1) 시스템이 안전 요구사항을 만족했으며 그리고 시스템과 관련된 위험이 허용 수준 이하로 감소되었음을 입증하는 데 필요한 문서들의 모임을 지칭한다.
준수 (compliance)	제품의 특성 또는 속성이 정해진 요구사항을 충족한다는 것의 입증이다.
철도 당국 (railway authority)	철도 시스템의 운영 규제에 관한 전반적인 책임을 갖는 기관이다.
추적성 (traceability)	개발 과정의 둘 또는 그 이상의 결과물들 간에 수립된 관계의 수준, 특히 서로에 대해 전/후 또는 주/종 관계를 연결하는 것이다.
컴포넌트 (component)	소프트웨어 구조 및 설계에 대하여 잘 정의된 인터페이스와 행위를 갖는 소프트웨어의 부품이다.
테스팅 (testing)	소프트웨어를 실제 실행해가면서, 요구사항 명세서와 비교하여 소프트웨어의 행위와 성능을 확인하는 활동이다.
통합 (integration)	구조 및 설계 명세서에 따라서 소프트웨어 컴포넌트를 조립하고 확인하는 활동이다.
평가 (assessment)	무슨 제품이 만들어지는지를 살펴보기 위해서, 제품에 초점을 맞추어서 조사한다. 보충1) 흔히들 독립 평가라고 불린다.
피해 (harm)	사고가 발생했을 때 겪는 고통이다. 보충1) 사람이 겪는 고통인 부상 및 사망 등이 있다. 또는 환경적 손실, 재산적 손해 등도 있다.
허용 위험도 (tolerable risk)	철도 당국에서 허용할 수 있는 제품 위험도의 최대 등급이다.
형상 관리 (configuration management)	형상 항목의 기능적 물리적 특성을 식별하고 이를 문서화하고, 이들에 대한 변경을 통제하고, 변경 처리를 기록하고 보고하는 활동이다.
형상 관리자 (configuration manager)	문서, 소프트웨어 그리고 관련된 도구들(변경 관리를 포함하는)의 형상을 관리하는 절차를 구현하고 수행하는 책임을 갖는 기관, 부서 또는 개인이다.
확인 (validation)	발주자의 요구사항이 충족되었음을 객관적인 입증으로 하는 것이다. 보충1) 철도 분야에서는 ‘validation’ 용어를 ‘검증’으로 사용하고 있다. 반면에, 소프트웨어 공학에서는 이를 ‘확인’으로 사용하고 있다.

용 어	해 설
FMEA (Failure Mode and Effects Analysis)	안전성 분석을 위해서 제품의 모든 고장 모드를 차례로 하나씩 검토해가면서 고장 모드에 따른 결과를 식별하는 기법이다.
FRACAS (Failure Reporting Analysis & Corrective Action System)	시스템 개발 및 운용과정에서 발생한 문제 및 고장 해결을 위한 신뢰성 관리 시스템이다. 장비 및 시스템에 대한 지속적인 모니터링을 통하여 고장과 관련 있는 다양한 데이터를 수집하고 고장 데이터의 분석을 통해 신뢰성 정보를 제공함으로써 기존 시스템을 개선, 운영하거나 새로운 시스템을 개발, 운영하는 생명주기 동안 신뢰성을 관리하는 종합 시스템이다.
FTA (Fault Tree Analysis)	트리의 루트에 해당하는 최상위 이벤트가 어떻게 발행할 수 있는지를 논리적인 조합으로 나타내고 분석하는 기법이다.
HAZOP (Hazard and Operability Study)	안내어를 사용하여 설계 의도로부터 벗어난 모든 이상현상을 식별하는 체계적이고 구조화된 기법이다. 보충1) ‘안내어’는 ‘지시어’라고도 불린다.
RBD (Reliability Block Diagram)	신뢰성 블록 다이어그램은 고장확률 및 신뢰도의 계산에 사용하는 직렬과 병렬의 조합 형태의 그림이다. 시스템 구조분석에서 나타난 기능을 중심으로 작성한다. 각블록은 물리적인 부품그룹 혹은 기능적인 개체와 일치하며 서로 독립적인 고장을 목적으로 선정된다.

부록 B. 철도 표준에서 사용되는 기법 및 대책에 관한 해설

B-1. 인공 지능 결함 수정

○ 목적

기법들과 프로세스 모델들 그리고 어떤 종류의 온라인 안전성 및 신뢰성 분석을 혼합(조합)하여 가능한 위험원에 대하여 유연한 방법으로 대처할 수 있음

○ 설명

규칙들이 명세로부터 직접 유도될 수 있고 그것들에 대하여 점검할 수 있으므로, 특히 결함 예측(추이 계산), 결함 수정, 유지보수 및 감독 행위는 시스템의 다양한 채널에서 매우 효과적인 방법으로 인공지능 기반 시스템에 의해서 지원될 수 있다.

이미 어떤 설계 및 구현 규칙들을 마음속에 갖고 있음으로써 은연중 명세에 주입된 어떤 공통적인 결함들은 이 접근법에 의해서 효과적으로 회피될 수 있는데, 특히 모델들과 기법들의 조합을 기능적 또는 설명적 방법으로 적용하는 경우이다. 요구된 안전성과 신뢰성을 만족하기 위하여 결함들이 수정될 수 있고 고장의 영향이 최소화되는 기법들이 선택된다.

B-2. 분석 가능한 프로그램

○ 목적

프로그램 분석이 쉽게 될 수 있도록 프로그램을 설계하는 것. 프로그램 행위는 그 분석에 기초하여 완전하게 시험할 수 있어야 한다.

○ 설명

이 의도는 정적 분석 기법으로 분석하기 쉬운 프로그램을 만드는 것이다. 이것을 달성하기 위하여 구조적 프로그래밍의 규칙들을 따라야 한다. 예를 들어서,

- 컴포넌트 제어 흐름은 구조화된 구성들 즉, 순서, 반복 및 선택으로 이루어져야 한다.
- 컴포넌트는 작아야 한다.
- 컴포넌트를 통하는 가능한 경로의 개수는 작아야 한다.
- 개별적인 프로그램 부분들은 가능하면 서로 비 결합적으로 설계되어야 한다.
- 입력과 출력 인자들 간의 관계는 가능한 단순해야 한다.
- 분기와 반복 결정들은 컴포넌트 입력 인자들과 단순하게 연관되어야 한다.
- 서로 다른 유형들의 맵핑 간의 경계는 단순해야 한다.

B-3. 과부하/스트레스 시험

○ 목적

시험 대상이 정상적인 작업 부하를 쉽게 견딘다는 것을 보여주기 위하여 시험 대상에 예외적으로 높은 작업 부하를 주는 것.

○ 설명

과부하/스트레스 시험에 적용할 수 있는 다양한 시험 조건들이 있다. 이들 시험 조건들 중 일부는 아래와 같다.

- 만약 폴링 모드에서 동작중이면 시험 대상은 정상 조건하에서 보다 단위 시간 당 훨씬 더 많은 입력 값의 변화를 갖는 조건
- 만약 요청 시 동작하는 모드이면 시험 대상에 대한 단위 시간 당 요청 수는 일반 조건을 넘어 증가하는 조건
- 데이터베이스의 크기가 중요한 역할을 한다면 정상 조건을 넘어 증가된 크기의 조건
- 중요한 장치들은 최대 속도 또는 최저 속도로 차례대로 조정하는 조건
- 극단적인 경우를 위해서, 모든 중요한 요소들은 가능한데 까지 동시에 경계 조건에 놓음

이러한 시험 조건들 하에서 시험 대상의 시간에 따른 반응이 평가될 수 있다. 부하 변화의 영향이 관찰될 수 있다. 내부 버퍼 또는 동적 변수, 스택 등의 정확한 크기가 검사될 수 있다.

B-4. 경계값 분석

○ 목적

파라미터의 한계 또는 경계에서 발생하는 소프트웨어 오류의 제거

○ 설명

프로그램의 입력 범위는 입력 클래스의 개수로 나눠진다. 시험은 클래스의 경계 및 극단적인 경우를 포함해야 한다. 시험은 명세의 입력 범위의 경계가 프로그램의 입력 범위의 경계와 일치하는지 검사한다. 직접 또는 간접적적인 변형에서 0 값의 사용은 자주 오류가 발생하는 경향이 있으며 특별한 주의가 요구된다.

- 0으로 나누기
- 인쇄 안 되는 제어 문자
- 비어 있는 스택 또는 비어 있는 리스트 원소
- null 행렬
- 0 테이블 원소

일반적으로 입력에 대한 경계는 출력 범위에 대한 경계와 직접 대응한다. 시험 케이스는 출력이 그것의 극한 값으로 되도록 작성되어야 한다.

또한 명세의 경계값을 초과하는 출력을 야기하는 테스트 케이스를 명세하는 것이 가능한지 고려한다. 만약 출력이 순차적인 데이터이면, 예를 들어 인쇄된 테이블, 첫 번째와 마지막 원소 그리고 비어 있거나, 1개, 2개의 원소가 있는 줄에 특별히 주의해야 한다.

B-5. 역방향 복구

○ 목적

하나 또는 그 이상의 고장에 직면하여 정확한 기능적 연산을 제공

○ 설명

만약 결함이 발견되면, 시스템은 이전의 내부 상태, 전에 증명된 상태로 복원 된다. 이 기법은 소위 잘 정의된 체크포인트에서 내부의 상태를 주기적으로 저장함을 내포한다. 이 저장은 전체적으로 (완전한 데이터베이스에 대하여) 또는 점진적으로 (체크포인트 간의 차이점만) 될 수 있다. 그 후에 시스템은 저널링(활동의 감사 추적), 보상(이들 변경의 모든 영향이 무효화됨) 또는 외부적(수동) 상호작용을 사용하여 그동안 발생한 변경에 대하여 보상해야 한다.

B-6. 원인 결과 다이어그램

○ 목적

시스템에서 개발할 수 있는 이벤트들의 순서를 기본적인 이벤트의 조합 결과로, 도식화된 모양으로 만드는 것

○ 설명

이 기법은 고장-트리와 이벤트-트리의 조합으로 간주될 수 있다. 중대한 이벤트로부터 시작해서, 원인-결과 그래프는 순방향과 역방향으로 추적 된다. 역방향에서는 중대한 이벤트를 최상위 이벤트로 갖는 고장 트리와 동등하다. 순방향에서는 이벤트로부터 발생하는 가능한 결과들이 식별된다.

그래프는 그 꼭짓점으로부터 여러 가지 분기를 따라 전파되는 조건들을 설명하는 꼭지점 기호를 포함할 수 있다. 시간 지연이 또한 포함될 수 있다. 이 조건들은 고장 트리와 함께 기술될 수 있다. 이벤트 전파의 연결선들은 다이어그램을 더욱 간결하게 만들기 위하여 논리적인 기호와 조합될 수 있다. 표준적인 기호들이 원인 결과 다이어그램

에서 사용되기 위하여 정의된다. 그 다이어그램들은 어떤 중대한 결과의 발생 확률을 계산하는데 사용될 수 있다.

B-7. 체크리스트

○ 목적

특정한 요구사항들을 처리하기 보다는 시스템의 모든 측면에서 중요한 평가를 촉진 시키는 것

○ 설명

체크리스트를 수행하는 사람에 의해서 완성되는 질문들의 세트. 많은 질문들이 일반적인 성향이며 평가자는 질문들을 평가되는 특정한 시스템에 가장 적합한 것처럼 해석해야 한다.

확인되는 소프트웨어와 시스템에서 다양한 변화에 적용을 위하여, 대부분의 체크리스트는 다양한 종류의 시스템에 적용 가능한 질문들을 포함한다. 결과적으로 사용되는 체크리스트에는 다루게 되는 시스템과 관련 없는 질문들이 있을 수 있으며 이 질문들은 무시되어야 한다.

동일하게 특정한 시스템에 대하여, 그 시스템에 구체적인 질문들을 표준 체크리스트에 보완하는 것이 필요할 수 있다. 어떤 경우에도 체크리스트의 사용은 체크리스트를 선택하고 적용하는 엔지니어의 전문 지식과 판단에 중대하게 의존적임이 명확해야 한다. 결과적으로 엔지니어에 의한 결정은, 선택된 체크리스트에 관한, 그리고 어떤 추가적인 또는 불필요한 질문들은 완전하게 문서화되고 정당성을 보여야 한다.

체크리스트의 적용이 검토될 수 있고 다른 기준이 사용되지 않는다면 동일한 결과가 달성됨을 확실하게 하는 것이 목적이다. 체크리스트를 완성하는데 있어서 목적은 가능한 간결하게 하는 것이다. 광범위한 정당성이 필요할 때는 추가적인 문서를 참조함으로써 되어야 한다.

각각의 질문에 대하여 결과를 기록하기 위하여 합격, 실패, 미정 또는 어떤 유사한 제한된 표시들이 사용되어야 한다. 이 간결함은 체크리스트 평가의 결과에 대하여 전체적인 결론에 도달하는 과정을 대단히 단순화한다.

B-8. 제어 흐름 분석

○ 목적

결함이 있거나 잠재적으로 정확하지 않은 프로그램 구조를 찾아냄

○ 설명

제어 흐름 분석은 좋은 프로그래밍 관습을 따르지 않는 의심스러운 코드 영역을 식별한다. 프로그램은 다음과 같이 분석될 수 있는 방향성 그래프 형식으로 분석된다.

- 접근되지 않는 코드, 예를 들어, 도달할 수 없는 코드의 블록을 나가는 조건 없는 점프
- 매듭진 코드, 제어 그래프가 연속적인 그래프 축소에 의해서 단일 노드로 감소되는 잘 구조화된 코드.
- 부실하게 구조화된 코드는 오직 여러 개의 노드로 구성된 매듭으로 축소될 수 있다.

B-9. 공통 원인 고장 분석

○ 목적

이중화 시스템 또는 이중화된 서브시스템에서 이중화의 혜택을 해치는 잠재적인 고장을 식별

○ 설명

컴퓨터 시스템은 하드웨어와 다수결 투표에 종종 이중화를 사용하여 설비의 안전성 보호를 의도했다. 이 기술은 컴퓨터 시스템에서 데이터의 정확한 처리를 방해하는 경향이 있는 임의의 컴포넌트 고장을 회피하기 위해 사용된다. 그러나 어떤 고장들은 하나의 컴포넌트 이상에 대해서 공통적인 고장이 될 수 있다. 예를 들어 만약 컴퓨터 시스템이 단일 공간에 설치된다면, 에어컨의 결점들이 이중화의 혜택을 감소시킬 수 있다.

화재, 홍수, 전자기적 간섭, 항공기 추락, 지진과 같은 컴퓨터 시스템에 대한 외부적 영향에 대한 것 역시 동일하다. 컴퓨터 시스템은 또한 운영 및 유지보수와 관련된 사고에 의해서도 영향을 받을 수 있다. 그러므로 적절하고 잘 문서화된 절차들이 운영 및 유지보수를 위해 제공되어야 하는 것이 필수적이다. 운영 및 유지보수 인력의 포괄적인 훈련 또한 필수적이다. 내부적인 영향들 또한 공통 원인 고장(CCF)에 대한 주범이다. 내부적인 영향들은 공통 또는 동일한 컴포넌트 및 인터페이스의 설계 오류, 뿐만 아니라 고물이 되어가는 컴포넌트들에서 기인할 수 있다. 공통 원인 고장 분석은 그러한 잠재적인 공통 고장에 대하여 시스템을 조사해야 한다.

공통 원인 고장 분석의 기법들은 독립적인 팀에 의한 일반적인 품질 관리, 설계 검토, 검증 및 테스팅과 유사한 시스템의 경험으로 실제 사건을 분석한다. 그러나 분석의 범위는 하드웨어를 넘는다.

'다양한 소프트웨어'가 이중화된 컴퓨터 시스템의 어려운 체인에서 사용될지라도, 공통 원인 고장에 단서가 될 수 있는 소프트웨어 접근법에서 어떤 공통성이 있을 수 있다. 예를 들면, 공통 명세서의 오류. 공통 원인 고장들이 정확하게 동일한 시간에 발생하지 않을 때, 모든 체인에서 이 고장이 공통 고장이 되기 전에 고장의 탐지를 이끌어야 하는 이중화 체인들 간에 비교 방법을 사용하여 예방이 취해질 수 있다. 공통 원인 고장 분석은 이 기술을 고려해야 한다.

B-10. 데이터 흐름 분석

○ 목적

결함이 있거나 잠재적으로 정확하지 않은 프로그램 구조를 찾아냄

○ 설명

데이터 흐름 분석은 제어 흐름 분석에서 얻어진 정보와 코드의 다양한 부분에서 변수들이 읽고, 써진 정보를 결합시킨다. 분석은 다음 사항을 검사할 수 있다.

- 값을 기록하기 전에 값을 읽은 변수들. 이것은 오류가 될 가능성이 아주 높으며, 확실히 나쁜 프로그래밍 관습이다.
- 값을 읽지 않고 한번 이상 값을 기록하는 변수. 이것은 누락된 코드를 가리키는 것일 수 있다.
- 값은 기록되었으나 값을 읽지 않는 변수. 이것은 중복된 코드를 가리키는 것일 수 있다.

실제 데이터 흐름들이(프로시저들 간에 그리고 프로시저 내부에서 모두) 설계 의도와 비교되는 정보 흐름 분석으로 알려진 데이터 흐름 분석의 확장이 있다. 이것은 일반적으로 도구가 읽을 수 있는 구조화된 주석을 사용해서 의도된 데이터 흐름들이 정의되는 컴퓨터 화된 도구에 의해 구현된다.

B-11. 데이터 흐름 다이어그램

○ 목적

프로그램을 통하는 데이터 흐름을 도식화된 형식으로 설명

○ 설명

데이터 흐름 다이어그램은 데이터 입력이 어떻게 출력으로 변환되는지 각각의 단계에서 다른 변화를 표현하는 다이어그램으로 문서화한다. 데이터 흐름 다이어그램이 기본적인 컴포넌트는 다음을 포함한다.

- 함수, 원으로 표시

- 데이터 흐름, 화살표로 표시
- 데이터 저장, 열린 사각형으로 표시
- 입/출력, 특별한 종류의 사각형

데이터 흐름 다이어그램은 입력이 어떻게 출력으로 변환되는지 설명한다. 데이터 흐름 다이어그램은 제어 정보 또는 순서 정보를 포함하지 않으며, 포함해서는 안 된다. 각각의 원은 그것의 입력들이 이용 가능하자마자 그것의 출력들로 변환하는 독립적인 블랙박스로 간주될 수 있다.

데이터 흐름 다이어그램의 주요한 이점중 하나는 이들 변환들이 어떻게 구현되는지에 대한 어떠한 가정도 없이 변환들을 보여주는 것이다. 데이터 흐름 다이어그램의 준비는 시스템 입력들과 시스템 출력들을 향하는 활동을 고려하는 접근법이 최고이다.

각각의 원은 다른 변화를 나타낸다. - 원의 출력은 어떻게 해서든 그것의 입력과 달라야 한다. 다이어그램의 전체적인 구조를 결정하는 규칙은 없으며 데이터 흐름 다이어그램을 구축하는 것은 시스템 설계의 창의적인 측면중 하나이다. 모든 설계처럼, 최종 다이어그램을 만드는 것은 초기 시도들과 각 단계에서 정제되는 반복적인 프로세스이다.

B-12. 데이터 기록 및 분석

○ 목적

개별 프로젝트 및 인력으로부터 관련된 데이터를 확인, 분석, 기록함으로서 소프트웨어 프로세스 향상을 용이하게 함. 데이터의 관련성은 조직의 전략적인 목표에 의해서 결정된다. 그 목표는 예를 들면 결함 방지 효율성에 대한 것 같이 그것에 대한 주장과 관계가 있는 특정한 소프트웨어 개발 방법의 평가로 향할 수 있다.

○ 설명

데이터 기록 및 분석은 소프트웨어 프로세스 향상의 필수적인 부분을 구성한다. 유효한 데이터의 기록은 소프트웨어 개발 프로세스에 관하여 더 많은 것을 배우고 다른 소프트웨어 개발 방법들을 평가하는 중요한 부분을 나타낸다.

상세한 기록들이 프로젝트 및 개개인 모두에 대해서 프로젝트 기간 동안 유지된다. 예를 들어, 어떤 엔지니어는 다음과 같은 것들을 기록하도록 요구받을 수 있다.

- 개별 컴포넌트에 대해 확장한 노력
- 각각의 컴포넌트에 대해 수행한 테스팅
- 결정 및 근거
- 프로젝트 마일스톤(이정표)의 달성
- 문제 및 해결책

프로젝트 기간 동안 그리고 프로젝트의 종료 시 이러한 기록들은 매우 다양한 정보를 수립하기 위하여 분석될 수 있다. 개발 프로젝트 동안 만들어진 어떤 결정들에 대한 근거가 유지보수 엔지니어에게 항상 알려지는 것은 아니므로 특히 데이터 기록은 컴퓨터 시스템의 유지보수에 매우 중요하다.

부족한 계획 때문에, 데이터 기록은 종종 과도한 분량 및 초점을 벗어나는 경향이 있다. 데이터 기록은 목적, 질문 그리고 무엇이 전략적으로 조직에 중요한지 연관된 기준들에 의해서 되어야 한다는 원칙을 따르는 것으로 이것을 피할 수 있다. 원하는 정확성을 달성하기 위하여, 데이터 기록과 확인 절차는 개발과 동시에 예를 들어, 형상 제어 절차의 일부로써 진행되어야 한다.

B-13. 결정 테이블 (진리표)

○ 목적

명확하고 일관성 있는 명세와 복잡한 논리적 조합 및 연관성의 분석을 제공

○ 설명

이들 관련된 방법들은 이진 프로그램 변수들 간의 논리적 연관성을 간결하게 기술하기 위하여 이차원 테이블을 사용한다. 두 가지 방법의 간결함과 테이블 모양 본질은 코드에 표현된 복잡한 논리적 조합의 분석 수단으로써 적합하다. 두 가지 방법은 명세로서 사용된다면 잠재적으로 실행 가능하다.

B-14. 방어적 프로그래밍

○ 목적

실행하는 중 비정상적인 제어 흐름, 데이터 흐름 또는 데이터 값을 탐지하고 이들에 대해서 미리 결정되고 허용할 수 있는 방식으로 대처하는 프로그램을 제작

○ 설명

많은 기법들이 제어 비정상 또는 데이터 비정상을 검사하기 위하여 프로그래밍 중 사용될 수 있다. 이들 기법은 잘못된 데이터 처리 가능성을 감소하기 위하여 시스템의 프로그래밍 처음부터 끝까지 체계적으로 적용될 수 있다.

방어적 기법들의 두 가지 공통 영역이 식별될 수 있다. 본질적으로 오류-안전 소프트웨어가 그 자신의 설계 결점을 수용하기 위하여 설계 된다. 이러한 결점들은 설계 또는 코딩의 간단한 오류, 또는 잘못된 요구사항 때문일 수 있다. 다음 리스트는 몇 가지 방어적 기법들이다.

- 변수들은 범위가 검사되어야 한다.
- 가능하다면, 값이 타당성 있는지 검사되어야 한다.
- 프로시저에 대한 인자들은 유형이 있어야 하며, 차원과 범위가 프로시저 시작에서 검사되어야 한다.

이 세 가지 권고는 프로그램에서 다루어지는 숫자들이 프로그램 기능적 및 변수 물리적 중요성 양쪽 관점에서 합리적임을 보증하는 것을 돕는다. 읽기-전용 및 읽기-쓰기 인자들은 분리되어야 하며 그들의 입출력은 검사되어야 한다. 함수들은 모든 인자들을 읽기-전용으로 다루어야 한다. 문자 상수는 쓰기-가능해서는 안 된다. 이것은 우연한 겹쳐 쓰기 또는 변수의 잘못된 사용 탐지를 돕는다. 오류 허용 소프트웨어는 그것의 환경에서 또는 명목적인 범위를 넘는 사용 또는 기대되는 조건에서 고장을 예상하고 설계되었다. 기법들은 다음을 포함한다.

- 입력 변수들과 물리적 중요성을 갖는 중간 변수들은 타당성에 대해서 검사되어야 한다.
- 출력 변수들의 영향은 되도록 관련된 시스템 상태 변경의 직접적인 관찰에 의해서 검사되어야 한다.
- 소프트웨어는 그것의 구성을 검사해야 한다. 이것은 기대되는 하드웨어의 존재 및 접근성 모두를 포함할 수 있으며 또한 소프트웨어 자체로 완전하다. 이것은 특별히 유지보수 절차 후에 무결성 유지를 위해 중요하다.

제어 흐름 순서 검사 같은 몇몇 방어적 프로그래밍 기법들은 또한 외부적인 고장을 극복한다.

B-15. 코딩 표준 및 형식 가이드

○ 목적

설계 문서와 생산된 코드의 일정한 구조를 보장하고 일관된 프로그래밍을 강제하고 오류들을 회피하는 표준 설계 방법을 강제함

○ 설명

코딩 표준들은 주어진 프로그래밍 언어에 대해서 그 언어를 사용할 때 만들어질 수 있는 잠재적인 고장들을 회피하기 위한 규칙들과 제약들이다. 코딩 표준 내용은 다음을 포함할 수 있다.

- 코딩 표준 변경을 위한 절차
- 잠재적인 고장의 분석 및 권장하는 처리
- 고장을 회피하기 위한 제약들
- 이식성

형식 가이드라인들은 형식 및 이름 명명과 같은 이슈들을 다룬다. 그리고 비록 그것이 매우 주관적이 될 수 있을지라도, 어떤 형식이든 당신의 코드의 가독성에 영향을 주는 것 이상이다. 프로젝트에 대한 공통적이고 일관된 형식의 수립은 한명 이상의 프로그래머에 의해서 개발된 코드의 이해와 유지보수를 쉽게 만들 것이다. 그리고 여러 명의 사람들이 동일한 프로그램의 개발에서 협력하는 것을 더욱 쉽게 만들 수 있다.

B-16. 다양화 프로그래밍

○ 목적

시스템의 안전 치명적인 고장을 방지하기 위하여 그리고 높은 신뢰성을 위한 연속적인 운영을 위하여 프로그램의 실행 동안 남아있는 소프트웨어 설계 결함을 탐지하고 가린다.

○ 설명

다양화 프로그래밍에서 주어진 프로그래밍 명세서는 다른 방법으로 N 번 구현된다. 동일한 입력값들이 N 가지 버전에 주어지고, N 가지 버전에서 생산된 결과들이 비교된다. 만약 그 결과가 타당하다고 간주된다면, 그 결과는 컴퓨터 출력으로 보내진다. N 가지 버전들은 각각의 컴퓨터에서 병렬적으로 실행할 수 있거나, 대신 모든 버전들이 동일한 컴퓨터에서 실행되고 결과는 내부 투표에 맡긴다. 다른 투표 전략들이 응용프로그램 요구사항에 따라 N 가지 버전에 대해서 사용될 수 있다.

- 만약 그 시스템이 안전한 상태를 가지면, 완전한 의견일치(N 가지 버전이 모두 동의)를 요청하는 것이 실현 가능하고 그렇지 않으면 고장에 대비한 출력값이 사용된다. 간단한 보호 시스템들에 대해서 투표는 안전한 방향으로 치우쳐질 수 있다. 이 경우 만약 어느 한쪽 버전이 보호를 요청하면 안전한 활동이 보호하게 된다. 이 접근방법은 일반적으로 오직 두 버전(N = 2)만 사용한다.
- 안전한 상태를 갖지 않는 시스템에 대해서, 다수결 전략들이 사용될 수 있다. 집단 동의가 없는 경우에 대해서 올바른 값을 선택하는 기회를 최대화하기 위하여 확률적 접근방법들이 사용될 수 있다. 예를 들어 중간 값을 취하고, 의견일치가 돌아올 때까지 출력을 임시로 멈추기 등.

이 기술은 남아 있는 소프트웨어 설계 오류를 제거하지는 않지만, 설계 오류들이 안전성에 영향을 줄 수 있기 전에 탐지하고 가리는 방책을 제공한다. 불행하게도, 실험과 분석 연구들은 N 버전 프로그래밍이 항상 원하는 것처럼 효과적이지는 않다는 것을 보여준다. 다른 알고리즘들이 사용되더라도, 다양한 소프트웨어 버전들이 동일한 입력에서 너무 자주 실패한다.

N 버전 프로그래밍에 대한 두 가지 대안은 설계 다양성과 기능적 다양성이다. 설계 다양성은 각각 다른 방법으로 설계된 그러나 동일한 기능을 구현한 여러 개의 컴포넌트들의 사용을 포함한다. 기능적 다양성은 동일한 문제의 해결을 기능적으로 다른 방법으로 해결하는 것을 포함한다. 접근방법을 고려하지 않고, 다양성의 수준을 평가하는 효과적인 방법이 현재는 없다.

B-17. 동적 재구성

○ 목적

내부적인 고장에도 불구하고 시스템 기능성을 유지

○ 설명

시스템의 논리적 구조는 시스템의 사용 가능한 자원의 부분집합으로 맵핑될 수 있어야 한다. 그 구조는 물리적인 자원의 고장을 감지할 수 있는 능력과 그리고 나서 논리적 구조를 기능이 남은 제한된 자원들로 되돌리게 재배치하는 능력이 필요하다. 이 개념은 전통적으로 장애가 발생한 하드웨어 유닛 로부터의 복구에 제한적으로 사용되었을지라도, 소프트웨어가 재시도 할 수 있거나 또는 고장을 격리시키는 충분히 이중화된 데이터가 있는 충분한 '실행시간 이중화' 가 있다면 이 개념은 또한 장애가 발생한 소프트웨어 유닛에도 사용될 수 있다.

전통적으로 하드웨어에 적용되었지만, 이 기술은 소프트웨어에 그리고 전체 시스템에 적용을 위해 개발되고 있는 중이다. 이 기술은 첫 번째 시스템 설계 단계에서 고려되어야 한다.

B-18. 동등 클래스 및 입력 분할 테스팅

○ 목적

최소한의 테스트 데이터를 사용하여 소프트웨어를 적절하게 테스트함. 그 테스트 데이터는 소프트웨어를 시험하는데 필요한 입력 범위의 분할된 부분을 선택함으로서 얻는다.

○ 설명

이 테스팅 전략은 입력 값 범위의 분할을 결정하는 입력값의 동등 관계에 기반을 둔다. 테스트 케이스는 이 분할된 부분의 모든 부분집합을 대신하는 목적으로 선택된다. 각각의 동등 클래스로부터 적어도 하나의 테스트 케이스를 가져온다. 입력 분할에 대한 두개의 기본적인 가능성이 있다.

- 동등 클래스는 명세에서 정의될 수 있다. 명세의 해석은 예를 들어 선택된 값들은 같은 방식으로 취급된다는 입력 지향적이거나 또는 예를 들어 값들의 집합이 같은 기능적 결과로 이끄는 출력 지향적 둘 중에 하나이어야 한다.
- 동등 클래스는 프로그램의 내부적 구조에서 정의될 수 있다. 이 경우 동등 클래스 결과는 프로그램의 정적 분석으로부터 결정된다. 예를 들어 동일한 경로로 실행을 이끄는 값의 집합

B-19. 오류 검출 및 정정 코드

○ 목적

민감한 정보에서 오류를 찾고 정정함

○ 설명

n 비트 정보에 대하여 오류를 찾고 정정할 수 있는 k 비트의 코드 블록이 생성된다. 다른 유형의 코드는 다음과 같다.

- 해밍 코드
- 순환 코드
- 다항식 코드
- 해쉬 코드
- 암호화 코드

B-20. 오류 추측

○ 목적

일반적인 프로그래밍 오류를 제거

○ 설명

테스팅 경험과 테스트 중인 시스템에 대한 지식과 호기심이 결합된 직관은 어떤 분류되지 않은 테스트 케이스를 설계된 테스트 케이스 세트에 추가할 수 있다. 특별한 값 또는 값들의 조합이 오류를 쉽게 발생할 수 있다. 몇몇 흥미로운 테스트 케이스는 검사 체크리스트로부터 파생될 수 있다. 이것은 또한 시스템이 충분히 강건한지 여부도 고려할 수 있다. 전면 패널의 버튼이 너무 빠르게 또는 너무 자주 눌러질 수 있는가? 만약 두 버튼이 동시에 눌러지면 무슨 일이 발생하는가?

B-21. 오류 삽입

○ 목적

테스트 케이스의 세트가 충분한지 확인

○ 설명

일부 알려진 오류 유형들이 프로그램에 삽입되고 그 프로그램이 시험조건 하에서 테스트 케이스와 함께 실행된다. 단지 삽입된 오류들의 일부가 발견된다면 테스트 케이스 세트가 충분하지 않은 것이다. 발견된 삽입된 오류들과 전체 삽입된 오류들의 비율이 전체 오류 중 발견된 실제 오류 비율의 추정이다. 이렇게 하면 남은 오류의 개수와 남은 테스트 노력을 추정할 수 있습니다.

$$\frac{\text{발견된 삽입된 오류들}}{\text{삽입된 전체 오류들 개수}} = \frac{\text{발견된 실제 오류들}}{\text{실제 전체 오류들의 개수}}$$

모든 삽입된 오류들의 검출은 테스트 케이스 세트가 충분하거나 또는 삽입된 오류들이 발견하기 너무 쉬운 것을 나타낸다. 이 방법의 한계들은 어떠한 사용가능한 결과를 얻기 위하여, 오류 형식뿐만 아니라 삽입 위치들은 실제 오류의 통계적 분표를 반영해야 한다. 만약 오류 삽입이 사용된다면 모든 오류의 위치는 기록되어야 하며 확인자는 소프트웨어가 출시되기 전에 모든 삽입된 오류들이 제거됨을 확인해야 한다.

B-22. 이벤트 트리 분석

○ 목적

도식화된 형식으로 초기 이벤트 이후 시스템에서 만들 수 있는 이벤트의 순서를 모델링하고, 그것에 의해 심각한 결과가 발생할 수 있음을 나타냄

○ 설명

다이어그램의 가장 위에는 분석의 목표인 초기 이벤트를 따르는 관련된 순차적 조건들을 작성한다. 초기 이벤트의 시작 하에서 순서에서 첫 번째 조건으로 선을 그린다. 거기에서 그 다이어그램은 미래의 전개가 그 조건에 의존함을 기술하는 '예', '아니오' 가지로 분기된다.

각각의 가지에 대하여 비슷한 방법으로 다음 조건에 대해서 계속한다. 그러나 모든 조건들이 모든 가지들과 관련된 것은 아니다. 순서의 끝까지 계속하고 이렇게 구축된 트리의 각가지들이 가능한 결과를 나타낸다. 이벤트 트리는 이벤트 순서상에서 확률과 조건의 개수에 따라 다양한 결과의 확률을 계산하는데 사용될 수 있다.

B-23. 페이건 정밀 검사

○ 목적

프로그램 개발의 모든 단계에서 오류를 밝히기 위함

○ 설명

오류 및 누락을 찾기 위한 목적으로 품질 보증 문서에 대한 '정형적' 감사. 정밀 검사 과정은 5 단계로 구성된다. 계획, 준비, 정밀 검사, 재작업 및 후속조치. 이들 각각의 단계는 자체적인 분리된 목표를 갖고 있다. 전체 시스템 개발(명세, 설계, 코딩 및 테스팅)은 정밀검사를 해야 한다.

B-24. 고장 단정 프로그래밍

○ 목적

소프트웨어 프로그램의 실행 중 남아 있는 결함을 탐지

○ 설명

단정 프로그래밍 방법은 사전 조건(일련의 명령문이 실행되기 전에, 초기 조건들의 타당성을 검사) 및 사후 조건(일련의 명령문의 실행 후 결과 검사) 의 검사에 대한 아이디어를 따른다. 만약 사전 조건 또는 사후 조건이 충족되지 않으면, 처리 과정은 오류와 함께 멈춘다. 예를 들어 아래와 같다.

```
assert <pre-condition>;
    action 1;
      :
      :
    action x;
assert <post-condition>;
```

B-25. SEEA - 소프트웨어 오류 영향 분석

○ 목적

소프트웨어 컴포넌트, 컴포넌트의 임계를 식별하기 위하여; 소프트웨어 오류 탐지 및 소프트웨어 견고성 향상을 위한 방법의 제안, 다양한 소프트웨어 컴포넌트에 필요한 타당성 확인의 양을 평가

○ 설명

1) 중요 소프트웨어 컴포넌트 식별

명세로부터, 각각의 소프트웨어 컴포넌트에 대해 필요한 분석의 깊이 결정(단일 명령

줄, 여러 줄의 명령, 컴포넌트 등)

2) 소프트웨어 오류 분석

이 단계의 결과는 다음 정보를 나열하는 표이다.

- 컴포넌트 이름
- 고려된 오류
- 모듈 수준에서 오류의 결과
- 시스템 수준에서 결과
- 위반된 안전 기준
- 오류 임계
- 제안된 오류 탐지 방법
- 만약 탐지 방법이 구현되면 위반된 기준
- 만약 탐지 방법이 구현되면 남은 임계

3) 합성

합성은 남은 안전하지 않은 시나리오 및 주어진 각 모듈의 중요성에 필요한 확인 작업을 식별한다. SEEA는 독립적인 팀에 의해서 수행된 심층 분석으로 강력한 버그 찾기 방법이다.

B-26. 결함 검출과 진단

○ 목적

시스템에서 고장을 유발할 수 있는 결함을 탐지, 고장의 결과를 최소화하기 대응책의 기초를 제공

○ 설명

결함 탐지는 시스템에 대한 오류 상태(이전에 설명한 것처럼, 검사되는 (하위)시스템 내의 결함에 의해서 원인이 되는)를 검사하는 과정이다. 결함 탐지의 주된 목표는 잘못된 결과의 영향을 금지하는 것이다. 올바른 결과를 제공하거나 전혀 결과를 제공하지 않는 시스템을 '자가 점검' 이라고 한다.

결함 탐지는 이중화(주로 하드웨어 결함 탐지) 및 다양성(소프트웨어 결함)의 원칙에 기반을 둔다. 결과의 정확성을 결정하기 위하여 어떤 종류의 투표가 필요하다. 적용 가능한 특별한 방법들은 단정 프로그래밍, N-버전 프로그래밍, 안전성 백 기법 그리고 하드웨어 단계에서는 센서 도입, 제어 루프, 오류 검사 코드 등이 있다.

결함 탐지는 값의 범위 또는 여러 가지 수준에서 시간의 범위, 특히 물리적(온도, 전압 등), 논리적(오류 탐지 코드), 기능적(단정) 또는 외부적 수준(타당성 검사) 검사함으로

써 달성할 수 있다. 이러한 검사의 결과는 고장 추적을 허용하도록 영향 받는 데이터와 함께 저장되고 연관될 수 있다. 복잡한 시스템은 하위 시스템으로 구성된다. 결함 탐지의 효율성, 진단 및 결함 보상은 결함의 전파에 영향을 주는 하위 시스템 간의 상호작용의 복잡성에 의존한다. 결함 진단은 식별할 수 있는 가장 작은 하위 시스템을 분리한다. 하위 시스템이 작을수록

더욱 상세한 결함(오류 상태의 식별)의 진단이 가능하다.

B-27. 유한 상태 기계 / 상태 전이 다이어그램

○ 목적

시스템의 제어구조를 정의하거나 구현하기 위함

○ 설명

많은 시스템들은 상태, 입력, 동작 관점에서 정의될 수 있다. 그러므로 상태 S1에서 입력 I를 수신하면 시스템은 동작 A를 수행하고 상태 S2로 이동할 수 있다. 모든 상태의 모든 입력에 대해서 시스템의 동작을 정의함으로써 시스템을 완전하게 정의할 수 있다.

이 시스템의 결과 모델을 유한 상태 기계(FSM)이라고 한다. 이것은 시스템이 한 상태에서 다른 상태로 이동하는 소위 상태 천이 다이어그램이라 불리는 그림 또는 차원이 상태와 입력이고 행렬의 셀은 동작과 주어진 상태에서 입력을 수신한 결과인 새로운 상태를 포함하는 행렬 이다. 시스템이 복잡하거나 자연스러운 구조를 갖는 경우 계층적인 유한 상태 기계로 반영될 수 있다. 유한 상태 기계로 표현된 명세 또는 설계는 완전성(시스템이 모든 상태의 모든 입력에 대하여 동작 및 새로운 상태를 가짐), 일관성(각각의 상태/입력 쌍에 대해서는 오직 하나의 상태 변경이 정의됨) 그리고 도달 가능성(입력의 순서에 의해서 하나의 상태에서 다른 상태로 갈 수 있는 가능성이 있는지 여부)을 검사할 수 있다. 이들은 치명적인 시스템에서 중요한 속성들이고 검사되어야 한다. 이들의 검사를 지원하는 도구들은 쉽게 작성된다. 유한상태기계 구현을 검증하거나 유한 상태 기계 모델을 애니메이션하기 위한 테스트 케이스의 자동생성을 허용하는 알고리즘 또한 존재한다.

복잡한 시스템의 행위의 설명을 향상하기 위하여 기본적인 FSM의 몇 가지 확장이 고안되었다. 상태차트는 계층 구조, 합성(병렬처리), 레벨 간의 전이, 이력 상태 등을 추가한다. 특별히 내부적 상태의 중첩과 전이가 필요에 따라 내부적 상태를 표시하거나 숨길 수 있는 유용한 기능이다. 상태차트는 UML(통합 모델링 언어)의 일부이며 많은 상용 도구들이 지원한다.

B-28. 정형 기법들

B-28.1 일반

○ 목적

“정형 기법들“은 소프트웨어 및 하드웨어 시스템의 명세, 설계 및 검증을 위한 수학적으로 엄격한 기술 및 도구들을 말한다.

○ 설명

“수학적으로 엄격한“은 정형 기법이 사용된 명세가 수학적인 논리에서 문법에 맞는 문장들이고 정형 검증은 그 논리에서 엄격한 추론이다. (즉, 각각의 단계는 추론 규칙을 따르므로 기계적 처리에 의해서 점검될 수 있다)

정형 기법의 가치는 디지털 설계(하드웨어 또는 소프트웨어)의 전체 상태 공간을 기호적으로 검사하고 모든 가능한 입력에 대하여 정확성 또는 참인 안전성 속성을 확립하는 수단을 제공하는 것이다. 그러나 실제 시스템의 엄청난 복잡성 때문에 오늘날 실제로는 (안전상 중요한 시스템의 중요한 컴포넌트를 제외하고) 거의 사용되지 않는다.

실제 시스템과 관련된 천문학적인 규모의 상태 공간을 극복하기 위하여 몇 가지 접근법이 사용된다.

- 정형 기법을 대부분의 상세함 들이 추상화되어 사라진 요구사항 및 상위 수준 설계에 적용한다.
- 정형 기법을 오직 가장 중요한 컴포넌트에만 적용한다.
- 변수들이 이산적이고 범위가 심하게 축소된 소프트웨어 및 하드웨어의 모델을 분석한다.
- 시스템 모델을 “분할하고 정복“이 가능한 계층적인 방식으로 분석한다.
- 가능한 한 많이 검증을 자동화한다.

수학적인 논리의 사용이 정형 기법의 학문을 가로지르는 단일화된 주제일지라도 가장 좋은 “정형 기법“은 없다. 각 응용 분야마다 서로 다른 모델링 기법과 다양한 증명 방법을 요구한다.

게다가 특정한 응용 분야 내에서 조차 생명 주기의 여러 단계들은 서로 다른 도구 및 기술로 가장 잘 사용될 수 있다. 예를 들어 정리 증명기는 고속 푸리에 변환 회로의 레지스터 전송 레벨 설명의 정확성을 분석하는데 가장 잘 사용되는 반면, 대수적 유도 기법은 게이트 수준의 설계로 설계 상세화의 정확성을 분석하는데 가장 잘 사용될 수 있다. 그러므로 전 세계적으로 수많은 정형 기법들이 개발되고 있다. 정형 기법의 몇 가지 예들이 이 참고문헌의 하위절 다음에 설명된다. 여기의 예제 목록은 완전한 것이 아니다.

설명되는 정형 기법들은 CSP, CCS, HOL, LOTOS, OBJ, 시제 논리, VDM, Z 기법, B 기법 및 모델 검증이다.

B-28.2 CSP - Communicating Sequential Processes

○ 목적

CSP는 동시성 소프트웨어 시스템 즉, 동시에 동작하는 통신 프로세스의 시스템의 명세를 위한 기술이다.

○ 설명

CSP는 프로세스 시스템의 명세에 대한 언어를 제공하고 프로세스의 구현이 명세를 만족하는지 검증을 위한 증명을 제공한다. (추적으로 설명되는 - 허용가능한 일련의 사건)

시스템은 독립적인 프로세스의 네트워크로 모델링된다. 각 프로세스는 모든 가능한 행위의 관점에서 기술된다. 시스템은 프로세스들을 순차적으로 또는 병렬로 구성함으로써 모델링된다. 프로세스들은 채널을 통해서 통신(동기화 또는 데이터 교환)할 수 있고 통신은 양쪽 프로세스가 준비되었을 때만 발생한다. 이벤트들의 상대적인 타이밍은 모델링 될 수 있다. CSP의 이론은 Inmos 트랜스퓨터의 구조에 직접적으로 통합되었다[11]. 그리고 occam 언어[12] 는 CSP로 명세된 시스템을 트랜스퓨터의 네트워크상에 직접적으로 구현하도록 허용한다.

B-28.3 CCS - Calculus of Communicating Systems

○ 목적

CCS는 동시적이고 통신하는 프로세스의 시스템의 행위에 대하여 설명하고 추론하는 방법이다.

○ 설명

CSP와 비슷하게, CCS는 시스템의 행위와 관련된 수학적 계산법이다. 시스템 설계는 순차적 또는 병렬로 동작하는 독립된 프로세스들의 네트워크로써 모델링된다. 프로세스들은 포트(CSP의 채널과 유사)를 통하여 통신할 수 있고 통신은 양쪽의 프로세스가 준비되었을 때만 수행된다. 비결정론이 모델링될 수 있다. 전체 시스템의 고수준 추상화 설명으로부터 시작하여 (추적) 시스템의 단계별 상세화는 전체 시스템이 요구하는 전체

11) Inmos는 80년대에 트랜스퓨터라고 하는 병렬처리를 위한 혁신적인 마이크로 프로세스 아키텍처를 제작한 영국의 반도체 회사였다. 나중에 Inmos는 SGS-Tomson의 일부가 되고 그리고 STMicroelectronics가 되었다.

12) occam은 Occam's Razor 명성의 William Ockham의 이름을 따서 만든 동시성 프로그래밍 언어이다. occam은 Inmos 트랜스퓨터의 기본 프로그래밍 언어이다.

행위를 하는 프로세스 통신의 구성으로 수행이 가능하다. 마찬가지로 프로세스들을 결합하고 합성 규칙과 관련된 추론 규칙을 사용하여 결과 시스템의 속성들을 추론하는 상향식으로 작업이 가능하다.

B-28.4 HOL - Higher Order Logic

○ 목적

HOL은 하드웨어 명세와 검증을 위한 기초로써 의도된 정형 언어 이다.

○ 설명

HOL(Higher Order Logic)은 캠브리지 컴퓨터 연구소에서 개발된 특정 논리 표기법과 그 논리의 기계 지원 시스템 모두를 말한다. 논리 표기법은 처치의 단순 타입 이론으로부터 취해지며 기계 지원 시스템은 LCF(Logic of Computable Functions) 시스템을 기반으로 한다.

B-28.5 LOTOS

○ 목적

LOTOS는 동시적이고 통신하는 프로세스의 시스템 행위에 대한 설명 및 추론을 위한 수단이다.

○ 설명

LOTOS(Language for Temporal Ordering Specification)은 연관된 대수 CSP와 CIRCAL(Circuit Calculus)로부터 추가적인 기능을 갖는 CCS를 기반으로 한다. LOTOS는 추상 데이터 타입 언어인 ACT ONE에 기반을 둔 두 번째 컴포넌트와 결합하여 데이터 구조 및 값 표현식의 처리에서 CCS의 약점을 극복한다. 그러나 LOTOS의 프로세스 정의 컴포넌트는 추상 데이터 타입의 설명에 대하여 다른 수학적 형식을 사용할 수 있다.

B-28.6 OBJ

○ 목적

구현에 앞서 사용자 의견과 시스템 확인을 통하여 정확한 시스템 명세를 제공

○ 설명

OBJ는 대수적 명세 언어이다. 사용자는 대수 방정식의 관점에서 요구사항을 명세한다. 시스템의 행위적 또는 구조적 관점이 추상 데이터 타입(ADT)에 대해 동작하는 연산의

관점에서 명세 된다. ADT는 연산자 행위는 보이는 반면 상세한 구현은 '숨겨진' Ada[13] 패키지와 같다. OBJ 명세와 후속 단계별 구현은 다른 정형 접근법처럼 동일한 정형 증명 기법을 적용할 수 있다. 게다가 OBJ 명세의 구조적 관점은 기계에서 실행 가능하므로, 명세 자체로부터 시스템 확인을 수행하는 것이 간단하다. 실행은 본질적으로 방정식 치환(re-writing)에 의한 함수의 평가로서, 특정한 출력값이 얻어질 때까지 계속된다. 이 실행 가능성은 의도된 시스템의 최종 사용자가 기본적인 정형 명세 기법에 익숙하지 않고도 명세 단계에서 최종 시스템의 'view'를 얻을 수 있도록 한다. 다른 모든 ADT 기술처럼, OBJ는 오직 순차 시스템 또는 동시적 시스템의 순차적 측변에만 적용 가능하다. OBJ는 소형 및 대형 산업 응용분야 모두의 명세로 널리 사용되었다.

B-28.7 시제 논리

○ 목적

안전성과 운영 요구사항의 직접적 표현 그리고 이러한 특성들이 후속 구현 단계에서 보존된다는 정형 증명

○ 설명

표준 일차 술어 논리에는 시간의 개념이 없다. 시제 논리는 시제 연산자(예: '앞으로', '언젠가는')를 추가하여 일차 술어 논리를 확장한다. 이러한 연산자들은 시스템에 대한 단정들을 검사하는데 사용될 수 있다. 예를 들어 안전 속성은 '앞으로' 유지해야 할 수도 있지만, 다른 원하는 시스템 상태는 '언젠가는' 어떤 다른 시작 상태에서 달성되어야 할 수도 있다. 시제 공식은 일련의 상태(행위)로 해석된다. '상태'를 구성하는 것은 선택한 설명의 수준에 달려있다. 시제논리는 전체 시스템, 시스템 컴포넌트 또는 컴퓨터 시스템을 언급할 수 있다. 정량적 시간 간격 및 제약조건들은 시제 논리에서 명시적으로 다룰 수 없다. 절대적인 타이밍은 상태 정의의 일부로 추가적인 시간 상태를 생성함으로써 다룰 수 있다.

B-28.8 VDM - Vienna Development Method

○ 목적

순차적 프로그램의 체계적인 명세 및 구현

○ 설명

VDM은 수학적 기반의 명세 기법이며 명세 관점에서 정확성을 증명하기 위한 방법으로

13) Ada는 Pascal 및 다른 언어로부터 확장된 구조적, 정적 타입, 명령형, 광범위한 스펙트럼 및 객체 지향식 고수준 컴퓨터 프로그래밍 언어이다.

구현을 정제하는 기술이다. 명세 기법은 모델 기반으로 시스템 상태가 불변식(술어식)으로 정의된 집합이론 구조로 모델링 되고 그 상태에 대한 연산은 시스템 상태에 대한 사전, 사후 조건을 맹세하는 것으로 모델링된다. 연산은 시스템 불변식을 보존하는 것으로 증명될 수 있다.

명세의 구현은 대상 언어의 데이터 구조 관점에서 시스템의 상태를 구체화 하는 것으로, 그리고 대상 언어의 프로그램 관점에서 연산을 정제하는 것으로 수행된다. 구체화 및 정제 단계는 명세의 정확성을 입증하는 증명 의무를 낳는다. 이 의무들이 수행되는지 여부는 설계자의 선택이다. VDM은 주로 명세 단계에서 사용되지만 소스코드로 이끄는 설계 및 구현 단계에서 사용될 수 있다. VDM은 오직 순차적 프로그램 또는 동시성 시스템에서 순차적 프로세스에만 적용될 수 있다.

B-28.9 Z 방법

○ 목적

Z는 순차적 시스템에 대한 명세 언어 표기법이며 설계자가 Z 명세에서 실행 가능한 알고리즘으로 진행하는데 명세에 관한 실행 가능한 알고리즘의 정확성을 증명할 수 있도록 하는 설계 기법이다. Z는 주로 명세 단계에서 사용되지만 명세에서 설계 및 구현으로 가는 방법이 고안되었다. Z는 데이터 지향, 순차적 시스템의 개발에 가장 적합하다.

○ 설명

VDM과 마찬가지로 이 명세 기법은 시스템 상태를 불변식(술어)으로 정의된 집합 이론 구조의 관점에서 모델링하고 해당 상태의 연산은 시스템 상태의 관점에서 사전, 사후 조건을 맹세하는 것으로 모델링한다. 연산은 시스템의 불변식을 보존함으로써 연산의 일관성을 보여주는 것이 증명될 수 있다. 명세의 정형적인 부분은 정제를 통하여 명세의 구조화를 허용하는 스키마로 나누어진다. 일반적으로 Z 명세는 정형적인 Z 와 비정형적인 자연어 설명문이 섞여 있다.

(정형적인 글은 너무 간결해서 쉽게 읽을 수 없어서 종종 목적을 설명할 필요가 있지만 비정형적 자연어는 쉽게 모호하고 부정확해지 수 있다) VDM과는 달리 Z는 완전한 방법이 아닌 표기법이다. 그러나 Z와 결합하여 사용할 수 있는 관련된(B라고 함) 기법이 개발되었다.

B-28.10 B 방법

○ 목적

VDM과 마찬가지로 B 방법의 목적은 정형적으로 시스템 또는 소프트웨어를 모델링하고 시스템 또는 소프트웨어의 행위가 모델링 중에 명시적으로 만들어진 속성들을 존중한다는 것을 증명하는 것이다.

○ 설명

B 모델링은 집합 이론에서 수학적 항목을 요청한다. 한편 불변식(즉, 술어)은 모델의 정적 속성을 정의한다. 반면에 연산은 사후 조건을 설정하여 모델의 동적 행위를 정의한다. 복잡한 시스템 또는 소프트웨어의 명세는 모델을 다른 의미를 갖는 연결선으로 함께 묶은 '기계들'로 분해함으로써 가능하다.

B 정형론을 사용하는 모델링의 두 가지 주요 분류를 구별할 수 있다.

- 전자(역사적으로는 첫 번째)는 소프트웨어 개발을 목표로 한다. 이 경우 목표는 명세를 중시하는 프로그램을 만드는 것이다. 모델은 추상 기계(반드시 결정론적일 필요는 없음)와 이 기계의 단계별 상세화로 구성되어 'B0' 라고 하는 의사 코드로 작성된 결정론적 구현을 유도한다. 그리고 이 의사 코드는 자동적으로 목적 프로그래밍 언어로 변환될 수 있다.
- 후자는 시스템 모델링을 목표로 하며 이 경우 "이벤트 B"에 대하여 말한다. "이벤트 B"의 목표는 모호하지 않고 일관성 있게 명시적인 속성을 충족하는 시스템을 명세하는 것이다. 모델은 시스템 자체와 환경을 고려한다.

시스템의 역동성은 '이벤트'로 모델링되며 상세화 기법은 시스템과 환경간의 정확한 상호작용을 위하여 사용된다. 증명 의무(B 정형 모델로 부터 추출된 가정으로부터 정형적으로 증명되는 논리적 단언들) 세트는 자동적으로 생성된다. 이 증명 의무는 다음을 보증한다.

- 모델의 정적 및 동적 속성을 충족하는 데이터의 존재,
- 연산(모델의 동적 행위)은 불변식을 존중하고,
- 데이터와 연산(그리고 필요하다면 B0 의사 코드)의 정제는 추상 기계로 작성된 명세와 모순되지 않음,
- 각 연산은 사전 조건의 문맥 내에서 호출된다,
- 소프트웨어 모델링의 경우 프로그램은 종료된다. (특히, 각 루프가 종료됨)
- 다른 증명 의무, 예를 들어 정수 오버플로 또는 언더플로우 검증 또한 생성된다.

B-28.11 모델 검증

○ 목적

주어진 시스템 모델에서, 이 모델이 주어진 명세를 만족하는지 여부를 자동적으로 시험한다.

○ 설명

모델 검증은 주어진 구조가 주어진 논리식들의 모델인지 여부를 검사하는 과정이다. 개념은 일반적이고 모든 종류의 논리 및 적절한 구조에 적용된다. 간단한 모델 검증 문제는 명제 논리에서 주어진 논리식들이 주어진 구조에 의해서 만족되는지 여부를 시험하는 것이다.

모델 검증 기법의 중요한 클래스는 정형 시스템을 알고리즘 적으로 검증하기 위해 개발되었다. 이것은 종종 하드웨어 또는 소프트웨어 설계로부터 파생된 구조가 정형 명세(일반적으로 시제 논리식)를 만족하는지 검증함으로써 달성된다.

모델 검증은 주로 하드웨어 설계에 적용된다. 소프트웨어의 경우 결정불가능성(계산 논리 참조) 때문에 이 접근법은 완전하게 알고리즘 적이 될 수 없다. 일반적으로 주어진 속성을 증명하거나 반증하는데 실패할 수 있다. 구조는 일반적으로 산업용 하드웨어 기술 언어 또는 특수 목적 언어의 소스코드 설명으로 제공된다. 이러한 프로그램은 유한 상태 기계(즉 노드와 에지로 구성된 방향성 그래프)와 일치한다.

원자적 명제의 집합은 각 노드와 관련되며 일반적으로 어떤 메모리 원소가 하나인지 분명히 말한다. 노드들은 시스템의 상태를 나타내고 에지는 상태를 변경할 수 있는 가능한 전이를 나타내는 반면 원자적 명제들은 실행의 시점에서 유지되는 기본 속성을 나타낸다.

공식적으로 문제는 다음과 같이 말할 수 있다: 주어진 시제 논리식 p로 표현된 원하는 속성과 초기 상태 s를 갖는 구조 M 에 대하여 결정한다. 만약 하드웨어에서처럼 M이 유한하면 모델 검증은 그래프 탐색으로 축소된다.

B-29. 정형 증명

○ 목적

이론적 및 수학적 모델과 규칙을 사용하여 실행하지 않고 프로그램 또는 모델의 정확성을 증명할 수 있다

○ 설명

프로그램의 다양한 위치에서 여러 가지 단언들이 명시되어 있고 이 단언들은 프로그램

의 다양한 경로에서 사전 및 사후 조건으로써 사용된다. 증명은 프로그램이 논리적 규칙의 집합에 따라 사전조건을 사후조건으로 바꾸고 프로그램이 종료하는 것을 보여주는 것으로 구성된다.

CCS, CSP, HOL, LOTOS, OBJ, 시제 논리, VDM 및 Z와 같은 몇 가지 정형 기법들이 이 부록에서 설명된다. 이들의 설명은 B-28 절에서 찾을 수 있다.

B-30. 전 방향 복구

○ 목적

하나 이상의 결함이 있는 경우 올바른 기능 작동을 제공

○ 설명

만약 결함이 감지되면 시스템의 현재 상태가 얼마 후 일관되게 되는 상태를 얻기 위하여 조작된다. 이 개념은 특별히 작은 데이터베이스와 내부 상태의 빠른 변화율을 갖는 실시간 시스템에 적합하다. 적어도 시스템 상태의 일부는 환경에 부과될 수 있고 오직 시스템의 상태들 중 일부만 환경에 영향을 받는(강제되는)것으로 가정한다.

B-31. 우아한 저하

○ 목적

고장인 경우에도 덜 중요한 기능들은 버리고 더 중요한 시스템 기능을 사용가능하도록 유지

○ 설명

이 기술은 시스템에서 수행될 다양한 기능에 우선순위를 부여한다. 그런 다음 설계는 모든 시스템 기능을 수행할 자원이 있으면 더 높은 우선순위 기능들이 낮은 순위의 기능보다 우선하여 수행되는 것을 보장한다. 예를 들어 오류 및 이벤트 기록 기능은 시스템 제어 기능보다 낮은 우선순위를 가질 수 있다. 이 경우 오류 기록과 관련된 하드웨어가 실패해도 시스템 제어는 계속된다. 다른 예는 신호 시스템으로 제어 센터와 통신이 끊어진 경우 자동적으로 지역의 라인사이드 장비가 우선순위가 가장 높은 트래픽에 의해 취해진 방향으로 가능한 경로를 설정한다. 이것은 우아한 저하가 될 수 있다. 왜냐하면 우선하는 경로의 열차는 제어 센터와 통신이 끊어져 영향을 받는 지역을 통과할 수 있으나, 궤도를 바꾸는 것 같은 다른 이동은 불가능하기 때문이다.

B-32. 영향 분석

○ 목적

소프트웨어의 변경 또는 개선이 그 소프트웨어의 다른 컴포넌트뿐만 아니라 다른 시스템에도 영향이 있는지 확인

○ 설명

소프트웨어에 수정 또는 개선이 수행되기에 앞서 소프트웨어에 대한 그 수정 또는 개선의 영향을 파악하고 영향 받는 소프트웨어 시스템과 컴포넌트를 식별하기 위한 분석이 수행되어야 한다. 분석이 완료된 후 소프트웨어 시스템의 재검증에 관한 결정이 필요하다. 이것은 영향 받는 컴포넌트의 개수, 영향 받는 컴포넌트의 중요도와 변경의 성격에 의존적이다. 가능한 결정들은 다음과 같다.

- 변경된 컴포넌트만 재검증
- 식별된 모든 영향 받는 컴포넌트 재검증; 그리고
- 전체 시스템 재검증

B-33. 정보 은닉 / 캡슐화

○ 목적

소프트웨어의 견고함과 유지보수성을 향상

○ 설명

모든 소프트웨어 컴포넌트가 전역적으로 접근 가능한 데이터는 이러한 컴포넌트 중 하나에 의해서 실수 또는 잘못 수정될 수 있다. 이 데이터 구조에 대한 어떤 변경이 코드의 상세한 검사 및 광범위한 수정을 요구할 수 있다.

정보 은닉은 이러한 어려움을 최소화하기 위한 일반적인 접근방법이다. 주요 데이터 구조는 '감춰진' 이며 정의된 접근 프로시저의 세트를 통해서만 조작할 수 있다. 이를 통하여 나머지 소프트웨어의 기능적 행위에 영향 없이 내부 구조를 수정하거나 더 프로시저를 추가할 수 있다. 예를 들어 명명된 디렉토리에는 접근 프로시저인 추가, 삭제 및 검색이 있을 수 있다. 접근 프로시저 및 내부 데이터 구조는 이들 프로시저를 사용하는 나머지 소프트웨어의 논리적 행위에 영향 없이 재작성(예를 들어, 다른 검색 방법 사용 또는 하드 디스크 상에 이름을 저장) 될 수 있다.

B-34. 인터페이스 시험

○ 목적

서브프로그램의 인터페이스가 어떠한 오류도 포함하지 않음을 입증하거나 또는 특정한 응용 소프트웨어에서 고장을 일으키는 오류를 포함하지 않는다는 것을 입증하거나 또는 관련 있을 수 있는 모든 오류들을 탐지

○ 설명

시험의 몇 가지 세부 수준 또는 완벽함이 실현 가능하다. 가장 중요한 수준은 시험이다.

- 모든 인터페이스 변수에 대해 그 변수의 맨 끝 값으로 시험
- 모든 인터페이스 변수에 대해 각 변수는 맨 끝 값으로, 나머지 인터페이스 변수는 정상 값으로 시험
- 각 인터페이스 변수는 범위의 모든 값으로, 나머지 인터페이스 변수는 정상 값으로 시험
- 모든 변수의 모든 값 조합(소규모 인터페이스의 경우만 가능)
- 각 서브루틴의 각 호출과 관련된 명세 시험 조건들에 대해 시험

이러한 시험은 인터페이스에 잘못된 인자 값을 감지하는 단언들을 포함하지 않는 경우 특히 중요하다. 이 시험은 기존 서브루틴의 새로운 구성이 생성된 후에도 중요하다.

B-35. 언어 하위집합

○ 목적

프로그래밍 결함을 넣을 확률을 줄이고 남은 결함을 발견할 확률을 높임

○ 설명

예를 들어 정적 분석 모델을 사용하여 오류가 발생하기 쉽거나 분석하기 어려운 프로그래밍 구조를 식별하기 위해 검사한다. 그런 다음 이러한 구성들을 제외하는 언어 하위집합이 정의된다.

B-36. 실행된 사례 기억

○ 목적

소프트웨어가 허가되지 않은 경로로 실행되면 안전 작동으로 동작시킴

○ 설명

허가된 기간 중에는 각 프로그램의 실행에 대한 모든 상세한 정보가 기록으로 만들어진다. 정상 운영 중에는 각 프로그램 실행은 허가된 실행 집합과 비교된다. 만약 다른 경우 안전 조치가 취해진다.

실행 기록은 개별 결정-결정 경로(DDPath)의 연속 또는 배열, 레코드 또는 볼륨, 또는 두 가지 모두에 대한 개별 접근의 연속일 수 있다. 실행 경로를 저장하는 다른 방법이 가능하다. 해시 코딩 방법은 실행 순서를 하나의 큰 수 또는 일련의 수로 맵핑하는데 사용될 수 있다. 정상 운영 중에는 출력 운영이 발생하기 전에 실행 경로 값이 저장된 경우에 반하여 검사되어야 한다.

한 프로그램에서 결정-결정 경로의 가능한 조합이 매우 크기 때문에 프로그램을 전체적으로 다루는 것은 적합하지 않을 수 있다. 이 경우 이 기술은 컴포넌트 수준에서 적용될 수 있다.

B-37. 측정 기준

○ 목적

소프트웨어 개발 또는 시험 기록이 아닌 소프트웨어 자체의 속성으로부터 프로그램의 속성을 예측

○ 설명

이 모델은 소프트웨어의 일부 구조적 속성을 평가하고 복잡도와 같은 원하는 특성과 관련시킨다. 대부분의 측정 기준을 평가하기 위하여 소프트웨어 도구가 필요하다. 적용할 수 있는 측정 기준 중 일부는 다음과 같다.

- 그래프 이론 복잡도: 이 측정 기준은 트레이드오프를 평가하기 위하여 생명주기 초기에 적용될 수 있으며 프로그램 제어 그래프의 복잡도(사이클로매틱 숫자로 표시)에 기반을 둔다.
- 특정 컴포넌트를 활성화 하는 방법의 개수(접근가능성): 컴포넌트가 더 많이 접근될수록 더 많이 디버깅될 가능성이 높다.
- Halstead 복잡도 측정: 이 측정 기준은 연산자 및 피연산자의 개수를 세어서 프로그램의 길이를 계산한다. 이것은 복잡도를 측정하고 개발자원을 추정한다.
- 컴포넌트 당 입구 및 출구의 개수: 입구/출구의 수를 최소화하는 것이 구조적 설계와 프로그래밍 기법의 중요한 특징이다.

B-38. 모듈 방식

○ 목적

소프트웨어의 복잡도를 제한하기 위하여 소프트웨어를 이해하기 쉬운 작은 부분으로 분해

○ 설명

모듈러 방식 또는 모듈화는 소프트웨어 프로젝트의 설계, 코딩 및 유지보수 단계에 대한 몇 가지 규칙을 포함한다. 이 규칙들은 설계 중에 사용된 설계 방법에 따라 다양하다. 대부분의 방법들은 다음과 같은 규칙을 포함한다.

- 모듈/컴포넌트는 수행해야하는 단일의 잘 정의된 작업 또는 기능을 가져야 한다.
- 모듈/컴포넌트간의 연결은 제한적이고 엄격하게 정의되어야 하며 하나의 모듈/컴포넌트에서 일관성은 강해야 한다.
- 여러 수준의 모듈/컴포넌트를 제공하는 서브프로그램의 모음을 구축해야 한다.
- 서브프로그램은 하나의 입구와 하나의 출구만을 가져야 한다.
- 모듈/컴포넌트는 다른 모듈/컴포넌트와 인터페이스를 통하여 통신해야 한다. 전역 또는 공통 변수들이 사용되는 경우, 각각의 경우 변수들은 잘 구조화되고 접근은 통제되어야하고 사용은 당위성이 있어야 한다.
- 모든 모듈/컴포넌트 인터페이스는 완전하게 문서화되어야 한다.
- 모든 모듈/컴포넌트 인터페이스는 모듈/컴포넌트 기능에 필요한 최소한의 필요한 인자만 포함해야 한다. 그리고
- 인자 개수의 적절한 제한이 명세 되어야 한다. 일반적으로 5 개이다.

B-39. 성능 모델링

○ 목적

시스템의 작동 용량이 명세 된 요구사항을 만족하기 충분한지 확인

○ 설명

요구사항 명세는 특정 기능에 대한 처리량 및 응답 요구사항을 포함하며 아마도 전체 시스템 자원의 사용에 대한 제약 조건과 결합된다. 제안된 시스템 설계는 다음에서 언급하는 요구사항과 비교된다.

- 시스템 프로세스의 모델과 그 상호작용을 정의,
- 각 프로세스 별 자원의 사용을 식별(예를 들어 프로세서 사용시간, 통신 대역폭, 저장장치 등),
- 평균적 및 최악의 조건하에서 시스템에 부과되는 수요의 분배 식별,
- 개별 시스템 기능에 대한 평균 및 최악의 처리량 및 응답시간 계산

간단한 시스템의 경우 분석 솔루션이 가능한 반면 더 복잡한 시스템의 경우 정확한 결과를 얻기 위하여 일부 시뮬레이션 형식이 필요하다.

상세 설계 전에 단순한 '리소스 예산' 점검표를 사용하여 모든 프로세스의 자원 요구사항을 합산할 수 있다. 만약 요구사항이 설계된 시스템 용량을 초과한다면 설계는 구현이 불가능하다. 설계가 이 점검표를 통과하더라도 성능 모델링은 자원 고갈로 인하여 과도한 지연과 응답시간이 발생할 수 있음을 보여줄 수 있다. 이러한 상황을 피하기 위하여 엔지니어들은 종종 자원 고갈의 가능성이 감소되도록 전체 자원의 일부(예: 50%)를 사용하도록 시스템을 설계한다.

B-40. 성능 요구사항

○ 목적

소프트웨어의 성능 요구사항이 만족되었음을 입증

○ 설명

모든 일반 및 특수, 명시적 및 암시적인 성능 요구사항을 식별하기 위하여 시스템과 소프트웨어 요구사항 명세 모두에 대한 분석이 수행된다. 각 성능 요구사항은 다음을 결정하기 위하여 차례로 검사된다.

- 얻어진 성능 기준
- 성공 기준에 반한 측정이 얻어질 수 있는지 여부
- 그러한 측정의 잠재적 정확성
- 측정이 추정될 수 있는 프로젝트의 단계, 그리고
- 측정이 가능한 프로젝트의 단계

성능 요구사항, 성공 기준 및 잠재적인 측정 목록을 얻기 위하여 각 성능 요구사항의 실행 가능성을 분석한다. 주요 목표는 다음과 같다.

- 각 성능 요구사항은 적어도 하나의 측정과 관련된다.
- 가능한 한 개발 과정의 초기에서 사용될 수 있는 가능하고 정확하고 효율적인 측정을 선택한다.
- 필수적이고 선택적인 성능 요구사항과 성공 기준을 식별한다. 그리고
- 가능하다면 하나 이상의 성능 요구사항에 대하여 단일 측정 사용 가능성의 이점을 취해야 한다.

B-41. 확률론적 시험

○ 목적

조사된 소프트웨어의 신뢰성 속성에 대한 정량적인 수치를 얻는 것. 이 수치는 관련된 수준의 신뢰와 중요성을 만족시킬 수 있으며

- 요청 당 고장 확률
- 특정 기간 동안 고장 확률, 그리고
- 오류 억제 확률

이러한 수치로부터 다음과 같은 다른 파라미터들이 유도될 수 있다.

- 고장 없는 실행의 확률
- 생존 확률
- 가용성
- MTBF 또는 고장률, 그리고
- 안전 실행 확률

○ 설명

확률론적 고려사항은 확률론적 시험 또는 운영 경험 중 하나에 기반을 둔다. 일반적으로 관찰된 운영 사례의 테스트 케이스 건수는 매우 많다. 시험을 용이하게 하기 위하여 일반적으로 자동화된 보조 장치가 사용된다. 이 장치들은 시험 데이터의 제공 및 시험 결과 감독의 세부사항에 관심이 있다. 대규모 시험은 적절한 프로세스 시뮬레이션 주변장치를 갖는 대형 호스트 컴퓨터에서 실행된다. 시험 데이터는 체계적 및 임의의 시점 두 가지에 따라서 선택된다.

첫 번째는 전반적인 시험 제어에 관한 것으로 예를 들어 시험 데이터 프로파일을 보장하는 것이다. 무작위 선택은 개발 테스트 케이스를 상세하게 취한다. 개별 시험 하네스, 시험 실행 및 시험 감독은 위에서 설명한 상세 시험 목표에 의해서 결정된다.

다른 중요한 조건들이 의도된 시험 목표의 관점에서 시험 평가를 가능하게 하기 위하여 충족되어야 하는 수학적 전제조건을 통해 주어진다. 시험 대상의 행위에 관한 확률론적 수치는 운영 경험으로부터 도출 될 수 있다. 동일한 조건이 충족되면, 시험 결과의 평가에 대하여 동일한 수학이 적용될 수 있다.

B-42. 프로세스 시뮬레이션

○ 목적

어떤 식으로든 그 소프트웨어가 실세계를 변경하는 것 없이, 소프트웨어의 기능을 그 소프트웨어의 외부 세계에 대한 인터페이스와 함께 시험

○ 설명

시험만을 목적으로 하는 시스템을 생성, 이 시스템은 시험 대상 시스템에 의해서 제어되는 시스템의 행위를 흉내 낸다. 시뮬레이션은 소프트웨어만이거나 소프트웨어와 하드웨어의 조합이 될 수 있다. 시뮬레이션은

- 시스템이 설치될 때 있게 될 시험 대상 시스템의 모든 입력을 제공한다.
- 제어하는 장비를 충실하게 대표하는 방식으로 시스템으로 부터의 출력에 응답한다.
- 시험 중인 시스템이 극복해야 하는 어떤 혼란을 제공하는 운영자 입력을 제공한다.

소프트웨어가 시험될 때, 시뮬레이션은 대상 하드웨어의 입력과 출력을 시뮬레이션 할 수 있다.

B-43. 프로토타입 / 애니메이션

○ 목적

주어진 제약사항에 대해 시스템 구현의 타당성을 점검, 오해를 찾기 위하여 명세자의 시스템의 해석을 고객에게 전함

○ 설명

시스템 기능, 제약 사항 및 성능 요구사항의 하위 집합이 선택된다. 프로토타입은 고수준 도구를 사용하여 만든다. 이 단계에서 대상 컴퓨터, 구현 언어, 프로그램 크기, 유지보수성 및 견고성과 같은 제약사항들은 고려할 필요가 없다. 프로토타입은 고객의 기준에 대하여 평가되고 시스템 요구사항은 이 평가에 비추어 수정될 수 있다.

B-44. 복구 블록

○ 목적

프로그램이 의도한 기능을 수행할 가능성을 높임

○ 설명

여러 개의 서로 다른 프로그램 섹션이 종종 독립적으로 작성되는데, 각 섹션이 동일한

원하는 기능을 수행하게 된다. 최종 프로그램은 이 섹션들로 구성된다. 기본이라고 하는 첫 번째 섹션이 제일 처음 실행된다. 그 다음 그것이 계산하는 결과의 합격 판정 시험이 이어진다. 만약 시험이 통과되면 결과가 받아들여지고 시스템의 후속 부분으로 전달된다. 만약 실패하면 첫 번째 섹션의 실행으로 발생된 변경사항들이 초기화되고 첫 번째의 대안인 두 번째 섹션이 실행된다. 이 역시 첫 번째 경우와 같이 합격 판정 시험을 받는다. 원하는 경우 두 번째, 세 번째 또는 그 이상의 대안을 제공할 수 있다.

B-45. 응답 시기 및 메모리 제약

○ 목적

시스템이 시간적인 요구사항 및 메모리 요구사항을 만족하는지 확인

○ 설명

시스템 및 소프트웨어 대한 요구사항 명세는 특정 기능에 대한 메모리와 응답 요구사항(아마도 전체 시스템 자원 사용에 대한 제약과 결합된)을 포함한다. 평균 및 최악 조건하에서 분배 요청을 식별하는 분석이 수행된다. 이 분석을 위해서는 각 시스템 기능의 자원 사용과 경과시간 예측을 해야 한다. 이 예측들은 여러 가지 방법(예를 들어 기존 시스템과 비교 또는 시간이 중요한 시스템의 프로토타입 및 벤치마킹)으로 얻을 수 있다.

B-46. 고장 복구 재시도 방법

○ 목적

감지된 고장 상태에서 재시도 방법을 통해 기능적 복구 시도

○ 설명

감지된 고장 또는 오류 상태의 경우에서 동일한 코드를 재실행하여 그 상황을 복구하려고 시도한다. 재 시도에 의한 복구는 소프트웨어 시간 초과 또는 작업 감시 작업 후 재부팅 및 재시작 또는 작은 재 스케줄링 및 작업 재시작으로 완료될 수 있다.

재시도 기술은 통신 오류에서 일반적으로 사용되며 오류 복구 및 재시도 조건들은 통신 프로토콜 오류(체크섬 등) 또는 통신 확인 응답 시간 초과로부터 표시될 수 있다.

B-47. 안전성 백

○ 목적

소프트웨어의 안전성에 악영향을 주는 잔여 명세 및 구현 결함을 방지

○ 설명

안전성 백은 다른 사양의 독립적인 컴퓨터에서 구현되는 외부 감시 이다. 안전성 백은 주 컴퓨터가 안전한 활동(정확한은 아닌)을 수행하도록 보장하기 위해서만 사용된다. 안전성 백은 주 컴퓨터를 끊임없이 감시한다. 안전성 백은 시스템이 안전하지 않은 상태로 진입하는 것을 방지한다. 또한 주 컴퓨터가 잠재적으로 위험한 상태로 진입하는 것을 감지하면 안전성 백 또는 주 컴퓨터 중 하나에 의해서 시스템이 안전한 상태로 회복되어야 한다.

B-48. 소프트웨어 형상 관리

○ 목적

소프트웨어 형상 관리는 구현 산출물들이 변경될 때 이들의 일관성을 보장하는 것을 목표로 한다. 일반적으로 형상 관리는 하드웨어 및 소프트웨어 개발 모두에 적용된다.

○ 설명

소프트웨어 형상 관리는 개발 처음부터 끝까지 사용되는 기술이다. 본질적으로 소프트웨어 형상 관리는 모든 중요한 산출물의 모든 버전의 생산을 기록하고 다른 산출물의 다른 버전 간의 모든 관계를 기록해야 한다. 기록 결과를 사용하여 설계자는 어떤 산출물의 변경으로 다른 산출물에 대한 영향을 결정할 수 있다. 특히 시스템 또는 서브시스템들이 일관된 컴포넌트 버전의 세트로 신뢰성 있게 재구축될 수 있다.

B-49. 엄격한 형식의 프로그래밍 언어

○ 목적

컴파일러에서 고 수준의 검사를 허용하는 언어를 사용하여 결함의 확률을 감소함

○ 설명

이러한 언어는 대게 기본 언어 데이터 타입(INTEGER, REAL 같은)으로부터 사용자 정의 데이터 유형을 정의할 수 있게 허용한다. 이렇게 사용자가 정의한 유형은 기본 유형과 완전히 동일한 방식으로 사용될 수 있지만 정확한 유형이 사용되는지 확인하기 위하여 엄격한 검사가 부과된다. 이러한 검사는 별도로 컴파일된 단위로 만들어진 경우에도 전체 프로그램에 부과된다. 또한 검사는 별도로 컴파일된 컴포넌트에서 참조되는 경우에도 프로시저 인자의 개수와 유형이 일치하는지 확인한다.

엄격한 형식적 언어는 잘 구조화된 프로그램으로 이어지는 쉽게 분석 가능한 제어 구조(예: IF ... THEN ... ELSE, DO ... WHILE 등) 같은 좋은 소프트웨어 공학 실무의 다른 측면을 또한 지원한다. 엄격한 형식적 언어의 전형적인 예는 Pascal, Ada 및

Modula-2 이다.

B-50. 구조 기반 시험

○ 목적

프로그램 구조의 일부 부분집합을 사용하는 시험 적용

○ 설명

프로그램의 분석에 기초하여 선택된 프로그램 요소의 큰 부분이 실행되도록 입력 데이터의 세트가 선택된다. 실행되는 프로그램 요소는 요구되는 엄격함의 수준에 따라 달라질 수 있다.

- 구문: 조건문의 분기를 실행하지 않고 모든 코드 구문을 실행할 수 있으므로 최소한의 엄격한 검사이다.
- 분기: 모든 분기의 양쪽을 점검해야 한다. 이는 어떤 종류의 방어적인 코드에 대해서는 실행 불가능할 수 있다.
- 복합 조건: 복합 조건 분기(즉, AND/OR로 연결된)의 모든 조건이 실행된다.
- LCSAJ(Linear Code Sequence And Jump): 선형 코드 연속 및 점프는 점프에 의해서 종료되는 조건 점프를 포함하는 임의의 코드 구문의 선형 연속이다. 많은 잠재적인 하위 경로들이 앞선 코드의 실행에 의해 부과된 입력 데이터에 대한 제약으로 인하여 실행이 불가능할 것이다.
- 데이터 흐름: 실행 경로는 예를 들어 동일한 변수가 기록되고 읽혀지는 경로와 같은 데이터 사용에 기초하여 선택된다.
- 호출 그래프: 프로그램은 다른 서브루틴에서 호출될 수 있는 서브루틴들로 구성된다. 호출 그래프는 프로그램의 서브루틴 호출 트리(tree)이다. 시험은 트리의 모든 호출을 포함하도록 설계된다.
- 전체 경로: 코드를 통해 모든 가능한 경로를 실행한다. 매우 많은 수의 잠재적인 경로로 인하여 완벽한 시험은 일반적으로 불가능하다.

B-51. 구조 다이어그램

○ 목적

프로그램의 구조를 도표로 보여줌

○ 설명

구조 다이어그램은 데이터 흐름 다이어그램을 보완하는 표기법이다. 구조 다이어그램은 프로그래밍 시스템과 부품의 계층구조를 설명하고 이것을 트리(tree)같은 그래픽 적으로 표시한다.

구조 다이어그램은 데이터 흐름 다이어그램의 요소가 프로그램 단위의 계층으로 어떻게 구현될 수 있는지 문서화한다. 구조 도표는 프로그램 단위의 활성화의 순서와 관련된 정보를 포함하지 않고 프로그램 단위들 간의 관계를 보여준다. 다음 세 가지 기호를 사용하여 그린다.

- 단위의 이름이 붙은 사각형
- 이 사각형들을 연결하는 화살표
- 구조 도표의 요소로부터 전달되는 데이터의 이름이 붙은 원이 있는 화살표.

일반적으로 원이 있는 화살표는 도표에서 사각형을 연결하는 화살표와 평행하게 그려진다. 어떤 중요한 데이터 흐름 다이어그램으로부터 다양한 구조 도표를 도출할 수 있다. 데이터 흐름 다이어그램으로부터 도출된 구조 도표는 시스템의 첫 번째 수준 구조를 나타내며 구조 도표의 각 상자는 데이터 흐름 다이어그램의 원(bubble)을 나타낸다. 당연히 더 깊은 수준은 동일한 기술을 사용하여 나타낼 수 있다.

B-52. 구조적 방법론

○ 목적

구조적 방법론의 주요 목표는 생명주기의 초기 부분에 집중하여 소프트웨어 개발의 품질을 향상시키는 것이다. 이 방법은 요구사항과 구현 기능의 존재를 논리적 순서와 구조적 방식으로 식별하기 위해 정확하고 직관적인 프로시저와 표기법(컴퓨터의 지원)을 통하여 이를 달성하는 것이 목표이다.

○ 설명

다양한 구조적 방법론이 존재한다. SSADM, LBMS와 같은 일부 방법론은 전통적인 데이터처리 및 트랜잭션 처리 기능을 위해 설계된 반면 다른 방법론들(MASCOT, JSD, 실시간 Yourdon)은 프로세스 제어 및 실시간 응용프로그램(더 안전성이 높은 경향이 있는)에 더 중점을 두고 있다. 구조적 방법은 본질적으로 문제 또는 시스템을 체계적으로 인지하고 분할하기 위한 '사고 도구' 이다. 주요 기능은 다음과 같다.

- 사고의 논리적 순서, 커다란 문제를 다루기 쉬운 단계로 분해
- 요구되는 시스템뿐만 아니라 환경을 포함한 전체 시스템의 식별
- 요구되는 시스템에서 데이터 및 기능의 분해
- 체크리스트, 즉 정의가 필요한 것들의 목록
- 낮은 지적 부담 - 단순하고, 직관적이고, 실용적

지원 표기법은 문제 및 시스템 개체(예: 프로세스 및 데이터 흐름)을 식별하기 위해 정확하지만 이들 개체가 수행하는 처리 기능은 비정형적 표기법을 사용하여 표현되는 경향이 있다. 그러나 일부 방법들은 (수학적인) 정형 표기법(예: JSD는 정규 표현식을 사

용, Yourdon, SOM, SDL은 유한 상태 기계를 사용)을 부분적으로 사용한다. 이 정확도는 오해의 범위를 줄일 뿐 아니라 자동 처리의 범위를 제공한다. 구조적 표기법의 다른 이점은 명세 또는 설계가 사용자에 의해서 직관적으로 확인을 가능하게 하는 가시성이다.

B-53. 구조적 프로그래밍

○ 목적

소프트웨어 컴포넌트의 실제 분석을 할 수 있는 방식으로 소프트웨어 컴포넌트를 설계하고 구현함. 이 분석은 모든 중요한 컴포넌트 행위들을 발견할 수 있어야 한다.

○ 설명

소프트웨어 컴포넌트는 최소한의 구조적 복잡도를 포함해야 한다. 복잡한 분기는 피해야 한다. 반복문 제약조건 및 분기는 (가능하면) 입력 인자들과 단순하게 관련되어야 한다. 소프트웨어 컴포넌트는 적절하게 작은 모듈로 나누어 져야 하며 이 모듈간의 상호 작용은 명시적이어야 한다. 위의 접근 방식을 장려하는 프로그래밍 언어의 특성은 효율성 같은 다른 특성, 효율성이 절대적인 우선순위를 차지하는 경우(예: 일부 안전성 중요 시스템)를 제외하면, 우선적으로 사용되어야 한다.

B-54. 적합한 프로그래밍 언어

○ 목적

가능한 한 많은 이 표준의 요구사항을 지원, 특히 방어적 프로그래밍, 강한 형식, 구조적 프로그래밍 및 가능한 단언들. 선택한 프로그래밍 언어는 최소한의 노력으로 쉽게 검증 가능한 코드를 이어지고 프로그램 개발, 검증 및 유지보수를 쉽게 한다.

○ 설명

언어는 완전하고 모호하지 않게 정의되어야 한다. 언어는 기계 지향적이 아닌 사용자 또는 문제 지향적이어야 한다. 널리 사용되는 언어 또는 그런 언어의 하위 집합이 특별한 목적의 언어보다 선호된다. 이미 참조된 특성들에 추가하여 언어는 다음을 제공해야 한다.

- 블록 구조
- 변환 시간 검사
- 실행 시 형(타입) 및 배열 경계 검사, 그리고
- 인자 검사

언어는 다음을 장려해야 한다.

- 작고 다루기 쉬운 컴포넌트의 사용
- 정의된 컴포넌트에서 데이터의 접근 제한
- 변수의 하위 범위 정의, 그리고
- 다른 오류 형(type)을 제한하는 구조

언어는 적절한 변환기, 기존 컴포넌트의 적절한 라이브러리, 디버거와 버전 관리 및 개발에 대한 도구들이 지원되는 것이 바람직하다. 검증을 어렵게 만드는 기능들 따라서 피해야하는 기능들을 다음과 같다.

- 서브루틴 호출을 제외한 무조건적 점프
- 재귀
- 포인터, 힙 또는 모든 유형의 동적 변수 또는 객체
- 소스코드 수준에서 인터럽트 처리
- 반복, 블록 또는 서브프로그램에서 여러 개의 입구 또는 출구
- 묵시적인 변수 초기화 또는 선언
- 가변적 레코드 및 동등성, 그리고
- 프로시저 인자화

저수준 언어, 특히 어셈블리 언어는 기계 지향적인 성격으로 인하여 문제가 발생한다.

B-55. 시간 페트리넷

○ 목적

시스템 행위의 관련된 측면을 모델링하고 분석 및 재설계를 통하여 안전성 및 운영 요구사항을 평가하고 가능한 향상함

○ 설명

페트리넷은 동시성 및 비동기적 행위를 나타내는 시스템에서 정보 및 제어 흐름을 나타내는데 적합한 그래프 이론적 모델의 부류에 속한다. 페트리넷은 플레이스와 전이들의 네트워크이다.

플레이스는 '표시' 또는 '표시 해제' 될 수 있다. 전이는 모든 입력 플레이스들이 표시될 때 '사용가능' 이 된다. 사용가능이 되면 '점화'가 허용(의무는 아님) 된다. 만약 점화되면 입력 표시가 제거되고 대신 전이로부터 각 출력 플레이스에 표시된다.

잠재적인 위험원은 모델에서 특정 상태(표시)로 표현된다. 확장된 페트리넷은 시스템의 타이밍 기능을 모델링 할 수 있다. '고전적'인 페트리넷이 제어 흐름 측면에 집중되어 있지만 데이터 흐름을 모델에 통합하기 위한 몇 가지 확장이 제안되었다.

B-56. 워크스루 / 설계 검토

○ 목적

가능한 한 빨리, 가능한 한 경제적으로 개발 과정의 일부 제품에서 오류를 감지

○ 설명

IEC는 정형적 설계 검토의 일반적인 설명, 목적, 다양한 설계 검토 유형의 세부 사항, 설계 검토 팀 구성 및 관련 업무와 책임을 포함하는 정형적 설계 검토에 대한 가이드인 IEC 61160을 발간했다. IEC 61160은 정형적 설계 검토를 계획하고 수행하기 위한 일반적인 가이드라인과 설계 검토 팀 내에서 독립적인 전문가 역할에 대한 세부사항을 제공한다.

IEC 61160은 “기능, 성능, 안전성, 신뢰성, 유지보수 가능성, 가용성, 비용 능력 및 최종 제품/과정에 영향을 주는 다른 특성, 사용자 또는 방관자가 영향을 미치는 모든 신제품/과정, 신규 응용프로그램 그리고 기존 제품 및 제조 과정의 수정에 대하여 정형적 설계 검토가 수행되어야 함“을 권고한다.

코드 워크스루는 소규모 종이 테스트 케이스 세트, 프로그램에 대한 대표적인 입력 및 해당하는 예상 출력 세트, 출력 워크스루 팀으로 구성된다. 시험 데이터는 프로그램의 논리를 통해 수동으로 추적된다.

B-57. 객체 지향 프로그래밍

○ 목적

신속한 프로토타이핑을 가능하게하고 기존 소프트웨어 컴포넌트를 더욱 쉽게 재사용하고, 정보 은닉을 달성하고, 전체 생명주기 동안 오류의 가능성을 줄이고, 유지보수 단계 동안 필요한 노력을 줄이고, 복잡한 문제를 더 쉽게 관리 가능한 작은 문제로 분해하고, 소프트웨어 컴포넌트 간의 의존성을 줄이고, 더욱 쉽게 확장 가능한 응용 프로그램을 생성함

○ 설명

객체 지향 프로그래밍은 계산적 추상화에 기반을 둔 것이 아닌 실세계에 존재하는 추상화를 기반으로 하는 소프트웨어에 대한 근본적으로 새로운 사고방식이다. 객체 지향 프로그래밍은 데이터 구조와 행위 모두를 포함하는 객체의 모음으로 소프트웨어를 구성한다. 이는 데이터 구조와 행위가 느슨하게 연결되는 전통적인 프로그래밍과는 대조적이다.

객체: 객체는 비공개 데이터 영역과 그 객체에 대한 연산들(메소드라고 하는)의 집합으로 구성된다.

메소드: 메소드는 공개 또는 비공개가 될 수 있다. 다른 소프트웨어 컴포넌트는 객체의 비공개 데이터를 직접적으로 읽거나 변경하는 것이 허용되지 않는다. 다른 모든 소프트웨어 컴포넌트는 그 객체의 비공개 데이터 영역의 데이터를 읽거나 수정하기 위해서는 객체의 공개된 메소드를 사용해야 한다.

객체 클래스: 객체 클래스(종종 유형 정의 형식으로)를 지정하여 동일한 클래스의 수많은 객체의 인스턴스화를 가능하게 한다. 즉, 모든 인스턴스화는 객체 클래스에 정의된 비공개 데이터 영역과 메소드를 갖는다.

(다중)상속: 객체 클래스는 비공개 데이터 영역과 비공개 데이터 추가, 메소드 추가 또는 상속받은 메소드의 구현 수정이 허용 되는 하나(또는 그 이상의)의 슈퍼 클래스(클래스 계층구조에서 그것 위의 객체 클래스들)를 상속받을 수 있다. 상속을 사용하여 다중 객체 클래스 트리를 만들 수 있다.

다형성: 동일한 연산이 다른 객체 클래스에서 다르게 동작할 수 있다. 예: 터미널 객체에 대한 쓰기 연산은 문자를 터미널에 쓰고 파일 객체에 대한 쓰기 연산은 문자를 파일에 쓴다.

단점: 객체 지향 프로그래밍 언어는 시스템 성능에 부정적인 영향을 미치는 리소스에 대한 추가적인 필요를 초래할 수 있다.

B-58. 추적성

○ 목적

추적성의 목적은 모든 요구사항이 적절하게 만족되고 추적할 수 없는 자료가 도입되지 않음을 보장하는 것이다.

○ 설명

요구사항에 대한 추적성은 시스템 확인에 중요한 고려사항이며 생명주기의 모든 단계를 통하여 입증이 가능하도록 수단이 제공되어야 한다. 추적성은 기능적 요구사항과 비 기능적 요구사항 모두에 적용 가능한 것으로 고려되어야 하며 특별히 다음을 만족해야 한다.

- 요구사항에서 설계 또는 그것을 만족하는 다른 객체에 대한 추적성
- 설계 객체에서 그것을 인스턴스화한 구현 객체에 대한 추적성
- 요구사항 및 설계 객체에서 시스템의 안전하고 적절한 사용에 적용되는 운영 및 유지보수 객체에 대한 추적성
- 요구사항, 설계, 구현, 운영 및 유지보수 객체에서 수용 가능성을 결정하게 될 검증과 시험 계획 및 명세에 대한 추적성
- 검증 및 시험 계획 및 명세에서 적용의 결과를 기록하는 시험 또는 다른 보고서에 대한 추적성

요구사항, 설계 또는 다른 객체들이 여러 개의 별도 문서로 인스턴스화 되는 경우, 추적성은 문서 구조 내에서 그리고 계층적인 방식으로 유지되어야 한다. 추적성 과정의 결과물은 정형적 형상 관리의 대상이 되어야 한다.

B-59. 메타 프로그래밍

○ 목적

메타 프로그래밍은 프로그래머가 모든 코드를 수동으로 작업하는데 걸리는 시간과 동일한 시간에 더 많은 작업을 할 수 있도록 한다.

○ 설명

메타 프로그래밍은 프로그램 작성 또는 데이터로써 다른 프로그램(또는 자체를)을 조작하는 또는 다른 경우 실행 시간에 수행되는 작업의 일부를 컴파일 시간에 하는 컴퓨터 프로그램을 작성하는 것이다.

메타 프로그래밍이 작성된 언어를 메타언어라 한다. 조작되는 프로그램의 언어는 목적언어라 한다. 프로그래밍 언어가 자신의 메타언어가 될 수 있는 능력을 반사(reflection) 또는 반영(reflexivity)라 한다. 반사는 메타 프로그래밍을 쉽게 하는 중요한 언어 기능이다.

첫 번째 클래스 데이터 유형으로 프로그래밍 언어 자체를(Lisp 처럼) 가지는 것은 매우 유용하다. 제네릭 프로그래밍은 언어 내에서 그것을 지원하는 언어로, 언어 내에서 메타 프로그래밍 기능을 호출한다.

메타 프로그래밍은 대게 두 가지 방법 중 하나를 통하여 작동한다. 첫 번째 방법은 API(Application Programming Interface)를 통하여 프로그래밍 코드의 실행 엔진의 내부 구조를 노출하는 것이다. 두 번째 방법은 프로그래밍 명령을 포함하는 문자열 식을 동적으로 실행하는 것이다. 그러므로 “프로그램이 프로그램을 작성할 수 있다“. 두 가지 접근방법 모두 사용될 수 있지만 대부분의 언어는 한쪽 또는 다른 쪽으로 기울어지는 경향이 있다.

B-60. 절차적 프로그래밍

○ 목적

프로그램이 원하는 상태에 도달하는데 걸리는 단계를 열거함

○ 설명

프로시저 호출의 개념에 기반을 두는 절차적 프로그래밍. 루틴, 서브루틴, 메소드 또는 함수 (수학적인 함수와 혼동하면 안 되지만, 함수적 프로그래밍에서 사용되는 것과 유사함) 라고 하는 프로시저는 수행되는 일련의 계산적 단계를 단순하게 포함한다. 주어진 프로시저는 다른 프로시저 또는 자체를 포함해서 프로그램의 실행 중 언제든지 호출될 수 있다.

B-61. 순차적 함수 도표

○ 목적

프로그램 알고리즘을 도식적으로 설명함

○ 설명

SFC 원소는 응용프로그램 알고리즘 단위를 방향성 링크로 연결된 단계 및 전이의 집합으로 나눌 수 있게 한다. 각의 단계와 연관된 동작들의 집합이 있고 각 전이는 전이 조건과 관련 있다. SFC 원소가 상태 정보를 저장해야 하므로 이들 원소를 사용하여 구조화 될 수 있는 응용프로그램 알고리즘의 유일한 단위는 함수 블록이다. (IEC 61131-3:2013, 6.7 참고)

B-62. 래더 다이어그램

○ 목적

프로그램을 도식적으로 설명함

○ 설명

IEC 61131-3:2013, 8.2 참고

B-63. 기능적 블록 다이어그램

○ 목적

입력 변수와 출력 변수간의 함수를 도식적으로 설명함

○ 설명

IEC 61131-3:2013, 8.3 참고

B-64. 상태 차트 또는 상태 다이어그램

○ 목적

시스템의 행위를 도식적으로 설명함

○ 설명

상태 차트 또는 상태 다이어그램은 시스템의 행위를 설명하는데 사용된다. 상태 다이어그램은 이벤트가 발생할 때 객체의 가능한 상태를 설명할 수 있다. 각 다이어그램은 일반적으로 단일 클래스의 객체를 나타내며 시스템을 통해 객체의 여러 가지 상태를 추적한다.

상태 다이어그램은 유한 상태 기계를 그래픽 적으로 표현하는데 사용될 수 있다. 이것은 Taylor Booth가 1967년에 저술한 “Sequential Machines and Automata Theory“에서 소개되었다. 다른 가능한 표현은 상태 전이 테이블이다. 유한 상태 기계에 대한 상태 다이어그램의 전형적인 형식은 방향성 그래프이다.

B-65. 데이터 모델링

○ 목적

데이터 모델을 생성

○ 설명

전산학에서 데이터 모델링은 데이터 모델링 기술을 사용하여 정형적인 데이터 모델

설명을 적용하여 데이터 모델을 생성하는 과정이다. 소프트웨어 공학에서 데이터 모델은 데이터가 표현되고 접근되는 방법을 설명하는 추상 모델이다. 데이터 모델은 관심 도메인에 대한 데이터 객체와 데이터 객체들 간의 관계를 형식적으로 정의한다. 데이터베이스 모델의 일부 일반적인 응용 프로그램은 데이터베이스의 개발 지원과 특정 관심 도메인에 대한 데이터 교환을 가능하게 하는 것을 포함한다. 데이터 모델은 데이터 모델링 언어로 명세 된다.

B-66. 제어 흐름 다이어그램/제어 흐름 그래프

○ 목적

시스템의 행위를 도식적으로 설명함

○ 설명

전산학에서 제어 흐름 다이어그램 또는 제어 흐름 그래프(CFG)는 프로그램 실행 중 프로그램을 통해 통과할 수 있는 모든 경로를 그래프 표기법을 사용해서 표현한 것이다. 그래프에서 각 노드는 기본 블록, 즉 점프 또는 점프 대상이 없는 직선의 코드 조각을 나타냄; 점프 대상은 블록을 시작하고 점프는 블록을 종료한다. 화살표는 제어 흐름에서 점프를 나타내는데 사용된다. 대부분의 발표에서는 두개의 특별히 설계된 블록이 있다: 제어가 흐름 그래프로 들어오는 시작 블록과 모든 제어 흐름이 빠져나가는 종료 블록이다.

CFG는 많은 컴파일러 최적화 및 정적 분석 도구에 필수적이다. 도달성은 최적화에 유용한 또 다른 그래프 속성이다. 만약 블록/서브그래프가 시작 블록을 포함하는 서브그래프로 연결되지 않으면 그 블록은 실행 중 도달할 수 없으므로 도달 할 수 없는 코드이다; 이 코드는 안전하게 제거될 수 있다.

만약 종료 블록이 시작 블록으로부터 도달할 수 없다면 이것은 무한루프를 나타낸다. 다시 말하지만 프로그래머가 다음과 같은 방식으로 명확하게 코드를 작성하지 않는 경우에도 실행되지 않는 코드와 무한루프가 가능하다. 점프 쓰레드 다음으로 상수 전파 및 상수 폴딩과 같은 최적화는 여러 기본 블록을 하나로 붕괴시킬 수 있으며 CFG에서 연결을 제거할 수 있음 등, 그러므로 그래프의 일부분이 연결 해제 될 수 있다.

B-67. 시퀀스 다이어그램

○ 목적

프로세스 또는 컴포넌트간의 상호작용을 도식적으로 설명함

○ 설명

시퀀스 다이어그램은 프로세스 또는 컴포넌트가 다른 프로세스와 어떻게 그리고 어떤 순서로 동작하는지 보여주는 일종의 상호작용 다이어그램이다.

B-68. 테이블 명세 기법

○ 목적

목표는 시스템의 데이터 기반 기능을 정의하는 표준화되고 잘 구조적인 방법을 제공하는 것이다.

○ 설명

신호 제어 테이블과 같은 테이블 표기법은 철도 신호 시스템에 대한 설치 특성 요구사항을 문서화하는 잘 정립된 방법이다. 이 기술은 시스템의 요소들 간 관계의 유형이 표준화되는 경우 적합하다.

장점: 테이블 형식과 각 필드의 가능한 항목은 검증 시 체크리스트 역할을 할 수 있다.

B-69. 응용 특화 언어

○ 목적

목표는 기존 프로그래밍 언어에 익숙하지 않은 응용 엔지니어가 쉽게 동화할 수 있는 개념과 용어를 사용하여 데이터 기반 시스템의 기능을 명세하는 방법을 제공하는 것이다.

○ 설명

응용 특화 언어는 일반적으로 기존의 고 수준 프로그래밍 언어와 유사한 제어 구조를 시스템 유형에 특화된 연산자와 결합한다. 이 기술은 이진 결정이 명세될 필요가 있는 경우 적합하지만 다른 곳에도 적용할 수 있다. 장점: 시스템이 처음 설계되었을 때 예견하지 못할 수 있는 비정상적인 상황에서 데이터를 생성할 수 있는 유연성

B-70. UML (통합 모델링 언어)

○ 목적

추상화를 통하여 복잡도를 줄일 수 있는 방식으로 소프트웨어 프로그램 및 관련 부산물을 표현함. 다양한 다이어그램 종류의 관점에서 기존 또는 계획된 설계의 모델링을 허용함으로써 UML은 적절한 수준의 표현을 기반으로 설계의 핵심 특성을 평가할 수 있다. UML은 상용 제품에 의해서 지원되는 소위 모델 중심 개발에 자주 사용된다. 이

개발 스타일은 고 수준 모델링 언어를 사용함으로써 소프트웨어의 품질과 개발자의 생산성을 향상시키는 것을 목표로 한다.

○ 설명

UML은 그래픽 지향 소프트웨어 명세 언어와 객체지향 프로그래밍 언어의 사용에서 시작된 표준화된 범용 모델링 언어이다. 이 전통을 기반으로 UML은 이전의 개념과 메소드를 많이 재사용한다. 모델은 구조적 다이어그램 또는 행위적 다이어그램으로 분류되는 하나 이상의 다이어그램 형식으로 작성되는데, 후자는 상호작용 다이어그램으로 분류된 네 가지 다이어그램 유형으로 구성된다.

B-70.1 구조 다이어그램

- 패키지 다이어그램: 각각의 관련 모델 요소를 포함하는 서로 다른 패키지 간의 내용 및 관계 표시
- 클래스 다이어그램: 전총적인 개체-관계 다이어그램의 적용을 기반으로 다른 특성 및 다른 객체 유형과의 관계로 객체 유형을 명세
- 객체 다이어그램: 서로 다른 객체(클래스 인스턴스)가 서로 어떻게 관련되어 있는지 보여줌
- 복합적 구조 다이어그램: 분류자(클래스 또는 컴포넌트와 같은)의 내부 구조 및 시스템의 다른 부분과의 상호작용 지점을 보여줌
- 컴포넌트 다이어그램: 시스템을 구성하는 컴포넌트, 그들 상호관계, 상호작용 및 외부 인터페이스를 구성하는 컴포넌트를 보여줌
- 적용 다이어그램: 소프트웨어가 실행 플랫폼을 통해 분산되는 방법을 명세

B-70.2 행위 다이어그램

- 활동 다이어그램: 데이터 전송과 동시 실행의 모델링을 허용하는 전통적인 순서도의 적용을 사용하여 알고리즘적인 행위를 설명
- 상태 기계 다이어그램: 유한 상태 기계(상태차트)를 사용하여 이벤트 중심 행위를 설명
- 유즈케이스 다이어그램: 특정한 유즈케이스를 얻기 위하여 시스템과 상호작용하는 모델 액터
- 상호작용 다이어그램: (통신 다이어그램, 상호작용 개요 다이어그램, 순서 다이어그램, 타이밍 다이어그램): 통신하는 객체들이 수행하는 활동으로 구성되는 시나리오를 설명

UML은 범용적 모델링 언어이지만 도메인에 특화된 해석을 프로파일을 통해 만들 수 있다. 표준 UML 개념을 정제함으로써 프로파일은 프로파일에 정의된 확장을 사용하여

이런 해석을 가능하게 한다. 이러한 방식으로 UML은 도메인 특화 언어를 정의하기 위한 기초로 사용된다.

B-71. 도메인 특화 언어

○ 목적

특정 도메인에 맞게 조정된 언어로 소프트웨어 프로그램 및 관련된 부산물을 표현

○ 설명

도메인 특화 언어(DSL)은 특정 응용 프로그램 도메인 또는 문제 도메인의 문제를 해결하거나, 특정한 기술을 사용하여 특별히 만들어진 프로그래밍, 명세 또는 모델링 언어이다. 이 언어는 이 도메인과 관련된 개념 및 기능을 기반으로 한다. 도메인 특화 언어는 Java와 UML 같은 또한 범용 언어 또는 모델링 언어와 달리 특수 목적 언어라고도 한다.

도메인 특화 언어의 중요한 이점 중 하나는 특정 도메인 내에서 일반 프로그래밍, 명세 또는 모델링에 대한 지식 없이도 문제를 표현하고 해결할 수 있는 가능성이 있다는 것이다. 결과적으로 프로그램, 명세 또는 모델은 최종 사용자에 의하여 더 고수준에서 생성될 수 있다. 이 도메인에 맞는 구조를 제공하고 자동화된 코드 생성을 위한 수단을 제공함으로써 DSL은 일반적으로 프로그래머의 생산성과 결과 제품의 품질을 향상시킨다. 코드 생성은 일반적으로 DSL을 입력으로 사용하는 응용 프로그램 생성기로 구현된다.

부록 C. 소프트웨어 SIL 1, 2의 T&M 적용 가이드

C-1. 기법 및 대책(T&M) 적용 가이드 개요

본 절에서는 철도 소프트웨어 개발 안전가이드 적용 시 소프트웨어 시스템의 안전성 확보를 위한 소프트웨어 개발 생명주기 상 전체 활동을 0~4까지의 안전 무결성 등급별로 충족해야 할 기법 및 대책을(T&M) 개발 조직이 용이하게 식별 가능하도록 목록표(Matrix)로 제공하며, 특히 소프트웨어 안전 무결성 등급 1, 2의 T&M을 M, HR, R 적용 요구기준에 따라 구체적인 활용 지침을 개념 설명과 예시를 포함해 제공함으로써 시스템의 안전등급을 확보할 수 있다.

C-2. 전체 T&M 전체 목록표(Matrix) 구성

본 절에서는 IEC 62279 표준의 소프트웨어 개발 생명 주기 상 안전과 관련한 전체 활동을 안전등급 0~4까지의 안전등급을 기준으로 적용해야 할 기술 및 대책을 목록표(Matrix)로 제공한다. (부록 H. 철도 표준에서 사용되는 기법 및 대책 목록 참조)

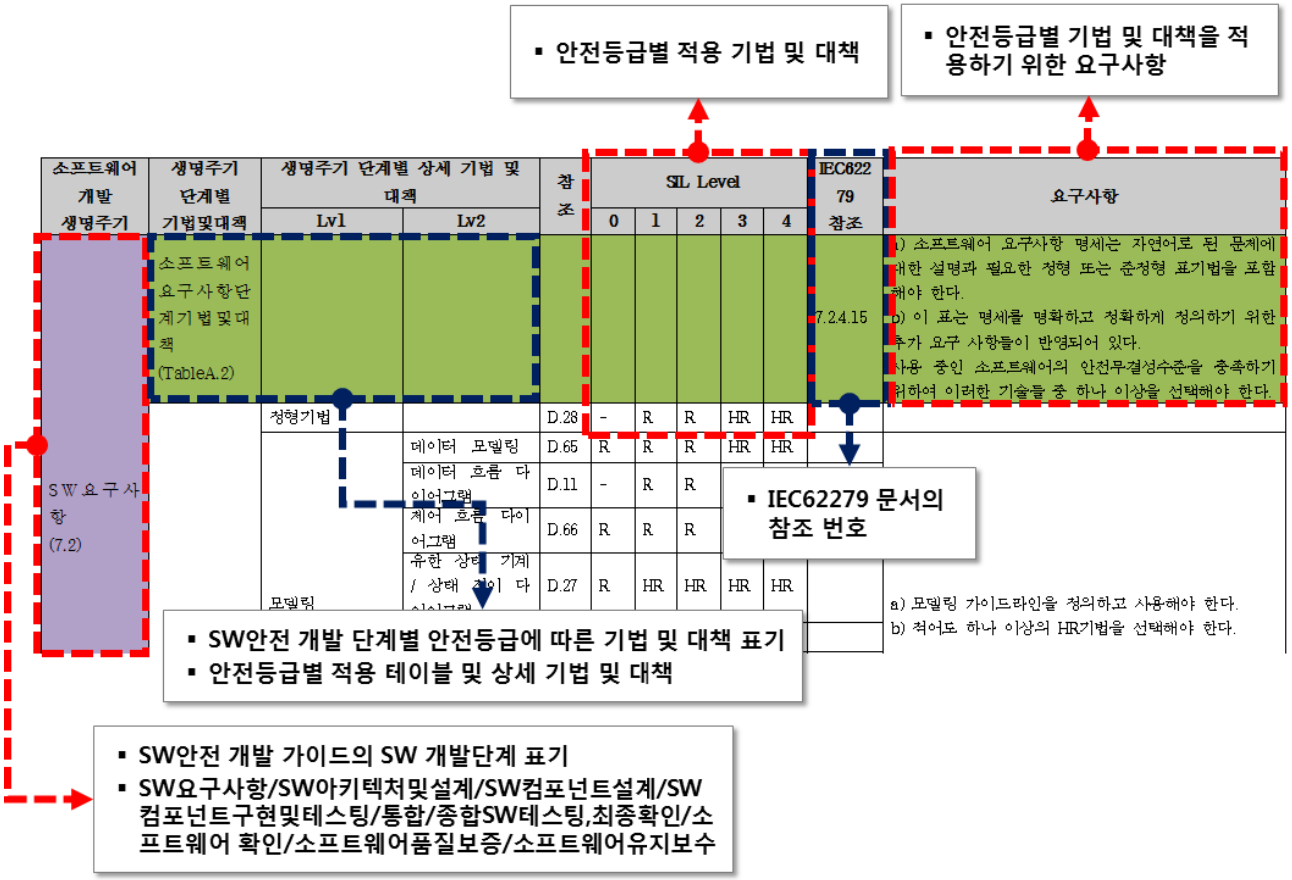


그림 217 소프트웨어 개발 생명 주기에 따른 적용 가능한 T&M 전체 목록표

C-3. 소프트웨어 SIL 1, 2 T&M 요구사항 목록표(Matrix)

소프트웨어 SIL 1, 2의 T&M 요구사항 목록표와 M, HR, R 적용 요구기준에 따른 각 T&M 별로 구체적인 활용 지침을 개념 설명과 예시를 포함함으로써 소프트웨어 시스템의 안전 무결성 등급을 달성할 수 있는 가이드를 제공한다.

소프트웨어 개발 생명주기	생명주기단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		구조적 프로그래밍		D.53	R	HR	HR		
		프로그래밍언어 (TableA.15)	ADA	D.54	R	HR	HR		
			MODULA-2	D.54	R	HR	HR		
			PASCAL	D.54	R	HR	HR		
		절차적 프로그래밍		D.60	R	HR	HR		
	확인및테스팅 (Table A.5)							7.4.4.10	소프트웨어 안전 무결성 수준 1 및 2의 경우, 승인된 기술 조합은 2, 3 또는 8중 하나와 함께 5가 함께 사용된다.
		정적 분석 (Table A.19)	제어 흐름 분석	D.8	-	HR	HR		
			데이터 흐름 분석	D.10	-	HR	HR		
			워크스루/설계 검토	D.56	HR	HR	HR		
		동적 분석 및 시험 (Table A.13)	경계값 분석으로부터 테스트 케이스 수행	D.4	-	HR	HR		
		추적성		D.58	R	HR	HR		
		코드시험적용범위 (TableA.21)	구문						
		기능 및 블랙박스 시험 (Table A.14)	경계값 분석						
			동등 클래스 및 입력 분할 테스팅	D.16	R	HR	HR		
						HR	HR		
SW 컴포넌트 구현 및 테스팅								7.5.4.7	
		구문		D.50	R	HR	HR		
통합	통합 (TableA.6)							7.6.4.6 7.6.4.10	
		기능 및 블랙박스 시	경계값 분석	D.4	R	HR	HR		

- SW-SIL 1,2 등급에 따른 적용 T&M 선별
- SW-SIL 1,2 등급에 따른 M, HR, R 적용 T&M 표 기술

- SW-SIL 1,2 등급 별 T&M 조합 적용 요구사항을 반영한 T&M 지침 상세화

그림 218 소프트웨어 생명 주기에 따른 적용 가능한 SIL 1, 2 T&M 목록표

C-4. 철도 표준에서 사용되는 기법 및 대책 목록 (SIL 1/2)

표 305 철도 표준에서 사용되는 기법 및 대책 목록 (SIL 1/2)

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
소프트웨어 요구사항	소프트웨어 요구사항 명세 (Table A.2)							7.2.4.15	
		모델링(Table A.17)	유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR		
		구조적 방법론		D.52	R	R	R		
	종합 소프트웨어 테스트 명세 (Table A.7)							6.2.4.5 7.2.4.18	
		성능시험 (TableA.18)	응답 시기 및 메모리 제약	D.45	-	HR	HR		
			성능 요구사항	D.40	-	HR	HR		
			경계값 분석	D.4	R	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR		
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR		
소프트웨어 아키텍처 및 설계	소프트웨어 아키텍처 (Table A.3)							7.3.4.14	
		방어적 프로그래밍		D.14	-	HR	HR		
		정보 캡슐화		D.33	R	HR	HR		
		완전하게 정의된 인터		D.38	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		페이스							
		구조적 방법론		D.52	R	HR	HR		
	소프트웨어설계 및 구현 (TableA.4)							7.3.4.24 7.4.4.6	
		모델링 (TableA.17)	유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR		
		구조적 방법론		D.52	R	HR	HR		
		모듈 방식		D.38	HR	M	M		
		컴포넌트 (TableA.20)	정보 캡슐화	D.33	R	HR	HR		
			완전하게 정의된 인터페이스	D.38	R	HR	HR		
		설계 및 코딩 표준 (Table A.12)	코딩 표준	D.15	HR	HR	HR		
			코딩 스타일 가이드	D.15	HR	HR	HR		
			조건 없는 점프 사용 금지	D.15	-	HR	HR		
			함수, 서브루틴과 메소드의 크기와 복잡도 제한	D.38	HR	HR	HR		
			함수, 서브루틴에 대한 진입/종료 시점 전략 및 방법	D.38	R	HR	HR		
			전역 변수 사용 제한	D.38	HR	HR	HR		
		분석 가능한 프로그램		D.2	HR	HR	HR		
		엄격한 형식의 프로그래밍 언어		D.49	R	HR	HR		
		구조적 프로그래밍		D.53	R	HR	HR		

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		프로그래밍언어 (TableA.15)	ADA	D.54	R	HR	HR		
			MODULA-2	D.54	R	HR	HR		
			PASCAL	D.54	R	HR	HR		
		절차적 프로그래밍		D.60	R	HR	HR		
	코딩 표준 (Table A.12)							7.3.4.25	
		코딩 표준		D.15	HR	HR	HR		
		코딩 스타일 가이드		D.15	HR	HR	HR		
		조건 없는 점프 사용 금지		D.15	-	HR	HR		
		함수, 서브루틴과 메소드의 크기와 복잡도 제한		D.38	HR	HR	HR		
		전역 변수 사용 제한		D.38	HR	HR	HR		
	확인 및 테스팅 (Table A.5)							7.3.4.32	소프트웨어 안전 무결성 등급 1 및 2의 경우, 승인된 기술 조합은 2, 3 또는 8중 하나와 함께 5가 함께 사용된다.
		정적 분석(Table A.19)	제어 흐름 분석	D.8	-	HR	HR		
			데이터 흐름 분석	D.10	-	HR	HR		
			워크스루/설계 검토	D.56	HR	HR	HR		
		동적 분석 및 시험 (Table A.13)	경계값 분석으로부터 테스트 케이스 수행	D.4	-	HR	HR		
		추적성		D.58	R	HR	HR		
		코드시험적용범위 (TableA.21)	구문	D.50	R	HR	HR		
		기능 및 블랙박스 시	경계값 분석	D.4	R	HR	HR		

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		험 (Table A.14)	동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR		
		인터페이스 시험		D.34	HR	HR	HR		
소프트웨어 컴포넌트 설계	소프트웨어 설계 및 구현 (TableA.4)							7.4.4.6	
		모델링 (Table A.17)	유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR		
		구조적 방법론		D.52	R	HR	HR		
		모듈 방식		D.38	HR	M	M		
		컴포넌트 (Table A.20)	정보 캡슐화	D.33	R	HR	HR		
			완전하게 정의된 인터페이스	D.38	R	HR	HR		
		설계 및 코딩 표준 (Table A.12)	코딩 표준	D.15	HR	HR	HR		
			코딩 스타일 가이드	D.15	HR	HR	HR		
			조건 없는 점프 사용 금지	D.15	-	HR	HR		
			함수, 서브루틴과 메소드의 크기와 복잡도 제한	D.38	HR	HR	HR		
			함수, 서브루틴에 대한 진입/종료 시점 전략 및 방법	D.38	R	HR	HR		
			전역 변수 사용 제한	D.38	HR	HR	HR		
		분석 가능한 프로그램		D.2	HR	HR	HR		
		엄격한 형식의 프로그래밍 언어		D.49	R	HR	HR		

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		구조적 프로그래밍		D.53	R	HR	HR		
		프로그래밍언어 (TableA.15)	ADA	D.54	R	HR	HR		
			MODULA-2	D.54	R	HR	HR		
			PASCAL	D.54	R	HR	HR		
		절차적 프로그래밍		D.60	R	HR	HR		
	확인 및 테스팅 (Table A.5)							7.4.4.10	소프트웨어 안전 무결성 등급 1 및 2의 경우, 승인된 기술 조합은 2, 3 또는 8중 하나와 함께 5가 함께 사용된다.
		정적 분석 (Table A.19)	제어 흐름 분석	D.8	-	HR	HR		
			데이터 흐름 분석	D.10	-	HR	HR		
			워크스루/설계 검토	D.56	HR	HR	HR		
		동적 분석 및 시험 (Table A.13)	경계값 분석으로부터 테스트 케이스 수행	D.4	-	HR	HR		
		추적성		D.58	R	HR	HR		
		코드시험적용범위 (TableA.21)	구문	D.50	R	HR	HR		
		기능 및 블랙박스 시험 (Table A.14)	경계값 분석	D.4	R	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR		
		인터페이스 시험		D.34	HR	HR	HR		
소프트웨어 컴포넌트 구현 및 테스팅	코드테스트커버리지 (TableA.21)							7.5.4.7	
		구문		D.50	R	HR	HR		
통합	통합 (TableA.6)							7.6.4.6 7.6.4.10	
		기능 및 블랙박스 시	경계값 분석	D.4	R	HR	HR		

소프트웨어 개발 생명주기	생명주기단계별 기법 및 대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level			IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2		
		험 (Table A.14)	동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR		
소프트웨어 유지보수	소프트웨어유지보수 (TableA.10)							9.2.4.6	
		영향 분석		D.32	R	HR	HR		
		데이터 기록 및 분석		D.12	HR	HR	HR		

C-5. SIL 1·2 상세 T&M 가이드 목록

아래의 T&M 가이드 목록은 SIL 1, 2의 T&M 목록표를 상세화한 후 해당 T&M에 대한 목차와 그에 속하는 세부 T&M, 참조 T&M 번호(No), 소프트웨어 생명 주기(Software Life Cycle)에서의 T&M 사용여부를 기술하였다. (소프트웨어 생애주기의 단계별 약자는 다음과 같다.)

표 306 소프트웨어 생애주기의 단계별 목록

No	약자	영문 단계명	한글 단계명
1	R	Software Requirement	소프트웨어 요구사항
2	AD	Architecture and Design	소프트웨어 아키텍처 및 설계
3	CD	Component Design	소프트웨어 컴포넌트 설계
4	CI&T	Component implementation and Testing	소프트웨어 컴포넌트 구현 및 테스팅
5	INT	Integration	통합
6	SM	Software Maintenance	소프트웨어 유지보수

T&M 요구사항 목록과 상세 T&M 요구사항의 관계는 다음의 그림과 같다.

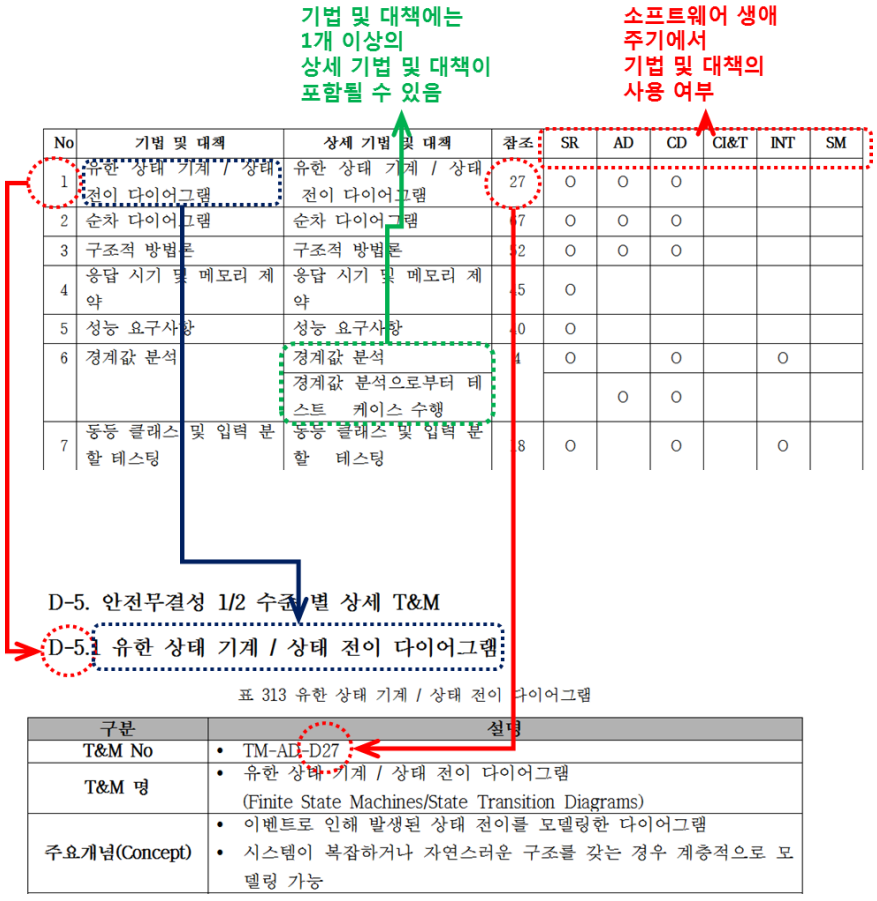


표 313 유한 상태 기계 / 상태 전이 다이어그램

구분	설명
T&M No	• TM-AD-D27
T&M 명	• 유한 상태 기계 / 상태 전이 다이어그램 (Finite State Machines/State Transition Diagrams)
주요개념(Concept)	• 이벤트로 인해 발생된 상태 전이를 모델링한 다이어그램 • 시스템이 복잡하거나 자연스러운 구조를 갖는 경우 계층적으로 모델링 가능

그림 219 T&M 가이드 목록과 상세 T&M의 관계 설명

표 307 T&M 가이드 목록

No	기법 및 대책	상세 기법 및 대책	참조	SR	AD	CD	CI&T	INT	SM
1	유한 상태 기계 / 상태 전이 다이어그램	유한 상태 기계 / 상태 전이 다이어그램	27	O	O	O			
2	순차 다이어그램	순차 다이어그램	67	O	O	O			
3	구조적 방법론	구조적 방법론	52	O	O	O			
4	응답 시기 및 메모리 제약	응답 시기 및 메모리 제약	45	O					
5	성능 요구사항	성능 요구사항	40	O					
6	경계값 분석	경계값 분석	4	O		O		O	
		경계값 분석으로부터 테스트 케이스 수행			O	O			
7	동등 클래스 및 입력 분할 테스팅	동등 클래스 및 입력 분할 테스팅	18	O		O		O	
8	방어적 프로그래밍	방어적 프로그래밍	14		O				
9	정보 은닉 / 캡슐화	정보 캡슐화	33		O	O			
10	모듈 방식	모듈 방식	38		O	O			
10	Modular Approach	모든 인터페이스 정의	38		O	O			
11	코딩 표준 및 형식 가이드	코딩 표준	15		O	O			
		코딩 스타일 가이드			O	O			
		동적 객체 사용 금지			O	O			
		동적 변수 사용 금지			O	O			
		포인터 사용 제한			O	O			
		재귀호출 사용 제한			O	O			
		조건 없는 점프 사용 금지			O	O			
12	분석 가능한 프로그램	분석 가능한 프로그램	2		O	O			
13	적합한 프로그래밍 언어	엄격한 형식의 프로그래밍 언어	54		O	O			
		ADA			O	O			
		MODULA-2			O	O			
		PASCAL			O	O			
		C Language			O	O			
		C++			O	O			
		C#			O	O			
14	절차적 프로그래밍	절차적 프로그래밍	60		O	O			
15	제어 흐름 분석	제어 흐름 분석	8		O				
16	데이터 흐름 분석	데이터 흐름 분석	10		O				
17	워크스루 / 설계 검토	워크스루 / 설계 검토	56		O				
18	추적성	추적성	58		O		O		
19	구조 기반 시험	구문	50		O	O	O		
20	인터페이스 시험	인터페이스 시험	34		O	O			
21	구조적 프로그래밍	구조적 프로그래밍	53		O	O			
22	영향 분석	영향 분석	32						O
23	데이터 기록 및 분석	데이터 기록 및 분석	12						O

C-6. SIL 1・2 상세 T&M

C-6.1 유한 상태 기계 / 상태 전이 다이어그램

구 분	설 명
T&M No	• TM-AD-D27
T&M 명	• 유한 상태 기계 / 상태 전이 다이어그램 (Finite State Machines/State Transition Diagrams)
주요개념(Concept)	• 이벤트로 인해 발생된 상태 전이를 모델링한 다이어그램 • 시스템이 복잡하거나 자연스러운 구조를 갖는 경우 계층적으로 모델링 가능
적용목적(Objective)	• 시스템의 제어구조를 정의하거나 구현 • 대상의 행위를 표현
적용단계(Phase)	• 소프트웨어 요구사항 • 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계 • 종합 소프트웨어 테스팅/최종확인
주요 기술 (Techniques)	• UML의 상태 전이 다이어그램의 표기법 참조

적용지침(Guideline)

- 적용 단계에 따른 사용
  - 상태 전이 다이어그램은 여러 적용 단계에 걸쳐 사용될 수 있으므로 적용 단계에 따라 깊이를 달리하여 적용 가능
  - 예컨대 소프트웨어 요구사항 명세 단계에 사용할 경우, 설계나 구현 단계에서의 사용될 때보다 더욱 추상적이고 모호하게 작성 가능
  - 요구사항 분석 단계에서의 다이어그램은 준정형적(semi-formal)으로 기술하는 하는 것도 가능
  - 설계, 구현 단계로 갈수록 더 구체적이고 명확하게 기술되어 구현 가능한 코드로 생성(Generation)할 수 있는 수준까지 정형적(formal)으로 표현해야 오해의 여지없이 구현 가능
- 상태 전이 다이어그램은 모든 케이스에 사용할 필요 없음
  - 요구 분석 단계에서 상태 전이 다이어그램을 사용하는 경우는 분석 과정 중 행위를 표현할 때 사용 가능
  - 행위 부분을 표현할 수 있는 다른 방법 (예. 시퀀스 다이어그램 등)이 있으므로 용도에 맞게 선택하여 사용하는 것을 권장
    - ✓ 이벤트에 대한 상태 변화가 중요한 경우는 상태 전이 다이어그램을 사용
    - ✓ 순차에 의한 요소들의 상호작용이 중요할 경우는 시퀀스 다이어그램을 사용
    - ✓ 핵심적인 부분이거나 필요할 경우는 다양한 기법을 취사선택하여 사용
- 상태 전이 다이어그램 작성 절차
  - 상태식별: 대상 요소가 가질 수 있는 상태를 식별
  - 상태 전이와 조건식별: 상태가 다른 상태로 바뀔 수 있는 전이를 식별하고 상태 전이의 조건을 식별
  - 다이어그램 작성: 표준 표기법에 맞춰 상태 전이 다이어그램을 작성

적용예시(Example)

- 건널목 제어 시스템 (Level Crossing Control System)
  - 건널목 제어 시스템을 통해 설계 시 상태 전이 다이어그램이 사용된 케이스와 그에 대한 설명을 보여준다.
  - 건널목 (Level Crossing)은 교량 또는 터널이 없는 동일한 레벨의 도로가 철도를 가로 지르는 경우를 말한다.

표 308 건널목 제어 시스템 요구사항 목록

No	요구사항
1	▪ 가장 중요한 안전 규칙은 건널목에서 도로 및 철도 교통을 제어하여 충돌을 피하는 것이다.
2	▪ 철도 건널목에는 차단기와 도로 신호등, 경보기가 있어 차량 통행을 제어한다.
3	▪ 철도에 두 개의 센서가 존재하여 건널목 안전 절차의 시작 (열차 입구)과 끝 (열차 출구)을 감지한다.
4	▪ 건널목 제어는 열차가 열차 입구에 들어오면 시작된다.
5	▪ 건널목 제어가 시작되면 신호등이 초록불(Green Fire)에서 노란불(Orange Fire)로, 노란불에서 빨간불(Red Fire)로 변경되며 차량 교통을 통제를 시작한다.
6	▪ 신호등이 바뀌면 차단기가 내려와 건널목을 폐쇄하여 차량 교통을 완전히 차단하고 열차가 지나갈 수 있다.
7	▪ 차단기는 정해진 시간 이내에 내려와야 한다.
8	▪ 열차가 두 번째 센서를(열차 출구) 지나가면 차단기가 올라가고 건널목 제어의 초기상태로 돌아간다.
9	▪ 신호등이나 센서나 차단기에 이상이 있을 경우 문제 상황을 담당자에게 알린다.
10	▪ 건널목 담당자는 수동으로 건널목 차단을 해제할 수 있다.

- 건널목 제어 시스템 상태 전이 다이어그램 작성
- 상태식별
  - 상태를 식별하기 위해 요구사항을 분석한다.
    - ✓ 대상 요소가 가질 수 있는 상태를 다음과 같이 9개의 상태로 식별하였다.
    - ✓ 요구사항 1,2의 모든 상태에 공통적으로 속한다.
    - ✓ 식별한 상태 외에 초기 상태(Initial State)가 존재한다.
  - 아래의 표는 식별한 상태에 대해 기술하고 있다. 식별된 상태, 각 상태에 대한 설명과 상태식별을 위하여 참조한 요구사항 번호를 같이 기술하였다.

표 309 건널목 제어 시스템의 상태식별 목록

No	상 태	설 명	참조요구사항번호
1	Idle	▪ 건널목 제어는 열차가 열차 입구에 들어오면 시작 ▪ 열차가 들어오기 전의 상태를 초기(Idle) 상태로 함	1 2
2	Activate	▪ 열차가 열차 입구에 들어오면 건널목 제어가 시작되는 활성화(Activate) 상태	3 4
“생 략”			
9	Canceling and Unsecured State	▪ 신호등이나 차단기에 결함이 발생한 상태	9 10

- 상태전이와 조건(Event) 식별
  - 상태들 간의 상태 전이를 식별하고 상태 전이를 하기 위한 조건(Event)을 식별한다.
    - ✓ 요구사항 1,2의 모든 상태 전이에 공통적으로 속한다.
    - ✓ 식별된 상태 전이, 각 상태 전이에 대한 설명과 상태 전이 식별을 하기 위하여 참조한 요구사항 번호를 같이 기술하였다.
    - ✓ 건널목 제어 시스템에서 식별한 상태 전이는 18번의 3건의 상태 전이를 포함하여 모두 23개의 상태 전이를 식별하였고 그에 따른 상태 전이 조건(Event)을 기술하였다.
  - 아래의 표는 식별한 상태 전이와 상태 전이 조건 그리고 이들을 도출하기 위해 참고한 요구사항 번호에 대해 기술하고 있다.

표 310 건널목 제어 시스템의 상태전이 목록

No	상태 전이	상태 전이 조건 (Event)	참조요구사항번호
1	Initial → Idle	▪ 상태 전이 다이어그램의 시작 상태에서 초기 시작 상태에서 건널목 제어 시스템의 대기 상태로 변경	1 2
2	Idle → Idle	▪ 열차가 들어오기 전까지 대기(Idle) 상태도 대기	3 4
3	Idle → Activate	▪ 열차가 열차 입구에 들어오면 대기(Idle) 상태에서 활성화(Activate) 상태로 변경	3 4
“생 략”			
18	┌Activate ├Blocking State 1 (Orange Fire) ├Blocking State 2 (Red Fire) └Blocking State 3 (Lowering Barrier) → Canceling and Unsecured State	▪ 4가지 상태 (활성화, 노란불, 빨간불, 차단기 내려옴)에서 신호등이나 차단기에 문제가 발생할 경우 취소 및 불안전 상태(Canceling and Unsecured State)로 변경하고 차단 오류 보고를 보냄 ▪ 4개의 상태 전이를 기술하고 있음	9
19	Canceling and Unsecured State → Canceling and Unsecured State	▪ 건널목의 차단이 풀릴 때까지 상태 대기	9 10
20	Canceling and Unsecured State → Unblock State	▪ 건널목의 차단이 풀리면 차단 해제 상태(Unblock State)로 변경	9 10

- 다이어그램 작성
  - 식별한 상태, 상태전이와 조건을 표준 표기법에 맞춰 상태 전이 다이어그램을 작성한다.
  - 상태와 상태전이 작성
    - ✓ 건널목 제어 상태 다이어그램은 1개의 초기 상태와 9개의 상태, 23개의 전이(Transition)과 그에 따른 상태 전이 조건(Event)을 갖는다.
    - ✓ 절차에 따라 식별한 상태와 상태 전이를 상태 전이 다이어그램 표기법에 맞춰 아래와 같은 다이어그램으로 작성할 수 있다.

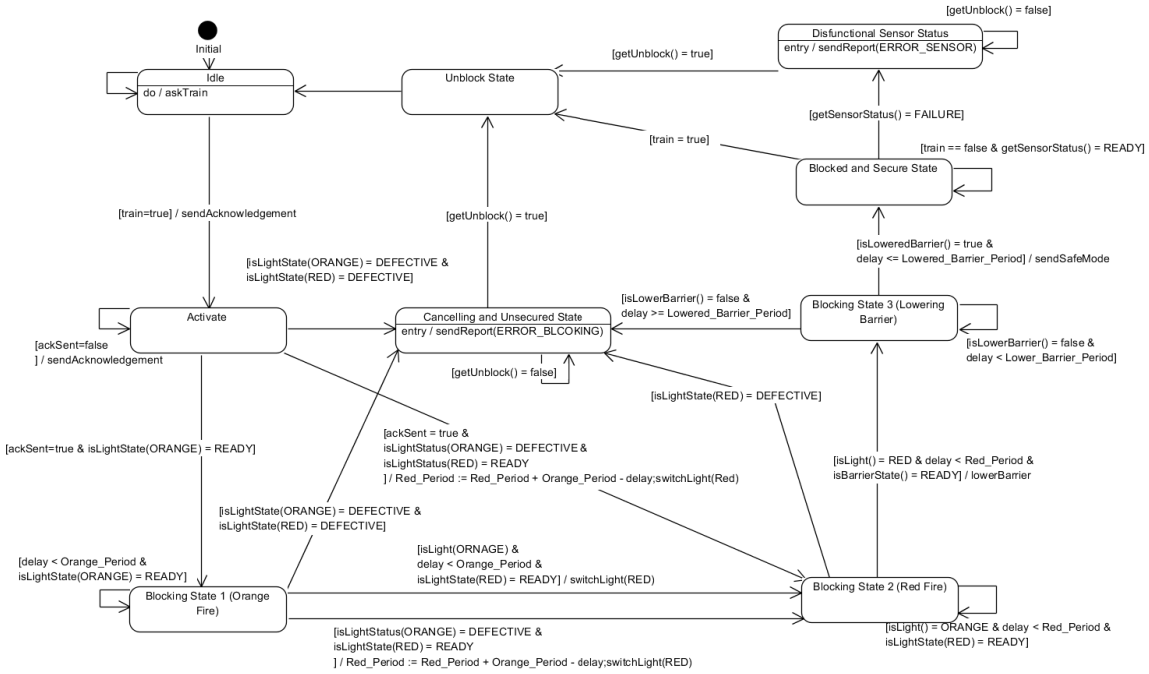


그림 220 건널목 제어 시스템의 상태 전이 다이어그램

적용 시 고려사항(Considerations & Constraints)

- 다이어그램 작성 시 고려사항
  - 플로 차트와 차이
    - ✓ 상태 전이 다이어그램은 시간이 흐르면 다음으로 진행되는 플로 차트와 달리 명시적인 이벤트에 의한 상태의 전이가 일어나므로 이벤트를 표시해야 함
  - 상태에 들어오는 이벤트와 나가는 이벤트 표시
    - ✓ 무한 루프에 빠질 수 있으므로 반드시 포함되어야 함
  - 시스템의 완전성
    - ✓ 모든 상태의 모든 입력에 대해서 시스템의 동작을 정의함으로써 시스템을 완전하게 정의 가능
  - 계층화
    - ✓ 시스템이 복잡할 경우 계층화된 상태 전이 다이어그램으로 구성
- 완전성 검사
  - 프로그램의 완전성은 안전 시스템일 경우 매우 중요한 속성이므로 반드시 체크되어야 함
  - 이를 지원하는 도구를 사용하여 쉽게 파악 할 수 있음
  - 모든 상태에 대한 입출력 상태 전이를 검사
    - ✓ 상태에 대한 입력 상태 전이만 존재하고 출력 상태 전이가 존재하지 않을 경우 무한 루프에 빠질 수 있음
  - 모든 상태에 대한 도달 가능성 검사
    - ✓ 입력 상태 전이가 존재하지 않는 상태의 경우 결코 도달 할 수 없는 상태로 제거할 상태인지 오류인지 검토가 필요
- 다이어그램 적용 단계에 대한 고려사항
  - 상태 전이 다이어그램은 여러 단계에 걸쳐 적용할 수 있는 기법이나 모든 단계에 반드시 적용해야 할 필요는 없음
  - 단계별로 적용할 수 있는 부분을 고려하여 상황에 적합한 경우, 선택적으로 사용해

야 함

표 311 상태 전이 다이어그램의 적용 단계에 대한 고려사항

적용단계	적용 가능한 상황
소프트웨어 요구사항	▪ 요구사항 분석 시 행위적 관점에서 분석 가능
소프트웨어 아키텍처 및 설계	▪ 시스템의 동적 설계 시 상태, 입력, 동작 관점으로 설계할 때 사용 가능
소프트웨어 컴포넌트 설계	▪ 컴포넌트 내부의 동적 설계 시 상태, 입력, 동작 관점으로 설계할 때 사용 가능 ▪ 상세 설계 후 자동화 도구를 통해 소스 코드로 생성(Generation) 가능
종합 소프트웨어 테스팅/최종확인	▪ 상태 전이 다이어그램의 각 상태, 입력, 동작을 추출하여 테스트 케이스를 추출하여 자동화 가능

- 상태 전이 다이어그램을 위한 지원 기능 존재
  - 상태 전이 다이어그램의 구현을 검증하기 위한 테스트 케이스 자동 생성 도구
  - 상태 전이 다이어그램을 구동을 시각화 도구
  - 상태 전이 다이어그램을 소스코드로 생성(generation)하는 도구
  - 소스코드를 상태 전이 다이어그램으로 리버스 엔지니어링(reverse engineering)하는 도구

- 상태 전이 다이어그램의 확장
  - 복잡한 시스템 동작에 대한 묘사를 향상시키기 위해 확장됨
    - ✓ 계층구조
    - ✓ 병렬처리
    - ✓ 하위 레벨 전이
    - ✓ 히스토리 상태 등

C-6.2 순차 다이어그램

구 분	설 명
T&M No	• TM-AD-D67
T&M 명	• 순차 다이어그램 (Sequence Diagram)
주요개념(Concept)	• 프로세스 또는 컴포넌트가 다른 프로세스와 어떻게 그리고 어떤 순서로 동작하는지 보여주는 일종의 상호작용 다이어그램
적용목적(Objective)	• 프로세스 또는 컴포넌트 간의 상호작용을 도식적으로 설명함
적용단계(Phase)	• 소프트웨어 요구사항 • 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계 • 종합 소프트웨어 테스팅/최종확인
주요 기술 (Techniques)	• UML의 시퀀스 다이어그램의 표기법을 참조

적용지침(Guideline)

- 적용 단계에 따른 사용
  - 시퀀스 다이어그램은 여러 적용 단계에 걸쳐 사용될 수 있으므로 적용 단계에 따라 깊이를 달리하여 적용 가능
  - 예컨대 소프트웨어 요구사항 분석 단계에 사용할 경우, 설계나 구현 단계에서의 사용될 때보다 더욱 추상적이고 모호하게 작성될 수 있음
  - 요구사항 분석 단계에서의 다이어그램은 도메인에서 사용하는 용어로 기술 하는 것도 가능
  - 설계, 구현 단계로 갈수록 좀 더 구체적이고 명확하게 기술되어 구현자가 설계를 보고 구현할 수 있는 수준까지 상세하게 표현해야 오해의 여지없이 구현이 가능함
- 시퀀스 전이 다이어그램을 모든 케이스에 사용할 필요는 없음
  - 요구 분석 단계에서 시퀀스 다이어그램을 사용하는 경우는 분석 과정 중 행위를 표현할 때 사용 가능
  - 행위 부분을 표현할 수 있는 다른 방법(예. 상태 전이 다이어그램)이 있으므로 용도에 맞게 선택하여 기술하도록 함
    - ✓ 순차에 의한 요소들의 상호작용이 중요할 경우는 시퀀스 다이어그램을 사용
    - ✓ 이벤트에 대한 상태 변화가 중요한 경우는 상태 전이 다이어그램을 사용
    - ✓ 핵심적인 부분이거나 필요할 경우는 다양한 기법을 취사선택하여 사용
- 시퀀스 다이어그램 작성 절차
  - 요소 식별: 요구사항으로부터 주체가 될 수 있는 요소들을 식별
  - 상호작용 식별: 시간 순에 따른 요소들 간의 상호작용(이벤트 혹은 오퍼레이션)을 식별
  - 다이어그램 작성: 표준 표기법에 맞춰 시퀀스 다이어그램을 작성

적용예시(Example)

- 건널목 제어 시스템 (Level Crossing Control System)
  - 본 예시는 도로가 철도를 가로지는 건널목 시스템의 시퀀스 다이어그램이 사용된 케이스와 그에 대한 설명을 보여준다.

- 시나리오
  - 건널목 제어 시스템 (Level Crossing Control System)을 기반으로 아래와 같은 시나리오가 만들어졌다.
  - 시나리오는 TM-AD-D28의 요구사항을 참조한다.

표 312 건널목 제어 시스템의 시나리오 목록

No	시나리오
1	▪ 건널목의 제어가 시작되면 열차 제어 시스템은 건널목 시스템을 감지하고 안전 모드를 시작한다.
2	▪ 열차 제어 시스템은 열차에 속도를 줄이도록 요청하고 열차운행 통신 시스템에 활성화를 요청한다.
3	▪ 활성화를 요청 받은 열차운행 통신 시스템은 건널목 통신 시스템과 건널목 메인 시스템에 활성화를 차례로 요청한다.
4	▪ 건널목 메인 시스템은 건널목 신호등 관리자에 노란 불로 변경하고 빨간 불로 변경할 것을 순차적으로 요청한 뒤 건널목 차단기 관리자에 차단기를 내릴 것을 요청한다.
5	▪ 건널목 메인 시스템은 건널목 통신 시스템에 안전한 상태라는 것을 알리고 건널목 통신 시스템은 다시 열차운행 통신 시스템에 안전 상태임을 메시지로 보낸다.
6	▪ 열차운행 통신 시스템이 열차 제어 시스템에 안전 모드가 종료되었음을 알리면 열차 제어 시스템은 열차에 속도를 올릴 것을 요청하고, 열차는 열차 센서를 활성화 시킨다.
7	▪ 열차 센서가 건널목 메인 시스템은 비활성화 시키면 건널목 메인 시스템은 건널목 차단기 관리자에게 차단기를 올리게 하고 건널목 신호등 관리자에게 스위치를 끄도록 한다.

- 건널목 제어 시스템 시퀀스 다이어그램 작성
- 요소식별
  - 요소를 식별하기 위해 시나리오를 분석한다. 시나리오로부터 다음과 같이 8개의 요소를 식별하였다.
  - 아래의 표는 식별한 요소에 대해 기술하고 있다. 식별된 요소명, 각 요소의 영문명과 요소 식별을 하기 위하여 참조한 시나리오 번호를 같이 기술하였다.
  - 식별한 요소는 시스템 외의 열차와 물리적인 요소도 포함한다.

표 313 건널목 제어 시스템의 요소 식별 목록

No	요 소 명	영 문 명	참조 시나리오
1	열차 제어 시스템	TrainbornControl System	1,2,6
2	열차	TrainPhysical	2,6
3	열차운행 통신 시스템	TrainbornCommunication System	2,3,5,6
4	건널목 통신 시스템	LevelCrossingCommunication System	3,5
5	건널목 메인 시스템	LCMain	3,4,5,7
6	건널목 신호등 관리자	LCLightManager	4,7
7	건널목 차단기 관리자	LCBarrierManager	4,7
8	열차센서	TrainSensor	6

- 상호작용 식별
  - 요소들 간의 순차적인 상호작용을 식별한다.
  - 아래의 표는 식별한 상호작용 대해 기술하고 있다. 식별된 상호작용, 각 상호작용

에 대한 설명과 상호작용 식별하기 위하여 참조한 요구사항 번호를 같이 기술하였다. 요구사항 1,2의 모든 상태 전이에 공통적으로 속한다.

- 건널목 제어 시스템에서는 모두 17개의 상호작용을 식별하였다.
- 식별한 상호 작용은 다음과 같이 이벤트 혹은 오퍼레이션으로 전환하였다.

표 314 건널목 제어 시스템의 상호작용 식별 목록

No	이용자	제공자	상호작용	참조 시나리오
1	열차 제어 시스템	열차 제어 시스템	건널목 시스템을 감지	1
2	열차 제어 시스템	열차 제어 시스템	안전모드 시작	1
3	열차 제어 시스템	열차	감속	2
4	열차 제어 시스템	열차운행 통신 시스템	활성화를 요청	2
5	열차운행 통신 시스템	건널목 통신 시스템	활성화를 요청	3
6	열차운행 통신 시스템	건널목 메인 시스템	활성화를 요청	3
7	건널목 메인 시스템	건널목 신호등 관리자	노란불로 변경	4
8	건널목 메인 시스템	건널목 신호등 관리자	빨간불로 변경	4
9	건널목 메인 시스템	건널목 차단기 관리자	차단기 내리기	4
10	건널목 메인 시스템	건널목 통신 시스템	안전 상태임을 알림	5
11	건널목 통신 시스템	열차운행 통신 시스템	안전 상태임을 알림	5
12	열차운행 통신 시스템	열차 제어 시스템	안전모드 종료	6
13	열차 제어 시스템	열차	가속	6
14	열차	열차센서	활성화를 요청	6
15	열차센서	건널목 메인 시스템	비활성화 요청	7
16	건널목 메인 시스템	건널목 차단기 관리자	차단기를 올리기	7
17	건널목 메인 시스템	건널목 신호등 관리자	스위치를 끄기	7

표 315 건널목 제어 시스템의 메시지 목록

No	상호작용	메시지(오퍼레이션)
1	건널목 시스템을 감지	detectLevelCrossingSystem()
2	안전모드 시작	startSafetyMode()
3	감속	decreaseSpeed()
4	활성화를 요청	askActivation()
5	활성화를 요청	send(ASK_ACTIVATION)
6	활성화를 요청	activate()
7	노란불로 변경	switchYellow()
8	빨간불로 변경	switchRed()
9	차단기 내리기	lowerBarrier()
10	안전 상태임을 알림	sendSafeStatus()
11	안전 상태임을 알림	send(SAFE_STATUS)
12	안전모드 종료	stopSafetyMode()
13	가속	increaseSpeed()
14	활성화를 요청	activate()
15	비활성화 요청	deactivate()
16	차단기를 올리기	upperBarrier
17	스위치를 끄기	switchOff()

- 다이어그램 작성
  - 다음의 시퀀스 다이어그램은 건널목을 제어하기 위한 8개 요소들의 17개 상호작용을 시간 순으로 보여주고 있다.
  - 절차에 따라 식별한 요소와 상호작용을 시퀀스 다이어그램 표기법에 맞춰 다음의 그림과 같은 다이어그램으로 작성할 수 있다.
  - 다이어그램은 건널목 제어 시스템의 요소들과 그들의 순차적 상호작용을 시각적으로 보여주고 있다.

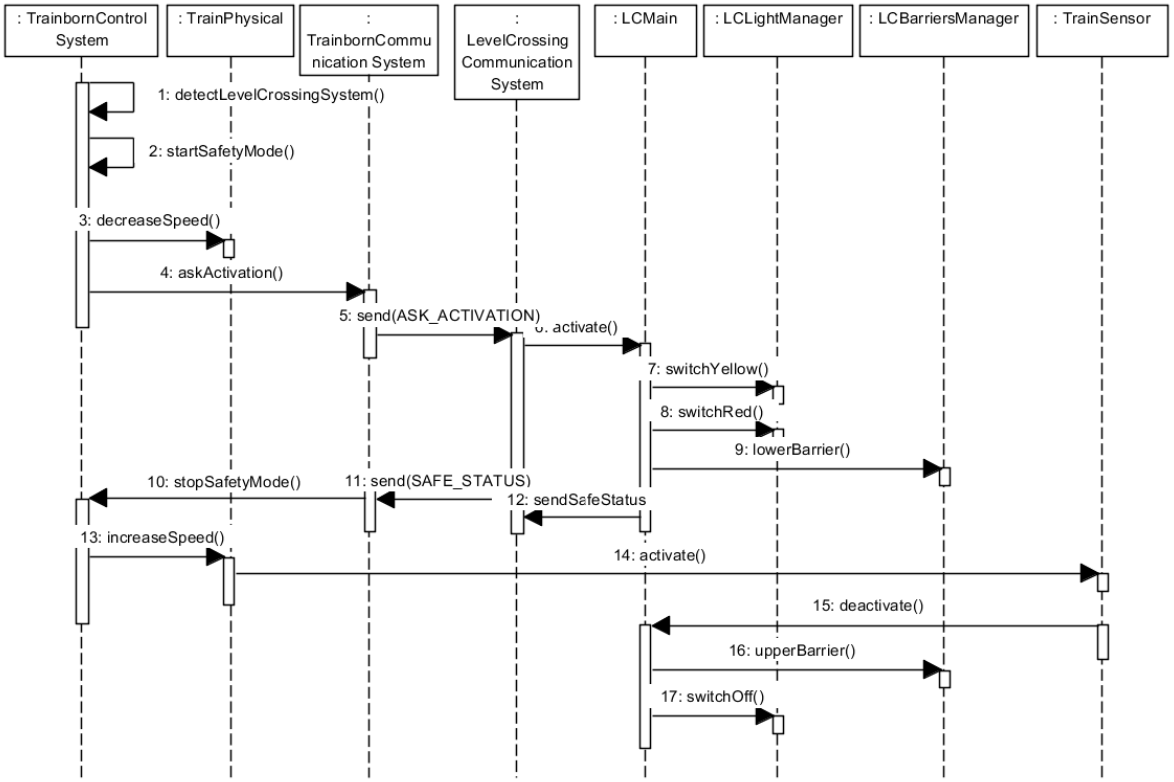


그림 221 건널목 제어 시스템의 시퀀스 다이어그램

적용 시 고려사항(Considerations & Constraints)

- 다이어그램 적용 단계에 대한 고려사항
  - 시퀀스 다이어그램은 여러 단계에 걸쳐 적용할 수 있는 기법이나 모든 단계에 반드시 적용해야 할 필요는 없음
  - 단계별로 적용할 수 있는 부분을 고려하여 상황에 적합한 경우, 선택적으로 사용해야 함

표 316 시퀀스 다이어그램의 적용 단계에 대한 고려사항

적용단계	적용 가능한 상황
소프트웨어 요구사항	▪ 요구사항 분석 시 행위적 관점에서 분석 가능
소프트웨어 아키텍처 및 설계	▪ 시스템의 동적 설계 시 컴포넌트 간의 순차적인 상호작용을 설계할 때 사용 가능
소프트웨어 컴포넌트 설계	▪ 컴포넌트 내의 동적 설계 시 순차적인 상호작용을 설계할 때 사용 가능
종합 소프트웨어 테스팅/최종 확인	▪ 각 요소 사이의 상호작용을 분석하여 테스트 케이스 추출에 이용 가능

C-6.3 구조적 방법론

구 분	설 명
T&M No	• TM-AD-D52
T&M 명	• 구조적 방법론 (Structured Methodology)
주요개념(Concept)	• Structured Methodology는 전체 소프트웨어 개발 생명주기 상에서 초기 부분에 집중하여 소프트웨어 개발의 품질을 향상시키는 방법
적용목적(Objective)	• 이 방법은 요구사항과 구현 기능의 식별을 위해 논리적인 순서와 구조화 된 방식으로 정확하고 직관적인 절차와 컴퓨터를 활용한 표기법을 통해 이를 달성하는 것을 목표로 함
적용단계(Phase)	• 소프트웨어 요구사항 • 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술 (Techniques)	• 컨텍스트 다이어그램(Context Diagram) • 데이터 흐름도(Data Flow Diagram)

적용 지침(Guideline)

- 구조화 방법론에는 다양한 방법이 존재하며, SSADM, LBMS와 같은 방법은 기존의 데이터 처리 및 트랜잭션 처리 기능을 위해 설계된 반면 MASCOT, JSD, Real Time Yourdon 방법은 더 안전한 프로세스 제어 및 Real time 어플리케이션 설계에 적합 할 수 있음.

- 구조화 방법론은 문제 또는 시스템을 체계적으로 인식하고 분할하기 위한 도구이며, 다음과 같은 특징을 가짐.
  - 논리적인 사고 순서, 규모가 큰 문제를 다루기 쉬운 단계로 분할,
  - 시스템 주변 환경을 포함한 전체 시스템의 식별,
  - 시스템의 데이터와 기능의 분해,
  - 분석 및 정의 시 필요한 체크리스트
  - 간단하고, 직관적이며, 실용적

- 문제 및 시스템의 개체(프로세스 및 데이터 흐름)를 식별하기 분석하기 위한 표기법은 정확해야 하지만, 이러한 작업을 수행하는 방법은 비공식 표기법을 사용해 표현하는 경향이 있음. 일부 방법은 (수학적으로) 정형 표기법을 부분적으로 사용하며, 이를 활용하여(JSD는 정규 표현식 사용, Yourdon, SOM 및 SDL은 유한 상태 머신 사용) 정확도를 제고함으로써, 범위를 잘못 이해하는 것을 줄이고 자동화 프로세스에 대한 적용 범위를 제공함.

적용예시(Example)

1) Yourdon-모델링 방법을 활용한 시스템 분석

- 시스템의 전체 범위와 기능 요구사항을 식별하고 분석하게 위해 Yourdon-모델링을 적용함으로써 시스템의 소프트웨어 기능을 하향식으로 분할해 시스템이 수행해야 할 기능에 대해 완전하고 모호하지 않게 단계 별로 상세화 할 수 있음

- 첫 번째 단계에서는 전체 시스템을 하나의 단일 프로세스로 기술하고 외부 시스템 또는 개체와의 인터페이스를 표시하는 컨텍스트 다이어그램을 작성하여 시스템의 전체 뷰를 추상적으로 표현할 수 있음

- 다음 단계에서는 시스템의 각 부분을 구성하는 상호작용을 포함한 개별 프로세스로 분할해 최상위 수준의 데이터 흐름을 기술하며, 컨텍스트 다이어그램과 유사한 형태로 프로세스의 흐름을 표현할 수 있다. 실선 화살표는 데이터 흐름을 나타내는 반면, 점선 화살표는 간단한 토큰(Boolean Value)만 전송하는 제어흐름을 표현한다.

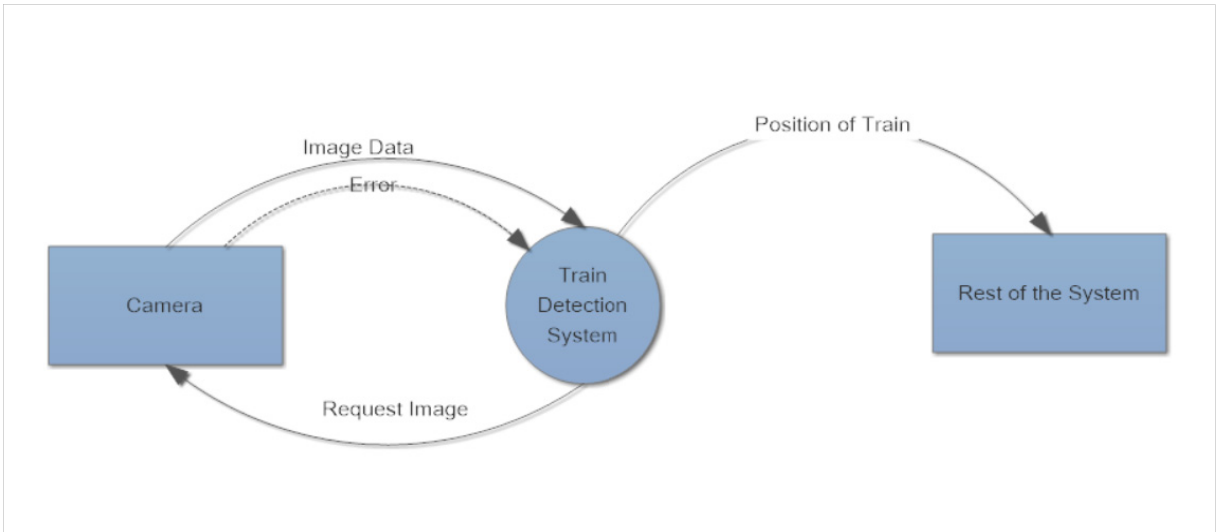


그림 222 Yourdon-모델링 방법을 활용한 시스템 분석의 예

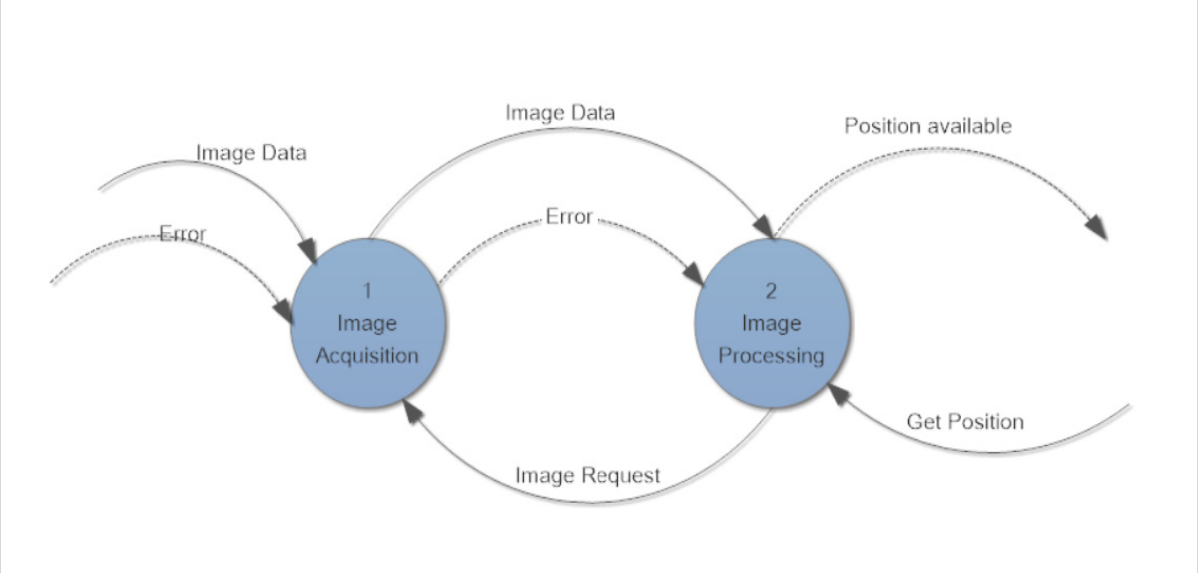


그림 223 Yourdon-모델링을 적용한 최상위 수준의 데이터 흐름도

- 시스템의 복잡성에 따라 최상위 데이터 흐름은 2 레벨, 3 레벨까지 상세화 할 수 있으며, 최하위 수준에서의 프로세스 버블은 더 이상 분할 할 수 없는 최소단위 프로세스로 식별한다.

- 다음의 그림과 같이 최종 데이터 흐름을 기술한 다이어그램을 기반으로 각 데이터 흐름 및 데이터 저장소의 내부 구조를 설명하는 데이터 사전을 개발할 수 있음.

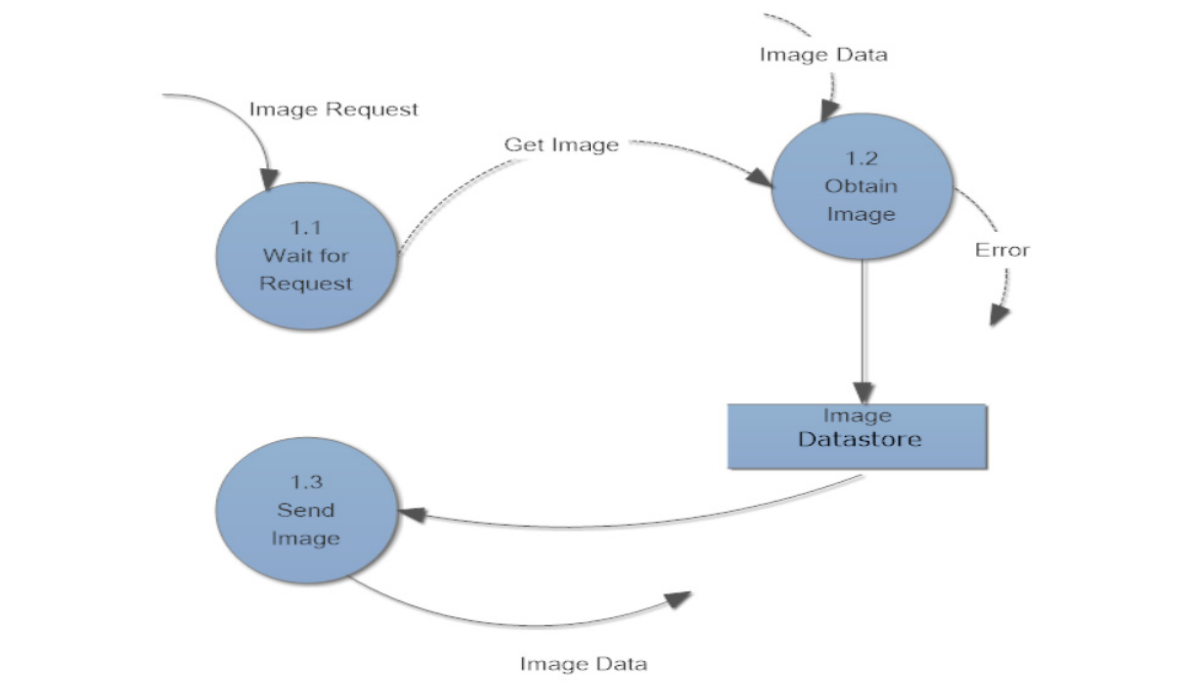


그림 224 Image Acquisition에 대한 2-레벨의 데이터 흐름도

- 데이터 사전은 데이터 흐름의 구조를 매우 신중한 방식으로 시각화 할 수 있으나, 복잡한 구조는 보여주기가 어렵거나 불가능하므로, 최하위로 식별된 프로세스에 대해 미니스펙(mini-spec)을 활용함으로써 프로세스가 데이터를 처리하는 설명을 기술한다.
- 미니스펙은 최하위 데이터 흐름에서 식별된 데이터 처리 항목에 처리 절차를 요약해 기술한 명세로 실제 구현을 위한 상세설계 내용을 기술하며, 다음 예는 최하위 데이터 흐름도에서 식별한 1.3 Send Image 데이터 처리 항목을 명세화한 예를 설명
- 1.3 SEND IMAGE - Mini-Specification
  - Initialize an offset with 0.
  - For each call, do the following:
  - Write the current callcount to the first 4 bytes of “Imagedata”
  - Initialize an offset2 with 5.
  - For the bytes of “Imagedata”, do the following:
  - Take the data from “Image” at offset and write it to “Imagedata” at offset2.
  - Increment offset and offset2.
  - If there is no data left in “Image” or “Imagedata” is full, stop the iteration.
  - Otherwise: Repeat.
  - Return “Imagedata”

적용 시 고려사항(Considerations & Constraints)

- 구조화 방법론 적용함에 있어 이 방법은 소프트웨어 개발뿐만 아니라 개발 초기 단계부터 기능 블록 간의 종속성과 계층 구조를 식별하고 기술함으로써 혼란을 피할 수 있는 다른 매우 복잡한 프로세스와 하드웨어에도 적용될 수 있음.
- 구조적 방법론 중 하나인 MASCOT은 영국 육군에 의해 개발되었으며 1980년대 후반과 1990 년대 초반에 적극적으로 사용되었으나, 민간 프로젝트에 거의 활용되지 못했으며, LSDM과, SSADM은 비교적 최근까지 사용되고 있으나, Real Time 시스템에는 적합성이 떨어지며, JSD와 Yourdon 방법 등이 활동되고 있음.

C-6.4 응답 시기 및 메모리 제약

구 분	설 명
T&M No	• TM-AD-D45
T&M 명	• 응답 시기 및 메모리 제약 (Response Timing and Memory Constraints)
주요개념(Concept)	• 시스템 및 소프트웨어 대한 요구사항 명세는 특정 기능에 대한 응답시간과 메모리 제한 요구사항을 포함 • 비기능적 요구사항인 응답시간과 메모리 제한의 준수 여부를 확인하기 위해 성능 테스트를 수행
적용목적(Objective)	• 시스템이 시간적인 요구사항 및 메모리 요구사항을 만족하는지 확인
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 구현 및 테스팅 • 통합 • 종합 소프트웨어 테스팅/최종확인
주요 기술 (Techniques)	• 성능 테스트

적용 지침(Guideline)

- 비기능적 요구사항
  - 특정 기능에 대한 응답시간과 메모리 등의 리소스 사용에 대한 제약 조건에 대한 요구사항

- 성능 테스트
  - 사용자의 요구사항 중 비기능적 요구사항을 소프트웨어가 준수하는지 여부를 확인하는 테스트를 수행
  - 성능 테스트 절차
    - ✓ 성능 요구사항 정의
    - ✓ 성능 테스트 수행 방법 정의
    - ✓ 성능 테스트 수행
    - ✓ 성능 테스트 수행 결과 분석

- 성능 요구사항 정의
  - 특정 기능에 대한 응답 시간과 메모리(자원) 제한 사항
    - ✓ 시간이 중요한 시스템의 경우 응답시간이 중요한 비기능 요구사항
    - ✓ 임베디드 시스템의 경우 제한된 자원(메모리, CPU 등)에서 시스템이 구동 되어야 함
  - 응답 시간과 메모리(자원) 제한 사항을 예측하기 위한 방안
    - ✓ 기존 시스템과 비교
    - ✓ 프로토타이핑을 선행
    - ✓ 경쟁 제품의 벤치마킹

- 성능 테스트 수행 방법 정의
  - 성능 요구사항을 달성했는지 여부를 측정하기 위한 구체적이고 객관적인 방법을 정의

- 테스트를 위한 다음의 항목들이 정의되어야 함
  - ✓ 테스트 일정
  - ✓ 테스트 주체
  - ✓ 테스트 환경
  - ✓ 테스트 대상
  - ✓ 테스트 수행방법(절차)

- 성능 테스트 수행
  - 테스트 수행을 위해 테스트를 위한 프로그램을 작성 가능
  - 테스트 자동화 도구를 사용 가능
    - ✓ 테스트 도구는 부하를 발생시키고 부하에 대한 응답시간 및 자원사용 결과를 분석할 수 있는 기능을 제공함
    - ✓ 예상된 부하에 대한 응답시간과 자원 사용량 측정 (평균, 최악)
      - ◆ 부하를 늘려가며(동시 요청수) 요청응답시간(RT: Response Time)과 초당 처리 건수(TPS: Throughput Per Second)와 메모리 사용량 측정

- 성능 테스트 수행 결과 분석
  - 소프트웨어가 성능 요구사항을 준수하였는지 여부를 평가 및 성능 분석
    - ✓ 통합 시스템에 대한 응답시간과 자원 사용량 측정
    - ✓ 성능 요구사항 만족 여부와 만족하지 못할 경우 원인 분석
  - 테스트 분석 도구는 테스트 결과 분석을 용이하도록 시각화 기능 지원

적용예시(Example)

- 건널목 제어 시스템 (Level Crossing Control System)
  - 본 예시는 도로가 철도를 가로지는 건널목 시스템의 성능 테스트에 대한 예시를 보여준다.

- 성능 요구사항 정의
  - 본 예시의 성능 요구사항은 TM-AD-D28의 기능 요구사항에 대한 응답시간에 대한 비기능 요구사항이다.
    - ✓ 기능 요구사항

표 317 건널목 제어 시스템의 요구사항 목록 (일부)

No	요구사항
3	▪ 철도에 두 개의 센서가 존재하여 건널목 안전 절차의 시작 (열차 입구)과 끝 (열차 출구)을 감지한다.
4	▪ 건널목 제어는 열차가 열차 입구에 들어오면 시작된다.

    - ✓ 비기능 요구사항

표 318 건널목 제어 시스템 비기능 요구사항

No	요구사항
1	▪ 건널목 제어는 열차가 열차 입구에 들어오면 즉시 시작되어야 한다. ▪ 열차가 건널목 구간에 시작점의 센서에 감지된 후, 8ms 이내에 열차의 감속이 시작되어야 한다.

  - 기능 요구사항 3, 4에 대해서 열차 감속과 관련된 성능요구사항 1을 정의하였다.

- 성능 테스트 수행 방법 정의

- 테스트 일정: 테스트 수행 시기는 소프트웨어 통합 후 1일
- 테스트 환경: 테스트는 실제 환경에서 테스트가 용이하지 않으므로 시뮬레이션 환경에서 수행
- 테스트 주체: 시스템 개발자
- 테스트 대상: 통합된 건널목 제어 시스템
- 테스트 수행방법(절차)
  - ✓ 측정 구간은 열차 입구의 센서가 열차를 감지한 직후부터 열차가 감속을 시작한 시점

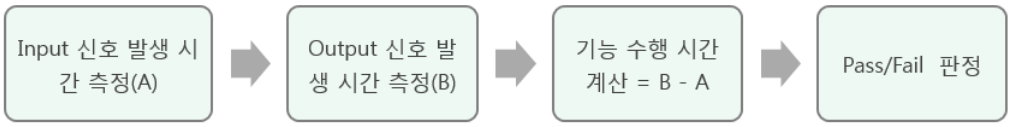


그림 225 테스트 시 측정 구간을 구하는 절차

  - ✓ 구동 성능을 측정하기 위한 테스트 도구 사용

- 성능 테스트 수행
  - 테스트 자동화 도구를 사용하여 테스트를 수행
  - 열차가 센서에 감지된 후 열차 감속까지 걸린 시간의 평균, 최단, 최장의 시간 등의 결과를 리포팅
  - 결과

표 319 성능 테스트 수행 결과

분류	평균	최단	최장
시간	6ms	4ms	10ms

- 성능 테스트 수행 결과 분석
  - 테스트 지원 도구에서는 시스템에 대한 응답시간을 측정하고 그 결과를 시각화해서 보여주어 분석을 지원한다.
    - ✓ 시각화 예제

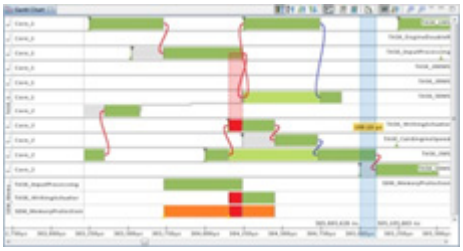


그림 226 테스트시각화 예제1

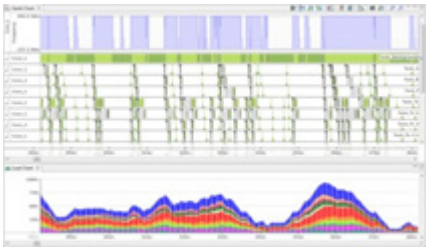


그림 227 테스트시각화 예제2

  - 결론
    - ✓ 평균적으로 성능 요구사항인 8ms를 달성하고 있으나 최장 시간으로 10ms를 기록하여 성능 요구사항을 달성하지 못하고 있다.
    - ✓ 성능 요구사항 만족 여부와 만족하지 못할 경우 원인 분석
      - ◆ 시간 관점에서 기능 분배가 적절히 이루어졌는지 확인
      - ◆ 구간별 타이밍 결과를 분석한 결과 열차 제어 시스템은 열차에 감속 이벤트를 전달하는 구간에서 전체 소요시간의 80%정도 소모
      - ◆ 이 구간의 최적화가 필요

적용 시 고려사항(Considerations & Constraints)

- 테스트 시 평균 및 최악 조건하에서 분배 요청을 식별하는 분석을 수행해야하기 위해 각 시스템 기능의 자원 사용과 경과시간 예측을 해야 하고 이는 평가 기준에 참고 될 수 있음

- 예측을 위해 기존 시스템과 비교하거나 시간이 중요한 시스템의 프로토타입 및 벤치마킹 가능

C-6.5 성능 요구사항

구분	설명
T&M No	• TM-AD-D40
T&M 명	• 성능 요구사항(Performance Requirements)
주요개념(Concept)	• 사용자가 요구하는 소프트웨어 시스템의 품질목표를 달성하기 위하여 반드시 고려되어야 할 성능과 관련한 요구사항을 식별하고, 이를 분석하는 활동
적용목적(Objective)	• 사용자가 요구하는 소프트웨어 시스템의 처리량, 응답 시간, 자원 사용 등의 성능목표를 달성하기 위한 요구사항을 식별하고 명세하는 기준을 제시
적용단계(Phase)	• 소프트웨어 요구사항
주요 기술 (Techniques)	• 성능 모델링(Performance Modeling) • 성능 테스트(Performance Testing)
적용 지침(Guideline)	

- 성능 요구사항은 소프트웨어 시스템의 특성, 업무 목표, 사용자 유형, 상위 요구사항에 따른 품질 속성의 하나로, 소프트웨어 시스템의 목표를 달성하기 위하여 반드시 고려되어야 할 사항임. 다음은 소프트웨어 품질 국제 표준(ISO 9126)을 기반으로 품질 특성과 부특성을 나타낸 표로, 성능 요구사항은 효율성(Efficiency), 신뢰성(Reliability) 품질 특성 및 부특성과 관련이 있음

표 320 소프트웨어 품질특성 및 관련 요구사항

품질특성	품질부특성	요구사항내용
효율성 (efficiency)	시간효율성 처리효율성 자원효율성	시스템이 특정 조건에서 제공할 시간과 자원에 대해 기술한다.
신뢰성 (reliability)	가용성 오류허용성 복구성	시스템 사용 시 필요한 시간과 자원에 대해 목표를 기술한다.
사용성 (usability)	이해성 학습성	사용자가 시스템을 쉽게 운용하거나 사용법을 쉽게 배울 수 있도록 관련된 요구사항을 기술한다.
유지보수성 (maintainability)	변경성 안전성 운영성	시스템의 변경 수행 절차 또는 문제 발생의 해결 방안을 기술한다.
이식성 (portability)	설치성 적응성 상호운용성	시스템의 설치 및 운용이 가능한 플랫폼이나 기술과 또한 기존 시스템이나 정보와의 호환성을 기술한다.
보안성 (Security)	기밀성 무결성	시스템 및 시스템 데이터에 대한 보호와 데이터 무결성 관련 요구사항을 기술한다.

준수성 (Compliance)	표준적합성	시스템과 관련된 각종 표준 관련 요구사항을 기술한다.

- 모든 일반 및 특수, 명시적 및 묵시적은 성능 요구사항을 식별하기 위하여 시스템과 소프트웨어 요구사항 명세 모두에 대한 분석이 수행되며, 각 성능 요구사항은 다음을 고려해 식별함.
  - 테스트 성공 기준에 반한 측정이 얻어질 수 있는지 여부
  - 그러한 측정의 잠재적 정확성
  - 측정이 추정될 수 있는 프로젝트의 단계 및 측정이 가능한 프로젝트의 단계
  - 성공 기준에 반한 측정이 얻어질 수 있는지 여부
  - 그러한 측정의 잠재적 정확성
  - 측정이 추정될 수 있는 프로젝트의 단계

- 성능 요구사항, 성공 기준 및 잠재적인 측정 목록을 얻기 위하여 각 성능 요구사항의 실행 가능성을 분석하며, 주요 목표는 다음과 같음.
  - 각 성능 요구사항은 적어도 하나의 측정과 관련
  - 가능한 한 개발 과정의 초기에서 사용될 수 있는 가능하고 정확하고 효율적인 측정을 선택
  - 필수적이고 선택적인 성능 요구사항과 성공 기준을 식별
  - 가능하다면 하나 이상의 성능 요구사항에 대하여 단일 측정 방법을 사용

적용예시(Example)

- 시스템이 특정 조건에서 요구된 기능을 제공할 수 있도록 시간효율성, 처리효율성 및 자원효율성과 관련된 [14]성능 요구사항에 대한 작성 기준과 예시는 다음과 같으며, 특히 각 성능 요구사항에 대한 성능 목표와 대상 정보의 목표 기준 값을 계량적으로 기술함.

  - 시간효율성 요구사항 (Time efficiency requirements)
    - ✓ 작성 기준 : 시스템이 명시된 조건에서 사용자나 시스템으로부터 요구된 기능을 수행할 때 필요한 시간을 정의
    - ✓ 작성 내용
      - ◆ 응답시간(response time) : 정상 또는 부하 상태에서 사용자가 시스템에 조회를 요구한 직후부터 응답 메시지가 출력되기 시작될 때까지 최대로 허용할 수 있는 시간
      - ◆ 반응시간(reaction time) : 정상 또는 부하 상태에서 외부 입력에 대하여 결과를 발생시킬 때까지 최대로 허용할 수 있는 시간
      - ◆ 전송시간(transmission time) : 정상 또는 부하 상태에서 데이터가 전송을 시작하여 최종 목적지까지 도달하는데 최대로 허용할 수 있는 시간

  - 처리효율성 요구사항 (Processing efficiency requirements)
    - ✓ 작성 기준 : 시스템이 명시된 조건에서 특정 기능을 수행하는 비율이나 동시에 처리해야 하는 목표 양을 정의한다. 처리효율성 요구사항은 목표 시스템의 작업 형태(온라인 배치 등)에 따라 성능 요구치가 다를 수 있는 것을 고려하여 정의

✓ 작성 내용 ◆ 동시 처리 능력 : 정상 또는 부하상태에서 동시 사용자 수, 동시 트랜잭션 수 ◆ 최대 처리 능력 : 정상 또는 부하상태에서 최대 사용자 수, 최대 트랜잭션 수
적용 시 고려사항(Considerations & Constraints)
• 해당 사항 없음

14) 성능요구사항 작성 기준 예시 : 한국정보통신기술협회(TTA) 소프트웨어 품질 요구사항 작성 지침 참조

C-6.6 경계값 분석

구분	설명
T&M No	• TM-VT-D4
T&M 명	• 경계값 분석 (Boundary Value Analysis)
주요개념(Concept)	• 동등 분할의 경계부분에 해당하는 입력값에서 결함이 발견된 확률이 경험적으로 높기 때문에 결함을 방지하기 위해 경계 값까지 포함하여 테스트 하는 기법
적용목적(Objective)	• 파라미터(Parameter) 제한 또는 경계에서 발생하는 에러 제거
적용단계(Phase)	• 소프트웨어 요구사항 • 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계 • 통합
주요 기술 (Techniques)	• Single Faultbased Testing • Robustness Testing • Worst Case Testing • Robustness Worst Case Testing

적용 지침(Guideline)

- 많은 수의 오류들이 입력 도메인의 “중앙” 보다는 경계에서 발생하며 이를 근거로 경계값분석(boundary value analysis) 테스팅 기법이 개발되었다.
- 경계값 분석은 동등 분할을 보완하는 테스트 케이스 설계 기법으로 동등 클래스에 속하는 입력 값 중에서 임의로 값을 선택하지 않고 클래스의 경계에 있는 테스트 입력 집합 중에서 입력값을 선택하여 테스트 케이스를 설계한다.
- 경계값 분석은 동등 분할과 마찬가지로 모든 테스트 레벨, 모든 테스트 형태, 모든 테스트 분류에 적용될 수 있다.
- 명세에 있는 입/출력 값을 활용하기 때문에 명세기반 테스트 기법이다.
- 결함 발견율이 높고, 적용하기 쉬운 장점이 있어 가장 많이 사용되는 테스트 케이스 설계 기법 중에 하나이다.
- 테스팅의 강도에 따라 Single Fault based Testing, Robustness Testing, Worst Case Testing, Robustness Worst Case Testing의 기법을 선택하여 적용한다.

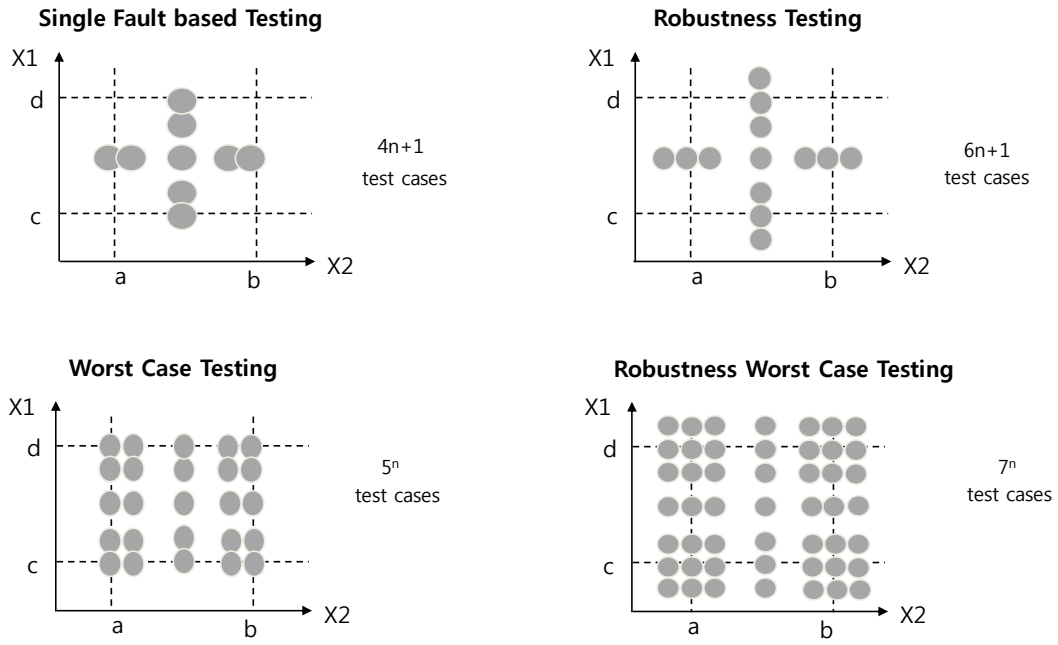


그림 228 경계값 분석의 기법들

- 경계값 분석 기법은 종종 동등 분할의 확장으로 여겨지며, 동등 분할과 동일한 방식으로 커버리지를 보장한다.
- 경계값 분석은 테스트 데이터 선택에도 사용될 수 있다.
- 제로 제수, 빈 스택, 인쇄제어 문자, 널 행렬 등의 값은 오류의 원인이 되니 사용 시 특별한 주의를 해야 한다.

적용예시(Example)

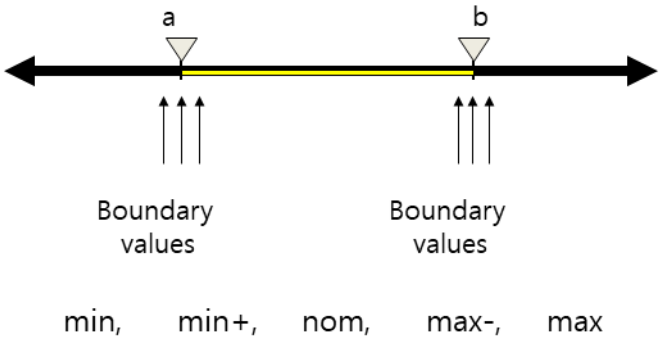


그림 229 경계값 분석

1. 입력 조건이 값 a와 b로 경계 범위를 지정한 경우, 테스트 케이스는 값 a와 b 그리고 a와 b의 바로 위, 아래 값을 갖도록 설계해야 한다.

2. 입력 조건이 많은 개수의 값들을 지정한 경우, 최소값과 최대값을 검사하는 테스트 케이스들을 개발해야 한다. 최대값과 최소값 바로 위, 아래 값들도 테스트해야 한다.

3. 위의 1,2번 항목을 출력 조건에 적용한다.
예를 들어, 분석 프로그램의 출력으로서 온도와 압력의 관계를 나타내는 표가 필요하다고

가정하자. 표에 들어갈 데이터 개수가 가장 많은 경우(그리고 가장 적은 경우)의 출력 보고서를 생산하는 테스트 케이스들을 설계해야 한다.

4. 내부 프로그램 데이터 구조의 범위가 미리 정해진 경우, 데이터 구조를 경계점에서 검사하는 테스트 케이스를 설계해야 한다.

- 입력 파라미터로 정수형 nX, NType를 가지는 FnSwitch(int nX, int nType) 함수는 입력값에 따라 특정 트리거 이벤트를 발생한다.
- nX 입력값의 범위는 0 ~ 255 이고 nType 입력값의 범위는 1 ~ 20 인 경우 경계값 분석 기법을 활용한 Fnswitch 함수의 테스트 케이스 설계 예이다.

표 321 경계값 분석을 활용한 테스트 케이스의 예

NO	테스트 케이스	비고
1	▪ TC_Fnswitch (-1, 10)	nX : min , nType : normal
2	▪ TC_Fnswitch (1, 10)	nX : min+ , nType : normal
3	▪ TC_Fnswitch (254, 10)	nX : max- , nType : normal
4	▪ TC_Fnswitch (256, 10)	nX : max+ , nType : normal
5	▪ TC_Fnswitch (120, 0)	nX : normal , nType : min
6	▪ TC_Fnswitch (120, 2)	nX : normal , nType : min+
7	▪ TC_Fnswitch (120, 19)	nX : normal , nType : max-
8	▪ TC_Fnswitch (120, 21)	nX : normal , nType : max+

적용 시 고려사항(Considerations & Constraints)

- 경계값분석 및 동등분할 테스트 케이스 설계 기법은 다음과 같은 한계점이 있다.
- 일련의 동작에 대한 조합을 테스트하기에는 적합하지 않음
- 입력조합이 상호간에 의존성이 없다는 가정에서만 적합
- 동등 분할 경계에 있는 값(유효 값)을 테스트 대상이 되는 요소로 선택하여 테스트 케이스를 도출해야 한다.
- 입력 영역뿐만 아니라 출력(결과) 영역도 고려해 테스트 케이스 도출해야 한다.

C-6.7 동등 클래스 및 입력 분할 테스팅

구 분	설 명
T&M No	• TM-VT-D18
T&M 명	• 동등 클래스 및 입력 분할 테스팅 (Equivalence Classes and Input Partition Testing)
주요개념(Concept)	• 입력 데이터의 영역을 유사한 특징을 가진 그룹으로 분할하여, 각 클래스에서 대표 값을 도출하여 테스트 하는 기법
적용목적(Objective)	• 최소한의 테스트 데이터를 사용하여 소프트웨어를 충분히 테스트 하기 위함
적용단계(Phase)	• 소프트웨어 요구사항 • 소프트웨어 컴포넌트 설계 • 통합
주요 기술 (Techniques)	• 등가 집합(Equivalence classes) • 명세기반 테스트

적용 지침(Guideline)

- 동등 분할 테스팅은 소프트웨어나 시스템의 입력의 결과로 나타날 결과 값이 동일한 경우 하나의 그룹으로 간주한다. 그리고 해당 그룹 내의 입력값은 내부적으로 같은 방식으로 처리됨을 가정한다.
- 동등 분할 그룹은 유효한 입력 데이터, 즉 허용할 데이터와 유효하지 않은 입력 데이터 모두에 대해 존재할 수 있다.
- 입력 값 이외에도 출력 값, 내부 값, 시간 관련 값(이벤트 이전과 이후), 모듈 간 인터페이스 파리미터에 대해서도 동등 분할을 정의할 수 있다.
- 동등 분할 테스트는 모든 레벨에 적용할 수 있다.
- 명세에 있는 입/출력 값을 활용하기 때문에 명세기반 테스트 기법이며 구조 기반, 경험 기반 기법에도 해당된다.

적용예시(Example)

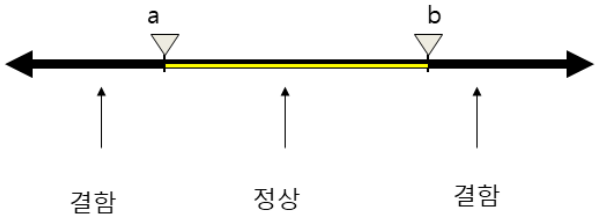


그림 230 동등 분할 기법

- 입력 파라미터로 정수형 nX, NType를 가지는 Fnswitch(int nX, int nType) 함수는 입력 값에 따라 특정 트리거 이벤트를 발생한다.
- nX 입력값의 범위는 0 ~ 255 이고 nType 입력값의 범위는 1 ~ 20 인 경우 동등 분할 기법을 활용한 Fnswitch 함수의 테스트 케이스 설계 예이다.

표 322 동등 분할 기법을 활용한 Fnswitch 함수의 테스트 케이스 설계 예

NO	테스트 케이스	비고
1	▪ TC_Fnswitch (100, 10)	nX : 정상 , nType : 정상
2	▪ TC_Fnswitch (-20, 10)	nX : 결함 , nType : 정상
3	▪ TC_Fnswitch (101, 20)	nX : 정상 , nType : 결함
4	▪ TC_Fnswitch (1000, 50)	nX : 결함 , nType : 결함

적용 시 고려사항(Considerations & Constraints)

- 경계값분석 및 동등분할 테스트 케이스 설계 기법은 다음과 같은 한계점이 있다.
  - 일련의 동작에 대한 조합을 테스트하기에는 적합하지 않음
  - 입력조합이 상호간에 의존성이 없다는 가정에서만 적합

C-6.8 방어적 프로그래밍

구 분	설 명
T&M No	• TM-AD-D14
T&M 명	• 방어적 프로그래밍 (Defensive Programming)
주요개념(Concept)	• 예상치 못한 입력에도 소프트웨어가 지속적이고 안정적으로 기능 수행을 보장할 수 있도록 고안된 소프트웨어 설계의 한 방법
적용목적(Objective)	• 소프트웨어 내부 로직의 비정상적인 제어 흐름, 데이터 흐름 또는 데이터 값을 실행 중에 탐지하고 이를 미리 결정된 허용가능한 방식으로 처리하는 소프트웨어 설계 및 구현을 위해 적용
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계
주요 기술 (Techniques)	• variables range checked • Input values checked for plausibility • Function parameters to procedures's type, dimension and range checked

적용 지침(Guideline)

- Safety-Critical 소프트웨어 시스템은 "garbage in, garbage out"보다 더 정교한 방법으로 Error를 처리하도록 개발되어야 하며, defensive programming 기술을 적용함으로써 시스템에서 발생 가능한 Error를 용이하게 발견하고 수정할 수 가 있음.
- Safety-Critical 소프트웨어 시스템에서의 "garbage in, garbage out" 는 충분하지 않으며, "garbage in, nothing out", "garbage in, error message out", "No garbage allowed in" 형태로 견고하고 안전하게 설계되어야 함.
- 소프트웨어 기능 실행 중 이상 제어 및 데이터 흐름과 데이터 값의 범위를 확인하고, 소프트웨어 형상의 일관성 유지 등을 확인 할 수 있는 자가 점검(self-checking) 기법을 적용 할 수 있음.

- Data(Variables) range checked

표 323 Data(Variables) range checked 목록

No.	세부 적용 지침	설명
01	Check the values from external sources	▪ 외부로부터 유입되는 데이터 - file data, users, network, external interfaces - 가 유효한 범위에 있는지 check ▪ 특히 numeric value는 허용 범위 내에 있는지, string value는 문자열을 처리하기에 충분히 짧은지 확인 필요 (허용범위내의 string 크기인지 check) ▪ String의 경우, 제한된 범위 내에서 표현하기 위한 것이라면 해당 string이 원래 용도로 유효한지 확인 필요. ▪ Security-critical 시스템의 경우, 시스템을 공격 할 수 있는 데이터의 check 필요 - Attempt buffer overflow - Inject SQL scripts, HTML, XML code
02	Check the values for all function/routine's input parameters	▪ 기본적으로 외부로부터 유입되는 데이터 의 유효성 check와 동일 ▪ 타 모듈에서의 function/routine 호출 시 입력되는 data의 유효성 check ▪ input parameter의 type, dimension 의 유효성 check

No.	세부 적용 지침	설명
03	Decide how to handle bad input value	▪ 외부/내부 입력 data가 유효하지 않다고 판정되는 경우, bad data의 처리 로직의 적용이 필요함(Error Handling Techniques)

- Assertion check(Plausibility checked)
  - Assertion은 안전 관련 프로그램의 실행 시 프로그램 내부의 로직 및 흐름을 자체적 Check 할 수 있는 프로그램으로 Assertion이 True일 경우, 설계자가 의도 및 예상대로 동작한다는 것을 의미하고, False 일 경우, 이는 프로그램에서 예기치 않은 오류가 있음을 의미하므로 보다 안전하고 신뢰성을 보장 할 수 있는 유용한 소프트웨어안전 설계 기법임.

표 324 Assertion check(Plausibility checked) 목록

No.	세부 적용 지침	설명
01	Use assertions for conditions that should never occur	▪ 발생예상 조건에 Error 처리 코드를 적용하며, 절대로 발생해서는 안 되는 조건에 대해 Assertion을 적용
02	Avoid putting executable code into assertions	▪ Assertion 적용 시 코드를 실행 코드를 직접 삽입 지양
03	Use assertion to verify pre-conditions and post-conditions	▪ Assertion 적용 시 기능 동작 수행에 대한 사전/사후 조건을 check

- Error Handling Techniques
  - Assertion이 절대로 발생해서는 안 되는 Error 처리를 위해 사용하다면 예상되는 Error에 대해서는 이를 안전하게 처리하는 Error 처리 기술이 필요함. 다음과 같은 다양한 Error 처리 전략이 있으며 상황에 따라 적절히 적용이 가능함.

표 325 Error Handling Techniques 목록

No.	세부 적용 지침	설명
01	Return a neutral value	▪ 악성 데이터 입력 시 기능수행의 연속성을 위해 안전하고 검증된 데이터를 반환(악성값 판단 시 Default Value - Numeric value => 0, String Value =>Empty String)
02	Return the same value as previous time	▪ 상황에 따라 바로 이전 데이터를 반환.(거래 승인 및 사용자 인증의 경우는 적용 시 주의 필요)
03	Substitute the next piece of valid data	▪ 데이터 스트림을 처리 시 상황에 따라 다음번의 유효한 데이터를 반환. 센서로부터 초당 100개의 데이터 취득 시, 한 번에 유효한 데이터를 취득하기 어려우므로 waiting time 후(1/100 second) 다음 번 데이터를 취득
04	Substitute the closet legal value	▪ 특정 데이터 반환 시 가장 근접한 유효 데이터를 반환. 취득 데이터의 유효 범위를 벗어나는 경우, 각 상한 및 하한 데이터로 변환해 반환.
05	Log a warning message to	▪ 악성 데이터 취득 또는 감지 시 경고 메시지를 파일에

No.	세부 적용 지침	설명
	a file	기록하고 기능을 수행(02, 04 방법과 함께 사용 가능) ▪ Error 로그를 사용하는 경우 안전하게 공개적으로 사용할 수 있는지 또는 암호화해야하는지 여부 고려 필요
06	Return an error code	▪ Error 감지 시 local Error 처리가 아닌 시스템의 다른 부분에 처리하도록 통지(호출 계층이 아닌 상위 루틴에서 Error 처리) ▪ Error 유형에 따른 상태 변수 지정, function 반환값으로 Error 유형/상태 반환, 개발 Language 에 따른 예외 메커니즘 적용(local 또는 시스템 의 다른 계층 처리 결정)
07	Call an error processing routines/procedures	▪ Error 처리 시 전역 Error 처리 루틴을 이용해 오류 처리를 중앙 집중화 ▪ Error 처리 책임을 중앙 집중화하여 디버깅을 쉽게 할 수 있는 장점이 있음. ▪ 시스템의 코드를 다른 시스템에서 재사용 시 재사용 코드와 함께 오류 처리 루틴을 가지고 와야 하는 단점 존재
08	Display error message	▪ Error 감지 시 사용자에게 메시지 표출. Error 처리 부담을 최소화 할 수 있는 방법
09	Shut down or Reset	▪ Error 감지 시 시스템을 재시작 또는 강제로 중지 시키는 방법. Safety-critical 시스템에 유용할 수 있는 방법

표 1 Defensive Programming

적용예시(Example)

- Input parameters range checked coding 예시

```
function int range_check(float para1, float para2, int para3)
    {
      // range checked
      ...
      ...
    }
```

- Assertion checked coding 예시

```
function int velocity (double int latitude, double int longitude,
                       double int elevation)
    {
        int velocity_calc

        // pre-conditions
        Debug.Assert (-90 <= latitude && latitude <=90)
        Debug.Assert (0 <= longitude && longitude <360)
        Debug.Assert (-500 <= elevation && elevation
```

```
                    <= 75000)
        ….

        // post-conditions
        Debug.Assert (0 <= return value and return value
                    <=600)

        // return value
        return velocity_calc
    }
```

적용 시 고려사항(Considerations & Constraints)

- defensive programming은 안전하고 견고한 소프트웨어 시스템의 품질을 제고할 수 있는 유용한 기법이지만, 가장 좋은 형태의 defensive 코딩은 처음부터 코드에 이를 삽입하는 것이 아니라, 결함 유입을 방지 할 수 있는 사전 활동을 강화하는 것임. 이와 같은 활동은 defensive programming 보다 더 높은 우선순위로 적용되어야 하며, defensive programming 기법과 병행함으로써 보다 효과적으로 소프트웨어 시스템의 안전성을 확보할 수 있음.
  - ✓ 반복적 개발 및 설계(Using Iterative Design)
  - ✓ 실제 코딩 전 의사코드 개발(pseudo-code before code)
  - ✓ 코딩 전 테스트 케이스 설계(test cases before code)
  - ✓ 설계 검토(design inspections)

C-6.9 정보 은닉 / 캡슐화

구 분	설 명
T&M No	• TM-AD-D33
T&M 명	• 정보 은닉 / 캡슐화 (Information Hiding/Encapsulation)
주요개념(Concept)	• 정보 은닉이란 데이터의 무결성을 위해 외부 소프트웨어 컴포넌트가 접근하지 못하도록 데이터를 숨기는 것 • 캡슐화란 데이터와 이에 접근하기 위한 오퍼레이션을 하나로 묶는 것
적용목적(Objective)	• 소프트웨어의 견고함과 유지보수성을 향상
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 설계 단계: UML의 클래스의 표기법 • 구현 단계: 객체 지향 언어의 접근 지정자

적용 지침(Guideline)

- 정보 은닉과 캡슐화의 필요성
  - 허락되지 않은 방법으로 접근하여 데이터를 변경하는 것을 막기 위하여 데이터 구조를 숨기고 약속한 방법으로 데이터를 변경하기 위해 인터페이스를 제공하여 데이터의 무결성과 데이터의 정당한 접근을 보장
  - 다른 소프트웨어의 동작에 영향을 미치지 않고 내부 구조를 변경하거나 추가할 수 있음

- 설계 단계 적용 방법
  - 데이터와 데이터에 접근하기 위한 인터페이스(오퍼레이션 세트)를 정의
  - 데이터 은닉과 인터페이스 공개
    - ✓ 데이터는 데이터의 접근제어 수준을 은닉으로 하여 내부에서만 데이터에 접근 가능하도록 함
    - ✓ 외부에 제공할 인터페이스는 접근제어 수준을 공개로 하여 외부에서 접근 가능하도록 함

- 구현 단계 적용 방법
  - 언어에 따라 다른 방법을 적용
  - 객체 지향 언어
    - ✓ C++, C#, Java 등의 객체 지향 언어는 접근 지정자(public, protected, private)를 통해 데이터와 메소드(오퍼레이션)의 접근제어 수준을 결정
    - ✓ 접근 지정자
      - ◆ public: 외부에서 접근할 수 있게 공개
      - ◆ protected: 외부에서는 접근할 수 없게 은닉되었으나 이를 상속받은 클래스 내에서는 접근 가능
      - ◆ private: 외부에서 접근할 수 없게 은닉됨
  - 절차적 언어
    - ✓ C언어와 같은 절차적 언어는 언어적으로 정보은닉과 캡슐화를 지원하지 않으나 기본 원칙으로 사용하여 적용

✓ 기본 원칙
- 데이터와 데이터에 접근 가능한 오퍼레이션 세트는 하나의 파일 세트(헤더 파일과 소스파일)에 작성
- 전역 변수 대신 로컬 변수 사용
- 소프트웨어 생명주기 동안 데이터가 유지되어야 하는 경우, 전역 변수 대신 정적 변수를 사용하여 외부 접근을 제어
- 데이터는 인터페이스(오퍼레이션 세트)로 데이터 접근이 가능하도록 헤더 파일(.h)에 선언하고 소스파일(.c)에 구현

- 정보은닉과 캡슐화의 특징
  - 캡슐화는 인터페이스로만 데이터에 접근이 가능해야 하므로 정보은닉이 전제되어야 함
  - 외부 컴포넌트가 데이터에 직접 접근하는 것을 차단하여 데이터를 손상과 오류로부터 보호하여 데이터의 무결성을 보장
  - 서비스를 제공하는 제공자와 이를 이용하는 이용자에 대한 역할을 명확히 하여 독립성을 확보
  - 이용자에게 인터페이스를 제공하여 내부 구현과 관계없이 이용할 수 있도록 변경 범위를 제한
  - 인터페이스를 통해 소프트웨어의 객체 혹은 컴포넌트 간의 상호작용을 하여 변경에 대한 영향도를 줄일 수 있고 대상 간의 결합도와 복잡도를 낮춤
  - 캡슐화를 통해 대상을 추상화

적용예시(Example)

- 대상: 신호등 관리자
  - 시퀀스 다이어그램으로 상호작용(오퍼레이션)을 도출 (TM-AD-D67의 예제 참조)

- 설계 단계
  - 데이터와 데이터에 접근하기 위한 인터페이스(오퍼레이션 세트)를 정의
    ✓ 데이터 정의
    - 신호등 관리자는 신호등의 색의 데이터(속성)를 가지고 있기 위해서 currentColor라는 데이터를 정의
    ✓ 오퍼레이션 정의
    - 신호등 관리자의 데이터를 외부에서 변경하기 위한 오퍼레이션 세트 정의

표 326 건널목 제어 시스템의 오퍼레이션 목록

오퍼레이션	설명
▪ switchRed	▪ 신호등을 빨간불로 변경
▪ switchYellow	▪ 신호등을 노란불로 변경
▪ switchOff	▪ 신호등을 종료

  - 데이터 은닉과 인터페이스 공개
    ✓ currentColor 접근 제어를 은닉으로 설정

표 327 건널목 제어 시스템의 데이터의 접근제어 목록

데이터	접근제어	설명
▪ currentColor	▪ 은닉	▪ 현재 신호등의 색

✓ 오퍼레이션의 세트는 접근 제어를 공개로 설정

표 328 건널목 제어 시스템의 오퍼레이션의 접근제어 목록

오퍼레이션	접근제어	설명
▪ switchRed	▪ 공개	▪ 신호등을 빨간불로 변경
▪ switchYellow	▪ 공개	▪ 신호등을 노란불로 변경
▪ switchOff	▪ 공개	▪ 신호등을 종료

- UML 표기

✓ 신호등 관리자를 UML의 클래스 표기법으로 표현하면 다음과 같다.

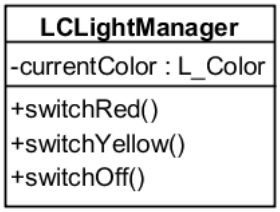


그림 231 신호등 관리자의 UML 표기법

- 구현 단계
  - 객체 지향 언어

✓ 객체 지향 언어에서 지원하는 접근지정자를 설정하여 데이터를 은닉하고 인터페이스 공개한다.

표 329 신호등 관리자의 java 코드

```
public class LCLightManager
{
        private L_Color currentColor;

        public void switchRed()
        {
                currentColor = L_Color.RED;
        }
                public void switchYellow()
        {
                currentColor = L_Color.YELLOW;
        }
        public void switchOff()
        {
                currentColor = L_Color.OFF;
        }
}
```

- 구조적 언어
  - ✓ 원칙을 적용하여 데이터를 은닉하고 인터페이스(오퍼레이션 세트)를 공개한다.

표 330 신호등 관리자의 c언어 코드

LCLightManager.h

```
typedef enum L_Color{OFF, YELLOW, RED} L_Color;

void LCLightManager_switchRed();
void LCLightManager_switchYellow();
void LCLightManager_switchOff();
```

LCLightManager.c

```
#include “LCLightManager.h“

static L_Color currentColor = OFF;

void LCLightManager_switchRed()
{
        currentColor = RED;
}
void LCLightManager_switchYellow()
{
        currentColor = YELLOW;
}
void LCLightManager_switchOff()
{
        currentColor = OFF;
}
```

적용 시 고려사항(Considerations & Constraints)

- 데이터 접근방식에 대한 일반적인 전략이 존재할 경우 정보은닉/캡슐화는 필수 사항은 아님
- 정보은닉/캡슐화를 데이터 접근 방식으로 결정했을 경우, 아키텍처 설계 단계와 컴포넌트 설계 및 구현 단계에 모두 적용하도록 함
- 객체 지향 언어와 같이 태생적으로 정보은닉과 캡슐화를 지원하는 언어가 있으나, 지원하지 않는 언어는 정보은닉/캡슐화를 기본 원칙을 적용가능
- 정보 은닉은 캡슐화의 전제조건으로 전역 변수 난발 등으로 정보은닉을 제대로 하지 않을 경우 정보은닉/캡슐화를 통해 얻고자 하는 효과를 얻을 수 없음

C-6.10 모듈 방식

구 분	설 명
T&M No	• TM-AD-D38
T&M 명	• 모듈 방식 (Modular Approach)
주요개념(Concept)	• 소프트웨어 복잡성을 제한하기 위해 소프트웨어를 이해하기 쉬운 작은 부분으로 분해하는 방법
적용목적(Objective)	• 모듈화 접근방법은 소프트웨어 설계 및 구현 시 프로그램의 기능을 독립적인 부품형태의 모듈로 분리함으로써 유지보수와 타 프로그램에서의 코드 재사용을 용이하게 하기 위해 적용하는 설계방법
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술 (Techniques)	• 해당사항 없음

적용 지침(Guideline)

- 모듈화 접근방법은 소프트웨어 개발 시 특정 문제 해결함에 있어 그대로 놓고 해결하는 것은 매우 어려운 일이므로 일반적으로 작은 단위로 쪼개어 그것을 하나씩 해결하는 “분할 및 정복”이라는 문제해결 전략을 적용해 소프트웨어를 실제로 개발할 수 있는 작은 단위로 나누는 것 기법이며 이는 ’모듈화(Modularization)’라고 한다.

- 모듈화란 프로그램의 기능을 독립적인 부품형태의 모듈로 분리함으로써 유지보수와 타 프로그램에서의 코드 재사용을 손쉽게 하는 소프트웨어 설계 기법을 말한다.

- 소프트웨어를 구성하는 모듈은 ’소프트웨어 시스템의 구조를 이루는 기본적인 구성단위’라고 할 수 있으며, ’하나 또는 몇 개의 논리적인 기능을 수행하기 위한 명령어들의 집합’이라고도 정의 할 수 있음. 따라서 독립 프로그램도 하나의 모듈이 될 수 있고, 함수들도 하나의 모듈로 구성할 수 있으며, 다양한 크기의 집합으로 모듈을 구성할 수 있음.(예 - 라이브러리 함수, 서브루틴, 프로시저, 객체, 메소드 등)

- 모듈은 다음과 같은 주요 특징들을 가지고 있음.
  - ✓ 독립적인 기능을 갖는 단위(unit)이다.
  - ✓ 식별 가능한 유일한 이름을 가져야 한다.
  - ✓ 독립적으로 컴파일이 가능하다.
  - ✓ 모듈에서 또 다른 모듈을 호출할 수 있다.
  - ✓ 다른 프로그램에서도 모듈을 호출할 수 있다.

- 모듈화는 소프트웨어 프로젝트의 설계, 코딩 및 유지 보수 단계에 대한 몇 가지 규칙을 포함하며, 이 규칙은 설계 중에 사용 된 설계 방법에 따라 다를 수 있으며 일반적으로 다음 규칙을 포함함.
  - ✓ 모듈/컴포넌트는 충족시킬 단일의 잘 정의 된 작업 또는 기능을 가져야한다.

✓ 모듈/컴포넌트 간의 연결은 제한적이고 엄격하게 정의되어야하며, 하나의 모듈/컴포넌트에서의 일관성은 강해야한다.

✓ 모듈/컴포넌트의 여러 수준을 제공하여 서브프로그램 집합을 구축해야 한다.

✓ 서브프로그램은 단일 entry와 단일 exit 만 가져야한다.

✓ 모듈/컴포넌트는 인터페이스를 통해 다른 모듈/컴포넌트와 통신해야합니다. 전역 변수 또는 공통 변수가 사용되는 경우, 그것들은 잘 구조화되어야하고, 접근은 통제되어야하며, 각각의 경우에 그 사용이 정당화되어야 한다.

✓ 모든 모듈/컴포넌트 인터페이스는 완전하게 문서화되어야한다.

✓ 모든 모듈/컴포넌트 인터페이스는 모듈/컴포넌트에 필요한 최소한의 매개변수를 포함해야한다.

✓ 매개변수 개수의 적절한 제한이 명시되어야하며, 전형적으로 5개 이하로 제한한다.

- 모듈 방식이 잘 반영된 소프트웨어를 설계하기 위해서는 다음과 같은 주요 원칙을 적용해 설계를 진행하는 것을 권장한다.

주요 원칙	설명
정보 은닉 (Information Hiding)	▪ 인터페이스를 프로그램 내부와 분리함으로써, 인터페이스를 통해 모듈로 구성된 프로그램은 프로그램 내부변경에 따른 재작업 감소
낮은 결합도/높은 응집도 (Low Coupling / High Cohesion)	▪ 모듈은 서로 독립적으로 낮은 결합도를 유지할 수 있도록, 함께 자주 사용되는 기능은 동일한 모듈에 속하도록 함으로써 모듈 내부의 높은 내부 응집력을 갖도록 설계
설계 변경 용이성 (Design for change)	▪ 변경 가능성이 있는 데이터에 대해 매개변수와 상수를 활용해 설계 변경에 대한 용이성 확보

적용예시(Example)

- 시스템을 구성하는 모듈 간의 의존도가 높게 설계된 시스템(Tight coupling)은 하나의 모듈 변경이 다른 모듈에 영향을 미침으로써 재작업, 유지보수 등의 비용을 증가 시킬 수 있으며, 특히 데이터 또는 변수, 제어 데이터 모듈 간 공유는 모듈간의 강결합을 야기할 수 있음.

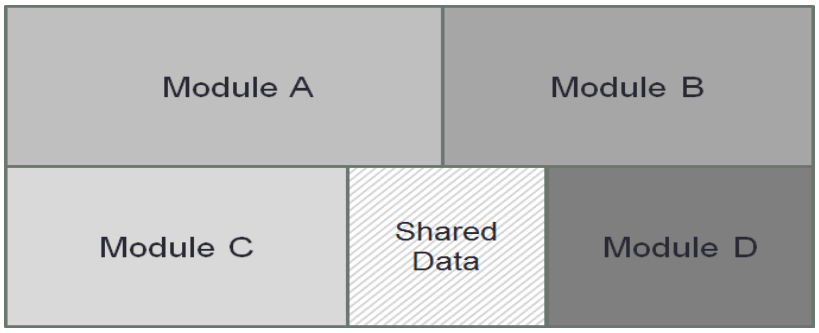


그림 232 강한 결합(Tight Coupling)으로 설계된 프로그램 예시

- 시스템을 구성하는 모듈 간의 의존도를 적절히 유지함으로써(Loose coupling) 하나의 모듈 변경이 다른 모듈에 미치는 영향을 최소화하고, 모듈간의 독립성을 보장 할 수 있음.

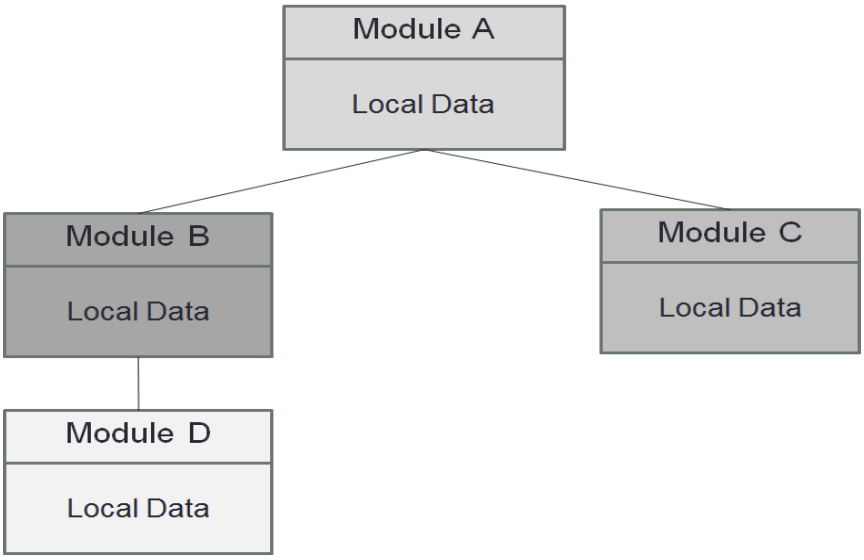


그림 233 약한 결합(Loose Coupling)으로 설계된 프로그램 예시

적용 시 고려사항(Considerations & Constraints)

- 해당사항 없음

C-6.11 코딩 표준 및 형식 가이드

구 분	설 명
T&M No	• TM-AD-D15
T&M 명	• 코딩 표준 및 형식 가이드 (Coding Standards and Style Guide)
주요개념(Concept)	• 코딩 표준은 프로그래밍 언어를 사용할 때 만들어질 수 있는 잠재적인 고장들을 회피하기 위한 규칙과 제약의 집합 • 코딩 스타일 가이드는 프로젝트에 대한 공통적이고 일관된 형식을 가이드 • 코딩 표준과 스타일 가이드는 다수의 프로그래머에 의해서 개발된 코드의 이해와 유지보수를 쉽게 만들고 여러 사람들이 동일한 프로그램의 개발에서 협력하는 것을 더욱 용이하게 함
적용목적(Objective)	• 설계 문서와 생산된 코드의 형식과 품질을 보장 • 프로그래밍의 일관성과 오류 회피를 위한 표준 설계 방법을 준수시키는 것이 목적
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술(Techniques)	• 동적인 객체 금지(No Dynamic Objects) • 동적인 변수 금지(No Dynamic Variables) • 제한적인 포인트 사용(Limited Use of Pointers) • 제한적인 재귀의 사용(Limited Use of Recursion) • 무조건적인 점프 금지(No Unconditional Jumps)

적용 지침(Guideline)

- 코딩 표준은 다음의 항목을 포함 가능
  - 언어 타당성(Language justification)
  - 범위와 기본 표준(Scope and base standard when available)
  - 기본 표준 - 코딩 스타일 가이드(Coding Style Guide)
  - 코딩 표준 변경을 위한 절차(Procedure for changing the coding standard)
  - 잠재적인 결함 분석 및 권장하는 처리(Analysis of the potential faults and recommended treatment)
  - 결함을 피하기 위한 제한사항(Restrictions to avoid the fault)
  - 이식성(portability)

- 언어 타당성
  - 각 프로젝트의 특성에 따라 언어를 선택할 수 있으며, 언어를 선택한 근거 및 타당성을 기술 가능

- 범위와 기본 표준
  - 코딩 표준의 적용 범위와 코딩의 기본 표준을 기술
  - 특정 도메인을 위한 언어는 기본 표준을 사용 불가능
  - 기본 표준 - 코딩 스타일 가이드(Coding Style Guide)는 다음을 포함 가능
  - 들여쓰기 규칙
    - ✓ 괄호 규칙
      - ◆ 모든 if, while 및 do 명령문은 중괄호가 있거나 한 줄에 작성
      - ◆ 닫는 중괄호에 주석을 추가하면 시작 중괄호를 찾을 필요가 없기 때문에

가독성이 향상됨

- 함수처럼 보일 수 있으므로 키워드 옆에 괄호 사용 금지
- 함수 이름 옆에 괄호를 넣기
- 필요하지 않을 경우 return 문에 괄호를 사용 금지
-

- 명명 규칙
  - ✓ 파일명, 타입명, 함수 이름, 구조체 이름, 변수명(전역 변수, 포인터 변수, 로컬 변수 등), 상수명, define, 매크로 명 등의 명명 규칙 및 예외 사항
- 주석 규칙
  - ✓ 모듈의 작성자, 버전, 매개 변수, 반환값, 현재 코드 블록이 무엇을 하는지 더 잘 이해할 수 있도록 파일/함수/변수에 대한 문서의 링크, 코드에서 발견된 버그 보고, 상태(함수가 더 이상 사용되지 않음) 등을 나타냄
- 구성요소 간의 일관성을 유지하기 위한 필요 사항
  - ✓ 한 줄 당 최대 문자 (화면과 출력을 고려하여 작성)
  - ✓ 한 줄 당 하나의 문장만 작성
  - ✓ 매크로 대신 인라인 함수 사용
  - ✓ 모든 변수의 초기화

- 코딩 표준 변경을 위한 절차
  - 코딩 표준의 변경 시 누가 어떤 절차로 변경할 지에 대한 프로세스를 정의 가능

- 잠재적인 결함 분석 및 권장하는 처리
  - 정적분석 참고
  - 제어 흐름 분석
    - ✓ 접근할 수 없는 코드
    - ✓ 무한루프
    - ✓ 잘 못 구조화된 코드
  - 데이터 흐름 분석
    - ✓ 할당되지 않는 변수
    - ✓ 코드 누락
    - ✓ 중복 코드

- 결함을 피하기 위한 제한사항
  - 동적인 객체 금지(No Dynamic Objects)
    - ✓ 동적인 객체를 사용할 경우 사용 후 명시적으로 객체를 삭제(delete)하여 메모리를 반납해야 함
    - ✓ 명시적으로 동적인 객체를 반납하지 않는 경우 메모리 누수(memory leak)가 발생
    - ✓ 동적 객체를 반납 후 다시 사용하려 할 경우 오류 발생 (dangling reference)
    - ✓ 동적인 객체를 사용할 경우 컴파일 타임에 오류를 발견하기 용이하지 않음
  - 동적인 변수 금지(No Dynamic Variables)
    - ✓ 런타임에 메모리가 할당되므로 예기치 않은 버퍼 오버플로가 발생 가능
    - ✓ 명시적으로 동적인 변수를 반납하지 않는 경우 메모리 누수(memory leak)가 발생
    - ✓ 동적 변수를 반납 후 다시 사용하려 할 경우 오류 발생 (dangling reference)
    - ✓ 동적인 변수를 사용할 경우 컴파일 타임에 오류를 발견하기 용이하지 않음
  - 제한적인 포인터 사용(Limited Use of Pointers)

✓ 포인터를 사용할 경우 모든 메모리의 데이터에 접근할 수 있는 강력한 기능
✓ 변수 포인터와 함수 포인터로 유연한 프로그래밍이 가능함
✓ 반면, 포인터를 계산하여 데이터에 쉽게 접근할 수 있으나 실수로 엉뚱한 데이터나 할당되지 않은 메모리에 접근하여 잘못된 데이터를 읽거나 조작하여 프로그램이 오작동 가능
✓ 신중한 사용이 필요

- 제한적인 재귀의 사용(Limited Use of Recursion)
  ✓ 재귀함수란 자기 자신을 재 참조하는 함수
  ✓ 문제점 발생 시 파악이 용이하지 않음
  ✓ 무한 재귀로 오버플로가 발생 가능
  ✓ 신중한 사용이 필요
- 무조건적인 점프 금지(No Unconditional Jumps)
  ✓ 점프는 명령문의 순차적 실행을 뛰어넘어 프로그램의 다른 지점에서 실행됨
  ✓ 점프 대상이 범위를 벗어난 경우 점프는 자동 변수의 파괴를 초래
  ✓ 명시적인 goto문을 사용하지 않아도 break, continue, return이 호출되면 goto가 발생
  ✓ goto문 을 많이 사용하는 코드는 가독성이 떨어짐
  ✓ 심하게 중첩 된 루프를 빠르게 종료하는 것과 같은 명확한 이점을 제공하는 경우에만 사용하도록 함
  ✓ goto 문 을 사용하는 모든 C 프로그램은 명령 없이도 작성 가능

- 이식성
  - 오류 없이 가능한 효율적으로 특정 환경에서 작동하는 것으로 가정하여 작성된 소프트웨어 프로그램을 다른 환경으로 이식하는 방법을 기술

적용예시(Example)

- 동적인 객체 금지(No Dynamic Objects)
  - 런타임에 확정되는 동적 객체를 사용하는 대신 컴파일 타임에 확정되는 정적 객체를 사용

표 331 동적인 객체 금지 예제

동적 객체 사용	정적 객체 사용
LCLightManager* lightManager = new LCLightManager();	LCLightManager lightManager;

- 동적인 변수 금지(No Dynamic Variables)
  - 런타임에 확정되는 동적 변수를 사용하는 대신 컴파일 타임에 확정되는 정적 변수를 사용

표 332 동적인 변수 금지 예제

동적 변수 사용	정적 변수 사용

void func(int n) {
int a[n]; /* 가변 길이 배열은 부적합 변수 n에 매우 큰 수가 올 경우 버퍼 오버플로의 위험*/
}
	#define MAX 1024
	void func(void) {
	int a[MAX]; /* 최대 길이의 배열 확보 준수 */
	}

- 제한적인 포인터 사용(Limited Use of Pointers)
  - 동적 변수를 할당하여 사용한 후 반납한 동적 변수를 다시 사용하려고 할 경우 Dangling Pointer 발생
  - 포인터 대신 배열(정적 변수)을 사용하여 회피 가능

표 333 제한적인 포인터 사용 예제

동적 변수 사용	배열 사용

char *dp = malloc(MAX_SIZE);
char *temp = dp;
/* df, temp를 사용한 로직 */
free(temp); /* dp는 dangling pointer */
/* df를 사용하려 할 경우 오류 */
dp = NULL; /* 명시적으로 dp에 NULL할당 dangling pointer 해소 */
	char dp[MAX_SIZE];
	char temp[MAX_SIZE];
	for (int i = 1; i < MAX_SIZE; i++)
	{
	temp[i] = dp[i];
	}
	/* df, temp를 사용한 로직 */

- 제한적인 재귀의 사용(Limited Use of Recursion)
  - 재귀 함수는 반복 제어문으로 변경 가능

표 334 제한적인 재귀의 사용 예제

재귀 함수	반복 제어문

int factorial(int num)
{
int result = 0;
if (num == 1)
result = 1;
else
result = num * factorial(num - 1);
return result;
}
	int factorial(int num)
	{
	int result = 1;
	for (int i = 1; i < num; i++)
	{
	result = result * i;
	}
	return result;
	}

  - 모든 재귀 함수가 제어문으로 쉽게 변경되지 못함
  - 재귀 함수를 사용할 경우 이해가 용이하고 재귀 함수에 더 적합한 경우 제한적으로 재귀 함수를 사용

- 무조건적인 점프 금지(No Unconditional Jumps)
  - 점프는 제어문으로 대체 가능

표 335 무조건적인 점프 금지 예제

무조건적인 점프 사용	제어문 사용

int sum()
{
int i = 1, sum = 0;
loop_start:
sum = sum + i;
if (i <= 100)
{
i = i + 1;
goto loop_end;
}
goto loop_start;
loop_end:
return sum;
}
	int sum()
	{
	int sum = 0;
	for (int i = 1; i <= 100; i++)
	{
	sum = sum + 1;
	}
	return sum;
	}

- 중첩된 루프를 빠르게 종료할 때와 같은 상황에 제한적으로 사용

표 336 점프의 제한적인 사용 예제

```
int loop_exit(int color[], int status[])
{
        int result = SUCCESS;
        for (int i = 0; i < 100; i++)
        {
                for (int j = 0; j < 100; j++)
                {
                        if (color[i] == -1 && status[j] == -1)
                        {
                                result = FAIL;
                                return result; // 한번에 2개의 루프를 빠져나옴
                        }
                }
                return result;
        }
}
```

적용 시 고려사항(Considerations & Constraints)

- 작성 시 고려사항
  - 코딩 표준과 스타일 가이드라인은 설계 단계나 그 이전에 정의되어 있어야 함

- 적용 시 이점
  - 프로젝트에 공통되고 일관된 스타일 가이드라인 적용 시 가독성이 높아짐
  - 프로젝트 개발자들의 협업이 용이
  - 사용 언어의 잠재적 결함을 피할 수 있음

- 적용 시 유의사항
  - 프로그램 개발자는 구현 단계에서 코딩 표준과 스타일 가이드라인을 준수해야 함

- 코딩 표준과 스타일 가이드라인은 작성자(writer)의 생산성보다는 읽는 이(reader)의 가독성을 중요시 함
- 기존의 코드와의 일관성 유지가 중요

• 적용 준수 확인방안
- 코딩 표준과 가이드라인은 준수 여부를 확인하기 위한 검사가 필요
- 코딩 표준의 준수 여부 검사는 수동으로도 가능하나 프로그램의 규모가 클 경우 신속한 검사가 불가능함으로 정적 분석 도구의 사용을 권장

• 적용범위
- 표준 및 가이드라인은 코딩 외에도 문서화 등에서 사용 가능

C-6.12 분석 가능한 프로그램

구 분	설 명
T&M No	• TM-AD-D2
T&M 명	• 분석 가능한 프로그램 (Analysable Programs)
주요개념(Concept)	• 정적 분석 기법으로 분석하기 쉬운 프로그램 개발
적용목적(Objective)	• 프로그램 분석을 쉽게 할 수 있도록 프로그램을 설계 • 프로그램 분석에 기초하여 완전하게 시험 가능해야 함
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술 (Techniques)	• 구조적 프로그래밍 기법

적용 지침(Guideline)

- 컴포넌트 제어 흐름은 구조화된 구성으로 이루어져야 함

표 337 구조화 프로그램의 구성

구성요소	상세
순차 (concatenation)	▪ 순서에 따라 구문을 수행 ▪ 좌측에서 우측으로, 상단에서 하단의 순서로 구문을 수행
반복 (repetition)	▪ 프로그램이 특정 상태에 도달할 때까지 구문들을 반복적으로 수행 ▪ while, do/while, for문과 같은 반복 제어문으로 반복을 처리
선택 (selection)	▪ 프로그램의 상태에 따라 구문들 중 하나를 수행함 ▪ if/else, switch/case문과 같은 선택 제어문으로 선택(분기)을 처리

- 컴포넌트는 작아야 함
  - 작성자는 이해하기 쉬운 크기의 작은 수준으로 나눠 복잡도를 낮춰야 함
  - 컴포넌트는 연관이 있는 기능을 모아 놓은 이해하기 쉬운 정도의 작은 크기로 분해되어야 함
  - 이는 큰 문제를 분할하여 해결하는 전형적인 방법

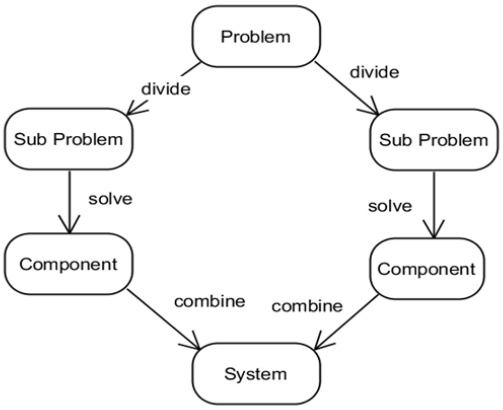


그림 234 문제 해결 방법-분할정복법

- 컴포넌트에 접근 가능한 경로의 개수는 적어야 함
  - 컴포넌트를 접근하는 경로는 인터페이스로 정의함

- 전역적 접근 등의 여러 경로를 허용할 경우 의도치 않은 데이터의 변경이 일어나 무결성을 해칠 수 있음
- 컴포넌트 이용자는 허용하는 인터페이스로만 접근하여야 정보은닉 및 캡슐화의 이점을 얻을 수 있음. (정보은닉 및 캡슐화 참조)
- 컴포넌트의 인터페이스는 한 개 이상일 수 있음

• 개별적인 프로그램들은 가능한 결합도가 낮게 설계되어야 함
  - 프로그램의 결합도가 낮으면 변경 영향도가 낮아짐
  - 컴포넌트에 변경이 발생할 경우 결합도가 낮을 경우, 인터페이스의 변경이 없다면 연관 외부 컴포넌트의 변경 없이 대상 컴포넌트 변경이 가능함

• 입력과 출력 인자(parameter)들 간의 관계는 가능한 단순해야 함
  - 컴포넌트의 입출력인자는 컴포넌트의 접근 경로 즉 인터페이스임
  - 인터페이스를 나타내는 입출력 인자들 간의 관계가 복잡하다면 가독성이 낮아짐
  - 이를 방지하기 위하여 입력과 출력 인자는 가능한 단순해야 함
    ✓ 필요한 최소한의 입력과 출력 인자를 포함해야 함
    ✓ 입력과 출력 인자의 개수는 적절히 제한되어야 함
    ✓ 입력과 출력 인자의 수가 많은 경우 구조체를 사용하여 단순화 함

• 복잡한 계산은 분기 및 반복 결정의 기초로 사용해서는 안 됨
  - 복잡한 계산을 분기나 반복 결정의 조건으로 삼을 경우 코드의 복잡도가 올라감
  - 코드의 복잡도가 올라가면 오류가 발생하기 쉽고 가독성이 낮아짐
  - 분기 및 반복 결정문이 복잡할 경우 함수로 분할

• 분기와 반복 결정들은 컴포넌트 입력 인자들과 단순하게 연관되어야 함
  - 컴포넌트의 입력 인자가 분기와 반복을 결정하기 위한 조건과 연관 될 수 있음
  - 이때 입력 인자를 직접적으로 이용하여 분기/반복 조건을 만들 경우 입력 인자의 변경에 분기/반복문이 영향을 받아 변경될 수 있음
  - 입력 인자와 분기/반복 조건의 조건 연산과 분기/반복문은 나누어 변경 영향도를 줄이고 가독성을 높이는 것이 바람직함

• 서로 다른 데이터 타입들의 맵핑 간의 경계는 단순해야 함
  - 다른 데이터 타입 간의 매핑은 명시적이고 분명해야 함
  - 묵시적인 매핑은 오류를 발생시키기 쉽고 애매하여 가독성이 낮아짐

적용예시(Example)

• 컴포넌트에 접근 가능한 경로의 개수는 적어야 함
  - 교차로 컴포넌트는 이용자에게 IRailCrossingComp, ILightManager라는 2개의 인터페이스를 제공함
  - 이용자는 이 2개의 인터페이스를 통해서만 RailCrossingComp를 이용 가능

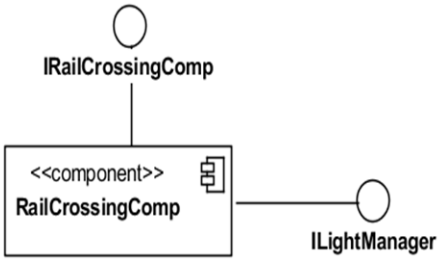


그림 235 인터페이스의 예

- 개별적인 프로그램들은 가능한 결합도가 낮게 설계되어야 함
  - 결합도가 낮으면 컴포넌트 RailCrossingComp의 내부가 변경되더라도 인터페이스 IRailCrossingComp와 ILightManager를 사용하고 있는 외부 컴포넌트 External Component1, External Component2는 변경 없이 사용 가능

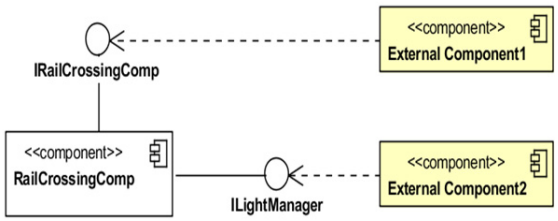


그림 236 인터페이스 사용의 예

- 복잡한 계산은 분기 및 반복 결정의 기초로 사용해서는 안 됨
  - 다음의 반복문의 반복 조건으로 복잡한 계산을 사용하여 코드가 복잡해진다.
  - 이 경우 반복 조건을 계산하는 부분을 다른 함수(isReady)로 분할하여 처리하면 복잡도가 낮아진다.

표 338 복잡한 계산은 분기 및 반복 결정의 기초로 사용하지 않는 예제

부적합	적합

while ((status1 == SUCCESS && status2 == READY) ¦¦ (status2 == READY && status3 == ACTIVE) ¦¦ (status2 == SUCCESS && status4 == NO_SIGNAL))
{
/* ... */
}
	while(isReady())
	{
	/* ... */
	}

- 서로 다른 데이터 타입들의 맵핑 간의 경계는 단순해야 함
  - 평균을 계산하기 위하여 변수 sum을 변수 count로 나눌 때 변수를 묵시적으로 데이터 타입 전환할 경우 결과가 int 형으로 나와 소수점 이하의 값이 버려진다.
  - 이와 같이 묵시적 데이터 타입 전환은 의도치 않은 결과가 나올 수 있으므로 명시적 데이터 타입 전환을 하여야 한다.

표 339 묵시적/명시적 타입 변환의 예제

묵시적 데이터 타입 변환	명시적 데이터 타입 변환

int sum = 7, count = 5;
double avg;
avg = sum / count;/* 묵시적 데이터 타입 변환 */
printf("Value of avg : %f\n", avg);
	int sum = 7, count = 5;
	double avg;
	avg = (double) sum / count; /* 명시적 데이터 타입 변환 */
	printf("Value of avg : %f\n", avg);

- 결과

표 340 묵시적/명시적 타입 변환의 결과

묵시적 데이터 타입 변환	명시적 데이터 타입 변환
Value of avg : 1.000000	Value of avg : 1.400000

적용 시 고려사항(Considerations & Constraints)

- 분석 가능한 프로그래밍은 프로그램 분석이 쉽게 될 수 있도록 즉, 읽는 이(reader)의 이해가 쉬운 프로그램

- 분석 가능한 프로그래밍을 위한 기법
  - 분석 가능한 프로그래밍을 위하여 구조적 프로그래밍 기법을 제안
  - 구조적 프로그래밍 기법이 분석 가능한 프로그래밍의 필요충분조건은 아님
  - 구조적 프로그래밍 기법 이외에도 코딩 표준과 가이드라인, 정보은닉과 캡슐화, 모듈화 기법 등을 적용 가능

- 구조적 프로그래밍은
  - 보통 모듈방식으로 이루어져 복잡도를 낮춤
  - 절차적 프로그래밍 언어에 사용하는 것이 적합하나 모든 프로그래밍 언어에 구조적 프로그래밍 기법을 적용하는 것은 가능함

- 프로그램을 정확하고 신속하게 분석하기 위하여 자동화된 정적 분석 도구를 사용 가능

C-6.13 적합한 프로그래밍 언어

구 분	설 명
T&M No	• TM-AD-D54
T&M 명	• 적합한 프로그래밍 언어 (Suitable Programming languages)
주요개념(Concept)	• 언어는 완전하고 모호하지 않게 정의되어야 함 • 언어는 기계 지향적이 아닌 사용자 또는 문제 지향적이어야 함 • 널리 사용되는 언어 또는 그런 언어의 하위 집합이 특별한 목적의 언어보다 선호됨
적용목적(Objective)	• 가능한 많이 표준 요구사항을 지원 - 특히 방어적 프로그래밍, 강력한 타입 체크, 구조적 프로그래밍 및 가능한 조건 확인(assertion) • 최소한의 노력으로 쉽게 검증 가능한 코드를 이끌어낼 수 있고 프로그램의 개발, 검증 및 유지보수가 용이한 프로그래밍 언어를 선택
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술 (Techniques)	• 프로그래밍 언어

적용 지침(Guideline)

- 프로그래밍 언어를 선택할 때 중요 요소
  - 요구사항 적합도
  - 안전에 필수적인 응용 프로그램 또는 구성 요소를 개발할 때는 프로그램의 안전성을 최우선적으로 고려
  - 후보 언어 간의 알고리즘의 메모리 크기 및 실행 속도
  - 특정 프로세서 및 개발 플랫폼에서 언어를 지원하는 도구(컴파일러, 디버거, 통합개발도구 등)의 존재 여부
  - 통합 개발도구의 편집기에서 언어 별 약속어의 색을 강조하거나 하여 문제를 찾는데 용이한 지 여부
  - 언어 교육 및 경험이 있는 소프트웨어 엔지니어의 가용성
  - 컴파일러의 인증 여부
  - 컴파일러의 알려진 결함이나 오류 목록 존재 여부
  - 컴파일러가 경고를 발생하여 정합성을 강화하도록 지원하는지 여부

- 안전한 프로그램 위한 언어의 특성
  - 언어는 다음을 제공해야 함
    - ✓ 블록 구조
    - ✓ 변환 시 검사
    - ✓ 실행 시 데이터 타입 및 배열 경계 검사
    - ✓ 매개 변수 검사
  - 언어는 다음을 장려해야 함
    - ✓ 작고 관리가 쉬운 컴포넌트의 사용
    - ✓ 정의된 컴포넌트에서 데이터의 접근 제한
    - ✓ 변수의 하위 범위 정의
    - ✓ 오류 유형을 제한
  - 언어는 편리한 개발도구들이 지원되어야 함

✓ 적합한 변환기
✓ 기존 컴포넌트의 적합한 라이브러리
✓ 디버거
✓ 버전 관리 및 개발을 위한 도구

- 언어는 검증을 어렵게 만드는 요소는 피해야 함
✓ 서브루틴 호출을 제외한 무조건적 점프
✓ 재귀
✓ 포인터, 힙 또는 모든 데이터 타입의 동적 변수 또는 객체
✓ 소스코드 수준의 인터럽트 처리
✓ 반복, 블록 ,서브프로그램에서 여러 개의 입구 또는 출구
✓ 묵시적인 변수 초기화 또는 선언
✓ 가변적 레코드 및 동등성
✓ 프로시저 매개 변수
- 언어는 엄격한 언어 규칙을 가져야 함
✓ 엄격한 데이터 타입 검사
✓ 배열의 경계 검사
- 코딩 표준을 사용하여 잘못 정의된 특정 기능을 제한 가능
✓ 일부 기능이 모호한 방식으로 정의되어 의도와 다르게 동작 가능
✓ 일부 기능이 지나치게 복잡하거나 오류가 발생하기 쉬움
- 저수준 언어, 특히 어셈블리 언어는 기계 지향적인 성격으로 인하여 문제가 발생 가능

• 나사의 소프트웨어 안전성 가이드 북 (NASA-GB-8719.13)에서는 프로그래밍 언어에 대해서 다음과 같이 가이드하고 있음

• 모든 언어에 공통적인 불확실성
- 모든 프로그래밍 언어는 정의나 구현 면에서 불안정함
- 새로운 언어 (또는 기존 언어 표준의 업데이트)는 구형 언어의 부족을 수정하는 동시에 기능을 추가하는데 이는 새로운 불안정성을 추가할 가능성이 있음
- 초기화되지 않은 변수의 사용
✓ 초기화되지 않았거나 부적절하게 초기화 된 포인터 (포인터를 지원하는 언어)는 종종 잠재적 오류를 유발
✓ 이로 인해 잘 작동되던 프로그램이 다른 환경 조건에서 다른 결과가 나올 수 있음
- 메모리 관리 문제
✓ 메모리를 할당 해제 시 포인터와 구조체에서 사용되는 메모리가 해제되는지 확인
✓ 특정 메모리 블록에 대해 하나의 할당 해제 호출 만 수행되는지 확인
✓ 사용하지 않는 메모리를 반납하지 않으면 메모리 누수로 이어짐
- 알 수 없는 컴파일러 동작
✓ 컴파일러의 특정 버전에서만 부작용이 발생하는 경우
✓ 계산식의 연산 우선순위
✓ 전역 변수와 정적 변수의 초기화 순서 등

• 언어 선택을 위한 평가 방법
- 임의의 위치로 점프 금지 여부
- 임의의 메모리 위치를 덮어 쓰지 못하는지 여부
- 정적 코드 분석이 가능하도록 언어의 의미(semantics)가 충분히 정의되어 있는지 여

부
- 언어의 표준 내에 정수 및 부동 소수점 연산의 엄격한 모델 존재 여부
- 대상 운영체제가 대상 프로세서에서 실행될 때 연산 모델을 따르는지 확인하는 절차 존재 여부
- 변수의 오용을 방지하기 위한 엄격한 데이터 타입 체크 여부
  - ✓ 엄격한 데이터 타입 체크는 데이터의 형 변환 시 명시적인 형 변환이 필요
- 런타임에 메모리 부족을 방지 할 수 있는 기능의 존재 여부
- 모듈 별 분할 컴파일이 가능하도록 모듈 경계를 넘어 데이터 타입 검사 기능 제공 여부
- 설계자와 프로그래머가 안전에 중요한 소프트웨어를 작성할 수 있도록 언어가 이해하기 쉬운지 여부
- 안전한 언어의 속성을 가진 언어의 하위 집합의 존재 여부

- 언어
  - ADA
    - ✓ 군사 및 안전에 중요한 응용 분야에서 가장 일반적으로 사용되는 언어 중 하나로 안전성과 안정성을 염두에 두고 미국방부에서 고안한 언어
    - ✓ 특징
      - 객체지향 지원
      - 엄격한 데이터 타입 체크
      - 데이터의 범위 검사로 오류 방지
      - 멀티태스킹과 스레드 지원
      - 소스 코드의 가독성이 좋음
      - 래퍼를 통해 다른 언어로 작성된 모듈 사용 가능
      - 교착상태 방지 등의 런타임 시스템 지원
      - 분산 시스템 지원
      - 비 객체지향 소프트웨어 개발 스타일 지원
    - ✓ 안전관련 기능 지원
      - 컴파일러 유효성 검사
      - ADA 컴파일러는 표준 테스트 세트를 통한 유효성 검사를 통과 요건 존재
      - 언어 제한 능력 - 안전하지 않은 기능을 해제할 수 있는 기능 지원
      - 스칼라 값의 유효성 검사 - 유효 속성을 추가하여 사용자가 스칼라 객체의 비트 패턴이 유효한지 여부를 확인할 수 있도록 함
      - 검토 가능한 객체 코드 - 컴파일러에서 생성된 객체 코드를 검토할 수 있는 메커니즘을 제공
  - MODULA-2
    - ✓ 시스템 프로그래밍 언어이나, 범용적으로 사용 가능
    - ✓ 특징
      - 절차적 언어
      - 별도의 컴파일 및 데이터 추상화를 직접적으로 지원
      - Pascal과 거의 유사하게 설계
      - 구문상의 모호성을 제거하고 모듈 개념을 추가
      - 다중 프로그래밍을 위한 직접 언어 지원을 제공

- PASCAL
  - ✓ Algol 언어의 간단 버전으로 만들어짐
  - ✓ 특징
    - ◆ 교육 언어로 사용됨
    - ◆ 사용자 정의 타입을 정의할 수 있음
    - ◆ 구조화 프로그램 언어
    - ◆ SPADE Pascal11은 안전에 필수적인 어플리케이션에 필요한 연구 및 검증을 거친 하위 세트임
  - ✓ 한계
    - ◆ 분할 컴파일을 지원하지 않음
    - ◆ 예외처리를 제공하지 않음
- C언어
  - ✓ 미국 벨연구소에서 개발한 운영 체제나 언어 처리계 등의 시스템 기술에 적합한 범용 프로그래밍 언어
  - ✓ 특징
    - ◆ 유연성 높음
    - ◆ 지원 환경이 다양함
    - ◆ 데이터 타입을 강제하지 않음
    - ◆ 암시적 전환을 허용
    - ◆ 임베디드 및 런타임 환경에서 자주 사용됨
    - ◆ 하드웨어 액세스가 매우 쉬움
    - ◆ 콤팩트한 코드 생성 가능
    - ◆ 숙련된 공급 업체 및 인력풀이 풍부함
    - ◆ 안전에 중요한 응용 분야에 필요한 엄격함이 부족
    - ◆ 하나의 플랫폼에서 개발된 코드와 다른 플랫폼에서 사용된 코드에 대한 무결성 문제가 제기되는 수십 개의 하위 세트가 존재
    - ◆ 문제가 있음에도 불구하고 많은 안전 관련 응용 프로그램은 C로 개발되어 심각한 결함 없이 작동
    - ◆ C가 선택되면 코드 및 데이터 시퀀스에 대한 철저한 검증과 기능 및 오류 처리를 검증하기위한 충분한 테스트를 제공하는 것이 개발자에게 부담
    - ◆ 엄격한 코딩 표준이 필요
  - ✓ 한계
    - ◆ 포인터 사용으로 인한 잠재적 문제
    - ◆ 경계값 검사 부족
    - ◆ 부동 소수점 연산
    - ◆ void* 변환
    - ◆ 전역 변수
  - ✓ 안전한 프로그램을 위해 지켜야할 점
    - ◆ 연산 우선순위를 명확히 하기 위해 괄호를 사용
    - ◆ 변수 이름을 중심으로 매크로 내에서 괄호를 사용
    - ◆ 복잡한 매크로 정의에 전처리기를 사용금지

- 변수를 명시적으로 형 변환
- 가능한 경우 void * 포인터를 사용금지
- 배열과 문자열에서 범위 밖 접근 여부를 확인
- 항상 함수 프로토타입을 사용하면 컴파일러가 변수를 함수에 전달할 때 일관성 없는 데이터 타입의 문제점을 찾을 수 있음
- 전역 변수 사용의 최소화
- switch/case 문에 항상 default 절을 포함
- 가능한 경우 재귀 함수 금지
- 오류 처리 절차 및 상태 및 오류 로깅을 사용

- C++
  - ✓ C언어의 확장 언어
    - C의 효율성을 유지하면서 객체 지향 기능을 추가
  - ✓ 특징
    - 객체 지향 지원
    - C언어 보다 강력한 데이터 타입 검사
    - 변수와 함수의 “불변성”을 강제하는 키워드 “Const”가 존재
    - 템플릿을 사용한 제네릭 프로그래밍 가능
    - 객체 지향 및 구조 디자인 및 프로그래밍 스타일을 모두 지원
    - 사용자 정의 데이터 타입(클래스)을 내장 데이터 타입과 비슷한 효율로 처리 가능
    - 예외 및 오류 처리 지원
    - 네임 스페이스 지원
    - 변수에 대한 참조 지원
    - 인라인 함수 지원
  - ✓ 안전한 프로그램을 위해 지켜야할 점
    - 다중 상속 금지
    - 상속 수준의 최소화
    - 추상 클래스는 인터페이스만을 기술하고 구현은 금지
    - 포인터 사용 최소화
    - 별칭(alias) 사용 금지
    - 클래스가 상속될 수 있으면 소멸자를 가상으로 작성
    - 기본 생성자를 정의
    - 클래스의 대입 연산자를 정의하거나 컴파일러가 생성한 것에 대한 주석을 추가
    - 오퍼레이션의 오버로딩을 간결하고 균일하게 작성
    - 런타임 데이터 타입 정보(RTTI: Real-time Type Information) 사용 금지
    - 전역 변수 금지
    - 생성자에서 할당한 모든 메모리를 소멸자에서 제거하는지 확인
    - 표준 템플릿 라이브러리 사용 시 주의 (스레드 세이프티하지 않음)
    - 배열을 delete할 때 주의 delete a[]

- C#
  - ✓ C/C++을 기반으로 만들어졌으나 여러 면에서 Java와 매우 유사한 언어
  - ✓ 특징
    - 특정 플랫폼(마이크로소프트 윈도우)에 의존적인 언어
    - 예외처리 지원
    - 가비지 컬렉션 지원
    - 배열 범위 검사 지원
    - 다차원 배열 지원
    - 스레드 지원
    - 전역 변수 없음
    - 동적 변수는 사용 전 초기화 (초기화를 하지 않고 사용할 경우 컴파일러에서 경고)
    - 산술 연산자의 오버플로 검사
    - foreach 지원
  - ✓ 한계
    - 의도하지 않은 시점의 가비지 컬렉션이 발생 가능
    - 특정 플랫폼(마이크로소프트 윈도우)에서만 작동됨

적용예시(Example)

- 해당사항 없음

적용 시 고려사항(Considerations & Constraints)

- 언어 선택 시 유의점
  - 특정 프로그래밍 언어를 제외시키는 결정을 정당화 할 필요 없음
  - T&M 표에 없는 언어라고 해도 자동으로 제외되지 않음
  - 적합한 언어 조항을 준수할 경우 선택 가능
  - 응용 프로그램을 실행하는데 필요한 선택된 언어와 관련된 런타임 시스템은 소프트웨어 안전 무결성 등급(SIL)에 따라 사용에 대한 근거가 존재해야 함
  - 어떤 언어를 사용할지 결정하는 것은 각 언어의 위험과 이점을 분석하여 요구사항에 적합한 언어를 선택

- 안전한 소프트웨어를 위한 언어
  - 안전한 소프트웨어는 모든 언어로 작성 가능
  - 코딩 표준은 보다 안전한 코드를 생성하기 위해 특정 언어로 프로그래밍 하는 방법을 지정 가능
  - 안전한 프로그래밍 언어는 소스코드에서 오브젝트 코드로의 변환을 엄격하게 검증할 수 있는 언어임
  - 컴파일러가 소스코드에서 오브젝트 코드로의 완전한 전환을 보장하는 인증을 받기도 함
  - 언어가 안전한 소프트웨어를 만드는 것을 보장하지는 않지만 언어의 선택은 소프트웨어의 안전성에 영향을 줄 수 있음
  - 일부 언어는 특정 오류에 취약하기 때문에 엄격한 개발, 검토 및 테스트가 중요
  - 좀 더 일반적인 의미의 안전한 언어는 좋은 프로그래밍 습관을 강요하고 런타임보

다는 컴파일 타임에 오류를 찾는 언어
- 특정 언어는 다른 언어에 비해 안전이 중요한 분야에 더 적합하여 위험이 적음
- 안정성이 떨어지는 언어를 선택한다면 소프트웨어의 안전성을 보장하기 위하여 추가적인 분석과 테스트가 필요

C-6.14 절차적 프로그래밍

구 분	설 명
T&M No	• TM-AD-D60
T&M 명	• 절차적 프로그래밍 (Procedural programming)
주요개념(Concept)	• 프로그램이 원하는 상태에 도달하는데 걸리는 단계를 명세하는 프로그래밍 패러다임으로, 프로시저 호출의 개념을 기반으로, 수행되어야 할 연속적인 계산 과정을 프로시저로 구성하는 프로그래밍 방법
적용목적(Objective)	• 절차적 프로그래밍을 통해 시스템의 복잡도가 지나치지 않고 유지보수를 용이하게 할 수 있으므로 단순한 순차적 프로그래밍이나 비구조적 프로그래밍에 비해 모듈화, 구조화를 구현하는데 많은 장점이 있음
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계
주요 기술 (Techniques)	• 해당 사항 없음

적용 지침(Guideline)

- 절차적 프로그래밍(procedural programming)은 절차지향 프로그래밍 혹은 절차 지향적 프로그래밍이라고도 하는 프로그래밍 패러다임의 일종으로, 명령형 프로그래밍과 동일한 의미로 쓰이기도 하며, 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍 패러다임임.

- 절차적 프로그래밍은 프로시저 호출을 사용함으로서 단순한 순차적 프로그래밍이나 비구조적 프로그래밍보다 적절한 복잡도와 유지보수의 용이성을 확보할 수 있음

- 절차적 프로그래밍의 장점은 다음과 같은 장점이 있음
  - Copy & Paste가 아닌 동일 코드를 다른 장소에서 재사용함으로 코드 전체의 복잡성을 감소시킬 수 있음
  - GOTO문, JUMP문 지양으로 프로그램의 흐름의 복잡도를 줄일 수 있음
  - 시스템의 모듈화, 구조화를 실현할 수 있음
- 절차적 프로그래밍을 지원하는 개발언어
  - ALGOL
  - FORTRAN
  - PL/1
  - MODULA-2
  - ADA
  - C

적용예시(Example)

- 해당 사항 없음

적용 시 고려사항(Considerations & Constraints)

- 해당 사항 없음

C-6.15 제어 흐름 분석

표 341 제어 흐름 분석

구 분	설 명
T&M No	• TM-AD-D8
T&M 명	• 제어 흐름 분석 (Control Flow Analysis)
주요개념(Concept)	• 프로그램의 제어 흐름을 결정하기 위한 정적 프로그램 분석 기법 • 모범 프로그래밍 사례를 따르지 않는 의심스러운 코드 영역을 식별
적용목적(Objective)	• 결함이 있거나 잠재적으로 정확하지 않은 프로그램 구조를 탐지
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 제어 흐름 그래프
적용 지침(Guideline)	

- 주요 제어 흐름의 구성에 따라 제어 흐름 그래프(CFG: Control Flow Graph) 형식으로 분석하여 결함이 있는 코드를 탐지

- 주요 제어 흐름 구성

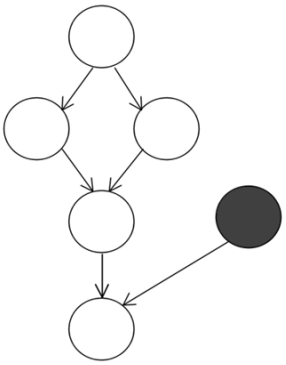
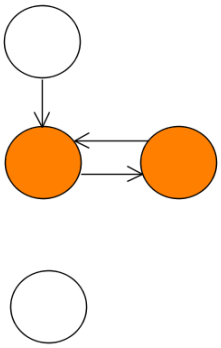
표 342 제어흐름의 구성 요소 목록

구성	설명
기본 블록	▪ 단일 진입점과 단일 종료점을 가지고 있고 내부 분기가 없는 연속 명령문의 순차로 이루어짐
루프	▪ 명령문들이 반복됨
메소드 호출	▪ 함수 호출을 받는 개체를 식별
예외처리	▪ 예외를 처리

- 제어 흐름 그래프는 가능한 모든 실행 경로를 상세화
  - 프로그램 구조를 파악하여 결함과 잠재적 문제점 탐지 가능

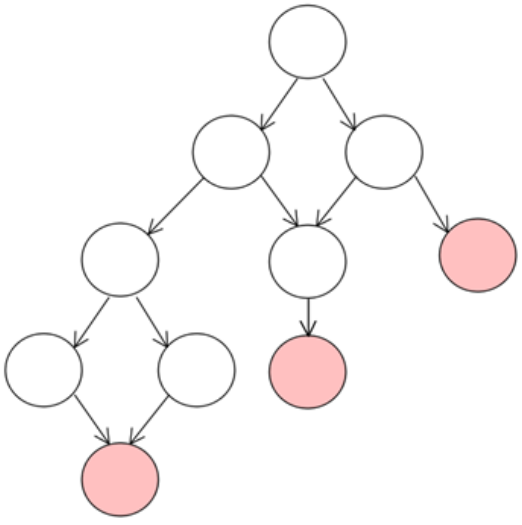
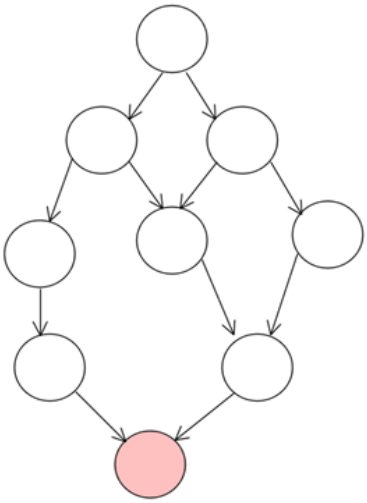
- 결함 코드 탐지

표 343 결함 코드 탐지의 예

접근할 수 없는 코드	무한루프
도달할 수 없는 코드 블록을 나가는 점프	루프를 나갈 수 있는 조건이 없을 경우
 그림 237 예제1	 그림 238 예제2

- 프로그램의 구조화
  - 잘 구조화 된 코드는 제어 그래프가 연속적인 그래프 축소에 의해서 단일 노드로 감소되는 코드
  - 부실하게 구조화된 코드는 오직 여러 개의 노드로 구성된 매듭으로 축소될 수 있음

표 344 프로그램의 비구조화/구조화 제어 흐름 그래프

잘못 구조화된 코드의 제어 흐름 그래프	잘 구조화된 코드의 제어 흐름 그래프
 그림 239 예제3	 그림 240 예제4

적용예시(Example)

- 해당 사항 없음

적용 시 고려사항(Considerations & Constraints)

- 적용 대상
  - 구조적 프로그래밍과 객체지향 프로그래밍 모두 적용 가능
  - 정교해지고 복잡해지는 소프트웨어의 질적 향상을 위한 기법

- 제안사항
  - 크기가 크고 복잡한 프로그램의 분석할 경우 자동화된 정적 분석 도구를 사용하면 신속하고 정확하게 분석 가능

C-6.16 데이터 흐름 분석

구 분	설 명
T&M No	• TM-AD-D10
T&M 명	• 데이터 흐름 분석 (Data Flow Analysis)
주요개념(Concept)	• 데이터 사용 흐름을 파악 • 데이터에 대한 접근과 수정을 추적
적용목적(Objective)	• 결함이 있거나 잠재적으로 정확하지 않은 프로그램 구조를 탐지
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 데이터 흐름 분석

적용 지침(Guideline)

- 데이터 흐름 분석은 제어 흐름 분석에서 얻은 정보와 코드의 다양한 부분에서 변수들이 읽고, 써진 정보를 결합하여 수행

- 제어 흐름 그래프(CFG Control Flow Analysis)의 각 노드에 데이터 흐름을 분석하여 결함이 있는 코드를 탐지

- 결함 코드
  - 할당되지 않은 변수 참조
    - ✓ 정의되지 않거나 값이 할당되지 않은 변수를 참조하는 경우는 오류가 될 가능성이 아주 높으며, 확실히 나쁜 프로그래밍 방법임
    - ✓ 변수의 초기화
  - 코드 누락
    - ✓ 변수를 여러 번 할당하였으나 사용하지 않는 변수는 누락된 코드를 가리키는 것일 수 있음
    - ✓ 코드 누락여부를 확인하여 누락 코드 추가
  - 중복 코드
    - ✓ 값은 할당하였으나 미사용 변수는 중복된 코드를 가리키는 것일 수 있음
    - ✓ 중복코드 삭제

적용예시(Example)

- 할당되지 않은 변수 참조
  - 변수의 초기화

표 345 할당되지 않은 변수의 예

할당되지 않은 변수	할당된 변수

int status;
if (status == READY)
{
/*...*/
}
	int status = USIGNED;
	if (status == READY)
	{
	/*...*/
	}

- 코드 누락
  - 여러 번 할당한 변수를 사용하는 코드 삽입

표 346 여러 번 할당한 변수 사용의 예

누락된 코드	누락되지 않은 코드

int result;
status = READY;
start_process();
/*...*/
status = START;
 	
return result;
	int result;
	status = READY;
	start_process();
	/*...*/
	status = START;
	result = status; // 누락된 코드
	return result;

- 중복 코드
  - 중복 코드 제거

표 347 중복 코드의 예

중복된 코드	중복 제거한 코드

int status = READY;
int result;
start_process();
/*...*/
status = START;
result = status;
return status;
	int status = READY;
	
	start_process();
	/*...*/
	status = START;
	
	return status;

적용 시 고려사항(Considerations & Constraints)

- 제어 흐름 분석과 마찬가지로 자동화된 정적 분석 도구를 사용할 경우 크기가 크고 복잡한 프로그램의 분석을 신속하고 정확하게 수행 가능

- 정보 흐름 분석은 데이터 흐름 분석을 확장한 것으로, 프로시저 간 혹은 프로시저 내의 실제 데이터 흐름과 설계 의도를 비교한 것으로 이는 보통 자동화된 도구로 구현되며, 도구가 해석할 수 있는 구조화된 주석을 사용하여 의도한 데이터 흐름을 정의함

C-6.17 워크스루 / 설계 검토

표 348 워크스루 / 설계 검토

구 분	설 명
T&M No	• TM-AD-D56
T&M 명	• 워크 스루 / 설계 검토 (Walkthroughs/Design Reviews)
주요개념(Concept)	• 개발자가 진행하는 검토 회의
적용목적(Objective)	• 가능한 빠르고 경제적으로 개발 프로세스에서 오류를 감지 • 시스템에 대한 이해
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 검토 회의 • 설계 검토

적용 지침(Guideline)

- 검토자
  - 개발자가 소집한 전문가에 의해 개발자의 작업이 검토됨
  - 프로젝트 팀장, QAO과 같은 전문가들이 미리 제공된 검토 자료를 정해진 절차에 따라 검토

- 검토 유의사항
  - 주로 요구사항 맞게 설계가 잘 되었는지 여부를 확인하고 작업 상황을 확인
  - 회의 자료는 회의 전에 배포되어야 함
  - 문제점을 발견하는데 초점(문제해결은 검토 회의 이후)
  - 문제점을 발견하면 목록을 개발자에게 전달

- 설계 검토 표준 IEC 61160
  - IEC는 정형적 설계 검토 가이드 IEC 61160을 발간
  - IEC 61160는 다음을 포함
    - ✓ 정형적 설계 검토를 계획하고 수행하기 위한 일반적인 가이드라인
    - ✓ 설계 검토 팀 구성
      - ◆ 설계 검토 팀 내의 독립적인 전문가 역할에 대한 세부사항
    - ✓ 관련 업무와 책임
  - IEC 61160 권고사항
    - ✓ 정형적 설계 검토는
      최종제품/프로세스, 사용자, 관계자에게 영향을 미치는 기능, 성능, 안전성, 신뢰성, 유지보수성, 비용, 기타 특성에 대하여
      모든 새로운 제품/프로세스, 어플리케이션, 기존의 제품과 제조 프로세스의 새로운 리비전을 대상으로 수행해야 함

적용예시(Example)

- 코드 검토 회의
  - 검토자
    - ✓ 개발팀, QA팀
  - 자료
    - ✓ 검토자들이 선택한 소규모 작은 종이 테스트 케이스 세트

✓ 대표적인 프로그램을 위한 입력과 정확한 예상 결과(출력) 세트

- 검토 방법
  - ✓ 프로그램의 로직에 따라 수동으로 추적
- 검토 후 절차
  - ✓ 시험결과를 개발자에게 전달

적용 시 고려사항(Considerations & Constraints)

- 검토 회의 시간은 너무 길지 않도록 함 (1~2시간 이내)
- 검토 회의 결과를 인사 평가 자료로 사용 금지

C-6.18 추적성

구 분	설 명
T&M No	• TM-AD-D58
T&M 명	• 추적성 (Traceability)
주요개념(Concept)	• 모든 요구사항에 대해 생명주기 전 단계에 걸친 대상(입출력)에 대한 추적성을 수립하고 관리
적용목적(Objective)	• 모든 요구사항이 적절하게 매핑 되고, 모든 대상에 대해 추적을 보장
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 해당사항 없음

적용 지침(Guideline)

- 요구사항에 대한 추적성은 시스템 확인에 중요한 고려 사항이며 생명 주기의 모든 단계를 통하여 입증이 가능하도록 수단이 제공되어야 함

- 추적성은 기능적 요구사항과 비기능적 요구사항 모두에 적용 가능한 것으로 고려되어야 함

- 추적성은 다음을 만족해야 함
  - 요구사항에서 설계 또는 그것을 만족하는 다른 객체에 대한 추적성
  - 설계 객체에서 그것을 인스턴스화한 구현 객체에 대한 추적성
  - 요구사항 및 설계 객체에서 시스템의 안전하고 적절한 사용에 적용되는 운영 및 유지보수 객체에 대한 추적성
  - 요구사항, 설계, 구현, 운영 및 유지보수 객체에서 수용 가능성을 결정하게 될 검증과 시험 계획 및 명세에 대한 추적성
  - 검증 및 시험 계획 및 명세에서 적용의 결과를 기록하는 시험 또는 다른 보고서에 대한 추적성

- 추적성 기술 방법
  - 추적성 매트릭스
  - 추적성 그래프

- 추적성 정보의 유용성
  - 변경 영향도 분석
  - 커버리지 분석
  - 프로젝트 상태 확인
  - 제품 구성 요소 재사용
  - 유지보수의 용이
  - 테스트 최적화

적용예시(Example)

- 요구사항과 테스트 케이스에 대한 추적성 매트릭스

Requirement	Reqs Tested	REQ 1.1	REQ 1.2	REQ 1.3	REQ 2.1	REQ 2.2	REQ 2.3	REQ 2.4	etc…	REQ 10.3
Test Cases	213	2	2	1	1	1	2	1		3
1.1.1	1	O								
1.1.2	1		O							
1.1.3	1			O						
1.1.4	3	O			O		O			
1.1.5	1					O				
1.1.6	1						O			
1.1.7	2		O					O		
etc…										
7.3.3	1									O

그림 241 요구사항과 테스트 케이스에 대한 추적성 매트릭스

적용 시 고려사항(Considerations & Constraints)

- 추적성의 관리는 안전 시스템을 개발할 때 특히 중요함
- 요구사항, 설계 등 기타 객체들이 여러 개의 별도 문서로 인스턴스화 되는 경우, 추적성은 문서 구조 내에서 계층적인 방식으로 유지되어야 함
- 추적성 프로세스의 결과물은 정형적 형상 관리의 대상이 되어야 함
- 모든 생애주기의 개체에 대한 추적성 관리를 하기 위해 추적표를 수동으로 만들어서 관리하기도 하나, 대상이 많을 경우 현행화 등 관리가 쉽지 않으므로 자동화 도구를 사용하는 것을 권장

C-6.19 구조 기반 시험

구 분	설 명
T&M No	• TM-AD-D50
T&M 명	• 구조 기반 시험 (Structure Based Testing)
주요개념(Concept)	• 테스트 케이스 스위트(Suite)에 의해 실행된 구문이 몇 퍼센트인지를 측정하는 것으로 테스트 케이스가 커버하는 실행 가능 문장들의 수를 테스트 중인 코드에 포함된 모든 실행 가능 문장의 수로 나눈 값으로 표시한다.
적용목적(Objective)	• 프로그램 구조의 특정 부분 집합을 시험하기 위한 테스트
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계 • 소프트웨어 컴포넌트 구현 및 테스팅
주요 기술 (Techniques)	• 해당 사항 없음

적용 지침(Guideline)

- 소프트웨어의 구조가 테스트 케이스에 의해 테스트된 정도를 커버리지(Coverage)라 하며, 특정 구조의 종류에 대해 커버된 백분율로 표시한다.
- 커버리지(Coverage)에는 그 강도에 따라 문장 커버리지 외에 다양한 종류가 존재하고 보장하는 범위가 다르다.
- 문장(Statement) 커버리지는 테스트 케이스에 의해 실행된 구문이 몇 퍼센트인지를 측정한 것으로 보장하는 범위가 가장 약하다.
- 문장(Statement) 커버리지 측정 수식은 다음과 같다.

  Statement = A / B * 100

  A. 테스트 수행된 문장(Statement) 수(개)

  B. 문장(Statement) 수(개)

- 문장 커버리지는 코드의 모든 구문을 실행할 수 있는 입력값이나 이벤트 등의 테스트 데이터를 가지는 테스트 케이스로 달성되며 일반적으로 테스트 도구를 활용하여 측정된다.
- 문장(Statement) 테스팅 기법은 문장 커버리지를 늘리기 위해 특정 문장을 테스트하는 테스트 케이스를 도출하는 것이다.

적용예시(Example)

표 349 문장 커버리지의 예

문장 커버리지 예

int fnChk(int nStatus)
{
1: f (nStatus < 0)
{
2 fnAssert();
: }
3:else {
4: fnlogic();
: }
}

- fnChk함수의 문장 커버리지를 100% 달성하려면 1,2,3,4 구문이 모두 실행되는 입력값을 가진 테스트 케이스를 설계해야 한다.

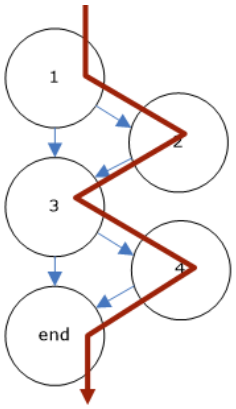


그림 242 문장 커버리지를 위한 구문 흐름 설계

표 350 테스트 커버리지 측정을 위한 테스트 케이스의 예

NO	테스트 케이스	비고
1	▪ TC_Fnchk (-1);	조건 체크, fnAssert() 구문 실행
2	▪ TC_Fnchk (10);	조건 체크, fnlogic() 구문 실행

- 테스트 도구 커버리지 측정 예시

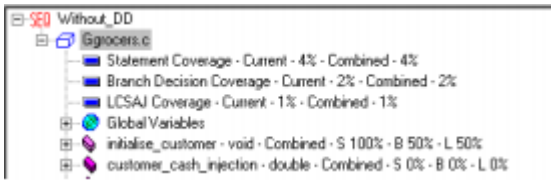


그림 243 테스트 도구 커버리지 측정 예시

적용 시 고려사항(Considerations & Constraints)

- 해당 사항 없음

C-6.20 인터페이스 시험

구 분	설 명
T&M No	• TM-AD-D34
T&M 명	• 인터페이스 시험 (Interface Testing)
주요개념(Concept)	• 주로 인터페이스에 포함된 변수를 대상으로 결함을 테스트하는 것으로 소프트웨어 관련 모듈 간의 인터페이스, 소프트웨어 모듈과 하드웨어의 인터페이스 검증을 포함한다.
적용목적(Objective)	• 서브프로그램의 인터페이스가 특정 응용 프로그램의 장애로 인해서 발생된 오류를 포함하지 않고 관련된 모든 오류를 감지하는 것을 보장
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계 • 소프트웨어 컴포넌트 설계
주요 기술 (Techniques)	• 해당 사항 없음

적용 지침(Guideline)

- 서로 연결된 다양한 프로그램이나 모듈 간 인터페이스의 결함을 확인하는 테스트 기법이다.
- 인터페이스 테스트는 프로그램 코드 또는 설계서를 기반으로 테스트 케이스를 도출해내는 구조적 테스트 방법으로 직접 데이터 흐름, 간접 데이터 흐름 방식으로 구분할 수 있다.
- 여러 수준의 상세 또는 완전성 테스트가 가능하며 인터페이스 변수의 극한 값 또는 정상 값, 모든 변수의 값 조합(작은 규모의 인터페이스인 경우만 가능), 서브루틴 호출과 같은 특정 테스트 조건 등을 주요 인터페이스 테스팅에 적용한다.
- 인터페이스에 잘못된 매개변수의 값을 감지하는 예외처리가 포함되어 있지 않은 경우, 기존 서브루틴에 새로운 구성이 추가된 경우에 인터페이스 테스팅은 특히 중요하다.

적용예시(Example)

- 해당 사항 없음

적용 시 고려사항(Considerations & Constraints)

- 해당 사항 없음

C-6.21 구조적 프로그래밍

구 분	설 명
T&M No	• TM-AD-D53
T&M 명	• 구조적 프로그래밍 (Structured Programming)
주요개념(Concept)	• 구조적 프로그래밍(structured programming)은 구조화 프로그래밍이라고 하며, 프로그래밍 패러다임의 일종인 절차적 프로그래밍의 하위 개념으로 볼 수 있음
적용목적(Objective)	• 구조적 프로그래밍 방법을 적용함으로써 시스템의 구조화와 복잡도를 낮출 수 있으며, 모듈화를 구현하기 위해 적용
적용단계(Phase)	• 소프트웨어 아키텍처 및 설계
주요 기술 (Techniques)	• 해당 사항 없음

적용 지침(Guideline)

- 구조적 프로그래밍(structured programming)은 구조화 프로그래밍으로도 하며, 프로그래밍 패러다임의 일종인 절차적 프로그래밍의 하위 개념으로 볼 수 있음.
- GOTO문을 없애거나 GOTO문에 대한 의존성을 줄여주는 특징이 있음.
- 구조적 프로그래밍은 일반적으로 하향식 설계(Top-Down)와 관련이 있으며, 하향식 설계 시, 설계자는 큰 규모의 프로그램을 더 작은 공정으로 나누어 구현하고, 각각 검사한 다음에 전체 프로그램으로 통합하는 형태로 진행 할 수 있음
- 구조적 프로그램의 주요 특징은 다음과 같음
  - 블록이라는 단위를 이용하여 프로그램을 작성
  - GOTO 문장의 사용 금지
  - 제한된 제어구조 만을 허용(순차구조 :Sequence, 반복구조:Repetition, 선택구조:Selection)
  - 특정 프로그램 내에서 하나의 시작점을 갖는 함수는 반드시 하나의 종료점을 가짐

표 351 구조적 프로그래밍에서 제시하는 제어구조 유형

제어구조 형태	설명
순차(concatenation)	구문 순서에 따라서 순서대로 수행되는 제어구조
선택(selection)	프로그램의 상태에 따라서 여러 구문들 중에서 하나를 수행하는 제어구조 (if..then..else..endif, switch, case와 같은 키워드)
반복(repetition)	프로그램이 특정 상태에 도달할 때까지 구문을 반복하여 수행하거나, 집합체의 각각의 원소들에 대해 어떤 구문을 반복 수행하는 제어구조 (while, repeat, for, do..until 같은 키워드)

- 구조적 프로그래밍을 지원하는 개발언어는 모든 절차적 프로그래밍 언어에서 구조적 프로그래밍을 할 수 있음. 잘 알려진 구조적 프로그래밍 언어는 다음과 같음
  - PASCAL
  - ADA

적용예시(Example)

- 해당 사항 없음

적용 시 고려사항(Considerations & Constraints)

- 해당 사항 없음

C-6.22 영향 분석

구 분	설 명
T&M No	• TM-AD-D32
T&M 명	• 영향 분석 (Impact Analysis)
주요개념(Concept)	• 소프트웨어에 대한 변경 또는 개선이 수행되기 전에 소프트웨어의 변경 또는 개선에 따른 영향을 식별하고 해당 시스템과 관련한 시스템과 구성요소를 식별하기 위한 영향도 분석을 수행
적용목적(Objective)	• 소프트웨어의 변경 및 개선이 타 시스템 또는 소프트웨어의 구성요소에 미치는 영향을 식별과 확인을 위함
적용단계(Phase)	• 소프트웨어 유지보수
주요 기술 (Techniques)	• 해당사항 없음

적용 지침(Guideline)

- 영향도 분석은 소프트웨어 시스템에서 변경하고자 하는 변경의 잠재적 결과를 확인하는데 도움이 기술임.

- 소프트웨어의 변경 및 개선은 비즈니스 요구사항 변경, 신규 요구사항, 새로운 기술 등과 같은 다양한 소스로부터 발생 할 수 있으나, 소프트웨어 시스템의 규모 및 복잡도에 따라 변경으로 인해 오류가 발생하거나 통제가 불가능해질 가능성이 존재함.

- 적절한 계획 없이 소프트웨어 시스템을 직접적으로 변경하는 것은 위험도가 크므로, 영향 분석 수행을 통해 실제 변경을 수행하기 전, 영향을 주는 내용을 파악하며, 할 수 있습니다. 이를 통해 변경범위를 결정하고 변경 계획에 필요한 자원을 추정하는 데 도움을 주는 기술임.

- 문제의 심각성/중요도에 따라 언제 어떻게 문제를 수정하는가를 결정해야 합니다. 그런 뒤 소프트웨어 엔지니어는 영향을 받는 컴포넌트를 식별합니다. 몇 가지 가능한 해결책을 제시하고, 가장 적절한 조치방안을 추천합니다. 유지보수성을 염두에 두고 설계된 소프트웨어는 쉽게 영향도 분석을 수행할 수 있습니다.

- 영향도 분석은 2 단계로 수행할 수 있으며, 시스템 변경 또는 개선에 따른 내부 구성요소 및 구성요소간의 연관관계를 식별과, 변경과 관련된 가능한 결과와 위험을 밝히기 위해 검토합니다. 관계를 찾는 두 가지 방법이 있습니다. 대상 요소 또는 모듈과 관련된 동적 연결뿐만 아니라 정적 관계를 연구합니다. 우리는 그것들을 추적 성 연구와 의존성 연구라고 부릅니다.

- 추적성 연구에서 소프트웨어 시스템의 사양 및 설계에 정의 된 요구사항, 사양, 설계 요소, 테스트 사례 등을 포함하여 해당 주제 요소와 관련된 모든 정적 관계를 검토할ㅇ 수 있으며, 의존성 연구에서 우리는 변수, 프로그램 로직, 모듈 아키텍처 등의 사용을 포함하여 주제 요소와 관련된 모든 논리적이고 역동적인 연결을 검토 할 수 있음

- 추적성 및 종속성 연구를 수행 한 후 주제 요소와 관련된 모든 관계를 얻을 수 있으며, 두 번째 단계로 이동하여 변경 사항을 적용 할 때 관련 요소에 대한 영향을 평가하기 위해 살펴봐야 할 내용과의 관계를 보면, 다음과 같음 관련 요소를 고려해야 한다.

- 변경으로 인해 다른 모듈을 실행할 수 없게 되는가?
- 변경으로 인해 시스템 성능이 저하될 수 있는가?
- 변경을 구현하려면 얼마나 많은 자원이 필요한가?

- 분석이 완료된 후 소프트웨어 시스템의 재검증에 관한 결정이 필요합니다. 이는 영향을 받는 구성 요소의 수, 영향을 받는 구성 요소의 중요성 및 변경의 성격에 따라 달라질 수 있음
  - 변경된 구성요소만 재검증
  - 식별된 모든 영향을 받는 구성요소 재검증
  - 전체 시스템 재검증

적용예시(Example)

- 해당사항 없음

적용 시 고려사항(Considerations & Constraints)

- 영향도 분석은 현재 소프트웨어에서 변경의 영향을 어떻게 완전하게, 비용-효과적으로, 분석을 할 수 있을까를 고민해야 하며,
- 유지보수 담당자(Maintainer)는 그 소프트웨어 구조와 내용에 대해 상세한 지식을 가지고 있어야 하고 이러한 지식을 활용해 소프트웨어 변경요청이 영향을 미치는 모든 시스템과 소프트웨어 구성요소들을 식별하고, 그 변경에 필요한 자원을 산정하는 영향도 분석을 수행해야 한다.
- 또한, 변경에 의해 발생되는 위험도 파악해야 합니다. 수정요청 (Modification Request (MR)), 문제 보고서 (Problem Report (PR))라고도 불리는 변경요청 (Change Request)이 가장 먼저 분석되어야 하며, 소프트웨어 용어로 해석되어져야 합니다. 이것은 변경요청이 소프트웨어 형상관리 프로세스에 등록된 후 실행되어져야 함을 의미한다.

C-6.23 데이터 기록 및 분석

구 분	설 명
T&M No	• TM-AD-D12
T&M 명	• 데이터 기록 및 분석 (Data Recording and Analysis)
주요개념(Concept)	• 유효한 데이터의 기록과 분석을 통해 소프트웨어 개발 프로세스에 관해 더 많은 것을 학습하고 대안 소프트웨어 개발 방법론에 대한 평가 • 데이터 기록 및 분석은 소프트웨어 프로세스 향상의 필수 요소
적용목적(Objective)	• 개별 프로젝트 및 인력과 관련된 데이터를 확인, 분석, 기록함으로써 소프트웨어 프로세스 향상을 용이하게 함
적용단계(Phase)	• 소프트웨어 유지보수
주요 기술 (Techniques)	• 해당사항 없음

적용 지침(Guideline)

- 계획 수립
  - 데이터의 기록 계획 수립 시 대상 데이터와 분석 기법의 선정은 조직의 전략적인 목표에 의해서 결정
  - 목표
    - ✓ 목표는 특정 소프트웨어 개발 방법에 대한 요청과 관련된 평가를 위한 것일 수 있음
    - ✓ 조직의 품질 목표 또는 정의된 상세 프로세스에 대한 성과 목표를 설정
  - 측정지표(Metric)를 도출
    - ✓ 품질 목표(Goal) 달성을 위한 필요 질문(Question)과 지표 도출
    - ✓ 도출된 지표의 수집 가능성 및 효용성 검토
  - 선정된 지표의 수집 및 분석을 위해 세부 특성을 정의
    - ✓ 관련 프로세스
    - ✓ 지표 측정 계산식 및 설명
    - ✓ 수집 데이터 항목
    - ✓ 데이터 수집 방법 (현 시스템에서의 수집 가능 여부 검토)
    - ✓ 지표 측정 주기
    - ✓ 분석 및 활용방법 등

- 데이터 기록
  - 프로젝트 기간 동안 조직의 목표에 맞게 데이터를 기록하도록 함
  - 프로젝트 및 프로젝트 참여자 모두에 대한 상세한 기록들이 프로젝트 기간 동안 유지되어야 함
  - 데이터의 정확성을 위해 데이터 기록과 확인 절차는 개발과 동시에 진행
  - 데이터의 정확성을 위해 기록은 형상 제어 절차의 일부로써 진행

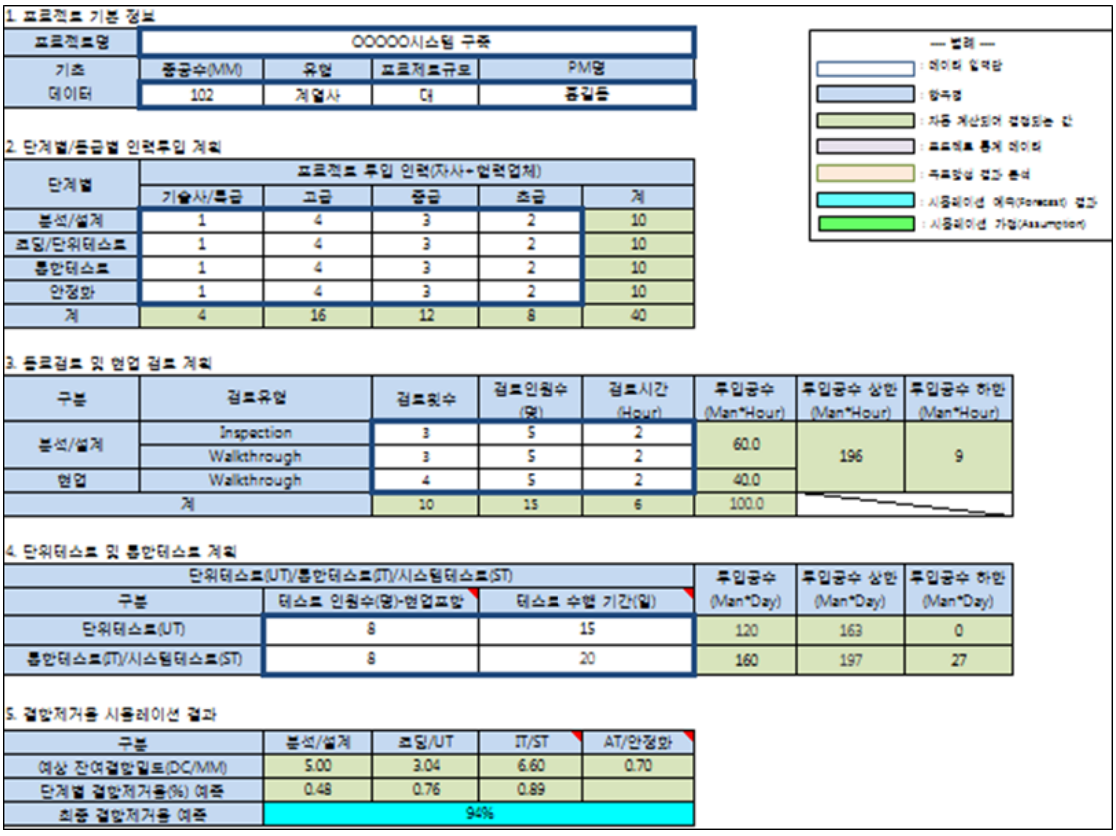
- 데이터 분석
  - 선정된 성과지표의 과거 데이터를 수집하고 통계 도구/기법을 활용하여 분석함으로써 기준선을 수립함
  - 목적에 맞는 기법으로 기록된 데이터를 분석하여 결론을 도출하고 그 결과를 프로젝트에 반영함

적용예시(Example)

- 과거 프로젝트 데이터의 통계 분석을 통해 프로세스와 제품에 대한 성과 기준을 정의하고, 성과를 예측할 수 있는 정량적 모델을 개발하여 새로 시작되는 프로젝트에서 활용

- 계획 수립
  - 목표: 프로세스 성과 예측
  - 측정지표: 총 공수, 단계별/등급별 인력 투입 인원 등

- 데이터 기록
  - 데이터 기록을 위하여 프로젝트의 정보를 기록 보존한다.
    - ✓ 단계별/등급별 인력투입 계획
    - ✓ 동료검토 및 현업 검토 계획
    - ✓ 단위테스트 및 통합테스트 계획 등



1. 프로젝트 기본 정보

프로젝트명	OOOOO시스템 구축			
기초 데이터	총공수(M/M)	유업	프로젝트규모	PM명
	102	계열사	대	홍길동

2. 단계별/등급별 인력투입 계획

단계별	프로젝트 투입 인력(자사+업력업체)				
	기술사/특급	고급	중급	초급	계
분석/설계	1	4	3	2	10
코딩/단위테스트	1	4	3	2	10
통합테스트	1	4	3	2	10
안정화	1	4	3	2	10
계	4	16	12	8	40

3. 동료검토 및 현업 검토 계획

구분	검토유형	검토회수	검토인원수 (명)	검토시간 (Hour)	투입공수 (Man*Hour)	투입공수 상한 (Man*Hour)	투입공수 하한 (Man*Hour)
분석/설계	Inspection	3	5	2	60.0	196	9
	Walkthrough	3	5	2			
현업	Walkthrough	4	5	2	40.0		
계		10	15	6	100.0		

4. 단위테스트 및 통합테스트 계획

단위테스트(UT)/통합테스트(IT)/시스템테스트(ST)			투입공수 (Man*Day)	투입공수 상한 (Man*Day)	투입공수 하한 (Man*Day)
구분	테스트 인원수(명)-현업포함	테스트 수행 기간(일)			
단위테스트(UT)	8	15	120	163	0
통합테스트(IT)/시스템테스트(ST)	8	20	160	197	27

5. 결함제거율 시뮬레이션 결과

구분	분석/설계	코딩/UT	IT/ST	AT/안정화
예상 잔여결함밀도(DC/MM)	5.00	3.04	6.60	0.70
단계별 결함제거율(%) 예측	0.48	0.76	0.89	
최종 결함제거율 예측	94%			

그림 244 프로젝트 정보 기록의 예

- 데이터 분석
  - 선정된 성과지표의 과거 데이터를 수집하고 통계 도구/기법을 활용하여 분석함으로써 기준선을 수립

- 새로운 프로젝트의 성과 예측 분석
  - 대상 프로젝트 (데이터)

단계	단계별 투입계획 (개월)	투입인원				
		특급	고급	중급	초급	계
설계	2.0	2	4			6
제작	3.0	1	3	2		6
테스트	3.0	1	3	2		6
계	8.0					

그림 245 분석에 대한 프로젝트의 예

- 분석 결과
  - ✓ 기준선을 기준으로 대상 프로젝트는 목표 달성 가능성(38%)이 낮음

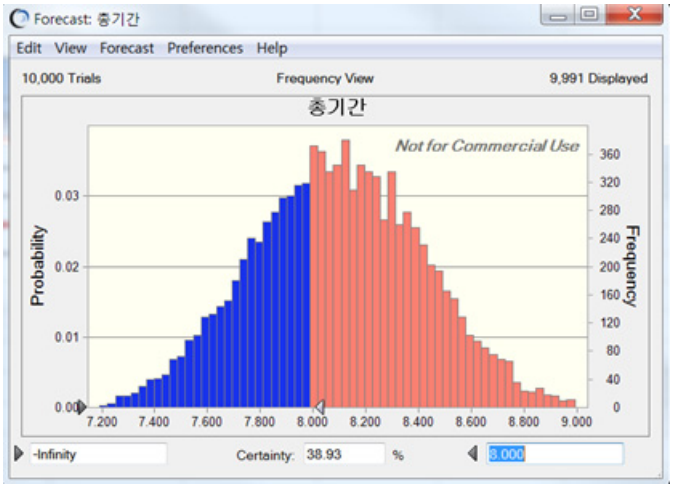


그림 246 프로젝트 분석의 예

- 분석된 결과를 기반으로 결함을 줄이기 위한 방안을 제시
  - ✓ 인력 투입 계획 수정
  - ✓ 고급 2명 추가 투입

단계	단계별 투입계획 (개월)	투입인원				
		특급	고급	중급	초급	계
설계	2.0	2	4			6
제작	3.0	1	4	2		7
테스트	3.0	1	4	2		7
계	8.0					

그림 247 프로젝트 분석 결과 반영의 예

적용 시 고려사항(Considerations & Constraints)

- 데이터 기록과 분석의 중요성
  - 프로젝트 기간 동안과 프로젝트의 종료 시 데이터 기록은 조직의 전략적인 목표 달성 및 예측을 위한 중요한 정보
  - 데이터 기록은 시스템의 유지보수에 매우 중요
  - 유지보수 엔지니어가 개발 프로젝트 동안 만들어진 어떤 결정에 대한 근거를 알 수 있는 바탕이 될 수 있음

- 데이터 기록 시 문제점과 해결책

- 문제점
  - ✓ 데이터 기록은 부실한 계획으로 인해 때때로 과도한 분량 및 초점을 벗어나는 경향이 있음
- 해결책
  - ✓ 이를 회피하기 위해 데이터 기록은 원칙에 따라 수행
  - ✓ 목적, 문제점 그리고 전략적으로 조직에 중요한 관련 항목에 의해 주도 되어야 함

부록 D. 안전성 분석 지원 도구 사용 방법

안전성 분석에 사용되는 FMEA, FTA에 자동화 도구들 중에서 널리 사용되고 있는 IQ-FMEA와 Reliability Workbench FaultTree+에 대해서 기술한다.

D-1. IQ-FMEA

○ 개요

FMEA 지원 도구인 IQ-FMEA는 구조분석을 시작으로 시스템을 분석 및 정의하는 과정을 수행하게 된다. 구조분석을 수행하고 나면 개별 구성품이 지니고 있는 기능을 분석하게 된다. 분석된 기능을 바탕으로 의도된 기능이 수행되지 못하는 상황 즉, 오작동에 대한 정의 및 영향에 대한 분석을 수행하게 된다. 또한 이러한 개별 고장이 시스템에 미치는 심각도를 평가하여 반영하게 된다. 평가된 지표를 바탕으로 설계 개선 요소를 찾고 이를 반영하는 순환적 프로세스를 지원한다.

○ 주요기능

IQ-FMEA는 FMEA 위험도 분석 기법에서 요구하는 템플릿 양식을 자체 Forms 및 다양한 FMEA 표준 약식을 지원하고 있다. 특히, 이러한 양식을 작성 및 지원하기 위한 다양한 분석 기법(구조분석, 기능분석, 오류분석, 추적성 확립)을 지원하고 있다. 그리고 FMEA 작성을 위한 Structure Trees, Functions Nets, Failure Nets, Cause and Effects Diagrams, Fault Trees를 기본기능으로 지원하고 있다.

- Structure Tree: 시스템의 구성요소를 나타냄
- Function Net: 시스템 요소들이 갖은 기능을 나타냄
- Failure Net: 각각의 기능에 대한 고장(오류)을 나타냄
- FMEA form: 다양한 FMEA 수행 글로벌 템플릿 양식 지원한다.

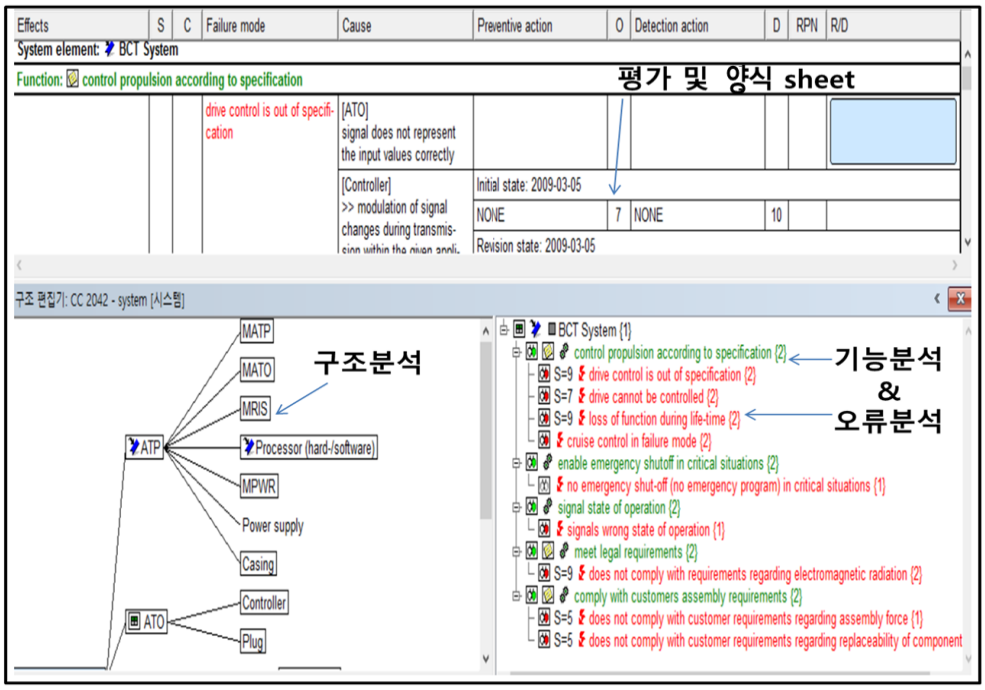


그림 248 IQ-FMEA 구성

○ 도구 사용법

- STEP 1. Structure Analysis(구조 분석) 수행

  - 아키텍처 산출물을 바탕으로 식별된 시스템 구성도 및 시스템 사양서를 기반으로 IQ-FMEA 도구 기반의 구조분석을 수행해야 한다. (아키텍처 정보를 기반으로 구성 요소에 대한 구조를 정의)

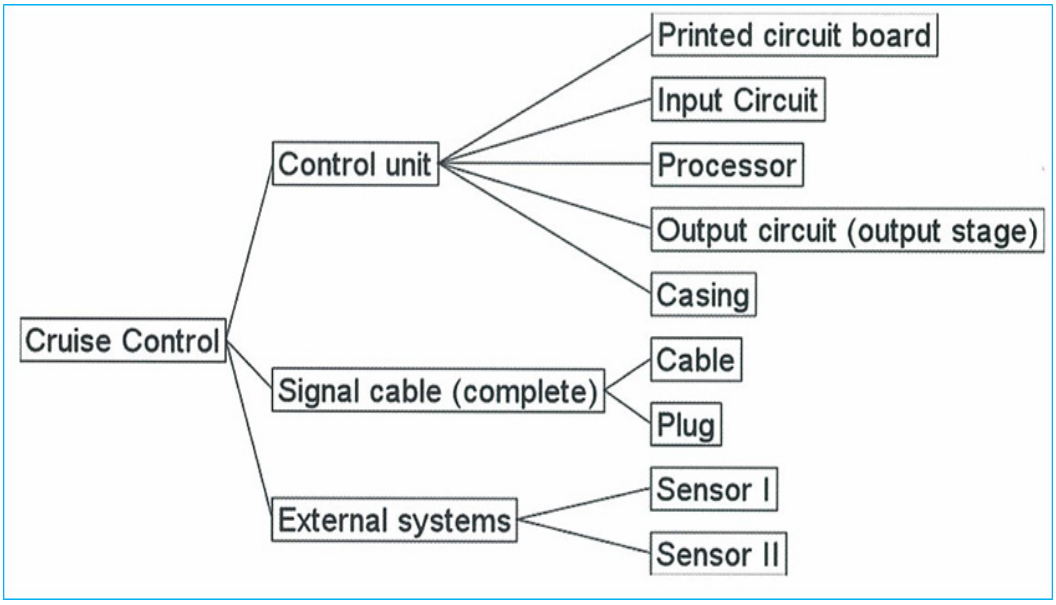


그림 249 구조 분석 화면

- STEP 2. Function Analysis (기능 분석) 수행
  - 앞서 STEP 1 수행을 통해 해당 구성품이 식별되어 도구 기반의 입력이 되었다면, 해당 구성품이 지니고 있는 기능요구사항을 입력하여야 한다. 입력된 기능적 정보 또는 구성품 간의 인터페이스 정보가 식별된다면, 해당 정보를 기반으로 IQ-FMEA에서는 드래그 형식으로 연동을 갖추어 추적성 관계를 확립한다.
  - 앞서 정의한 구조적 정보가 지닌 기능들 간의 관계 정립을 수행하기 위해 기능들 사이의 관계를 Tree형태로 추적성 관계를 설정해야 한다. 이러한 관계 정립을 통해 IQ-FMEA를 통해 Function Net를 구성한다.

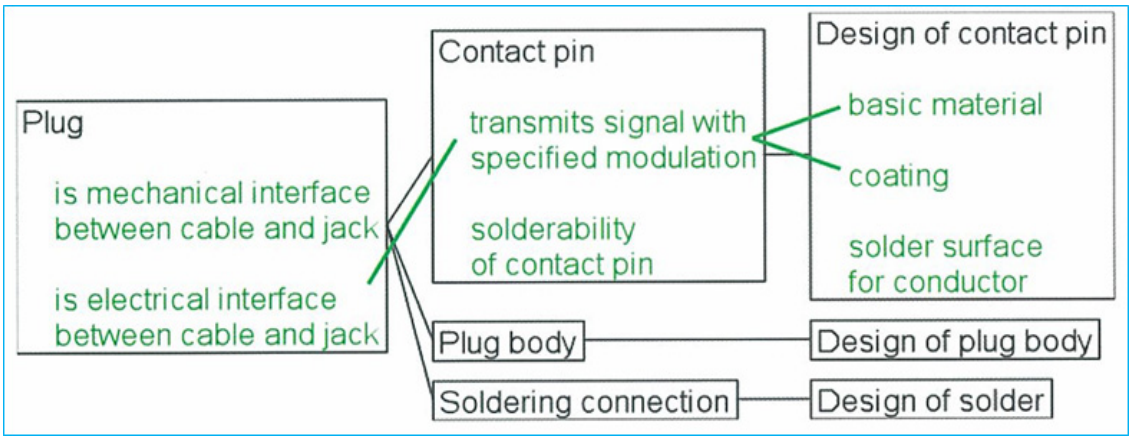


그림 250 기능 분석 화면

- STEP 3. Failure Analysis (고장 분석) 수행
  - IQ-FMEA를 통해 개별 구성품/기능이 식별되어 입력되었다면, 개별 기능요구사항의 오류로부터 어떠한 오작동이 발생되는지에 대한 문장을 기능과 연동해 추적성을 확립해야 한다. 추적성은 위에서 언급한 바와 같이, 드래그 앤 드래그 형식으로 표현이 가능하다.
  - 개별 기능으로부터 발생될 수 있는 오작동을 나열하고 상호 어떠한 인과 관계로 영향이 미치는지 분석한다.

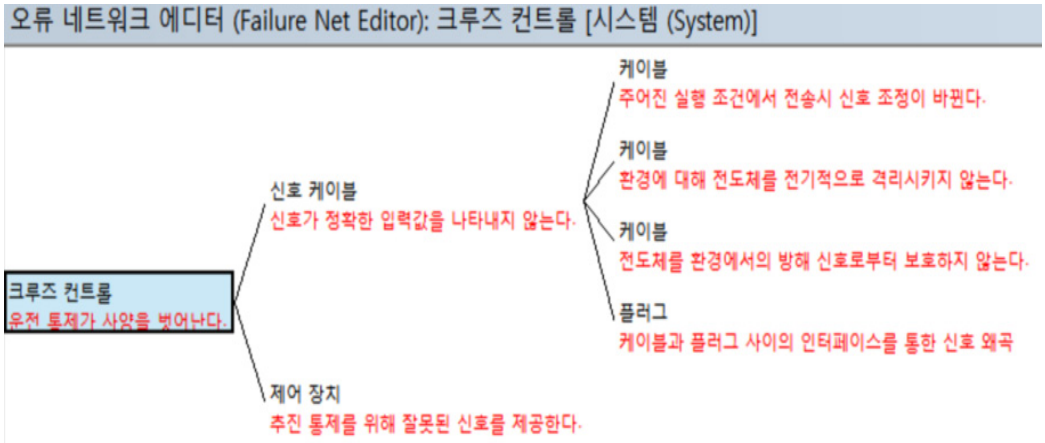
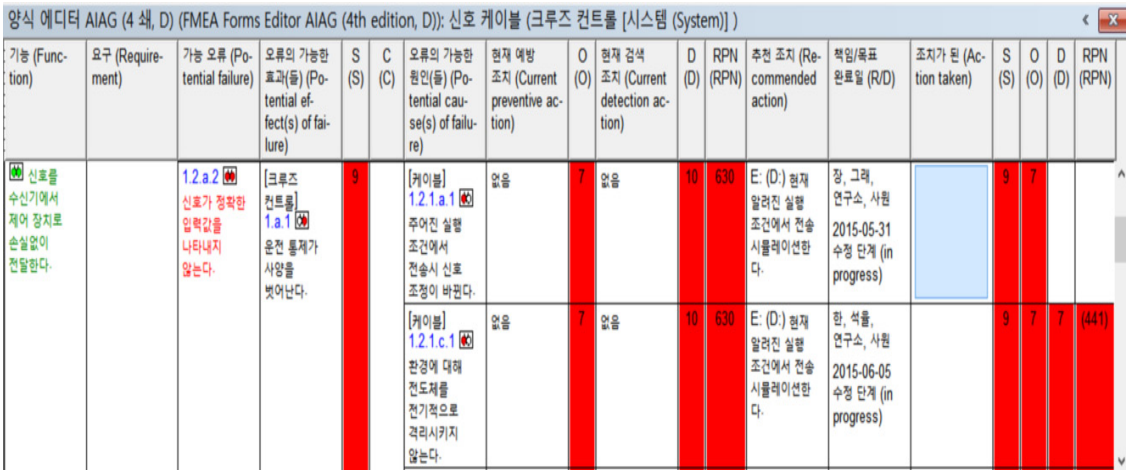


그림 251 고장 분석 화면

- **STEP 4.** Action Analysis (활동 분석 및 조치)
  - IQ-FMEA의 강점은 다양한 FMEA 작성 시트를 지원한다. 표준 템플릿 양식을 하나 선정하게 되면, 앞서 진행을 통해, 입력한 구성품, 기능, 오류, 추적성 정보가 자동으로 입력되게 된다. 이러한 정보를 근간으로 위험도 평가가 수행되어야 한다. 위험도 평가를 통해, 허용치 보다 높게 식별된 항목은 설계에 조치가 취해져야 한다. 특히, 이러한 요소는 시스템 안전성 확보를 하는데 있어서 중요한 요소가 된다.


양식 에디터 AIAG (4 쇄, D) (FMEA Forms Editor AIAG (4th edition, D)): 신호 케이블 (크루즈 컨트롤 [시스템 (System)])

기능 (Function)	요구 (Requirement)	가능 오류 (Potential failure)	오류의 가능한 효과(들) (Potential effect(s) of failure)	S (S)	C (C)	오류의 가능한 원인(들) (Potential cause(s) of failure)	현재 예방 조치 (Current preventive action)	O (O)	현재 검사 조치 (Current detection action)	D (D)	RPN (RPN)	추천 조치 (Recommended action)	책임/목표 완료일 (R/D)	조치가 된 (Action taken)	S (S)	O (O)	D (D)	RPN (RPN)
신호를 수신기에서 제어 장치로 손실없이 전달한다.		1.2.a.2 신호가 정확한 입력값을 나타내지 않는다.	[크루즈 컨트롤] 1.a.1 운전 통제가 사양을 벗어난다.	9		[케이블] 1.2.1.a.1 주어진 실행 조건에서 전송시 신호 조정이 바뀐다.	없음	7	없음	10	630	E: (D:) 현재 알려진 실행 조건에서 전송 시뮬레이션한다.	장, 그래, 연구소, 사원 2015-05-31 수정 단계 (in progress)		9	7		
						[케이블] 1.2.1.c.1 환경에 대해 전도체를 전기적으로 격리시키지 않는다.	없음	7	없음	10	630	E: (D:) 현재 알려진 실행 조건에서 전송 시뮬레이션한다.	한, 석률, 연구소, 사원 2015-06-05 수정 단계 (in progress)		9	7	7	(441)

그림 252 활동 분석 및 조치 화면

D-2. Reliability Workbench FaultTree++

○ 개요

Fault Tree Analysis(FTA)는 Top Event라고 불리는 최상위 고장으로부터 시작해서 최하부의 발생 가능한 Event까지 추적해 가는 Top-down 방식의 분석 방법이다. 따라서 하위 Level의 개별 고장들이 각 Top-Event로 연결되는 고장의 경로(Failure Path)를 시각적으로 보여주는 위험 분석 기법이다.

Reliability Workbench FaultTree+ 도구는 이러한 FTA 위험 분석 기법을 지원하는 도구이다. 본 도구는 세계적으로 가장 유명한 Fault Tree 소프트웨어 패키지인 FaultTree+가 Reliability Workbench에 통합된 것이다. FaultTree+는 전 세계적으로 항공, 방위, 철도, 자동차, 석유 & 가스, 원자력, 화학 공정 등의 다양한 주요 산업 분야에서 시스템의 안전을 평가하는데 사용되고 있다. Reliability Workbench의 FaultTree+는 Fault Tree, Event Tree, Markov 분석 기능이 통합된 강력한 시스템 신뢰성 분석 도구이다.

○ 주요기능

- Fault Tree 분석

FaultTree+ 모듈은 Fault Tree 다이어그램을 작성하는데 사용하기 쉬운 인터페이스를 제공한다. 또한, Fault Tree 다이어그램은 시스템을 구성하는 하부 구성품간의 계층, 구성 요소의 고장, 구성요소들이 어떻게 시스템 고장의 원인이 되는지를 나타낼 수 있다. Fault Tree의 Top Event는 이후 Event의 상세화 과정을 거쳐 구성품과 이벤트의 고장 종속관계를 표현할 수 있기에 컴포넌트 고장들을 연결해서 볼 수 있다. 고장과 수리적 데이터는 시스템 구성요소에 할당된다. Fault Tree 모듈은 시스템과 식별된 위험 구성요소의 신뢰성, 유효성 파라미터를 계산하기 위한 세밀한 분석 수행을 지원한다.

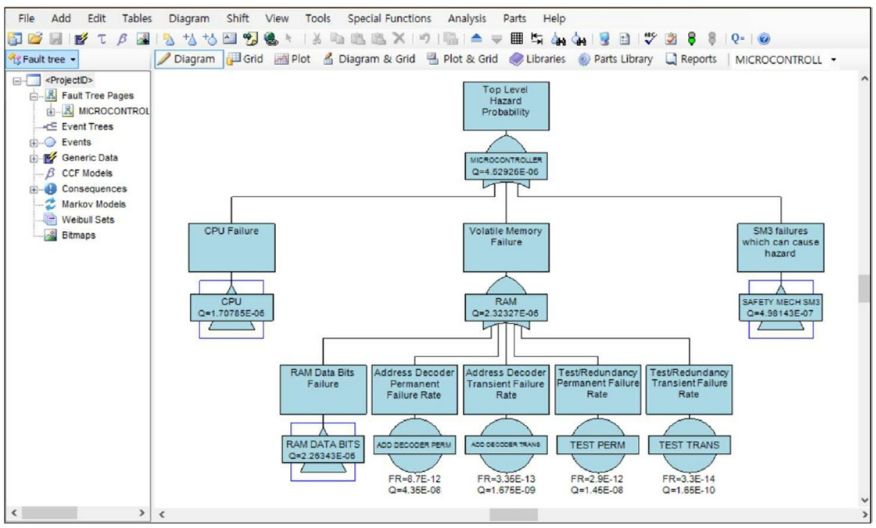


그림 253 Fault Tree 분석 화면

- Event Tree 분석

Event Tree 모듈은 큰 문제들을 다룰 수 있고, 완벽하게 성공 로직을 처리할 수 있다. Event Tree 모델은 Fault Tree 모델을 별도로 만들거나, Event Tree 확률의 소스로 Fault Tree 분석 게이트 결과를 사용할 수 있다. Event Tree 모듈은 주 이벤트와 보조 이벤트 트리의 다양한 가지들과 다양한 결과 카테고리 모두를 다룰 수 있다.

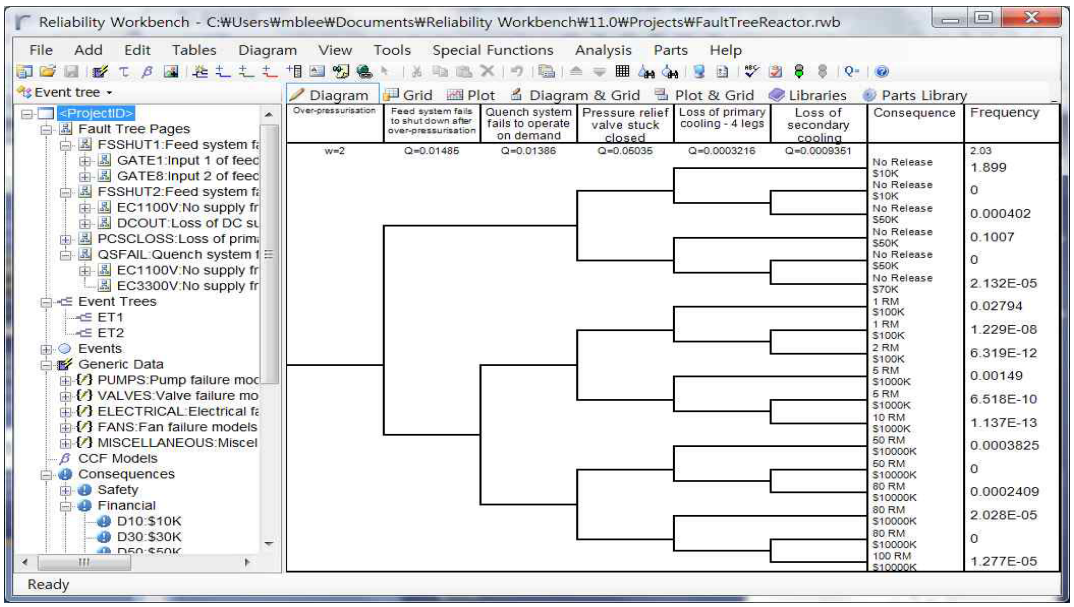


그림 254 Event Tree 분석 화면

- Markov 분석

Markov 모듈은 사용자가 쉽게 상태 전이 다이어그램을 만들고, 복잡한 문제들을 해결하기 위한 숫자 통합을 수행할 수 있도록 한다. Mark 분석 모델로 만들어진 모들은 Fault Tree와 Event Tree 분석 모듈 안에 있는 Basic Event와 연결되어 진다.

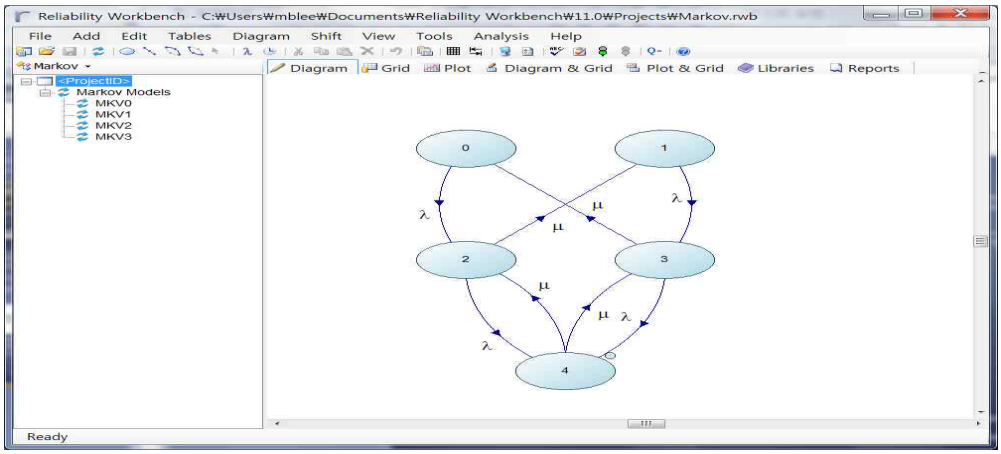


그림 255 Markov 분석 화면

○ 도구 사용법

- STEP 1. (최상위 이벤트 식별 수행)
  - 도구기반의 FTA 수행 시 첫 번째 단계로 FTA 최상위 이벤트는 분석하고자 하는 대상에 의해 발생 할 수 있는 위험원(Hazard)을 식별해 선정하는 단계이다. 따라서 시스템의 주요 고장을 시스템의 Top Event로 정의되어야 한다. Top Event로 정의되기 위해서는 발생여부에 대한 명확한 정의가 되어야 하며, Event가 하부로 분해(Decompose)되어야 한다.
  - 식별된 Top Event와 하부 이벤트는 Reliability Workbench FaultTree+의 지원 기호 및 Logic Gate의 조합을 통해 나타내어야 한다.

- STEP 2. (Fault Tree 작성) 수행
  - 정의된 최상위 이벤트에 직접적으로 영향을 주는 모든 원인을 분석하기 위한 Fault Tree Model을 작성해야 한다. Fault Tree는 원인과 인과관계를 확인하고 다른 원인을 찾을 수 없을 때 까지 분석하고 작성되어야 한다.

- STEP 3. (Fault Tree Analysis) 수행
  - 앞서 Fault-Tree 작성이 완료된다면, Fault-Tree 에 대한 분석이 수행되어야 한다. 개별 구성품이 지니고 있는 Event가 지닌 발생확률을 기반으로 시스템의 신뢰도 분석이 수행이 가능하여 도구기반의 정량적인 신뢰도 분석이 가능해진다.

○ 도구 기반 FTA 작성 유의사항

- 도구에서 제공하는 기호와 이벤트 블록을 활용하여 하나의 결함을 명확하게 식별해야 한다.
- 각각의 고장을 부품 혹은 시스템 고장으로 명확히 정의해야 한다.
- 하나의 게이트(Gate)에 대한 모든 입력들은 하위로 전개되기 전에 완전히 정의 되어야 한다.
- 주어진 게이트들의 조합에 대한 모든 입력들은 Fault Event 이거나 Basic Event 이어야 한다.
- 다른 분기(branch)를 작성하기 전에 하나의 분기에 대해서 basic event까지 완전하게 정의해야 한다.
- 하나의 Fault Tree는 중복 부분이 없어야 한다.
- 분석을 수행하기 전에 Fault Tree의 작성이 완료 되어야 한다.

부록 E. 철도 안전성 분석 기법 선정 및 활용에 관한 해설

E-1. 안전성 분석을 수행하기 위한 준비

○ 목적

국내 많은 철도 도메인 업체들이 대기업 위주의 기업이 아닌 많은 영세기업을 주축으로 철도 산업이 형성이 되어 있다. 이와 관련해, 안전성 분석을 수행하기 위해서는 선행적으로 공학적 설계기법에 대한 충분한 인지와 설계 산출물을 바탕으로 안전성 분석에 대한 시작을 수행할 수 있다. 따라서 앞서 언급한, 조직이 갖추지 못한 설계와 관련한 산출물이 없거나 또는, 부족한 정보를 가지고, 본 안전성 분석 가이드를 활용하기에 어려움이 상대적으로 많이 느끼는 대상(개인 및 업체)의 활용에 대한 가이드를 별도로 제공하고자 한다.

E-2. 안전성 분석을 수행 전략

○ 설명

앞서 설명한바와 같이, 안전성 분석을 수행하기 위해서는 다양한 설계 산출물을 기반으로 수행이 되어야 한다. 토대가 되는 자료의 부족 또는 부적절한 산출물을 보유 시 수행방안에 대해 정의하고자 한다.

○ 수행방안

첫 번째, 안전성 분석을 수행하기 위해서, 기존 설계 산출물에 대한 자료가 없거나 또는 부적절한지에 대한 의문을 갖아야 한다.

기존 자료가 없거나 부적절할 시 안전성 분석을 수행하기 위한 적합한 활동은 시스템 분석과 관련 모드로부터 사상(Event) 발생빈도를 추정할 필요가 있다. 이때 활용되는 기법들이 결함수 분석(FTA)이나 사상수 분석(ETA) 등이 있다. 입력 자료로는 운용상의 경험이나 출판 자료와 같은, 고장이나 인적오류를 포함한 관련 사상(Event)에 관한 자료들이 바람직하지 않은 사상(Event)의 빈도 추정치를 얻기 위하여 결합된다. 특히, 예측기법을 이용할 때에는 적절한 허용오차가 도입되었는가를 확인하는 일이 중요하며, 운용 장비의 노후화에 따른 장비 및 구조적 고장 빈도를 추정하기 위해서는 시뮬레이션 기법이 필요할 수도 있다.

아무런 데이터가 없이 위험원 분석(Hazard Analysis)을 수행하기 위해서는 개발 중인 대상 시스템에 대한 정의가 선행적으로 수행되어야 한다. 대상 시스템이 지닌 운영 조건을 식별하고 해당 운용조건에서 시스템이 지녀야할 기능을 식별하는 과정을 거쳐 기능을 식별하게 된다. 또한, 시스템의 운용환경에 분석과정을 거쳐 시스템 환경의 확인

을 수행하는 과정을 거치게 된다. 따라서 운용환경에서의 위험사건의 분석을 통해 위험요인을 확인하게 된다. 초기 위험원 식별을 위해서는 예비위험원분석(PHA) 기법이 대표적으로 수행된다. 또한, 고장 원인을 분석하기 위해서는(FMECA/FTA) 수행을 통해, 확인할 수 있는 방법이다. 식별된 위험원을 평가하기 위해서는 시나리오 개발과 발생빈도, 사고에 따른 결과 심각도에 대한 평가가 수반되어야 하기 때문에, 평가지표가 될 수 있는 기준이 필요로 하다. 위험도에 대한 평가된 결과는 허용 수준인지 아닌지에 대한 평가를 통해, 위험도 경감 대책의 수립 또는 안전대책이 설계에 반영되는 과정의 연속으로 진행되어야 한다.

두 번째, “만약 ~한다면(What if)? 하는 질문을 떠올리게 함으로써 위험원 발굴에 관한 지식을 가지고 앞서 예측하도록 하는 방법이다. 대표적인 것으로 위험원 및 위험도 및 운용도 연구(Hazard and Operability Study; HAZOP), 고장 모드 및 영향분석(FMEA) 등이 있다. 하지만, 위험성의 판단에는 주관적인 요소가 있다는 점과, 발굴된 위험요소들만이 시스템을 위협하는 유일한 위험요소가 아니라는 부분에 대해서 인지가 필요하다. 전통적으로 안전성 분석 기법들과 사상수 분석(Event Tree Analysis) 등이 해당이 된다. 더욱이 채택된 실제 기법에 관계없이, 일반적인 위험원 식별 과정에서 사람과 조직의 과오로부터 많은 사고의 중요한 요인이라는 사실을 올바로 인식하는 것 또한 중요한 요소이다. 그러므로 사람과 조직의 과오 및 오작동을 포함한 재해사고의 시나리오 위험원 발굴과정에서 포함 되어야 하며, ‘하드웨어’ 및 ‘소프트웨어’ 측면만을 지향해서는 안 된다. 앞서, 설명한 바와 같이 위에서 언급한 안전성 분석을 수행하기 위해서는 아래 표의 산출물을 근간으로 필요로 하게 된다. 특히, ‘1.3 시스템 요구사항’과 관련된 문서는 시스템 안전성 분석을 수행하는데 있어서 중요한 근간의 문서로 활용되어 진다.

표 352 SIL 인증을 수행하기 위한 생명주기 단계별 안전성 평가 수행 산출물

No.	영역	ID	생명주기 단계	산출물
1	시스템	1.1	계획	Safety plan
				Quality management plan
				System configuration management plan
				System verification and validation plan
		1.2	위험도 분석	Risk analysis report (PHA)
				Hazard Log
		1.3	시스템 요구사항	System requirements specification
				System safety requirements specification
				System requirements verification report
				System Validation test plan
				Traceability Matrix

No.	영역	ID	생명주기 단계	산출물
		1.4	H/W 설계	System architecture specification
				System FMEA
				HW design specification
				Drawing
				HW test plan and report
				RAM prediction
				Detail hardware FMEA
				System reliability modelling: FTA
				HW architecture and design verification report
				Manuals for operation, maintenance, installation
		1.5	시스템검증	System validation test report
				Environmental Stress and EMC test report
				Internal audit report for quality and safety management
				Generic Application Safety Case

부록 F. 철도 안전 관련 법령 및 기준

○ 목적

국내 철도산업에서 고려되는 철도안전 관련 지침 및 법령에 관해 다양하게 존재한다. 하지만, 안전이라는 목적을 달성위한 지향하는 바가 같기에 상당부분 유사 또는 연계성을 지니고 있다. 따라서 철도관련 시스템안전성 관련 부분과 관련된 부분의 사항에 대해 다음과 같이 요약 정리하였다.

○ 요약

국내 철도산업에서 준수해야하는 관련 대표 안전 지침/법령에는 “국토교통부 고시 제 2015-477호”, “철도안전법”이 있으며, 특히, 철도안전법의 제 7조 5항에 따라, 고시한 “철도 안전관리체계 기술기준”을 따른다면, 철도안전법에서 요구하는 IEC 62278에서 언급하는 사항에 대해 상당부분 커버할 수 있는 수행을 할 수 있게 된다.

○ 내용

“철도안전관리체계”란 철도운영자 및 철도시설관리자(이하 “철도 운영자 등”이라 한다)가 철도를 운영하거나 철도시설을 관리하기 위하여 갖추어야 하는 인력, 시설, 장비, 운영절차 및 비상대응계획 등 안전관리에 관한 유기적 체계를 말하며, 철도안전관리시스템(SMS : Safety Management System), 열차운행체계 및 유지관리체계로 구성된다.

표 353 철도안전관리체계 주요 용어

‘철도안전관리체계’ 주요 용어	
위험도(Risk)	위험 요인에 의한 발생가능성(Probability)과 심각도(Severity)에 따라 측정되는 위험의 정도를 말한다.
위험도 평가 (Risk Assessment)	위험 요인을 분석하고 해당 위험 요인에 의한 철도 사고 및 장애 등의 발생가능성과 심각도를 평가한 후 저감대책을 수립 · 시행하는 일련의 과정을 말한다.
안전정보	철도안전관리에 활용될 수 있는 모든 자료를 말한다.
변경관리	철도 운영자 등의 안전관리체계에 영향을 주는 내 · 외부의 변화에 따라 새롭게 발생하거나 변경되는 위험을 파악하고, 통제하는 것을 말한다.
적격성	철도안전관리 업무수행에 적합한 지식, 경험 및 능력을 갖춘 정도를 말한다.
철도관련법령	「철도산업발전기본법」, 「철도사업법」, 「철도안전법」, 「철도건설법」, 「도시철도법」 등을 말한다.

철도안전관리시스템(SMS) 프로그램에는 다음의 항목이 포함되어야 한다.

1) 철도안전관리시스템 개요

2) 철도안전경영

3) 문서화

4) 위험관리

5) 요구사항 준수

6) 사고조사 및 보고

7) 내부 점검

8) 비상대응

9) 교육훈련

10) 안전정보

11) 안전문화

특히, 위 “4. 위험관리”에 해당하는 사항의 구성은 다음과 같다.

4.1 위험 관리

4.1.1 위험도 평가 및 관리

4.1.2 위험관리 절차

4.1.3 위험도 평가 절차

4.1.4 위험도 관리 기준

4.1.5 설계단계 위험도 평가의 활용

철도 운영자 등은 철도차량과 철도시설의 설계단계부터 식별한 위험요인, 위험도 평가 및 결정된 안전대책 등의 적정성을 확인하고, 이에 대한 결과물을 운영단계에 활용하여야 한다.

4.1.6 위험도 평가 등의 문서화

철도 운영자 등은 식별된 위험요인, 위험도 평가 및 결정된 안전대책 결과를 향상 최신의 것으로 문서화하고 유지하여야 한다.

4.2 안전대책

4.2.1 안전대책 수립

철도 운영자 등은 위험도를 관리하기 위한 안전대책을 수립, 실행 및 유지하여야 한다.

한국철도기술연구원 “안전성 분석 매뉴얼 목차” 구성은 다음과 같다. 구성 목차를 바탕으로 수행해야 할 활동 및 준비되어야 할 항목에 대해서 참조할 수 있을 것이다.

1. 일반사항

1.1 개요

1.2 관련 규격 및 법규

1.3 목적 및 적용범위

2. 열차제어시스템의 안전성

2.1 안정성 활동 정의

2.2 안전성 활동 생명주기

3. 위험원 도출

4. 위험원 분석기법

4.1 PHA

4.2 SHA

4.3 IHA

4.4 OSHA

5. 위험도 평가

6. 위험도 수용기준

7. V&V

8. Safety Case

8.1 Safety Case 개요

8.2 품질관리보고서

8.3 안전관리보고서

8.4 기술안전보고서

9. 안전성 분석 산출 문서

10. 결론

부록 G. 철도 분야 안전성 분석 수행을 위한 위험원 분석 기법 분류

G-1. 시스템 개발 단계에 따른 분류

○ 목적

IEC 62278 표준은 시스템 개념 설계 단계로부터 폐기 단계에 이르기 까지 전 생명주기 단계를 커버한 안전성 설계 활동을 지원하고 있다. 마찬가지로 위험원 기법 역시, 생명주기 단계별 적재적소의 기법이 선정 및 활용되어야 한다. 관련해, 시스템 개발 단계에 따른 분류가 보편적 기준이기에 다음과 같이 정의한다.

○ 내용

시스템이 지는 생애주기를 시스템 생명주기라고 하는데, 시스템 개발 단계 중 언제 분석하느냐에 따라 시스템에 관련된 정보량도 다르며, 시스템의 세부 설계수준도 차이가 있다. 아래 기법이 생명주기 단계별 순서라고 보면 된다.

- 예비위험원분석(PHA)
- 결함위험원분석(FHA)
- 시스템위험원분석(SHA)
- 서브시스템위험원분석(SSHA)
- 운용위험원분석(O&SHA)

G-2. 수리적 방법에 따른 분류

해당 분류는 안전성 분석 활동에 대한 분석한 결과를 수학적으로 어떻게 정리하고 제시하느냐에 따른 분류이다. 수리적 방법은 크게 정성적 분석(Qualitative Analysis)과 정량적 분석(Quantitative Analysis)로 구분될 수 있다. 정량적 분석 방법은 시스템 구조를 기능적으로 분석하되, 수학적 평가지수를 활용하지 않은 방법이다. 주로 신뢰도/가용도 모형을 가정하고, 도표를 이용하여 시스템/ 부품의 결함 모드, 고장 메커니즘, 고장의 영향 및 결과 등을 결정하여, 보전 및 수리 전략 등을 결정한다. 현장에서 관련 전공 지식이 낮은 수준의 관련자라도 쉽게 이해할 수 있으며, 정량적인 방법에 비해 노력이 적게 든다는 장점이 있으나, 논리력과 합리성이 약하다는 단점이 지적되기도 한다.

○ 정성적 분석 기법 관련 대표기법

고장 모드 및 영향분석(Failure Modes and Effects Analysis, FMEA)이 있으며, 전통적인 위험성 분석기법들은 모두 여기에 해당한다.

① 수학적 평가지수 활용보다는 시스템 구조를 기능적으로 분석한다.

② 지식수준이 낮은 사람도 쉽게 이해할 수 있다.

③ 정량적 방법에 비해 노력이 적게 든다.

④ 논리력과 합리성이 약하다.

정량적 분석(Quantitative Analysis)은 시스템의 기능특성을 나타내는 수학적 신뢰도, 가용도 모형을 구성하고, 부품 신뢰도 자료를 획득하거나 정의하여 고장률이나 고장확률 같은 수학적 특성을 가지고 시스템을 평가하는 방법이다. 정성적 분석에 비해, 전문적인 지식을 요구하고, 노력도 많이 소요된다는 단점이 있으나, 부품의 시스템 안전 요구사항(Criticality)나 민감도 분석(Sensitivity Analysis)도 가능하여, 중복구조와 보전정책으로 인한 시스템 성능의 개선도를 평가하는 등 다양하고 깊이 있는 분석결과를 얻을 수 있다는 장점이 있다.

○ 정량적 분석과 관련한 위험분석의 대표 기법

시스템 안전 요구사항 분석(Criticality Analysis, CA)이나 신뢰도 블록 다이어그램 분석 (Reliability Block Diagram Analysis) 등이 있다.

① 고장률 또는 고장확률과 같은 수리적 특성으로 시스템을 평가한다.

② 부품의 criticality 또는 sensitivity analysis도 가능하다.

③ 다양하고 깊이 있는 분석결과를 얻을 수 있다.

④ 정성적 분석에 비해 전문적 지식을 요구된다.

⑤ 정량적 접근에 따른 노력이 많이 소요된다.

G-3. 논리적 방법에 따른 분류

○ 목적

일반적으로 설계 관점에 대해서도 접근하는 방식에 따라, Top-down 방식의 접근과, 반대로 Bottom-up 방식의 접근을 기반으로 수행이 가능하다. 안전성 분석 역시, 목표하는 바에 부합하는 방식의 안전성 분석 기법을 선정하여 수행한다면, 보다 효율적이고 원하는 목표하는 부합하는 안전성 분석 기법을 선정하여야 한다.

○ 귀납적 방법(Inductive Method)

귀납적 방법은 시간경과에 다라 원인으로부터 시작하여 결과를 추론해 나아가는 것이다. 즉, 시스템 안전성 분석의 경우, 개별 단위 부품이라는 하위 수준의 결함이 다음 상위 수준의 성능에 미칠 효과를 추정하고, 그 다음 상위 수준에 미칠 영향, 이렇게 분

석수준을 높여가면서(Bottom-up) 결함모드를 찾아내는 것이다. 그리하여 분석수준이 최고수준인 시스템에 이르게 되면 위험성 분석이 끝나게 되는 것이다. 그렇지만, 시스템이 점차 완성 단계에 이르러야 부품의 결함 모드와 효과가 확인되는 것이 보통이므로, 이 방법은 대체로 이후의 설계단계에 이용되는데, 모든 단일 결함(Single Fault)를 찾아내는 것이 힘들다는 단점을 지니고 있다. 이와 관련한 기법으로는 결함수 분석(Fault-Tree Analysis)를 제외하고 앞서 언급한 모든 분석 기법들이 여기에 속한다. 앞서 설명한 귀납적 접근 방식에 대한 특성을 요약하자면 다음과 같다.

① 시간의 경과에 따라 원인에서 결과를 추론한다.

② Bottom-Up 접근 방법으로 수행한다.

③ 후기 설계단계에서 이용을 한다.

④ 단일 결함(Single Fault)을 찾는 것이 매우 힘들다.

○ 연역적 방법(deductive Method)

연역적 방법의 본질은 시간의 경과를 거슬러 결과로부터 원인으로 추론해 가는 것이다. 다시 말해, 시스템과 같은 높은 수준에서부터 서브-시스템이나 부품으로, 차례대로 분석수준을 낮춰가며(Top-down) 상위 수준의 바람직하지 않은 결과를 가져 온 원인을 찾아가는 것이다. 그러므로 이러한 방법은 시스템의 상세규격이 미쳐 가주어지지 않은 시스템 설계의 초기 구상단계에서 유용한, 사상(Event) 지향적인 분석 방법이다. 또한, 연쇄적으로 관련된 시스템 고장을 포함하여 다중 고장을 평가하거나, 공통원인에 의한 결함이 있는가, 없는가, 또는 시스템 복잡성 때문에 시스템 결함을 나열하여 분석하는 것이 편한 경우에 어디든지 사용될 수 있다. 따라서 연역적 방법에 대한 특성을 요약한다면 다음과 같다.

① 시간 경과를 거슬러 결과에서 원인을 추론한다.

② Top-Down 접근 방법으로 수행한다.

③ 시스템 설계 초기 구성단계에서 매우 유용하다.

④ 사상지향적인(Event-Oriented) 분석 방법에 적합하다.

⑤ 다중 고장(Multiple-Fault)을 평가하고 공통원인에 의한 결함의 유무 파악에 있어서 매우 유용하다.

부록 H. 철도 표준에서 사용되는 기법 및 대책 목록

표 354 철도 표준에서 사용되는 기법 및 대책 목록

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
소프트웨어 요구사항 (7.2)	소프트웨어 요구사항 단계 기법 및 대책 (TableA.2)									7.2.4.15	a) 소프트웨어 요구사항 명세는 자연어로 된 문제에 대한 설명과 필요한 정형 또는 준정형 표기법을 포함해야 한다. b) 이 표는 명세를 명확하고 정확하게 정의하기 위한 추가 요구사항들이 반영되어 있다. 사용 중인 소프트웨어의 안전 무결성 등급을 충족하기 위하여 이러한 기술들 중 하나 이상을 선택해야 한다.
		정형기법		D.28	-	R	R	HR	HR		
		모델링 (TableA.17)	데이터 모델링	D.65	R	R	R	HR	HR		a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나 이상의 HR기법을 선택해야 한다.
			데이터 흐름 다이어그램	D.11	-	R	R	HR	HR		
			제어 흐름 다이어그램	D.66	R	R	R	HR	HR		
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR	HR	HR		
			시간 패트리넷	D.55	-	R	R	HR	HR		
			결정/진리 테이블	D.13	R	R	R	HR	HR		
			정형 기법들	D.28	-	R	R	HR	HR		
			성능 모델링	D.39	-	R	R	HR	HR		
			프로토타입/애니메이션	D.43	-	R	R	R	R		
			구조 다이어그램	D.51	-	R	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
		구조적 방법론		D.52	R	R	R	HR	HR		
		결정 테이블		D.13	R	R	R	HR	HR		
소프트웨어 아키텍처 및 설계 (7.3)	소프트웨어 아키텍처 단계 기법 및 대책 (TableA.3)									7.3.4.14	a) 소프트웨어 안전 무결성 등급 3, 4에 대한 기술의 승인된 조합은 다음과 같다. 1) 1, 7, 19, 22와 4, 5, 12, 21 중 한 가지; 2) 1, 4, 19, 22와 2, 5, 12, 15, 21 중 한 가지 b) 소프트웨어 안전 무결성 등급 1, 2에 대한 기술의 승인된 조합은 1, 19, 22와 2, 4, 5, 7, 12, 15, 21 중 한 가지이다. c) 이러한 문제 중 일부는 시스템 수준에서 정의될 수 있다. d) IEC 62279 요구사항에 따라 오류 검출 코드를 사용할 수 있다. 참고 기법 및 대책 19는 외부 인터페이스 용도로 사용된다.
		방어적 프로그래밍		D.14	-	HR	HR	HR	HR		
		결함 검출 & 진단		D.26	-	R	R	HR	HR		
		오류 정정 코드		D.19	-	-	-	-	-		
		오류 검출 코드		D.19	-	R	R	HR	HR		
		고장 단정 프로그래밍		D.24	-	R	R	HR	HR		
		안전성 백 기법		D.47	-	R	R	R	R		
		다양화 프로그래밍		D.16	-	R	R	HR	HR		
		복구 블록		D.44	-	R	R	R	R		
		역방향 복구		D.5	-	NR	NR	NR	NR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
		전방향 복구		D.30	-	NR	NR	NR	NR		
		고장 복구 재시도 방법		D.46	-	R	R	R	R		
		실행된 사례 기억		D.36	-	R	R	HR	HR		
		인공지능-결함 정정		D.1	-	NR	NR	NR	NR		
		소프트웨어의 동적 재구성		D.17	-	NR	NR	NR	NR		
		소프트웨어 오류 영향 분석		D.25	-	R	R	HR	HR		
		우아한 저하		D.31	-	R	R	HR	HR		
		정보 은닉		D.33	-	-	-	-	-		
		정보 캡슐화		D.33	R	HR	HR	HR	HR		
		완전하게 정의된 인터페이스		D.38	HR	HR	HR	M	M		
		정형기법		D.28	-	R	R	HR	HR		
		모델링 (TableA.17)	데이터 모델링	D.65	R	R	R	HR	HR		a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나 이상의 HR기법을 선택해야 한다.
			데이터 흐름 다이어그램	D.11	-	R	R	HR	HR		
			제어 흐름 다이어그램	D.66	R	R	R	HR	HR		
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR	HR	HR		
			시간 패트리넷	D.55	-	R	R	HR	HR		
			결정/진리 테이블	D.13	R	R	R	HR	HR		
			정형 기법들	D.28	-	R	R	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			성능 모델링	D.39	-	R	R	HR	HR		
			프로토타입/애니메이션	D.43	-	R	R	R	R		
			구조 다이어그램	D.51	-	R	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR	HR	HR		
		구조적 방법론		D.52	R	HR	HR	HR	HR		
		컴퓨터 지원 설계 및 명세도구를 통한 모델링	데이터 모델링	D.65	R	R	R	HR	HR		a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나 이상의 HR기법을 선택해야 한다.
			데이터 흐름 다이어그램	D.11	-	R	R	HR	HR		
			제어 흐름 다이어그램	D.66	R	R	R	HR	HR		
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR	HR	HR		
			시간 패트리넷	D.55	-	R	R	HR	HR		
			결정/진리 테이블	D.13	R	R	R	HR	HR		
			정형 기법들	D.28	-	R	R	HR	HR		
			성능 모델링	D.39	-	R	R	HR	HR		
			프로토타입/애니메이션	D.43	-	R	R	R	R		
			구조 다이어그램	D.51	-	R	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR	HR	HR		
소프트웨어 컴포넌트 설계 (7.4)	소프트웨어 설계 및 구현 단계 기법 및 대책									7.3.4.24 7.4.4.6	a) 소프트웨어 안전 무결성 등급 3, 4에 대해 승인된 기술 조합은 4, 5, 6, 8과 1, 2 중 한 가지이다. b) 소프트웨어 안전 무결성 등급 1, 2에 대한 승인된 기술 조합은 3, 4, 5, 6과 8, 9, 10 중 한 가지이다.

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
소프트웨어 컴포넌트 구현 및 테스팅 (7.5)	(TableA.4)										c) 메타프로그래밍은 컴파일하기 전 소프트웨어 소스 코드의 생성으로 제한되어야 한다.
		정형기법		D.28	-	R	R	HR	HR		
		모델링 (TableA.17)	데이터 모델링	D.65	R	R	R	HR	HR		a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나 이상의 HR기법을 선택해야 한다.
			데이터 흐름 다이어그램	D.11	-	R	R	HR	HR		
			제어 흐름 다이어그램	D.66	R	R	R	HR	HR		
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR	HR	HR		
			시간 패트리넷	D.55	-	R	R	HR	HR		
			결정/진리 테이블	D.13	R	R	R	HR	HR		
			정형 기법들	D.28	-	R	R	HR	HR		
			성능 모델링	D.39	-	R	R	HR	HR		
			프로토타입/애니메이션	D.43	-	R	R	R	R		
			구조 다이어그램	D.51	-	R	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR	HR	HR		
		구조적 방법론		D.52	R	HR	HR	HR	HR		
		모듈 방식		D.38	HR	M	M	M	M		
		컴포넌트 (TableA.20)	정보 은닉	D.33	-	-	-	-	-		a) 정보 은닉 및 캡슐화는 데이터 액세스에 대한 일반적인 전략이 없는 경우에만 권장된다. 참고 기법 및 대책 4는 내부 인터페이스를 위한 것이다.
			정보 캡슐화	D.33	R	HR	HR	HR	HR		
			파라미터 수 제한	D.38	R	R	R	R	R		
			모든 인터페이스	D.38	R	HR	HR	M	M		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			정의								
		설계 및 코딩 표준	코딩 표준	D.15	HR	HR	HR	M	M		a) 검증된 컴파일러 또는 번역기(translator)의 한 부분으로서 3, 4, 5의 기법이 제시될 수 있다.
			코딩 스타일 가이드	D.15	HR	HR	HR	HR	HR		
			동적 객체 사용 금지	D.15	-	R	R	HR	HR		
			동적 변수 사용 금지	D.15	-	R	R	HR	HR		
			포인터 사용 제한	D.15	-	R	R	R	R		
			재귀호출 사용 제한	D.15	-	R	R	HR	HR		
			조건 없는 점프 사용 금지	D.15	-	HR	HR	HR	HR		
			함수, 서브루틴과 메소드의 크기와 복잡도 제한	D.38	HR	HR	HR	HR	HR		
			함수, 서브루틴에 대한 진입/종료 시점 전략 및 방법	D.38	R	HR	HR	HR	HR		
			서브루틴 인자 수 제한	D.38	R	R	R	R	R		
			전역 변수 사용 제한	D.38	HR	HR	HR	M	M		
		분석 가능한 프로그램		D.2	HR	HR	HR	HR	HR		
		엄격한 형식의 프로그래밍 언어		D.49	R	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
		구조적 프로그래밍		D.53	R	HR	HR	HR	HR		
		프로그래밍언어 (TableA.15)	ADA	D.54	R	HR	HR	HR	HR		a) 언어 선택은 6.7 및 7.3에 제시된 요구사항에 기반을 두어야 한다. b) 특정 프로그래밍 언어를 배제하는 결정을 정당화할 필요는 없다. 참고1 프로그래밍 언어의 적합성 평가에 대한 자세한 내용은 D.54 '적합한 프로그래밍 언어'의 항목을 참조한다. 참고2 특정 언어가 테이블에 없으면 자동으로 제외되지 않는다. D.54를 준수해야 한다. 참고3 어플리케이션 프로그램을 실행하는데 필요한 선택된 언어와 관련된 런타임 시스템은 소프트웨어 안전무결성 등급에 따라 사용할 수 있어야 한다.
			MODULA-2	D.54	R	HR	HR	HR	HR		
			PASCAL	D.54	R	HR	HR	HR	HR		
			C or C++	D.54 D.35	R	R	R	R	R		
			PL/M	D.54	R	R	R	NR	NR		
			BASIC	D.54	R	NR	NR	NR	NR		
			Assembler	D.54	R	R	R	R	R		
			C#	D.54 D.35	R	R	R	R	R		
			JAVA	D.54 D.35	R	R	R	R	R		
			Statement List	D.54	R	R	R	R	R		
		언어 하위집합		D.35	-	-	-	HR	HR		
		객체지향 프로그래밍 (TableA.22)	객체지향 프로그래밍	D.57	R	R	R	R	R		a) 기존 프레임과 설계 패턴을 사용할 때 기존 소프트웨어의 요구사항은 이러한 프레임과 패턴에 적용된다. 참고1 객체 지향 접근법은 절차적 접근법과는 다른 정보를 제공하며, 다음 목록은 구체적인 고려 사항이 필요한 권고안을 포함한다. - 클래스 구조를 이해하고, 주어진 메소드 호출 시 실행될 소프트웨어 기능의 식별(기존 클래스 라이브러리를 사용할 때 포함) - 구조 기반 시험(Table A.13) 어플리케이션 도메인에서 클래스 구조로의 추적성은 덜 중요하다.

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
											참고2 의도된 소프트웨어의 일부에 대하여 유사한 작업을 성공적으로 해결하고 개발 담당자에게 잘 알려진 기존 소프트웨어에서 존재할 수 있다. 그런 다음 해당 프레임을 사용하는 것은 좋은 습관으로 간주된다.
			아키텍처 클래스들에 대한 어플리케이션 도메인 컨셉의 추적성	-	R	R	R	HR	HR		
			일반적으로 사용되는 클래스들과 디자인패턴의 조합과 같은 적절한 프레임 사용	-	R	R	R	HR	HR		
			객체지향 상세 설계	Table A.23	R	R	R	HR	HR		
			클래스들은 오직 하나의 목표만 가진다.	-	R	R	R	HR	HR		
			파생 클래스가 기본 클래스에 의해 구체화 되는 경우에만 상속이 사용된다.	-	R	HR	HR	HR	HR		a) 하나의 클래스는 하나의 책임, 즉 밀접하게 연결된 데이터 및 이러한 데이터에 대한 연산을 처리하는 것을 특징으로 한다. b) 객체 간의 순환 종속성을 방지하려면 주의가 필요하다.
			상속의 깊이는 코딩 표준에 의해 제한된다.	-	R	R	R	HR	HR		
			메소드의 상속은 엄격한 제어 하에 수행	-	R	R	R	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			다중 상속은 오직 인터페이스 클래스들에만 사용된다.	-	R	HR	HR	HR	HR		
			알 수 없는 클래스로 부터의 상속	-	-	-	-	NR	NR		
		절차적 프로그래밍		D.60	R	HR	HR	HR	HR		
		메타프로그래밍		D.59	R	R	R	R	R		
소프트웨어 아키텍처 및 설계 (7.3) 소프트웨어 컴포넌트 구현 및 테스팅 (7.5)	검증 및 시험 단계 기법 및 대책 (TableA.5)									6.2.4.5 7.3.4.32 7.3.4.39 7.4.4.10	a) 소프트웨어 안전 무결성 등급 3, 4의 경우 승인된 기법 조합은 3, 5, 7, 8과 1, 2, 6 중 한 가지이다. b) 소프트웨어 안전 무결성 등급 1, 2의 경우 승인된 기법 조합은 5와 2, 3, 8 중 한 가지이다. 참고1: 기법 및 대책 1, 2, 4, 5, 6, 7은 검증 활동을 위한 것이다. 참고2: 기법 및 대책 3, 8, 9, 10은 시험 활동을 위한 것이다.
		정형 증명		D.29	-	R	R	HR	HR		
		정적분석 (TableA.19)	경계값 분석	D.4	-	R	R	HR	HR		-
			체크리스트	D.7	-	R	R	R	R		
			제어 흐름 분석	D.8	-	HR	HR	HR	HR		
			데이터 흐름 분석	D.10	-	HR	HR	HR	HR		
			오류 추측	D.20	-	R	R	R	R		
			워크스루/설계 검토	D.56	HR	HR	HR	HR	HR		
		동적분석 및 시험	경계값 분석으	D.4	-	HR	HR	HR	HR		a) 테스트 케이스에 대한 분석은 서브시스템 수준에

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
		(TableA.13)	로부터 테스트 케이스 수행								서 이루어지며 규격 및/또는 규격 및 코드를 기반으로 한다.
			오류 추측으로부터 테스트 케이스 수행	D.20	R	R	R	R	R		
			오류 삽입으로부터 테스트 케이스 수행	D.21	-	R	R	R	R		
			성능 모델링	D.39	-	R	R	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	R	R	HR	HR		
			구조 기반 시험	D.50	-	R	R	HR	HR		
		측정 기준		D.37	-	R	R	R	R		
		추적성		D.58	R	HR	HR	M	M		
		소프트웨어 오류 영향 분석		D.25	-	R	R	HR	HR		
		코드시험적용범위 (TableA.21)	구문	D.50	R	HR	HR	HR	HR		a) 모든 SIL에 대해 수행 된 테스트에 대해 정량화된 측정 척도를 개발해야 한다. 이는 테스트에서 얻은 자신감과 추가 기법의 필요성에 대한 판단을 뒷받침할 수 있다. b) SIL 3 또는 4의 경우 컴포넌트 레벨에서 테스트 범위는 다음에 따라 측정해야 한다. - 2 및 3; 또는 - 2 및 4; 또는 - 5 또는 통합 수준에서의 테스트 범위는 하나 이상의 2, 3, 4 또는 5에 따라 측정해야 한다.
			분기	D.50	-	R	R	HR	HR		
			복합 조건	D.50	-	R	R	HR	HR		
			데이터 흐름	D.50	-	R	R	HR	HR		
			경로	D.50	-	R	R	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
											c) 이를 정당화할 수 있다는 점을 감안할 때 다른 테스트 범위 기준을 사용할 수 있다. 이러한 기준은 소프트웨어 아키텍처(Table A.3 참조)와 프로그래밍 언어(Table A.15 및 Table A.16 참조)에 따라 달라진다. d) 테스트할 수 없는 코드를 사용하는 모든 코드는 적절한 기법을 사용하여 정확함을 입증해야 한다.(예: Table A.19의 정적 분석) 참고1 구문 범위는 항목 2에서 5까지 자동으로 달성된다. 참고2 표의 테스트 범위 기준은 구조 기반(코드 기반, 화이트박스) 테스트에 사용된다. 기능(명세 기반, 블랙박스) 테스트를 위한 기법 및 대책은 Table A.14에 제시되어 있다. 참고3 적용 범위의 높은 비율은 일반적으로 달성하기 어렵다. 경계값(조항 D.4)과 동등 클래스 및 입력 분할 테스팅(조항 D.18)의 테스트 케이스 실행은 더 적은 수의 테스트를 통해 충분한 적용 범위를 얻을 수 있다. 참고4 2와 3의 차이점은 실제로 프로그래밍 언어의 수준과 복합 조건의 사용에 따라 달라진다. 예를 들어 단일 조건을 사용할 때, 2,3의 결과는 동일한 것으로 간주된다.
		기능 및 블랙박스 시험 (TableA.14)	원인 결과 다이어그램으로부터 테스트 케이스 수행	D.6	-	-	-	R	R		a) 시뮬레이션의 완성도는 소프트웨어 안전 무결성 등급, 복잡성 및 응용 프로그램 범위에 따라 달라진다.
			프로토타입/애니메이션	D.43	-	-	-	R	R		
			경계값 분석	D.4	R	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR	HR	HR		
			프로세스 시뮬레이션	D.42	R	R	R	R	R		
		성능시험 (TableA.18)	과부하/스트레스 시험	D.3	-	R	R	HR	HR		-
			응답 시기 및 메모리 제약	D.45	-	HR	HR	HR	HR		
			성능 요구사항	D.40	-	HR	HR	HR	HR		
		인터페이스 시험		D.34	HR	HR	HR	HR	HR		
통합 (7.6)	통합 단계 기법 및 대책 (TableA.6)									6.2.4.5 7.6.4.6 7.6.4.10	-
		기능 및 블랙박스 시험 (TableA.14)	원인 결과 다이어그램으로부터 테스트 케이스 수행	D.6	-	-	-	R	R		a) 시뮬레이션의 완성도는 소프트웨어 안전 무결성 등급, 복잡성 및 응용 프로그램 범위에 따라 달라진다.
			프로토타입/애니메이션	D.43	-	-	-	R	R		
			경계값 분석	D.4	R	HR	HR	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR	HR	HR		
			프로세스 시뮬레이션	D.42	R	R	R	R	R		
		성능시험 (TableA.18)	과부하/스트레스 시험	D.3	-	R	R	HR	HR		-

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			응답 시기 및 메모리 제약	D.45	-	HR	HR	HR	HR		
			성능 요구사항	D.40	-	HR	HR	HR	HR		
종합소프트웨어 테스팅 / 최종 확인 (7.7)	종합 소프트웨어 시험 단계 기법 및 대책 (Table A.7)									6.2.4.5 7.2.4.18	a) 소프트웨어 안전 무결성 등급 1, 2의 경우 승인된 기술조합은 1, 2 이다.
		성능시험 (TableA.18)	과부하/스트레스 시험	D.3	-	R	R	HR	HR		-
			응답 시기 및 메모리 제약	D.45	-	HR	HR	HR	HR		
			성능 요구사항	D.40	-	HR	HR	HR	HR		
		기능 및 블랙박스 시험 (TableA.14)	원인 결과 다이어그램으로부터 테스트 케이스 수행	D.6	-	-	-	R	R		a) 시뮬레이션의 완성도는 소프트웨어 안전 무결성 등급, 복잡성 및 응용 프로그램 범위에 따라 달라진다.
			프로토타입/애니메이션	D.43	-	-	-	R	R		
			경계값 분석	D.4	R	HR	HR	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	HR	HR	HR	HR		
			프로세스 시뮬레이션	D.42	R	R	R	R	R		
		모델링 (TableA.17)	데이터 모델링	D.65	R	R	R	HR	HR		a) 모델링 가이드라인을 정의하고 사용해야 한다. b) 적어도 하나 이상의 HR기법을 선택해야 한다.
			데이터 흐름 다이어그램	D.11	-	R	R	HR	HR		
			제어 흐름 다이	D.66	R	R	R	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			어그램								
			유한 상태 기계 / 상태 전이 다이어그램	D.27	R	HR	HR	HR	HR		
			시간 패트리넷	D.55	-	R	R	HR	HR		
			결정/진리 테이블	D.13	R	R	R	HR	HR		
			정형 기법들	D.28	-	R	R	HR	HR		
			성능 모델링	D.39	-	R	R	HR	HR		
			프로토타입/애니메이션	D.43	-	R	R	R	R		
			구조 다이어그램	D.51	-	R	R	HR	HR		
			순차 다이어그램	D.67	R	HR	HR	HR	HR		
	소프트웨어 분석 기법 및 대책 (TableA.8)									6.2.4.5	a) 하나 이상의 기법들은 사용되는 소프트웨어의 SIL 등급을 만족시키기 위해 선택되어야 한다.
소프트웨어 확인 (6.3)		정적소프트웨어분석 (TableA.19)		D.13 D.37	R	HR	HR	HR	HR		
			경계값 분석	D.4	-	R	R	HR	HR		
			체크리스트	D.7	-	R	R	R	R		
			제어 흐름 분석	D.8	-	HR	HR	HR	HR		
			데이터 흐름 분석	D.10	-	HR	HR	HR	HR		
			오류 추측	D.20	-	R	R	R	R		
			워크스루/설계 검토	D.56	HR	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
		동적소프트웨어분석 (TableA.13 TableA.14)		TableA.13 TableA.14	-	R	R	HR	HR		
		(Table A.13)	경계값 분석으로부터 테스트 케이스 수행	D.4	-	HR	HR	HR	HR		a) 테스트 케이스에 대한 분석은 서브시스템 수준에서 이루어지며 규격 및/또는 규격 및 코드를 기반으로 한다.
			오류 추측으로부터 테스트 케이스 수행	D.20	R	R	R	R	R		
			오류 삽입으로부터 테스트 케이스 수행	D.21	-	R	R	R	R		
			성능 모델링	D.39	-	R	R	HR	HR		
			동등 클래스 및 입력 분할 테스팅	D.18	R	R	R	HR	HR		
			구조 기반 시험	D.50	-	R	R	HR	HR		
		(Table A.14)	원인 결과 다이어그램으로부터 테스트 케이스 수행	D.6	-	-	-	R	R		a) 시뮬레이션의 완성도는 소프트웨어 안전 무결성 등급, 복잡성 및 응용 프로그램 범위에 따라 달라진다.
			프로토타입/애니메이션	D.43	-	-	-	R	R		
			경계값 분석	D.4	R	HR	HR	HR	HR		
			동등 클래스 및 입력 분할 테스	D.18	R	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
			팅								
			프로세스 시뮬레이션	D.42	R	R	R	R	R		
		원인 결과 다이어그램		D.6	R	R	R	R	R		
		이벤트 트리 분석		D.22	-	R	R	R	R		
		소프트웨어 오류 영향 분석		D.25	-	R	R	HR	HR		
소프트웨어 품질보증 (6.5)	소프트웨어 품질 보증 기법 및 대책 (TableA.9)										a) 테이블은 모든 단계의 담당자들에게 적용된다.
		ISO 9001 인증		7.1	R	HR	HR	HR	HR		
		ISO 9001 준수		7.1	M	M	M	M	M		
		ISO/IEC 90003 준수		7.1	R	R	R	R	R		
		회사 품질 시스템		7.1	M	M	M	M	M		
		소프트웨어 형상 관리		D.48	M	M	M	M	M		
		체크리스트		D.7	R	HR	HR	HR	HR		
		추적성		D.58	R	HR	HR	M	M		
		데이터 기록 및 분석		D.12	HR	HR	HR	M	M		
소프트웨어 유지보수 (9.2)	소프트웨어 유지보수 기법 및 대책									9.2.4.16	-

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		
	(TableA.10)										
		영향 분석		D.32	R	HR	HR	M	M		
		데이터 기록 및 분석		D.12	HR	HR	HR	M	M		
어플리케이션 데이터 혹은 알고리즘 개발: 어플리케이션 혹은 알고리즘의 시스템 컨피그 (8.4)	데이터 준비기법 및 대책 (TableA.11)									8.4.1.4	a) 소프트웨어 SIL 1및 2에서 승인된 기법의 조합은 1과 4이다. b) 소프트웨어 SIL 3및 4에서 승인된 기법의 조합은 1, 4, 5와 7 또는 2, 3, 6이다. 참고: B-29에 대한 설명은 프로그램에 관한 것이고, 이 맥락에서 기법 8은 데이터의 정확성에 대해서 정형 증명을 적용한다는 것이다.
		테이블 명세 기법		D.68	R	R	R	R	R		
		응용 특화 언어		D.69	R	R	R	R	R		
		시뮬레이션		D.42	R	HR	HR	HR	HR		
		기능 테스트		D.42	M	M	M	M	M		
		체크리스트		D.7	R	HR	HR	M	M		
		페이건 정밀 검사		D.23	-	R	R	R	R		
		정형 디자인 리뷰		D.56	R	HR	HR	HR	HR		
		정형 증명 (데이터)		D.29	-	-	-	HR	HR		
		워크스루		D.56	R	R	R	HR	HR		
어플리케이션 데이터/알고리즘 제품 (8.4.4)		어플리케이션 알고리즘을 위한 다이어그램언어 (TableA.16)	기능적 블록 다이어그램	D.63	R	R	R	R	R	8.4.4.1	-
			순차적 함수 도표(차트)	D.61	-	HR	HR	HR	HR		
			래더 다이어그램	D.62	R	R	R	R	R		
			상태 차트	D.64	R	HR	HR	HR	HR		

소프트웨어 개발 생명주기	생명주기 단계별 기법및대책	생명주기 단계별 상세 기법 및 대책		참조	SIL Level					IEC62279 참조	요구사항
		Lv1	Lv2		0	1	2	3	4		

부록 I. 형식 승인 vs. 철도 안전 가이드 소프트웨어 산출물 조견표

표 355 형식 승인 vs. 철도 안전 가이드 소프트웨어 산출물 조견표

항목	형식승인		철도 안전 가이드	
	기술검토서	주요 요구사항	관련 산출물	관련 내용 및 항목
위험도 분석	화재안전 위험도	-	예비위험원분석서 - 위험원 리스트 - 위험원 결과에 따른 시스템 결과 - 안전대책 / SIL 정의 및 할당	안전성 분석 가이드
	충돌안전 위험도	예비위험원(PHA) 분석 활동		
	탈선안전 위험도	예비위험원(PHA) 분석 활동		
계획	소프트웨어 안전 계획서	-	안전계획서	안전성 분석 가이드
	품질보증 계획서	- 소프트웨어 개발 요건 - 확인 및 검증 - 안전성 분석 요건	-	※ IEC 62279 - 6.5 Software quality assurance
		- 계획, 설계, 구현, 시험 - 설치 운영 및 유지보수 요건	-	
	형상관리 계획서	- 형상항목 식별 - 형상항목 통제 - 형상상태 기록 및 보고	-	
요구 사항	요구사항 정의	- 소프트웨어 기능, 성능 - 외부 연계 인터페이스 - 신뢰성 관련 요구사항 - 안전성 관련 요구사항 - 보안성 관련 요구사항	소프트웨어 요구사항 명세서	소프트웨어 요구사항
	요구사항 확인	- 시스템 요구사항과의 추적성 - 요구사항 적합성 - 하드웨어 및 외부 시스템과의 연계 적합성	소프트웨어 요구사항 검증 보고서	
설계	설계 정의	- 요구사항 기반 소프트웨어 구조 - 소프트웨어 상세 설계	소프트웨어 아키텍처 명세서	
			소프트웨어 설계 명세서	
			소프트웨어 컴포넌트 설계 명세서	
	설계 확인	- 요구사항과의 추적성	소프트웨어 아키텍처 및 설계 검증 보고서	소프트웨어 아키텍처 및 설계

항목	형식승인		철도 안전 가이드	
	기술검토서	주요 요구사항	관련 산출물	관련 내용 및 항목
		- 설계요서, 공정특성의 적합성 - 하드웨어 및 외부 시스템과의 연계 적합성	소프트웨어 컴포넌트 설계 검증 보고서	소프트웨어 컴포넌트 설계
			소프트웨어 인터페이스 명세서	소프트웨어 아키텍처 및 설계
구현	구현 정의	- 소프트웨어 소스코드로 구현 - 통합계획에 따라 통합	소스코드	소프트웨어 컴포넌트 구현 및 테스팅
	구현 확인	- 설계와 소스코드 추적성 - 소스코드 구성요소의 공정 특성 적합성 - 하드웨어 및 외부 시스템과의 연계 적합성	소프트웨어 소스코드 검증 보고서 소프트웨어 검증 보고서	
시험	시험 계획	- 단위시험 - 통합시험 - 시스템 시험 계획 및 절차 수립	소프트웨어 테스트 명세서	소프트웨어 요구사항
			소프트웨어 통합테스트 명세서 소프트웨어 / 하드웨어 통합테스트 명세서	소프트웨어 아키텍처 및 설계
			소프트웨어 컴포넌트 테스트 명세서	소프트웨어 컴포넌트 구현 설계
	시험 확인	- 시험계획 및 절차에 따라 시험 수행 결과 작성	소프트웨어 통합 테스트 결과서	통합
			소프트웨어 / 하드웨어 통합테스트 결과서	
			종합 소프트웨어 테스트 보고서	종합 소프트웨어 테스팅 / 최종 확인
		- 시험계획서, 절차서 및 보고서 내의 상호 관계에 대한 추적성	소프트웨어 통합 검증 보고서	통합
			종합 소프트웨어 테스트 검증 보고서	종합 소프트웨어 테스팅 / 최종 확인
설치	설치 수행	- 시스템에 정확히 설치되었고 요구되는 기능을 수행하는지 확인	소프트웨어 확인 보고서	종합 소프트웨어 테스팅 / 최종 확인
			소프트웨어 릴리스 및 배포 계획서 소프트웨어 배포 매뉴얼 릴리스 노트	소프트웨어 배포
		- 설치 관련 안전 요구사항 준수 여부	배포 검증 보고서	
유지 보수	유지보수 활동	- 소프트웨어 변경 요구 확인 - 변경 관련 문제점 보고 - 변경 적용 시 소프트웨어 생명주기 활동 재수행	릴리스 노트 소프트웨어 유지보수 계획서 소프트웨어 변경 기록 소프트웨어 유지보수 기록 소프트웨어 검증 보고서	소프트웨어 배포
	변경활동 확인	- 소프트웨어 운영 시 부적합 사항으로 인한 영향 평가 - 소프트웨어의 변경 사항에 대한 확인 및 검증 업무의 반복 정도 - 승인 된 변경 사항에 적합한 확인 및		소프트웨어 유지 보수

항목	형식승인		철도 안전 가이드	
	기술검토서	주요 요구사항	관련 산출물	관련 내용 및 항목
		검증 계획 개정 - 소프트웨어 생명주기에 따른 확인 및 검증 업무 재수행		

부록 J. 철도 산업 현황 조사를 위한 설문지

0. 일반사항

1. 기업명:

2. 직급:

3. 경력:

4. 제품 개발 분야: 신호 / 제어 / 관제 / 통신 / 전력 / 차량 / 기타 ( )

5. 제품 개발 영역: 하드웨어 / 펌웨어 / 응용소프트웨어 / 기타 ( )

1. 소프트웨어 공학 및 표준의 이해

1. 소프트웨어 개발업무에 있어서 체계적인 소프트웨어공학 기법의 적용이 중요하다고 생각하십니까?
   ① 매우 그렇다 ② 그렇다 ③ 보통 ④ 아니다 ⑤ 매우 아니다

2. 체계적인 업무 수행을 위해서라면 업무가 다소 가중되어도 소프트웨어 공학기법을 적용 할 의향이 있으십니까?
   ① 매우 그렇다 ② 그렇다 ③ 보통 ④ 아니다 ⑤ 매우 아니다

3. 다음 프로세스의 적용 경험과 본인의 이해 수준에 대해 모두 체크해 주시기 바랍니다.

프로세스 항목	적용경험	이해수준
프로젝트 관리 관련 프로세스	유 / 무	상 / 중 / 하
위험관리 관련 프로세스	유 / 무	상 / 중 / 하
이슈관리 관련 프로세스	유 / 무	상 / 중 / 하
형상관리 관련 프로세스	유 / 무	상 / 중 / 하
품질보증 관련 프로세스	유 / 무	상 / 중 / 하
측정 관련 프로세스	유 / 무	상 / 중 / 하
요구사항 관리 관련 프로세스	유 / 무	상 / 중 / 하
개발관련 프로세스(분석, 설계, 구현)	유 / 무	상 / 중 / 하
테스트 관련 프로세스	유 / 무	상 / 중 / 하
검토(Review) 관련 프로세스	유 / 무	상 / 중 / 하
유지보수 관련 프로세스	유 / 무	상 / 중 / 하

4. 제품 개발 시 소프트웨어 관련 표준 규격 또는 가이드를 적용하고 계시면 해당 표준 규격이나 가이드의 이름 또는 분류번호를 적어주십시오 (예, IEC 62279 또는 철도 어플리케이션 소프트웨어 개발 국제 표준 등)

5. (4번 항목에서 사용하시는 표준 규격 또는 가이드가 있을 경우) 해당 표준 규격이나 가이드의 적용 범위(회사, 팀, 특정 제품)와 적용 시 어려움이 많은 분야에 대한 의견을 적어주십시오 (예, 회사 전체의 모든 제품에 적용하고 있으며, 안전 요구사항 분석 시 표준에서 제시하고 있는 방법을 적용하는 것이 어렵다. 등)

2. 안전성 분석

6. 현업에서 사용하시거나 알고 계신 용어를 모두 선택해 주십시오.
   ① 기능 안전 (Functional Safety)
   ② 안전 무결성 등급 (SIL, Safety Integrity Level)
   ③ 안전 요구사항 (Safety Requirement)
   ④ 위험 분석 (Risk Analysis)

7. 안전 무결성 등급(SIL)을 반영한 제품이 있습니까? 있으면 제품 정보와 등급을 적어주십시오 (만약, 보안상의 이유가 있다면 제품의 수와 안전 무결성 등급(SIL)만을 적어주십시오)

8. 제품 개발 시 안전 무결성 등급(SIL)은 어떻게 산정하십니까?
   ① 발주처에서 제공
   ② 기존 관행대로 선정
   ③ 상황에 맞게 계산 및 선정
   ④ 모르겠음

9. 제품 개발 시 사전에 안전성 분석 프로세스를 수행 하십니까?
   ① 그렇다
   ② 아니다

※ 위 9번 항목에서 "그렇다"라고 답변 했을 경우 작성 (10번 ~ 13번)

10. 위험 분석 시 기록 관리되는 정보와 분석 된 정보를 어떤 산출물에 기록 관리하고 있는지 적어주십시오 (예, 위험 분석 기법을 적용해서 나온 모든 정보를 별도의 위험분석 보고서로 작성해서 관리한다.)

11. 안전성 분석 결과를 어떻게 활용하고 있는지 적어주십시오 (예, 별도의 안전 요구사항으로 식별하여 제품개발을 진행한다.)

12. 안전성 분석을 위해 사용하고 있는 기법을 모두 선택해 주시기 바랍니다.

① FTA (Fault Tree Analysis)
② FMEA (Failure Mode and Effects Analysis)
③ HAZOP (HAZard and OPerability study)
④ 기타 (
⑤ 없음

13. 회사에서 제품 개발 시 사용하는 안전성 분석 도구가 있으면, 해당 도구의 이름과 활용 방법을 적어주십시오

3. 소프트웨어 개발 방법

14. 제품 개발에 적용하시는 개발방법론 유형을 모두 선택해 주시기 바랍니다.
    ① 구조적 방법론 (또는 구조적 방법론 기반 자체 정의)
    ② 정보공학 방법론 (또는 정보공학 방법론 기반 자체 정의)
    ③ 객체지향 방법론 (또는 객체지향 방법론 기반 자체 정의)
    ④ CBD 개발 방법론 (또는 CBD 개발 방법론 기반 자체 정의)
    ⑤ Agile 개발 방법론 (또는 Agile 개발 방법론 기반 자체 정의)
    ⑥ 프로토타입 개발 방법론 (또는 프로토타입 개발 방법론 기반 자체 정의)
    ⑦ 잘 모르겠음 또는 별도 정의되지 않음

15. 다음 중 요구사항 정의 시 정의하는 정보를 모두 선택해 주시기 바랍니다.
    ① 고객이 제시한 명시적인 요구사항 정리
    ② 과거경험, 벤치마킹, 과거 문제점 등을 고려하여 요구사항 정의
    ③ 구현하여야 하는 모든 기능 리스트
    ④ 비-기능 요구사항
    ⑤ 제약 사항
    ⑥ 인터페이스 대상
    ⑦ 위험 분석 결과 도출 된 안전 기능
    ⑧ 해당하는 사항 없음

16. 다음 중 요구사항 분석 결과에 포함되는 것을 모두 선택해 주시기 바랍니다.
    ① 구현 할 모든 기능의 식별과 기능에 대한 설명
    ② 구현 할 모든 기능에 대한 상세 흐름 정의 (기본, 선택, 에러 핸들링 흐름 등)
    ③ 각 기능 구현 시 필요한 정보 또는 데이터 리스트
    ④ 각 기능 구현 시 Function으로 도출 가능한 대상
    ⑤ 구현 할 기능간의 연관 관계 파악
    ⑥ 기능 구성에 대한 모델링
    ⑦ 기능 동작에 대한 모델링
    ⑧ 필요 데이터에 대한 모델링
    ⑨ 해당하는 사항 없음

17. 소프트웨어 설계를 위해 수행하는 활동을 모두 선택해 주시기 바랍니다.
    ① 자체 개발 부분, 솔루션 도입 부분, 기존 시스템 재사용 영역 식별
    ② 아키텍처 설계
    ③ 상위 설계 및 데이터 설계 활동을 수행하고 문서화 함
    ④ 상세 설계 및 데이터 설계 활동을 수행하고 문서화 함
    ⑤ 내부 시스템 간 또는 외부 시스템과의 인터페이스에 대한 설계를 수행하고, 문서화 함
    ⑥ 위의 활동들을 수행하지 않음

18. 소프트웨어 설계 시 적용하고 있는 모델링 방법을 모두 선택해 주시기 바랍니다.
    ① 데이터 모델링
    ② 데이터 흐름 다이어그램
    ③ 제어 흐름 다이어그램
    ④ 유한 상태 머신 또는 상태 전이 다이어그램
    ⑤ 시간 패트리넷
    ⑥ 정형 기법
    ⑦ 결정/진리 테이블
    ⑧ 성능 모델링
    ⑨ 구조 다이어그램
    ⑩ 프로토타이핑/애니메이션
    ⑪ 순차 다이어그램

19. 철도 분야 소프트웨어 개발 시 사용하는 프로그래밍 언어를 모두 선택해 주시기 바랍니다.
    ① C or C++
    ② JAVA
    ③ ADA
    ④ MODULA-2
    ⑤ PASCAL
    ⑥ PL/M
    ⑦ Assembler
    ⑧ C#

20. 소프트웨어 개발 시 사용하는 코딩 가이드라인을 선택해 주시기 바랍니다.
    ① MISRA-C
    ② 자체 개발
    ③ 컨설팅 업체 제안
    ④ 사용하지 않음

21. 제품 개발 시 수행하는 테스트를 모두 선택해 주시기 바랍니다.

① 단위 테스트
② 통합 테스트
③ 시스템 테스트
④ 회귀 테스트
⑤ 별도 정의되지 않았음

22. 테스트 설계는 어떤 방식으로 이루어지고 있는지 선택해 주시기 바랍니다.

① 각 담당 개발자가 각자의 방식으로 테스트 시나리오 및 케이스를 설계함
② 각 담당 개발자가 정의 된 방식에 따라 테스트 시나리오 및 케이스를 설계 함
③ 테스트 전문 인력의 지원을 받아 테스트 시나리오 및 케이스를 설계함
④ 잘 모르겠음 또는 별도 정의되지 않음

23. 회사에서 제품 개발 시 사용하는 소프트웨어 개발 관련 도구가 있으면, 해당 도구의 이름과 구체적인 활용 영역을 적어주십시오 (Doors로 요구사항 관리, StarUML로 아키텍처 및 설계에 사용, MS Visio로 아키텍처 및 설계에 활용, PMD로 코딩 규칙 확인 등)

4. 품질 및 형상관리

24. 제품 개발 시 형상관리를 진행하고 있는 산출물을 모두 선택해 주시기 바랍니다.
    ① 각종 계획서
    ② 요구사항
    ③ 분석 단계 개발 산출물
    ④ 설계 단계 개발 산출물
    ⑤ 각종 모델링 파일
    ⑥ 소스코드
    ⑦ 테스트 시나리오 및 케이스
    ⑧ 각종 회의록
    ⑨ 각종 보고서
    ⑩ 각종 관리 대장
    ⑪ 개발 도구(컴파일러 포함)버전 및 설정정보

25. 변경 요청 시 기록 관리하는 정보를 모두 선택해 주시기 바랍니다.
    ① 변경 요청자, 변경 요청일, 변경 요청 내용
    ② 해당 변경 요청 건에 의하여 변경이 필요한 대상 산출물
    ③ 해당 변경 요청 건에 의하여 변경이 필요한 대상 소스코드
    ④ 변경에 소요되는 예상 공수
    ⑤ 변경 처리 일정 계획
    ⑥ 변경 승인 회의 결과
    ⑦ 변경 처리 결과
    ⑧ 변경 처리에 소요된 실제 공수
    ⑨ 변경 요청에 대해서 별도로 기록 관리하지 않음

26. 구현 단계에서 변경 발생 시 변경 범위를 파악하기 위해 확인하는 산출물을 모두 선택해 주시기 바랍니다.
    ① 소스 코드
    ② 분석 개발 산출물
    ③ 설계 개발 산출물
    ④ 기능 연관관계를 파악할 수 있는 특정 개발 산출물
    ⑤ 요구사항 추적표
    ⑥ 해당 개발자의 경험에 의한 판단

27. 품질 점검 시 검토되는 항목을 모두 선택해 주시기 바랍니다.
    ① 정의 된 프로세스 준수 여부
    ② 각 단계별 작성 산출물 내용의 적절성
    ③ 프로젝트 진행(진척, 투입공수 등)의 적절성
    ④ 프로젝트 이슈 및 위험 대응의 적절성
    ⑤ 잘 모르겠음

28. 회사에 제품 개발 시 품질 담당 전문 인력이나 조직이 있는지와 있다면 역할과 권한에 대해서 적어 주시기 바랍니다. (예, 별도의 품질관리 팀이 있으며, 제품의 테스트 수행과 평가 및 보고 기능을 수행한다.)

29. 제품 개발 시 사용 하는 산출물 양식이 있으면 모두 선택해 주시기 바랍니다.
    ① 프로젝트 관리 관련 양식
    ② 위험관리 관련 양식
    ③ 이슈관리 관련 양식
    ④ 형상관리 관련 양식
    ⑤ 품질보증 관련 양식
    ⑥ 요구사항관리(변경 및 추적성 관리) 관련 양식
    ⑦ 분석 단계 관련 양식
    ⑧ 설계 단계 관련 양식
    ⑨ 테스트 단계 관련 양식
    ⑩ 검토 관련 양식
    ⑪ 유지보수 관련 양식
    ⑫ 그 외 양식

30. 회사에서 품질 및 형상 관리 시 사용하는 도구가 있으면, 해당 도구의 이름과 구체적인 활용 영역을 적어주십시오 (예, SVN으로 소스코드 형상 관리, GIT으로 소스코드 및 산출물 버전 관리 등)

설문에 응해 주셔서 감사합니다. 소중히 활용하겠습니다. – 경기대학교 & ㈜에스피아이디 컨소시엄 –