# 2022년 SW안전 제품·서비스 실증 지원 시범사업

## [공유자료]

과제명: 상용차 자율주행 테스트베드 구축을 위한
상용차 자율주행 차량 및 SW 개발의 SW안전기술 적용

2023. 1.

주관기관: (주)스카이오토넷
참여기관: (주)네오피엠
자동차융합기술원

# 목 차

# Ⅰ. 사업개요

## 1. 사업 목적

□ 향후 상용차 군집주행/자율주행 기술을 확보하기 위한 테스트베드에 활용될 Level 3 자율주행 상용차 장치 및 SW 개발에 ISO26262 ASIL C 등급에서 요구하는 안전기술을 적용하여 SW안전을 확보하고자 함


[그림 1] 사업 목적 및 주요 사업 범위

※ SW안전 제품·서비스 실증 지원 시범사업의 목적

○ SW안전 중요분야에 SW안전기술을 적용한 수요맞춤형 실증 지원을 통해 SW안전 新시장 창출 및 관련 전문기업 육성 추진

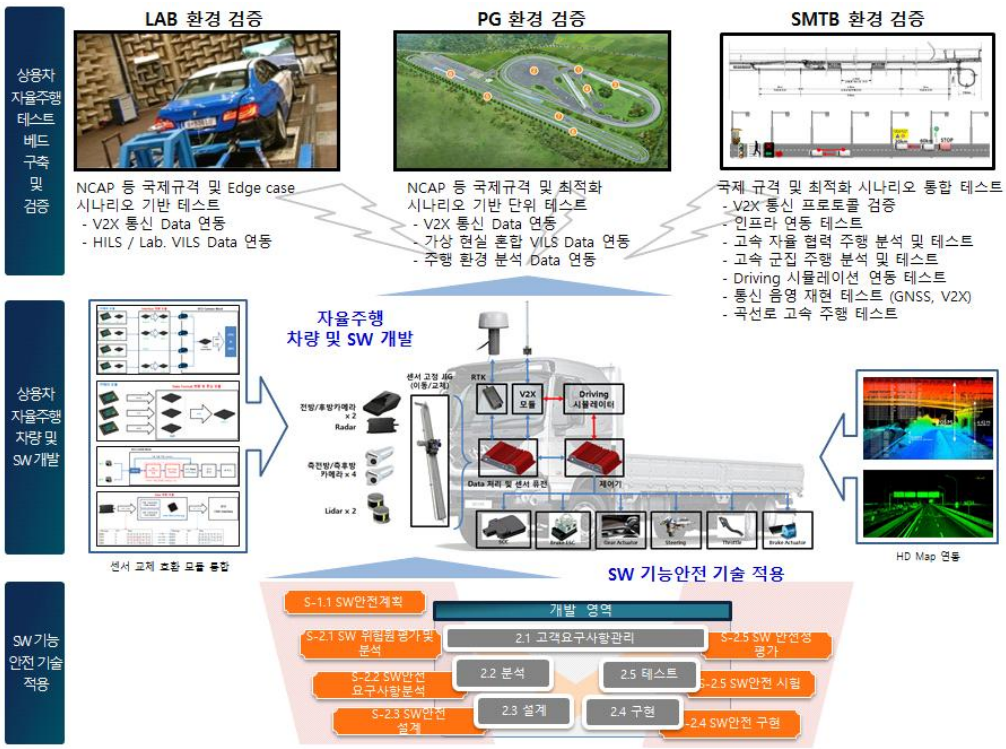
## 2. 목표 서비스 및 주요 사업 내용

### 가. 목표 서비스

□ 주관기관은 새만금지역 상용차 자율주행 테스트베드에서 운영될 상용차 자율주행(level 3 자율주행) 차량 및 SW를 개발하고 검증하여 수요기업에 제공

□ 수요기업은 “새만금지역 상용차 자율주행 테스트베드 구축사업”과 관련한 상용차 자율(군집)주행 인지·제어 핵심부품의 실차 적용성을 검증하고, 실차 기반 시스템 성능 분석·검증 플랫폼, 가상 현실 통합 평가 플랫폼과 연계 활용을 위하여 이를 적용하여, 국내 상용차 자율주행 기술 및 부품 개발 고도화에 기여함

□ 컨설팅 기업과 주관기관은 상용차 자율주행 SW와 센서 퓨전 SW의 개발에 기능안전 개발 프로세스와 기술을 적용하여 SW의 안전성을 제고함




[그림 2] 목표서비스 개념도

## 나. 주요 사업 내용

| 연도 | 분류 (구분) | 기술/서비스명 | 기술/서비스의 기능 정의 및 주요 내용 |
|---|---|---|---|
| 1차년도 | 상용차 자율주행(Lv3) 차량 및 SW 개발 | 상용차 자율주행(Lv3) 차량 제작 (기 보유기술) | o 자율주행 레벨 3을 지원하기 위한 차량 및 엑츄에이터 장비 구성<br>o 군집/자율협력 주행 개발을 위한 센서 이동 설치/추가 장착 가능 구조<br>o 자율주행차량 확보 및 개조: 5톤 이상 상용차(트럭) 확보 및 사용자 활용 편의성을 위한 HMI 최적화 |
| | | 자율주행차량 S/W 개발 | o ROS2 기반 자율주행(Lv3) 제어 시스템 개발<br>o 제어 모듈 독립화 및 Fail Safe / Take over 기능 SW 적용 |
| | | 자율주행 기술 개발을 위한 센서 평가 지원 시스템 | o 센서별 이동 및 추가 설치가 가능<br>o 센서별 DATA 변환 및 활용, 검증 지원 (자율주행/ADAS 등) |

| 연도 | 분류 (구분) | 기술/서비스명 | 기술/서비스의 기능 정의 및 주요 내용 |
|---|---|---|---|
| | | 개발 (기 보유기술) | |
| | | V2X 통신 개발 (기 보유기술) | o V2X 통신 (인프라/Lab/PG/SMTB/관제) 장비 연동 |
| | 상용차 자율주행 테스트베드 내 기술성 검증 | 상용차 자율주행 테스트베드 내 실증 테스트 | o 새만금지역 상용차 자율주행 테스트베드 구축을 위한 통합시험<br>o 상용차량 및 자율주행 SW 의 테스트베드 내 실증 테스트 |
| | ASIL B 등급 SW기능안전 프로세스 및 기술 적용 | SW 안전 요구정의 | o ISO26262 ASIL B 등급 요건 만족 프로세스 수립<br>- SW기능안전 프로세스의 Tailoring 적용<br>- SW기능안전 기술 적용 지침 보완<br>- 지침 기반의 교육 수행<br>o SW 위험원 분석 및 평가(HARA)<br>o FMEA 및 SW요구사항 분석 |
| | | SW 안전설계, 구현 및 시험 | o SW 안전 아키텍처 설계<br>o SW 안전 설계 및 구현<br>o 통합테스트 및 안전 시험<br>o 새만금 테스트베드 내 실차 테스트 및 안전 시험 |
| 2차년도 | ASIL C 등급 기능안전 프로세스 및 기술 적용 | ASIL C 등급 지원 SW 안전 구현 및 시험 | o 고속도로 자율주행 테스트를 위한 SW고도화에 따른 안전요구사항 및 안전메카니즘 설계<br>o ISO26262 ASIL C 등급 지원 SW 안전 설계 검증 및 SW 안전 구현<br>o 고도화 SW 코드 분석 및 안전 시험 |
| | 상용차 자율주행(Lv3) 차량 및 SW 고도화 | 자율주행차량 S/W 고도화 | o 고속도로 자율주행 테스트를 위한 SW 고도화<br>o 제어 모듈 독립화 및 Fail Safe / Take over 기능 SW 고도화<br>o 1차년도 기술성 검증결과와 안전분석 결과를 반영하여 SW 고도화 |
| | | V2X 통신 및 HMI 고도화 | o V2X 통신 (인프라/Lab/PG/SMTB/관제) 장비 연동 변경에 따른 지원<br>o 군집 자율주행 상태 및 센서 정보에 따른 HMI 개선 |
| | 자율(군집) 주행 검증 및 평가 활용 | 상용차 자율주행 테스트베드 활용 지원 | o 새만금지역 상용차 군집/자율협력 주행 테스트<br>o 테스트베드 기업 지원 및 환경 고도화 |

## 다. 당해년도 주요 사업 내용

### 1) ASIL C 등급 SW기능안전 프로세스 및 기술 적용

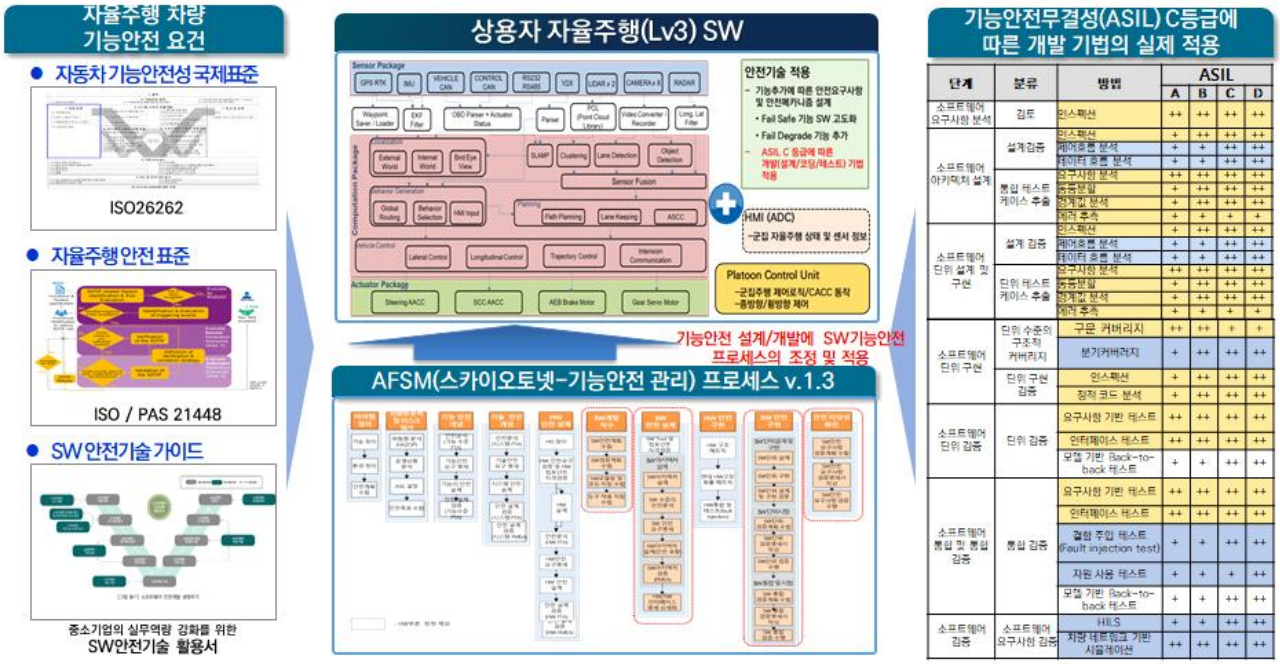
○ SW기능안전 프로세스 및 지침 보완 후에 안전기술을 개발에 적용

– 현재 주관기관이 보유하고 있는 기능안전 프로세스 v1.3를 조정하여 상용

차 자율주행(Lv3) SW 개발에 적용

– 기능안전 프로세스를 적용하여 신규 및 고도화 기능에 대한 기능안전 요구사항 및 안전메카니즘을 설계

– ASIL C 등급을 만족하는 설계/개발/시험 기법을 실제 개발 과정에 적용




[그림 3] 당해연도 SW 안전 기술 적용 개요 및 범위

### 가) ISO26262 ASIL C 등급 요건 만족 프로세스 및 지침 보완

○ 현재 주관기관이 보유하고 있는 기능안전 프로세스 v1.3을 ASIL C 등급에 맞게 보완하여 상용차 자율주행(Lv3) SW 개발에 적용

– 현재 수립된 기능안전 프로세스는 ISO26262의 ASIL C 등급을 만족하도록 수립되었으나, ISO26262 표준이 가지고 있는 한계로 자율주행 시스템 영역의 안전 표준을 구체적 지침 및 작업별 업무수행가이드에 구체적으로 제시하지 못하고 있음

- 상용차 자율주행차량(L3)의 S/W 개발은 기본적으로 ASIL D 등급을 만족하여야 하나, 테스트베드 내 Fail-safe 기능 제공 및 운전자 개입으로 ASIL C 등급인 것으로 정의됨(1차년도 HARA 결과)

– 현재 개발하고자 하는 상용차 자율주행(Lv3) SW는 ASIL C등급을 만족하여야 하나, 일부 ASIL C 등급에서 요구하는 기술의 적용은 현실적으로 1년내에 적용하기 어려워 1차년도에는 ASIL B 등급을 만족하고, 필수적으로 요구되는 기술 위주로 적용하였음

○ 안전무결성수준(ASIL) C 등급 만족을 위한 개발기법 지침 보완

- 현재 적용하지 못하고 있는 ISO26262 ASIL C 등급 요건을 만족하는 개발을 위한 다양한 개발 및 검증 활동 기법을 2차년도에 적용하기 위한 지침을 개발
- ASIL C등급에서 요구하는 개발기법을 적용하기 위한 업무수행가이드 보완
  - SW아키텍처 설계 업무수행가이드
  - SW단위시험 업무수행가이드
  - SW통합시험 업무수행가이드
  - HIL시험 업무수행가이드 등
- 각 작업에 대한 구체적 업무가이드를 수립
  - 각 작업별 업무수행가이드는 표준 관리/개발 프로세스의 각 작업별로 절차, 역할, 입력물/출력물, 기법, 도구, 핵심체크사항 등을 제공함
  - 각 산출물별로 산출물의 주요 내용(목차) 및 각 목차별 작성 지침 및 작성 사례 등을 기술하여 각 작업자가 쉽게 산출물을 작성하도록 산출물 템플리트를 제시
- ASIL C등급에서 요구하는 설계 및 구현 단계의 SW공학원칙을 만족하기 위한 지침 보완
  - SW 시험 지침
  - 시험케이스 설계 지침
  - 정적분석도구 활용 지침
  - 코딩가이드
  - 코드개선 지침 등
  - NIPA에서 공개한 『자동차 SW안전가이드』, 『SW안전가이드-공통분야』, 『중소기업의 실무역량 강화를 위한 SW안전기술 활용서』를 활용하여 수립(절차 및 가이드, 산출물 템플리트 정의)


[그림 4] 적용 안전기술 및 지침 보완 범위

○ 지침 기반의 교육 수행: 실무자들의 소프트웨어공학 기법에 대한 인식을 높이고 관련 지식 및 경험 수준을 향상시키기 위하여 프로젝트 진행 중 지속적인 교육 및 세미나, 토론 활동을 통하여 전문성을 배양하도록 함

– ISO26262를 지원하는 HARA 분석 및 안전분석 기법 교육

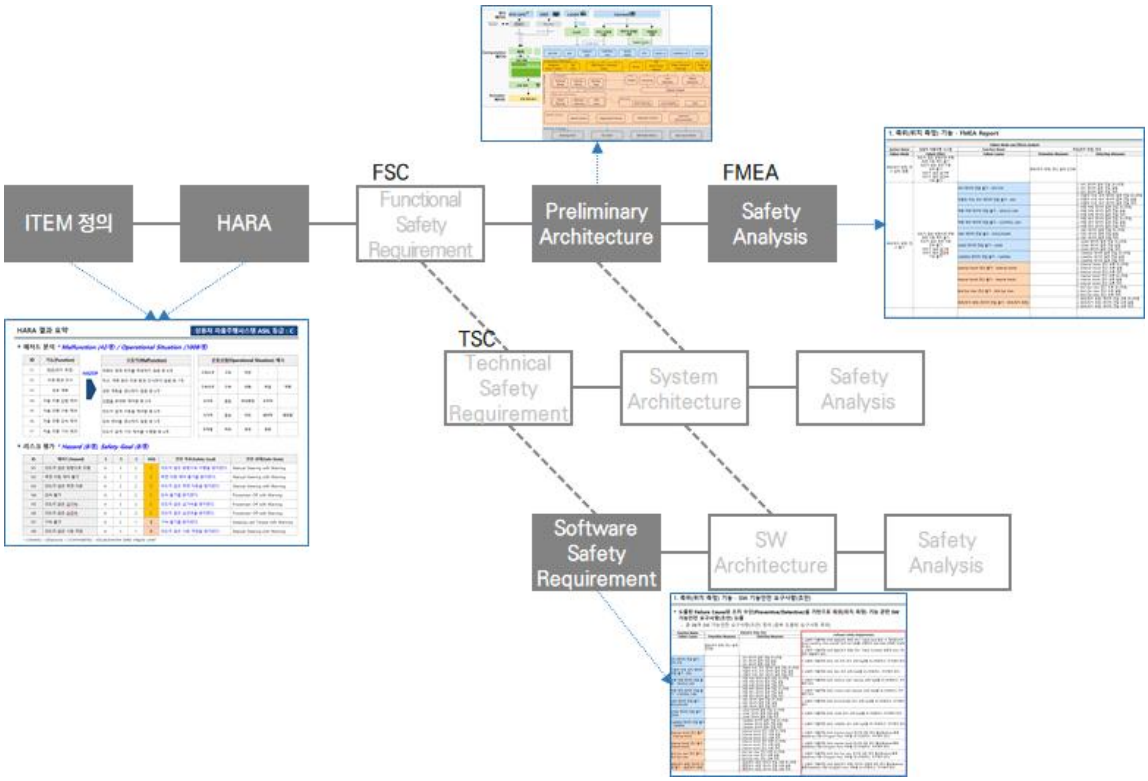
– 업무수행가이드 및 지침 기반의 교육(안전 설계, 정적분석, 테스트 교육)

### 나) SW 안전 요구 정의

○ 자동차 레벨의 HARA 수행 후, 작성된 아키텍처를 기반으로 FMEA 안전 분석을 수행하고, 이미 널리 알려진 SW 기능안전 메커니즘과 도출된 결함 원인을 맵핑하여 SW 기능안전 요구사항을 도출

– 1차년도에 수행한 HARA 및 FMEA를 기반으로 고속도로 자율주행 테스트를 위한 추가개발사항을 포함하여 HARA(위험원 분석 및 리스크 평가) 및 FMEA(고장모드 및 영향분석) 수행

– HARA 및 FMEA 수행범위, 안전요구사항 정의 방안은 1차년도 내용과 유사함




[그림 5] 안전요구사항 정의

### 다) SW 안전 설계 및 구현

#### ① SW 안전 아키텍처 설계

○ 고속도로 자율주행 테스트를 위하여 추가되는 기능(HD-map 기능개선 등)을 포함하는 자율주행의 전체 시스템이 안전하다고 주장하기 위해 존재해야 하는 안전기능을 설계

– 군집/자율주행을 위한 SW고도화에 따른 안전메카니즘 설계

• 1차년도에 자율주행면허시험 획득을 위하여 추가 설계/개발한 안전기능을 포함하여 전반적인 안전메카니즘 재설계 및 개발

- FS (페일 세이프 기능)와 FD (페일 디그레이드 기능) 일부 포함

○ SW안전 아키텍처 설계 전술과 안전 아키텍처 패턴을 참고하여 체계적으로 설계

– 각 요소에는 개별 시스템 설계 측면에 따라 오류 방지 및 / 또는 자동 오류 모드가 있도록 설계

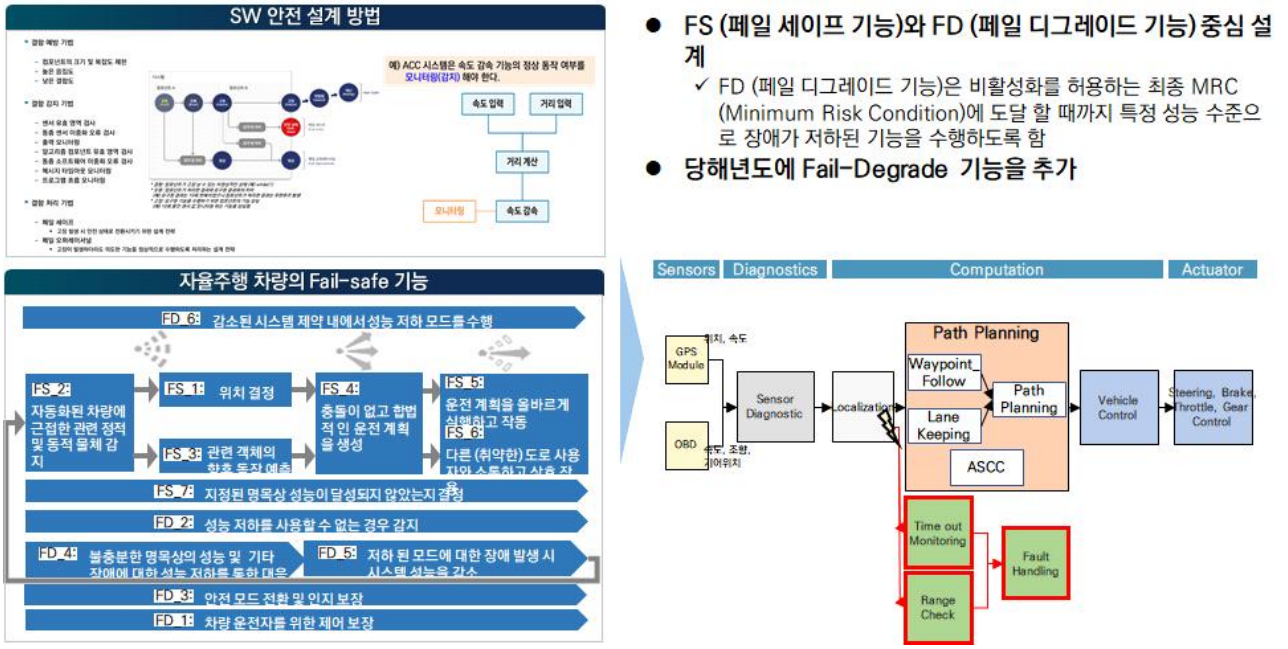
– 어떤 경우 든 개별 요소 또는 요소 조합의 현재 성능과 장애를 관찰하고 시스템 모니터에 보고하도록 설계

○ 결함을 사전에 예방하고, 감지 가능성을 높이고, 감지된 결함에 대해 적절한 처리를 수행할 수 있게 설계

– 결함 예방 : SW 컴포넌트에서 결함이 발생하지 않도록 사전에 아키텍처 설계 시 준수해야 하는 제약사항을 설정

– 결함 감지: SW 및 시스템 구성요소에서 발생할 수 있는 추가적인 구성요소를 아키텍처 설계 시에 반영

– 결함처리 : 결함이 감지되면 안전한 상태로 전환하도록 설계(Fail-Safe)하거나, 고장이 발생하도라도 정상적으로 수행하는 것을 목표로 설계(Fail-Operational)




[그림 6] SW 안전메카니즘 설계

### ② SW 안전 설계 및 구현

○ 결함 예방을 위하여 ISO26262에서는 ASIL(안전무결성 등급)별로 SW 설계, 구축, 시험 단계에서 요건을 만족하는 다양한 개발 및 검증 방법의 수행이 필요하며, 이를 당해연도에는 아래와 같이 적용하고자 함

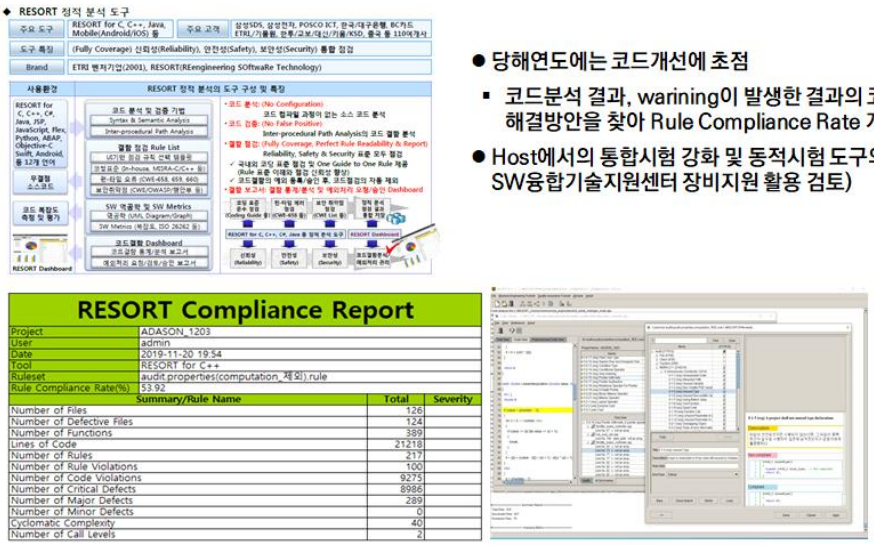
| 단계 | 분류 | 방법 | ASIL | | | | 현재 수준 | 주요 개선 필요사항 | 필요 지침 및 도구 | 적용 여부 및 적용 방안 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | A | B | C | D | | | | |
| 소프트웨어 단위 설계 및 구현 | 설계 검증 | 워크스루 | ++ | + | 0 | 0 | LA | 없음 | 산출물 검토(인스펙션) 지침 | 기적용 |
| | | 인스펙션 | + | ++ | ++ | ++ | LA | 인스펙션 결과서 지속적 유지 필요 | 산출물 검토(인스펙션) 지침 | 1차년도 적용 |
| | | **제어흐름 분석** | + | + | ++ | ++ | **NA** | **설계 검증을 위한 정적분석 도구 및 가이드 필요** | **아키텍처 설계지침(설계 검증 보완) 정적분석 도구 및 활용 지침** | **2차년도 적용** |
| | | **데이터 흐름 분석** | + | + | ++ | ++ | **NA** | **설계 검증을 위한 정적분석 도구 및 가이드 필요** | **아키텍처 설계지침(설계 검증 보완) 정적분석 도구 및 활용 지침** | **2차년도 적용** |
| | 단위 테스트 케이스 추출 | 요구사항 분석 | ++ | ++ | ++ | ++ | PA | 테스트케이스 설계를 위한 명확한 가이드 부족 | **테스트케이스 작성 지침(보완)** | 1차년도 적용 |
| | | 동등분할 | + | ++ | ++ | ++ | | | | |
| | | 경계값 분석 | + | ++ | ++ | ++ | | | | |
| | | 에러 추측 | + | + | + | + | | | | |
| | 단위 수준의 구조적 커버리지 | 구문 커버리지 | ++ | ++ | + | + | NA | **커버리지 분석 필요** | **단위테스트 지침 테스트케이스 작성 지침 테스트지원도구(커버리지 분석 도구)** | **2차년도 적용 (단위테스트 도구 적용 및 커버리지 분석 예정)** |
| | | 분기커버리지 | + | ++ | ++ | ++ | | | | |
| 소프트웨어 단위 구현 | 단위 구현 검증 | 인스펙션 | + | ++ | ++ | ++ | LA | 인스펙션 결과서 지속적 유지 필요 | 산출물 검토(인스펙션) 지침 | 1차년도 적용 |
| | | 정적 코드 분석 | + | ++ | ++ | ++ | NA | 정적 분석 필요 MISRA C/C++ 적용 필요 | **C, C++ 코딩 가이드(개선) 정적분석 도구 활용 지침** | **1차년도 적용 (일단 분석도구의 적용을 통한 분석 적용 후 일부 Rule 위반 사항은 코드 개선) 2차년도 확대 적용(코드개선)** |
| 소프트웨어 단위 검증 | 단위 검증 | 요구사항 기반 테스트 | ++ | ++ | ++ | ++ | PA | 단위테스트의 결과를 검증하는 기록 부재 | 단위테스트 지침 | 1차년도 일부 적용 (테스트 지침 교육 강화) |
| | | 인터페이스 테스트 | ++ | ++ | ++ | ++ | | | | |
| | | 모델 기반 Back-to-back 테스트 | + | + | ++ | ++ | NA | | | 미적용 (Model 기반 개발 적용하지 못하고 있음) |

[그림 7] SW 단위 설계/구현 단계의 SW안전기술 적용 방안

### ③ 코드 분석

○ ISO26262 요건 만족 및 사내 주요 문제점의 해결을 위한 단위/통합시험의 안전성 보장을 위한 정적분석 도구 활용 및 코드 개선

- 국제 표준 코딩룰(MISRA C/C++)을 적용한 잠재적 결함 검출 및 소프트웨어 품질 측정을 위한 Metrics을 지원하는 정적분석 도구의 활용
- 당해연도에는 코드개선에 초점
  - 코드분석 결과, warining이 발생한 결과의 코드 개선 및 그 해결방안을 찾아 Rule Compliance Rate 개선
- Host에서의 통합시험 강화 및 동적시험 도구의 도입 검토(일부 SW융합기술지원센터 장비지원 활용 검토)




**RESORT Compliance Report**

| | |
|---|---|
| Project | ADASON_1203 |
| User | admin |
| Date | 2019-11-20 19:54 |
| Tool | RESORT for C++ |
| Ruleset | audit.properties(computation_제외).rule |
| Rule Compliance Rate(%) | 53.92 |

| Summary/Rule Name | Total | Severity |
|---|---|---|
| Number of Files | 126 | |
| Number of Defective Files | 124 | |
| Number of Functions | 389 | |
| Lines of Code | 21218 | |
| Number of Rules | 217 | |
| Number of Rule Violations | 100 | |
| Number of Code Violations | 9275 | |
| Number of Critical Defects | 8986 | |
| Number of Major Defects | 289 | |
| Number of Minor Defects | 0 | |
| Cyclomatic Complexity | 40 | |
| Number of Call Levels | 2 | |



[그림 8] RESORT 도구를 활용한 코드정적분석 및 코드개선

### 라) 통합테스트 및 안전 검증

○ 개발된 시스템에 대해서 통합적인 테스트와 성능테스트를 진행

– 구현된 시스템 및 연계 시스템 간 유기적인 통합테스트 진행

○ SW 안전 시험을 위하여 다음과 같은 SW 안전시험 기법을 적용

– HILS 시험과 차량네트워크 시험이 시행되고 있으나, ISO26262의 요건을 만족함을 보장하지 못하는 수준이어서, 테스트를 위한 환경 구성 및 절차, 템플리트 정의 후 엄격한 테스트 진행 예정

〈 SW 통합 및 통합시험 단계의 SW안전기술 적용 방안 〉

| 단계 | 분류 | 방법 | ASIL | | | | 현재 수준 | 주요 개선 필요사항 | 필요 지침 및 도구 | 적용 여부 및 적용 방안 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | A | B | C | D | | | | |
| 소프트웨어 통합 및 통합 검증 | 통합 검증 | 요구사항 기반 테스트 | ++ | ++ | ++ | ++ | PA | 단위테스트의 결과를 검증하는 기록 부재 | 단위테스트 지침 | 1차년도 일부 적용 (테스트 지침 교육 강화) |
| | | 인터페이스 테스트 | ++ | ++ | ++ | ++ | | | | |
| | | 결함 주입 테스트 (Fault injection test) | + | + | ++ | ++ | NA | | 테스트 지침 | 2차년도 적용 (단계적으로 적용) |
| | | 자원 사용 테스트 (Resource usage test) | + | + | + | ++ | PA | 메모리 누수, 성능 관련 테스트 도구 활용 필요 | 테스트 지침 동적분석도구 | 2차년도 적용 (동적분석도구의 시범 적용) |
| | | 모델 기반 Back-to-back 테스트 | + | + | ++ | ++ | NA | | | 미적용 (Model 기반 개발 적용하지 못하고 있음) |

〈 SW 검증 단계의 SW안전기술 적용 방안 〉

| 단계 | 분류 | 방법 | ASIL | | | | 현재 수준 | 주요 개선 필요사항 | 필요 지침 및 도구 | 적용 여부 및 적용 방안 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | A | B | C | D | | | | |
| 시스템 검증 | 소프트웨어 요구사항 확인 | HILS | + | + | ++ | ++ | PA | 일부 target에서의 시험 적용하나 명확한 지침 필요 | HILS 테스트 지침 | SW 컴포넌트에 대하여 적용 |
| | | 차량 네트워크 시험 | ++ | ++ | ++ | ++ | NA | 차량네트워크 테스트를 위한 환경 구성 및 절차, 템플리트 정의 필요 | 차량네트워크 테스트 지침 | 2차년도 적용 (센서 퓨전, 위치 확인, 인지 SW 컴포넌트에 대하여 적용) |
| | | 세만금 테스트베드 시험 | ++ | ++ | ++ | ++ | PA | 시나리오 및 성능요건 정의 | 시나리오 정의 | 2차년도 적용 |

[그림 9] SW 통합 안전시험 방안

## 2) 상용차 자율주행(Lv3) SW 고도화

### 가) 자율주행차량 S/W 개발

#### ① 자율주행 기반 데이터 처리 및 분석 시스템 개발

○ ROS2 기반 자율주행(Lv3) 연산처리장치(기보유기술) 기반으로, 자율주행 제어 시스템 을 개발하고, Fail Safe 기능 등의 기능안전 SW 추가 적용

– 고속도로 자율주행 테스트를 위한 SW 고도화 : HD-map 기능 개선 및 Autoware Platform 도입을 통한 SW 고도화

• Autoware 플랫폼 도입 및 SW 고도화

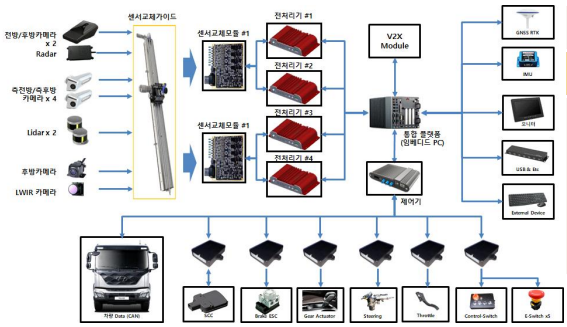
- ROS 기반의 자율주행 플랫폼으로 도입 및 재개발

- 센서 데이터를 통해 시뮬레이션이 가능

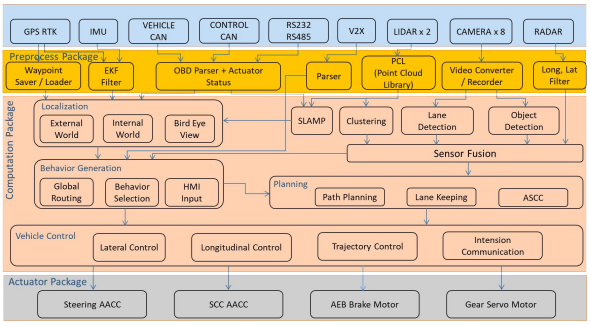
- 개별의 모듈화로 되어 있어 구조적으로 이점이 있음
- HD map 기능을 고도화

– 안전기술 적용: Fail Safe 기능 SW 고도화, Fail Degrade 기능 추가
– 1차년도 기술성 검증결과와 안전분석 결과를 반영하여 SW 고도화 : ASIL C 등급에 따른 개발(설계/코딩/테스트) 기법 적용

*** 자세한 자율주행 차량 SW 기능 및 구조는 기업 비밀 보호를 위하여 생략**




[그림 10] 상용차 자율주행 차량 시스템 구성




[그림 11] 상용차 자율주행 시스템 SW Architecture

## ② 자율주행 기반 상용차량 제어 시스템 개발

○ 자율주행 제어 소프트웨어는 차량의 기어포지션, 차량의 가속/감속 제어와 조향(Steering wheel) 제어로 구성됨

○ 센서별 객체 탐지 및 위치 측정을 통한 차량제어

– GPS를 통한 위치 좌표 데이터 및 주행 경로 데이터, 전/후/측방 카메라를 통한 2차원(2D) 객체 데이터 및 vision 데이터, 전/후방 LiDAR를 통한 3차원(3D) 객체 데이터 및 vision 데이터, OBD의 차량 정보 등을 활용하여 Path planning을 도출할 수 있다. 도출된 Path planning 값을 활용하여 가/감속 제어, 제동 제어, 조향 제어 등에 활용하여 진행

○ ASCC(Advanced Smart Cruise Control) 시스템 알고리즘 적용

– 운전자가 조작하는 가속 및 브레이크 페달, 조향을 보조해 운전자의 피로도를 낮춰주는 편의 기능

○ HD Map을 활용한 자율주행 기능

– 정밀맵을 활용 자율주행 가능하고 GPS가 아닌 Lidar를 활용하여 기 구축된 정밀맵(vactor map)상에 매칭을 통한 상대위치 인지 가능

– 주행 차량별 이동 객체와 연석/가드레일 등 정적 객체 맵핑 가능

○ 주행 환경 인식 후 주행경로에 의한 자율주행 제어 기능

*** 자세한 자율주행 차량의 제어 시스템 기능 및 구조는 기업 비밀 보호를 위하여 생략**

### 나) V2X 통신 HMI 고도화

○ V2X(Vehicle to Everything) 시스템은 실시간 정보 교환을 목적으로 진행하며, 차량(V2V), 인프라(V2I), Cloud(V2C) 등 모든 통신을 포함

- 각각의 센서들의 융합을 이용하여 도출하는 정보를 바탕으로 IEEE 802.11p의 WAVE 통신을 이용하여 시스템을 구현

O V2X 통신 (인프라/Lab/PG/SMTB/관제) 장비 연동 변경에 따른 지원

O 군집 자율주행 상태 및 센서 정보에 따른 HMI 개선

○ 또한, V2X 시스템을 이용하여 자율주행 제어의 검증을 진행

## 3) 상용차 자율주행 테스트베드 내 기술성 검증

### 가) 새만금지역 상용차 자율주행 테스트베드 구축을 위한 통합 시험

○ 실시간 통합관제평가시스템 구축을 위한 시스템 통합 검증

### 나) 군집자율주행 SW 의 테스트베드 내 실증 테스트

○ 실도로 환경에서 고속으로 합류·분류, 이탈, 가속, 장애물 인지·회피 등 다양한 연속성 시나리오 제공하여 시험·실증

○ 군집 자율주행 차량이 실도로 환경에서 운영되지를 검증할 시험시나리오를 개발하고 시험·실증

# Ⅱ. 적용 SW 공학 기술

## 1. SW공학기술 도입 필요성 및 적용 영역 선정 배경

□ 자율주행 기술이 미래 상용차 산업의 핵심으로 떠오르고 있음

○ 정부는 자율주행기술개발혁신사업을 통하여 2027년 융합형 Lv.4+ 자율주행 상용화 기반 완성을 목표로 2021년부터 2027년까지 총 1조 974억원 투입 예정임

○ 국교통연구원은 2040년까지 중대형 상용차의 약 56%가 완전자율주행차로 대체될 것으로 전망함

○ 자율주행 상용차가  지닌 대형사고 감소, 인건비 및 유류비 절감, 운송 정시성 향상 등의 효과 때문임

□ 자율주행 차량 및 자동차 전장부품에 **기능안전 기술을 적용하는 것이 필수적임**

○ 자동차에 탑재되는 전기전자시스템의 오류로 인한 사고방지를 위해 개발 과정에서 기능의 안전성을 확보를 목적으로 2011년 11월 국제표준화기구(ISO)에서 자동차기능안전 국제 표준이 ISO26262(자동차 기능안전성 국제 표준)을 제정함

– ISO 26262는 프로세스 모델과 함께 요구되는 활동, 유무형의 증거물, 그리고 개발과 생산에 사용되는 기법을 정의함

○ 자동차 안전 요구사항을 지정하는 지표로는 ASIL등급을 사용하고 차량 사고 발생 시, OEM이 차량의 전체 생명주기에 걸쳐 최신의 안전기술을 적용했음을 입증해야 최소한의 면책 요건을 부합할 수 있음

– Toyota는 급발진 사고 시에 이를 입증하지 못하여 리콜비용(24억 달러) 외에 16억 달러를 배상함

○ 안전지능주행(Safe-Smart driving)은 ADAS 기술에서 운전자의 편의성과 사고방지를 강조한 개념으로 안전주행은 사고 가능성의 감지 기술, 사고 방지를 위한 경고 또는 주행제어가 필요한 기술로서 외부 상황 인식, 판단 및 전장시스템에 의한 주행제어 기술이 복합적으로 요구되는 기술임

– 안전주행에 있어서 전장시스템의 동작고장은 필요한 순간에 정상동작을 하지 못하게 되므로 심각한 기능결함으로 이어질 수 있으므로 최근의 안전주행 시스템은 사고방지를 위한 경고 시스템을 구현하고 시스템 오동작 방지를 위하여 SW안전기술을 도입하고 있음

□ 당사는 자동차 안전운전 단말기와  자율주행 플랫폼, 차량의 실시간 관리를 전문으

로 하는 회사로 차량의 안전을 위한 기능안전 기술을 모든 제품 및 서비스에 적용하는 것을 목표로 하며, **특히 Level 3 이상의 자율주행 상용차 개발을 위해서는 기능안전 기술의 적용이 필수적임**

○ 당사는 미국 CES에서 2회의 Innovation Award 수상과 다수의 대규모 국책연구개발사업의 주관기관 등으로 전문기술을 보유하고 있으나, 시스템 개발을 위한 기법 및 도구 활용, 기능안전성 관리(Functional Safety Management) 능력은 ISO26262 인증을 획득하기에는 여전히 부족한 실정임

○ 수년간에 걸쳐(2017년부터 ISO9001, CMMI Level 3 인증 추진) 표준 프로세스를 개선하고, 일부 안전관리 프로세스를 수립/적용하였으며, 위험원 분석 및 리스크 평가, 안전 분석(FMEA, HAZOP), 안전요구사항 명세 등 개념 및 시스템 개발 단계의 안전관리 프로세스를 일부 적용함

– 자율주행 Level 3 이상의 상용차 자율주행 플랫폼 개발에 필수적인 **ISO26262 ASIL C 등급 이상에서 요구하는 SW공학 원칙 및 기법의 적용이 필수적임**

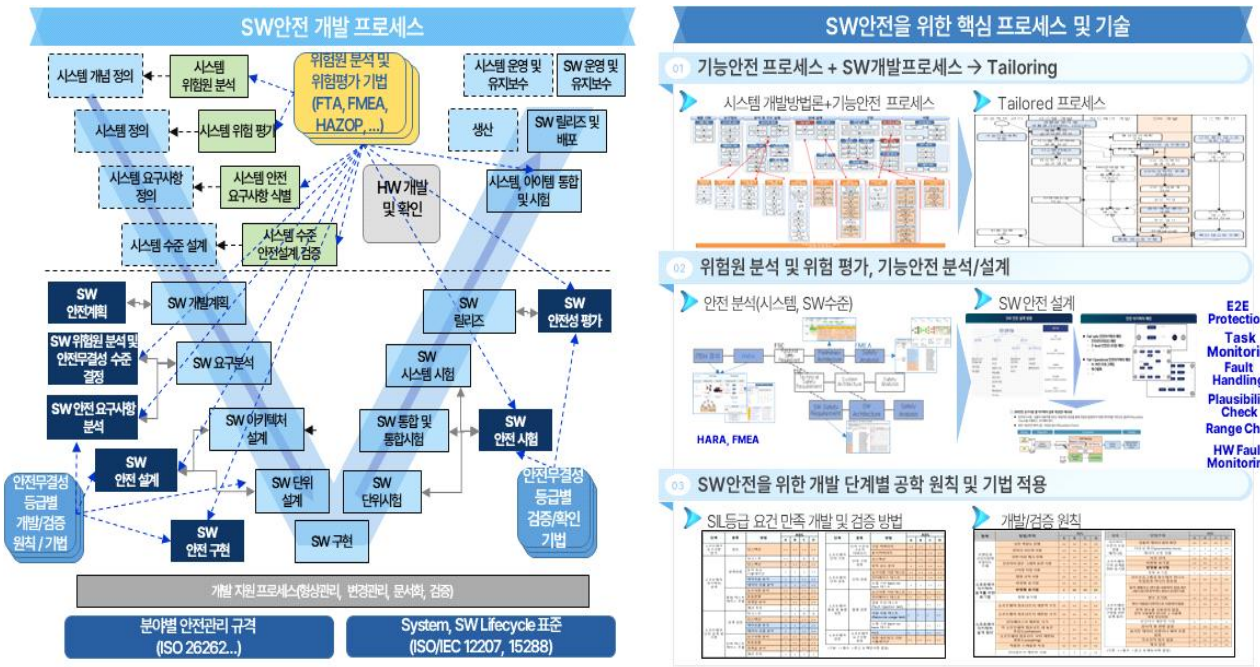
– SW 설계, 구축, 시험 단계에서 일부 기법 및 가이드가 수립되어 있으나, ISO26262에서 요구하는 개발 기법의 적용 및 도구 도입, SW 개발의 안전관리 후반부의 적용이 절실한 상황임

○ 최근 투자사 및 관련 공급사 등에서 투자 및 제품 공급을 위한 제품 검증, 개발관리 프로세스 능력 및 안전관리 프로세스 적용에 대한 실사 요구가 급증하고 있어 이에 대비하여 최소한 ASIL B 등급의 기능안전관리 체계를 구축해야 할 필요가 있음

## 2. SW공학기술 도입 및 적용 영역과 과정

□ SW로 인해 사고가 발생하지 않도록 **SW품질수준을 확보하는 것(Safety Integrity)** 외에 발생 가능한 사고를 **SW로 구현된 안전기능으로 감소/예방하기 위한 조치(Safety Function)**가 필요하며, 이를 위한 **기능안전 프로세스와 SW공학기술의 적용 필요**

○ SW 기능안전을 확보하기 위해서는 잠재적으로 위험한 요건을 분석하여 위험 사건의 방지를 위한 안전메커니즘의 작동 또는 위험 사건의 영향 완화를 위한 조치 제공 등의 **안전기능이 개발**되어야 하며, 당연히 SW 품질수준을 확보하기 위하여 **엄격한 SW공학 기법의 적용**을 요구함




[그림 12] Safety Integrity 보장 및 Safety Function 개발 필요성

□ 당사는 그간의 노력으로 표준 프로세스와 안전관리 프로세스를 수립하고, ISO9001 인증을 획득(2018년 3월), CMMI Level 3 인증(2018년 11월) 등의 성과를 거두었으나, ISO26262 ASIL C 등급을 만족하는 설계, 구축, 시험 단계의 구체적 기법 및 도구의 적용이 시급한 상황임

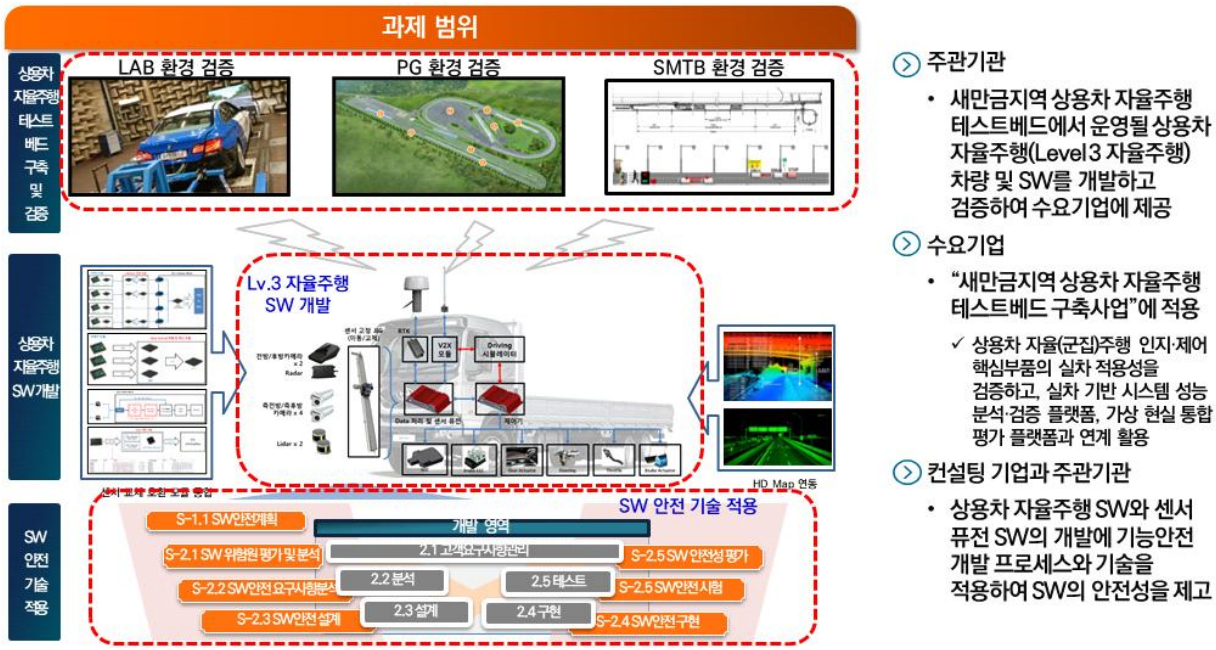
○ 현재 개발.공급 중인 상용차 자율주행 플랫폼과 차선유지시스템(LKAS, Lane keeping Assist System), 자동긴급제동장치(AEB : Autonomous Emergency Braking)등의 제품이 ISO26262 기능안전등급(ASIL) B~C 등급을 요구하고 있으나

○ ISO 26262를 충족시키기 위해서는 상당한 기술력과 자본이 필요하여 현재 CMMI(Capability Maturity Model Integrated) Level 3 인증을 우선 획득하였음

※ 일반적으로 ISO26262의 ASIL C 이상의 안전요구사항을 충족하기 위해서는 기본적으로 국제적으로 활용되는 개발능력 평가모형인 CMMI(Capability Maturity Model Integrated) Level 3 이상의 능력 성숙도를 기본적으로 보유해야 함

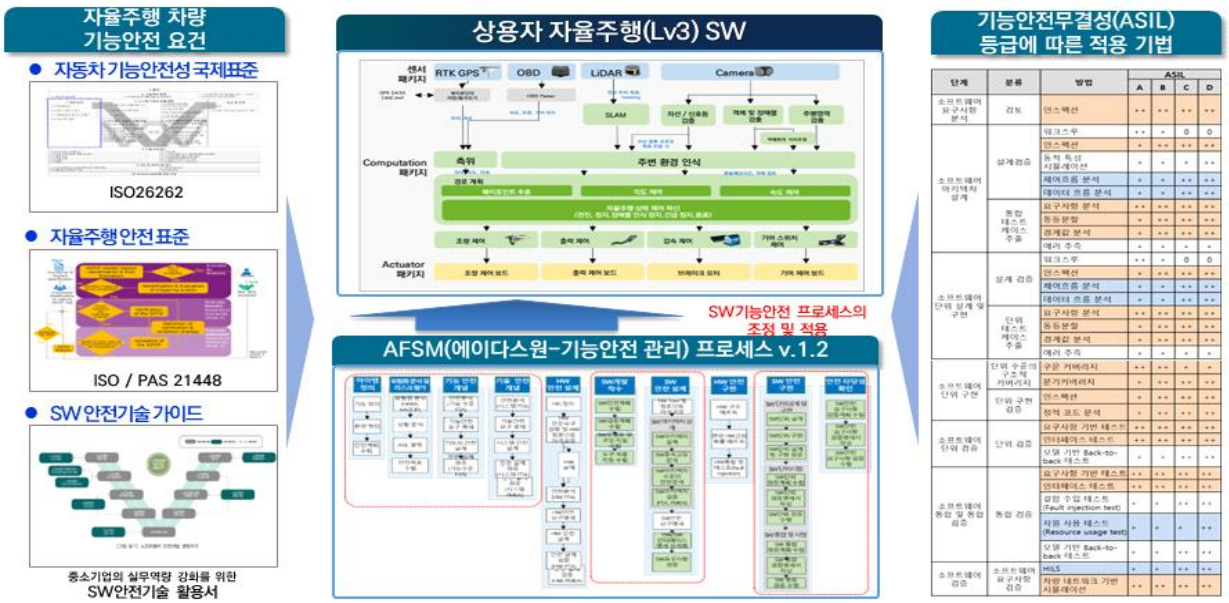
□ **“2021년도 SW안전 제품·서비스 실증 지원 시범사업”**으로 새만금지역 상용차 자율주행 테스트베드 내 상용차 자율주행(Lv3) SW 개발에 SW안전기술을 적용하고 실증 환경에서 검증하는 것을 범위로 하는 사업을 수행함

○ “상용차 자율주행 테스트베드 구축을 위한 상용차 자율주행 차량 및 SW 개발의 SW안전기술 적용” 사업을 통하여 SW안전기술을 적용하며, 그 범위는 다음과 같음




[그림 13] 상용차 자율주행 차량 및 SW 개발의 SW안전기술 적용

○ SW공학 기술의 적용 범위는 ASIL B 등급(일부 C등급 요건 포함) 요건을 만족하는 SW기능안전 프로세스 및 기술을 적용하고자 하였으며, 현재 주관기관이 보유하고 있는 기능안전 프로세스 v1.2를 tailoring하여 상용차 자율주행 SW 개발에 적용하며, 특히, 최근에 발표된 ISO/PAS 21448 자동차 주행 기능안전 표준을 일부 반영하여 안전 분석 및 검증에 활용할 수 있도록 기법을 적용하고자 하였습니다.




[그림 14] 기능안전 프로세스 및 기술 적용

□ 본 사업에서 적용하는 SW공학 기술은 크게 다음의 4가지로 요약될 수 있음

○ 전 생명주기를 지원하는 활동, 도구 및 기법 등의 SW공학기술 적용

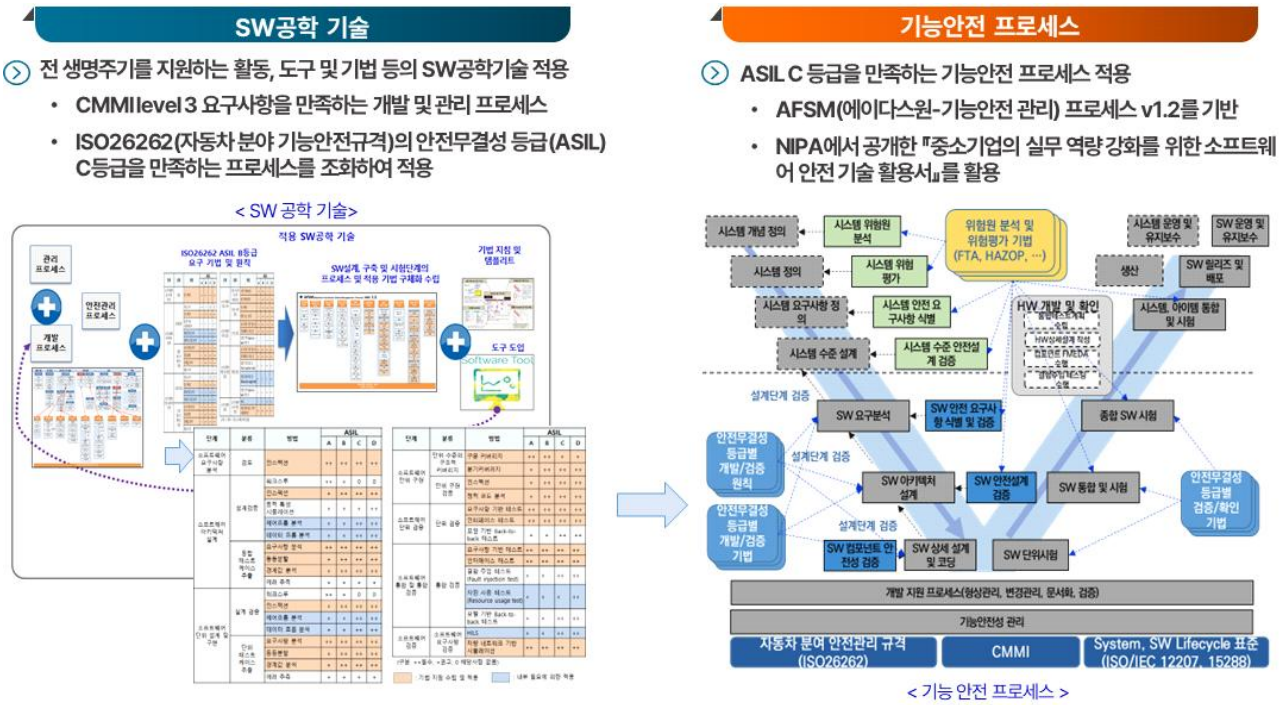
– CMMI level 3 요구사항을 만족하는 개발 및 관리 프로세스

– ISO26262(자동차 분야 기능안전규격)의 기능안전등급(ASIL) C등급을 만족하는 프로세스를 조화하여 적용

○ ASIL C 등급을 만족하는 기능안전 프로세스 적용

– AFSM(에이다스원-기능안전 관리) 프로세스 v1.2를 기반

– NIPA에서 공개한 『중소기업의 실무 역량 강화를 위한 소프트웨어 안전 기술 활용서』를 활용




[그림 15] 적용기술: SW공학기술과 기능안전프로세스

○ 안전 분석 기법을 적용

– 위험원분석 및 위험평가 (HARA:Hazard Analysis and Risk Assessment)

- 품목의 오작동이 유발할 수 있는 위험을 식별하고 분류
- 위험한 사건의 예방 또는 완화와 관련된 안전 목표의 공식화를 위한 위험 회피

– 고장영향분석(FMEA)

- 시스템의 발생 가능한 고장 모드(Failure Modes)를 정의하고, 영향(Effect)과 원인(Cause)을 분석하여, 해결 또는 예방책을 식별하고 안전메카니즘을 설계

○ 기능안전을 구현하는데 기본적을 품질을 보장하는 SW 공학기술(구현 및 시

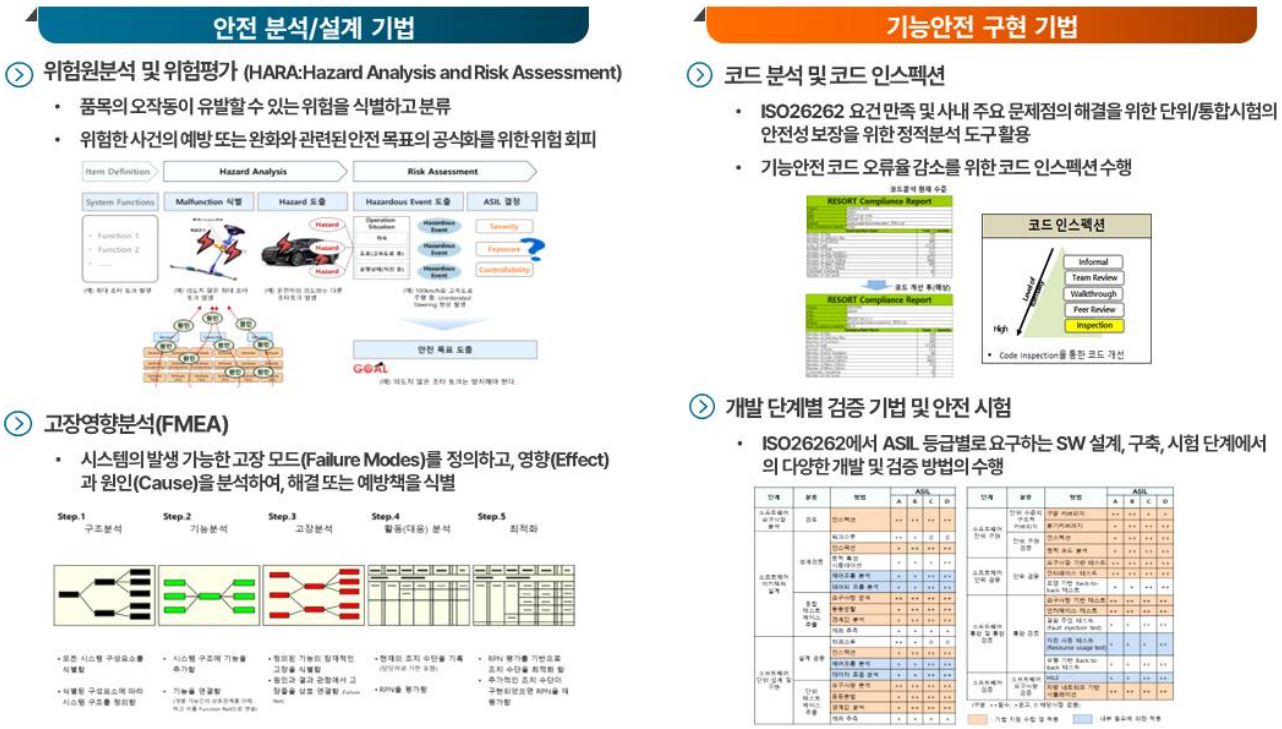
험 기술)을 적용

– 코드 분석 및 코드 인스펙션

- ISO26262 요건 만족 및 사내 주요 문제점의 해결을 위한 단위/통합시험의 안전성 보장을 위한 정적분석 도구 활용
- 기능안전 코드 오류율 감소를 위한 코드 인스펙션 수행

– 개발 단계별 검증 기법 및 안전 시험

- ISO26262에서 ASIL 등급별로 요구하는 SW 설계, 구축, 시험 단계에서의 다양한 개발 및 검증 방법의 수행




[그림 16] 적용 기술: 안전분석/설계 기법 및 기능안전 구현 기법

# III. 적용 영역별 주요 추진 내용

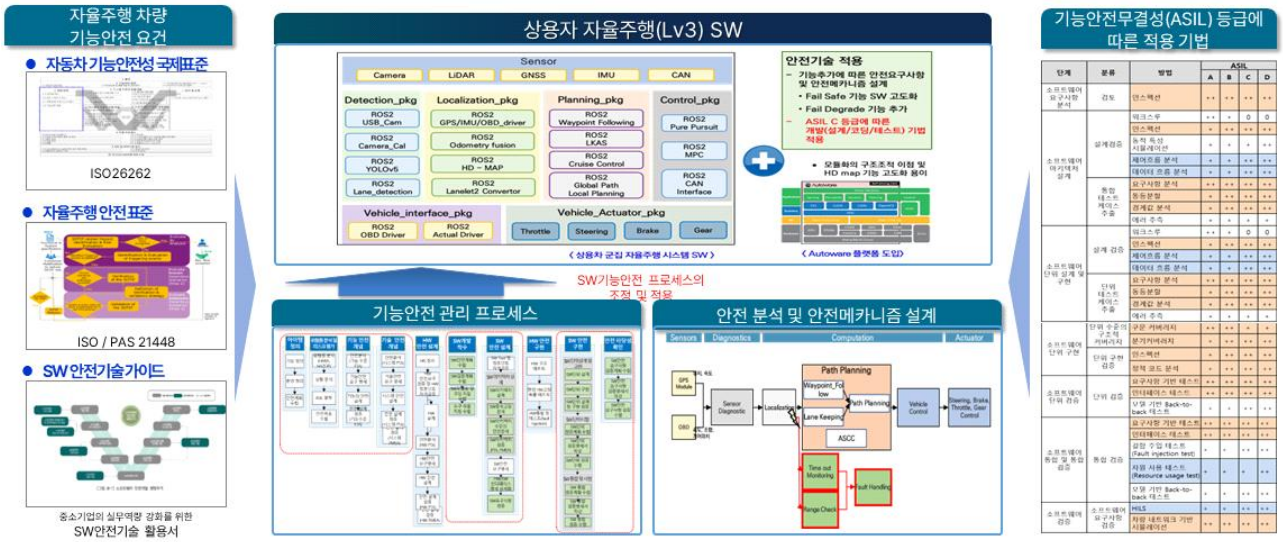
## 1. ASIL C 등급 SW기능안전 프로세스 및 기술 적용

□ SW기능안전 프로세스 및 지침 보완 후에 안전기술을 개발에 적용

○ 현재 주관기관이 보유하고 있는 기능안전 프로세스 v1.3를 조정하여 상용차 자율주행(Lv3) SW 개발에 적용

○ 기능안전 프로세스를 적용하여 신규 및 고도화 기능에 대한 기능안전 요구사항 및 안전메카니즘을 설계

○ ASIL C 등급을 만족하는 설계/개발/시험 기법을 실제 개발 과정에 적용




[그림 17] 당해연도 SW 안전 기술 적용 범위

### 가. ISO26262 ASIL C 등급 요건 만족 프로세스 및 지침 보완

□ ISO26262 요건 만족 프로세스 수립 및 단계적 적용

○ SW공학기술은 (주)에이다스원이 2017년 한양정보통신으로부터 분사한 이후부터 전사의 모든 프로젝트에 적용함을 원칙으로 하고 있음.

– 2021년 7월에 (주)에이다스원은 (주)스카이오토넷과 합병함

○ 2018년부터 개발방법론 및 도구의 활용을 위주로 SW공학기술을 적용하였고, CMMI level 3 요구사항을 만족하는 개발 및 관리 프로세스에 더하여 2019년부터 본격적으로 **ISO26262(자동차 분야 기능안전규격)의 기능안전등급(ASIL) C등급을 만족하는 프로세스를 구체화 하고, 요구되는 개발 방법(기법)과 개발 원칙을 수립하여 단계적으로 적용**하고 있음

– 지속적으로 적용하고자 하는 SW 공학기술은 구축, 테스트, 도구 및 기법임

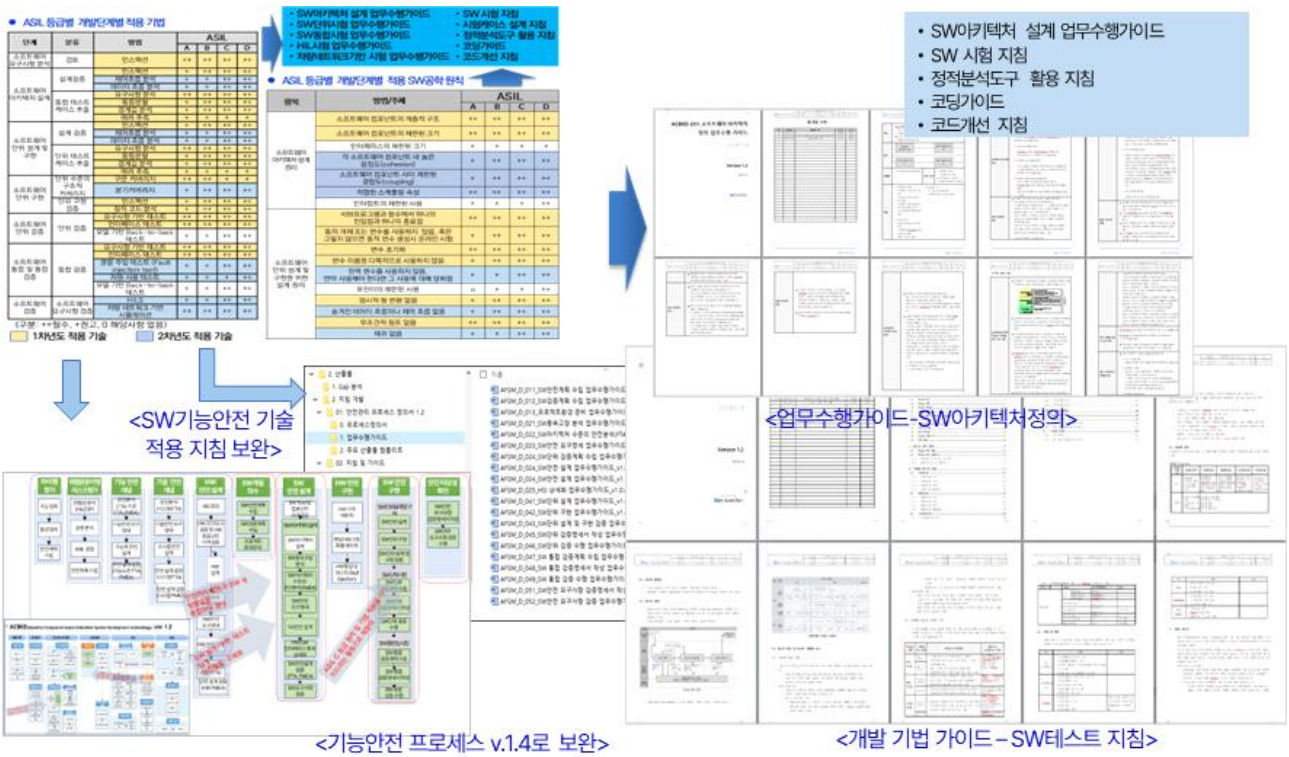
□ SW 기능 안전 프로세스 수립

○ 주관기관이 보유한 기능 안전 프로세스를 자율주행 시스템 영역의 안전 표준 및 프로세스를 반영하여 보완 수립

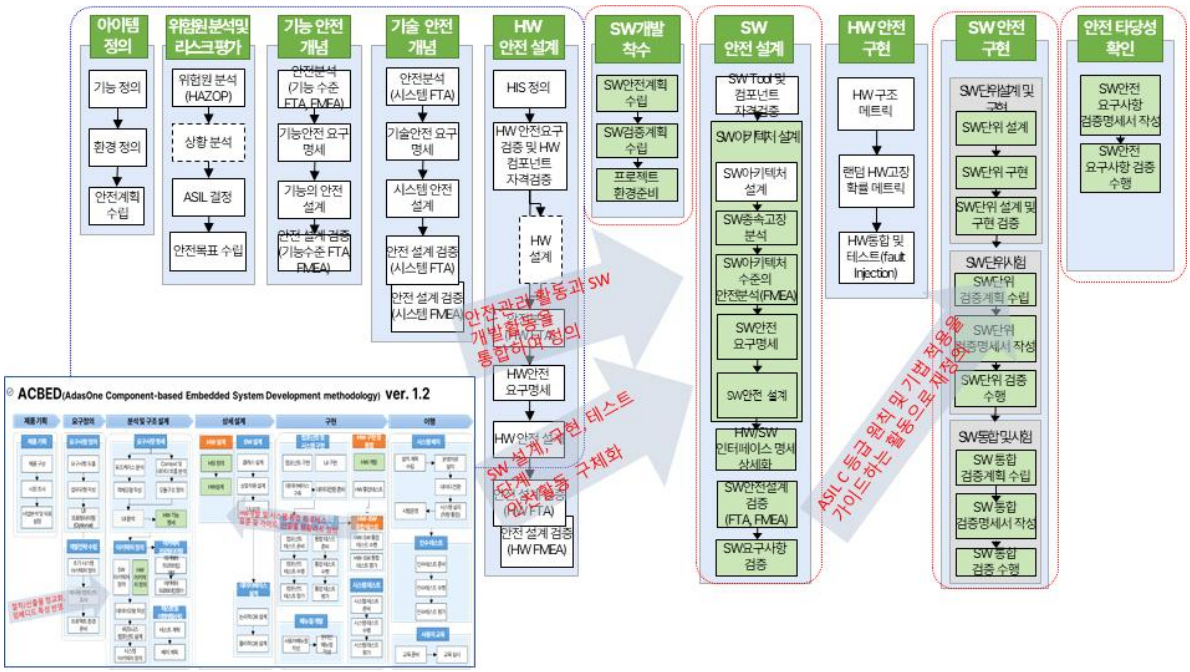
○ 기능안전 관리 프로세스의 각 활동에 대한 흐름과 요약(작업 개요 및 산출물 등)을 정리한 프로세스 정의서를 작성/보완

○ 기 수립된 개발방법론 및 기능안전관리 프로세스(AFSM: ADASOne Functional Safety Management Process) ver. 1.3)을 기반으로 SW 설계 단계 이후의 기능안전 프로세스를 구체화 보완

○ ISO26262 ASIL C 등급 요건을 만족하는 개발을 위한 다양한 개발 및 검증 활동의 수행이 필요하며, SW의 설계 단계에서부터 시험단계까지 이를 구체화 함




[그림 18] 프로세스 및 지침 보완 개요



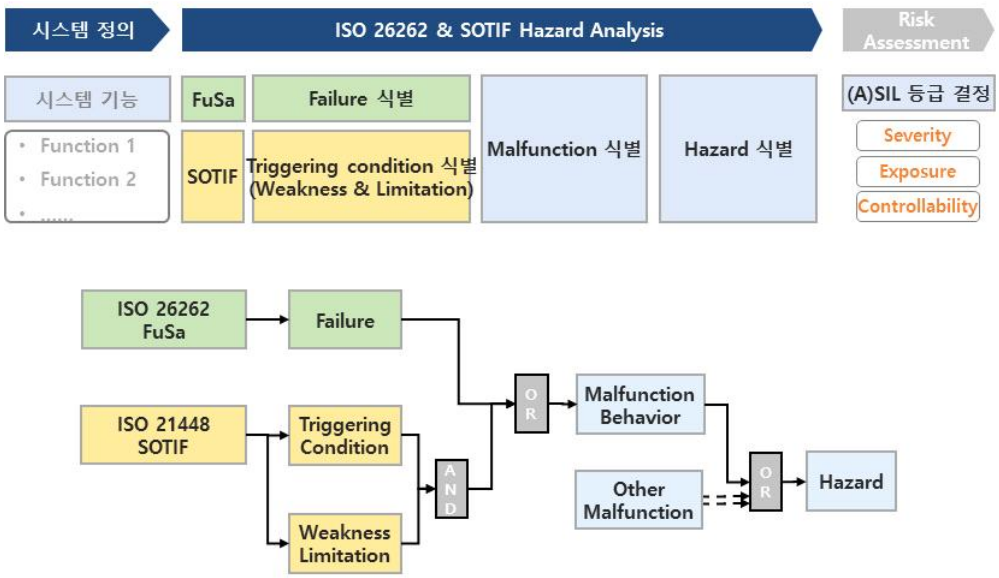

[그림 19] 기능안전 프로세스 1.4 수립

□ ISO/PAS 21448 표준, UNECE/WP.29 등을 반영한 자율주행의 비 결함 시나리오 및 시스템 사용 사례의 분석 및 검증을 위한 개발 프로세스 수립

○ ISO/PAS 21448 표준에 대응하는 SOTIF Process는 단계적 적용

– 자율주행의 비결함 시나리오 및 시스템 사용 사례의 분석, 검증 및 검증을 위한 개발 프로세스(SOTIF HARA, SOTIF Concept Verification, SOTIF verification & Validation) 적용

○ FuSa(기능안전) & SOTIF HARA 수행 프로세스를 아래와 같이 수립함




[그림 20] ISO 26262 & SOTIF Hazard Analysis

□ SW 기능 안전 프로세스의 Tailoring 적용 : SW FMEA 등에 집중하고 ASIL C 등급을 만족하는 필수 SW 개발 기술 위주로 조정하여 적용

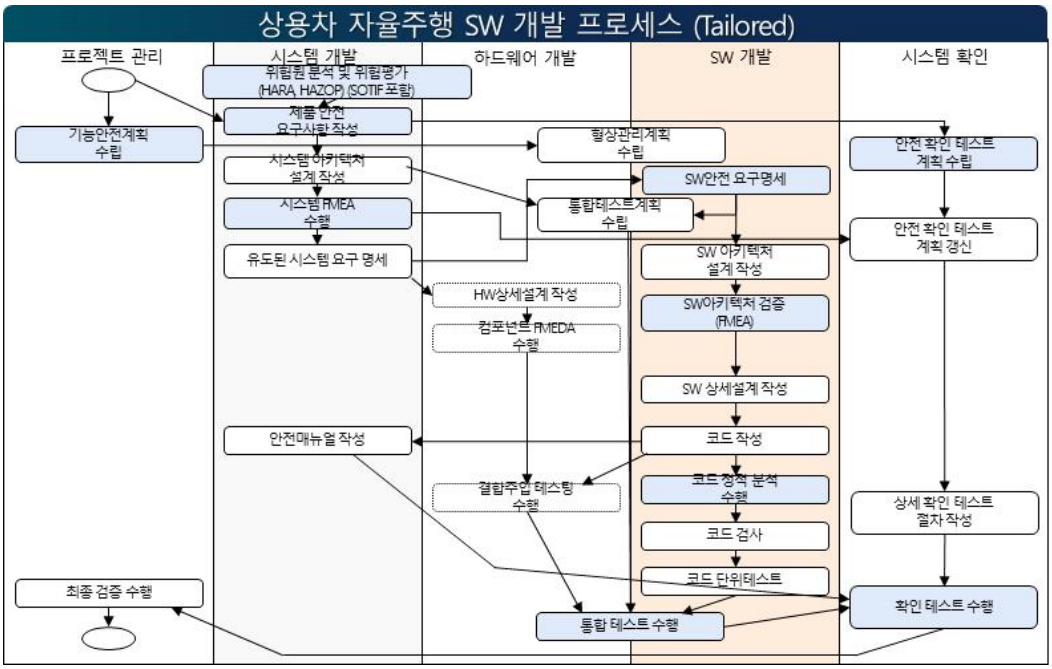
○ 2차년도는 ASIL C등급을 만족하는 활동 중에서 필수적으로 요구되는 기술 위주로 수행하도록 프로세스를 조정하여 적용함

– HARA 및 SW FMEA 중심으로 안전분석 수행

– 인스펙션 및 코드 분석, 테스트 중심으로 SW 공학 기법 적용

○ ISO/PAS 21448 표준에 대응하는 SOTIF Process는 단계적으로 적용

– 자율주행의 비결함 시나리오 및 시스템 사용 사례의 분석, 검증을 위한 SOTIF HARA 프로세스를 우선 적용하고 FMEA에도 이를 반영




[그림 21] 자율주행 기능 안전 프로세스 수립 및 Tailoring 적용

□ 안전무결성수준(ASIL) C 등급 만족을 위한 개발기법 지침 보완

○ 현재 적용하지 못하고 있는 ISO26262 ASIL C 등급 요건을 만족하는 개발을 위한 다양한 개발 및 검증 활동 기법을 2차년도에 적용하기 위한 SW 설계, 구축, 시험단계의 안전관리 프로세스 지침 및 개발기법 가이드 수립

○ 안전관리 프로세스를 효과적으로 수행하기 위한 작업별 업무수행 가이드와 산출물 템플리트를 수립하여 지원

○ ISO26262 요건 만족 프로세스 수립 및 업무수행가이드 수립

– SW개발 착수, SW아키텍처 설계, SW안전요구사항 검증, 단위 설계 및 구현, 단위시험, SW통합 및 시험, HIL시험, 안전 타당성 확인 등 각 작업에 대한 구체적 업무가이드를 수립함

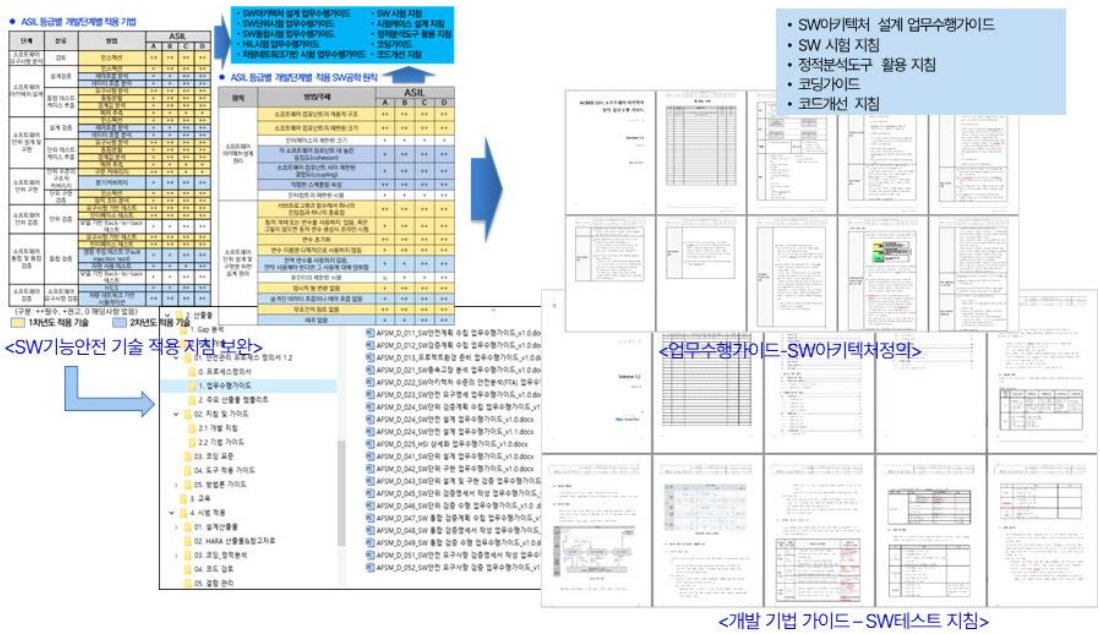
– 각 작업별 업무수행가이드는 표준 관리/개발 프로세스의 각 작업별로 절차, 역할, 입력물/출력물, 기법, 도구, 핵심체크사항 등을 제공함

○ SW 설계, 구축, 시험단계에서 ASIL C 등급이 요구하는 개발기법의 적용을 위한 기법 가이드 수립

– 업무수행가이드 중 SW아키텍처 설계 업무수행가이드는 요구되는 개발기법을 적용하기 위하여 업무수행가이드 보완

– 일부 지침은 설계 및 구현 단계의 SW공학원칙을 만족하기 위한 지침 보완

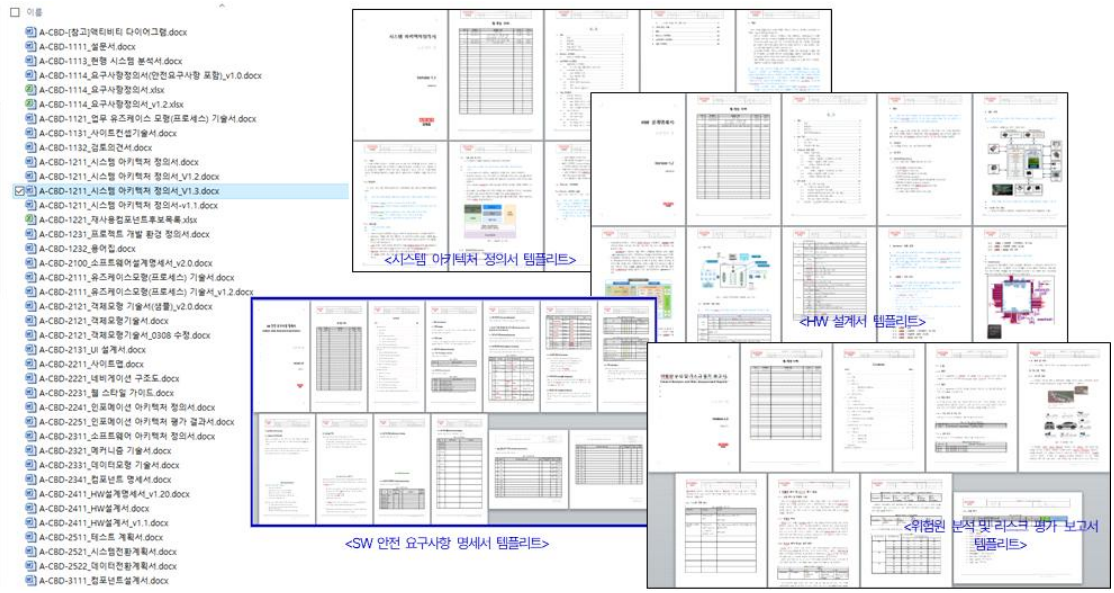
- SW 시험 지침
- 정적분석도구 활용 지침
- 코딩가이드
- 코드개선 지침




[그림 22] ISO26262 요건 만족 프로세스 수립 및 개발 지침 수립

○ 프로젝트 실용적 적용을 위한 작업별 산출물 템플리트 수립

– 각 산출별로 산출물의 주요 내용(목차) 및 각 목차별 작성 지침 및 작성 사례 등을 기술하여 각 작업자가 쉽게 산출물을 작성하도록 산출물 템플리트를 제시함




[그림 23] 산출물 템플리트 수립

□ 지침 기반의 교육 수행: 지속적인 교육 및 세미나, 토론 활동을 통하여 전문성을 배양하도록 함

○ ISO26262를 지원하는 HARA 분석 및 안전분석 기법 교육

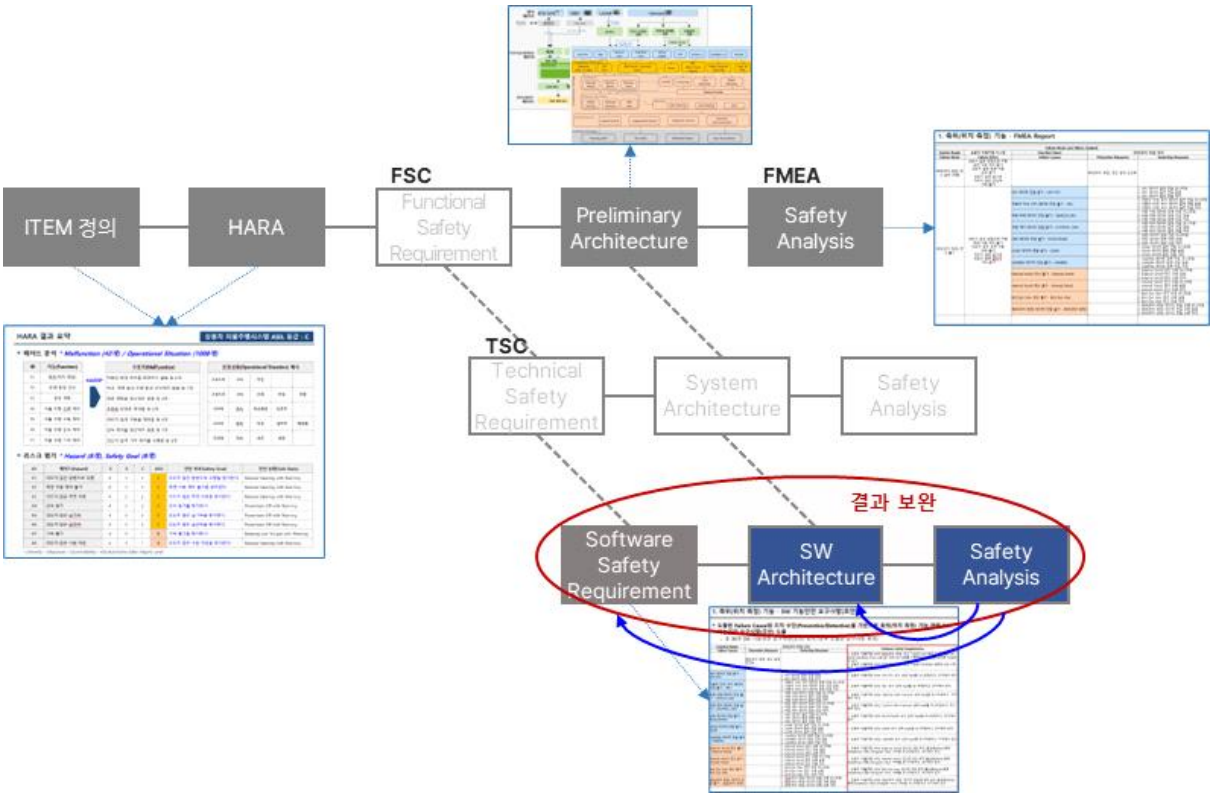
○ 업무수행가이드 및 지침 기반의 교육(안전 설계, 정적분석, 테스트 교육)

## 나. SW 안전 요구 정의

□ 자동차 레벨의 HARA 수행 후, 작성된 아키텍처를 기반으로 SW FMEA 안전 분석을 수행하고, 이미 널리 알려진 SW 기능안전 메커니즘과 도출된 결함 원인을 맵핑하여 SW 기능안전 요구사항을 도출

○ 1차년도에 수행한 HARA 및 FMEA를 기반으로 고속도로 자율주행 테스트를 위한 추가개발사항을 포함하여 HARA(위험원 분석 및 리스크 평가) 및 FMEA(고장모드 및 영향분석) 수행

○ HARA 및 FMEA 수행범위, 안전요구사항 정의 방안은 1차년도 내용과 유사하나 의도된 기능안전(SOTIF)의 안전분석을 포함하여 수행




[그림 24] 2차년도 안전분석 및 안전요구사항 정의

□ Level 3 자율주행 상용차 개발에 결함으로부터 기인한 안전분석 및 검증(ISO26262 요건) 외에 비결함 시나리오 및 시스템 사용 사례의 분석, 검증 및 검증을 위한 개발 프로세스(SOTIF HARA, SOTIF Concept Verification, SOTIF verification & Validation) (ISO/PAS 21448 요건) 적용

○ 자율주행차의 위험원 분석 및 리스크 평가, 고장모드 및 영향 분석을 통한 안전 목표 및 안전요구사항을 도출

○ 자동차 전장부품의 안전분석 및 안전메카니즘은 많은 사례가 있지만, 자율주행차의 안전분석 및 안전메카니즘의 설계는 국내에서는 선도적인 사례임

○ 특히, 자율주행의 비결함 시나리오 및 시스템 사용 사례의 분석, 검증 및 검증을 위한 개발 프로세스(SOTIF HARA, SOTIF FMEA 및 검증) 적용한 부분은 거의 첫 사례로 보임

– 국제적으로, 의도된 기능의 안전”(Safety of the Intended Functionality, SOTIF)이라는 별도의 표준 ISO/PAS 21448을 개발 중이며, 자율주행차의 경우에는 이의 적용이 필요함

• ISO 26262를 준수하는 완벽한 소프트웨어 및 하드웨어를 탑재한 차량이 센서나 시스템의 성능 제한, 예기치 않은 도로 환경의 변화, 예상할 수 없는 운전자의 기능 오용으로 인해 사고가 난 예가 많음

- 차량의 하드웨어나 소프트웨어에 의한 오작동이 없는 경우에도 ADAS나 자율주행 차량(Autonomous Vehicles, AV)에서 불합리한 위험(Unreasonable risk)을 방지해야 함

-

## 다. 자율주행 SW HARA(위험원 분석 및 위험평가)

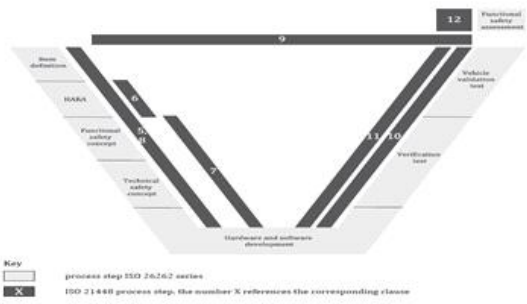
□ SW공학기술 및 기능안전 기술의 적용으로, 고장으로 인한 위험으로부터 안전한 자율주행차량 SW를 개발하여, **자율주행 임시운행면허를 획득**하였음

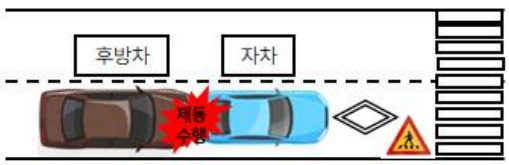
○ 자율주행 시스템의 응답율, Fail-safe 기능 만족도, 긴급제동브레이크 시스템 성능 평가 만족도를 100% 달성

○ fail-safe를 위한 기능안전 메커니즘 도입 건수도 10개로 100%달성

○ SW공학기술 도입 지표측면에서는, 동료검토 효율성은 시간당 결함발견 2.1개로 100% 달성하였으며, 소스코드 복잡도도 복잡도 위반 46건, Call Level 위배 4건으로 100%달성

| 표준 | Cause for Hazardous event (Pre-condition) | 내/외부 |
|---|---|---|
| ISO 26262 | E/E 시스템의 고장 | 내부 |
| ISO 21448 (SOTIF) | 성능 제한 | 내부 |
| | 예측 가능한 사용자 오용 | 내/외부 |
| | 차량 주변 환경으로부터의 영향 | 외부 |
| ISO SAE 21434 (Cyber Security) | 차량 보안 취약점을 이용한 사이버 공격 | 외부 |



| 구분 | ISO 26262 | SOTIF |
|---|---|---|
| 목적 | E/E 시스템에서 발생하는 고장에 대한 회피 및 방지로 차량 안전성 확보 | E/E 시스템이 동작하는 성능 및 환경요인 대비 충분성 확보 |
| 시나리오 | 자율주행 기능 사용 시 의도치 않은 제동 기능으로 인한 차량 추돌사고 발생 | |
| 세부 시나리오 | 기능의 고장으로 제동이 불가하여 전방차량과의 사고 발생 | 성능 제한 및 오인식으로 전방차량이 있다고 판단하여, 의도치 않은 제동으로 후방차량과의 사고 발생 |
| |  |  |
| 원인 | 시스템 안전 요구사항 검증 미흡<br>소프트웨어 코드 메트릭기준 미달성<br>하드웨어 부품의 고장 | 좋지 않은 환경 조건(짙은 안개, 강한 햇빛, 폭설 등)<br>표지판/바닥의 그림 오인식<br>의도치 않은 상황에 의한 사용자 오용 |

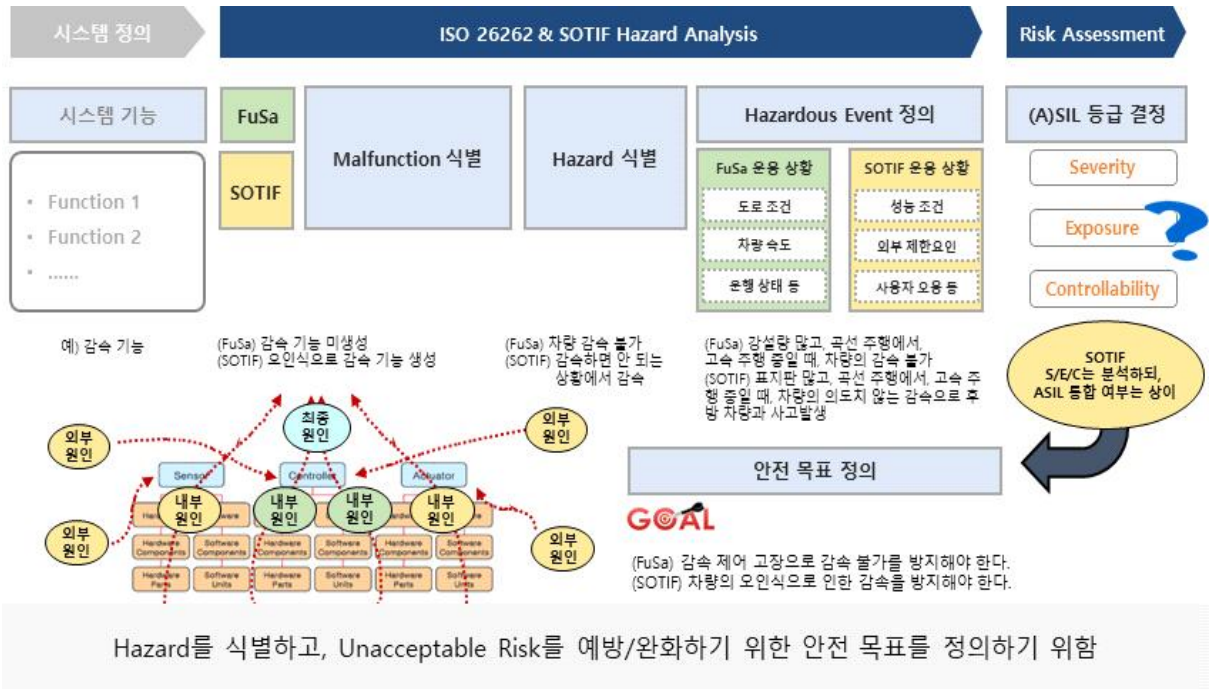
[그림 25] SOTIF 프로세스 적용

□ 위험원 분석 및 리스크 평가(HARA) : Lv.3 자율주행 상용차 Use case를 기반으로 주요 위험원을 분석하여 기능 안전 무결성 등급(ASIL)과 주요 안전 목표를 정의

○ ISO 21448 (SOTIF) HARA는 추가 분석

– 결함으로부터 기인한 안전분석 및 검증(ISO26262 요건) 외에 의도된 기능의 안전을 위한 개발 프로세스 (ISO/PAS 21448 요건) 적용

– 자율주행의 비결함 시나리오 및 시스템 사용 사례의 분석, 검증 및 검증을 위한 개발 프로세스(SOTIF HARA, SOTIF FMEA 및 검증) 적용

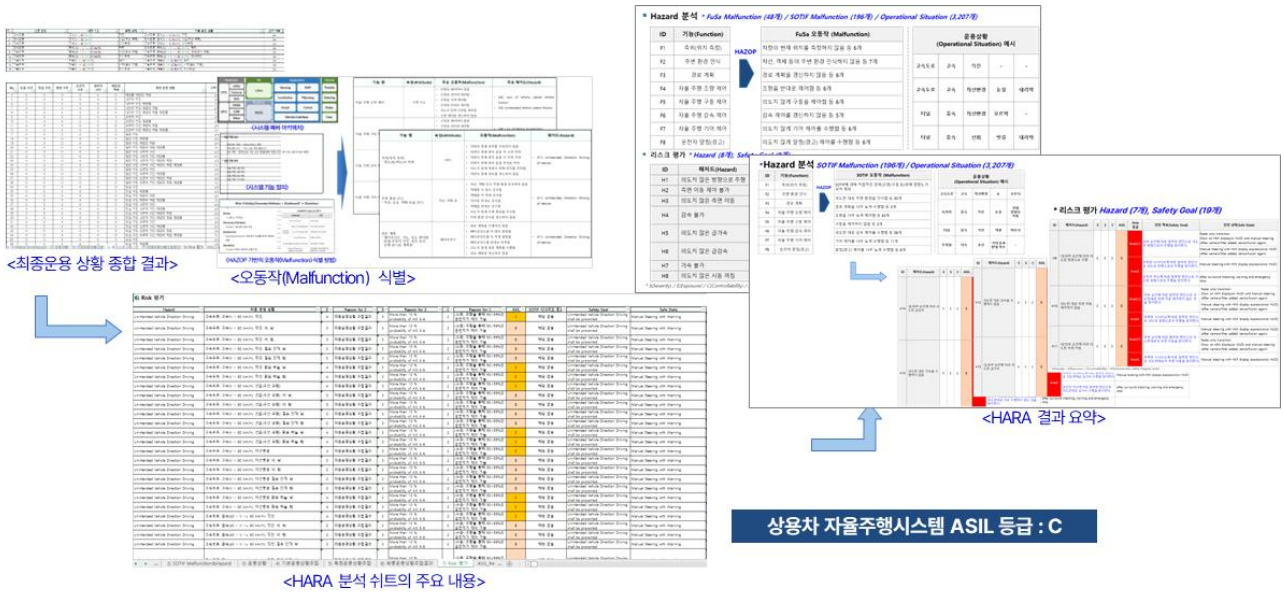



[그림 26] ISO26262 & SOTIF HARA 적용

□ 위험원 분석 및 리스크 평가(HARA) - 절차 및 결과 요약

– Lv3. 상용차 자율주행 운용환경 등을 고려하여 위험원을 분석하고 위험을 평가하여 안전목표를 정의함

– HARA의 결과로 도출되는 ASIL(안전무결성 등급) 및 Safety Goal(안전목표) 정의

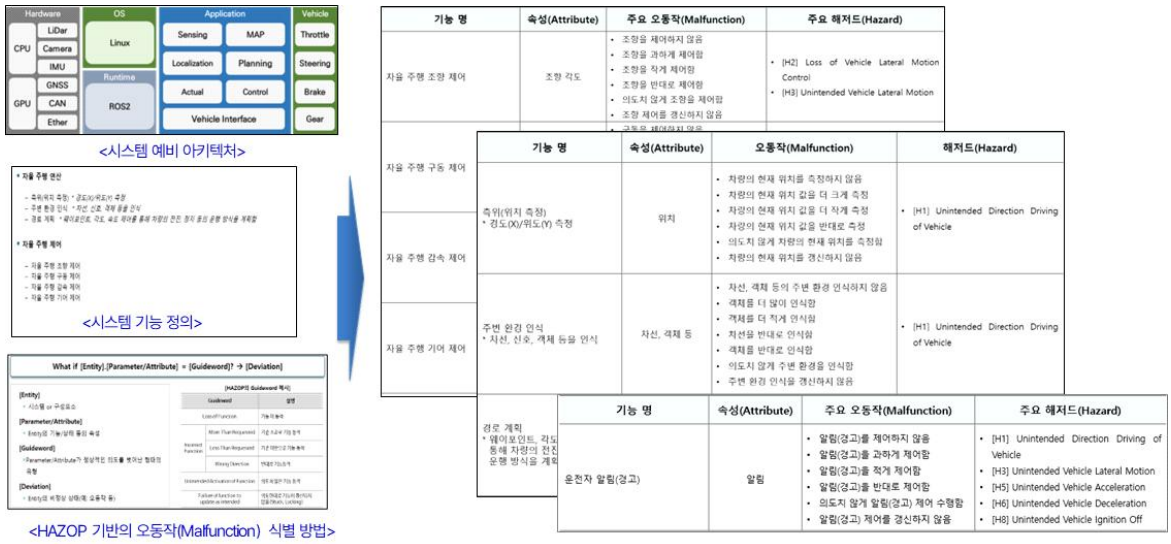



[그림 27] 위험원 분석 및 리스크 평가(HARA) - 요약

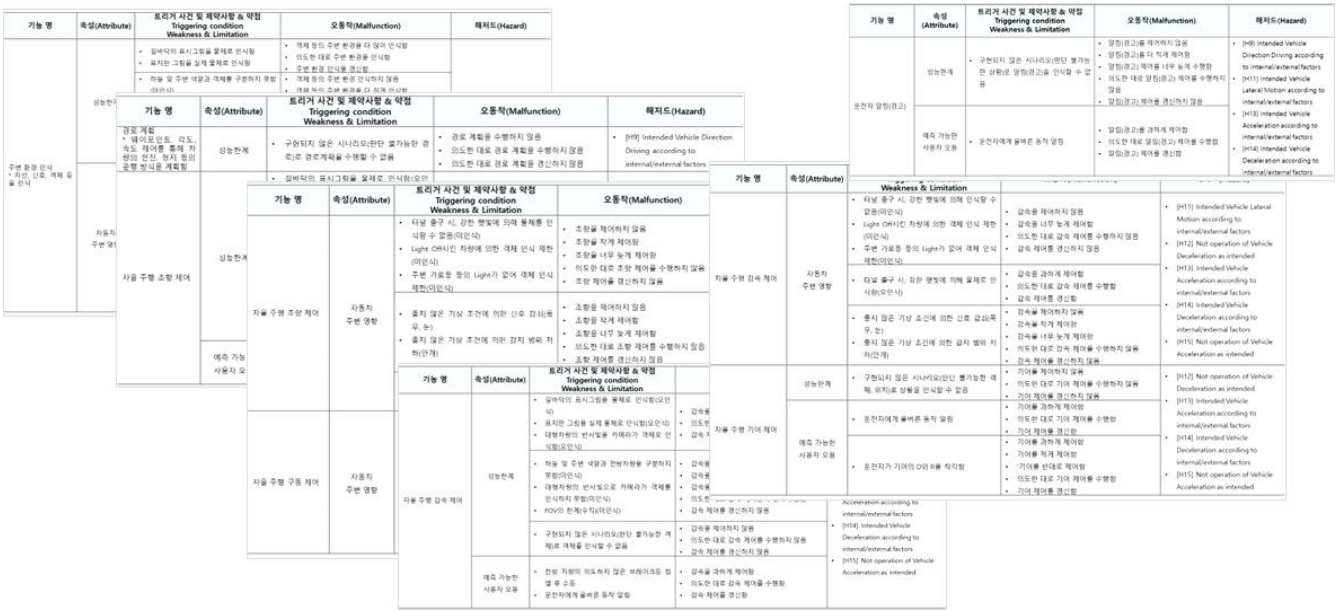
- 주요 기능에 대한 오동작을 위험원(헤저드)으로 분석하고 각 위험원에 대하여 운영상황 및 Exposure, 심각도, 통제가능성을 분석하여 위험을 평가
- 해저드 분석을 통해 고장에 의한 오작동에 대한 3,207개의 운영상황과 주요 Hazard 15개(기능안전 8개, 의도된 기능의 안전 7개)의 조합에 의한 총 20,916개의 Hazardous Event가 분석되어 안전무결성 등급을 결정하고 안전목표를 정의
- 고속도로, 거점연결 시가지 중심 등의 운용상황을 고려

○ 위험원 분석 및 리스크 평가(HARA) - 오동작 식별

- 시스템 기능 정의, 예비 아키텍처를 기반으로, 시스템의 정상 기능/상태 등에 HAZOP Guideword를 적용하여 오동작(Malfunction)을 식별
- 기능 별 기능안전(FuSa) 오동작(Malfunction) 및 해저드 정의
- 기능 별 의도된 기능의 안전(SOTIF) 오동작(Malfunction) 및 해저드 정의




[그림 28] 위험원 분석 및 리스크 평가(HARA) – 오동작 식별

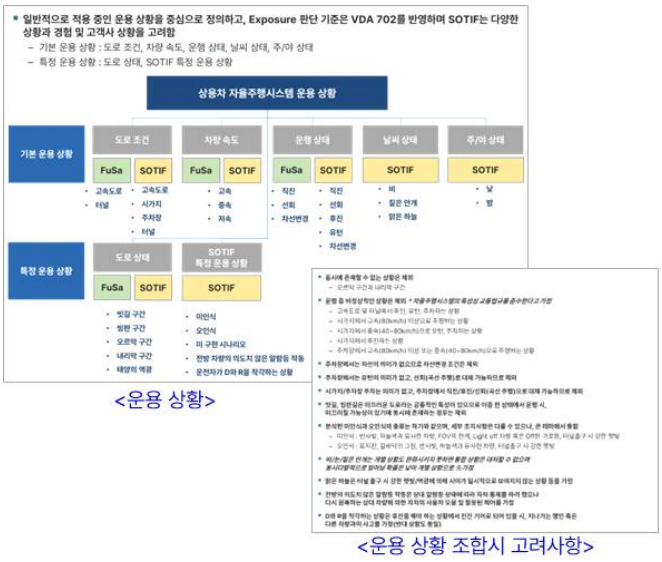


[그림 29] SOTIF 오동작 및 해저드 정의

○ 위험원 분석 및 리스크 평가(HARA) - 운용상황 정의

– 운용상황을 정의하고, Exposure 판단 기준은 VDA 702를 반영하고 기본적 상황을 고려하여 Risk 평가

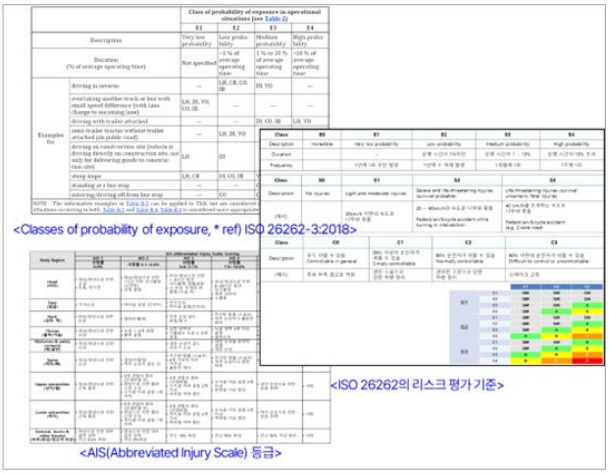
– Exposure, Severity, Controllability를 고려하여 Risk 평가




<운용 상황>



<운용 상황 조합시 고려사항>



<Classes of probability of exposure, * ref) ISO 26262-3:2018>

<ISO 26262의 리스크 평가 기준>

<AIS(Abbreviated Injury Scale) 등급>

<Risk 평가>

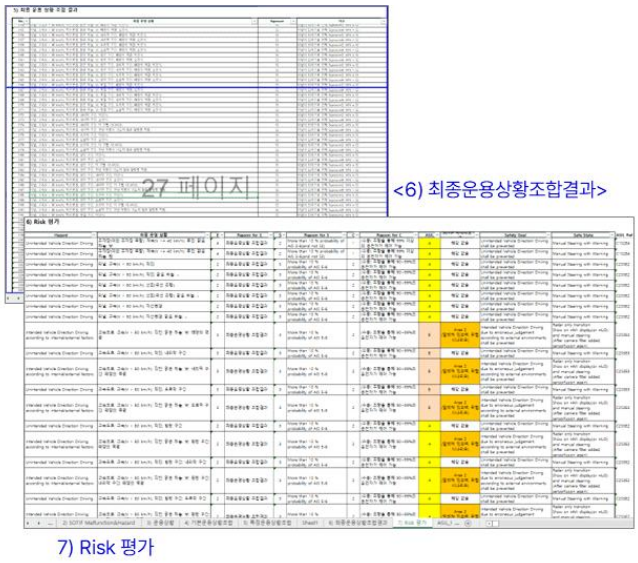
[그림 30] 운용 상황 정의 및 Risk 평가

○ 위험원 분석 및 리스크 평가(HARA) – HARA 분석 쉬트

– HARA 분석 쉬트를 활용하여 각 운용상황별로 Exposure, Severity, Controllability를 고려하여 Risk 평가


<1) FuSa Malfunction&Hazard>

<2) SOTIF Malfunction&Hazard>




<6) 최종운용상황조합결과>

7) Risk 평가

[그림 31] 위험원 분석 및 리스크 평가(HARA) – HARA 분석 쉬트

○ 기능안전(Functional Safety) HARA 분석 : 결과 요약

– 8개의 주요 기능에 대하여 48개의 기능안전 관련 오작동과 8개의 해저드를 식별

– 각 헤저드에 대하여 안전목표와 안전상태를 정의

- Hazard 분석 ** FuSa Malfunction (48개) / SOTIF Malfunction (196개) / Operational Situation (3,207개)*

| ID | 기능(Function) |
|---|---|
| F1 | 측위(위치 측정) |
| F2 | 주변 환경 인식 |
| F3 | 경로 계획 |
| F4 | 자율 주행 조향 제어 |
| F5 | 자율 주행 구동 제어 |
| F6 | 자율 주행 감속 제어 |
| F7 | 자율 주행 기어 제어 |
| F8 | 운전자 알림(경고) |

HAZOP

| FuSa 오동작 (Malfunction) |
|---|
| 차량의 현재 위치를 측정하지 않음 등 6개 |
| 차선, 객체 등의 주변 환경 인식하지 않음 등 7개 |
| 경로 계획을 갱신하지 않음 등 6개 |
| 조향을 반대로 제어함 등 6개 |
| 의도치 않게 구동을 제어함 등 6개 |
| 감속 제어를 갱신하지 않음 등 5개 |
| 의도치 않게 기어 제어를 수행함 등 6개 |
| 의도치 않게 알림(경고) 제어를 수행함 등 6개 |

| 운용상황 (Operational Situation) 예시 | | | | |
|---|---|---|---|---|
| 고속도로 | 고속 | 직진 | - | - |
| 고속도로 | 고속 | 차선변경 | 눈길 | 내리막 |
| 터널 | 중속 | 차선변경 | 오르막 | - |
| 터널 | 중속 | 선회 | 빗길 | 내리막 |

- 리스크 평가 ** Hazard (8개), Safety Goal (8개)*

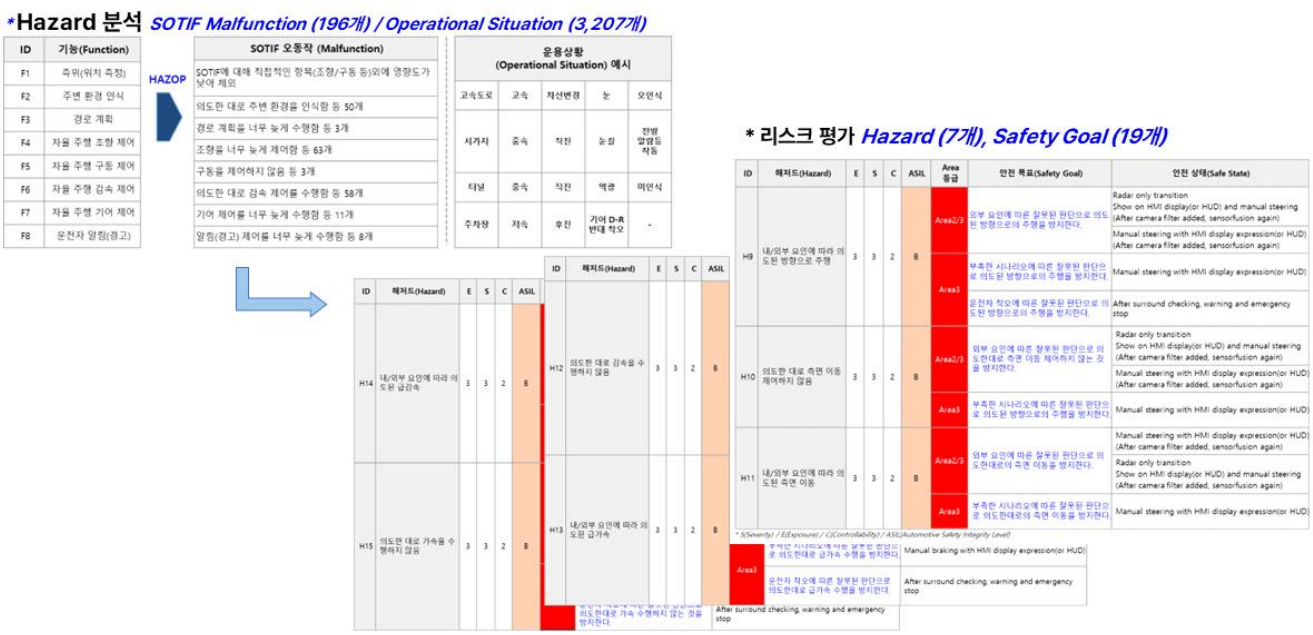
| ID | 해저드(Hazard) | E | S | C | ASIL | 안전 목표(Safety Goal) | 안전 상태(Safe State) |
|---|---|---|---|---|---|---|---|
| H1 | 의도치 않은 방향으로 주행 | 4 | 3 | 2 | C | 의도치 않은 방향으로 주행을 방지한다. | Manual Steering with Warning |
| H2 | 측면 이동 제어 불가 | 4 | 3 | 2 | C | 측면 이동 제어 불가를 방지한다. | Manual Steering with Warning |
| H3 | 의도치 않은 측면 이동 | 4 | 3 | 2 | C | 의도치 않은 측면 이동을 방지한다. | Manual Steering with Warning |
| H4 | 감속 불가 | 4 | 3 | 2 | C | 감속 불가를 방지한다. | Manual Braking with warning & Assisted braking torque |
| H5 | 의도치 않은 급가속 | 4 | 3 | 2 | C | 의도치 않은 급가속을 방지한다. | Manual Braking with warning & Assisted braking torque |
| H6 | 의도치 않은 급감속 | 4 | 3 | 2 | C | 의도치 않은 급감속을 방지한다. | Manual Braking with warning & Assisted braking torque |
| H7 | 가속 불가 | 4 | 3 | 2 | C | 가속 불가를 방지한다. | Keeping Last Torque with Warning |
| H8 | 의도치 않은 시동 꺼짐 | 4 | 3 | 1 | B | 의도치 않은 시동 꺼짐을 방지한다. | Manual Steering with Warning |

[그림 32] 기능안전(Functional Safety) HARA 분석 : 결과 요약

○ 의도된 기능안전(SOTIF) HARA 분석 : 결과 요약

– 8개의 주요 기능에 대하여 196개의 SOTIF 오작동과 7개의 해저드를 식별

– 각 헤저드에 대하여 안전목표와 안전상태를 정의




[그림 33] SOTIF HARA 분석 : 결과 요약

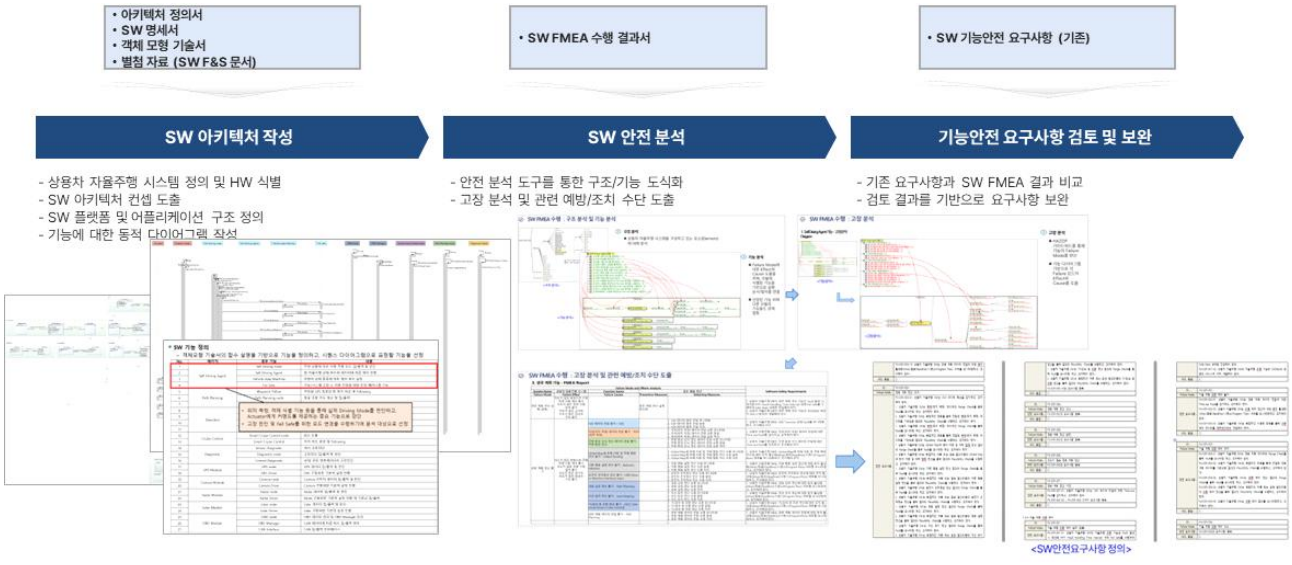
## 라. 자율주행 SW FMEA( 고장모드 및 영향분석)

□ 시스템 구조 및 기능 분석을 통하여 고장분석 결과를 도출하고 대응분석을 통하여 SW 안전 요구사항을 정의

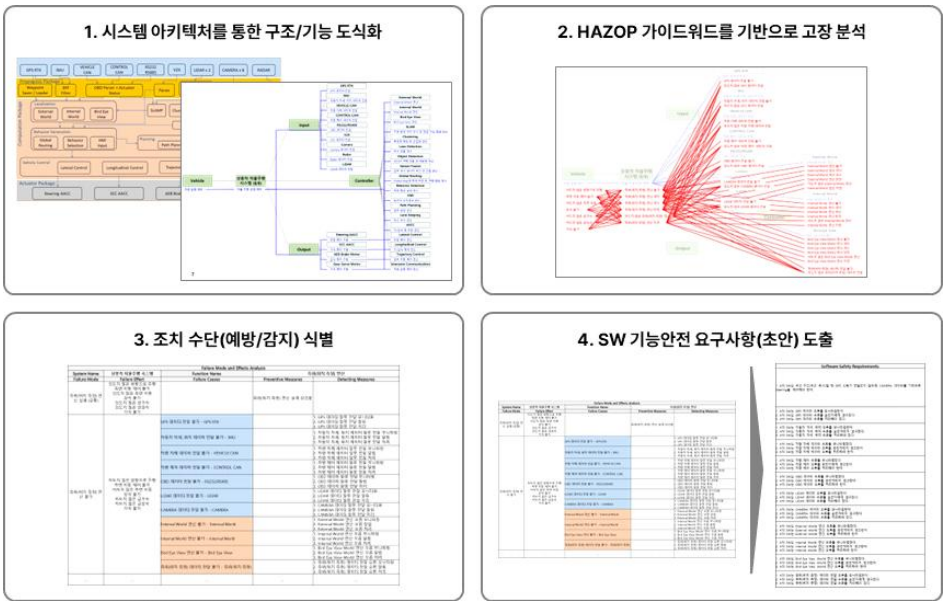
○ 기능안전 메커니즘 설계가 충분한지 판단하기 위해, SW 아키텍처 설계서 작성 및 SW FMEA 수행

– 도출된 SW 기능안전 요구사항에 대한 완전성 검토

– 상세화된 시스템 아키텍처를 기반으로 FMEA 수행하며, SW 모듈 단위로 Failure Cause를 분석 및 연관된 SW 기능안전 요구사항(초안) 작성




[그림 34] SW FMEA 수행




[그림 35] FMEA 수행 : 절차

○ Network Diagram 도식화 및 FMEA Report 작성

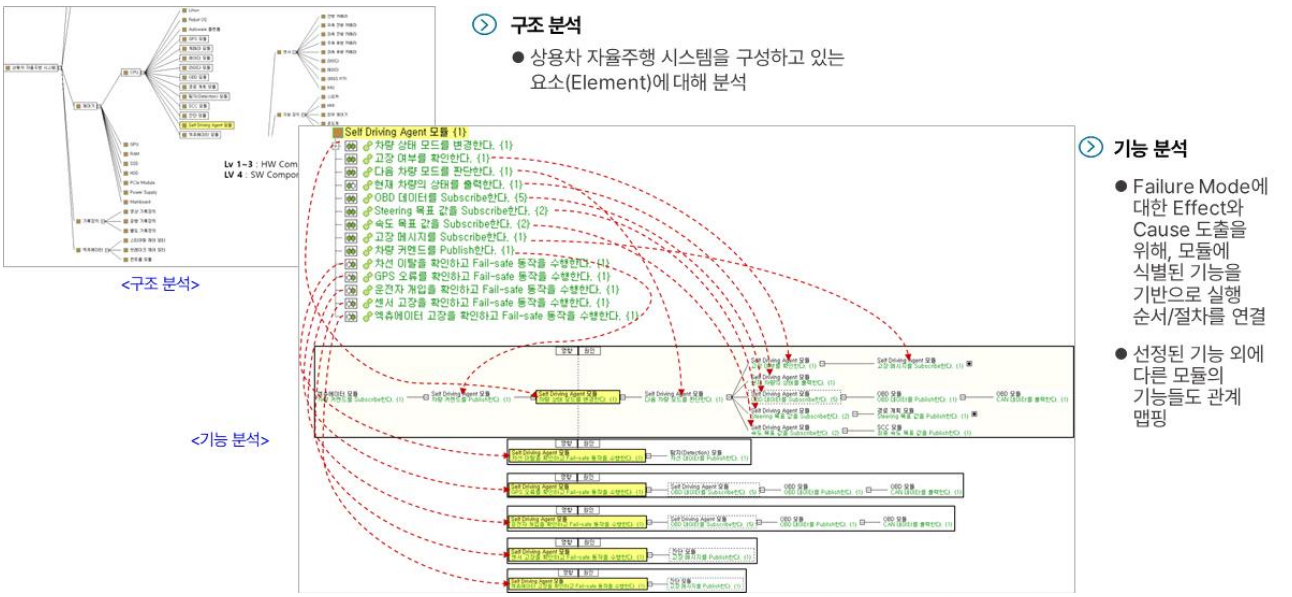
– FMEA 프로세스에 따른 구조/기능/고장 Network Diagram을 도식화

– 예방/감지 수단 및 SW기능안전 요구사항을 FMEA Report로 작성

○ 기능 분석 : 구성 요소별 기능을 정의하고, 상위 기능을 달성하기 위해 필요한 하위 기능을 원인(cause)-결과(effect)의 관계로 연결

– Failure Mode에 대한 Effect와 Cause 도출을 위해, 모듈에 식별된 기능을 기반으로 실행 순서/절차를 연결

– 선정된 기능 외에 다른 모듈의 기능들도 관계 맵핑



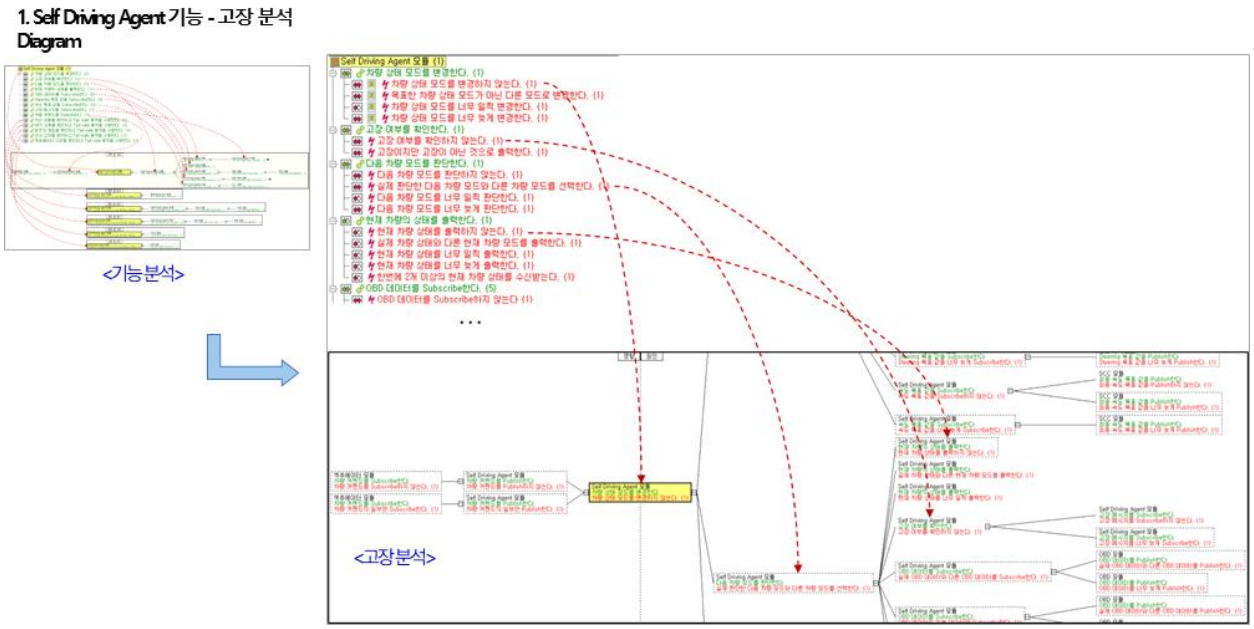

[그림 36] 고장모드 및 영향분석(FMEA): 구조 분석 및 기능분석

○ 고장 분석 : Malfunction의 식별과 연결

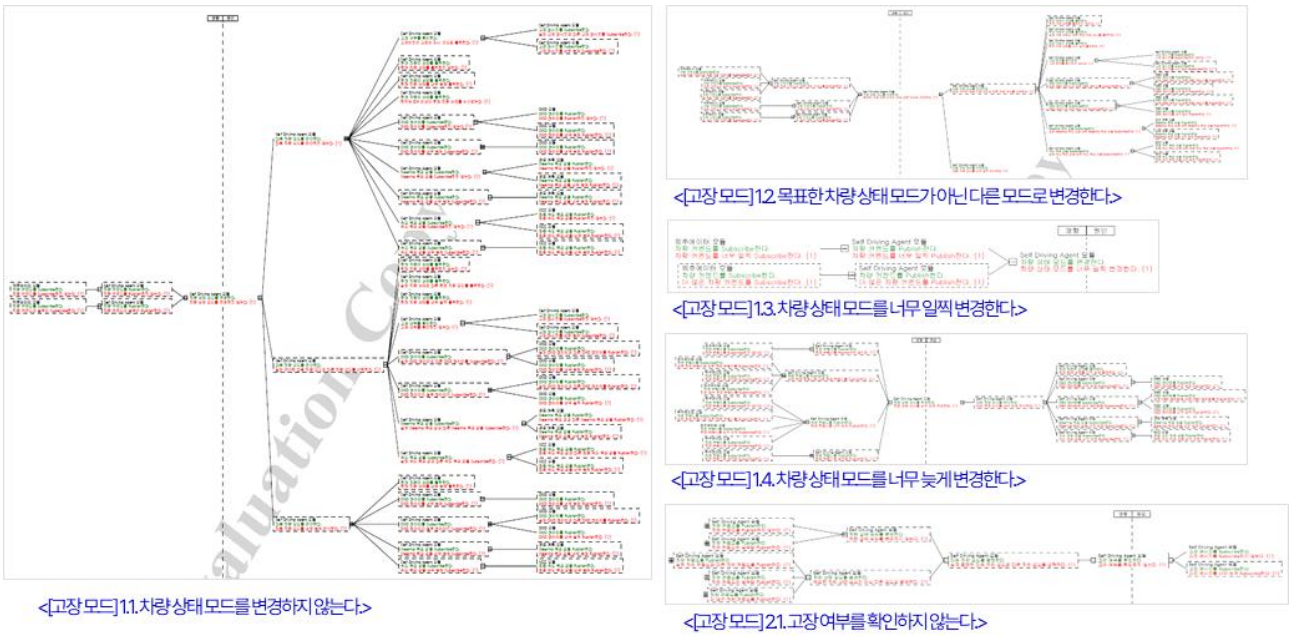
– 정의한 기능 별 오동작(Malfunction)을 정의함: HAZOP 가이드워드를 통해 기능의 Failure Mode를 판단

– 상위 Malfunction을 유발하는 하위의 Malfunction을 원인(cause)-결과(effect)의 관계로 연결함 : 기능 다이어그램 기반으로 각 Failure 모드의 Effect와 Cause를 도출

– FMEA Worksheet로 표현




[그림 37] 고장모드 및 영향분석(FMEA) : 고장분석




[그림 38] 고장모드 및 영향분석(FMEA) : 고장분석 사례

○ 활동(대응) 분석 및 요구사항 정의

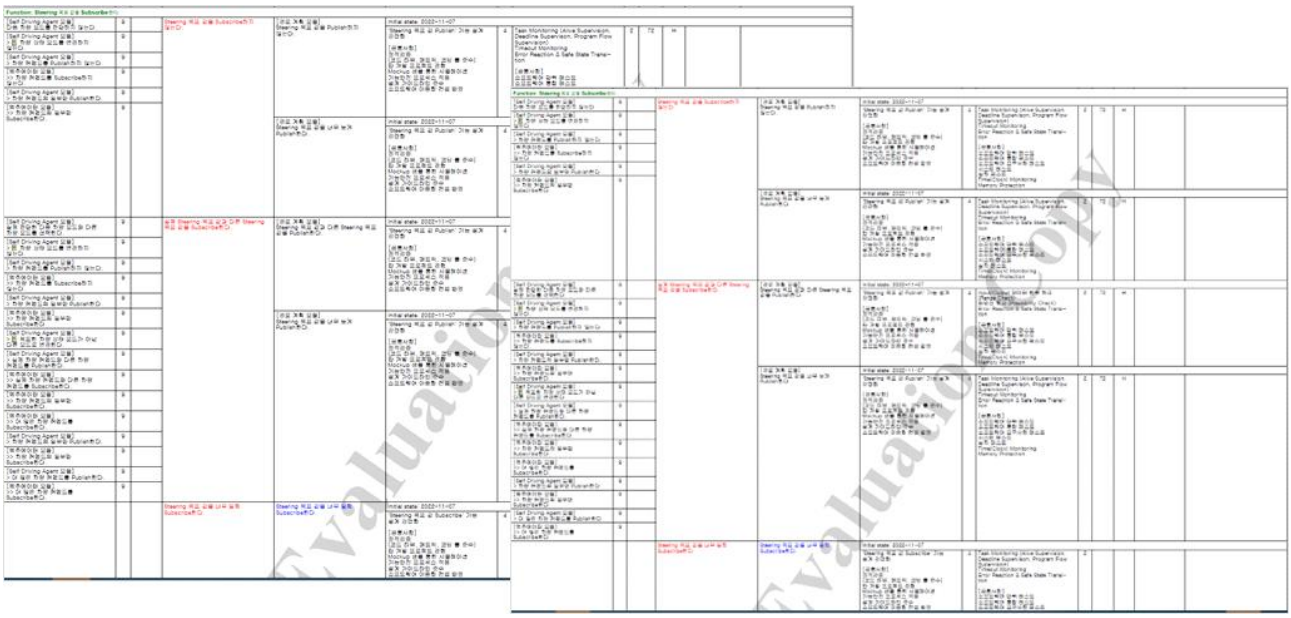
- 고장 분석을 바탕으로 Safety Measure를 고려하여 소프트웨어 안전 요구사항을 정의함
- 주요 감지(detection Measure)와 Fail-safe 기능을 우선 정의함
- 설계 관점에서 고장 예방 또는 감지할 수 있는 수단(Measure) 식별
- 총 770개 항목에 대한 분석 결과 도출

| | Preventive Action | Detection Action |
|---|---|---|
| 공통 사항 | • 정적검증 (코드 리뷰, 메트릭, 코딩 룰 준수)<br>• 타 개발 프로젝트 경험<br>• Mockup 샘플 통한 시뮬레이션<br>• 기능안전 프로세스 적용<br>• 설계 가이드라인 준수<br>• 소프트웨어 이중화 컨셉 반영 | • 소프트웨어 단위 테스트<br>• 소프트웨어 통합 테스트<br>• 소프트웨어 요구사항 테스트<br>• 시스템 테스트<br>• 실차 테스트<br>• Time(Clock) Monitoring<br>• Memory Protection |
| 기능별 | • 기능별 설계 강건화 (ASIL 개발) | • Task Monitoring (Alive Supervision, Deadline Supervison, Program Flow Supervision)<br>• Timeout Monitoring<br>• Input/Output 데이터 범위 체크 (Range Check)<br>• 타당성 체크 (Plausibility Check)<br>• E2E Protection (CRC, Sequence Counter, Alive Counter, Timeout)<br>• Error Reaction & Safe State Transition |

| Effect | S | Spec. Charac-teristic | Failure mode | Cause | Preventive action | O | Detection action | D | RPN |
|---|---|---|---|---|---|---|---|---|---|
| > 차량 상태 모드를 변경하지 않는다. | 9 | | 실제 고장 메시지와 다른 고장 메시지를 Subscribe한다. | 실제 고장 메시지와 다른 고장 메시지를 Subscribe한다. | '고장 메시지 Subscribe' 기능 설계 강건화<br>[공통사항]<br>정적검증<br>(코드 리뷰, 메트릭, 코딩 룰 준수)<br>타 개발 프로젝트 경험<br>Mockup 샘플 통한 시뮬레이션<br>기능안전 프로세스 적용<br>설계 가이드라인 준수<br>소프트웨어 이중화 컨셉 반영 | 4 | Input/Output 데이터 범위 체크 (Range Check)<br>타당성 체크 (Plausibility Check)<br>Error Reaction & Safe State Transition<br>[공통사항]<br>소프트웨어 단위 테스트<br>소프트웨어 통합 테스트<br>소프트웨어 요구사항 테스트<br>시스템 테스트<br>실차 테스트<br>Time(Clock) Monitoring<br>Memory Protection | 2 | 72 |
| [Self Driving Agent 모듈]<br>> 차량 커맨드를 Publish하지 않는다. | 9 | | | | | | | | |
| [역추에이터 모듈]<br>>> 차량 커맨드를 Subscribe하지 않는다. | 9 | | | | | | | | |
| [Self Driving Agent 모듈]<br>> 차량 커맨드의 일부만 Publish한다. | 9 | | | | | | | | |
| [역추에이터 모듈]<br>>> 차량 커맨드의 일부만 Subscribe한다. | 9 | | | | | | | | |
| [Self Driving Agent 모듈]<br>고장 여부를 확인하지 않는다. | 9 | | 고장 메시지를 너무 늦게 Subscribe한다. | 고장 메시지를 너무 늦게 Subscribe한다. | Initial state: 2022-11-07 | | | | |
| [Self Driving Agent 모듈]<br>> 실제 확인한 다음 차량 모드와 다른 차량 모드를 견택한다. | 9 | | | | '고장 메시지 Subscribe' 기능 설계 강건화<br>[공통사항]<br>정적검증<br>(코드 리뷰, 메트릭, 코딩 룰 준수)<br>타 개발 프로젝트 경험<br>Mockup 샘플 통한 시뮬레이션<br>기능안전 프로세스 적용<br>설계 가이드라인 준수<br>소프트웨어 이중화 컨셉 반영 | 4 | Input/Output 데이터 범위 체크 (Range Check)<br>타당성 체크 (Plausibility Check)<br>E2E Protection (CRC, Sequence Counter, Alive Counter, Timeout)<br>Error Reaction & Safe State Transition<br>[공통사항]<br>소프트웨어 단위 테스트<br>소프트웨어 통합 테스트<br>소프트웨어 요구사항 테스트<br>시스템 테스트 | 2 | 72 |
| [Self Driving Agent 모듈]<br>> 차량 상태 모드를 변경하지 않는다. | 9 | | | | | | | | |
| [Self Driving Agent 모듈]<br>> 차량 커맨드를 Publish하지 않는다. | 9 | | | | | | | | |

*별첨 1. FMEA SOD Rating 기준 (AIAG-VDA FMEA)

[그림 39] 예방/조치 수단 도출



[그림 40] FMEA Sheet

○ 고장모드 및 영향분석(FMEA) 및 SW 요구사항 분석 - 자율주행 시스템 FMEA 수행

– 도출된 Failure Cause와 조치 수단(Preventive/Detective)을 기반으로 SW 기능안전 요구사항 도출

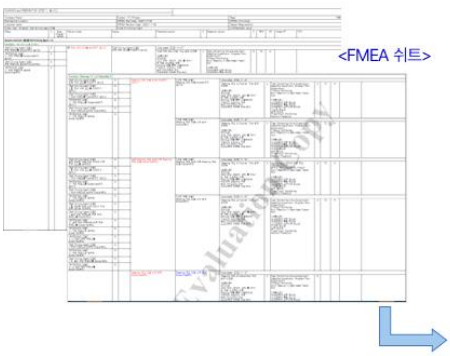
– 예시(일부): 측위(위치 측정) 기능 관련 SW 기능안전 요구사항(초안) 정의

| '21년 SW 기능안전 요구사항 (28개) | '22년 SW 기능안전 요구사항 (77개) |
|---|---|
| 상용차 자율주행 SW의 경로 계획 연산 기능은 Fault 발생 시, 정의된 FHTI (Fault Handling Time Interval) 내에 Fail Safe를 수행하여 Safe State 상태로 진입해야 한다. | 상용차 자율주행 SW의 경로 계획 연산 기능은 Fault 발생 시, 정의된 FHTI (Fault Handling Time Interval) 내에 Fail Safe를 수행하여 Safe State 상태로 진입해야 한다. |
| 상용차 자율주행 SW의 경로 계획 연산 기능은 ISO26262 표준의 ASIL-C에 따라 개발해야 한다. | 상용차 자율주행 SW의 '차량 상태 모드 변경' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '고장 여부 확인' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '다음 차량 모드 판단' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '현재 차량 상태 출력' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 'OBD 데이터 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 'Steering 목표 값 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '속도 목표 값 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '고장 메시지 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '차량 커맨드 Publish' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '차선 이탈 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 'GPS 오류 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '운전자 개입 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '센서 고장 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다.<br>상용차 자율주행 SW의 '액츄에이터 고장 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. |
| 상용차 자율주행 SW는 Global Map에 현재 차량 및 객체 맵핑 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | - |
| 상용차 자율주행 SW는 차량 행동 설정 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | 상용차 자율주행 SW는 '차량 상태 모드 변경' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| - | 상용차 자율주행 SW는 '고장 여부 확인' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| - | 상용차 자율주행 SW는 '다음 차량 모드 판단' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| - | 상용차 자율주행 SW는 '현재 차량 상태 출력' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| - | 상용차 자율주행 SW는 '고장 메시지 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| 상용차 자율주행 SW는 운전자 조작정보 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | 상용차 자율주행 SW는 'OBD 데이터 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| 상용차 자율주행 SW는 경로 설정 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | - |
| 상용차 자율주행 SW는 차선 유지 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | 상용차 자율주행 SW는 'Steering 목표 값 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| 상용차 자율주행 SW는 가/감속 및 조향 연산에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | 상용차 자율주행 SW는 '차량 커맨드 Publish' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| 상용차 자율주행 SW는 경로 계획 데이터 전달에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | 상용차 자율주행 SW는 '속도 목표 값 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |
| - | 상용차 자율주행 SW는 '차선 이탈 확인 및 Fail-safe 동작' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. |

[그림 41] '21년 / '22년 SW 기능안전 요구사항과 결과 비교

(중복 도출된 요구사항 제외) 등

- 총 77개 SW 기능안전 요구사항 정의
- 전년 대비 SW 기능안전 요구사항 59건 추가 도출




<SW안전요구사항 정의>

| SW 기능안전 요구사항 | ASIL | 출처 | 조치 수단 타입 | 비고 |
|---|---|---|---|---|
| 상용차 자율주행 SW의 경로 계획 연산 기능은 Fault 발생 시, 정의된 FHTI (Fault Handling Time Interval) 내에 Fail Safe를 수행하여 Safe State 상태로 진입해야 한다. | ASIL C | - | Preventive Action | |
| 상용차 자율주행 SW의 '차량 상태 모드 변경' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '고장 여부 확인' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '다음 차량 모드 판단' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '현재 차량 상태 출력' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 'OBD 데이터 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 'Steering 목표 값 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '속도 목표 값 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '고장 메시지 Subscribe' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '차량 커맨드 Publish' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | '액츄에이터 모듈 FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '차선 이탈 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | CPU FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 'GPS 오류 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | CPU FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '운전자 개입 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | CPU FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '센서 고장 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | CPU FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW의 '액츄에이터 고장 확인 및 Fail-safe 동작' 기능은 ASIL-C에 따라 개발해야 한다. | ASIL C | CPU FMEA 결과 | Preventive Action | |
| 상용차 자율주행 SW는 Time(Clock)에 대한 모니터링을 수행해야 한다. | ASIL C(C) | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | Timer QM(C)와 ASIL Decomposition 수행 |
| 상용차 자율주행 SW는 QM 모듈이 ASIL 메모리 영역에 쓰는 것을 방지(Protection)해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '차량 상태 모드 변경' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '고장 여부 확인' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '다음 차량 모드 판단' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '현재 차량 상태 출력' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 'OBD 데이터 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 'Steering 목표 값 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '속도 목표 값 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '고장 메시지 Subscribe' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '차량 커맨드 Publish' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '차선 이탈 확인 및 Fail-safe 동작' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 'GPS 오류 확인 및 Fail-safe 동작' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '운전자 개입 확인 및 Fail-safe 동작' 기능에 대한 로직 활성화(Alive)/종료(Deadline)/시퀀스(Program Flow) 여부를 모니터링하고, 조치해야 한다. | ASIL C | Self_Driving_Agent 모듈 FMEA 결과 | Detection Action | |
| 상용차 자율주행 SW는 '센서 고장 확인 및 Fail-safe 동작' 기능에 대한 로직 활성화(Alive)/종료 | | | | |

[그림 42] SW 기능안전 요구사항 도출

## 마. SW 안전 설계 및 구현

### 1) SW 안전 아키텍처 설계

□ 초기 안전메카니즘 설계 및 구현

○ SW안전 메커니즘의 설계 이전의 시스템은 Fail Safe class에서 기본적인 lane 이탈, GPS 오류, 센서고장, 엑츄에이터 고장 등을 Vehicle_State_Machine class에서 운전자모드로 전환하는 단순한 형태로 구현

○ 기본적인 fail-safe 기능 중심으로 구현됨

- Self_Driving_Agent 패키지 내의 Fail_safe Class 에 5가지 고장에 대한 조치수단으로 운전자모드로 전환하는 Method를 정의

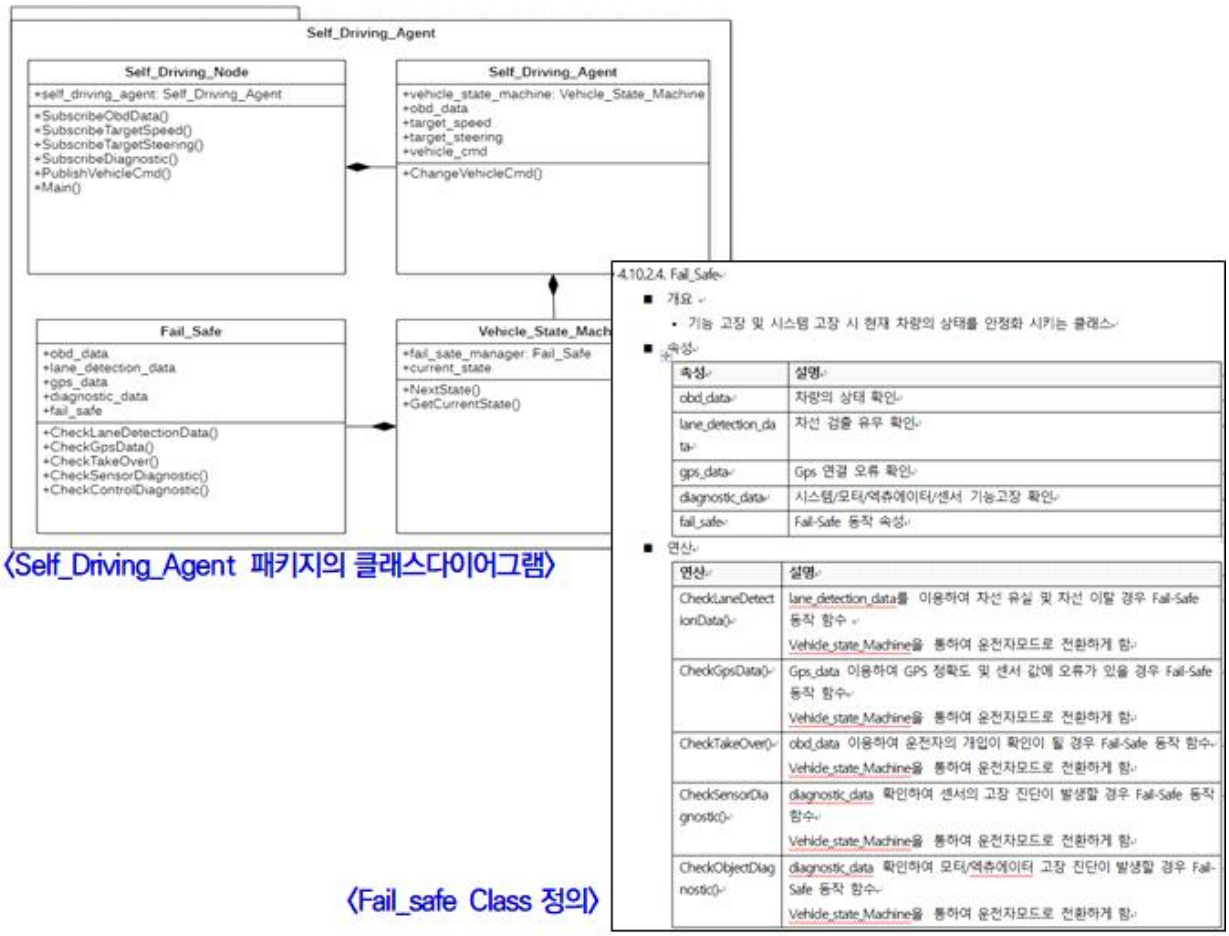
○ Safe-state가 운전자모드로의 전환을 기본으로 하고 있음

- 향후 FS (페일 세이프 기능)와 FD (페일 디그레이드 기능) 중심 설계 필요: 운전자모드로 전환하기 전에 기본적으로 Fail-degrade 상태를 safe-state로 정의하고 이에 도달할 수 있게 구현 필요

□ Fail-safe 기능 추가(일부)

○ 자율주행임시운행 허가 획득을 위한 SW 안전 아키텍처 설계 및 구현

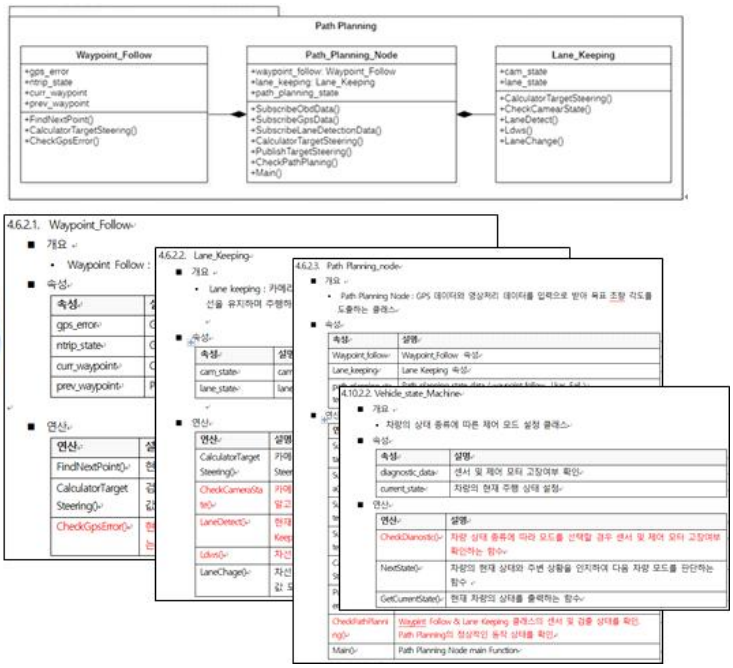
- 기존 5개의 안전기능 외에 자율주행 임시운행 허가를 위한 시험을 통과하기 위하여 필요한 안전기능(8개)을 추가로 설계/개발




[그림 43] 초기 Safe-Mechanism: Self_Driving_Agent 패키지 내의 Fail_safe Class 설명

〈JIAT 자율주행임시운행 허가 재인증 획득을 위한 안전기능 개발〉

| 시험 항목 | 기능 추가 필요 사항(필수요소) |
|---|---|
| 곡선로 곡선률에 따른 속도 감속기능 | 차량의 한계치 이상의 곡선로 발생시 감속기능 추가 |
| 차선이탈시 페일-세이프 | LDWS 자체 알림기능 추가<br>차선이탈 발생시 운전자 운전모드 변경 |
| 레이더/ 퓨전 정확도 | 퓨전 필터링 기능 추가 구현 |
| E-STOP 스위치 | 스위치박스에서 제어모듈 전원 차단 스위치 추가 |
| 카메라 이물질 감지 기능 | 비전 검출 기능 추가 |
| 제어 엑츄에이터 모니터링 기능 | 제어 엑츄에이터 모니터링 기능 추가 |
| 화물 적재시 제어 특성 변경 | 적재하중에 따른 제어 모드 기능 추가 |



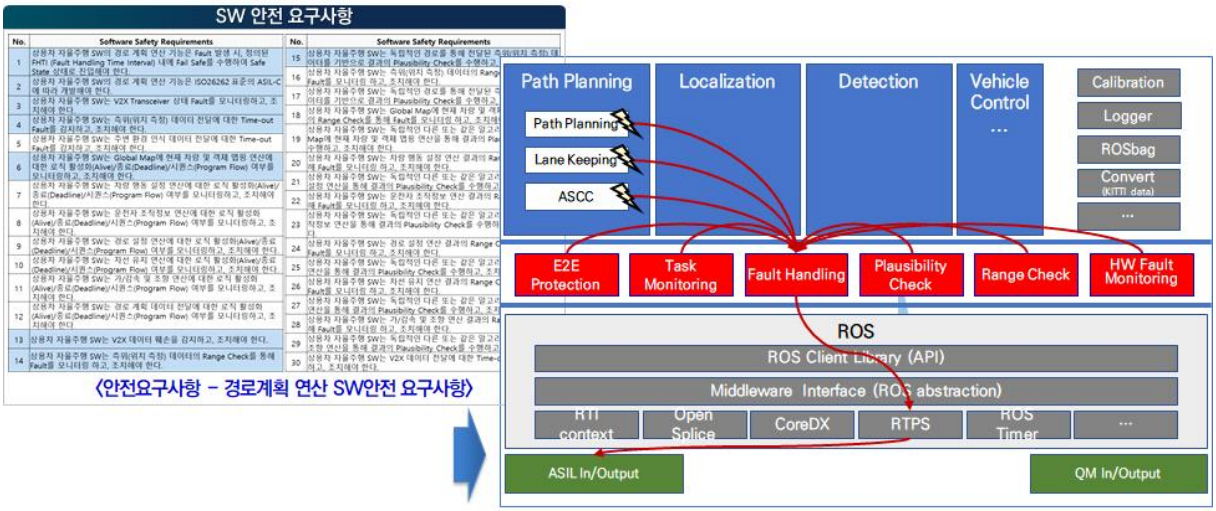

[그림 44] Fail-safe 기능 추가

□ SW 안전 아키텍처 설계 - 안전메카니즘 설계

○ 고속도로 자율주행 테스트를 위하여 추가되는 기능(HD-map 기능개선 등)을 포함하는 자율주행의 전체 시스템이 안전하다고 주장하기 위해 존재해야 하는 안전기능을 설계

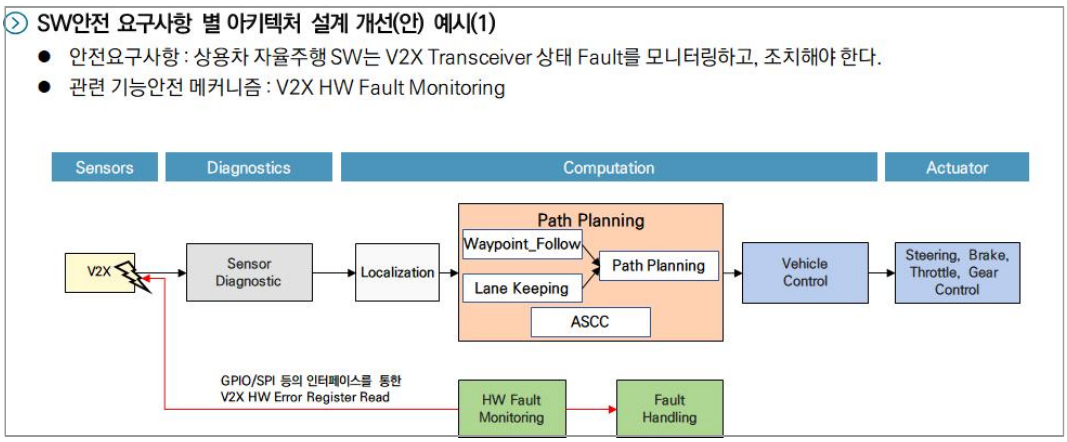
○ ROS2 SW 플랫폼 기반 상용차 시스템 SW에 대한 전체 모듈 식별

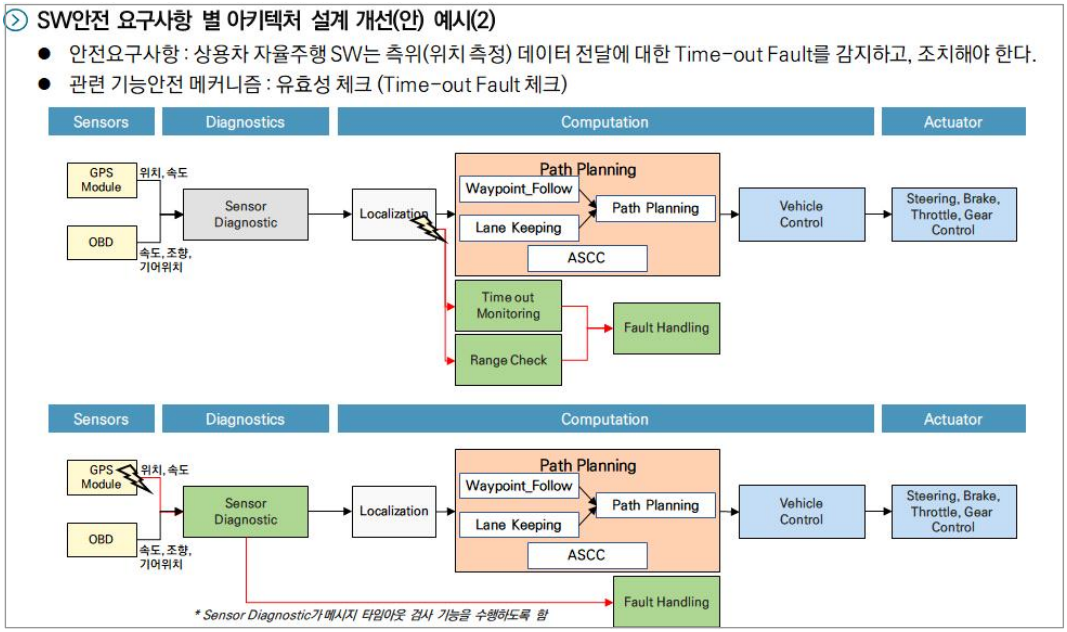
– 1차년도에 자율주행면허시험 획득을 위하여 추가 설계/개발한 안전기능을 포함하여 전반적인 안전메카니즘 재설계 및 개발

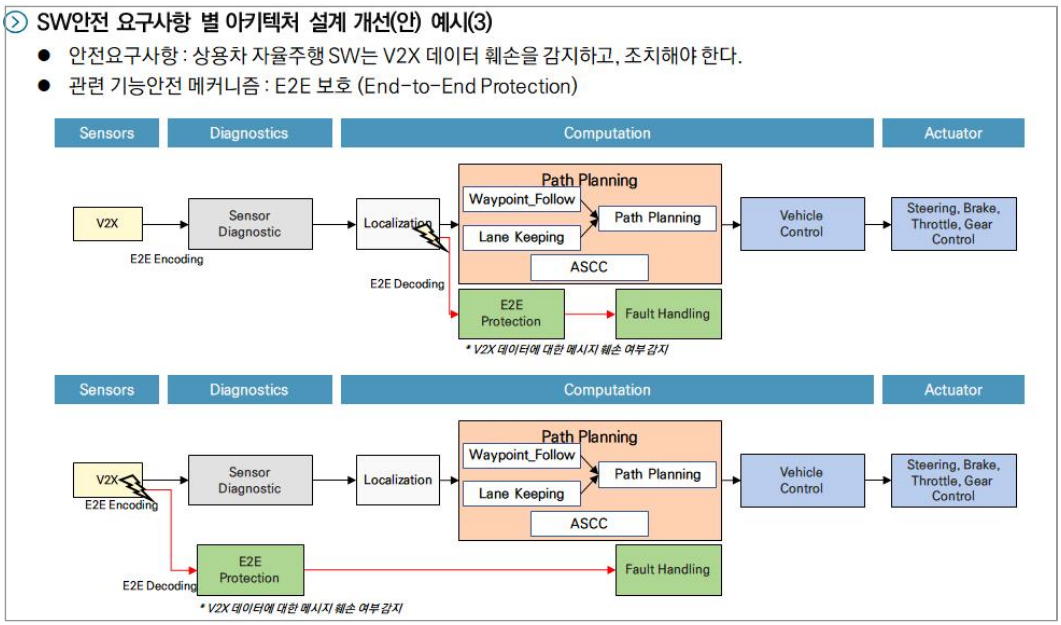



[그림 45] 각 안전요구사항에 대한 안전메카니즘 설계

– SW안전 요구사항 별 아키텍처 설계 개선(안) 중 일부를 예시하면 다음과 같음

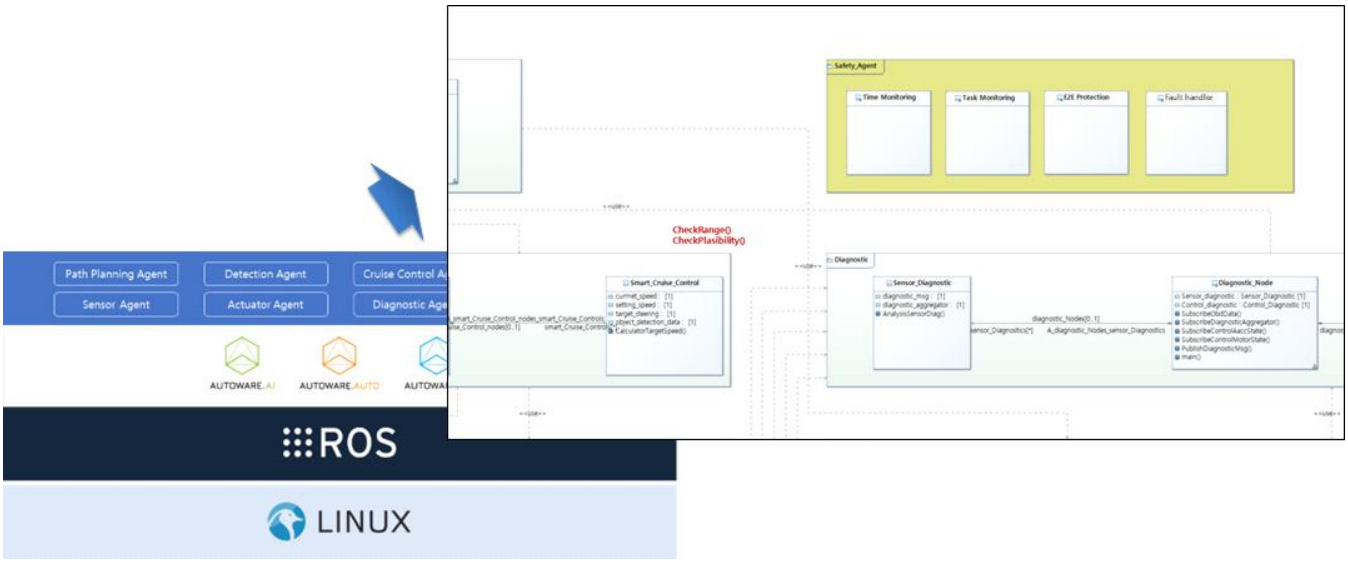





○ SW 아키텍처 보완

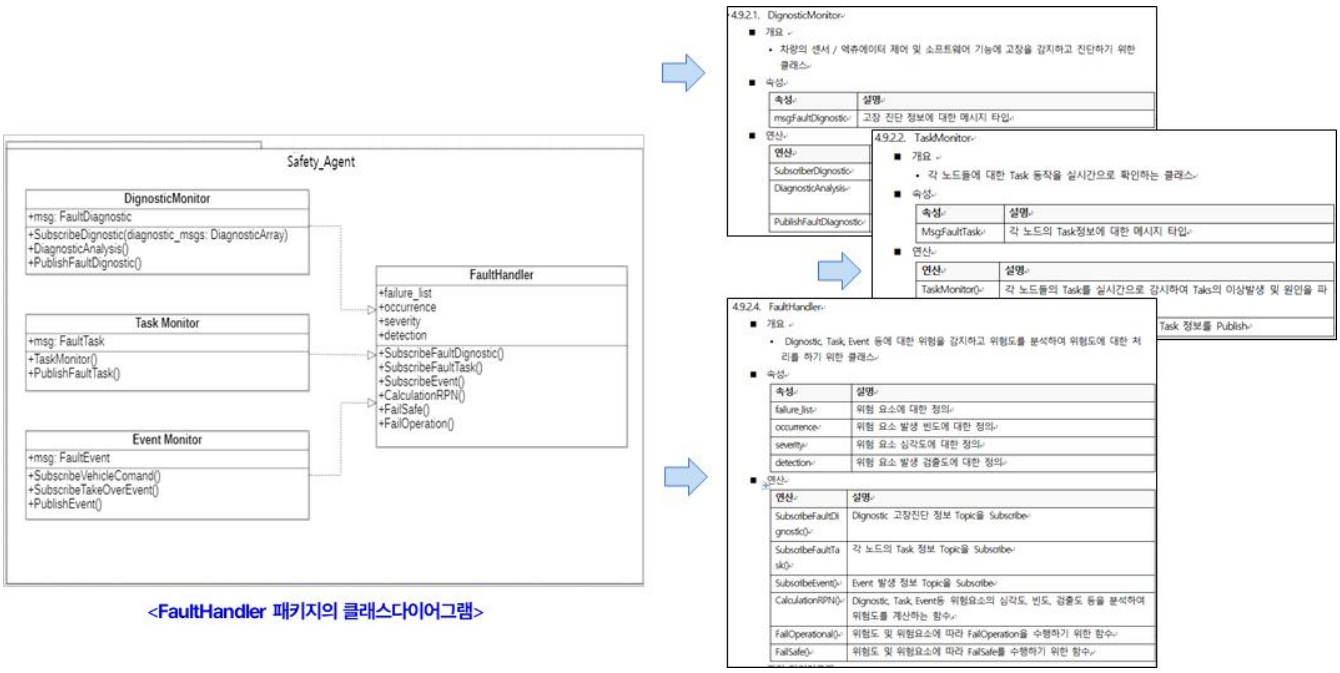
– 안전 분석 결과와 SW 기능안전 요구사항을 기반으로 SW 기능안전 메커니즘을 아키텍처에 반영




[그림 49] 아키텍처 보완

○ SW 안전 메카니즘 설계

- Safety_Agent 패키지를 설계
- 주요 클래스와 Method를 식별
- Real-time 성능을 고려하여 추가 모듈 상세설계 및 구현




[그림 50] SW안전 메카니즘 설계

### 2) SW 안전 설계 및 구현

□ 구현 및 시험 기술 적용

○ ISO26262 ASIL B 등급 요건 만족을 위한 인스펙션 수행

– SW 설계 및 코드 검토를 위한 인스펙션 교육 및 이행

– 동료검토 효율성을 시간당 2.1개의 결함을 찾을 수 있게 개선

○ 결함 예방을 위하여 ISO26262에서는 ASIL C등급에서 요구하는 SW 설계, 구현, 시험 단계에서 요건을 만족하는 다양한 개발 및 검증 방법을 지침에 반영하여 아래와 같이 단계적으로 확대 적용하고 있음

< SW 단위 설계/구현 단계의 SW안전기술 적용 >

| 단계 | 분류 | 방법 | ASIL A | ASIL B | ASIL C | ASIL D | 지침 및 도구 | 적용 여부 |
|---|---|---|---|---|---|---|---|---|
| 소프트웨어 단위 설계 및 구현 | 설계 검증 | 워크스루 | ++ | + | 0 | 0 | 산출물 검토(인스펙션) 지침 | 기적용 |
| | | 인스펙션 | + | ++ | ++ | ++ | 산출물 검토(인스펙션) 지침 | 1차년도 적용 |
| | | **제어흐름 분석** | **+** | **+** | **++** | **++** | **아키텍처 설계지침(설계 검증 보완) 정적분석 도구 및 활용 지침** | **2차년도 적용** |
| | | **데이터 흐름 분석** | **+** | **+** | **++** | **++** | **아키텍처 설계지침(설계 검증 보완) 정적분석 도구 및 활용 지침** | **2차년도 적용** |
| | 단위 테스트 케이스 추출 | 요구사항 분석 | ++ | ++ | ++ | ++ | **테스트케이스 작성 지침** | 1차년도 적용 |
| | | 동등분할 | + | ++ | ++ | ++ | | |
| | | 경계값 분석 | + | ++ | ++ | ++ | | |
| | | 에러 추측 | + | + | + | + | | |
| 소프트웨어 단위 구현 | 단위 수준의 구조적 커버리지 | 구문 커버리지 | ++ | ++ | + | + | **단위테스트 지침 테스트케이스 작성 지침(보완) 테스트지원도구(커버리지 분석 도구)** | **2차년도 적용 (단위테스트 도구 적용 및 커버리지 분석 예정)** |
| | | 분기커버리지 | + | ++ | ++ | ++ | | |
| | 단위 구현 검증 | 인스펙션 | + | ++ | ++ | ++ | 산출물 검토(인스펙션) 지침 | 1차년도 적용 |
| | | 정적 코드 분석 | + | ++ | ++ | ++ | **C, C++ 코딩 가이드(개선) 정적분석 도구 활용 지침** | **1차년도 적용 (일단 분석도구의 적용을 통한 분석 적용 후 일부 Rule 위반 사항은 코드 개선) 2차년도 확대 적용(코드개선)** |
| 소프트웨어 단위 검증 | 단위 검증 | 요구사항 기반 테스트 | ++ | ++ | ++ | ++ | 테스트 지침 | 1차년도 일부 적용 (테스트 지침 교육 강화) |
| | | 인터페이스 테스트 | ++ | ++ | ++ | ++ | | |
| | | 모델 기반 Back-to-back 테스트 | + | + | ++ | ++ | | 미적용 (Model 기반 개발 적용하지 못하고 있음) |

< SW 통합 및 통합시험 단계의 SW안전기술 적용 >

| 단계 | 분류 | 방법 | ASIL A | ASIL B | ASIL C | ASIL D | 지침 및 도구 | 적용 여부 |
|---|---|---|---|---|---|---|---|---|
| 소프트웨어 통합 및 통합 검증 | 통합 검증 | 요구사항 기반 테스트 | + + | + + | + + | + + | 테스트 지침 | 1차년도 일부 적용 (테스트 지침 교육 강화) |
| | | 인터페이스 테스트 | + + | + + | + + | + + | | |
| | | **결함 주입 테스트 (Fault injection test)** | **+** | **+** | **+ +** | **+ +** | **테스트 지침** | **2차년도 적용 (단계적으로 적용)** |
| | | **자원 사용 테스트 (Resource usage test)** | **+** | **+** | **+** | **+ +** | **테스트 지침 동적분석도구** | **2차년도 적용 (동적분석도구의 시범 적용)** |
| | | 모델 기반 Back-to-back 테스트 | + | + | + + | + + | | 미적용 (Model 기반 개발 적용하지 못하고 있음) |

[그림 51] SW 설계 및 구현 기법 적용

### 3) 코드 분석

□ ISO26262 요건 만족 및 사내 주요 문제점의 해결을 위한 단위/통합시험의 안전성 보장을 위한 정적분석 도구 활용 및 코드 개선

○ 국제 표준 코딩룰(MISRA C/C++)을 적용한 잠재적 결함 검출 및 소프트웨어 품질 측정을 위한 Metrics을 지원하는 정적분석 도구의 활용

○ 당해연도에는 코드개선에 초점

– 코드분석 결과, warining이 발생한 결과의 코드 개선 및 그 해결방안을 찾아 Rule Compliance Rate 개선

○ RESORT C/C++ 도구를 활용하여 코드 정적 분석

– 코드 구조 분석을 통한 코드 구조의 개선: 소스코드 복잡도 위반 46건, Call Level 위배 4건으로 줄임

– 10만건 이상이던 MISRA 코딩룰 위배 건수도 만건 이하로 줄임

도구 적용전

**RESORT Compliance Report**

| | |
|---|---|
| Project | JIAT |
| User | admin |
| Date | 2021-09-03 22:17 |
| Tool | RESORT for C++ |
| Ruleset | all |
| Rule Compliance Rate(%) | 33.18 |

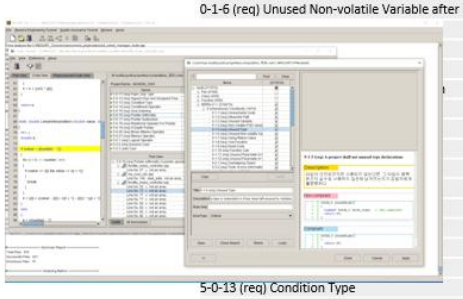
| Summary/Rule Name | Total | Severity |
|---|---|---|
| Number of Files | 465 | |
| Number of Defective Files | 463 | |
| Number of Functions | 1000 | |
| Lines of Code | 280577 | |
| Number of Rules | 217 | |
| Number of Rule Violations | 145 | |
| Number of Code Violations | 222340 | |
| Number of Critical Defects | 216239 | |
| Number of Major Defects | 6101 | |
| Cyclomatic Complexity | 125 | |
| Number of Call levels | 27 | |

도구 적용후

**RESORT Compliance Report**

| | |
|---|---|
| Project | JIAT 2.0 |
| User | admin |
| Date | 2021-01-03 19:54 |
| Tool | RESORT for C++ |
| Ruleset | auditproperties(computation_제외)rule |
| Rule Compliance Rate(%) | 48.92 |

| Summary/Rule Name | Total | Severity |
|---|---|---|
| Number of Files | 126 | |
| Number of Defective Files | 124 | |
| Number of Functions | 389 | |
| Lines of Code | 21218 | |
| Number of Rules | 217 | |
| Number of Rule Violations | 100 | |
| Number of Code Violations | 9275 | |
| Number of Critical Defects | 8986 | |
| Number of Major Defects | 289 | |
| Number of Minor Defects | 0 | |
| Cyclomatic Complexity | 46 | |



| 항목 | 수정전 | 수정후 |
|---|---|---|
| 0-1-10 (req) Function Call | 6688 | 390 |
| 0-1-3 (req) Unused Variable | 2483 | 99 |
| 0-1-4 (req) Non-Volatile POD Variable | 1774 | 91 |
| 0-1-5 (req) Unused Type | 5696 | 20 |
| 0-1-6 (req) Unused Non-volatile Variable after Definition | 4697 | 76 |
| | 52500 | 230 |
| | 4048 | 260 |
| | 1910 | 70 |
| | 1107 | 27 |
| | 2579 | 36 |
| | 16128 | 1316 |
| | 12988 | 1163 |
| | 19006 | 1458 |
| | 1289 | 17 |
| | 2975 | 49 |
| | 4989 | 74 |
| | 4467 | 42 |
| | 1577 | 127 |
| 5-0-13 (req) Condition Type | 4832 | 74 |
| 5-0-15 (req) Array Indexing | 5076 | 70 |
| 5-0-16 (req) Pointer Arithmetic | 3799 | 68 |
| 5-0-3 (req) Cvalue Expression | 1981 | 67 |
| 5-3-1 (req) Bool Not, Logical && Or Logical \|\| Operators | 1940 | 47 |
| 6-4-1 (req) If Statement | 3305 | 140 |
| 6-5-6 (req) Loop Control Variable | 1943 | 14 |
| 6-6-1 (req) Goto Statement | 1231 | 0 |
| 6-6-5 (req) One Exit Point | 3228 | 173 |
| 7-1-1 (req) Const Variable | 10590 | 348 |
| 7-3-1 (req) Global Namespace | 9769 | 833 |
| 8-0-1 (req) Single Declarator | 3098 | 18 |
| 8-4-3 (req) Explicit Return Statement | 1221 | 49 |

[그림 52] 정적분석도구 활용 및 코드 개선

## 2. 상용차 자율주행(Lv3) SW 고도화

### 가. 자율주행차량 S/W 개발

#### 1) 자율주행 기반 데이터 처리 및 분석 시스템 개발

□ ROS2 기반 자율주행(Lv3) 연산처리장치(기보유기술) 기반으로, 자율주행 제어 시스템을 개발하고, Fail Safe 기능 등의 기능안전 SW 추가 적용

○ HD Map을 활용한 자율주행 기능 강화를 위한 Autoware Platform 도입 및 Autoware 기반 SW아키텍처 재설계

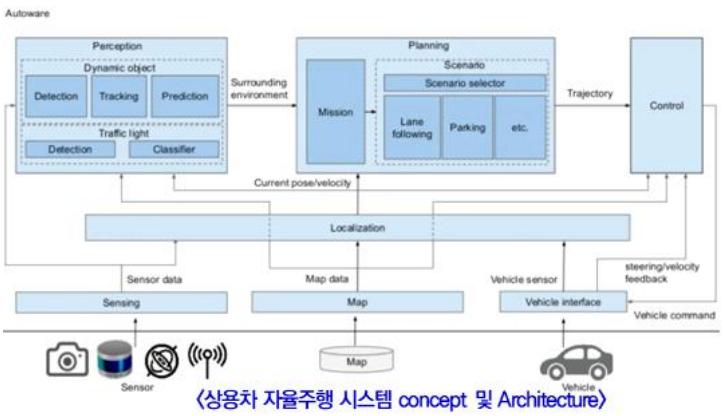
– Autoware 플랫폼 도입 및 SW 고도화

- ROS 기반의 자율주행 플랫폼으로 도입 및 재개발
- 센서 데이터를 통해 시뮬레이션이 가능
- 개별의 모듈화로 되어 있어 구조적으로 이점이 있음
- HD map 기능을 고도화

○ 안전기술 적용: Fail Safe 기능 SW 고도화, Fail Degrade 기능 추가

○ 1차년도 기술성 검증결과와 안전분석 결과를 반영하여 SW 고도화 : ASIL C 등급에 따른 개발(설계/코딩/테스트) 기법 적용

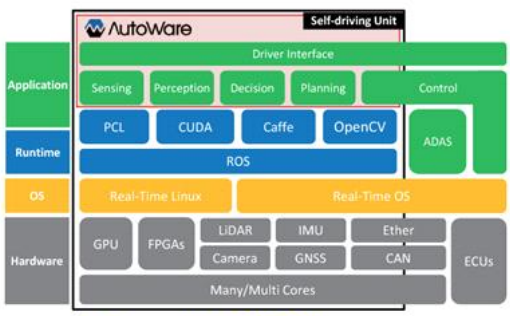
*** 자세한 자율주행 차량 SW 기능 및 구조는 기업 비밀 보호를 위하여 생략**




〈상용차 자율주행 시스템 concept 및 Architecture〉

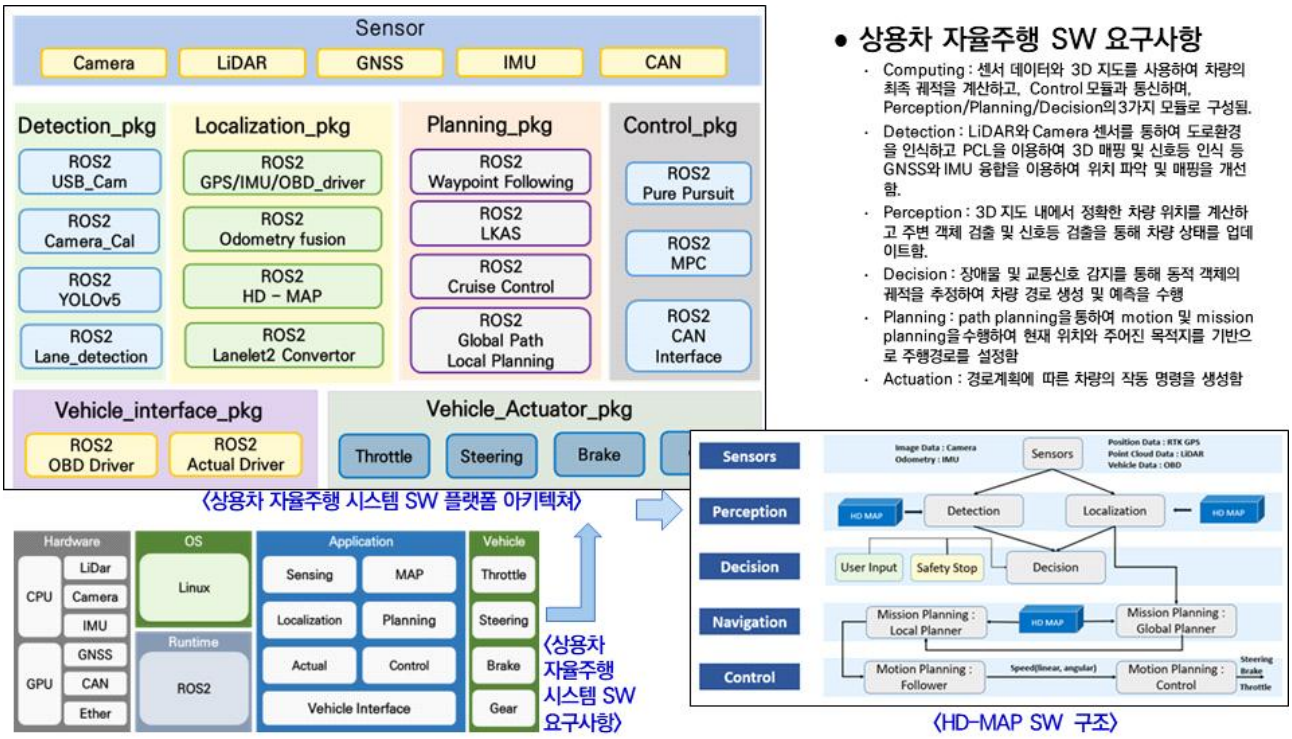
- Autoware 시스템 SW 요건
  - PCL/ CUDA/ Caffe/ OpenCV등의 tools를 사용하여 Detection, Sensing, Decision, Planning, Control등 소프트웨어 패키지를 적용
  - PCL은 Pointcloud library 로서, 데이터 시각화와 3D 매핑에 사용되며, Caffe는 표현, 속도, 모듈성을 염두에 두고 설계된 딥러닝 프레임워크임
  - CUDA는 프로그래밍 프레임워크 GPU 컴퓨팅 작업을 수행하고 자율주행과 관련된 연산의 집약적 작업을 처리하는 데에 사용됨
  - OpenCV는 이미지 저리를 위한 컴퓨터 비전 라이브러리임

- Autoware 플랫폼 필요성
  - ROS 기반의 자율주행 플랫폼으로 개발되어 Handling 가능하며, 센서 데이터를 통해 시뮬레이션이 가능
  - 각 센서들의 Input 데이터를 처리하는데 있어 개별의 모듈화로 되어 있어 구조적으로 이점이 있음
  - HD map 기능을 고도화 하기 위한 Open source 제공




〈Autoware 시스템 SW 요건〉

[그림 53] SW고도화 및 플랫폼 도입




● 상용차 자율주행 SW 요구사항

- Computing : 센서 데이터와 3D 지도를 사용하여 차량의 최적 궤적을 계산하고, Control 모듈과 통신하며, Perception/Planning/Decision의 3가지 모듈로 구성됨.
- Detection : LiDAR와 Camera 센서를 통하여 도로환경을 인식하고 PCL을 이용하여 3D 매핑 및 신호등 인식 등 GNSS와 IMU 융합을 이용하여 위치 파악 및 매핑을 개선함.
- Perception : 3D 지도 내에서 정확한 차량 위치를 계산하고 주변 객체 검출 및 신호등 검출을 통해 차량 상태를 업데이트함.
- Decision : 장애물 및 교통신호 감지를 통해 동적 객체의 궤적을 추정하여 차량 경로 생성 및 예측을 수행
- Planning : path planning을 통하여 motion 및 mission planning을 수행하여 현재 위치와 주어진 목적지를 기반으로 주행경로를 설정함
- Actuation : 경로계획에 따른 차량의 작동 명령을 생성함

〈상용차 자율주행 시스템 SW 플랫폼 아키텍처〉

〈상용차 자율주행 시스템 SW 요구사항〉

〈HD-MAP SW 구조〉

[그림 54] 사용차 자율주행 플랫폼 아키텍처 및 SW 요구사항

### 2) 자율주행 기반 상용차량 제어 시스템 개발

□ 자율주행차량 S/W 요구사항 정의 및 설계

○ 요구사항 정의 : 시범운행 현장분석, 요구사항 분석, 시스템 구성 정의

○ 시스템 설계/제작 : 차량 Data 분석, 가공물 설계/제작, 제어 모듈 설계/제작

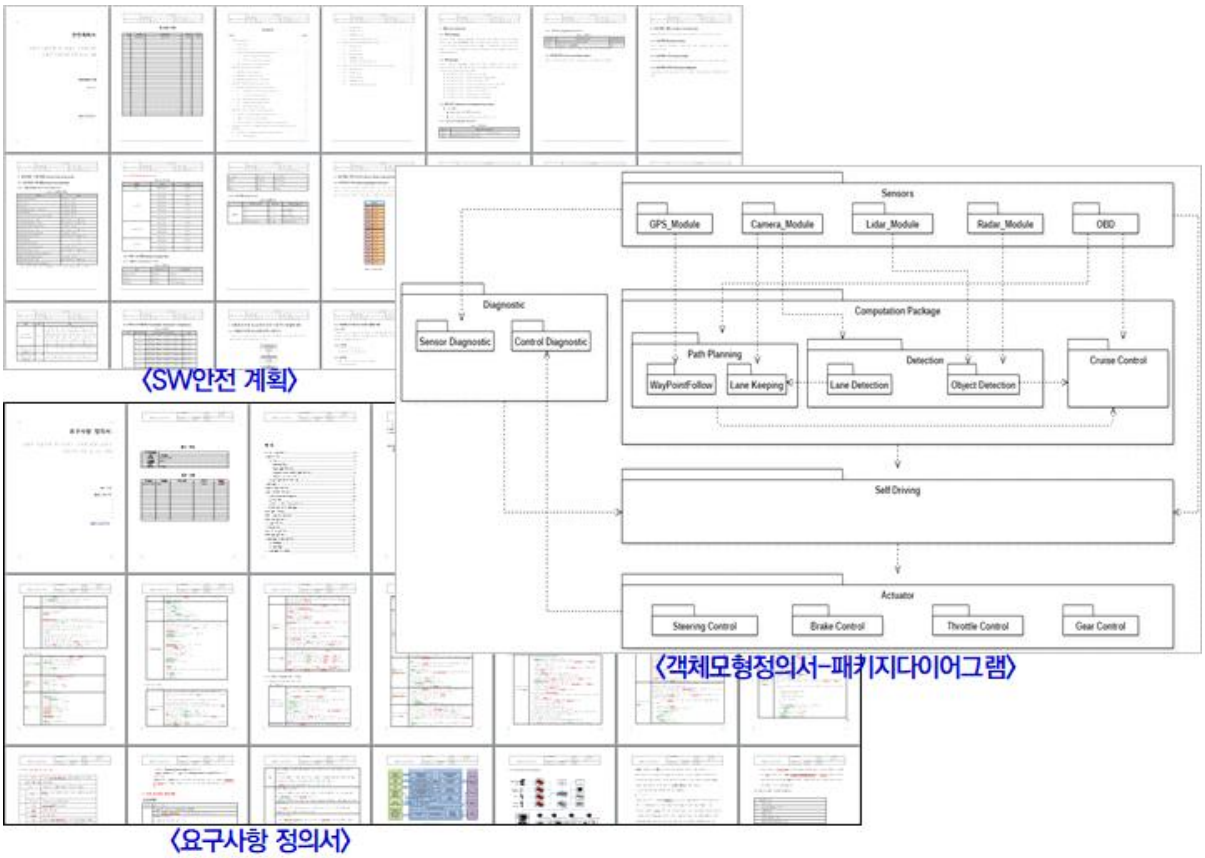
○ SW 설계 및 개발 : 운영환경 Data 취득/분석/학습, 센서/제어 SW 구조 설계, 통신/보안 SW 구조 설계, fail safe /take over 기능 정의, 객체인지/센서융합 개발

□ Lv3. 상용차 자율주행을 위한 SW 요구사항 및 SW 아키텍처를 정의하고, SW 모듈을 설계

○ 제어 모듈 독립화 및 Fail Safe / Take over 기능 SW 적용

○ SW아키텍처는 ASIL C 수준의 설계 원칙 및 SW안전을 일부 고려함

*** 자세한 자율주행 차량의 제어 시스템 기능 및 구조는 기업 비밀 보호를 위하여 생략**




[그림 55] Lv.3 자율주행 SW 요구사항 정의 및 설계 산출물

## 나. V2X 통신 HMI 고도화

□ V2X(Vehicle to Everything) 시스템은 실시간 정보 교환을 목적으로 진행하며, 차량(V2V), 인프라(V2I), Cloud(V2C) 등 모든 통신을 포함

○ 각각의 센서들의 융합을 이용하여 도출하는 정보를 바탕으로 IEEE 802.11p의 WAVE 통신을 이용하여 시스템을 구현

○ V2X 통신 (인프라/Lab/PG/SMTB/관제) 장비 연동 변경에 따른 지원

○ 또한, V2X 시스템을 이용하여 자율주행 제어의 검증을 지원

*** 자세한 개발 내용은 기업 비밀 보호를 위하여 생략**

## 3. 상용차 자율주행 테스트베드 내 기술성 검증

### 가. 새만금지역 상용차 자율주행 테스트베드 구축을 위한 통합 시험

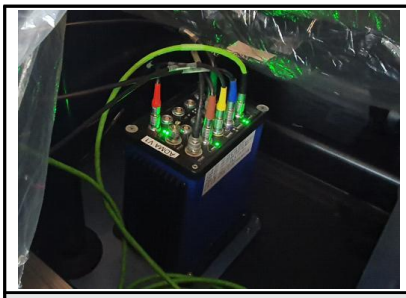
□ 실시간 통합관제평가시스템 구축을 위한 시스템 통합 검증

○ 시험 차량 플랫폼 및 시험 장비 구축

○ SMTB/SMPG 인프라내 주행 기본 기능 주행시나리오 운행을 위한 시험 코스 및 환경 조건 정의

○ CACC 기반 주행시스템의 기본 운행 기능 평가항목에 대한 시험 운용 및 결과 데이터 분석

| | | |
|---|---|---|
|  |  |  |
| 전면 | 측면 | 후면 |

| | | |
|---|---|---|
|  |  |  |
| GPS/INS | DAQ | 고속카메라 |
|  |  |  |
| 카메라/마이크로폰 | 충돌더미(차량타겟) | 중량 측정사진 |

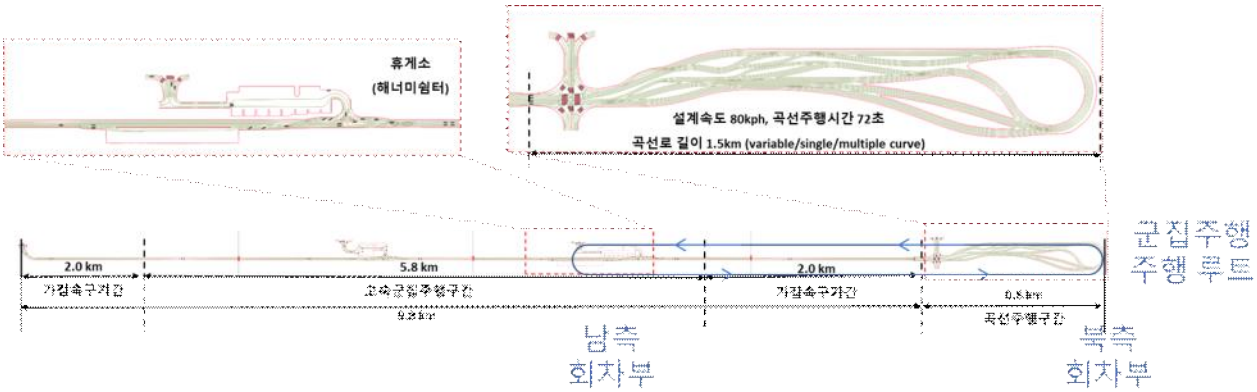
### 나. 군집자율주행 SW 의 테스트베드 내 실증 테스트

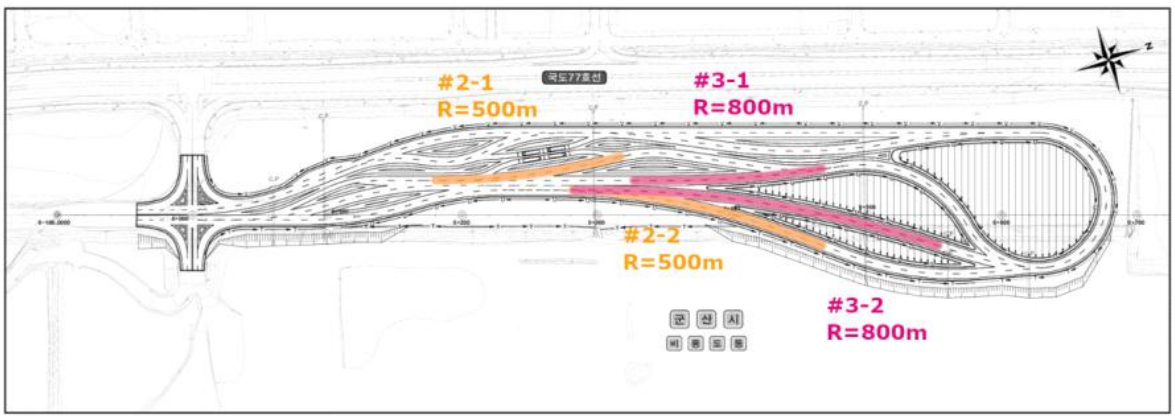
□ SMTB 새만금 자율주행시험도로 환경에서 CACC 기반 자율협력주행 제어기가 적용된 차량을 활용하여 형성／합류／유지／이탈 시나리오에 대한 실차 시험 주행을 실시함

□

□ 시험 환경

| 새만금 주행시험장 | 종합시험로 |
| --- | --- |
|  |  |




□ 평가 시나리오

| 시나리오 구분 | 세부 시험 시나리오 설명 | 비고 |
|---|---|---|
| 상태 변경 | LV주행속도 30kph, LV-FV 차간 간격 변경 (Time gap 1.5s -> 1.0s) | |
| | LV주행속도 40kph, LV-FV 차간 간격 변경 (Time gap 1.5s -> 1.0s) | |
| | LV주행속도 40kph, LV-FV 차간 간격 변경 (Time gap 1.0s -> 1.5s) | |
| | LV 주행속도 변경 (30 -> 40kph) | |
| | LV 주행속도 변경 (40 -> 50kph) | |
| | LV 주행속도 변경 (50 -> 40kph) | |
| | LV 주행속도 변경 (40 -> 30kph) | |
| 추종모드 유지 | 직선로 추종모드 유지 | |
| | 곡선로 추종모드 유지 (R500) | |
| | 곡선로 추종모드 유지 (R100) | |
| 주행 환경 | 외부 차량 주행 컷인 | |

## □ 시험 결과 예시

● 차간 간격 감소 시험 (1.5 sec -> 1.0 sec)

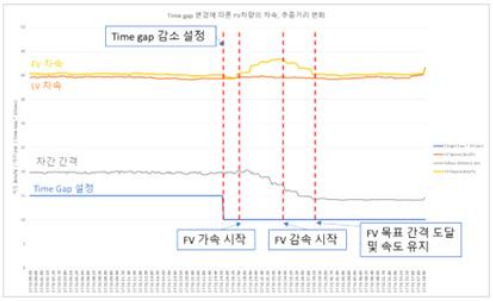
✓차속 40kph 유지 환경에서 LV-FV간 차간 간격을 1.5초에서 1.0초로 감소 조정하는 기능에 대한 시험 수행

✓시험 목표 : 정상 상태 조건에서 기존 대형의 간격 변동 안정성 검증

✓추종모드의 LV 차량에서 차간 간격 설정을 1.5sec에서 1.0sec로 변경하여, 차간 간격 조정의 기능 및 간격 조정에 대한 성능 평가



〈FV 차량 HMI 표출 이미지 (차간 간격 감소) 〉




〈LV-FV 차간간격 감소시의 차속 , 차간간격 변화 그래프〉

[그림 69] 시험 결과 예시

# Ⅳ. SW품질개선 노력 및 내재화 수준

## 1. SW품질 개선 의지 및 노력

□ SW품질관리 의지 및 노력

○ 전사적 품질관리 및 SW 공학 적용 의지

– 당사는 전사적인 품질관리 추진으로 2018년에 ISO 9001 인증과 CMMI Level 3 인증을 획득하였음

– 현재는 ISO26262 ASIL C 등급 달성 등을 목표로 전사적으로 SW공학기술을 지속적으로 적용하고 있음

○ 품질관리 및 프로세스 개선 조직 수립 의지

– 당사는 품질관리 담당자를 충원하였으며, 개발과 양산의 품질관리를 위한 인력과 당사의 EPG(Engineering Process Group) 인력을 양성하고자 노력하고 있음

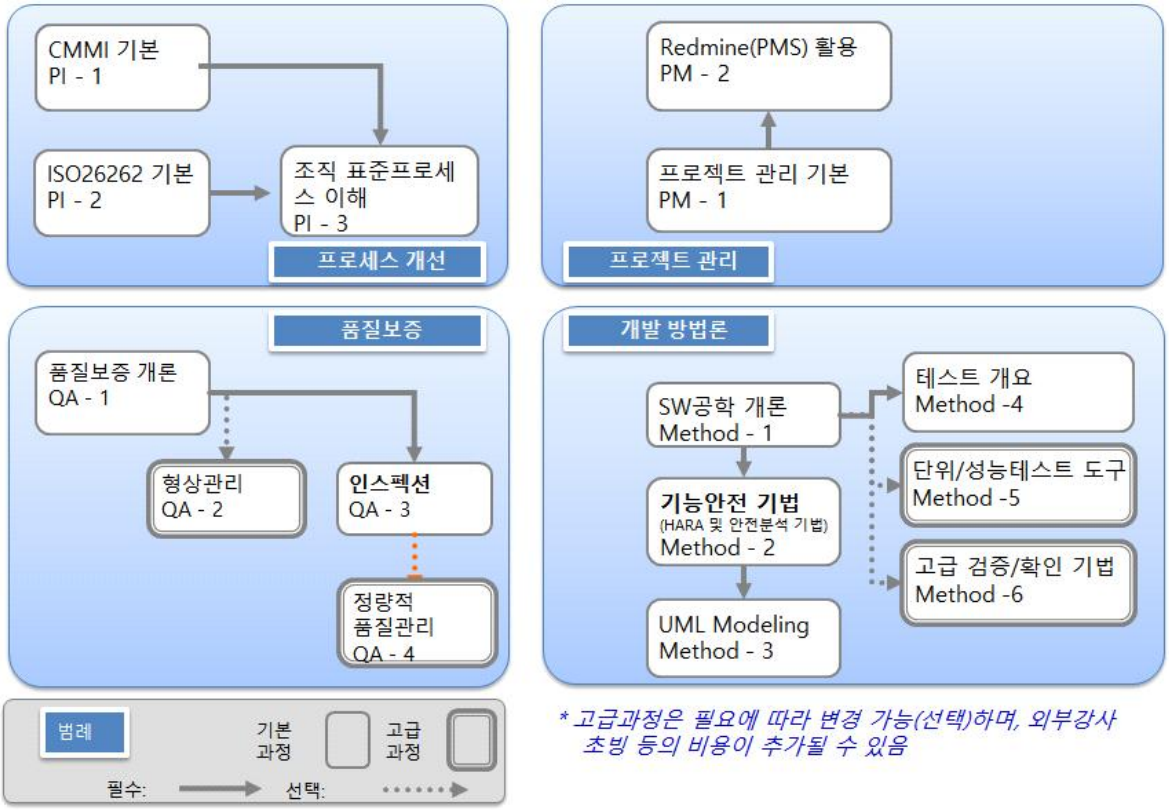
○ SW공학 관련 교육 실시

– 프로젝트 수행 역할자의 역량제고 및 구축된 표준 프로세스의 원활한 사용을 위해 프로젝트수행 및 프로세스 운영과정에서 요구되는 각종 기술들을 단기간에 습득할 수 있도록 다양한 역량강화 교육을 실시함

- 품질관리 조직 및 품질보증 담당자(QA) 교육, CMMI 기본, ISO26262 기본 교육

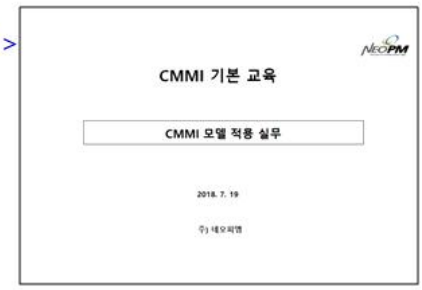
– 실무자들의 소프트웨어공학에 대한 인식을 높이고 관련 지식 및 경험 수준을 향상시키기 위하여 프로젝트 진행 중 지속적인 교육 및 세미나, 토론 활동을 통하여 전문성을 배양하도록 함

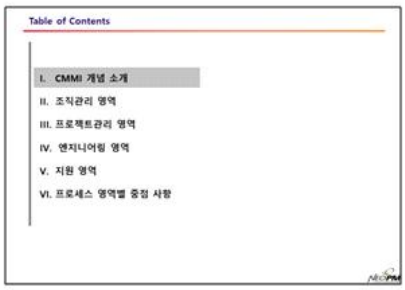
- SW공학 개론 및 자사 방법론 기본 교육
- ISO26262를 지원하는 HARA 분석 및 안전분석 기법 교육
- 분석/설계 모델링(UML 모델링 교육) 및 인스펙션, 테스트 기본 교육
- 프로젝트 관리 교육(PM 및 개발자 대상)



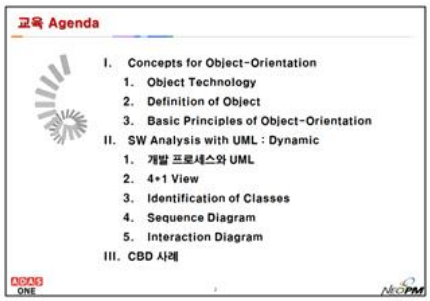

[그림 70] SW공학 기술 교육 내용

< CMMI 기본 교육- module 2 >








<SW공학 교육>

[그림 71] 주요 SW공학기술 교육 교재 사례

## 2. SW공학기술 활용 및 내재화 수준

□ 당사는 2018년에 CMMI(시스템 개발능력 인증) Level 3 요건을 만족하는지를 객관적으로 입증하기 위하여 CMMI Level 3 심사를 수행하였고, **CMMI Level 3 인증을 획득**하여 전사적인 프로세스 능력과 품질능력을 객관적으로 인증 받았음

○ 수립된 방법론을 실제로 이행하도록 시범적용 프로젝트에 적용하고 SW개발 활동을 수행하였음

○ 2017년 이후 ISO9001 품질경영시스템 인증을 유지하고 있음

○ 2020년에 **SW프로세스 품질인증(SP인증) 2등급 인증을 획득**함

□ 현재 개념단계, 시스템 개발단계, SW개발의 요구분석/설계 과정에서 ISO26262 ASIL B 등급 요건을 만족하는 개발을 위한 **SW공학기법은 대부분 적용하고 있음**

○ HARA 분석 등 ISO26262 요건을 만족하기 위한 분석. 설계 단계의 안전분석에 대한 Practices를 적용하고 있어, 향후 고객 및 투자사들의 요구사항을 만족할 수 있을 것으로 보임

□ ISO26262의 ASIL C 등급을 만족하는 SW공학 원칙 및 기법을 단계적으로 적용하고 있음

○ SW의 설계 단계에서부터 시험단계까지 ISO26262의 ASIL C 등급을 만족하는 SW공학기법을 단계적으로 적용하고자, 구체적 기법 가이드 및 산출물 템플리트를 수립함

○ 올해 구현 및 테스트 단계에서 필수적으로 요구하는 기법을 단계적으로 적용하고 있음

## 3. SW공학기술 적용에 따른 개선효과

□ SW공학기술 적용에 따른 효과성

○ ISO26262를 만족하는 SW공학기술의 적용으로, 안전분석과 안전메카니즘 설계 및 구현, 상용차 자율주행 실증 테스트를 통하여, 자율주행차의 안전을 확보하고, 2차의 면허시험을 거쳐 **자율주행 임시운행 면허**를 획득함


[그림 72] 자율주행 임시운행 면허 획득

○ 1차년도에 기능안전 프로세스 및 ISO26262 ASIL B 등급을 만족하는 SW공학기술의 적용으로 자율주행 상용차의 성능과 안전성을 확보하고, SW품질 역량을 강화하였음

- 자율주행차량 SW개발 관련하여 자율주행 운행시 자율주행 시스템의 응답율, Fail-safe 기능 만족도, 긴급제동브레이크 시스템 성능 평가 만족도를 100% 달성하여 자율주행 임시운행면허를 획득
- fail-safe를 위한 기능안전 메커니즘 도입 건수도 10개로 100% 달성
- SW공학기술 도입 지표측면에서는, 동료검토 효율성은 시간당 결함발견 2.1개로 100% 달성하여 동료검토 효율성을 높였음
- MISRA C/C++ Rule에 적합한 소스코드의 개발 측면에서는, 소스코드 복잡도 위반 46건, Call Level 위배 4건으로 당초 목표를 100% 달성하여 소스코드의 품질을 개선하였음

○ 2차년도에 SW공학기술의 적용으로 강화하고자 하는 자율주행 상용차의 성능과 안전성을 확보, SW품질 역량 강화 성과목표는 아래와 같으며, 이를 달성할 수 있을 것으로 보임

- 전년도의 기능안전 메카니즘의 구현에 더하여 자율주행 SW 플랫폼을 전면 개편하고, 고속도로 자율주행을 고려한 안전분석을 통하여 10개 이상의 기능안전 메카니즘을 추가 구현 예정임
- ISO26262의 요건은 ASIL B 등급에서 요구하는 SW공학기법의 적용을 80% 이상 적용하고 내재화 될 것으로 기대됨

| 구분 | 성과지표 | 성과목표 | 비고(설명 또는 산출방법) |
|---|---|---|---|
| ASIL C 등급 SW기능안전기술 적용 | 기능안전 메카니즘 추가 도입 건수 | 5개 이상 | HARA, FMEA를 통한 SW 기능안전요구사항과 이를 만족하기 위하여 추가 설계/구현된 안전조치 SW 기능(모듈 또는 함수) 수 |
| | SW 개발 및 시험 지침 추가 도입 건수 | 5개 이상 | ASIL C등급에서 요구하는 설계 및 구현 단계의 SW공학원칙을 만족하기 위한 지침 추가 : SW 시험(결함주입시험 등) 지침, 시험케이스 설계 지침, 코드개선 지침 등 |
| | ISO 26262 요건 항목 만족율 | 80% 이상 | 자동차 기능안전성 국제표준인 ISO 26262의 요건을 만족하는 정도<br>- TUV-SUD 등 외부전문기관의 testing and Assessment service에 의한 인증 획득은 현실적으로 1년 이내에 달성하기 어려워, 자체적인 체크리스트를 기반으로 평가를 수행<br>- 총 요건 항목에 대한 만족율을 계산 (요건 항목의 개수는 HW, SW에 할당/분해된 형태에 따라 달라질 수 있음) |
| 상용차 자율주행 (Lv3) SW 고도화 | 고속도로 자율주행 테스트 시 자율주행시스템의 응답 | 100% | 고속도로 실주행 테스트를 통한 운행 상황 시나리오에 따른 시스템의 응답을 테스트 |
| | 신시장 개척을 통한 매출증대기여 | 10억원 | 국내 매출 기여액(국내 사업수주 금액 증가분 x 50%) |

○ 개발방법론 및 관리방법론, SW공학기술의 적용으로 SW개발의 생산성과 품질이 높아지고 있음

– 기본적으로 SW공학기술의 도입으로 개발을 위한 주요 Task와 WBS를 수립하고, 이를 기반으로 진척도를 확인하여 기존에 6개월 이상 지연되던 프로젝트 진척도 상의 문제는 현재 8% 이내의 편차로 줄어듬

– 개발자가 개발을 위하여 공유하여야 할 요구사항, 아키텍처, 주요 설계사항을 문서화 하여 공유하고, 설계된 개발 모듈 리스트를 통하여 개발관리를 함으로써 개발의 진행상태가 블랙박스이던 문제점을 개선하고 있음

– 요구사항을 기반으로 운영환경 및 운영시나리오를 고려한 테스트 요구사항 및 테스트 시나리오 준비로 테스트를 강화하여 출시 후 결함 감소 예상

# [첨부] 관련 문의처

| 소속기관 | 문 의 처 | | | 담당 내용 |
|---|---|---|---|---|
| | 당당자명/직위 | 연락처 | 이메일 | |
| (주)스카이오토넷 | 이석수/상무이사 | 070-4616-3121 | seoksoo.lee@skyautonet.com | 상용차 자율주행(Lv3) SW 고도화 |
| (주)네오피엠 | 안유환/대표이사 | 010-2464-5441 | ywahn@neopm.co.kr | ASIL C 등급 SW기능안전 프로세스 및 기술 적용 |
| 자동차융합기술원 | 전재석/전임연구원 | 010-2816-4569 | jsjeon@jiat.re.kr | 상용차 자율주행 테스트베드 내 기술성 검증 |