StudyTemplate

소프트웨어 유닛 시험 계획서

**Version < x.x >**

**<YYYY/MM/DD>**

문서 목적

**<문서의 목적을 기술>**

(예시) 이 문서는 VW AQ 프로젝트의 SW 검증 계획이다.

개발된 SW가 SW 요건 및 안전 요구사항을 충족하는지 검증하기 위해 SW 레벨 검증 계획을 수립하는 데 사용된다. SW unit 검증 단계에서는 개발된 unit 기능이 SW 상세설계를 충족하고 원치 않는 기능을 포함하지 않는지 검증하기 위해 수립된 Test방법 및 Test사례 설계기법 적용계획을 수립하고, 정적분석 계획을 수립하여 개발된 코드의 신뢰성을 확인한다. 또한, 검증된 unit 기능을 통합하여 SW 컴포넌트를 구축할 때, SW 인터페이스가 SW 아키텍처 설계에 부합하는지 검증하기 위한 시험 방법 및 테스트케이스 설계 방식의 적용 계획을 수립하고, 임베디드 SW에 대해 SW 아키텍처 설계에 부합하는지와 전체 component가 SW 요구사항 및 안전 요구사항을 충족하는지 검증하기 위한 시험 환경 구축 계획을 수립한다.

- 목 차 -

[1. 개요 1](#_Toc86840152)

[**1.1. 목적** 1](#_Toc86840153)

[**1.2. 유닛 테스트 목표** 1](#_Toc86840154)

[2. 테스트 방법론 1](#_Toc86840155)

[**2.1. 역할과 책임 (Roles and Responsibility)** 1](#_Toc86840156)

[**2.2. 테스트 활동 방법** 1](#_Toc86840157)

[**2.3. 테스트 설계 기법** 2](#_Toc86840158)

[**2.4. 테스트 산출물** 3](#_Toc86840159)

[**2.5. 테스트 위배 관리** 3](#_Toc86840160)

[3. 소프트웨어 유닛 테스트 4](#_Toc86840161)

[**3.1. 유닛 테스트 전략** 4](#_Toc86840162)

[**3.2. 유닛 테스트 기준** 4](#_Toc86840163)

[**3.2.1. 소프트웨어 유닛 테스트** 5](#_Toc86840164)

[**3.2.2. 테스트 Entry/Suspension/Resumption 기준** 5](#_Toc86840165)

[**3.2.3. 테스트 완료 기준** 6](#_Toc86840166)

[**3.3. 유닛 테스트 환경** 6](#_Toc86840167)

[**3.3.1. 테스트 환경 개요** 6](#_Toc86840168)

[**3.3.2. 테스트 데이터 요구사항** 6](#_Toc86840169)

[**3.3.3. 테스트 환경 요구사항** 6](#_Toc86840170)

2. **개요**
   1. **목적**

<일반 목적에 대해서 기술>

(예) SW 상세 설계와 비교하여 SW unit이 의도된 대로 기능하고 의도하지 않은 기능이 없는지 확인한다. 또한 SW code 신뢰성을 위해 정적 분석을 수행한다.

* 1. **유닛 시험 목표**

<유닛 시험 목적에 대해 기술>

(예) 소프트웨어 유닛 시험의 목적으로 시험 목표는 다음을 준수해야 합니다. 이러한 목표는 PMBook에서 가져온 것입니다.

- 아키텍처 인터페이스/동적 동작 및 인터페이스는 테스트케이스에서 다룹니다.

- SWAD(행동 범주) 및 SwUTC 추적 가능성의

시험 완료 목표는 이 문서(SwUTC)의 시험 완료 기준(x.x.x)에 설명되어 있습니다.

Table 1 시험 기준

|  |  |  |
| --- | --- | --- |
| Milestones | Targets | Remarks |
| SBS1 | - Behavior of SwDD and SwUTC Traceability : 100%  - Interface of SwDD and SwUTC Traceability : 100% | V300 |
| SBS2.1 | - Behavior of SwDD and SwUTC Traceability : 100%  - Interface of SwDD and SwUTC Traceability : 100% | V400 |
|  |  |  |

1. **시험 방법론**
   1. **역할과 책임 (Roles and Responsibility)**

이 프로젝트에 대한 역할과 책임은 RnR 회의 보고(RnR)에 문서화되어 있습니다.

* 1. **시험 활동 방법**

(예) 소프트웨어 유닛 테스트케이스는 다음 설계 방법에서 파생되어야 하며 테스트케이스 설계 방법은 소프트웨어 유닛 테스트케이스 설계 전에 정의되어야 합니다.

Table 2 테스트케이스 디자인 방법

|  |  |  |
| --- | --- | --- |
| Test Case Methods | Description of Method | Remarks |
| Analysis of requirements (Architecture Design) | Tests directly from the architecture design. One requirement needs one or more tests for sufficient verification. | Applied for sbs1 |
| Analysis of external and internal interface | External interfaces which are connected to the external system shall be checked  Internal interfaces in the A/T system shall be checked |  |
|  |  |  |
|  |  |  |

Table 3 시험 목표와 방법

|  |  |  |
| --- | --- | --- |
| Test Case Methods | Description of Method | Remarks |
| Analysis of requirements (Architecture Design) | Verification of functional performance, accuracy and timing against system requirements (system requirements, system design, etc.) | Applied for sbs1 |
| Analysis of external and internal interface | Verify the internal interface in the system integration test and the external interface in the system test |  |
|  |  |  |

* 1. **시험 설계 기법**

(예제) (1) Specification-based test design

- Equivalence partitioning

- Boundary value analysis

- Decision table testing

- Scenario testing

(2) Structure based test design

- Statement testing

- Branch testing

- Decision testing

* 1. **시험 산출물**

Table 4 Test Related Documents

|  |  |
| --- | --- |
| Work product | Time of Writing / Updating |
| Static Analysis Report | After completion of static analysis (generated from static analysis tool) |
| SWUTC | After developing/updating of SW unit design |
| SWUT Report | After every SW unit testing (automatically generated from Testing Tool) |
| SWUT Summary Report | Software unit verification summary report |
| SWITC | After developing/updating of SW architecture design |

* 1. **시험 위배 관리**

(예제) 이 전략은 문제 해결 관리 계획(PRMP)의 시험 결함 해결 절차 x.x 장에 문서화되어 있습니다.

식별된 결함은 SUP.9의 전략에 의해 관리되어야 하고 공급자의 결함은 SM이 관리해야 하며, 결함에 대한 사전 분석을 통해 PM에게 결함 티켓을 생성합니다.

* Internal Test Defect

HTS에서 실시하는 SW 레벨 검증의 defect은 다음과 같이 관리한다.

* SWUT : SW unit 별 unit test는 Test 중 모든 하자가 발생한 경우에 한하여 완료한다. Unit test 중 확인된 결함은 자체 단계 내에서 교정하는 것을 원칙으로 하며, 단계에서 작성한 Test report에 그 결함을 관리할 수 있다.
* Static Analysis : SW component Static Analysis report에 결함을 기록 관리한다. SW PL 및 SW Test Leader는 요약 보고서에 모든 SW unit에 대한 Static Analysis 결함 총수를 수집하여 관리한다.
* Acceptance Test Defect

공급업체의 작업 제품에 대한 Acceptance test에서 확인된 defect는 다음과 같이 관리한다.

* Acceptance test defect는 defect list(PTC의 작업 항목)을 이용하여 관리
* Defect list는 Test report와 함께 공급업체 책임자를 통해 공급업체에 전달되며, 조치 결과에 대한 피드백을 받는다.

1. **소프트웨어 유닛 시험**
   1. **유닛 시험 전략**

(예제) 모든 유닛 시험 활동의 계획 및 범위는 유닛 시험 전략에 따라 달성됩니다.

SweDD는 유닛 시험 전략의 기본 문서입니다.

(1) 검증 도구: Matlab/Simulink

(2) 검증 규칙:

- MAAB 3.0, MISRA C 2012, Code Metrics, Code generation advisor

- Boundary and equivalent analysis Test (Black Box Test):

> Boundary test 목적에 따른 입력값의 범위 및 경계값의 설계 및 Test

> Boundary test는 모든 입력 신호(min, min+1, center, Max-1, max-1, max)에 대해 최소값, 최대값, 최대값의 최소값, 최대값 등을 확인하고 Over flow, Data flip 등 공통 오류 없이 장치가 작동하는지 확인한다.

> SWDD에 기술된 Simulink 옵션에 따라 Lookup Table 또는 Operator Block에서 입력 범위를 벗어난 값이 오류 없이 작동하는지 확인

> 조건에 대한 경계값 검증

- Missing Coverage Test (White Box Test): defect찾기

> Missing Coverage Test는 Boundary test 및 requirement based test 후 분석 및 Test한다.

> Missing Coverage report는 SWDD 엔지니어들과 논의해야 한다.

> SWDD 엔지니어들은 왜 특정 누락된 커버리지가 있는지 정당화해야 한다.

- SWC Test (Unit Integration Test)

> SWDD 상세설명에 따른 Test사례 설계

> Test결과 SWC가 실행 순서에 따라 동작하는지 확인하여 dynamic behavior 확인

(3) Purpose: SWDD에 따른 SWU 및 소스 코드 확인

(4) Input: SWDD

(5) Output: SWUV 결과보고서, SWUV 요약보고서

(6) ASPICE BPs: SWE4. BP4

* 1. **유닛 시험 기준**
     1. **소프트웨어 유닛 시험**

컴포넌트 내 개발된 모든 유닛은 소프트웨어 유닛 시험을 받는다.

* + 1. **시험 Entry/Suspension/Resumption 기준**

Table 5 Test suspension/Resumption Criteria for Static Analysis

|  |  |  |
| --- | --- | --- |
| criteria | | Remark |
| Entry | PTC CM 예외 입력 기준에서 SWDD의 해제된 체크포인트를 확인한 후: 작업 일수가 5일 이상 지연될 것으로 예상되는 시점에서, 위에서 정의한 진입 기준에 도달하지 않았더라도 정적 분석은 활동을 진행할 수 있다. | check the version history view project History |
| Suspending Condition | 각 릴리즈 기준으로 SW Unit 또는 Component에 결함이 있는 경우 | create a defect |
| Resuming Condition | 결함 수정 후 |  |

Table 6 Test suspension/Resumption Criteria for SWUT

|  |  |  |
| --- | --- | --- |
| criteria | | Remark |
| Entry | PTC CM 예외 입력 기준에서 SWDD의 해제된 체크포인트를 확인한 후: 작업 일수가 5일 이상 지연될 것으로 예상되는 시점에서, 위에서 정의한 진입 기준에 도달하지 않았더라도 정적 분석은 활동을 진행할 수 있다. |  |
| Suspending Condition | 각 릴리즈 기준으로 SW Unit 또는 Component에 결함이 있는 경우 | create a defect |
| Resuming Condition | 결함 수정 후 |  |

Table 7 Test suspension/Resumption Criteria for SWCT

|  |  |  |
| --- | --- | --- |
| criteria | | Remark |
| Entry | SW Unit에 결함이 없음을 확인한 후 |  |
| Suspending Condition | SW Component에 결함이 있는 경우 | create a defect |
| Resuming Condition | 결함 수정 후 |  |

Table 8 Test suspension/Resumption Criteria for Regression Test

|  |  |  |
| --- | --- | --- |
| criteria | | Remark |
| Entry | 결함 또는 CR(변경 요청) 발생 시 SW Unit 수정 시 |  |
| Suspending Condition | SW Unit이나 SW Component에 결함이 있는 경우 | create a defect |
| Resuming Condition | 결함 수정 후 |  |

* + 1. **시험 완료 기준**

이 프로젝트에서 시스템 통합 시험의 시험 완료 조건은 컨트롤러 PM과 함께 각 단계에 대해 다음 조건에 의해 판단됩니다.

* 1. **유닛 시험 환경**
     1. **시험 환경 개요**

소프트웨어 유닛 시험은 아래와 같은 소프트웨어 유닛 시험 환경에서 진행되어야 합니다. 소프트웨어 아키텍처 설계에 따라 제어 시스템을 확인하기 위한 전반적인 시험.

* + 1. **시험 데이터 요구사항**

CAN 데이터베이스 파일, DCM 사전 설정 교정 데이터, HIL 시뮬레이션 플랜트 및 시험 벡터와 같은 모든 시험 데이터는 PTC의 CM에 의해 저장 및 관리됩니다. 시험용 데이터의 모든 버전 및 정보는 시험 보고서에 작성됩니다.

* + 1. **시험 환경 요구사항**

Table 9 Test environment of system Integration test

|  |  |  |
| --- | --- | --- |
| Item | Description | Remark |
| HILS | * Interfaces verification on HILs * HW ETAS LABCAR-HIL * SW LABCAR-OPERATOR | HIL envireonment & conficuration |
| Vehicle | * Interfaces verification on Vehicle side * VW Atlas |  |
|  |  |  |
|  |  |  |