2019

소프트웨어 적합성 인증 지침

GC-30-K

한 국 선 급

“소프트웨어 적합성 인증 지침”의 적용

1. 별도로 명시하지 않는 한, 이 지침은 2019년 7월 1일 이후 소프트웨어 적합성에 대하여 인증을 받고자 하는 소프트웨어에 적용한다.

차 례

제 1 장 일반사항

제 1 절 일반사항

101. 적용

1. 이 지침은 선박 및 해양플랜트에 탑재되는 제어, 감시 및 안전시스템 소프트웨어, 선박 및 해양구조물의 업무지원 소프트웨어, 조선해양 관련 제품 설계와 기술/공학용 소프트웨어에 적용한다.
2. 이 지침에서 규정하지 아니하는 사항은 **선급 및 강선규칙**의 각 해당 요건에 따른다. 다만, 소프트웨어에 적용할 수 없는 요건은 제외한다.
3. 이 지침에 포함되지 않은 사항에 대하여는 우리 선급이 적절하다고 인정하는 바에 따라 ISO, IEC 또는 이와 동등 이상의 인정된 기준에 따를 수 있다.
4. 본 지침의 규정 이외에 IMO 등 국제 규정에 따른 별도 요건이 있는 경우 추가적인 고려사항 및 요건이 요구될 수 있다.

102. 용어의 정의

용어의 정의는 여기에 별도로 정하는 경우를 제외하고는 **선급 및 강선규칙**에 따른다.

1. **소프트웨어 제품(Software Product)**이라 함은 일련의 컴퓨터 프로그램, 절차 및 관련된 문서와 데이터를 말한다.
2. **요구사항(Requirements)**이라 함은 성취되거나 구현되어야 하는 것들의 인지된 필요에 대한 표현을 말한다.
3. **생애주기(Life cycle)**라 함은 소프트웨어 제품의 개발, 운영 및 유지 보수에 관련된 프로세스, 활동 및 과업을 담고 있고, 시스템 요구사항 정의에서부터 시스템의 사용 종료에 이르는 시스템의 수명을 말한다.
4. **시스템(System)**이라 함은 하나 이상의 특정 목적을 달성하기 위해 구성되어 상호작용하는 요소들의 조합을 말한다.
5. **검증(Verification)**이라 함은 객관적 증거의 규정에 의해 명시된 요구사항이 충족되었다는 것을 확인하는 것을 말한다.
6. **형상 항목(Configuration Item)**이라 소프트웨어 제품의 개발, 운영 및 유지 보수에 관련된 프로세스, 활동 및 과업을 말한다.
7. **형상 관리(Configuration Management)**라 함은 소프트웨어 항목을 포함한 형상 항목의 설계, 개발 및 지원에 기술적 및 행정적인 방향을 제시하는 관리를 말한다.
8. **베이스라인(Baseline)**이라 함은 매체에 상관없이 형상 항목의 생애주기 동안의 특정한 시간에 공식적으로 지정되고 고정된, 공식적으로 승인된 형상 항목의 버전을 말한다.
9. **형상 통제 위원회(Configuration Control Board)**라 함은 형상 항목의 변경을 수락하거나 거절하는 역할을 하는 집단을 말한다.
10. **인터페이스(Interface)**라 함은 하나의 시스템을 구성하는 하드웨어와 소프트웨어 또는 2개의 시스템이 상호 작용할 수 있도록 접속되는 경계나 이 경계에서 상호 접속하기 위한 하드웨어, 소프트웨어, 조건, 규약 등을 포괄적으로 가리키는 것을 말한다.
11. **버스(Bus)**라 함은 컴퓨터 내·외부 각종 신호원 간의 데이터나 전원 전송용 공통 전송로를 말한다.
12. **저장 장치(Storage Devices)**라 함은 컴퓨터가 사용하는 프로그램과 자료를 저장하는 장치를 말한다.
13. **무결성(Integrity)**라 함은 소프트웨어의 정보가 완벽하고 정확하며 정당함을 보장하고 인가되지 않은 변경이 없음을 보장하는 것을 말한다.
14. **로깅 프로세스(Logging Process)**라 함은 시스템을 작동할 때 시스템 작동 상태의 기록과 보존, 사용자의 습성 조사 및 시스템 동작의 분석 등을 수행하기 위해 작동중의 각종 정보를 기록하는 절차를 말한다.
15. **아키텍처(Architecture)**라 함은 시스템을 추론하기 위해 필요한 구조의 집합이며, 이 구조는 요소와 관계 그리고 요소속성과 관계속성을 포함하는 것을 말한다.
16. **데이터베이스(Database)**라 함은 자료항목의 중복을 피하고 정보를 일원화하여 처리를 효율적으로 수행하기 위해서 서로 관련성을 가지며 중복이 없는 데이터의 집합을 유지하는 것을 말한다.
17. **메커니즘(Mechanism)**이라 함은 어떤 소프트웨어의 동작하는 원리를 말한다.
18. **어플리케이션(Application)**이라 함은 특정한 업무를 수행하기 위해 개발된 응용 소프트웨어를 말한다.

19. **힙 기반 메모리(Heap-based Memory)**라 함은 프로그램의 실행 도중에 요구되는 기억 장소를 할당하기 위하여 운영 체제에 예약되어 있는 기억 장소 영역을 말한다.
20. **스택 기반 메모리(Stack-based Memory)**라 함은 후입선출(LIFO, Last-in First-out)과 같은 방식을 통해서 산술 계산, 지역 변수 또는 내부 동작을 추적하기 위해 사용되는 예약된 메모리를 말한다.
21. **진입점(Entry Point)**이라 함은 어느 프로그램을 실행하고 있을 때 다른 프로그램 기능이 필요하게 되어 호출된 프로그램이 최초로 실행하는 명령이 저장되어 있는 어드레스를 말한다.
22. **쓰레기 수집(Garbage Collection)**이라 함은 메모리 시스템에 할당 해제 또는 반환할 개체를 수동으로 지정하지 않고 더 이상 사용하지 않는 메모리를 되찾는 관리 형태를 말한다.
23. **조기 메모리 해제(Premature Deallocation)**라 함은 프로그램의 다른 곳에서 메모리를 사용하지 않는다는 확인하기 전에는 반드시 힙에게 메모리를 반환하지 말아야 하는 것을 말한다.
24. **단편화 기아(Fragmentation Starvation)**라 함은 주기억 장치의 활용도를 극대화하기 위해 프로그램을 비연속적인 곳에 적재하는 작업을 할 때 프로세스가 필요한 자원들의 권한을 영원히 얻지 못해 자신의 일을 하지 못하는 경우를 말한다.
25. **호출 그래프(Call Graphs)**라 함은 컴퓨터 프로그램에서 서브루틴 사이의 호출 관계를 나타내는 흐름 분석 그래프를 말한다.
26. **모듈성(Modularity)**이라 함은 서로 상호작용하는 작은 서브시스템으로 분할되는 소프트웨어의 특성을 의미한다. 모듈은 기능에 따라 구분되고 프로그래머는 다른 모듈의 기능들에 관여하지 않아서 새로운 기능을 개별 모듈에서 쉽게 프로그래밍 할 수 있다.
27. **런타임 오류(Run-time Errors)**라 함은 컴퓨터 언어 안에 쓰인 프로그램을 관리하기 위해 특정한 컴파일러나 가상 머신이 사용하는 기본 코드의 라이브러리나 프로그램의 오류를 말한다.
28. **상용제품(Commercial Off-The-Shelf)**이라 함은 특정 용도에 맞춘 컴퓨터 하드웨어 또는 소프트웨어 제품이고 회사나 일반 대중이 이용할 수 있게 만든 제품을 의미한다.
29. **시뮬레이터(Simulator)**라 함은 다른 컴퓨터 환경을 위해 만들어진 응용 프로그램을 실행하여, 실제 프로세스 또는 시스템을 모방할 수 있게 해주는 소프트웨어를 말한다.
30. **시나리오(Scenario)**라 함은 시스템 환경과 사용자의 상호 작용 및 해당 구성 요소 간의 상호 작용을 포함하는 일반적인 사고의 단계에 대한 설명을 말한다.
31. **회귀 테스트(Regression test)**라 함은 고장의 일부 변경으로 응용 프로그램의 다른 부분에 새로운 고장이 발생했는지 여부를 결정하기 위한 시험을 말한다.

103. 동등효력

이 지침의 규정에서 정한 것과 동등 이상의 적합성이 있다고 우리 선급이 인정하는 경우에는 이 지침의 규정에 적합한 것으로 본다.

104. 제외사항

우리 선급은 이 지침에 명시되지 않은 기타 기술적인 특성 및 허가받지 않은 상용제품의 사용에 대하여는 책임을 지지 아니한다. ⚓

제 2 장 적합성 인증 절차

제 1 절 인증 신청

101. 일반사항

1. 소프트웨어 적합성 인증을 받고자 하는 신청자는 적합성 인증 신청서 1부와 함께 다음 **102.** 및 **103.**에 기재된 자료 (승인용 3부 및 참고용 1부)를 우리선급에 제출하여야 한다.
2. 모든 자료는 ISO 9001에서 규정한 문서 및 기록 요건에 맞게 작성하여야 한다.
3. 우리 선급은 제출 자료를 심사하여 **3장**의 요건에 적합할 경우 제출된 자료를 승인하여 신청자에게 송부한다.

102. 승인용 자료

1. 제출 문서
   (1) 사업수행 계획서
   (2) 형상 관리 계획서
   (3) 요구사항 명세서
   (4) 기본 및 상세 설계서
   (5) 소스 코드 또는 소스 코드 검증 보고서
   (6) 평가 계획서
   (7) 평가 결과서 (오류수정 보고서 포함)
   (8) 운영 및 유지관리 계획서

103. 참고용 자료

1. 개발조직 개요
   (1) 개발조직 구성, 개발조직 관리체계
   (2) 개발품 목록 및 판매실적
   (3) 핵심 개발 인력 경력사항
2. 사용자 및/또는 운영자 설명서, 설치 설명서
3. 이전에 취득한 소프트웨어 적합성 관련 시험 인증서 및 보고서
4. 기타 우리선급이 필요하다고 인정하는 자료

제 2 절 심사 및 시험

201. 자료 심사

1. **102.**와 **103.**의 자료는 **표 2.1**에 따라 각 소프트웨어 생애주기에 맞춰 제출 및 승인을 받아야 한다.

202. 인증 시험

1. 인증 시험은 평가 계획서가 승인된 후 시행하여야 한다.
2. 인증 시험은 원칙적으로 검사원의 입회하에 실시하여야 한다. 단, 우리 선급에서 인정하는 공인된 검사기관에서 수행할 경우 생략할 수 있다.
3. 인증 시험 항목은 소프트웨어의 요구사항에 따른 기능이 적절하게 동작하는지를 확인할 수 있는 항목으로 구성하여야 한다.
4. 시험 결과는 합격/불합격으로 판정되고, 불합격 항목이 발생한 경우 소프트웨어 수정 없이 재시험을 할 수 없다.
5. 다른 장비와 연동되는 시험의 경우 실제 연동대상 장비를 사용하거나, 이를 모사할 수 있고 우리 선급이 적절하다고 인정하는 시스템을 적용할 수 있다.

6. 신청자는 인증 시험 완료 후, 평가 결과서를 우리 선급에 제출하여야 한다.

표 2.1 생애주기별 제출하여야 할 문서

생애주기	제출 문서	비고
계획 수립	사업수행 계획서, 형상 관리 계획서	
요구사항 분석	요구사항 명세서	
설계 및 구현	기본 및 상세설계서, 소스 코드(또는 소스 코드 검증 보고서)	
소프트웨어 검증	평가 계획서, 평가 결과서	평가 결과서는 평가가 완료된 후 제출하여야 한다.
운영 및 유지 보수	운영 및 유지 관리 계획서	

제 3 절 인증

301. 인증 통지

1. 인증 시험 완료 후 제출된 평가 결과서가 양호하다고 인정되는 경우, 우리 선급은 적합성 인증서를 발행하여 신청자에게 송부하여야 한다.
2. 적합성 인증서에는 제품의 적합성, 인증 범위 및 부가된 제한 및 제약사항이 명확히 기술하여야 한다.
3. 소프트웨어 적합성 인증서에는 다음과 같은 최소 정보가 포함하고 있어야 한다.
   (1) 소프트웨어 개발자가 등록된 사무실의 이름과 주소
   (2) 공급업자가 등록된 사무실의 이름과 주소(다른 경우)
   (3) 설명, 버전 번호, 기타 관련 설정 정보
   (4) 소프트웨어가 동작하는 것이 보장되는 환경(기타 작업 제한 또는 조건 포함)
   (5) 소프트웨어가 적합한 것으로 판명되는 특정 평가 모듈
   (6) 인증서 번호와 발행일
   (7) 인증서의 유효기간

302. 인증 내용의 변경

1. 신청자는 이미 인증을 받은 소프트웨어에 대해 변경이 있는 경우 우리 선급에 통보하여야 한다.
2. 모든 변경에 대해서 신청자는 소프트웨어 버전과 관련된 품질절차에 따라 3항의 기록을 유지하고 관리하여야 하고, 우리 선급의 요청 시 해당 기록을 제출하여야 한다. 이 때 우리 선급은 변경 내용을 심사한 후, 필요에 따라 확인 시험을 요구할 수 있다.
3. 모든 변경에 대한 추적을 목적으로 형상 관리 되는 기록은 다음과 같다.
   (1) 프로젝트에서의 변경 횟수
   (2) 최근 소프트웨어 항목의 버전, 배포 방법 식별자, 배포 횟수, 배포 버전 간의 비교 내용
   (3) 베이스라인의 상태
   (4) 형상 통제 내역(변경 제어 상태)
   (5) 형상 통제 위원회 활동 내역
4. 다음의 경우에는 변경 사항으로 인정하지 않고 재인증 받아야 한다. 단, 기존 제출 자료와 동일한 경우 자료 제출을 생략할 수 있다.
   (1) 이미 평가된 소프트웨어에 추가 버전의 소프트웨어를 포함하는 경우
   (2) 추가적인 사양 또는 표준을 가지는 경우
   (3) 복제하여 새로운 환경에 설치하는 경우

303. 인증서의 유효기간 및 갱신

1. 적합성 인증서의 유효기간은 증서발행일로부터 3년으로 한다. 다만, 이전 302.에 따라 인증서를 재교부할 경우에는 인증서 유효기간은 기존 증서의 유효기간으로 한다.

2. 구성요소 또는 구조의 변경이 없다면 우리 선급은 재시험 없이 적합성 인증서를 갱신할 수 있다.
3. 적합성 인증서의 갱신 시에는, 소프트웨어 사용 중 장애보고 및 고객 불만과 같은 추가 정보에 관한 검토와 필요시 추가 검사를 요구 할 수 있다. ⚓

제 3 장 소프트웨어 생애주기별 요건

제 1 절 일반사항

101. 일반사항

1. 이 지침은 다음의 생애주기를 통해서 개발된 소프트웨어의 적합성 인증을 한다.
   (1) 계획 수립
   (2) 요구 사항
   (3) 설계 및 구현
   (4) 검증
   (5) 운영 및 유지 보수
2. 소프트웨어는 원칙적으로 영어를 이용하여 개발되고 사용하여야 한다.
3. 생애주기별 각 단계마다 검증 활동을 할 수 있도록 계획하여야 한다.

제 2 절 계획 수립

201. 일반사항

1. 사업수행 계획서에는 소프트웨어 개발 생애주기별 각 활동의 입력, 산출물, 행위자 및 책임으로 구분하여 식별하고 기술하여야 한다.
2. 신청자 또는 소프트웨어 공급자는 형상 관리 담당자를 지정하여야 한다. 이때, 소프트웨어 개발 담당자와 형상 관리 담당자는 동일 할 수 있다.
3. 형상 관리 계획서는 202.의 관한 내용을 포함하여 제출하여야 한다.

202. 형상 관리 담당자

1. 형상 관리 담당자는 프로젝트 관리자가 정의한 형상 관리 계획서에 따라 운영 환경을 구현하고 형상 관리 활동을 수행한다.
2. 형상 관리 담당자의 역할은 다음과 같다.
   (1) 형상 관리 계획서 작성 참여
   (2) 형상 관리 환경 구축 및 형상 관리 저장소 생성
   (3) 형상 관리 절차 개발 및 문서화
   (4) 베이스라인의 기준을 선정
   (5) 파일, 문서, 버전에 관한 형상 식별자 규칙 선정
   (6) 주기적으로 형상 상태 보고

203. 형상 통제

1. 형상 통제 위원회의 구성원은 다음과 같다.
   (1) 프로젝트 관리자
   (2) 품질 담당자
   (3) 기술 담당자 및 고객 측 담당자
2. 형상 통제 위원회의 역할은 다음과 같다.
   (1) 형상 항목 결정
   (2) 베이스라인 수립 여부 결정
   (3) 승인된 변경에 대한 책임 및 보증
   (4) 베이스라인의 변경 요청이 필요한 경우 이에 대한 검토 및 승인
3. 형상 관리 담당자에 의해서 식별된 형상 관리 대상에 관하여 다음의 절차를 통해 관리한다.
   (1) 변경 요청

고객이나 개발자는 변경 사항이 생겼을 때 변경 요청서를 작성하여 형상 관리 담당자에게 제출하여야 한다.

(2) 변경 심사

고객이나 개발자가 작성한 변경 요청서가 들어오면 형상 통제 위원회는 다음을 검토하여 변경 요청 여부를 결정 하여야 한다.

(가) 변경 이유의 타당성

(나) 변경 범위

(다) 변경이 미치는 영향(기간, 예산, 인력 등)

(3) 변경 실시

변경을 위해서는 저장소에 보관 중인 해당 항목을 가져오고 형상 항목을 변경한다. 변경 시 만들어지는 결과물에 대해서는 이력 관리가 이루어져야 한다.

(4) 변경 확인

형상 통제 위원회는 변경된 내역을 확인 및 승인하여야 하고 저장소에 새로이 저장된 변경 항목은 다시 베이스라인으로 수립하여야 한다

제 3 절 요구 사항

301. 일반사항

1. 요구사항 명세서는 기능 및 비기능적 요구사항에 대한 획득, 분석, 명세 및 검증과 관련된 활동을 포함하여야 한다.
2. 소프트웨어 요구사항은 기능적 요구사항과 비기능적 요구사항으로 구분한다.
3. 소프트웨어 요구사항은 각 소프트웨어 구성요소에 따라 세분화하여 식별하여야 한다.

302. 소프트웨어 요구사항

1. 소프트웨어의 기능성 요구사항은 다음과 같은 내용을 포함하여야 한다.
   (1) 운영상 요구 사항
   (2) 매개 변수 요구 사항
   (3) 소프트웨어 시스템에 대한 입력 및 출력 정의 (예: 데이터의 범위와 유형, 한계 및 기본값)
   (4) 인터페이스 요구 사항
      (가) 소프트웨어 시스템과 하드웨어 시스템 (예: 데이터 버스, 드라이버, 대상 플랫폼, 저장 장치)
      (나) 소프트웨어의 시스템 간의 인터페이스 설명
2. 소프트웨어의 비기능성 요구사항은 적어도 다음과 같은 내용을 포함한다.
   (1) 소프트웨어를 고장 없이 사용할 수 있는 시간 및 환경
   (2) 예상 응답 시간 및 저장 기간
   (3) 데이터 분류 수준, 민감한 데이터 관리 제한, 인증 수단, 권한 수준, 무결성 검사 및 로깅 프로세스
   (4) 안전 요구사항
      (가) 대체 위치
      (나) 중복 전략 및 오류 감지 기능
   (5) 수동 작동에 대한 지원 및 인력 상호 작용 관리
   (6) 소프트웨어에 의해 생성 된 경보, 경고 및 운영자 메시지
   (7) 소프트웨어 파티션, 분리된 프로세서 사용 및 기타 분리 전략
   (8) 개발 및 유지관리 장소에 인도된 소프트웨어 버전의 설치 및 수락
   (9) 제품 또는 사용자 문서의 구조
   (10) 국가 또는 국제 표준
   (11) 규제 요구 사항(해당되는 경우)

303. 요구사항의 검증

1. 소프트웨어 요구 사항은 다음의 내용에 대하여 검증하고 기록하여야 한다.

(1) 완전성(Completeness)
필요한 정보가 누락되지 않고 모두 서술 하여야 한다.
(2) 명확성(Unambiguity)
표현을 명확히 하여 누구나 동일한 의미로 해석할 수 있어야 한다.
(3) 일관성(Consistency)
서로 상반된 요구뿐 아니라 불일치한 요구가 명세서에 존재해서는 안 된다.
(4) 추적 가능성(Traceability)
절차나 기법, 담당자뿐만 아니라 요구 사항 명세서를 변경하고자 한다면 해당되는 설계 사양서와 코드 부분까지도 추적할 수 있어야 한다.
(5) 변경 용이성(Modifiability)
요구사항은 모순이나 불일치가 발생하지 않기 위해 독립적으로 서술하여야 한다.
(6) 검증 가능성(Verifiability)
문장에 애매모호한 표현이 포함되어 있지 않고 체계적으로 검사할 수 있게 작성하여야 한다.

제 4 절 설계 및 구현

401. 일반사항

1. 소프트웨어 설계는 다음의 내용을 고려해서 수행하여야 한다.
(1) 요구사항 명세서의 내용이 기본 및 상세 설계서에 모두 포함하여야 한다.
(2) 유지 보수가 용이하도록 추적이 가능하여야 한다.
(3) 변화에 쉽게 적응할 수 있어야 한다.
(4) 시스템 변경으로 인한 영향이 최소화 하여야 한다.
(5) 설계서는 쉽게 읽고 이해할 수 있어야 한다.
2. 설계 및 구현은 302.의 소프트웨어 요구 사항을 통해서 소프트웨어 아키텍처를 정의하여야 한다.
3. 소프트웨어 아키텍처는 소프트웨어 구성 요소들 사이의 관계(인터페이스)를 식별하여야 한다.
4. 설계 및 구현은 기본 설계와 상세 설계가 수행하여야 한다. 단, 기본 설계가 별도로 필요하지 않은 경우에는 생략할 수 있다.

402. 설계 요구사항

1. 302.의 소프트웨어 요구 사항에 추가하여, 소프트웨어 개발자는 다음을 포함한 소프트웨어 설계 요구 사항을 정의하고 기록하여야 한다.
(1) 소프트웨어 개발 언어 및 소프트웨어 구성 요소 간의 호환성
(2) 운영체제 및 네트워크 인프라의 타이밍 및 메모리 특성
(3) 데이터의 범위와 유형, 한계 및 기본값
(4) 소프트웨어 구성 요소 간의 인터페이스 설명 및 외부 인터페이스 설명
(5) 각 소프트웨어 구성 요소에 의해 생성 된 경보, 경고 및 운영자 메시지
(6) 이름 지정 및 형식화 규칙 및 변환 함수
(7) 구성요소에 할당 된 기능, 성능, 보안, 안전 및 오류 격리
2. 1항의 요구사항은 다음의 내용에 대하여 검증하고 기록하여야 한다.
(1) 소프트웨어 아키텍처는 일관성이 있어야 한다.
(2) 서로 모순되지 않아야 한다.
(3) 분명한 용어로 표현하여야 한다. (형식적인 방법을 사용 할 수 있다.)
(4) 설계 요구 사항은 확인 및 검증 활동을 위한 기초가 될 수 있다.
(5) 고유하게 식별 할 수 있다.
(6) 소프트웨어 요구 사항 또는 다른 요구사항의 출처를 확인 할 수 있어야 한다.

403. 아키텍처

아키텍처의 특징과 기능은 다음과 같은 내용을 포함 하여야 한다.

(1) 개발할 소프트웨어에 대한 전체적인 구조를 다뤄야 한다.
(2) 소프트웨어를 이루고 있는 여러 구성요소(서브시스템, 컴포넌트)를 다뤄야 한다.
(3) 구성 요소들이 인터페이스를 통해서 어떻게 상호작용하는지를 정의하여야 한다.
(3) 인터페이스를 통한 구성 요소들 사이의 상호작용에 대해 정의하여야 한다.
(4) 세부 내용보다는 중요한 부분만을 다뤄야 한다.
(5) 시스템 설계와 개발 시 적용되는 규정이 있어야 한다.
(6) 의사소통 도구로 활용할 수 있어야 한다.
(7) 구현에 대한 제약 사항을 정의하여야 한다.
(8) 품질 속성을 결정하여야 한다.
(9) 재사용할 수 있게 설계하여야 한다.

404. 기본 설계

1. 기본 설계는 아키텍처, 어플리케이션, 인터페이스, 데이터베이스를 정의한다.
2. 개발자는 소프트웨어 구성 요소와 외부 구성 요소(예를 들어, 다른 시스템 소프트웨어 및 하드웨어 시스템) 간의 모든 인터페이스를 설명하여야 한다. 이 때, 외부 인터페이스에는 버스(Bus), 저장 장치(RAM, EEPROM, FLASH 등) 및 소프트웨어 시스템에서 사용할 수 있는 입력/출력 장치가 포함 된다.
3. 아키텍처는 타이밍 요구 사항 측면에서 다음을 포함한 소프트웨어 시스템의 동작을 지정하여야 한다.
   (1) 데이터 간의 우선순위, 최대 실행 시간 및 활성화 조건
   (2) 다른 입력 지점의 동기화 및 스케줄링
   (3) 공유 변수에 대한 보호 메커니즘 (그리고 연동 차단의 정당성)
   (4) 공유 변수의 식별
4. 소프트웨어 구성 요소의 진입점은 작업요청, 중단 또는 소프트웨어 시스템의 주체를 호출하여야 한다.
5. 아키텍처는 메모리 관리 요구 사항에 대해 다음의 경우를 포함한 소프트웨어 시스템의 동작을 지정하여야 한다.
   (1) 힙 기반 또는 스택 기반 메모리 할당, 쓰레기 수집
   (2) 힙 메모리 부족, 조기 메모리 해제 또는 단편화 기아
6. 보호 메커니즘은 호출 그래프의 최대 깊이에 의해 정의 하여야 한다.

405. 기본 설계의 검증

기본 설계는 다음에 대하여 검증하고 기록하여야 한다.
(1) 소프트웨어 아키텍처는 소프트웨어 요구 사항을 구현할 수 있어야 한다.
(2) 소프트웨어 아키텍처는 모듈성(Modularity)이 일관되어야 한다.
(3) 소프트웨어 아키텍처는 소프트웨어 시스템의 모든 내부 및 외부 인터페이스를 설명하여야 한다.

406. 상세 설계

1. 소프트웨어 아키텍처의 각 소프트웨어 구성요소는 해당되는 단위 소프트웨어로 나뉘어져야 한다.
2. 소프트웨어 구성 요소의 각각의 단위 소프트웨어에 대하여 설계하여야 한다. 각 단위 소프트웨어의 실제 입력 및 출력과 해당 유형 및 기능 영역 정의가 포함하여야 한다.
3. 단위 소프트웨어와 외부 구성요소(하드웨어 및 소프트웨어) 간의 인터페이스에 대한 설계가 이루어 져야 한다.

407. 상세 설계의 검증

다음의 사항에 대하여 소프트웨어 상세 설계를 검증하고 기록하여야 한다.
(1) 소프트웨어 구성 요소
(2) 소프트웨어 아키텍처
(3) 설계 기준
(4) 소프트웨어의 일관성

408. 구현

1. 기본 설계 및 상세 설계사항에 따라서 각 단위 소프트웨어를 구현 하여야 한다.
2. 소스 코드(source code)는 다음의 내용을 별도로 정의하고 적용하여야 한다.
   (1) 명명 규칙
   (2) 스타일 관례 (파일, 헤더, 기능, 코드의 주석 등)
   (3) 계량적 목표 (복잡성, 기능의 크기 등)
   (4) 환경 제약 조건 (컴파일러 옵션 등)
   (5) 프로그래밍 언어 제한 (위험하거나 복잡한 구성의 사용을 금지 또는 통제)
3. 소스 코드의 데드 코드(dead code, 필요 없는 코드 또는 사용하지 않는 코드)와 런타임 오류를 감지하여야 한다. 재귀적 문구는 최대 반복 횟수로 제어되는 경우에만 허용된다.
4. 수치의 정확성이 예상되고 검증하여야 한다.

409. 구현 검증

1. 소스 코드는 다음의 내용에 대하여 검증을 하고 기록하여야 한다.
   (1) 소프트웨어의 소스 코드는 정의된 프로그래밍 절차 및/또는 코딩 규칙을 준수하여야 한다.
   (2) 구현 결과가 일치하지 않는 부분을 분석하고 수정하여야 한다.
   (3) 소스 코드는 소프트웨어 요구사항과 설계 요구사항에 따라 실수 없이 작성하여야 한다.
   (4) 소스 코드의 로직은 명확하고 불필요하게 복잡하지 않아야 한다.
   (5) 소스 코드의 로직은 읽기 쉽고 주석을 작성하여야 한다. 상용제품의 구성은 통합, 등록자 그리고 구성에 대한 명확한 주석을 작성하여야 한다.
   (6) 예상치 않은 성능 병목 현상은 없어야 한다.
2. 1항의 소스 코드 검증 사항은 우리 선급이 인정하는 기관에서 수행 할 수 있다. 이 경우 소스 코드 검증 보고서를 우리선급에 제출 및 승인을 받아야 한다.
3. 코드 분석 도구는 데드 코드 및 런타임 오류를 확인하는 데 사용 할 수 있다.

제 5 절 검증 및 확인

501. 일반사항

구현이 완료된 소프트웨어의 구성요소를 통합하여 요구사항을 만족하는지 검증하여야 한다. 하드웨어에 내장되는 소프트웨어의 경우에는 해당되는 하드웨어에 설치하여 검증하여야 한다.

502. 소프트웨어 평가 계획

1. 소프트웨어 평가 계획서에는 다음의 내용을 포함하여 소프트웨어의 적합성 평가를 위한 계획, 방법 및 절차를 수립하여야 한다.
   (1) 제품 설명 및 사용 설명서에 있는 모든 기능
   (2) 시험 목적, 입력 데이터, 예상 결과 및 합격 또는 불합격 기준
   (3) 시험을 수행하고 결과를 기록하는 데 필요한 준비 작업을 포함하는 시험절차
2. 소프트웨어의 각 기능 및 특징은 적어도 두 가지 이상의 경우를 적용하여 시험한다.
3. 제품의 설명서와 사용자 문서에 표시된 설치 및 조작상의 제한을 시험하여야 한다.
4. 시험 계획은 소프트웨어 제품 설명에 언급 된 모든 구성을 포함하여 시험 할 하드웨어 및 소프트웨어-구성을 명시하여야 한다. 필요시 소프트웨어 시험에 필요한 모든 도구를 지정하여야 한다.
5. 유효성 검사 전략에는 활동 목표에 도달하기 위해 필요한 테스트 벤치를 정의 하여야 한다. 또한 시뮬레이터는 소프트웨어 시스템이 실제 환경에서 시험되는 것을 의미하므로 적합한 방법으로 간주 할 수 있다.
6. 하드웨어 및/또는 소프트웨어 구성 요소의 상호 작용은 평가 계획서 상의 소프트웨어의 시험 시 고려하여야 한다.
7. 시험 결과가 기준을 만족하지 못하는 경우 관련 기능 또는 특징을 재시험하기 위한 절차를 수립하여야 한다.

503. 소프트웨어 검증 기준

소프트웨어에 대한 허용 기준은 다음을 고려하여 검증하여야 한다.

(1) 정상적인 조건에서 예상되는 요구사항
(2) 정상적인 조건에서 예기치 않은 요구사항
(3) 비정상 조건에서 예상되는 요구사항
(4) 비정상적인 조건에서 예기치 않은 요구사항
(5) 기존에 식별된 발생 가능한 오류로 이어질 수 있는 조건

504. 소프트웨어 검증

1. 적합성 평가 계획에 따라 소프트웨어 구성 요소의 적합성을 검사하여야 한다. 이 활동은 최종 목표 환경에서 수행하여야 한다.
2. 모든 검증 과정 및 결과(불일치 목록뿐만 아니라 합격 또는 불합격)가 문서화하고 보존하여야 한다.
3. 평가 결과의 문서화는 다음의 내용을 고려하여야 한다.
   (1) 소프트웨어 요구사항 및 설계에 대한 추적
   (2) 소프트웨어 요구사항 및 설계의 일관성
   (3) 설계 요구사항 간의 일관성
   (4) 소프트웨어의 테스트 범위
   (5) 운영 그리고 유지관리의 가능성
4. 시험에 사용되는 소프트웨어는 적합성 평가 중인 소프트웨어와 동일하여야 한다.

505. 평가 결과서

평가 결과서는 다음의 사항을 포함해서 작성히여야 한다.

(1) 모든 시뮬레이션 설계, 시뮬레이션 시나리오, 시뮬레이션 절차 및 시뮬레이션 결과
(2) 모든 평가가 계획에 따라 실행되었는지에 대한 증명
(3) 평가 날짜, 시험기의 명칭 및 기능, 발견된 예외 목록 및 해당 예외 보고서에 대한 참조
(4) 각 평가 사례에 대한 보고서

506. 재시험

1. 평가 결과가 승인된 검증 기준을 만족하지 못할 경우 다음을 사항을 포함해서 문서화 하여야 한다.
   (1) 평가에 의해 식별된 특이사항 및 특이사항이 식별된 시험 사례
   (2) 특이사항의 수정 및 재시험 검증 계획
   (3) 특이점의 수정 및 반영결과 입증을 위한 다음의 항목을 추가로 기술 하여야 한다.
      (가) 수정의 식별자
      (나) 수정 날짜
      (다) 교정자의 이름
      (라) 수정에 대응되는 변경의 식별자
      (마) 수정의 영향
      (바) 수정을 수행한 사람의 의견
   (4) 특이점의 수정이 필요하지 않은 경우, 관련 자료를 우리선급에 제출하여 승인을 받아야 한다.
2. 소프트웨어는 식별된 특이사항을 수정하여 503.에서 정의된 검증 기준이 충족됨을 확인하여야 한다.
3. 재시험이 이루어진 기능은 다음의 항목이 추가로 평가 결과서에 기술하여야 한다.
   (1) 검증의 식별자
   (2) 검증 일자
   (3) 검증자의 이름

제 6 절 운영 및 유지 보수

601. 일반사항

1. 요구사항에 따라 소프트웨어의 운영 및 유지 보수와 관련된 제반 활동을 수행한다.
2. 소프트웨어는 허용 가능한 품질 경영 시스템의 범위 내에서 복제, 인도 및 유지하여야 한다.
3. 형상 관리 시스템을 통해서 소프트웨어 유지 관리가 체계적으로 이루어져야 한다.

602. 운영 및 유지 보수

1. 품질관리

소프트웨어 제품의 품질 관리 계획은 소프트웨어 제품 생애주기의 모든 단계에서 형상 관리 프로세스의 구현 및 검증과 관련된 책임과 권한을 식별하고 설명하는 것이다. 형상 관리 프로세스와 관련된 다양한 활동들과 구현 활동을 검증하기 위한 책임 있는 권한의 식별 간의 인터페이스가 정의 하여야 한다.

2. 복제 및 인도

소프트웨어에 대한 복제 기록에는 적어도 다음의 내용을 포함하여야 한다.

(1) 포맷, 변형 및 버전을 포함한 소프트웨어의 원본 및 사본.
(2) 사용 된 매체 유형 및 관련 라벨링.
(3) 식별 및 포장을 포함한 관련 소프트웨어 제품 설명 및 소프트웨어 사용자 매뉴얼, 라이센스 및 배포 기록.
(4) 소프트웨어 제품의 제공 된 사본의 정확성과 완전성에 대한 검증
(5) 소프트웨어 제품이 배송 중 손상 또는 손상되지 않도록 보호하기 위한 조치

603. 유지 보수

1. 소프트웨어 제품에 대한 유지 보수는 다음과 같다.
   (1) 수정 유지 보수
   개발된 소프트웨어를 사용자가 인도받은 후 사용하면서 발견되는 오류를 해결하여야 한다.
   (2) 적응 유지 보수
   개발된 소프트웨어가 운영체제, 주변장치, 네트워크 환경 등 환경이 바뀌어도 이에 맞도록 수정 및 보완하여야 한다.
   (3) 기능 보강 유지 보수
   개발된 소프트웨어는 새로운 기능의 추가, 변경 및 성능 개선을 요구하는 경우 기능 보강 유지 보수를 하여야 한다.
   (4) 예방 유지 보수
   어떤 문제가 발생하였을 때 처리하는 것이 아니라 미리 예상되거나 예측되는 오류를 찾아 수정하여야 한다.
2. 소프트웨어는 유지 보수 기록을 유지하도록 계획 및 관리 하여야 하며 적어도 다음을 포함하여야 한다.
   (1) 접수 된 문제 보고서의 목록과 현재 상태
   (2) 시정 조치 이행을 담당하는 당국
   (3) 시정 조치 우선순위 부여
   (4) 시정 조치 결과
   (5) 소프트웨어 제품의 구매자에게 예정된 추후 변화에 대해 안내하는 방법.
   (6) 변경 사항이 적용되었는지 확인하기 위한 조치는 다른 문제를 야기하지 않아야 한다.

604. 변경 관리

1. 각 변경 요청에 대하여 기록 및 분석을 하여야 한다.
2. 변경 계획 수립 후 소프트웨어의 출시 버전에 대한 영향 분석을 하여야 한다. 영향 분석 시 반복하여야 하는 소프트웨어 개발 활동 및 갱신될 문서에 대한 계획이 수립하여야 한다.
3. 소프트웨어의 배포 버전을 수정하는 경우, 변경 사항에 의해 새로이 유입된 오류가 없는지 확인하기 위한 회귀 테스트를 수행 하여야 한다.
4. 소프트웨어 변경 유형 및 범주에 따라 검증 활동의 범위를 식별하여야 한다.

605. 형상 감사

1. 형상 감사는 형상 관리 담당자에 의해 실시되며 형상 관리 계획서 상에 형상 감사를 위한 계획서를 수립하여야 한다.
2. 형상 감사는 적어도 다음과 같은 내용을 중심으로 검증 하여야 한다.
   (1) 승인된 변경 요청의 반영에 대한 검증
   (2) 승인되지 않은 내용의 반영에 대한 검증
   (3) 승인된 변경과 관련된 항목의 갱신에 대한 검증

606. 소프트웨어 배포

소프트웨어의 버전을 출시하기 위해 다음을 수행 하여야 한다.
(1) 모든 검증 활동이 완료되었는지 확인하여야 한다.
(2) 소프트웨어에 존재하는 각각의 예외 사항을 문서화 하여야 한다.
(3) 소프트웨어의 버전은 고유하게 식별 할 수 있어야 한다.
(4) 소프트웨어와 관련된 문서, 모델 파일, 테스트 결과, 소스 코드 등과 같은 모든 산출물들은 수집되고 식별 하여야 한다.
(5) 소프트웨어 시스템을 설치하기 위한 절차를 문서화 하여야 한다.
(6) 설치 후 최종 환경에서 소프트웨어 시스템을 시험하기 위한 절차를 문서화하여야 한다.
(7) 운영 분야에서 피드백을 얻기 위한 절차를 문서화 하여야 한다.

607. 운영 및 유지관리 계획서

1. 운영 및 유지관리 계획서에는 운영 매뉴얼 등의 문서를 포함 하여야 한다.
2. 소프트웨어의 운영 및 유지 관리자를 지정하여야 한다. ⚓

인 쇄 2019년 3월 24일
발 행 2019년 4월 01일

소프트웨어 적합성 인증 지침

발행인 이 정 기
발행처 한 국 선 급
부산광역시 강서구 명지오션시티 9로 36
전 화 : 070-8799-7114
FAX : 070-8799-8999
Website : http://www.krs.co.kr

신고번호 : 제 2014-000001호 (93. 12. 01)