

부 록

1. 연구개발사업 제안요청서 내 소프트웨어분야 포함항목(예)
 2. 구매사업 제안요청서 내 소프트웨어분야 포함항목(예)
 3. 개발단계별 산출물 점검표
 4. 형상관리 및 품질보증 점검표
 5. 국방규격화 검토용 점검표
 6. 무기체계 소프트웨어 코딩규칙
 7. 소프트웨어 신뢰성/보안성 시험 절차
 8. 국방 아키텍처 필수작성 산출물
 9. 표준 개발 프로세스와 국방 CBD 개발 프로세스 간 매핑표
 10. 분류체계 식별자 관리
 11. 형상변경 참조 양식
 12. 품질보증 참조 양식
 13. 감리 참조 양식
 14. 공개SW 무기체계 적용 가이드라인
 15. 약어표
- ※ 별책 : 무기체계 소프트웨어 기술문서 작성 가이드 ('20.2.13일 개정)

제 안 요 청 서(안)

(00000 사업)

본 내용은 방위사업관리규정 별지 제1호서식 제안요청서(연구개발 사업)에 포함되는 소프트웨어분야 항목(예)으로서, SW개발에 대한 주요내용을 참고할 수 있도록 제시하였으며, 무기체계 사업특성을 고려하여 적용바랍니다.

2020. 0. 0

사업관리기관의 승인 없이 사용을 금함

※ 복제 / 복사 및 배포 금지

방위사업청 0000사업팀

(중략)

아. 소프트웨어 개발계획

1) 소프트웨어 개발 방안

- 가) 무기체계 소프트웨어는 방위사업청 『무기체계 소프트웨어 개발 및 관리 매뉴얼』을 적용하여 개발한다.
- 나) 무기체계 소프트웨어는 소프트웨어형상항목(CSCI : Computer Software Configuration Item)별로 주요기능, 개발방안(국내개발, 국내구매, 해외기술협력 개발, 해외구매, 재사용 등)을 식별하고 관리 되어야 한다.
- 다) 무기체계에 대한 하드웨어와 소프트웨어가 연계된 구조도를 작성하여 제시한다.
- 라) 재사용되는 소프트웨어를 식별하고, 관리계획(신뢰성시험방안 포함)을 제시한다.
- 마) 해외구매 등 외산소프트웨어에 대해 국산 소프트웨어 적용계획을 제시한다.
- 바) 무기체계 소프트웨어는 국내개발 및 국산 소프트웨어를 우선 적용하여야 한다.
- 사) 상용 소프트웨어 활용시 정부인증 GS(Good Software)를 우선 선정하여야 한다.
- 아) 소프트웨어가 최대한 재사용 및 확장될 수 있도록 체계개발 단계별로 소프트웨어 분류체계를 설계하고 개발해야 한다.
- 자) 소프트웨어의 체계적인 개발을 위한 소프트웨어검토회의, 소프트웨어 점검 및 확인, 소프트웨어 신뢰성시험 등 소프트웨어 개발관리 활동을 수행한다.
- 차) 체계규모 및 특성에 따른 최적의 개발방안을 구체화하고 소프트웨어형상항목별로 개발조직, 개발단계별 일정 및 기술자료 제출일정, SW통합방안 등을 제시한다.

2) 소프트웨어 개발방법론

- 가) 소프트웨어 개발단계별 표준화된 개발방법론 및 개발도구를 사용하여 최적의 소프트웨어가 개발되도록 추진한다.
- 나) 소프트웨어 개발단계별 자동화도구(요구사항관리, 분석설계, 구현 등) 적용방안 및 선정사유를 기술하고 체계개발에 사용한다. 개발 편리성 및 향후 운용유지를 위하여 통합개발환경도구를 적용하여 개발하며, 운용자 중심의 HMI(Human Machine Interface) 구현, 일관성 있는 GUI(Graphic User Interface)를 디자인하여 개발한다.

다) 무기체계 소프트웨어 개발시 유지보수성을 고려하여 설계에 반영하고 향후 유지관리가 용이하도록 개발하여야 한다.

- ① 무기체계 소프트웨어 개발에 필요한 모든 소프트웨어[개발환경]
- ② 무기체계에 탑재되어 운영되는 소프트웨어[운용환경]
- ③ 무기체계 운영유지에 필요한 소프트웨어

라) 무기체계 운용유지단계의 소프트웨어 유지보수 등 소프트웨어의 종합군수지원 요소 개발은 사업관리부서장과 연구개발주관기관의 협의를 통해 개발 범위와 세부내용을 결정하여 종합군수지원계획서(ILS-P)에 구체화하여 반영한다.

3) 소프트웨어 품질보증

가) 형상관리 프로세스 및 형상관리 도구가 포함된 소프트웨어 형상관리계획을 제시하고 체계개발에 적용한다.

나) 소프트웨어 버전관리를 위한 소프트웨어 형상관리 도구를 사용하여야 하며 선정사유를 기술한다

다) 소프트웨어 품질보증계획을 제시하고 체계개발에 적용한다.

라) 소프트웨어 품질관리를 위해 개발단계별 품질관리 기준(순환복잡도 등 품질메트릭(Metric) 적용계획 등), 자동화도구 산출물을 활용한 품질관리방안과 품질관리부서의 품질보증방안을 제안한다.

(1) 사업 착수 초기에 제안한 품질보증계획을 구체화하여 SW 품질보증계획서를 별도 제출한다.

마) CBD 개발방법론 적용시 개발단계별 산출물 작성간 문제점을 식별, 단계별 산출물간 상호연계성과 품질향상 위한 최적방안을 제안한다.

바) 성능개량 SW의 신뢰성 확인 및 데이터베이스 성능을 보장할 수 있도록 품질보증방안을 제안한다.

4) 소프트웨어 신뢰성 시험

가) 소프트웨어 신뢰성 향상을 위해 정적시험 및 동적시험을 포함한 신뢰성시험방안을 제시한다.

나) 소프트웨어 신뢰성시험은 자동화도구를 사용하여 수행하며 개발시험평가 전까지 결함을 수정하고, 개발시험평가(DT&E)항목에 신뢰성시험을 반영하여 평가한다.

다) 자동화도구에 의한 소프트웨어 신뢰성시험결과는 시험결과서 및 전산파일(DATA 및 Log)로 제출하여야 한다.

5) 소프트웨어 보안성 시험(전장관리정보체계)

- 가) 소프트웨어 보안성 확보를 위한 소프트웨어 보안성시험 방안을 제시한다.
- 나) 소프트웨어 보안성시험을 위해 행자부 소프트웨어 개발보안가이드 준수여부를 국정원 CC(Common-Criteria) 인증된 자동화 도구로 시험하여 개발시험평가 전 까지 결함을 수정하고, 개발시험평가(DT&E) 항목에 보안성시험을 반영하여 평가한다.
- 다) 자동화도구에 의한 소프트웨어 보안성시험결과는 시험결과서 및 전산 파일(DATA 및 Log)로 제출하여야 한다.

6) 소프트웨어 원천기술 식별 및 확보계획

- 가) 소프트웨어 구성품 및 단위소프트웨어별로 상용, 저작권등록, 비공개 소프트웨어 등의 원천기술을 식별하고 획득방법을 명시한다.
- 나) 제안업체가 보유한 원천기술은 증빙서류(저작권 등록증 등)를 제안서에 첨부하고, 보유하지 않은 원천기술은 수수료를 지급하거나 소프트웨어를 구매한다.
- 다) 등록되지 않은 원천기술은 등록을 전제로 확보계획을 수립한다.
- 라) 상용 소프트웨어, 저작권 등록 또는 특허 출원(등록) 소프트웨어 등 국방규격 제정 제외대상을 식별한다.
- 마) 양산을 위한 소프트웨어 사용권 확보방안을 제시한다.

7) 국산 소프트웨어 적용계획

- 가) 해외구매 등 외산소프트웨어에 대한 국산 및 외산소프트웨어 동시 적용개발 등 국산 적용방안을 제시한다.

※ 기타 소프트웨어 기능 및 개발계획·관리에 관한 내용 추가할 사항 기술

(이후 생략)

【부록 2】 구매사업 제안요청서 내 소프트웨어분야 포함항목(예)

제 안 요 청 서(안)

(OOOOO 사업)

본 내용은 방위사업관리규정 별지 제1호서식 제안요청서(구매사업)에 포함되는 소프트웨어분야 항목(예)으로서, 항공기 도입사업 사례 중에서 소프트웨어 관련부분을 발췌하여 수록하였으며, 무기체계 사업특성과 군 요구도, 획득환경 등을 고려하여 적용바랍니다.

2020. 0. 0

사업관리기관의 승인 없이 사용을 금함

※ 복제 / 복사 및 배포 금지

방위사업청 OOOO사업팀

제 4 부
제 안 사 항
(업체 제안서에 포함될 내용)

Volume IV
Request for Proposal
(Contents of Company's Proposal)

4.2. 소프트웨어 지원 : 부록 -27, 28 참조

Software Support : Refer to Appendix -27, 28

4.2.1. 소프트웨어 지원 능력

Software Support Capability

4.2.1.1. 소프트웨어 기술자료/소스코드 목록과 가격

Software technology data/source code list and price

4.2.1.2. 지원능력 구비대상 추천 소프트웨어

Recommended software for support capability

4.2.1.3. 추천 판단자료 및 제공불가 소프트웨어 목록/사유

Recommended data and non-releasable software list including its reason

4.2.1.4. 추천된 소프트웨어 지원능력 구비를 위한 제안가격

Proposal price for recommended software support capability

4.2.1.5. 제작사 기술용역 지원계획(인원, 기간, 비용)

Support plan for Manufacturer technical support services(number of persons, period, cost price)

4.2.1.6. 소프트웨어 지원(계획, 요구, 설계, 구현, 시험, 설치)과 관련된 standard 규정이 적용된 기술자료 목록

(plan, requirement, design, embodiment, examination, installation)

4.2.1.7. 소프트웨어 시험 종류 및 절차

Type of software test and procedure

4.2.1.8. 소프트웨어 관련 사후관리 지원계획

After-sales service management support plan for software

4.2.1.9. 소프트웨어 기술이전 가능분야(Radar, ATE용 정비 소프트웨어 등)

Possible technology transfer area in software (Radar, maintenance software for ATE and etc.)

4.2.1.9.1. 기술이전 가능분야(핵심 기술이전 범위 및 내용)

Possible field of technology transfer(Extent and substance of core technology transfer)

4.2.1.9.2. 로열티 및 특허 사용조건

Royalty and permission for use of patents

4.2.1.9.3. 제작사의 법적 지위 관련 서류 일체

All documents in relation to Manufacturer's legal position

4.2.1.9.4. 이전자 및 전수자의 권리와 의무

Right and duty of recipient and

4.2.2. 개발/시험 장비

Development/examination equipment

4.2.2.1. 소프트웨어 개발 및 개조장비 종류, 수량 및 가격

Type, quantity and price of Software development and modification equipment

4.2.2.2. 소프트웨어 개발 및 개조장비 운영에 필수 소요되는 보조장비 종류, 수량 및 가격

Type, quantity and price of assistance equipment required for software development and modification equipment operation

4.2.2.3. 소프트웨어 시험장비 종류, 수량 및 가격

Type, quantity and price of software test equipment

4.2.2.4. 소프트웨어 시험장비 운영에 필수 소요되는 보조장비 종류, 수량 및 가격

Type, quantity and price of assistance equipment required for software development and test equipment operation

4.2.2.5. 소프트웨어 지원(개발/시험)용 정밀측정장비 종류, 수량 및 가격

Type, quantity and price of precision measure equipment for software support(development and test)

4.2.2.6. 소프트웨어 지원용 특수 장비 종류, 수량 및 가격

Type, quantity and price of special equipment for software support (development and test)

4.2.2.7. 소프트웨어 지원용 일반 및 특수공구 종류, 수량 및 가격

Type, quantity and price of general/special tools for software support (development and test)

4.2.2.8. 추천된 장비 제원 설명자료 및 산출근거

Specification explanation data and its estimation rationale for recommended equipment

4.2.3. 소프트웨어 지원 Tool

Software support tool

4.2.3.1. 개발용 소프트웨어 및 지원 소프트웨어 Tool 종류 및 가격

Tool type and price of development software and support software

4.2.3.2. 시험용 소프트웨어 및 지원 소프트웨어 Tool 종류 및 가격

Tool type and price of test software and support software

4.2.3.3. 기타 소프트웨어 지원과 관련 한 소프트웨어 Tool 종류 및 가격

Type and price of software Tool other software support

4.2.3.4. 형상관리 Tool 종류 및 가격

Type and price of configuration management Tool

4.2.3.5. 문서 관리 Tool 종류 및 가격

Type and price of document management Tool

4.2.4. 보급지원

Supply support

4.2.4.1. 소프트웨어 지원용 장비 연간 정비 실적

Annual Maintenance record for software support equipment

4.2.4.2. 소프트웨어 지원용 주/보조/시험/특수장비에 대한 수리부속 지속 지원 계획

CSP continuous support plan for main/sub/test/special equipment for software support

4.2.4.3. 소프트웨어 지원용 주/보조/시험/특수장비의 연간 수리부속 운영 실적

CSP operating result about main/sub/test/special equipment for software support

4.2.4.4. (시간교환 품목), 시한성 품목

4.2.5. 인력운용

Personnel Management

4.2.5.1. 소프트웨어 지원을 위한 조직, 편성 및 인력소요

Organization and personnel requirements for software support

4.2.5.2. 소프트웨어 시험을 위한 조직, 편성 및 인력소요

Organization and personnel requirements for software test

4.2.6. 교육훈련

Education and training

4.2.6.1. 소프트웨어 지원을 위한 교육훈련(과정, 기간, 인원, 장소, 비용)

Education training for software support (process, period, personnel, place, cost)

4.2.6.2. 소프트웨어 지원을 위한 추천 교육훈련(과정, 기간, 인원, 장소, 비용)

Recommended education training for software support (process, period, personnel, place, cost)

4.2.6.3. 교육훈련에 소요되는 교보재 / 장비

Education aids/equipment required on education training

4.2.7. 시설

Facility

4.2.7.1. 소프트웨어 지원에 필요한 시설소요

Facility Requirement for software support

4.2.7.2. 시설별 특수 환경조건(온도, 습도, 먼지, 진동, 조명, 접지, 소음 등)

Special environment condition (temperature, humidity, dust, shock, lighting, ground connection, noise etc.)

4.2.7.3. 시설별 전원 소요(전력, 비상발전기, UPS, 직류/교류, 전원 Cycle 등)

Power requirement for facility (electric power, emergency generator, UPS, DC/AC, Power Cycle etc)

4.2.7.4. 시설별 특수 요구시설(출입문 크기, 레이더 설치유무, 무반 향실 등)

Special requirement facility (entrance size, radar establishment existence and nonexistence, radish echo chamber etc.)

4.2.7.5. 안전 대책

Safety measures

부록 LK-27 소프트웨어 지원

Appendix LK-27 Software Support

1. 소프트웨어 지원

Software Support Capability

1.1. 소프트웨어 지원능력 개념 및 체계

Concept and System of software support capability

1.1.1. 운용프로그램 리스트

Operating Program list

1.1.2. 소프트웨어 리스트

OOO software (OFP/MDF) list

1.1.3. 운용 프로그램 소스코드 가격

Price of Operating Program Source code

1.1.4. OOO 소스코드 무상지원 불가시 가격

Price of source code when not provided for free of charge

1.1.5. 임무계획체계 소스코드무상지원 불가 시 가격

Price of mission plan system source code when not provided for free of charge

1.1.6. ATE 소스코드 무상지원 불가시 가격

Price of ATE source code when not provided for free of charge

1.1.6. ATE 소스코드 무상지원 불가시 가격

Price of ATE source code when not provided for free of charge

1.1.7. 제작국 공군의 소프트웨어(OFP, EWSP(OFP/MDF), 임무지원체계, ATE) 지원능력 및 사유

Software (OFP, EWSP(OFP/MDF), mission support system, ,ATE) support capability of Air Force of manufacturer's country

1.1.8. 비행운용 프로그램(OFP), EWSP 프로그램, 임무지원체계 소프트웨어, ATE 소프트웨어 지원을 위한 추천 소프트웨어

Recommended software support for OFPs, EWSP program, mission support system software, recommended software for ATE software support.

1.1.8.1. 비행운용 프로그램(OFP), EWSP 프로그램, 임무지원체계 소프트웨어, ATE 소프트웨어 추천 판단자료 및 제공불가 소프트웨어와 불가사유

Justification for recommendation or non-recommendation for OFPs, EWSP program, mission support system software, recommended software for ATE software support.

1.1.9. 소프트웨어 기술이전 가능성(OFP, EWSP(OFP/MDF), 임무지원체계, IFF, ATE용 정비 소프트웨어)

Possibility of software technology transfer (OFP, EWSP(OFP/MDF), mission support system, IFF, maintenance software for ATE)

1.1.10. EWSP 전자전 위협자료 제공 가능성

Probability of providing EW threat data

1.2. 소프트웨어 지원능력 업무량 추정 및 분석자료

Estimated work load and analysis data of software support capability

1.2.1. 추천된 소프트웨어 지원능력 구비를 위한 제안가격(OFP, EWSP, 임무지원체계, ATE)

Proposed price for recommended software support capability (OFP, EWSP, misstion support system, ATE)

2. 지원/시험 장비

Maintenance/Test equipment of EWSP(OFP/MDF)/Mission Support System/ATE software

2.1. 소프트웨어 지원용 개발/개조장비 종류, 수량 및 가격

Type, quantity and cost of software maintenance equipment

2.2. 소프트웨어 개발/개조장비 운영에 필수 소요되는 보조장비(보조기억 장비 등) 종류/수량/가격

Type, quantity, cost of auxiliary software for software development/modification equipment operation (secondary storage devices, etc.).

2.3. 소프트웨어 지원용 시험장비 종류, 수량 및 가격

Type, quantity and cost of software test equipments

2.4. 소프트웨어 시험장비 운영에 필수 소요되는 보조장비(보조기억장비 등) 종류/수량/가격

Type, quantity and cost of auxiliary software required for software test equipment operation (secondary storage etc.).

2.5. 소프트웨어 지원(개발/시험)용 정밀측정 장비 종류, 수량 및 가격

Type, quantity and cost of PME for maintenance and test

2.6. 소프트웨어 지원용 특수장비(물자취급 장비 등) 종류, 수량 및 가격

Type, quantity and cost of special equipments for maintenance and test

2.7. 소프트웨어 지원용 일반/특수공구 종류, 수량 및 가격

Type, quantity and cost of general/special tools for maintenance and test

2.8. 장비 제원 설명자료(크기, 중량, 소모전력, 소음크기, 발열량, 제열방법, 일반장비와 호환성, Power backup 필요성/요구시간 등) 및 설계도면

Equipment characteristics(size, weight, power consumption, noise strength, heat radiation, cooling method, interoperability with commercial equipment, necessity of power backup and required power backup time etc.) and drawing for maintenance and test

2.9. 소요장비 산출 자료

Calculation method and data of equipment recommendation

3. 소프트웨어 지원 Tools

Software support tools

3.1. 개발용 소프트웨어 및 지원 소프트웨어 tool 종류/가격

Type/Cost of support software tool for developing OFP/ EWSP(OFP/MDF) software

3.2. 시험용 소프트웨어 및 지원 소프트웨어 tool 종류/가격

Type/Cost of support software tool for testing OFP/ EWSP(OFP/MDF) software

3.3. 형상관리(Configuration Management) tool 종류/가격

Type/Cost of software tool for configuration management of OFP/ EWSP(OFP/MDF) software

3.4. 문서관리(Documentation) tool 종류/가격

Type/Cost of software tool for documentation of OFP/ EWSP(OFP/MDF) software

3.5. 기타 소프트웨어 종류/가격

(Schedule/Cost Estimate tool, networking, file transform 등)

Type/Cost of other software tools for supporting OFP/ EWSP(OFP/MDF) software development (schedule/cost estimating, networking and file transform etc.)

4. 보급지원

Supply support

4.1. 소프트웨어 지원용 장비 연간 정비실적

Annual maintenance status of software support equipment(maintenance/test)

4.2. 소프트웨어 지원용 특수제작 주요장비/보조/시험/특수장비에 대한 수리부속 지속지원 계획(업체의 보급지원 보장대책)

Spare part support plan(including manufacturer's guarantee) of non off-the-shelf main/auxiliary/test/special equipment

4.3. 소프트웨어 지원능력용 주요장비/보조/시험/특수장비의 연간 수리부속 list 및 운영실적

Annual operational status and list of spare parts for main / auxiliary / test / special equipment

4.4. 시한성, 시효성 품목현황

List of Self-life and Service life items

5. 인력운영

EWSP(OFP/MDF)/Mission Support System/ATE Personnel Operation

5.1. 소프트웨어 지원을 위한 조직 및 편성(OFP, EWSP(OFP/MDF))

Organization of software support capability(tree structure) for OFP, EWSP (OFP/MDF)

5.2. 소프트웨어 지원을 위한 인력소요(OFP, EWSP(OFP/MDF))

Required number of persons to software support for OFP, EWSP(OFP/MDF)

6. 교육훈련 및 교보재

Training and Training equipment/matrical & aids

6.1. 소프트웨어 지원을 위한 교육과정(교육분야, 소요기간, 인원, 장소, 비용)

Training curriculum(program, period, personnel, place, cost) for software support

6.2. 소프트웨어 지원을 위한 추천 교육계획 (교육분야, 소요기간, 인원, 장소, 비용)

Recommended training program(program, period, personnel, place, cost) for OFP & EWSP(OFP/MDF), Misson Support System, ATE software support capability)

6.3. 교육훈련에 소요되는 교보재/장비

Required equipment/manual for education and training

7. 소프트웨어 지원(설계, 구현, 시험)과 관련된 기술자료 목록

EWSP(OFP/MDF)/Mission Support System/ATE Software development documents list(design, implementation, test etc.)

8. 시설

Facilities

8.1. 소프트웨어 지원능력에 필요한 정비/시험시설 소요

Required maintenance/test facility for software support capability

8.1.1. 도면(크기)

Blueprint(size)

8.1.2. 정비환경 : 온도, 습도, 환기, 먼지, 조명, 진동, 소음, 전자장, 접지 등

Maintenance environment : temperature, humidity, dust, ventilation, light, vibration, noise, electromagnetic, grounding etc.(including specification)

8.1.3. 소요 전원 : 전력, 비상발전기, UPS, 1f/3f, 직류/교류, 전원 Cycle 등

Power requirement : Total required power(AC/DC), emergency generator, UPS, power phase(1f/3f), power cycle etc. (including specification)

8.1.4. 장비 배치도

Equipment Layout

8.1.5. 안전대책

Safety considerations

8.1.6. 출입문 크기

Gate Size

8.1.7. 특수시설(무반향실 등)

Special facilities(anechoic chamber etc.)

8.1.8. 기타 필요사항(물, air, 각종 가스 등)

Other requirements(water, air, gas etc.)

<부표/Attachment>

소프트웨어 지원능력(OFP, EWSP(OFP/MDF), IFF) 개념 및 체계

EWSP(OFP/MDF)/Mission Support System/ATE Software support capability
Concept and System(OFP, EWSP, IFF)

CSCI ①	Size ②	기 능 Function n	Cost	제작국 운영군 Original service force③	추 천 Recommend ed④	근 거 Justification ⑤

① CSCI 명칭

CSCI noun

② CSCI 크기(Kilo Byte)

CSCI size(Kilo Byte)

③ 제작국 운영군 소프트웨어 지원능력가능표시(V)

Software support capability from original service force (Mark 'V'
if applicable)

④ 추천된 소프트웨어 지원능력 능력 표시(V)

Recommended software support capability(mark 'V')

⑤ 추천된 이유, 미 추천된 이유(예, 경제성, 비밀 등)

Reason/Justificaion for recommendation or non-recommendation
(ex, economical, government restrict etc.)

※ CSCI : Computer Software Configuration Item

부록 LK-28 소프트웨어장비

Appendix LK-28 Software Equipment

1. 자료요구 목적 및 범위

Description/Purpose

본 자료는 부대 정비 단계 및 야전 정비단계에서 대상 무기체계를 운영 유지하기 위하여 필요한 운영 소프트웨어 지원 장비 및 수명주기 기간 소프트웨어 유지보수 및 성능개량을 대한민국 공군이 독자적으로 수행하기 위한 소프트웨어 개발/시험/지원 장비 내역 및 가격을 기술하는 것이다.

This is to describe the list and price of software development/test/support equipment in order to support ROKAF's independent operation for maintaining the proposed system operation S/W at the stage of organizational maintenance and intermediate maintenance and software maintenance/enhancement capability during the life cycle of the proposed system

2. 양식 및 작성 요령

Format and Content Instructions

아래에서 제시된 양식에 맞추어 목록을 제시하여야 한다. 단, 아래 양식의 내용이 모두 포함된 업체 나름의 양식 존재한다면 아래의 양식은 업체 고유의 양식에 의하여 대체될 수 있다.

Describe using the following format. However, if the proposing company has its own format that includes the following contents, the proposing company may replace the following format with its own.

LK-28-1 S/W 운영 지원장비 목록

Support Equipment List for S/W Operation

○ 구분 Classification

차 례	품 명	재 고 번 호	부 분 품 번 호	모 델	불 출 단 위	가 격	정 비 수 준	제 작 사	제 작 사 국 적	구 매 조 달 원	정 비 원	기 능	계 통	구 매 조 달 기 간	정 비 조 달 기 간	연 간 소 요		20년 간 소 요	추 천 량	추 천 사 유	연 간 사 용 빈 도	IC & D	크 기	비 고	
																(q)	(r)								
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	수 량	금 액	수 량	금 액	(s)	(t)	(u)	(v)	(w)	(x)
A																									

* Excel 자료로 별도 제출

Submit separately in EXCEL format

* 각 장비별 사진 목록 제출(별지 LK-7-1 참조)

Submit list of photographs per each equipment(Refer to Attachment LK-7-1)

LK-28-2 S/W 능력구비 지원장비 목록

Development/Test/Support Equipment List for S/W Maintenance

○ 구분 Classification

차 례	품 명	재 고 번 호	부 분 품 번 호	모 델	불 출 단 위	가 격	정 비 수 준	제 작 사	제 작 사 국 적	구 매 조 달 원	정 비 원	기 능	계 통	구 매 조 달 기 간	정 비 조 달 기 간	연 간 소 요(q)		20년 간 소 요(r)		추 천 량	추 천 사 유	연 간 사 용 빈 도	IC & D	크 기	비 고
																수 량	금 액	수 량	금 액						
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)					(s)	(t)	(u)	(v)	(w)	(x)
A																									

* Excel 자료로 별도 제출

Submit separately in EXCEL format

* 각 장비별 사진 목록 제출(별지 LK-7-1 참조)

Submit list of photographs per each equipment(Refer to Attachment LK-7-1)

(a) 차례 : A1, A2, ..., A10, ...

Number : A1, A2, ..., A10, ...

(b) 품명 : 해당장비의 명칭

Name : Name of concerned equipment

(c) 재고번호 : 국가재고번호(또는 나토재고번호)

NSN : National Stock Number(or NATO Stock Number, whichever is applicable)

(d) 부분품번호 : 제작 부분품 번호

Part Number : Part Number of manufactured component

(e) 모델 : 해당품목의 모델명

Model : Model of concerned item

(f) 불출단위 : 구매를 위한 최소 단위

Unit of Issue : Minimal Unit for purchase

(g) 가격 : 불출단위에 따른 가격

Unit Price : Price per Unit of Issue

(h) 정비수준 : 해당 장비를 사용하는 정비수준

부대정비 : O, 야전정비 : I,

Maintenance Level : Maintenance level which the concerned equipment is used in Organization : O, Intermediate : I

(i) 제작사 : 해당 품목의 제작사

Manufacturer : Manufacturer of the concerned equipment

(j) 제작사 국가 : 해당 품목 제작사의 국적

Manufacturer's Country : Nationality of manufacturer

(k) 구매조달원 : 해당 품목 구매를 위한 조달업체 및 국가

Source of Procurement : Manufacturer and country selling the concerned equipment

(l) 정비원 : 해당품목 정비를 위한 정비업체 및 국가

Source of Maintenance : Facility and country which performs maintenance for the concerned equipment

(m) 기능 : 해당품목의 주요기능

Function : Major functions of the concerned equipment

(n) 계통 : 주요기능이 포함된 부류

기체 : F, 항전 : A, 계기 : I, 유압 : H, 통신 : C, 전기 : E, 기타 : O

Systematic Classification : Category which the concerned equipment is used in
Fuselage : F, Avionics : A, Instrumentation : I, Hydraulics : H, Communication : C, Electric : E, Others : O

(o) 구매조달기간 : 구매 신청후 선적까지 소요되는 예측기간

Procurement Period : Estimated period from ordering until shipment

(p) 정비조달기간 : 정비를 위해정비원에 도착부터 정비후 선적까지 예상기간

Maintenance Period : Estimated period from arrival at Source of Maintenance until shipment after maintenance is completed

(q) 연간 소요 : 해당 품명의 연간 예측 수량 및 금액

Annual Requirement : Estimated quantity and price of the concerned equipment on a yearly basis

(r) 20년간 소요 : 해당 품명의 20년간 예측 수량 및 금액

Requirement for twenty(20) years : Estimated quantity and price of the concerned equipment for twenty(20) years

(s) 추천량 : 제작사 추천량

Recommended Quantity : Manufacturer's recommended quantity

(t) 추천사유 : 제작사 추천사유, 추천방법

Justification : Reason for and method of recommendation

(u) 연간사용빈도 : 연간 해당장비의 사용빈도

Annual Usage Frequency : Used frequency of the concerned equipment on a yearly basis

(v) IC&D : 설치 및 사용시범 필요여부(필요시 무상, 가격기록)

IC&D : Whether or not Installation Checkout & Demonstration(IC&D) is necessary for the concerned equipment(if necessary, support by free of charge and fill in with its cost)

(w) 크기 : 가로 × 세로 × 높이, 중량

Dimensions : Width × Length × Height, Weight

개발단계별 산출물 점검표

【체크리스트 작성을 위한 사전 준비】

1. 점검결과 표시 방법

양호, 보통, 미흡, N/A(점검항목 해당사항 없음)

2. 기술문서 검토 전 사전 확인 문서

검토 문서	사전 확인 문서
① SW개발계획서	체계개발실행계획서, 업체 비용분석서
② SW요구사항명세서	SW개발계획서, 체계설계기술서
③ SW설계기술서	SW요구사항명세서
④ SW통합시험절차서	SW설계기술서
⑤ SW통합시험결과서	SW통합시험절차서
⑥ SW산출물명세서	소스코드 및 실행파일 목록

① 소프트웨어개발계획서

장절	점검항목	점검결과	점검의견
전체	전체 체계별 1건으로 작성하였는가? 체계별 분리 필요성이 있으면 이를 검토한 후 승인		
1.2	<체계운용개념도> 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부문서/비정부문서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3.2	<SW기술문서 작성대상> SW기술문서 작성대상이 타당한가?		
4.2.1	<SW개발방법론> 1) 소프트웨어 형상항목(CSCI)별 적절한 개발방법론 및 자동화 도구를 사용하였는가?		
	2) 개발방법론 및 자동화 도구를 선택한 사유를 기술하였는가?		
4.2.2.3	<분류체계 식별자 부여 규칙> 개발단계별 분류체계 식별자 적용활동을 기술하였는가?		
4.2.2.4	<코딩 규칙> 매뉴얼 부록의 소프트웨어 코딩 규칙에 대한 적용 계획과 체계의 특성을 고려하여 가장 적합한 관련표준 코딩규칙 추가 적용계획을 기술하였는가?		
	소프트웨어 코딩시 품질메트릭 적용 계획을 제시하였 는가?		
4.2.3	<재사용 가능한 소프트웨어 산출물> 1) 신규개발, 수정하여 개발(modify), 재사용(reuse) 범위를 구분하여 제시하였는가?		
	2) 재사용 소프트웨어(공개 SW 포함)가 신뢰성이 있는지 검토하였는가?		
	3) SW 개발시 향후 재사용이 가능하도록 계획을 수립 하였는가?		
4.2.4	<SW신뢰성/보안성 시험> SW신뢰성 및 보안성 확보활동이 개발단계별로 기술되 었는가?		
4.2.6.1	<국외도입 SW 식별> 국외도입 SW를 식별 및 사유를 기술하였는가?		

장절	점검항목	점검결과	점검의견
4.2.7	<컴퓨터 하드웨어 자원 활용> 1) 컴퓨터 하드웨어 자원을 세부장비별 기술하였는가?		
	2) 사용한 하드웨어 자원이 요즘 많이 사용되는 자원으로 단종 예정은 없는지 확인하였는가?		
5	<세부적인 소프트웨어 개발활동에 대한 계획> 세부적인 소프트웨어 개발활동 계획을 일반적으로 기술한 것이 아니라 실제 개발하려고 하는 체계에 맞게 구체적(대상, 방법)으로 계획되었는가?		
5.2	<소프트웨어 개발환경의 구축> 1) 소프트웨어 개발환경이 적절한가?		
	2) 소프트웨어 시험환경이 해당 체계에 맞게 구체적 (대상, 방법 모두 포함)으로 기술되었는가?		
5.3 ~ 5.12	항목별 해당 체계에 맞게 구체적(대상, 방법 모두 포함)으로 기술되었는가?		
5.13	<형상관리> 소프트웨어 산출물에 대하여 형상관리 및 변경관리 계획이 적절한가?		
5.15	<품질관리> 개발부서의 품질관리계획 및 품질보증(QA)부서의 품질을 보증하기 위한 계획, 측정방법, 통제방법이 일반적으로 기술한 것이 아니라 해당 체계 및 업체에 맞게 구체적(대상, 방법 모두 포함)으로 기술되었는가?		
6	<일정계획 및 활동계획표> 소프트웨어 검토회의를 포함한 SW개발일정 계획이 사업기간 중에 수행 가능하도록 수립되었는가?		
7	<사업조직과 자원> SW개발에 참여한 협력업체와 협력업체 개발범위가 기술되었는가?		

② 소프트웨어요구사항명세서

장절	점검항목	점검결과	점검의견
전체	소프트웨어 형상항목(CSCI)별로 작성되었는가?		
1.2	<체계개요> 1) 전체 체계에 대한 설명 후 해당 소프트웨어 형상항목(CSCI)이 탑재되는 체계에 대한 설명이 추가되었는가?		
	2) 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부문서/비정부문서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3	<요구사항> 요구사항식별자 명칭 구성이 문서식별자-장비명-기능구분-일련번호인가?		
3.2	체계/부체계 요구사항 및 규격서의 내용과 비교하여 SW분야로 분석된 요구사항이 누락된 부분 없이 적절한가? SW 요구사항은 시험이 가능한 수준으로 세분화되어 있는가? * 시험이 가능한 수준 : 기능 시험 뿐만 아니라 신뢰성 시험(단위시험 시 커버리지 확인)용 테스트케이스를 만들 수 있을 정도로 세분화		
3.3	<외부 인터페이스 요구사항> 1) 체계/부체계 요구사항에서 제시하는 외부 인터페이스 관련사항이 SW 외부 인터페이스 요구사항에 반영되었는가?		
	2) 소프트웨어 형상항목(CSCI)별 외부 인터페이스의 일관성이 있는가?		
3.4	<내부 인터페이스 요구사항> 1) 소프트웨어 형상항목(CSCI) 내부 기능간 제어흐름과 데이터흐름에 대한 요구사항이 반영되었는가?		
	2) 사용자인터페이스 요구사항이 식별되었는가?		
3.9	<SW 품질 요소> SW 신뢰성 및 보안성 시험을 위한 요구사항 등 SW 품질 요소(신뢰성, 보안성, 기능적합성, 유지보수성, 사용성, 효율성, 이식성, 호환성)에 대한 요구사항이 식별되었는가?		

장절	점검항목	점검결과	점검의견
3.5 ~ 3.16	해당 소프트웨어 형상항목(CSCI)에 맞게 요구사항이 구체적(대상, 방법 모두 포함)으로 작성되었는가?		
4	<개발시험 평가 조항> 요구사항식별자별 자격부여방법이 적절한가?		
5	<요구사항 추적성> 체계/부체계 요구사항과 SW 설계기술서 간 양방향 추적이 가능한가?		

③ 소프트웨어설계기술서

장절	점검항목	점검결과	점검의견
전체	소프트웨어 형상항목(CSCI)별로 작성되었는가?		
전체	설계단계에서는 문서에 TBD로 기술할 수 있으나 규격화 시점에서는 모두 구체화하여야 한다.		
1.2	<체계개요> 1) 전체 체계에 대한 설명 후 해당 소프트웨어 형상항목(CSCI)이 탑재되는 체계에 대한 설명이 추가되었는가?		
	2) 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부문서/비정부문서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3	<SW설계시 고려사항> 1) 일반적인 문구가 아니라 대상 소프트웨어 형상항목 (CSCI)에 맞게 내용이 구체적으로 작성되었는가?		
	2) SW개발계획서와 상충되는 부분이 있는가?		
3.3	<상용제품 활용방안> 상용제품 활용방안이 구체적으로 제시되었는가?		
3.7	요구사항에 대한 설계결정사항이 기술되어 있는가?		
4	1) HW/SW 연계한 구조도가 기술되어 있는가?		
	2) 소프트웨어 형상항목(CSCI) 구조도는 CSC-CSU 계층으로 구분되어 있는가?		
	3) 마이크로 프로세서 탑재형 SW 및 HDL을 사용하는 PLD 탑재형 SW로 구분되어 있는가?		
	* 개발자와 통화하여 PLD 탑재형 SW가 모두 식별되었는지 확인 필요		
	* PLD 탑재형 SW 경우 설계도면을 사용하였으면 SDD에 기록하였는지 확인 필요		
4장 전체	1) 설계 내용이 검토자가 이해할 수 있는 수준으로 충분히 구체화되었는가?		
	2) 용어 및 내용이 전체적으로 일관성이 있는가?		
	3) 기술 순서는 CSC-CSU-설계식별자 순으로 되어 있는가? * 사업의 규모에 따라 CSU와 설계식별자는 동일 레벨일 수 있음		
4.3	<인터페이스 설계> 1) 체계 내/외부 인터페이스가 명확히 식별되었는가?		

장절	점검항목	점검결과	점검의견
	2) 인터페이스 구성도는 이해하기 쉽게 그려졌는가?		
	3) 인터페이스 설계 내용이 검토자가 이해할 수 있는 수준으로 충분히 구체화되었는가?		
	4) 내용 중 누락된 부분은 없는가?		
	5) 사용된 연동통제문서는 정부/비정부문서에 기록되어 있는가? * 이 문서는 규격화시 납품되는지 확인하여야 함		
4.4	<데이터베이스 설계> 1) 데이터베이스 설계 내용이 검토자가 이해할 수 있는 수준으로 충분히 구체화되었는가?		
	2) 데이터베이스 각 필드별 속성이 정의되어 있는가?		
5장 전체	<SW 형상항목 상세설계> 1) 상세설계 내용이 검토자가 이해할 수 있는 수준으로 충분히 구체화되었는가?		
	2) 용어 및 내용이 전체적으로 일관성이 있는가?		
	3) 기술 순서는 CSC-CSU-설계식별자 순으로 되어 있는가? * 사업의 규모에 따라 CSU와 설계식별자는 동일 레벨일 수 있음		
	4) 함수 목록에는 개발한 모든 함수명이 기록되어 있는가?		
6	<추적성> SW 요구사항과 SW 시험 간 양방향 추적이 가능한가?		

④ 소프트웨어통합시험절차서

장절	점검항목	점검결과	점검의견
전체	소프트웨어 형상항목(CSCI)별로 작성되었는가?		
1.2	<체계개요> 1) 전체 체계에 대한 설명 후 해당 소프트웨어 형상항목(CSCI)이 탑재되는 체계에 대한 설명이 추가되었는가?		
	2) 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부분서/비정부분서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3	<시험 개요> 시험 목적, 범위, 장소, 일정, 조직, 방법 등이 구체적으로 기술되었는가?		
5.2.6	1) 시험절차가 요구사항 및 설계를 만족시키는가?		
5.x ~ y	-소프트웨어 신뢰성시험(정적, 동적) 및 보안성시험(전장관리정보체계) 절차가 구체적으로 기술되었는가?		
6	<요구사항 추적성> SW 시험절차서 식별자는 양방향 추적이 가능한가?		

⑤ 소프트웨어통합시험결과서

장절	점검항목	점검결과	점검의견
전체	소프트웨어 형상항목(CSCI)별로 작성되었는가?		
1.2	<체계개요> 1) 전체 체계에 대한 설명 후 해당 소프트웨어 형상항목(CSCI)이 탑재되는 체계에 대한 설명이 추가되었는가?		
	2) 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부문서/비정부문서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3	소프트웨어통합 일정, 결과가 기술되어 있는가?		
4	소프트웨어 통합시험 목적, 범위, 장소, 일정, 조직, 방법이 구체적으로 기술되어 있는가?		
5	소프트웨어 통합시험결과에 대한 개요가 충실히 기술되어 있는가?		
6	1) 각 소프트웨어요구사항에 대한 통합시험결과가 기술되어 있는가?		
	2) SW 신뢰성 및 보안성 시험 결과가 포함되었는가?		
7	소프트웨어요구사항명세서(상위 요구사항) 및 소프트웨어설계기술서(하위 요구사항)에 명시된 요구사항이 100% 구현되어 시험을 수행하였는지 분석 결과를 포함하였는가?		
부록	<시험일지> 시험 일시, 장소 및 시험 입회자 서명 등을 확인하여 실제 시험 일자와 일치하는지 확인하였는가?		

⑥ 소프트웨어산출물명세서

장절	점검항목	점검결과	점검의견
전체	소프트웨어 형상항목(CSCI)별로 작성되었는가?		
1.2	<체계개요> 1) 전체 체계에 대한 설명 후 해당 소프트웨어 형상항목(CSCI)이 탑재되는 체계에 대한 설명이 추가되었는가?		
	2) 전체 체계운용개념도를 기준으로 해당 소프트웨어 형상항목(CSCI)이 식별되는 그림을 그렸는가?		
2.1	<정부분서/비정부분서> SW개발시 참조한 연동통제문서(ICD)의 목록이 해당 문서목록에 기술되었는가?		
3	디바이스 목록의 도면번호와 부품번호는 실제 자료와 일치하는가? - 도면번호 : 해당 CSCI가 탑재되는 장비의 도면번호 - 부품번호 : 해당 CSCI가 탑재되는 장비의 HW 부품번호		
3.1	실행파일 목록은 제출된 실행파일과 일치하는가? (파일명, 크기, 체크섬, 생성일 등 각 항목별 확인 필요)		
	프로젝트 파일 등 컴파일시 필요한 각종 환경파일 및 목록이 제출되었는가?		
3.2	원본(소스)파일 목록은 제출된 원본(소스)파일과 일치하는가? (파일명, 크기, 체크섬, 생성일 등 각 항목별 확인 필요)		
4	소프트웨어 생성 및 수정절차가 실제 절차를 이미지로 넣어서 작업자가 이해할 수 있을 정도로 쉽게 구성되어 있는가?		
5	<추적성> 요구사항부터 소스코드까지 양방향 추적성이 확보되었는가?		

형상관리 및 품질보증 점검표

【체크리스트 작성을 위한 사전 준비】

1. 점검결과 표시 방법

양호, 보통, 미흡, N/A(점검항목 해당사항 없음)

2. 점검 주기

- PDR, CDR : 개발기관 자체 점검을 통한 체크리스트 제출
- 구현단계 : 개발기관 실사시 체크리스트 활용 점검

3. 프로세스 점검 대상

프로세스	점검항목
① 형상관리	세부항목 참조
② 품질관리	세부항목 참조

① 형상관리

분야	점검항목	점검 결과	점검의견/담당자
형상항목 식별	소프트웨어 형상관리 계획에 의거하여 형상항목 및 작업 산출물을 선정한다.		
	고유한 식별자 코드를 형상항목에 부여한다.(형상번호)		
	형상항목의 중요한 특성을 명시한다.(변경대상, 고객 제출여부 등)		
	형상항목의 책임자를 식별한다.		
형상관리 시스템	소프트웨어 작업 산출물의 통제를 위해 형상관리 및 변경 관리 시스템을 관리한다.		
형상 기준선	소프트웨어 형상항목의 형상 기준선을 생성하거나 이관하기 전에 형상통제위원회의 승인을 받은 후 이관한다.		
형상항목 변경 통제	전체 수명주기 전반에 걸쳐 소프트웨어 형상항목의 변경을 통제한다.		
	변경한 소프트웨어 형상항목을 형상관리시스템에 반영하기 전에 적절한 승인을 얻는다.		
	소프트웨어 형상항목의 변경과 그 변경 사유를 기록한다.		

분야	점검항목	점검 결과	점검의견/담당자
형상 확인	소프트웨어 형상 기준선의 무결성을 평가한다.		
	소프트웨어 형상 기록이 형상항목의 현상을 올바르게 식별하고 있는지를 확인한다.		
	소프트웨어 형상관리시스템 내에 있는 항목들의 구조와 무결성을 검토한다.		
	소프트웨어 형상관리시스템 내에 있는 항목들의 완전성과 정확성을 확인한다.		
	소프트웨어 형상관리 표준 및 절차가 적용상에 문제가 없는지를 확인한다.		
	소프트웨어 형상확인 결과 도출된 부적합사항에 대하여 완료시까지 추적한다.		

② 품질보증

장절	점검항목	점검 결과	점검의견/담당자
프로세스 평가	소프트웨어 개발 관련 프로세스 평가에 대한 기준을 명확하게 설정하고 관리한다.(평가 대상, 시기, 주기, 방법, 담당자)		
	수행한 프로세스가 명시된 기준에 따라 프로세스 정의서, 표준, 절차를 준수하는지를 평가한다.		
	평가 중에 발견된 부적합사항을 식별한다.		
	소프트웨어 품질 평가결과를 기록하고 문서화한다.		
	향후 타 체계 개발에 대한 프로세스를 개선시킬 수 있는 교훈(Lessons Learned)을 식별한다.		
작업 산출물 평가	평가할 소프트웨어 작업 산출물을 선정한다.		
	소프트웨어 작업 산출물 평가에 대한 기준을 명확하게 설정하고 관리한다.(평가 대상, 시기, 주기, 방법, 담당자)		
	소프트웨어 작업 산출물의 평가기준은 프로세스에 명시된 절차와 표준에 부합하는지의 여부를 포함한다.		
	명시된 기준에 따라 획득자에게 인도되기 이전에 소프트웨어 작업 산출물을 평가한다.		
	개발 단계의 주요 마일스톤에 소프트웨어 작업 산출물을 평가한다.		
	평가방법은 검토, 테스트, 감사 등의 활동을 활용한다.		
	평가 중에 발견한 부적합사항을 식별한다.		

장절	점검항목	점검 결과	점검의견/담당자
	품질 평가결과를 기록하고 문서화한다.		
	향후 타 개발에 대한 작업 산출물 평가를 개선시킬 수 있는 교훈(Lessons Learned)을 식별한다.		
부적합 이슈 해결	소프트웨어 품질 보증 중 부적합사항의 이슈를 제기한다.		
	사업 내에서 해결하지 못한 부적합사항의 이슈를 조직차원의 관리자에게 보고하여 의사결정을 받는다.		
	관련된 이해관계자가 평가 및 품질의 결과를 적시에 인지할 수 있도록 한다.		
	이슈에 대해 조치 담당자를 할당하여 해결하도록 한다.		
	이슈를 해결할 때까지 추적한다.		
	해결된 이슈에 대해 시정조치 결과를 기록하여 관리한다.		
품질보증 활동 기록	소프트웨어 개발 프로세스 및 작업 산출물에 대한 품질보증 활동의 상태 및 결과를 충분히 알 수 있도록 구체적으로 기록한다.		
	필요한 경우 소프트웨어 품질보증 활동의 상태 및 이력을 개정한다.		
협력업체 관리	소프트웨어 개발 관련 협력업체를 선정 후 계약을 체결한다.		
	계약에서 정의된 협력업체의 진행사항과 성과(일정, 공수, 비용, 기술적 성능)를 모니터링하고 문서화한다.		

【부록 5】 국방규격화 검토용 점검표

국방규격화 검토용 점검표

(SW기술문서)

【점검방법】

- ▶ 국방규격 제·개정시 점검순서에 따라 점검표를 이용하여 점검한다.
- ▶ 국방규격 개정 시에는 형상 변경된 사항에 대해서만 점검을 수행할 수 있다.
- ▶ 각 점검 항목에 대하여 점검내용에 따라 점검결과를 기록한다. (예시 참조)
- ▶ 각 점검 항목의 점검결과에 대해 양호/보완/미흡 등으로 확인한다. 보완/미흡으로 식별된 항목에 대해서는 사유를 별도로 제시하고, 이를 보완하여 재점검을 실시한다.

【점검순서】

순 서	점검 항목	자료명
①	국방규격 작성 근거 점검	계약문서 및 법령
②	CSCI 식별 및 제출자료 점검	SCS 및 제출자료(국방규격서 등)
③	SW기술문서 점검 (CSCI별)	SRS, SDD, STD, STR, SPS 등
④	SW 산출물 점검 (CSCI별)	SPS
⑤	도면 연계 사항 점검 (CSCI별)	SPS, 도면
⑥	소프트웨어 목록 점검	SCS

① 국방규격 작성 근거 점검표

점검항목	점검내용	확인
국방규격 작성 근거 법령	▶ 검토할 국방규격 작성에 대한 근거 법령(발령일)을 확인하였는가?	
	▶ 점검결과: (예시) ○ 방위사업청 훈령 제000호 (201X.00.00) 방위사업관리규정 ○ 방위사업청 예규 제000호 (201X.00.00) 표준화 업무지침 ○ 방위사업청 예규 제000호 (201X.00.00) 국방규격표준서의 서식 및 작성에 관한 지침 ○ 방위사업청 매뉴얼 제000호 (201X.00.00) 무기체계 소프트웨어 개발 및 관리 매뉴얼	
규격화 대상	▶ 검토할 국방규격의 소프트웨어 기술문서 규격화 범위를 확인하였는가? ☞ 참고 • 방위사업관리규정 제38조에 따라 소프트웨어 기술문서는 관련지침에 따라 작성하되, 사업관리부서장은 작성 대상이 되는 소프트웨어 기술문서의 범위를 조정할 수 있음 • 소프트웨어 기술문서 작성 대상은 체계개발실행계획서 또는 소프트웨어개발계획서(SDP)의 산출물 목록으로 제시됨	
	▶ 점검결과: (예시) ○ 규격화 대상 소프트웨어 기술문서 - 소프트웨어 목록 명세서 (SCS) - 각 CSCI별 소프트웨어 요구사항 명세서 (SRS) - 각 CSCI별 소프트웨어 설계 기술서 (SDD) - 각 CSCI별 소프트웨어통합시험절차서 (STD) - 각 CSCI별 소프트웨어통합시험결과서 (STR) - 각 CSCI별 소프트웨어 산출물 명세서 (SPS)	

② CSCI 식별 및 자료제출 점검표

점검항목	점검내용	확인																									
체계 구성 및 CSCI 식별	▶ 체계 구성 및 체계에 포함된 CSCI를 식별하였는가? ☞ 참고 • 체계구성 및 CSCI 구조도는 SCS에 포함하여야 함																										
	▶ 점검결과: (예시) ○ 총 2개의 CSCI로 식별되었음 (SCS 3장 체계구조도 참조) <table><tr><td>CSCI</td><td>도면번호</td></tr><tr><td>가</td><td>40002400</td></tr><tr><td>나</td><td>40002410</td></tr></table>		CSCI	도면번호	가	40002400	나	40002410																			
	CSCI		도면번호																								
가	40002400																										
나	40002410																										
기술자료 번호 구성	▶ 모든 소프트웨어 기술자료에 대하여 기술자료 번호를 구성하였는가? ☞ 참고 • CSCI별로 <도면번호 + 영문약어> 형태로 기술자료 번호를 구성하여야 함 (국방규격·표준서의 서식 및 작성에 관한 지침 참조)																										
	▶ 점검결과: (예시) ○ 기술자료 번호 구성 <table><tr><td>CSCI</td><td>기술자료 번호</td><td>설명</td></tr><tr><td rowspan="5">가</td><td>40002400SRS</td><td>소프트웨어 요구사항명세서</td></tr><tr><td>40002400SDD</td><td>소프트웨어 설계기술서</td></tr><tr><td>40002400STD</td><td>소프트웨어통합시험절차서</td></tr><tr><td>40002400STR</td><td>소프트웨어통합시험결과서</td></tr><tr><td>40002400SPS</td><td>소프트웨어 산출물명세서</td></tr><tr><td rowspan="5">나</td><td>40002410SRS</td><td>소프트웨어 요구사항명세서</td></tr><tr><td>40002410SDD</td><td>소프트웨어 설계기술서</td></tr><tr><td>40002410STD</td><td>소프트웨어통합시험절차서</td></tr><tr><td>40002410STR</td><td>소프트웨어통합시험결과서</td></tr><tr><td>40002410SPS</td><td>소프트웨어 산출물명세서</td></tr></table>		CSCI	기술자료 번호	설명	가	40002400SRS	소프트웨어 요구사항명세서	40002400SDD	소프트웨어 설계기술서	40002400STD	소프트웨어통합시험절차서	40002400STR	소프트웨어통합시험결과서	40002400SPS	소프트웨어 산출물명세서	나	40002410SRS	소프트웨어 요구사항명세서	40002410SDD	소프트웨어 설계기술서	40002410STD	소프트웨어통합시험절차서	40002410STR	소프트웨어통합시험결과서	40002410SPS	소프트웨어 산출물명세서
	CSCI		기술자료 번호	설명																							
가	40002400SRS	소프트웨어 요구사항명세서																									
	40002400SDD	소프트웨어 설계기술서																									
	40002400STD	소프트웨어통합시험절차서																									
	40002400STR	소프트웨어통합시험결과서																									
	40002400SPS	소프트웨어 산출물명세서																									
나	40002410SRS	소프트웨어 요구사항명세서																									
	40002410SDD	소프트웨어 설계기술서																									
	40002410STD	소프트웨어통합시험절차서																									
	40002410STR	소프트웨어통합시험결과서																									
	40002410SPS	소프트웨어 산출물명세서																									
국방규격 제출자료	▶ 국방규격에 포함된 자료가 모두 제출되었는가?																										
	▶ 점검결과: (예시) ○ 제출자료(CD)에 포함된 자료 - 국방규격서 - 도면 - 품질보증요구서(QAR) - SCS - CSCI별 소프트웨어 기술문서/원시파일/실행파일 - 자료목록																										

국방규격
제출자료
파일명

▶ 제출자료의 전산파일명이 지침에 따라 작성되었는가?

▶ 점검결과:

(예시)

○ 전산파일 목록

CSCI	구분	내용	전산파일명
가	기술 문서	40002400SR5	SW40002400SR5R1.hwp
		40002400SDD	SW40002400SDDR1.hwp
		40002400STD	SW40002400STDR1.hwp
		40002400STR	SW40002400STRR1.hwp
		40002400SPS	SW40002400SPSR1.hwp
	소스 코드	소스코드 및 실행파일	40002400SWPR1.zip
나	기술 문서	40002410SR5	SW40002410SR5R1.hwp
		40002410SDD	SW40002410SDDR1.hwp
		40002410STD	SW40002410STDR1.hwp
		40002410STR	SW40002410STRR1.hwp
		40002410SPS	SW40002410SPSR1.hwp
	소스 코드	소스코드 및 실행파일	40002410SWPR1.zip

국방규격서
QAR
연계

▶ 국방규격서 및 품질보증요구서(QAR)에 소프트웨어
기술자료 관련사항이 기록되어 있는가?

▶ 점검결과:

(예시)

○ 국방규격서 2항 적용문서에 소프트웨어 기술자료 연계

○ QAR 2항 적용문서에 소프트웨어 기술자료 연계

③ SW 기술문서 점검표

점검대상	(예시) 무기통제장치 CSCI	
점검항목	점검내용	확인
서식	▶ 각 SW 기술문서의 서식을 준수하였는가? - 표지(명칭, 작성기관명) - 내지(작성자, 검토자, 승인자)	
	▶ 점검결과: (예시) ○ 표지 확인 - 명칭 확인 - 작성기관명 확인 ○ 내지 확인 - 작성자, 검토자, 승인자 확인	
버전	▶ 각 문서의 버전 및 이력을 확인하였는가?	
	▶ 점검결과: (예시) ○ 버전 확인 - 버전 1.0 ○ 이력 확인 - 이력 확인	
추적성	▶ 기술문서간 추적성이 유지되었는가?	
	▶ 점검결과: (예시) ○ 각 문서간 식별자 추적성 참조 (별첨)	

④ SW 산출물 점검표

점검대상	(예시) 가 CSCI					
점검항목	점검내용	확인				
디바이스 목록	▶ SPS에 디바이스 목록이 작성되었는가? ☞ 참고 각 디바이스별로 식별자, 디바이스명, 용도, 도면번호, 부품번호가 포함되어야 함					
	▶ 점검결과: (예시) ○ 디바이스 1종 식별 (SPS 표9 참조)					
파일 목록	▶ 디바이스별 파일 목록(실행파일/원시파일)이 작성되어 있고, 목록 내 각 파일에 대해 속성 정보가 기록되어 있는가?					
	▶ 점검결과: (예시) ○ 디바이스별 파일 목록 <table border="1"> <thead> <tr> <th>디바이스</th><th>실행파일</th><th>원시파일</th></tr> </thead> <tbody> <tr> <td>FDEV-001</td><td>4</td><td>290</td></tr> </tbody> </table>		디바이스	실행파일	원시파일	FDEV-001
디바이스	실행파일	원시파일				
FDEV-001	4	290				
체크섬	▶ 각 파일에 대한 체크섬 생성시 사용된 유틸리티 명칭이 기록되어 있는가?					
	▶ 점검결과: (예시) ○ 체크섬 유틸리티 : ChkSumGenerator 버전 2.1 (ChkSumUtil 사)					
생성/탑재 절차	▶ 각 디바이스별 실행파일을 생성 및 탑재하기 위한 절차가 기록되어 있는가? 이 때, 문서에 기술되어 있는 파일 명이 파일목록에 모두 포함되어 있는가? ☞ 참고 실행파일 생성 및 탑재절차를 FIG 등 별도의 기술문서로 작성하였을 경우, 규격화 대상에 포함하여야 함					
	▶ 점검결과: (예시) ○ 디바이스별 생성 및 탑재 절차 확인 결과 <table border="1"> <thead> <tr> <th>디바이스</th><th>생성절차</th><th>탑재절차</th></tr> </thead> <tbody> <tr> <td>FDEV-001</td><td>정상</td><td>정상</td></tr> </tbody> </table>		디바이스	생성절차	탑재절차	FDEV-001
디바이스	생성절차	탑재절차				
FDEV-001	정상	정상				
분류체계	▶ 분류체계 목록이 포함되어 있는가?					
	▶ 점검결과: (예시) ○ 분류체계 목록 포함 (SPS 부록 참조)					

점검대상	(예시) 나 CSCI								
점검항목	점검내용	확인							
디바이스 목록	▶ SPS에 디바이스 목록이 작성되었는가? ☞ 참고 각 디바이스별로 식별자, 디바이스명, 용도, 도면번호, 부품번호가 포함되어야 함								
	▶ 점검결과: (예시) ○ 디바이스 2종 식별 (SPS 표9 참조)								
파일 목록	▶ 디바이스별 파일 목록(실행파일/원시파일)이 작성되어 있고, 목록 내 각 파일에 대해 속성 정보가 기록되어 있는가?								
	▶ 점검결과: (예시) ○ 디바이스별 파일 목록 <table border="1"> <thead> <tr> <th>디바이스</th><th>실행파일</th><th>원시파일</th></tr> </thead> <tbody> <tr> <td>FDEV-001</td><td>2</td><td>145</td></tr> <tr> <td>FDEV-002</td><td>1</td><td>92</td></tr> </tbody> </table>		디바이스	실행파일	원시파일	FDEV-001	2	145	FDEV-002
디바이스	실행파일	원시파일							
FDEV-001	2	145							
FDEV-002	1	92							
체크섬	▶ 각 파일에 대한 체크섬 생성시 사용된 유틸리티 명칭이 기록되어 있는가?								
	▶ 점검결과: (예시) ○ 체크섬 유틸리티 : ChkSumGenerator 버전 2.1 (ChkSumUtl 사)								
생성/탑재 절차	▶ 각 디바이스별 실행파일을 생성 및 탑재하기 위한 절차가 기록되어 있는가? 이 때, 문서에 기술되어 있는 파일명이 파일목록에 모두 포함되어 있는가? ☞ 참고 실행파일 생성 및 탑재절차를 FIG 등 별도의 기술문서로 작성하였을 경우, 규격화 대상에 포함하여야 함								
	▶ 점검결과: (예시) ○ 디바이스별 생성 및 탑재 절차 확인 결과 <table border="1"> <thead> <tr> <th>디바이스</th><th>생성절차</th><th>탑재절차</th></tr> </thead> <tbody> <tr> <td>FDEV-001</td><td>정상</td><td>정상</td></tr> <tr> <td>FDEV-002</td><td>정상</td><td>정상</td></tr> </tbody> </table>		디바이스	생성절차	탑재절차	FDEV-001	정상	정상	FDEV-002
디바이스	생성절차	탑재절차							
FDEV-001	정상	정상							
FDEV-002	정상	정상							
분류체계	▶ 분류체계 목록이 포함되어 있는가?								
	▶ 점검결과: (예시) ○ 분류체계 목록 포함 (SPS 부록 참조)								

⑤ 도면 연계사항 점검표

점검대상	(예시) 가 CSCI																				
점검항목	점검내용		확인																		
디바이스 연계	▶ SPS의 디바이스 목록에 있는 각 디바이스와 연계된 도면이 존재하는가? 또한, 해당 도면 내의 부품번호가 일치하는가?																				
	▶ 점검결과: (예시) ○ 디바이스별 파일 목록																				
	<table> <tr> <td>디바이스</td> <td>도면번호</td> <td>부품번호</td> </tr> <tr> <td>FDEV-001</td> <td>40002400</td> <td>CHIP-001</td> </tr> </table>	디바이스		도면번호	부품번호	FDEV-001	40002400	CHIP-001													
디바이스	도면번호	부품번호																			
FDEV-001	40002400	CHIP-001																			
디바이스 부품 연계	▶ SPS의 디바이스 목록에 있는 각 디바이스에 탑재되는 실행파일 목록과 해당 도면 주기란의 소프트웨어 부품번호를 비교하였을 때, 수량과 번호가 일치하는가?																				
	▶ 점검결과: (예시) ○ 디바이스별 실행파일 및 소프트웨어 부품 비교 결과																				
	<table> <tr> <td>품번</td> <td>탑재된 소프트웨어</td> <td>부품번호</td> <td>관련문서</td> </tr> <tr> <td>1</td> <td>GaRun.exe</td> <td>40002400E1</td> <td>40002400SPS</td> </tr> <tr> <td>1</td> <td>Garun.lib</td> <td>40002400E2</td> <td>40002400SPS</td> </tr> <tr> <td>1</td> <td>Garun.dat</td> <td>40002400E3</td> <td>40002400SPS</td> </tr> <tr> <td>1</td> <td>Garun.cfg</td> <td>40002400E4</td> <td>40002400SPS</td> </tr> </table>	품번		탑재된 소프트웨어	부품번호	관련문서	1	GaRun.exe	40002400E1	40002400SPS	1	Garun.lib	40002400E2	40002400SPS	1	Garun.dat	40002400E3	40002400SPS	1	Garun.cfg	40002400E4
품번	탑재된 소프트웨어	부품번호	관련문서																		
1	GaRun.exe	40002400E1	40002400SPS																		
1	Garun.lib	40002400E2	40002400SPS																		
1	Garun.dat	40002400E3	40002400SPS																		
1	Garun.cfg	40002400E4	40002400SPS																		

점검대상	(예시) 나 CSCI																
점검항목	점검내용		확인														
디바이스 연계	▶ SPS의 디바이스 목록에 있는 각 디바이스와 연계된 도면이 존재하는가? 또한, 해당 도면 내의 부품번호가 일치하는가?																
	▶ 점검결과: (예시) ○ 디바이스별 파일 목록																
	<table><tr><td>디바이스</td><td>도면번호</td><td>부품번호</td></tr><tr><td>FDEV-001</td><td>40002410</td><td>CHIP-003</td></tr><tr><td>FDEV-002</td><td>40002410</td><td>CHIP-005</td></tr></table>			디바이스	도면번호	부품번호	FDEV-001	40002410	CHIP-003	FDEV-002	40002410	CHIP-005					
디바이스	도면번호	부품번호															
FDEV-001	40002410	CHIP-003															
FDEV-002	40002410	CHIP-005															
디바이스 부품 연계	▶ SPS의 디바이스 목록에 있는 각 디바이스에 탑재되는 실행파일 목록과 해당 도면 주기란의 소프트웨어 부품번호를 비교하였을 때, 수량과 번호가 일치하는가?																
	▶ 점검결과: (예시) ○ 디바이스별 실행파일 및 소프트웨어 부품 비교 결과																
	<table><tr><td>품번</td><td>탑재된 소프트웨어</td><td>부품번호</td><td>관련문서</td></tr><tr><td>3</td><td>NaRun.exe</td><td>40002410E1</td><td>40002410SPS</td></tr><tr><td>3</td><td>Narun.bmp</td><td>40002410E2</td><td>40002410SPS</td></tr><tr><td>5</td><td>Narun.cfg</td><td>40002410E3</td><td>40002410SPS</td></tr></table>			품번	탑재된 소프트웨어	부품번호	관련문서	3	NaRun.exe	40002410E1	40002410SPS	3	Narun.bmp	40002410E2	40002410SPS	5	Narun.cfg
품번	탑재된 소프트웨어	부품번호	관련문서														
3	NaRun.exe	40002410E1	40002410SPS														
3	Narun.bmp	40002410E2	40002410SPS														
5	Narun.cfg	40002410E3	40002410SPS														

⑥ 소프트웨어 목록 점검표

점검항목	점검내용	확인													
CSCI별 실행파일 목록	▶ SCS의 실행파일 목록의 각 속성정보가 정확한가?														
	▶ 점검결과: (예시) ○ 각 CSCI별 실행파일 요약 <table border="1"> <tr> <th>CSCI</th> <th>총 실행 파일 수</th> <th>연구개발</th> <th>국내구매</th> <th>해외구매</th> </tr> <tr> <td>가</td> <td>4</td> <td>4</td> <td>1</td> <td>0</td> </tr> <tr> <td>나</td> <td>3</td> <td>3</td> <td>0</td> <td>0</td> </tr> </table>		CSCI	총 실행 파일 수	연구개발	국내구매	해외구매	가	4	4	1	0	나	3	3
CSCI	총 실행 파일 수	연구개발	국내구매	해외구매											
가	4	4	1	0											
나	3	3	0	0											
구매 소프트웨어 식별	▶ 구매 소프트웨어가 존재할 경우, 각 구매 소프트웨어 대한 식별정보가 모두 기술되어 있는가?														
	▶ 점검결과: (예시) ○ 구매 소프트웨어 총 9종 식별정보 기술 (SCS 표8 참조)														

무기체계 소프트웨어 코딩규칙

(Software Coding Rule Guide, SCR-G)

제 1 장 개 요

1. 목 적

이 규칙은 방위력 개선사업으로 개발되는 소프트웨어의 품질 및 신뢰성 향상을 위해 프로그램 코딩 시 준수해야 할 규칙을 정의한 문서이다.(약어명칭 : DAPA SCR-G)

2. 적용 범위

- 1) 적용언어 : C, C++, C#, Java
- 2) 적용대상 : 방위력 개선사업으로 개발되는 소프트웨어

3. 관련 표준

[코딩 규칙]

- 1) MISRA-C : 2012
- 2) MISRA-C++ : 2008
- 3) JSF++(Joint Strike Fighter Air Vehicle C++)
- 4) Code conventions for the Java Programming Language(Oracle)
- 5) C# Coding conventions 또는 .NET Framework Design Guideline(Microsoft)

[산업 표준]

- 1) IEC 61508
- 2) ISO 26262
- 3) DO-178B/C
- 4) IEC 62279
- 5) IEC 60880
- 6) EN 50128

4. 코딩규칙 적용 기준

체계의 특성에 가장 적합한 위 ‘3항’의 관련 표준 준수를 원칙으로 한다. 단, 관련 표준 적용시 조정이 필요한 항목은 사유를 제시하고 사업관리회의를 통해 확정한다. (부품국산화 등 표준 적용이 제한되는 사업의 경우는 무기체계 소프트웨어 코딩규칙 제3장에 정의된 프로그램 작성 규칙을 따를 수 있다.)

5. 역할 및 책임

가. 방위사업청(방위사업정책국)

- 1) 코딩 규칙 제정 및 개선

나. 사업관리본부(사업관리부서)

- 1) 코딩규칙 적용 계획 확정
- 2) 코딩규칙 준수 여부 확인 및 관리

다. 연구개발주관기관

- 1) 코딩규칙 적용 계획 수립
- 2) 코딩규칙을 준수하여 프로그램 작성
- 3) 코딩규칙 위반사항 점검 및 조치

라. 기술지원기관

- 1) 코딩 규칙 개정 소요 제기
- 2) 코딩규칙 관련 표준 적용계획의 적절성 검토
- 3) 코딩규칙 준수 여부 확인 및 기술지원

6. 용어 정의

가. 함수 : 작업을 수행하도록 작성된 독립적이고 규격화된 프로그램

나. 예약어 : 의미를 정해 놓고 사용되는 단어로 사용자가 임의로 정의 할 수 없는 단어

다. 매크로 : 함수대신 사용하는 것으로 대체로 짧은 문장에 주로 사용하고 복잡한 문장을 단순화하기 위한 용도로 사용

라. 배열 : 같은 형태의 데이터가 연속으로 나열된 구조

마. 주석 : 프로그램에 영향을 주지 않으며 해당부분의 내용을 설명, 해설, 참조하는 등의 정보

바. Argument : 인자, 인수, 함수 호출시 매개변수에 넣어주는 변수 또는 값

사. Assignment : 대입되거나 할당 되는 연산자

아. Linkage : Translation Unit 내외부 참조 가능 여부

자. Parameter : 매개변수, 함수 선언 시 각각의 변수

차. Scope : 선언된 식별자가 사용될 수 있는 영역

카. Storage Duration

1) Automatic Storage Duration : Block Scope이며, 'extern'으로 선언되지 않은 변수

2) Static Storage Duration : External 또는 Internal Linkage를 갖거나, 'static'으로 선언된 변수

타. Side Effect : 코드의 수정 또는 동작이 실행에 영향을 미치는 현상

파. Sequence Point : 각각의 C Statement가 종료되는 지점

제2장 주석 및 명명 규칙

1. 주석 규칙 (Comment Rule)

- 가. C 프로그램 경우 “/* ~ */”, C++ 프로그램 경우 “/* ~ */” 또는 “// ~”를 사용한다. (단, C 프로그램의 경우 컴파일러가 “// ~”를 지원할 경우에 사용할 수 있다.)
 - 나. 주석 내에는 작성자, 날짜, 수정이력, 기능설명, 설계문서와 연계되는 정보 등의 필수 항목은 반드시 포함한다.
 - 다. 주석 내에는 Nested Comment를 삽입해서는 안된다.
 - 라. 기타 주석 작성은 개발자 특성에 맞는 규칙을 사용하되 조직차원에서의 통일된 지침을 수립하여 적용한다.
 - 마. 소스코드 내 해당내용을 자세하고 정확하게 설명할 수 있는 주석을 작성하고, 작성된 주석은 파일별 소스코드의 30%이상으로 한다.
- ※ 계산방법 : 주석문의 수 / 실행문의 수

2. 명명 규칙 (Naming Rule)

가. 파일명

- 1) 함수/클래스일 경우, 파일명을 일치시키고 복수일 경우 파일명은 기능을 잘 설명해 줄 수 있는 명칭으로 한다.
- 2) 파일의 첫글자는 대문자 또는 소문자로 통일하여 표기한다.
- 3) 파일을 잘 설명할 수 있는 이름으로 파일명을 작성한다.

나. 함수/클래스/메소드명

- 1) 기능을 잘 표현할 수 있는 명칭을 사용한다.
- 2) 함수 생성 시 “동사 + (명사)”순으로 한다.
- 3) 클래스 생성 시 “ClassName” 형태로 조합하되 첫 단어의 글자는 대문자로 작성한다.
- 4) 클래스, 함수, 메소드명을 작성할 경우, 실제 기능을 유추할 수 있는 단어를 조합하여 작성한다.

다. 변수명

- 1) Scope, Linkage를 구분할 수 있어야 하며, 각각 Internal, Extern을 구분하여 작성한다.

예) “Pointer 변수는 p, Global 변수는 g, Static 변수는 s,
Count 변수는 n” 으로 표기

- 2) 변수명의 첫 글자는 반드시 문자를 사용하고 31자를 넘기지 않는다.
- 3) 변수명은 헝가리언, 카멜, 파스칼 표기법 또는 조직 내부 작성지침에 따라 사용한다.

라. 매크로명

- 1) 매크로명은 전체를 대문자 또는 “_”를 사용한다.
- 2) 파일, 매크로명은 표준에 있는 예약어를 사용하지 않는다.

예) _변수명, scanf, printf 등

제3장 프로그램 작성 규칙

1. 적용 규칙(66개)

구 분		주 요 내 용	개수
C, C++ 공통적 용 (46개)	스타일	한 라인씩 구성, goto 문, 다중 return 문 자제 등	12
	초기화	변수, 문자열 배열, 포인터 초기화 등	3
	식별자	전역변수 중복 식별자 자제 등	4
	조건식	float 자료형 비교, 조건문 결과 항상 True, False 등	5
	변 환	함수 변환, 데이터 형변환, 상수형 자료 변환 등	8
	포 인 터/ 배열	Null 포인터 참조, 배열 인덱스 초과값 사용 등	5
	연산자	0 나누기 연산, side effect, 음수 shift 자제 등	9
C 전용 적용		scanf, malloc, 함수 파라미터 등	5
C++ 전용 적용		throw, virtual function, exception 처리, 생성자, 소멸자 등	15

가. 공통 적용 규칙 (C, C++)

1) 스타일 규칙

가) 소스코드 내 해당내용을 자세하고 정확하게 설명할 수 있도록 “제2장 주석 및 명명규칙”에 따라 주석을 작성하고, 작성 된 주석은 파일별 소스코드의 30%이상으로 한다.

나) Switch 구문에서 첫번째 Label 전에 코드 구문이 존재하면 안된다.

틀린 예	<pre> void foo(int m) { switch (m) { n = n + 1; /* 위반 */ case 1: --n; break; default: ++n; break; } } </pre>
바른 예	<pre> void foo(int m) { switch (m) { case 1: --n; break; default: ++n; break; } } </pre>
설 명	switch 구문은 Label로 바로 이동하여 수행됨. n=n+1은 수행되지 않는 소스코드임.

다) 함수/변수의 선언 시 type을 명시해야 한다.

틀린 예	<pre> extern i; /* 위반 */ extern foo(void); /* 위반 */ </pre>
바른 예	<pre> extern int i; extern int foo(void); </pre>
설 명	void 리턴 형이 아닌 함수의 return 문에 값을 명시적으로 표현하지 않으면 묵시적으로 스택의 임의의 값이 리턴 될 수 있음.

라) 의미 없는 구문은 사용하지 말아야 한다.(side effect)

틀린 예	<pre>void foo(int x) { int n = 0; n==x; /* 위반 */ }</pre>
바른 예	<pre>void foo(int x) { int n = 0; n=x; }</pre>
설 명	n==x를 통하여 발생된 boolean 값을 사용하지 않음. 불필요한 소스코드임(n==x는 프로그램의 실행에 영향을 주지 않음)

마) 함수 선언시 사용된 리턴형과 함수 내에서 사용되는 리턴형은 일치해야 한다.

틀린 예	<pre>int f1(int value) { return; /* 위반 */ } void f2(int value) { return value; /* 위반 */ } int f1(int value) { } /* 위반 */</pre>
바른 예	<pre>int f2(int value) { return value; }</pre>
설 명	예상에서 벗어나는 값이 리턴 될 수 있음.

바) 외부 함수 사용시 이를 명시하고 사용해야 한다.

틀린 예	<pre>int foo(void) { int r; r = func(0); /* 위반 */ return r; }</pre>
바른 예	<pre>extern int func(int value); int foo(void) { int r; r = func(0); return r; }</pre>
설 명	링커는 object로 되어있는 파일로부터, 함수의 식별자와 리턴 값의 형만 검사하여 파라미터가 올바르게 사용되었는지 확인할 수 없음.

사) 전처리에 사용되는 매크로는 사용 전에 반드시 정의되어야 한다.

틀린 예	<pre>#define M1 5 #if(M2==5) /* 위반 */ int y = 0; #endif</pre>
바른 예	<pre>#ifndef M3 #define M3 0 #endif #if (M3 == 5) int z = 0; #endif</pre>
설 명	정상적인 전처리를 위하여 #if문에서 사용되는 식별자는 반드시 사용 전에 선언되어 있어야함. 정의없이 macro를 사용하면 전처리기는 때로 아무런 warnig을 주지 않고 value를 zero로 가정. #ifdef, #ifndef, defined()를 사용하면 macro의 존재여부를 확인하게 됨 M1은 정의가 되어있어서 문제없으나, M2는 정의 없이 전처리 비교문에 사용하였기 때문에 문제가 되며, 바른예인 M3는 #ifndef 구문이 확인하는 구문이기 때문에 문제 발생하지 않음.

아) goto 문을 사용하지 말아야 한다.

틀린 예	<pre> short test_1404(void) { if (abc > 10) { goto mylabel; /* 위반 */ } abc++; mylabel: ... } </pre>
바른 예	<pre> short test_1404(void) { while (abc ≤ 10) { abc++; } ... } </pre>
설 명	goto 문을 사용은 프로그램의 흐름을 복잡하게 하여 가독성 및 문제가 될 수 있는 코드를 작성할 수 있음.

자) 하나의 함수는 하나의 Exit Point를 가져야 한다.

틀린 예	<pre> int S16test_1407(void) { if (s16a > 0) { return 0; } else { return 1; } } /* 위반 */ </pre>
바른 예	<pre> int S16test_1407(void) { int temp; if (s16a > 0) { temp = 0; } else { temp = 1; } return temp; } </pre>
설 명	하나의 함수에 여러 개의 Return 문을 사용할 경우 가독성 및 문제 발생시 수행 경로에 대하여 판단하기 힘들.

차) switch ~ case 문은 default 문이 포함되어야 한다.

틀린 예	<pre> switch (s16a) { case 0: x = 2; break; case 1: x = 1; break; } /* 위반 */ </pre>
바른 예	<pre> switch (s16a) { case 0: x = 2; break; case 1: x = 1; break; default: x = 0; break; } </pre>
설 명	코드의 가독성을 높이고 예외 case에 대한 방어를 위해 switch 문에는 항상 default문을 사용함.

카) 한 줄에 하나의 명령문을 사용한다.

틀린 예	<pre> r=p; p++; /* 위반 */ </pre>
바른 예	<pre> r=p; p++; </pre>
설 명	코드의 가독성을 높이고 실수를 방지하기 위함.

타) if - else if 문은 else 문도 반드시 포함 시킨다.

틀린 예	<pre> if (s16a == 2) { r = 3; } else if (s16a==3) /* 위반 */ { r = 2; } </pre>
바른 예	<pre> if (s16a == 2) { r = 3; } else if (s16a==3) { r = 2; } else { r = 1; } </pre>
설 명	코드의 가독성을 높이고 예외상황에 대한 방어를 위해 if-else if 문을 사용 시 반드시 else 문을 사용함.

2) 초기화 규칙

가) char 배열의 초기화시 char 배열의 크기는 초기화 문자열의 크기와 Null을 포함하는 크기로 선언되어야 한다.

틀린 예	<pre> char h1[4] = "Hello"; /* 위반 */ char h2[5] = "Hello"; /* 위반 */ </pre>
바른 예	<pre> char h1[6] = "Hello"; </pre>
설 명	hello는 5개의 문자열이기 때문에 최소 크기가 6 byte 이상의 배열을 사용해야 Null이 보장됨

나) 변수는 사용전 반드시 초기화되어야 한다.

틀린 예	<pre>void foo(void) { int i; int j; j = i+2; /* 위반 */ }</pre>
바른 예	<pre>void foo(void) { int i; int j; i = 0; j = i+2; }</pre>
설 명	지역변수 i의 값을 초기화하지 않고 사용함.

다) 값이 초기화되지 않은 변수의 포인터를 함수의 Read-only(const)로 사용해서는 안된다.

틀린 예	<pre>extern int foo(const int *p); void foo1(void) { int y; int r; r = foo(&y); /* 위반 */ }</pre>
바른 예	<pre>extern int foo(const int *p); void foo1(void) { int y; int r; y = 100; r = foo(&y); }</pre>
설 명	y의 값에 대한 초기화 없이 함수의 Read-only로 사용하였음. 그 값을 참조하여 사용하는 의도가 명확함에도 불구하고 값을 설정하지 않음.

3) 식별자 규칙

가) external과 internal linkage의 특성을 동시에 가질 수 없다.

틀린 예	<pre>extern int foo(int value); extern int number; /* ... */ static int number; /* 위반 */ static int foo(int value); /* 위반 */</pre>
바른 예	<pre>extern int foo(int value); extern int number; /* ... */ extern int number; extern int foo(int value);</pre>
설 명	두 가지 storage class를 모두 갖고 있을 경우 어떻게 동작 할지 표준에 정의되어 있지 않음.

나) external linkage scope에서 선언된 함수나 Object의 이름은 유일해야 한다.

틀린 예	<pre>// TU1.c int press; void foo(void) /* 위반 */ { } // TU2.c float press; /* 위반 */ void foo(int) /* 위반 */ { }</pre>
바른 예	<pre>// TU1.c int press; void foo(void) { } // TU2.c float local_press; void foo_tu2(int) { }</pre>
설 명	external linkage 특성을 갖는 식별자는 여러 파일에서 구현될 경우 링커가 에러를 발생시킴.

다) external linkage scope에서 정의된 함수나 Object의 데이터형은 선언 시 정의와 동일해야 한다.

틀린 예	<pre>// TU1.c extern int i; /* 위반 */ extern void foo(int); /* 위반 */ // TU2.c float i = 0.0; int foo(int){ }</pre>
바른 예	<pre>// TU1.c extern float i; extern int foo(int); // TU2.c float i = 0.0; int foo(int){ }</pre>
설 명	i 가 float로 실제 정의되어 있는데 extern int형으로 선언하여 사용하고 있음. foo함수는 int형을 return하도록 구현되어있는데 extern void형으로 선언하여 사용하고 있음.

라) 외부 scope에서 정의된 변수를 내부 scope에서 재정의 해서는 안된다.

틀린 예	<pre>void func(int a) { int variable = 6; if (a > 0) { int variable; /* 위반 */ variable = 10; } func2(variable); /* variable은 10이 아니라 6 */ }</pre>
바른 예	<pre>int variable; void func (void) { int local_variable; local_variable = 10; }</pre>
설 명	바깥 scope의 변수 사용 의도와 내부 scope의 변수 사용 의도가 혼용되어 가독성이 떨어짐. 특히 내부 scope에서 변경한 값이 외부 스코프에서 그대로 있을 것이라고 착각하게 되어 논리적 오류가 발생할 여지가 많아짐.

4) 조건식 규칙

가) float 자료형에서 동등성 비교연산을 수행하지 말아야 한다.

틀린 예	<pre> int foo(float a, float b) { int r = 0; if (a == b) /* 위반 */ { r = 1; } return r; } </pre>
바른 예	<pre> int foo(float a, float b) { int r = 0; if (a < b) { r = 1; } return r; } </pre>
설 명	부동소수점 특성 상 예상한 비교 결과가 나오지 않을 수 있음. 게다가 실행해보기 전 까지는 행위를 예측할 수 없고, 컴파일러마다 다를 수도 있기 때문에 수식의 비교가 참을 기대했을 때 종종 참으로 판단되지 않음.

나) 조건문의 결과가 항상 True 혹은 False라면 이는 조건문으로 작성해서는 안된다.

틀린 예	<pre> void foo(unsigned char x, int y) { if (y < 20) { if (y < 30) /* 위반 - True */ { y = 30; } if (y > 30) /* 위반 - False */ { y = 30; } } if (x < 0) /* 위반 - False */ { x = 0; } } </pre>
바른 예	<pre> void foo(unsigned char x, int y) { if (y < 30) { if (y < 20) { y = 30; } } } </pre>
설 명	y의 값이 20 보다 작을 경우 항상 30보다 작기 때문에 비교구문이 항상 True임. y의 값이 20보다 작으면서 30보다 큰 값을 가질 수 없기 때문에 항상 False 임.

다) switch의 case의 조건을 만족할 수 없는 Label을 사용하지 않는다.

틀린 예	<pre> extern int x; void foo(<i>unsigned int</i> n) { switch (n) { case -1: /* 위반 */ ++x; break; default: --x; break; } } </pre>
바른 예	<pre> extern int x; void foo(unsigned int n) { switch (n) { case 1: ++x; break; default: --x; break; } } </pre>
설 명	n은 항상 unsigned 값이며 case는 음수임. 실행될 수 없는 case로 Dead Code 발생 가능함.

라) Switch 구문에서 Expression을 논리적 연산으로 사용하지 말아야 한다.

틀린 예	<pre> extern int x; void foo(int option, int pick) { switch (option == pick) /* 위반 */ { case 0: ++x; break; case 1: x = 0; break; } } </pre>
바른 예	<pre> extern int x; void foo(int option, int pick) { if (option == pick) { ++x; } else { x = 0; } } </pre>
설 명	논리적 연산은 if 문 사용을 권장하고 논리적 연산을 switch 구문에 사용 시 문제가 될 수 있음.

마) 수행되지 않는 소스코드를 작성하지 말아야 한다.

틀린 예	<pre> void foo(int n) { switch (n) { case 1: --n; break; n = n + 1; /* 위반 */ default: ++n; break; } } </pre>
바른 예	<pre> void foo(int n) { switch (n) { case 1: --n; break; default: ++n; break; } } </pre>
설 명	예제에서는 break 구문 후에 n=n+1 코드를 삽입하여 절대 수행될 수 없는 코드를 작성 하였음.

5) 변환 규칙

가) 선언된 데이터 형으로 표현할 수 있는 숫자의 영역을 초과하는 값을 할당하지 말아야 한다.

틀린 예	<pre>void foo(short n) { short x; x = 10000 * 4; /* 위반 */ }</pre>
바른 예	<pre>void foo(long n) { long x; x = 10000 * 4; }</pre>
설 명	short형의 값은 최대 양수 32767까지 표현할 수 있지만 예제에서는 40000 값을 할당해서 Overflow가 발생함.

나) 가변인수를 받는 함수의 Conversion 지시자와 Argument의 type은 동일해야 한다.

틀린 예	<pre>#include <stdio.h> extern void foo(int n, unsigned char *s) { printf("The total is %s", n); /* 위반 */ }</pre>
바른 예	<pre>#include <stdio.h> extern void foo(int n, unsigned char *s) { printf("The total is %d", n); }</pre>
설 명	int형 변수 n을 %s로 출력했을 경우 어떠한 값으로 표현될지 표준에 정의되어 있지 않음.

다) 가변인수를 받는 함수의 Conversion 지시자와 Argument의 개수는 동일해야 한다.

틀린 예	<pre>#include <stdio.h> extern void foo(int test) { printf("The score for test %d is %d", test); /* 위반 */ }</pre>
바른 예	<pre>#include <stdio.h> extern void foo(int test) { printf("The score for test is %d", test); }</pre>
설 명	Argument의 개수가 틀릴 경우 동작여부가 보장 안됨.

라) Object 저장 값을 표현할 수 없는 데이터로의 형 변환을 하지 말아야 한다.

틀린 예	<pre>void foo(signed int sia) { signed char sca; if (sia > 127) { sca = sia; /* 위반 */ } }</pre>
바른 예	<pre>void foo(unsigned int sia) { unsigned char sca; if (sia < 255) { sca = (unsigned char)sia; } }</pre>
설 명	sia의 값이 -128~127의 범위를 벗어나는 integer 값이 할당될 경우 데이터 값이 변형되어 유효하지 않은 값이 될 수 있음.

마) 음수 값을 unsigned type으로 변환해서는 안된다.

틀린 예	<pre> void foo(int sia) { unsigned char ucx; if (sia < 0) { ucx = sia; /* 위반 */ } } </pre>
바른 예	<pre> void foo(signed char sia) { signed short ucx; if (sia < 0) { ucx = (signed short)sia; } } </pre>
설 명	변수 sia에 저장된 음수값을 unsigned로 변환할 경우 문제가 될 수 있음.

바) Character 문자열과 Wide character 문자열을 혼용하지 말아야 한다.

틀린 예	<pre>#include <stddef.h> #define HN "Hello" #define WN " World.\n" #define HW L"Hello" #define WW L" World.\n" char * t2 = HN WW; /* 위반 */</pre>
바른 예	<pre>#include <stddef.h> #define HN "Hello" #define WN " World.\n" #define HW L"Hello" #define WW L" World.\n" char * t1 = HN WN; wchar_t * t3 = HW WW;</pre>
설 명	Character 문자열 HN과 Wide Character 문자열 WW를 한번에 혼용하여 사용하였음. char 자료형 포인터에 할당되어 어떻게 동작할지 예측할 수 없음.

사) 포인터 Cast의 결과로 인한 이전 포인터의 CONST 특성의 상실을 유의해야 한다.

틀린 예	<pre>void foo(const int *pci) { int *pi; pi = (int *)pci; /* 위반 */ }</pre>
바른 예	<pre>void foo(const int *pci) { int *pi; *pi = *pci; }</pre>
설 명	const의 특성을 casting을 함으로 특성을 상실함, 함수 선언부에서 변환되지 않는 Read-Only로 판단하여 오류를 일으킬 수 있음. (값을 변환할 수 없는 변수에 변환 시도)

아) 포인터 Cast의 결과로 인한 이전 포인터의 volatile 특성 상실을 유의해야 한다.

틀린 예	<pre>void foo(volatile int *pvi) { int *pi; pi = (int *)pvi; /* 위반 */ }</pre>
바른 예	<pre>void foo(volatile int *pvi) { int *pi; *pi = *pvi; }</pre>
설 명	volatile의 특성을 casting에 의하여 상실할 수 있음

6) 포인터/배열 규칙

가) 포인터는 참조전에 NULL 여부를 반드시 확인하여야 한다.

틀린 예	<pre>#define NULL (void *)0 void foo(int *p) { int r; r = *p; /* 위반 */ }</pre>
바른 예	<pre>#define NULL (void *)0 void foo(int *p) { int r; if (p != NULL) { r = *p; } }</pre>
설 명	p의 주소가 NULL임에도 불구하고 참조함. Null pointer를 참조할 경우 System Crash 발생함.

나) 지역변수의 주소값을 더 넓은 scope를 가진 변수에 할당하지 말아야 한다.

틀린 예	<pre>extern int *pi; void foo(void) { int a; pi = &a; /* 위반 */ }</pre>
바른 예	<pre>void foo(void) { int a; int *pi; pi = &a; }</pre>
설 명	지역변수 a의 주소값을 외부에 넓은 scope를 가진 변수에 할당했음. 지역변수는 함수의 종료와 함께 유효하지 못한 값이 되기 때문에 보호받지 못하는 데이터 영역임.

다) 지역변수의 주소값을 함수의 리턴값으로 사용하지 말아야 한다.

틀린 예	<pre>int *func (void) { int a = 0; return &a; /* 위반 */ }</pre>
바른 예	<pre>int value; int *func (void) { return &value; }</pre>
설 명	지역변수는 함수의 종료 시 보호받지 못하는 데이터 영역으로, 그 주소값을 사용했을 경우 문제가 발생할수 있음.

라) 선언된 배열의 크기를 초과하는 인덱스 값을 사용하지 말아야 한다.

틀린 예	<pre>extern int a[10]; extern int foo(void) { int r; r = a[10]; /* 위반 */ a[10] = 0; /* 위반 */ *(a+10) = 0; /* 위반 */ return r; }</pre>
바른 예	<pre>extern int a[10]; extern int foo(void) { int r; r = a[9]; a[9] = 0; *(a+9) = 0; return r; }</pre>
설 명	배열 a의 index는 0~9까지 사용할 수 있음. 예제는 10을 사용함으로 메모리 침범이 발생됨.(Buffer Overflow)

마) Null Pointer를 산술연산 하지 않는다.

틀린 예	<pre>#define NULL (void *)0 void foo(int *q) { if (q == NULL) { ++q; /* 위반 */ } }</pre>
바른 예	<pre>#define NULL (void *)0 void foo(int *q) { if (q == NULL) { } else { ++q; } return; }</pre>
설 명	컴파일러나 시스템에 따라서 Null Pointer의 내부 처리 방식이 상이할 수 있어 이식성에 문제가 발생할 수 있음.

7) 연산자 규칙

가) 하나의 Sequence Point 내에서 하나의 Object Value를 두 번 이상 변경하지 않아야 한다.

틀린 예	<pre>extern void foo(void){ int x = 7; x = x++ * x++; /* 위반 */ }</pre>
바른 예	<pre>extern void foo(void) { int x = 7; int y; y = x * x; y++; y++; }</pre>
설 명	개발환경에 따라 값이 일정하지 않음. 항상 의도한 값이 예측되도록 코드를 작성해야 함. 예) Visual studio는 51, CodeWarrior HC12는 56 값 나옴.

나) 나누는 값이 변수일 경우 0인지를 반드시 확인하여야 한다.

틀린 예	<pre>void foo(int x, int n){ int r; r = x / n; /* 위반 */ }</pre>
바른 예	<pre>void foo(int x, int n) { int r; if(n != 0) { r = x / n; } }</pre>
설 명	0 으로 나누기 연산을 했을 경우 System Crash 발생함.

다) 하나의 Sequence Point내에서 Object의 값을 변경하고 Access 하지 않아야 한다.

틀린 예	<pre>extern void foo(int index) { int r; r = (2 * index) + index++; /* 위반 */ }</pre>
바른 예	<pre>extern void foo(int index) { int r; r = (3 * index); index++; }</pre>
설 명	ISO:1990 표준의 어떠한 우선순위에 의한 계산이라도 결과는 항상 같아야 함. index의 값이 어떠한 우선순위로 변경 될지, 되지 않을지 예상이 되지 않음.

라) 음수 값 또는 데이터 사이즈를 초과하는 값을 사용하여 Shift operator를 하지 않는다.

틀린 예	<pre>extern unsigned int uia; extern void foo(void) { uia = uia << -2; /* 위반 */ uia = uia << 40; /* 위반 */ }</pre>
바른 예	<pre>extern unsigned int uia; extern void foo(void) { uia = uia >> 2; }</pre>
설 명	shift value값을 음수로 사용했을 경우 가독성이 떨어져 오해 및 실수의 가능성이 높아짐.

마) Underlying type이 Unsigned인 데이터에 '-'를 사용하여 결과를 assign하지 말아야 한다.

틀린 예	<pre>void foo(unsigned char i) { long j; j = -i; /* 위반 */ }</pre>
바른 예	<pre>void foo(signed char i) { long j; j = i; }</pre>
설 명	unsigned 타입의 데이터형에 '-' 연산자를 사용하는 경우 의도하지 않은 결과를 발생시킬 수 있음.

바) 'sizeof'의 인자로 side effect를 포함하는 식을 사용하면 안된다.

틀린 예	<pre>long foo(int x) { long y; unsigned long un; y = x; un = sizeof(++y); /* 위반 */ return y; }</pre>
바른 예	<pre>long foo(int x) { long y; unsigned long un; y = x; y = y + 1; un = sizeof(y); return y; }</pre>
설 명	개발자는 sizeof의 인자인 ++y가 수행될 것이라고 기대하지만, 일부 컴파일러는 sizeof를 통하여 데이터 연산이 가능하나, 표준에 의해 정확하게 정의 되어 있지 않기 때문에 일부 컴파일러에서는 변환되지 않을 수 있음.

사) Boolean 표현 값에는 논리 연산자(&&, ||, !)만을 사용하여야 한다.

틀린 예	<pre> int foo(int a, int b, int c, int d) { int x; x = ~((a > b) && (b > 10)); /* 위반 */ x = (a > b) + (c > d); /* 위반 */ x = (a + b) > (c > d); /* 위반 */ return x; } </pre>
바른 예	<pre> int foo(int a, int b, int c, int d) { int x; if ((a > b) && (b > 10)) { ... } return x; } </pre>
설 명	일반적으로 True는 1로 판단하지만 ISO-C90 표준에 의하여 명확하게 정의 되어 있지 않으므로 이식성을 위하여 논리 연산자와 산술 연산자를 분리하여 사용해야 함.

아) condition에 assignment operator를 사용하지 말아야 한다.

틀린 예	<pre>if(var2 = var1) { /* 위반 */ }</pre>
바른 예	<pre>var2 = var1; if(var2 != 0) { } if((var2 == var1) != 0) { } if (var2 == var1) { }</pre>
설 명	조건문에서 사용된 경우 비교 연산자 "=="과 혼동하여 사용되었을 우려가 높음. Expression의 가장 상위가 assignment이지만 확인함.

자) Signed Value에서 Bitwise연산자(<<, ~, |, ^ 등)로 인한 Negative Value를 유의해야 한다.

틀린 예	<pre>void foo(long sl) { long slr; slr = sl << 4; /* 위반 */ }</pre>
바른 예	<pre>void foo(unsigned int sl) { unsigned int slr; slr = sl << 4; }</pre>
설 명	Signed값을 Shift연산은 했을 경우 sign bit값의 변환에 유의해야함. 양수가 음수로 변환될 수 있음.

나. C 전용 규칙

1) Scanf의 Argument 에는 Object Value가 저장된 주소값이 입력되어야 한다.

틀린 예	<pre>#include <stdio.h> extern void foo(void) { int n; scanf("%d", n); /* 위반 */ }</pre>
바른 예	<pre>#include <stdio.h> extern void foo(void) { int n; scanf("%d", &n); }</pre>
설 명	scanf에 사용되는 식별자는 pointer type로 사용되어야함.

2) #include 구문에서 표준에 맞지 않는 Character set을 사용하지 않아야 한다.

틀린 예	<pre>#include "taemin's.h" /* 위반 */ #include "donghyun.h/*temporary*/ /* 위반 */ #include "Jady\$.h" /* 위반 */ #include "June's-header.h" /* 위반 */</pre>
바른 예	<pre>#include "kim_tae_min.h"</pre>
설 명	일부 컴파일러에서 호환되지 않을 수 있음.

3) malloc에서 사용하는 인자값은 assign될 변수 type의 정수배이어야 한다.

틀린 예	<pre> void foo(void) { long * ptd; ptd = (long *) malloc(3); /* 위반 */ ptd = (long *) malloc(11); /* 위반 */ ptd = (long *) malloc(13); /* 위반 */ } </pre>
바른 예	<pre> void foo(void) { long * ptd; ptd = (long *) malloc(12); } </pre>
설 명	Multiple size가 아닐 경우 할당되지 않은 메모리를 침범하여 사용할 확률이 높아짐.

4) 함수의 Argument type과 개수는 함수의 Prototype, 선언, 정의가 모두 같아야 한다.

틀린 예	<pre> extern int foo (unsigned char a); void test(float a, unsigned char b) { int r; r = foo(a); /* 위반 */ r = foo(b, a); /* 위반 */ } </pre>
바른 예	<pre> extern int foo (unsigned char a); void test(unsigned char a) { int r; r = foo(a); } </pre>
설 명	함수의 Argument에 대하여 다르게 입력할 시 데이터 값이 변환되어 의도하지 않은 동작을 할 위험이 있음. (함수의 Argument의 타입은 반드시 표준에 정의된 Compatible한 타입으로 사용해야 함.)

- 5) 구조체/배열의 초기화 시 default 초기화 값(0)을 제외하고, 구조에 맞게 ‘{ }’를 사용하여 선언된 Size에 맞게 초기화 해야 한다.

틀린 예	<pre> struct s { int i; int j; } ; const struct s s1 = { 8}; /* 위반 */ int arr[10] = { 10}; /* 위반 */ int arr1[3][2] = { 1, 2, 3, 4, 5, 6}; /* 위반 */ </pre>
바른 예	<pre> struct s { int i; int j; } ; const struct s s0 = { 0}; const struct s s1 = { 8, 0}; const struct s s2 = { 0, 0}; int arr[10] = { 0}; int arr1[3][2] = { { 1, 2}, { 3, 4}, { 5, 6} }; </pre>
설 명	어떠한 값이 초기화되는지 오해를 일으킬 수 있음. 가독성을 위해 default 초기값(0) 이외의 초기값을 입력할 때는 반드시 전체 멤버 변수에 대하여 초기화를 해야 함

다. C++ 전용 규칙

- 1) 동적할당된 변수를 release 할 때에는 변수의 type과 일치하게 release 하여야 한다.

틀린 예	<pre> void bar(int num) { int* p1 = new int[num]; int* p2 = new int; delete p1; // 위반 delete[] p2; // 위반 } </pre>
바른 예	<pre> void bar(int num) { int* p1 = new int[num]; int* p2 = new int; delete[] p1; delete p2; } </pre>
설 명	동적할당을 할 때 배열형식으로 할당하고, 해제할 때 단일 형식으로 해제해야 함. 배열의 크기가 클수록 다량의 메모리 누수가 발생함.

2) 지역 변수의 주소 값을 처리하는 handle을 return하지 말아야 한다.

틀린 예	<pre> class A{ public: void f(); }; A& foo() { A a; return a; // 위반 } void bar(){ foo().f(); } </pre>
바른 예	<pre> class A{ public: void f(); }; A foo() { A a; return a; } void bar(){ foo().f(); } </pre>
설 명	return 문은 localdata의 값을 리턴하지만, Method가 &를 이용하도록 되어있어서 주소만 리턴하게 됨.

3) 함수 parameter의 주소 값을 처리하는 handle을 return하지 말아야 한다.

틀린 예	<pre> class A { public: void f(); }; A& foo(A a) { return a; // 위반 } void bar() { A a; foo(a).f(); } </pre>
바른 예	<pre> class A { public: void f(); }; A foo(A a) { return a; } void bar() { A a; foo(a).f(); } </pre>
설 명	return 문은 foo함수의 parameter 값을 리턴하지만, Method가 &를 사용하도록 되어 있어서 주소만 리턴 함.

4) 소멸자내에서 처리할 수 없는 예외 상황을 발생시키지 말아야 한다.

틀린 예	<pre> void funcThrowsEverything(); class C { public: ~C(); private: bool b; }; C::~~C() { funcThrowsEverything(); if (b) { throw int(); } } // 위반 C gc; </pre>
바른 예	<pre> void funcThrowsEverything(); class C { public: ~C(); private: bool b; }; C::~~C() { try{ funcThrowsEverything(); if (b) { throw int(); } } catch(...){ } } C gc; </pre>
설 명	소멸자에서 예외가 발생하면 C++에서는 프로그램을 강제 종료시킴. 처리되지 않은 예외가 있는 경우 해당클래스는 종료되지 않음.

5) 사용되지 않는 예외 처리 문을 작성하지 말아야 한다.

틀린 예	<pre> classE{ }; void boo() throw(int, E); void foo() throw() { try { boo(); }catch (...){ }catch (E& e) { // 위반 } } </pre>
바른 예	<pre> classE{ }; void boo() throw(int, E); void foo() throw() { try { boo(); }catch (E& e) { }catch (...){ } } </pre>
설 명	첫 번째 catch 문에서 두 번째 catch 문의 예외 처리가 되나, 두 번째 catch 문의 예외처리는 전혀 발생하지 않음.

- 6) exception specification에 기술되지 않은 모든 throw에 대하여 예외처리를 해야만 한다.

틀린 예	<pre>void foo(int a) throw(int){ if (a > 0) { throw int(); } else { throw float(); // 위반 } }</pre>
바른 예	<pre>void foo(int a) throw (int) { try { throw float(); } catch (...) { } }</pre>
설 명	throw float();에서 발생하는 예외를 처리하는 코드가 존재하지않음. 처리할수 없는 예외는 unexpected()에 의해서 처리됨. 만일 사용자 개체(Implementation defined object)의 경우 unexpected()에서도 처리되지 않으므로 terminate()를 호출하여 프로그램 강제 종료가 발생함.

- 7) main 함수에서 처리되지 않는 throw를 작성하지 말아야 한다.

틀린 예	<pre>int main(){ try { throw int(); } catch (int) { } throw int(); // 위반 }</pre>
바른 예	<pre>int main(){ try { throw int(); } catch (int) { } }</pre>
설 명	main 함수에 처리되지 않는 예외가 존재함. 시스템에서 terminate()를 호출하고 프로그램을 강제종료 시킴.

8) release 된 메모리 영역을 사용하지 말아야 한다.

틀린 예	<pre> void bar(int); void foo() { int * i = new int; * i = 1; bar(* i); delete i; bar(* i); // 위반 } </pre>
바른 예	<pre> void bar(int); void foo() { int * i = new int; * i = 1; bar(* i); delete i; } </pre>
설 명	bar 함수가 참조하는 변수 i는 이미 delete됨. 잘못된 메모리 영역 참조로 시스템 crash 또는 오동작 가능함.

9) copy operator를 통해서, 복사되지 않는 멤버 변수가 존재하지 말아야 한다.

틀린 예	<pre> class C { public: ~C(); C(const C& rhs); C& operator=(const C& rhs) { //위반 : a는 복사가 되지만, b는 복사가 되지 않음 a = rhs.a; return *this; } private: int a; int b; }; </pre>
바른 예	<pre> class C { public: ~C(); C(const C& rhs); C& operator=(const C& rhs) { a = rhs.a; b = rhs.b; return *this; } private: int a; int b; }; </pre>
설 명	Copy operator가 존재하는 클래스에 적용되는 코딩 규칙임. 모든 멤버가 복사되는지 검사. 위의 클래스에서 복사를 하면 독립된 클래스가 멤버 b의 경우 다른값을 저장하고 있음.

10) C++에서 동적 메모리 할당은 new 연산자를 이용하여야 한다.

틀린 예	<pre>#include<cstdlib> class C { }; void foo(){ C* p=static_cast<C*>(std::malloc(sizeof(C))); // 위반 delete p; }</pre>
바른 예	<pre>#include<cstdlib> class C { }; void foo(){ C* p = new C; delete p; }</pre>
설 명	C style의 메모리 할당은 생성자를 호출하지 않음. delete 수행과정에서 잘못된 heap으로 인하여 프로그램 강제종료 현상 발생 가능함.

11) 순수 가상함수는 반드시 0으로 초기화 되어야 한다.

틀린 예	<pre> namespace { class A { <i>virtual void f () = 1;</i> // 위반 }; } </pre>
바른 예	<pre> namespace { class A { <i>virtual void f () = 0;</i> }; } </pre>
설 명	순수 가상함수를 0으로 초기화하지 않을 경우 가상함수가 쓰레기 값을 가지고 있을 수 있음.

12) 순수함수는 반드시 가상함수로 선언되어야 한다.

틀린 예	<pre> class Abstract { public: void foo() = 0; // 위반 }; class Derived : public Abstract { public: void foo() {} }; </pre>
바른 예	<pre> class Abstract { public: virtual void bar() = 0; }; class Derived : public Abstract { public: virtual void bar() {} }; </pre>
설 명	순수 함수가 정의된 클래스를 상속받은 서브 클래스에서 순수 함수를 재정의하여 사용할 경우 함수 매핑시 오류가 발생할 수 있음.

- 13) virtual base 클래스의 포인터는 derived 클래스의 포인터로 cast 할 때에는 dynamic_cast만 사용해야 한다.

틀린 예	<pre> class Base { public: virtual ~Base (); virtual class Derived * getAsDerived () = 0; }; class Derived : virtual public Base { public: virtual class Derived * getAsDerived () { return this; } }; void foo (Base * bp, Derived * dp) { dp = static_cast <Derived *> (bp); // 위반 dp = (Derived *) (bp); // 위반 dp = reinterpret_cast <Derived *> (bp); // 위반 } </pre>
바른 예	<pre> class Base { public: virtual ~Base (); virtual class Derived * getAsDerived () = 0; }; class Derived : virtual public Base { public: virtual class Derived * getAsDerived () { return this; } }; void foo (Base * bp, Derived * dp) { dp = dynamic_cast <Derived *> (bp); dp = bp->getAsDerived (); } </pre>
설 명	base 클래스의 클래스 포인터를 안전하지 않은 casting 방법으로 상속받은 클래스 포인터의 형변환은 안전성을 보장할 수 없음. dynamic_cast를 사용해야 함.

- 14) 생성자/소멸자 내에서 가상함수는 식별자 없이 호출하면 안된다.(The virtual function is called in constructor/destructor without qualifier.)

틀린 예	<pre> class B { public: virtual ~B()=0; virtual void func(); protected: B(); B(const B&); B& operator=(const B&); }; class D : public B { public: D () : B() {} virtual ~D(); virtual void func(); private: D(const D&); D& operator=(const D&); }; B::B() { func(); // 위반 } </pre>
바른 예	<pre> class B { public: virtual ~B()=0; virtual void func(); protected: B(); B(const B&); B& operator=(const B&); }; class D : public B { public: D () : B() {} virtual ~D(); </pre>

	<pre> virtual void func(); private: D(const D&); D& operator=(const D&); }; B::B() { B::func(); } </pre>
설 명	Base클래스의 생성자에서 파생 클래스의 객체는 instance가 유효한 상태가 아니므로 Base클래스의 생성자에서 호출되면 안됨.

15) 생성자/소멸자에 `dynamic_cast`를 사용해서는 안된다.

틀린 예	<pre> class B2 { public: virtual ~B2 (); virtual void foo (); B2 () { typeid (B2); // 위반 foo (); // 위반 dynamic_cast< B2* > (this); // 위반 } }; </pre>
바른 예	<pre> class B1 { public: B1 () { typeid (B1); } }; class B2 { public: virtual ~B2 (); virtual void foo (); B2 () { B2::foo (); } }; </pre>
설 명	생성과 소멸중의 타입은 완전히 생성된 객체의 타입과 다를 수 있기 때문에, 생성자와 소멸자에서의 객체의 다이내믹 타입의 사용은 개발자의 의도와 다를 수 있음.

소프트웨어 신뢰성/보안성 시험 절차

제 1 장 개 요

1. 목 적

본 절차서는 무기체계 연구개발간 소프트웨어 신뢰성 및 보안성 시험을 정량적이고 객관적인 기준에 따라 체계적으로 실시하기 위하여 시험 절차, 기준 등 세부사항을 제공함에 그 목적이 있다.

2. 관련 근거

- 가. 「방위사업관리규정」
- 나. 「무기체계 소프트웨어 개발 지원에 관한 규정」
- 다. 「무기체계 소프트웨어 개발 및 관리 매뉴얼」

3. 적용 범위

가. 적용 사업

- 1) 무기체계 연구개발사업
- 2) 핵심기술(시험개발), 핵심SW(시험개발)
- 3) 신개념기술시범사업(ACTD)
- 4) 부품국산화(핵심부품)

※ 기초연구, 응용연구, 구매사업, 부품국산화(일반부품), 탐색개발 등의 사업은 사업특성을 고려하여 사업관리회의를 통해 적용 여부 및 범위 조정 가능

나. 적용 대상 언어 : C, C++, Java, C#을 원칙으로 하되, 자동화 도구가 지원하지 않는 경우에 대해서는 제외

다. 적용 대상 소프트웨어

구 분	시험 대상여부
개발 소프트웨어	○
기개발 소프트웨어	×
상용(COTS) 소프트웨어 (HW와 함께 제공되는 SW 포함)	×
공개(Open) 소프트웨어	○
자동 생성 Code	○

- 1) 기개발, 상용 소프트웨어는 소프트웨어 상세설계 단계에서 수정없이 재사용이 확정된 경우 시험 대상에서 제외 할 수 있으나 수정을 한 경우는 해당 함수에 대하여 신뢰성시험을 수행하여야 한다. 단, 기개발 소프트웨어는 무기체계 소프트웨어 신뢰성시험을 실시한 소프트웨어를 말한다.
- 2) 공개 소프트웨어는 부록14. 공개SW 무기체계 적용 가이드라인에 따라 소프트웨어 신뢰성 시험을 수행한다.
- 3) 자동 생성 Code는 타당한 사유를 제시할 경우 신뢰성 시험 대상에서 제외할 수 있다.

4. 방침

- 가. 소프트웨어 신뢰성시험은 정적시험(코딩규칙 검증, 취약점 점검, 소스코드 메트릭 점검)과 동적시험(코드실행률 점검)으로 구분하여 수행한다.
- 나. 소프트웨어 보안성시험은 전장관리정보체계에 대해 행자부의 『소프트웨어 개발보안 가이드』 준수여부를 시험한다.
- 다. 기술지원기관은 소프트웨어 동적시험에 대하여 제3자 검증이나 연구개발 주관기관 자체 실시 결과를 검토한다.
- 라. 소프트웨어 신뢰성 및 보안성 시험은 소프트웨어 형상항목(CSCI) 단위로 실시하는 것을 원칙으로 한다. (신뢰성 시험 중 동적시험은 소프트웨어 구성품(CSC), 단위소프트웨어(CSU) 단위로 적용 가능)
- 마. 소프트웨어 신뢰성 및 보안성 시험에 대한 기본계획(적용대상 CSCI, 평가항목, 사용도구 및 기타 사업별 필요 사항)은 체계개발실행계획서(또는 연구개발계획서)에 반영하고, 상세설계단계에서 소프트웨어 검토회의 등을 통해 확정한다.
- 바. [소프트웨어통합시험](#)항목에 대한 세부계획, 절차 및 결과는 소프트웨어 산출물(STP, STD, STR)에 포함하여 추적성을 유지하여야 하며, 규격화 단계에서 최신화 하여야 한다.
- 사. 시험도구는 본 매뉴얼 [부록7의 소프트웨어 신뢰성/보안성 시험 기준](#)을 지원하는 도구인지 확인하되, 확인 제한 시에는 소프트웨어 검토회의(소프트웨어 신뢰성시험계획)를 통해 결정한다.
- 아. 시험도구가 지원되지 않는 개발언어로 구현된 소프트웨어는 개발도구 등이 지원하는 기능을 이용하여 시험 할 수 있으며 소프트웨어 검토회의(소프트웨어 신뢰성시험계획) 또는 사업관리회의에서 시험방안을 결정하여 적용한다.
- 자. 하드웨어의 제한사항으로 인하여 실제 하드웨어(Target)에서 동적시험이 불가능한 경우에는 개발주관기관은 객관적으로 타당한 사유를 제시하고 사업관리회의에서 동적시험방안을 결정하여 적용한다.

5. 소프트웨어 신뢰성/보안성 시험 시기

가. 소프트웨어 구현, 소프트웨어 통합 및 시험 단계

- 1) 소프트웨어(통합)시험계획에 소프트웨어 신뢰성 및 보안성 시험 항목을 반영한다.
- 2) 연구개발 주관기관은 소프트웨어 신뢰성 및 보안성 시험을 실시하고, 시험 결과를 소프트웨어통합시험결과서(STR)에 반영한다.
 - 가) 정적 시험은 소프트웨어 개발 초기부터 수행하고 통합시험단계에서 종합적으로 실시한다.
 - 나) 동적 시험은 연구개발주관기관 자체 또는 제3자 검증을 통해 수행한다.
 - 다) 동적 시험은 소프트웨어를 실제 하드웨어(Target)에 탑재한 상태에서 소프트웨어통합시험절차서에 기술된 시험절차를 이용하여 요구사항기반으로 소프트웨어 코드 실행물을 점검한다.
 - ※ 하드웨어가 가용하지 않을 경우에는 체계 통합시험단계에서 수행
 - 라) 보안성시험은 전장관리정보체계에 대해 행자부의 『소프트웨어 개발보안 가이드』에 따른 보안성시험계획 준수여부를 시험한다.
- 3) 기술지원기관은 소프트웨어 신뢰성 및 보안성 시험을 지원 및 점검하고 시험결과를 확인한다.

나. 시험평가 단계

- 1) 기술지원기관은 개발시험평가계획에 반영된 소프트웨어 신뢰성 및 보안성 시험 항목에 대한 내용을 확인한다.
- 2) 연구개발 주관기관은 신뢰성시험 이후 시험평가 수행 중에 결함 수정 등으로 인해 소프트웨어를 변경한 경우 해당 소프트웨어에 대해 신뢰성 및 보안성 시험을 재실시하고 변경된 기능 내역, 해당 소프트웨어 목록(파일명, 생성일자, 체크섬, 파일크기 등 포함) 및 신뢰성 및 보안성 시험결과(소프트웨어통합시험결과서(STR) 수정본 포함)를 사업관리 부서를 경유하여 기술지원기관에 제출한다.
- 3) 기술지원기관은 연구개발주관기관에서 제출한 시험평가 수행과정에서 변경된 기능에 대한 신뢰성 및 보안성 시험결과를 검토한다.

다. 규격화 단계

- 1) 연구개발 주관기관은 시험평가간 수정사항이 반영되어 신뢰성 및 보안성 시험을 완료한 최종 소스코드(파일명, 생성일자, 체크섬, 파일크기 등 포함)와 보안성 및 신뢰성 시험결과(STR 수정본 포함)를 규격화 자료로 제출한다.
- 2) 기술지원기관은 연구개발주관기관에서 제출한 소프트웨어의 신뢰성 및 보안성 시험결과와 소스코드의 최신성을 검토하여 그 결과를 규격화 자료 검토결과와 함께 사업관리부서에 제출한다.

라. 양산 단계

- 1) 형상통제 제안기관(업체 포함)은 수정되는 소프트웨어에 대해 무기체계 개발단계의 소프트웨어 신뢰성 및 보안성 확보 활동 기준에 따라 신뢰성 및 보안성 시험을 실시하고, 그 결과(기술지원기관의 검토 필요)를 형상통제 제안자료(형상통제제안서, 제안기관 자체검토서, 수정 후 국방규격 기술자료, 제안내용을 입증할 수 있는 관련자료 등)에 반영하여 형상관리책임기관에 제출한다.

6. 용어 설명

소프트웨어 신뢰성 및 보안성 시험과 관련된 용어에 대한 설명은 다음과 같다.

가. “소프트웨어 신뢰성 시험”이란 소프트웨어가 동작할 수 있는 다양한 경우의 수를 확인함으로써 소프트웨어가 일으킬 수 있는 결함을 식별하는 시험을 말하며 정적시험 및 동적시험으로 구분한다.

나. “소프트웨어 신뢰성 시험 및 보안성 시험 확인”이란 소프트웨어의 정적, 동적 신뢰성 시험과 전장관리정보체계 소프트웨어의 보안 취약점 제거를 위한 보안성 시험이 기술적인 정확성과 적절성을 가지고 정해진 요구사항을 충족하는 지를 결정하기 위해 검증을 통해 수집된 자료와 사실에 대해 검토 및 확인하는 과정을 말한다.

다. “소프트웨어 정적(Static) 시험”이란 소프트웨어를 실행하지 않은 상태에서 잠재적인 결함을 검출하는 시험을 말하며, 코딩 규칙(Coding Rule) 검증 시험, 취약점 점검 시험 및 소스코드 메트릭(Code Metrics) 점검을 의미한다.

- 라. “소프트웨어 동적(Dynamic) 시험”이란 소프트웨어를 실제 하드웨어(Target)에 탑재한 상태에서 소프트웨어통합시험절차서에 기술된 시험 절차에 따라 요구사항기반으로 소프트웨어 코드 실행률(Coverage)을 점검하는 것을 말한다.
- 마. “소프트웨어 보안성 시험”이란 해킹 등 사이버공격의 원인인 보안약점을 개발단계에서 사전에 제거하기 위해 행자부 「소프트웨어 개발보안 가이드」를 적용하여 개발하였는지 확인하는 시험을 말하며 전장관리정보체계를 대상으로 수행한다.
- 바. “소프트웨어 신뢰성 및 보안성 시험도구(Testing Tool)”란 소프트웨어 신뢰성 및 보안성 시험을 수행하기 위해 사용되는 자동화 도구를 말하며, 시험도구는 관련표준에서 인증된 제품을 사용하여야 한다.
- 사. “코딩 규칙”이란 소프트웨어 구현에 적용하는 소스 코드 작성 규칙으로서 【부록 6】 “무기체계 소프트웨어 코딩 규칙”을 따른다.
- 아. “취약점 점검”이란 소프트웨어 소스 코드가 CWE(Common Weakness Enumeration) 목록에 정의된 취약점을 포함하고 있는지 점검하는 활동을 말한다.
- 자. “소스코드 메트릭”이란 소프트웨어의 복잡도 감소, 유지보수 용이성 증대 등 소프트웨어 품질향상을 위한 소스코드의 품질 측정지표를 말한다.
- 차. “거짓경보(False Alarm)”란 시험도구를 이용하여 결함 검출한 내용이 실제 오류가 아닌 결함을 오류로 검출하는 것을 말하며, 시험도구는 결함으로 탐지하였지만 기능 구현을 위해서 반드시 사용해야 하는 것은 거짓경보에 포함할 수 있다.
- 카. “코드 실행률(Code Coverage)”이란 동적 시험 수준을 결정하는 기준으로서 전체 소프트웨어 소스 코드에서 시험된 코드를 백분률(%)로 나타낸 것이며, 시험 수준에 따라 문장(Statement) 실행률, 분기(Branch) 실행률, MC/DC (Modified Condition/Decision)로 구분된다.
- 타. “문장(Statement) 실행률”이란 코드 실행률의 가장 기본적인 수준에 해당되는 것으로 시험대상 소프트웨어 소스 코드내의 문장 중 동적 시험간 적어도 한 번 이상 시험된 문장의 비율(%)을 의미한다.

파. “분기(Branch) 실행률”이란 시험대상 소프트웨어 소스 코드내의 분기문 중 동적 시험간 참(True), 거짓(False)이 적어도 한 번 이상 시험된 비율(%)을 의미한다.

하. “MC/DC(Modified Condition/Decision Coverage)”이란 가장 높은 수준의 코드 실행률로써 시험대상 소프트웨어 소스 코드내 분기문에 있는 모든 조건식 중 개별 조건식의 독립적인 변화가 분기문의 참, 거짓에 영향을 미치는 모든 조합에 대해 동적 시험간 적어도 한 번 이상 시험된 비율(%)을 의미한다.

제 2 장 소프트웨어 정적 시험 절차

1. 소프트웨어 정적 시험계획 수립 절차

가. 연구개발 주관기관은 소프트웨어개발계획서에 개략적인 소프트웨어 정적 시험계획(설계, 구현 및 시험 방안 등)을 포함하여 작성한다.

나. 연구개발주관기관은 소프트웨어요구사항명세서에 정의한 정적시험 요구 사항에 따라 소프트웨어통합시험계획서(STP) 및 소프트웨어통합시험절차서(STD)에 소프트웨어 정적시험 계획 및 절차를 작성한다. 소프트웨어 정적시험 계획은 다음 내용을 포함한다.

1) 시험대상 항목(CSCI, CSC, CSU 등이며 최소 시험단위는 파일임) 및 시험 기준 설정 방법

2) 정적 시험 시기 및 환경 구축 방안 (시험 도구 선정 등)

다. 사업관리부서는 관련 부서에서 검토한 결과를 종합하여 사업관리회의시 소프트웨어 정적 시험계획을 검토한다.

※ 상세한 소프트웨어 시험계획(정적 시험 포함)은 상세설계검토(CDR) 단계에서 결정

2. 소프트웨어 정적시험 절차

가. 연구개발 주관기관은 소프트웨어 통합시험단계에서 소프트웨어 정적시험을 자체적으로 수행하고 그 결과를 관련자료와 함께 사업관리부서로 제출한다.

※ 자체 시험결과 및 관련자료 제출자료 양식은 별지1 참조

나. 사업관리부서장은 연구개발주관기관이 제출한 자체 시험결과와 관련자료를 기술지원기관에게 통보하고 정적시험 확인을 의뢰한다.

다. 기술지원기관은 사업관리부서장으로부터 정적시험 확인 의뢰를 접수한 경우 연구개발주관기관과 협의하여 정적시험결과를 검증한다.

라. 연구개발 주관기관은 소프트웨어 정적 시험결과를 소프트웨어통합시험결과서 (STR)에 반영한다.

마. 소프트웨어통합시험단계에서 정적시험을 수행한 이후 소프트웨어 변경 사항이 없는 경우는 개발시험평가시 정적시험을 생략할 수 있다.

3. 시험 기준

가. 결함이 검출되지 않아야 하고 소스코드 메트릭 제한값을 준수해야 한다.

결함 검출시 소프트웨어 소스 코드 수정 후 재시험하여 결함 존재 여부를 확인한다. 단, 거짓정보 및 시험 대상이 아닌 것은 결함에 포함하지 않는다.

나. 시험 종류

1) 코딩 규칙 검증 : 본 매뉴얼 【부록 6】 무기체계 소프트웨어 코딩 규칙 준수 여부

2) 취약점 점검 : 소프트웨어 취약점 점검(CWE-658/659/660)

* 자동화 도구에서 지원하는 취약점 점검 항목은 모두 적용하고, CWE 목록 중 도구가 지원하지 않는 항목에 대해서는 대안과 사유를 제출하고 사업관리부서, 기술지원기관과 협의하여 결정

※ CWE(Common Weakness Enumeration)는 소프트웨어 보안 및 품질 강화를 위해 개발 시에 참고할 수 있도록 소프트웨어 취약점을 표준화한 목록으로서, 미국 비영리 공익 [기관인](#) MITRE가 제공

※ CWE-658/659/660은 C/C++/Java 언어로 소프트웨어를 개발할 때 소프트웨어 소스 코드에 발생 빈도가 높은 취약점 목록

3) 소스코드 메트릭(Code Metrics) 점검(함수 기준)

메트릭 종류	제한값	비 고
Cyclomatic Complexity	20이하	
Number of Call Levels	6이하	함수 내 조건문 등의 최대 중첩(Nesting)의 깊이
Number of Function Parameters	8이하	
Number of Calling Functions	8이하	이 함수가 몇 개의 다른 함수에 의해 호출되는가?
Number of Called Functions	10이하	이 함수가 몇 개의 다른 함수를 호출 하는가? * 같은 함수를 호출시는 1로 계산
Number of Executable Code Lines	200이하	

※ “메트릭 종류” 및 “제한값” 적용이 제한되는 경우에는 연구개발주관 기관에서 타당한 사유를 제시하고 사업관리회의를 통해 결정한다.

4. 시험확인서 : 별지2 참조

5. 시험확인서 제출시 포함사항(별도 첨부) 및 후속 조치 사항

가. 소프트웨어 신뢰성 시험도구의 한계로 인한 결함 중 거짓경보의 경우 해당 소스 코드를 확인하여 거짓경보 여부를 분석하고, 거짓경보에 대한 설명 자료를 추가한다.

나. 규격화시 소프트웨어통합시험결과서(STR)에 최신화 반영한다.

제 3 장 소프트웨어 동적 시험 절차

1. 소프트웨어 동적 시험계획 수립 절차

가. 연구개발 주관기관은 소프트웨어개발계획서에 개략적인 소프트웨어 동적 시험계획을 포함하여 작성한다. 소프트웨어 동적시험 계획은 다음 내용을 포함한다.

1) 시험 기준 설정 방법

결함의 발생빈도, 영향성 및 제어가능성을 평가한 후 수준별로 코드 실행률 시험종류를 설정한다.

※ 단, 체계/사업 특성에 따라 관련 국제 표준(MIL-STD-882E, DO-178, IEC 61508, ISO 26262 등)을 적용하여 개발할 경우에는 해당 표준을 따를 수 있으며, 주체계 운영과 직접적인 관련이 없는 장비(예: 지원장비, CBT(Computer based training) 등)는 동적 시험을 제외할 수 있다.

<세부 설정 기준>

결함 영향도 (Severity)	결함 발생빈도 (Exposure)	결함 제어가능성 (Controllability)		
		C1	C2	C3
S1	E1	S	S	S
	E2	S	S	S
	E3	S	S	S
	E4	S	S	S
S2	E1	S	S	S
	E2	S	S	S
	E3	S	S	S
	E4	S	S	B
S3	E1	S	B	B
	E2	B	B	B
	E3	B	B	B
	E4	B	B	M
S4	E1	B	B	M
	E2	B	M	M
	E3	B	M	M
	E4	M	M	M

* S : 문장(Statement), B : 분기(Branch), M : MC/DC(Modified Condition/Decision Coverage)

가) 결함 영향도

수준	내 용
S1	무시 가능(Negligible)
S2	사소함(Marginal)
S3	심각함(Critical)
S4	치명적(Catastrophic)

(한등급간 차이 10배 수준)

* 예 : S2는 S1보다 10배 심각

나) 결함 발생빈도

수준	내 용
E1	매우 낮은 확률 (발생가능성 거의없음)
E2	낮은 확률(수명주기당 수회)
E3	중간확률(1년에 수회)
E4	높은 확률(1달에 수회)

(한등급간 차이 10배 수준)

다) 결함 제어 가능성

수준	내 용
C1	간단히 제어가능
C2	보통의 경우 제어가능
C3	제어하기 어렵거나 불가능

(한등급간 차이 10배 수준)

2) 동적시험을 고려한 소프트웨어 설계, 구현 및 시험 방안

(복잡성, 시험용이성(Testability), 유지보수성 등)

3) 동적시험 시기 및 환경 구축방안 (시험 도구 선정 등)

나. 연구개발주관기관은 소프트웨어요구사항명세서에 정의한 동적시험 요구사항에 따라 소프트웨어통합시험계획서(STP) 및 소프트웨어통합시험절차서(STD)에 소프트웨어 동적 시험계획 및 절차를 작성한다.

다. 사업관리부서는 관련 부서에서 검토한 결과를 종합하여 사업관리회의시 소프트웨어 동적 시험계획을 검토한다.

※ 상세한 소프트웨어 시험계획(동적 시험 포함)은 상세설계검토(CDR) 단계에서 결정

2. 소프트웨어 동적시험 절차

가. 동적 시험은 연구개발주관기관 자체 또는 제3자 검증을 통해 수행한다.

나. 연구개발 주관기관은 소프트웨어 통합시험단계에서 소프트웨어를 실제 하드웨어 (Target)에 탑재한 상태에서 소프트웨어통합시험절차서에 기술된 시험절차를 이용하여 요구사항기반으로 소프트웨어 코드 실행물을 점검하여 목표 값을 달성해야 하며 그 결과를 관련자료와 함께 사업관리부서로 제출한다.

※ 하드웨어가 가용하지 않을 경우에는 체계 통합시험단계에서 수행. 자체 또는 3자검증 시험결과 및 관련자료 제출자료 양식은 별지1 참조

※ 자체 또는 3자검증 시험결과 및 관련자료 제출자료 양식은 별지1 참조

다. 사업관리부서장은 연구개발주관기관이 제출한 자체 시험결과와 관련자료를 기술지원기관에게 통보하고 동적시험 검토를 의뢰한다.

라. 기술지원기관은 사업관리부서장으로부터 동적시험 검토 의뢰를 접수한 경우 연구개발주관기관과 협의하여 동적시험결과를 검토한다.

마. 연구개발 주관기관은 소프트웨어 동적 시험결과를 소프트웨어통합시험결과서(STR)에 반영한다.

바. 소프트웨어통합시험 또는 체계통합시험 단계에서 동적시험을 수행한 이후 소프트웨어 변경사항이 없는 경우는 개발시험평가지 동적시험을 생략할 수 있다.

사. 목표값 미달시 시험대상 소프트웨어 소스 코드를 수정하거나 또는 필요한 조치를 취한 후 재시험하여 목표값 달성 여부를 확인한다.

3. 시험 기준

가. 설정된 시험종류(문장, 분기, MC/DC)에 대한 코드 실행률을 100% 달성하여야 한다. 단, 협의된 예외사항(방어코드 등)은 제외할 수 있다.

4. 시험확인서 : 별지2

5. 시험확인서 제출시 포함사항(별도 첨부) 및 후속 조치사항

가. 방어 코드(Defensive Code) 등 소프트웨어 신뢰성 시험도구의 한계로 인해 목표값을 달성 할 수 없는 경우에는 도구가 시험 할 수 없는 해당 소스 코드를 분석하고, 설명 자료를 추가한다.

나. 소스코드의 확장성을 고려하여 작성된 코드인 경우에는 해당 소스코드를 분석하고 관련 설명 자료를 추가한다.

제 4 장 소프트웨어 보안성 시험 절차

1. 소프트웨어 보안성 시험계획 수립 절차

가. 연구개발 주관기관은 전장관리정보체계 소프트웨어개발계획서에 개략적인 소프트웨어 보안성 시험계획을 포함하여 작성한다. 소프트웨어 보안성시험 계획은 다음 내용을 포함한다.

- 1) 시험대상 항목(CSCI, CSC, CSU 등) 및 시험기준 설정 방법
- 2) 보안성시험을 고려한 소프트웨어 설계, 구현 및 시험 방안
- 3) 보안성시험 시기 및 환경 구축방안 (시험 도구 선정 등)

나. 연구개발 주관기관은 소프트웨어요구사항명세서에 정의한 보안성시험 요구 사항에 따라 소프트웨어통합시험계획서(STP) 및 소프트웨어통합시험절차서(STD)에 소프트웨어 보안성 시험계획 및 절차를 작성한다.

다. 사업관리부서는 관련 부서에서 검토한 결과를 종합하여 사업관리회의시에 소프트웨어 보안성 시험계획을 검토한다.

※ 상세한 소프트웨어 보안성시험 계획은 상세설계검토(CDR) 단계에서 결정

2. 소프트웨어 보안성시험 절차

가. 연구개발 주관기관은 소프트웨어 통합시험단계에서 소프트웨어 보안성시험을 자체적으로 수행하고 그 결과를 관련자료와 함께 사업관리부서로 제출한다.

※ 자체 시험결과 및 관련자료 제출자료 양식은 별지1 참조

나. 사업관리부서장은 연구개발주관기관이 제출한 자체 시험결과와 관련자료를 기술지원기관에게 통보하고 보안성시험 확인을 의뢰한다.

다. 기술지원기관은 사업관리부서장으로 부터 보안성시험 확인 의뢰를 접수한 경우 연구개발주관기관과 협의하여 보안성시험결과를 검증한다.

라. 연구개발 주관기관은 소프트웨어 보안성 시험결과를 소프트웨어통합시험 결과서(STR)에 반영한다.

3. 시험 기준 : 소프트웨어 보안성 시험 충족여부(국정원 CC 인증된 도구를 사용하여 행자부 소프트웨어 개발보안 가이드 준수 여부, 전장관리정보체계에 한함).
4. 시험확인서 : 별지2
5. 시험확인서 제출시 포함사항(별도 첨부) 및 후속 조치 사항
 - 가. 소프트웨어 보안성 시험도구의 한계로 인한 결함 중 거짓경보의 경우 해당 소스 코드를 확인하여 거짓경보 여부를 분석하고, 거짓경보에 대한 설명자료를 추가한다.
 - 나. 규격화시 소프트웨어통합시험결과서(STR)에 최신화 반영한다.

(별지1)

신뢰성/보안성 시험 사전 수행결과 및 관련자료 제출양식

1. 표지

000체계 소프트웨어 신뢰성/보안성 시험 사전 수행결과 및 관련 자료 제출

정적시험 기간 : 00 ~ 00

동적시험 기간 : 00 ~ 00

보안성시험 기간 : 00 ~ 00

별첨1. HW/SW 형상목록

☐

별첨2. 신뢰성/보안성 시험대상 파일목록

☐

별첨3. 라이브러리/COTS/GOTS목록

☐

별첨4. 정적시험/보안성시험 자체수행결과 요약표

☐

별첨5. 동적시험 자체 또는 3자검증 수행결과 요약표

☐

별첨6. 자동화도구 시험결과 출력화면 캡취자료

☐

PM : 성명 (서명)

품질보증팀담당 : 성명 (서명)

(별지1-별첨1) HW/SW 형상 목록

AAA 체계 HW/SW 형상 목록															
번호	부체계 HW 형상	부체계 HW명	CSCI명	임무용도	분류	MCU / CPU	OS	개발언어	개발구분	출처	개발환경	정적 시험 대상	동적 시험 대상	보안성 시험대상	비고
1		레이더 신호처리	신호처리 #1 CSCI	레이더 신호처리 #1	Firmware	TMS32 0C6800	-	C	BSP	TI제공	CCS 5.0	X	X	X	
								C	BIOS	TI제공	CCS 5.0	X	X	X	미수정리포트
								C	BootLoa der	TI제공	CCS 5.0	X	X	X	미수정리포트
								C/C++	Applicat ion	자체개발	CCS 5.0	O	Δ	O	
2		안테나 신호수신	신호 수신 CSCI	신호 수신1	VHDL	Altera xx	-	VHDL				X	X	X	
				신호 수신2	Applicat ion			C				O	O	O	

...

체크리스트: 모든 HW 및 SW의 형상 목록을 누락 없이 제시하였는가?

예 ☐ 아니오 ☐

* SW가 없는 HW도 식별, 제시해야 함

(별지1-별첨2) 신뢰성/보안성 시험대상 파일 목록

[illegible]

* 재사용 : 기규격화된 소스코드 수정없이 사용

* 재활용 : 기규격화된 소스코드 일부 수정 사용

체크리스트 : ○ 모든 SW(Firmware, HDL(VHDL, verilog...), Labview 등 포함)를 누락 없이 예 ☐ 아니오 ☐

○ SW를 신규, 재사용, 재활용으로 구분하여 제시하였는가? 예 ☐ 아니오 ☐

*PLD(FPGA,CPLD등) 등에 구현된 SW도 제시 필요

(별지1-별첨3) 라이브러리/COTS/GOTS 목록

AAA체계 라이브러리/COTS/GOTS 목록								
시험 비대상 라이브러리 목록								
번호	라이브러리 파일명	라이브러리명	버전	구분	용도	사용CSCI	소스코드	예외사유
1	AVLib.lib	카메라 영상 캡춰 라이브러리	1.0	COTS	BB CSU에서 카메라 영상 획득용		없음	상용
2	BBB.c	oo 소프트웨어		GOTS	oo 기능 SW		BBB.c	관급
4	CCC.dll	영상 출력 라이브러리		GOTS			없음	관급

* BSP(Board Support Package) 등 HW와 함께 제공되는 SW 포함

시험 대상 라이브러리 목록								
번호	라이브러리 파일명	라이브러리명	버전	구분	용도	사용CSCI	소스코드	폴더 위치
1	Test.ocx	영상 처리 라 이브러리	0.1	자체개발	영상신호 처리		a.c, b.c, ...	-

체크리스트 : 모든 라이브러리/COTS/GOTS 목록을 제시하였는가?

예 아 니 오
☐ ☐ ☐ ☐

* 라이브러리가 없는 경우에도 "없음" 으로 표시

(별지1-별첨4) 정적시험/보안성시험 자체수행결과 요약표

AAA체계 정적시험/보안성시험 자체수행결과 요약표																				
표기어	부체계명	구성품명	CSC-명	CSC-명	CSU-명	시험단위	코딩규칙		취약점 점검		보안성시험		개발환경		정적시험도구		보안성 시험 도구	Line 수	재사용 / 재활용	비고
							결함	거짓 경보 (False Alarm)	결함	거짓 경보 (False Alarm)	결함	거짓 경보 (False Alarm)	언어	개발 도구	코딩 규칙	취약점 점검				
1	표적탐지부	기록장치	A	B	C	기록 SW	0	15	0	20	0	5	C	Visual studio x.x	도구명 및 버전	도구명 및 버전	도구명 및 버전			
		읽기장치	A	B	D															

체크리스트 : 결함 목록 및 False Alarm 목록/사유를 별도로 제출하였는가?

예
☐

아니오 ☐

(별지1-별첨5) 소스코드 메트릭 점검 결과

AAA 체계 소스코드 메트릭 점검 결과					
번호	항목	기준치	점검 결과	시험도구	비고
1				도구명 및 버전	
2					
3					
4					
5					
6					

체크리스트 : 예외 부분 및 사유를 첨부하였는가?

예 ☐ 아니오 ☐

(별지1-별첨6) 동적시험 자체 또는 제3자검증 수행결과 요약표

AAA 체계 동적시험 자체수행결과 요약표							
번호	파일명	기준치		측정치		시험도구	비고
1	ABC.cpp	Statement	100%	Statement	100%	도구명 및 버전	
2	DEF.cpp	Branch	100%	Branch	100%		
3	GHI.cpp	MC/DC	100%	MC/DC	95%		

체크리스트 : 예외 부분 및 사유를 첨부하였는가?

예 ☐ 아니오 ☐

체크리스트 : 소프트웨어통합시험결과서에 기술된 요구사항기반 시험 수행률을 첨부하였는가?

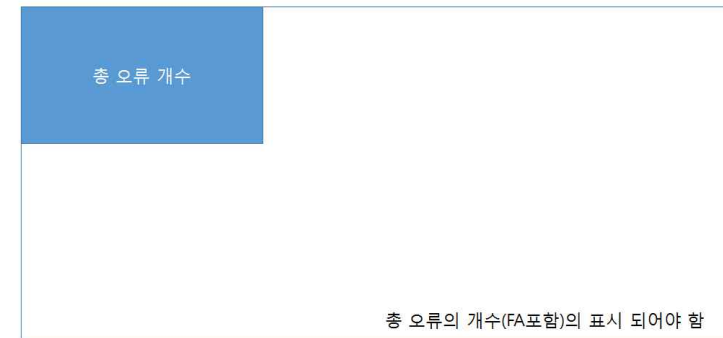
예 ☐ 아니오 ☐

(별지1-별첨7) 자동화도구 시험결과 출력화면 캡취자료

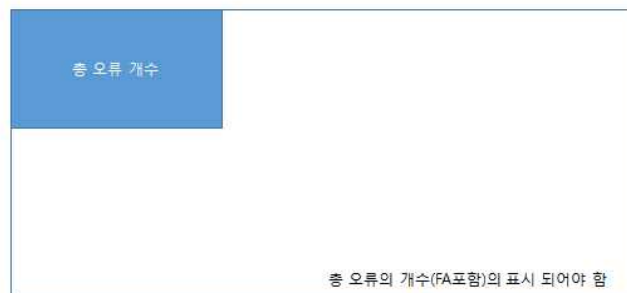
AAA체계 SW정적시험
자동화도구 화면 캡취

참고 : 시험 도구의 결과 화면을 캡취하며, FA등의 오류 개수등이 보여야 함
(SW 신뢰성/보안성 시험 단위별로 캡취 필요)

BBB CSU SW 코딩규칙 시험 결과



BBB CSU SW 취약점 점검 시험 결과



BBB CSU SW 보안성 시험 결과



(별지2)

SW 시험확인서(□신뢰성 □보안성)

사업명		OOO체계 연구개발 사업				
시험시제				시험일자	20yy. mm. dd	
시험장소				시험방법	자동화도구 이용 시험 및 분석	
시험자		소속	성명 (서명)			
관련근거		- OOO사업팀-xxx('yy.mm.dd) OOO체계 SW신뢰성 시험계획 통보 - 시험평가O과-xxx('yy.mm.dd) OOO체계 개발시험평가 계획				
시험항목		시험기준	자동화 도구명	확인결과		
정적 시험	코딩규칙 검증	총 xx개 규칙 (첨부)				
	취약점 점검결과	총 xx개 규칙 (첨부)				
	소스코드 메트릭	메트릭 종류 및 제한값 (첨부)				
동적 시험	코드 실행률	목표값 달성여부 OOO Coverage 100%				
보안성 시험		행자부 SW 개발보안 가이드				
첨부자료		- 시험 결과 요약 : 별첨 - 세부 시험 결과 : 별지 1 - 기타				
의견 (기술지원기관)						
입 회 기 관		소속	성명	서명	비고	
사업관리부서						
기술지원기관						
개발주관기관						
...						

(별지2-별첨1)

시험 결과 요약(예시)

1. 시험 개요

담당 IPT	000사업팀
사업명 / 시험대상	000체계 / 0000CSCI
시험종류 / 적용도구	정적시험 / XXXX 적용
시험 기간	2011. 4. 10 ~ 6. 10
시험 장소	

2. 시험 결과 종합

SW CSCI	검사 라인수 (SLOC)	오류 검출건수	거짓 경보 (False Alarm) 검출건수	결과 판정
				충족 or 미달
총 계				

(SLOC : Source Line Of Code)

3. SW 형상항목(CSCI)별 세부 시험결과

순 번	시험 차수	시험일자	시험대상 SW	검사 라인수 (Sloc)	오류수	False Alarm	비고
1	1차	2011.05.01					
	1차	2011.05.01					
총 계							

4. 자동화도구 시험결과보고서

- 별도 붙임

(별지2-별첨2)

OOO체계개발사업 OOOO분석장치 운용 CSC

SW신뢰성 정적시험 코딩규칙검증 False Alarm 보고서(예시)

코딩규칙	조건식 규칙 -나)	FA내용	조건문의 결과가 항상 True 혹은 False라면 이는 조건문으로 작성해서는 안된다.		
개수	4	파일명	abcefg.c	코드위치	665
코드내용 (pseudo code) (도구에서 오류 발생한 코드 부분 화면 출력 혹은 pseudo Code 작성)					
time()함수에서 현재 시각을 가져오는 부분에서 비정상적인 리턴이 넘어올 경우 예외처리를 위해 ERROR값 여부를 판정하는 조건문에서 Alarm이 발생하였으나 다음과 같이 OS 제조사에서 제공하는 Reference 확인 결과 ERROR는 가능한 리턴값임					
FA분석	time()함수에서 시스템의 현재 시각을 가져오는 부분에서 정상적으로 시각을 가져오지 못하는 상황에 대한 예외 처리를 하였으나 제조사 Reference를 참고하여 변수의 타입을 설정하였음에도 SW신뢰성 시험 도구가 오류로 검출한 내용임.				
결론	본 내용은 OS사에서 제공하는 함수를 사용하였으며, 해당 리턴값을 준수하였으므로 기능이 정상적으로 동작하는 것을 unit test 및 integration test에서 확인하였음. SW신뢰성 및 기능에 문제가 없음.				
동일사항 파일명 및 코드위치					
번호	파일명		코드위치		비고
1	abcdefg.c		700		
2	abcdefg.c		801		
3	defg.c		254		

※ 코드 위치는 라인 정보 대신 소스코드 상에서 해당 라인을 식별할 수 있는 주석에 대한 설명으로 대체 가능 (예: /* ○○○-Rule XX.X False Alarm */)

(별지2-별첨3)

OOO체계개발사업 OOOO분석장치 운용 CSC

SW신뢰성 동적시험 미달성 분석 보고서(예시)

개수	6	파일명	abcdef.cpp	코드위치	643~645
코드내용 (pseudo code) (도구에서 오류 발생한 코드 부분 화면 출력 혹은 pseudo Code 작성)					
미달성 사유	입력되는 파라미터의 값의 NULL체크를 위한 구문으로 시스템에서 매우 예외적인 상황에서만 동작하는 방어 코드임. 본 소스코드를 분석하여 시스템의 예외적인 상 황에서도 동작하는 코드임을 확인함 [X-XXX-SFR-033] 오디오 예외처리				
미달성 동일사항 파일명 및 코드위치					
번호	파일명		코드위치		비고
1	abc.cpp		343~345		
2	abc.cpp		242~247		
3	abcdef.cpp		289~291		
4	abcdef.cpp		156~158		
5	frgh.cpp		567~569		

※ 코드 위치는 라인 정보 대신 소스코드 상에서 해당 라인을 식별할 수 있는 주석에 대한 설명으로 대체 가능 (예: /* ○○○-Rule XX.X False Alarm */)

【부록 8】 국방아키텍처 필수 작성 산출물

국방아키텍처 필수 작성 산출물

● : 필수

국방정보기술 아키텍처 구분	구분	AV				OV								SV													TV			
		1	2	3	4	1	2	3	4	5	6	7			1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3
												a	b	c																
기관별 아키텍처/ 전군적 아키텍처 구축	현행 아키텍처	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	목표 아키텍처	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	이행계획	일반 문서로 작성																												
단위시스템 아키텍처 구축	소요기획 단계	OV-1(운용개념도), SV-1b(체계관계도) 만 일반문서로 작성																												
	선행연구 ·탐색개 발단계	●	●			●	●	●	●	●		●	●		●	●		●	●	●	●	●				●	●	●		
	시스템 개발단계	●	●			●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●				●	●	●	●	

- AV(All View) : 공통관점 산출물
- OV(Operational View) : 운용관점 산출물
 - * OV6은 6a, 6b, 6c로 구성
- SV(Systems View) : 체계관점 산출물
 - * SV1은 1a, 1b, SV10은 10a, 10b, 10c로 구성
- TV(Technical standards View) : 기술표준관점 산출물

※ 국방아키텍처 프레임워크(MND-AF) 최신 버전을 준용하여 필수 산출물을 전자적으로 작성하고, 획득 단계별 아키텍처 산출물 작성 시 이전 단계의 산출물을 보완 구체화하는 것이 원칙이다. 단 소요기획 단계 필수 산출물은 일반문서로 작성하여 소요제기 관련 문서에 첨부하여 제출할 수 있다.

※ 국방아키텍처 산출물 종류

관점	식별자	명칭	설명
공통	AV-1	아키텍처 개요/요약정보	아키텍처 목적, 범위, 사용자, 환경 및 분석 결과
공통	AV-2	통합사전	AF에 사용되는 용어의 정의
공통	AV-3	아키텍처 원칙 정의서	OA, SA, TA, AF 관리의 원칙정의
공통	AV-4	기관의 비전 및 미션	국방 비전 및 미션, 목표, 전략을 기술하고 이행과제를 정의
운용	OV-1	운용개념도	임무수행, 핵심운용조직, 운용능력 등에 대한 총체적인 모습을 표현, 계획자 시각에서 기관의 업무와 관련조직, 관련 체계를 정의
운용	OV-2	운용노드연결기술서	운용노드에서 수행되는 활동 및 노드간의 정보흐름 표현
운용	OV-3	운용정보교환목록	노드간 교환되는 정보(IEX)의 속성 표
운용	OV-4	조직관계도	조직간의 지휘, 통제, 조정의 관계도
운용	OV-5	운용활동모델	업무활동의 계층과 흐름도
운용	OV-6a	운용규칙모델	임무, 작전, 업무 및 아키텍처 등에 대한 제약사항을 정의
운용	OV-6b	운용상태전이 기술서	운용노드 또는 활동의 상태가 변화함으로써 다양한 사건에 응답하는 방법을 그래픽적인 방법으로 기술
운용	OV-6c	운용사건추적 기술서	특정 시나리오에 따른 대상 운용노드간의 정보 교환 내용을 시간중심으로 표현
운용	OV-7a	데이터 구성도	데이터 분류체계를 구성하는 주요 구성항목을 포함관계 형태로 작성
운용	OV-7b	개념데이터 모델	데이터 분류체계의 데이터 그룹 단위로 개념엔티티 타입 간의 관계를 작성
운용	OV-7c	논리데이터 모델	논리엔티티타입을 식별하고 논리엔티티타입간의 관계를 작성
체계	SV-1a	체계정의 기술서	논리적인 체계정의서, Portfolio, Inter/Extra/Internet, 기반체계 식별 등
체계	SV-1b	체계인터페이스 기술서	체계 노드간 인터페이스 정의 기술서
체계	SV-2	체계통신 기술서	물리적 통신노드와 이들 간의 통신경로 표현
체계	SV-3	체계관계추적 상관표	정의된 체계간의 관계도 정의

체계	SV-4	체계기능 기술서	체계 기능정의
체계	SV-5	운용활동 대 체계기능 추적 상관표	운용활동과 체계기능과의 대응관계 표현
체계	SV-6	체계데이터교환 목록	체계간 데이터 교환 내역 표현
체계	SV-7	체계성능요소 목록	정량적, 정성적 체계 성능특성 기술서
체계	SV-8	체계진화 기술서	시간에 따른 체계 현대화 계획을 기술
체계	SV-9	체계기술예측 목록	체계진화에 요구되는 기술표준/제품을 기술
체계	SV-10 a	체계규칙모델	아키텍처, 체계 또는 체계 컴포넌트, 체계 기능 등에 대한 제약사항 정의
체계	SV-10 b	체계상태전이 기술서	체계 또는 체계기능이 그들의 상태를 변화함으로써 다양한 사건에 응답하는 방법을 그래픽적인 방법으로 기술
체계	SV-10 c	체계사건추적 기술서	특정 시나리오에 따른 대상 체계/체계기능 간의 체계데이터 교환 내용을 시간중심으로 표현
체계	SV-11	물리구조 모델	물리적 데이터 모델정의
체계	SV-12	체계기반구조 기술서	체계 Technology 관계, 하드웨어 및 소프트웨어 구성 표현
체계	SV-13	체계보안 기술서	보안 정책 및 영역정의, 체계 보안 구성요소별 Logical Model
기술	TV-1	기술표준 목록	아키텍처에 적용할 수 있는 기술표준
기술	TV-2	미래표준 목록	예상되는 새로운 표준
기술	TV-3	제품 목록	적용할 솔루션에 대한 서술

【부록 9】 표준 개발프로세스와 국방 CBD 개발 프로세스간 매핑표

표준 개발프로세스와 국방CBD 개발프로세스 간 매핑표

※ 표준 개발산출물 중 “√” 표시된 필수 산출물을 제외하고는 상응하는 국방CBD산출물로 대체가능하며, 필수 산출물에 해당하는 CBD산출물은 생략 가능

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
개발준비		소프트웨어 수명주기 모델 선택	• 체계개발실행계획서				관련 계획서 검토 및 확정
		개발계획 작성					
분석	체계 요구사항 분석	체계/요구사항 정의	• 체계요구사항명세서(✓) • 소프트웨어개발계획서 (체계개발실행사에 미포함시) • MND-AF 산출물 (AV,OV)(✓)	요구사항 식별	도메인 모델링	• 도메인 명세서 • 용어집	체계 요구사항 확정(SRR)
		체계 운용개념/요구 사항 검토			현행시스템 분석	• 현행 시스템 분석서	
					비즈니스 모델링	• 비즈니스 정의서	
					요구사항 정의	• 요구사항 정의서 (요구사항 추적표)	
	체계 구조설계	체계 구조설계	• 체계설계기술서(✓) • MND-AF 산출물 (SV,TV)(✓)	아키텍처 정의	시스템 아키텍처정의	• 시스템 아키텍처 정의서	체계 설계 확정(SFR)
		체계 구조설계 검토			표준 지침 수립	• 표준 지침서	

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
분석	소프트웨어 요구사항 분석	소프트웨어 요구사항 정의	• 소프트웨어요구사항 명세서(✓)	요구사항 분석	유스케이스 모델링	• 유스케이스 명세서	소프트웨어 요구사항 확정 (SSR)
					연동소요 분석	• 연동소요 정의서 (연동항목소요서 IER)	
		소프트웨어 요구사항 검토			사용자 인터페이스 정의	• 사용자인터페이스 정의서	
					클래스 모델링	• 클래스 정의서	
					자료구축 계획수립	• 자료구축 계획서	
설계	소프트웨어 구조설계	소프트웨어 구조 정의 및 설계	• (개략)소프트웨어설계기술서 • (개략)인터페이스설계기술서 • (개략)데이터베이스설계기술서 • MND-AF 산출물(갱신)	개략설계	컴포넌트 식별	• 컴포넌트 아키텍처 명세서	소프트웨어 개략설계 확정 (PDR)
		인터페이스 설계			인터페이스 상호작용 명세	• 인터페이스 상호작용 명세서	
		데이터베이스 설계			컴포넌트 명세	• 컴포넌트 명세서	
					사용자 인터페이스 명세	• 사용자인터페이스 설계서	
		소프트웨어 구조설계 검토			데이터모델링	• 데이터 설계서 (코드설계서)	

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
설계	소프트웨어 상세설계	소프트웨어 구성요소 상세설계	• (상세)소프트웨어설계기술서 • (상세)인터페이스설계기술서 • (상세)데이터베이스설계기술서 • 인터페이스통제문서(✓) • 소프트웨어통합시험계획서(초안) • MND-AF 산출물	상세설계	컴포넌트 구현 설계	• 컴포넌트 설계서 • 트랜잭션 설계서	소프트웨어 상세 설계 확정 (CDR)
		사용자 인터페이스 구현 설계			• 사용자 인터페이스 설계서		
		데이터베이스 설계			• 데이터베이스 설계서		
		연동 설계			• 연동 설계서 (연동항목 테이블 매핑 설계서)		
		자료 구축 설계			• 자료구축 설계서		
		시스템 보안설계			• 시스템 보안설계서		
		테스트 준비		테스트 계획	• 테스트계획서		
				소프트웨어 단위테스트 준비	• 소프트웨어 단위테스트기술서		
	소프트웨어 통합테스트 준비	• 소프트웨어 통합 테스트 기술서					

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
구현 및 테스트	소프트웨어 구현	단위 소프트웨어 구현 및 데이터베이스 개발	- 소스코드/실행 프로그램 코드 (라이브러리/오브젝트코드 포함) - 소프트웨어통합시험계획서 - 소프트웨어통합시험절차서	구현	데이터베이스 구현	• 물리적 데이터 베이스	
		단위 소프트웨어 및 데이터베이스 시험 준비			컴포넌트 구현	• 컴포넌트 코드	
		단위 소프트웨어 시험			사용자 인터페이스 구현	• 사용자 인터페이스 코드	
		소프트웨어 코드 및 단위시험 결과 검토		단위 테스트 소프트웨어통 합테스트 준비	단위테스트 수행	• 소프트웨어 단위 테스트기술서	
		소프트웨어 통합시험계획서 갱신 및 시험절차서 개발			소프트웨어 통합테스트 준비	• 소프트웨어 통합 테스트 기술서	

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
구현 및 테스트	소프트웨어 통합 및 시험	소프트웨어 통합 및 시험	• 소프트웨어통합시험결과서 • 사용자/관리자 문서(초안) (사용자지침서, 체계운영자 지침서, 소프트웨어버전 기술서, 소프트웨어산출물 명세서) • 체계통합시험계획서 • 체계통합시험절차서	소프트웨어 통합 및 테스트	소프트웨어 통합 및 테스트 수행	• 소프트웨어 통합 테스트 기술서	
		소프트웨어 통합결과 검토		지침서 작성	지침서 작성	• 사용자지침서 • 운용자지침서	
		사용자/관리자 문서 개발					
		체계통합시험 계획서 및 시험절차서 개발		시스템 통합 테스트 준비	시스템 통합테스트 준비	• 시스템 통합테스트 기술서	
	체계 통합 및 시험	체계 통합 및 시험	• 체계통합시험결과서 • 소프트웨어설치계획서 • 소프트웨어전이계획서 • 소프트웨어설치절차서 • 소프트웨어목록명세서 • 개발시험평가계획서(안) • 개발시험평가절차서	시스템 통합 및 테스트	시스템 통합 및 테스트 수행	• 시스템 통합테스트 기술서	시험평가계획 승인 (TRR)
		체계 통합결과 검토		시스템 설치 준비	시스템 설치계획 수립	• 시스템설치계획서	
		소프트웨어 설치계획 작성					
		개발시험평가 계획서(안) 및 시험절차 개발					

표준 개발단계		표준 개발작업	표준 개발산출물	CBD 활동	CBD 작업	CBD 산출물	공식 이정표 (합동검토 시점)
시험평가 및 인도	개발시험 평가	개발시험평가 환경구성 및 시험평가	- 개발시험평가결과보고서 - 운용시험평가지원계획서 - 사용자/관리자 문서				
		개발시험평가 결과 검토					
		운용시험평가지원 계획 작성					
	운용시험 평가	운용시험평가 지원	운용시험평가지원결과서				
	소프트웨어 설치	소프트웨어 설치 수행	소프트웨어설치결과서	시스템 설치	시스템 설치 실시	시스템 설치 결과서	
	규격화	소프트웨어 국방규격화	규격화 기술자료				
	인도	소프트웨어 산출물 인도 및 발주자지원		인수 지원	인수 지원 실시		

분류체계 식별자 관리

목 차

【별지 1】 분류체계 식별자 부여 규칙

【별지 2】 분류체계 식별자 초안

【별지 3】 분류체계 식별자 목록

【별지 4】 분류체계 내장형 소프트웨어 총괄자료 목록

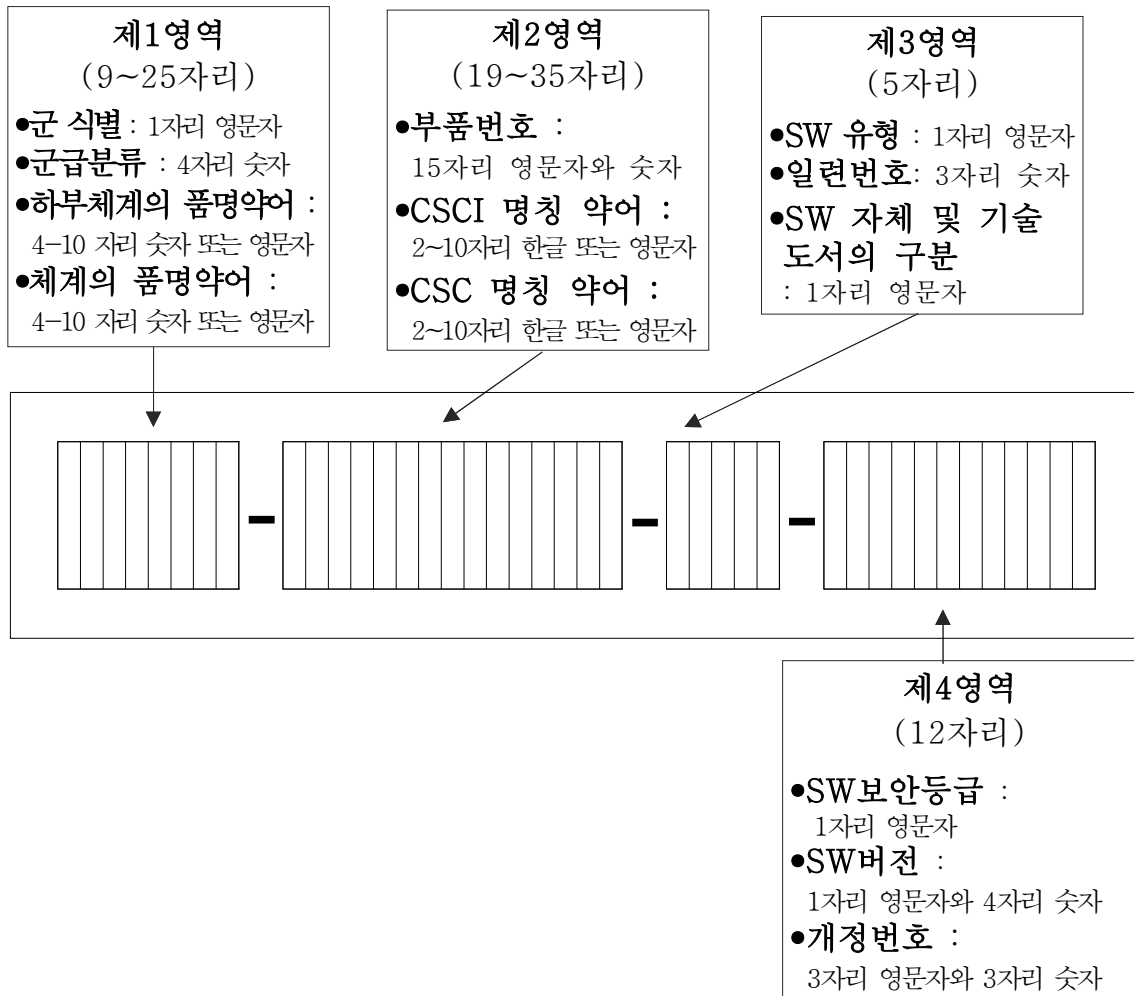
【별지 5】 분류체계 속성자료 목록

【별지 1】 분류체계 식별자 부여 규칙

1. 식별번호 구성

가. 연구개발 주관 기관은 소프트웨어 형상관리 품목과 형상관리 문서에 분류체계 식별자를 부여한다.

나. 무기체계 내장형 소프트웨어의 분류체계 식별자는 45~77자리의 영문 및 숫자와 4개 영역을 구분하는 “-” 로 구성한다.



예) A2350SCUK2-FPD75740/SCU/CON-F001A-CV5.40REV001

2. 제1영역 식별번호 부여 규칙

가. 제1영역의 첫 번째 식별자인 군 식별은 그 체계를 사용하는 군(軍)을 식별하기 위하여 한 자리 위치의 영문자로 표시한다. 군 식별자는 육군(A), 해군(N), 공군(F), 해병대(M), 공통(C)로 나타내는 것을 기본으로 한다.

나. 제1영역의 두 번째 식별자인 군급분류는 목록화 교범인 군급분류번호집(H2)에 의하여 작성하며, 무기체계 및 하부체계의 분류를 식별하기 위한 4자리 숫자로 표시한다.

- 다. 제1영역의 세 번째 식별자인 하부체계명 및 주체계는 무기체계명(장비명)을 최소 각각 4자리에서 최대 10자리의 한글 또는 영문자로 표시한다. 이때, 품명에 붙이는 AN, 사선(/) 및 구분 표시(-)를 생략한다.(예를 들면, AN/VRC-46은 VRC46으로만 식별자에 표시하며, K-2전차의 경우 K2로 사용한다.)
- 라. 해·공군 무기체계일 경우 무기체계 부분품을 탑재하고 있는 주체계명을 하부체계명의 식별자에 이어 표시할 수 있다.(예를 들면, VRC46PKX 또는 VRC46F15C와 같이 필요시 추가가 가능하다.)
- 마. 제1영역의 세 개의 식별자를 종합적으로 나타낸 식별자 표시방법은 “A2350SCUK2-”로서 이 의미는 ‘육군에서 사용하는 기동무기체계중 전투형 전차 K-2흑표전차의 현수전자 제어장치’임을 나타낸다.

군 식별 (1자리 영문자)	무기체계 분류 (4자리 숫자)	하부체계명 (4자리 또는 10자리 한글 또는 영문자)	체계명 (4자리 또는 10자리 한글 또는 영문자)	-
-------------------	---------------------	--	--------------------------------------	---

3. 제2영역 식별번호 부여 규칙

- 가. 제2영역의 첫 번째 식별자인 부품번호는 형상관리품목으로 지정된 수준의 CSCI는 물론 그 하부체계인 CSC의 식별을 용이하게 하기 위하여 『국방규격·표준서의 서식 및 작성에 관한 지침』에 의거 부여된 부품번호를 사용한다. 이때 상용품은 상용품의 부품번호를 그대로 적용하고, 외국에서 도입된 품목인 경우 외국의 부품 그대로 적용한다.
- 나. 제2영역의 두 번째 식별자는 내장형 소프트웨어의 기능을 나타내는 CSCI 명칭/CSC 명칭 순으로 표시하며 영문약어 또는 한글약어를 사용한다. 이때, CSCI/CSC 식별자의 자리수는 각각 최소 2개에서 최대 10개로 한다.
- 다. 제2영역의 식별자 표시방법은 “FPD75740/SCU/CON-”로서 제2영역의 식별자의 의미는 “현수전자제어 CSCI의 제어 기능을 수행하는 제어 CSC 소프트웨어”를 의미한다.

부품번호 (15자리 영문자 및 숫자)	CSCI명칭 (2~10자리 영문약어 또는 한글약어)	CSC명칭 (2~ 10자리 영문약어 또는 한글약어)	-
-------------------------	------------------------------------	------------------------------------	---

4. 제3영역 식별번호 부여 규칙

가. 제3영역의 첫 번째 식별자는 내장형 소프트웨어의 기능별 유형을 식별하기 위한 소프트웨어 유형을 나타내며, 소프트웨어 유형은 다음 각호와 같다.

- F : 운용기능을 수행하는 소프트웨어
- S : 지원기능을 수행하는 소프트웨어
- C : 운용과 지원 기능을 복합적으로 수행하는 소프트웨어
- T : 자체 진단용 목적의 기능을 수행하는 소프트웨어
- U : 현재 개발을 위한 시험중에 있는 소프트웨어

나. 제3영역의 두 번째 식별자는 내장형 소프트웨어의 CSCI, CSC 식별번호를 지정한 순서를 나타내는 일련번호이다. 일련번호는 001에서 999까지 계속되는 세 자리 숫자로서 나타낸다.

다. 제3영역의 세 번째 식별자는 소프트웨어 와 기술도서를 구분하기 위한 것으로 소프트웨어를 나타낼 때는 A, 관련 기술문서를 나타낼 때는 D를 붙여 나타낸다.

라. 제3영역의 식별자 표시방법은 “F001A-”로서 제3영역의 식별자의 의미는 “운용 용도로 식별번호를 처음 부여한 소프트웨어”를 의미한다.

소프트웨어 유형 (1자리 영문자)	일련번호 (3자리 숫자)	소프트웨어자체 또는 기술도서를 구분 (1자리 영문자)	-
-----------------------	------------------	-------------------------------------	---

5. 제4영역 식별번호 부여 규칙

가. 제4영역의 첫 번째 식별자는 내장형 소프트웨어의 비밀등급을 나타내며, 한 자리 영문자로 다음 각 호와 같이 나타낸다.

- T : 1급 비밀
- S : 2급 비밀
- C : 3급 비밀
- F : 대외비
- U : 일반

나. 제4영역의 두 번째 식별자는 버전번호를 의미하며, 버전번호는 CSCI, CSC의 기준선 또는 각각의 변경을 나타내는 식별자로서 영문자 V와 4자리 위치의 구독점이 있는 숫자로 나타낸다.(예를 들면, 버전 5.40이면 V5.40으로 표시한다.)

다. 제4영역의 세 번째 식별자는 CSCI의 결함을 발견하여 이를 수정하였을 때 부여하는 수정번호로서 수정이 완료되면 형상관리품목의 수준에 따라 최초의 기준선 CSCI, CSC를 대체하게 된다. 이 때 개정번호는 001, 002, 003, ... 과 같이 세 자리 숫자를 수정의 의미를 나타내는 영문자 REV 다음에 나타낸다.

라. 제4영역의 식별자 표시방법은 “CV5.40REV001”로서 제4영역의 식별자 의미는 “3급 비밀로 버전은 5.40이며 처음 작성된 소프트웨어”라는 것을 의미한다.

소프트웨어 보안등급 (1자리 영문자)	소프트웨어버전 (1자리 영문자 및 4자리 숫자)	소프트웨어수정번호 (3자리 영문자 및 3자리 숫자)
-------------------------	----------------------------------	------------------------------------

【별지 2】 분류체계 식별자 초안

분류체계 식별자 초안

순번	분류코드명칭	CSCI명	CSC명	분류체계 코드 1영역		
				군 식별	군급분류	하부체계명 및 체계명
1	K2SW	현수제어	제어	A	1230	K2HSC
2	K2SW	현수제어	운용통제	A	1230	K2HSC
	①	②	③	④	⑤	⑥

※ 별지 2는 PDR시 제출한다.

< 작 성 방 법 >

- ① 분류코드명칭 : “무기체계 약어 + SW”로 기술한다.
- ② CSCI명 : 소프트웨어 형상관리 단위인 CSCI명칭을 기술한다.
- ③ CSC명 : 소프트웨어 형상관리 단위인 CSC명칭을 기술한다.
- ④ 군식별 : 체계를 사용하는 군을 기술한다. 예) 육군 : A, 공군 : F, 해군 : N, 해병 : M, 공통 : C
- ⑤ 군급분류 : 목록화 교범인 군급분류집(H2)에 의해 작성하며 무기체계 및 하부체계의 분류를 식별하기위한 4자리의 숫자를 기술한다.
- ⑥ 하부체계명 및 무기체계명 : 4자리에서 최대 10자리의 숫자 또는 영문자로 표시한다. 여기서 유의할 것은 통상 품명(nomenclature)에 붙이는 AN과, 사선(/) 및 구분 표시(-)를 생략한다. 예를 들면, AN/VRC-46은 VRC46으로만 식별자에 표시하며, K-2전차의 경우 K2로 사용한다.

【별지 3】 분류체계 식별자 목록

분류체계 식별자 목록

순번	분류코드명칭	CSCI명	CSC명	분류체계 코드											
				1영역			2영역			3영역			4영역		
				군식별	군급분류	하부체계명 및 체계명	부품번호	CSCI 약어	CSC 약어	SW 유형	일련 번호	SW / 기술도서 구분	SW 보안 등급	SW 버전	SW수정 번호
1	K2SW	현수제어	제어	A	1230	K2HSC	FPD75740	SUC	CON	C	001	D	C	V1.00	REV001
	별지 1 설명 참조						①	②	③	④	⑤	⑥	⑦	⑧	⑨

※ 별지 3은 CDR시 제출한다.

< 작 성 방 법 >

- ① 부품번호 : 형상관리품목으로 지정된 수준의 CSCI는 물론 그 하부체계인 CSC의 식별을 용이하게 하기 위하여 『국방규격·표준서의 서식 및 작성에 관한 지침』에 의거 부여된 부품번호를 사용한다. 이때, 상용품은 상용품의 부품번호를 그대로 적용하고, 외국에서 도입된 품목인 경우 외국의 부품 그대로 적용한다. 단, CDR시 부품번호 생성이 불가능한 경우 규격화시 추가 작성한다.
- ② ~ ③ CSCI/CSC명칭 : 소프트웨어 형상관리 단위인 CSCI/CSC 명칭의 영문(한글약어 사용가능) 약어로 기술한다. 이때, CSCI/CSC 식별자의 자리수는 각각 4 ~ 10자리로 한다.
- ④ 소프트웨어 유형 : 내장형 소프트웨어의 기능별 유형을 식별하기 위한 소프트웨어 유형을 나타낸다.

예) **F** : 운용기능을 수행하는 소프트웨어, **S** : 지원기능을 수행하는 소프트웨어, **C** : 운용과 지원기능을 복합적으로 수행하는 소프트웨어,

T : 자체진단용 목적의 기능을 수행하는 소프트웨어, **U** : 현재 개발을 위한 시험중에 있는 소프트웨어

⑤ 일련번호 : 내장형 소프트웨어의 CSCI, CSC 식별번호를 지정한 순서를 나타내는 일련번호이다. 일련번호는 001에서 999까지 계속되는 세 자리 숫자로 기술한다

⑥ 소프트웨어 자체 또는 기술도서를 구분 : 소프트웨어 와 기술도서를 구분하기 위한 것으로 소프트웨어를 나타낼 때는 A, 관련 기술 문서를 나타낼 때는 D로 기술한다.

⑦ 소프트웨어 보안등급 : 내장형 소프트웨어의 비밀등급을 나타낸다. 1급 비밀 : T, 2급 비밀 : S, 3급 비밀 : C, 대외비(군전용) : F, 일반(상용) : U로 기술한다.

⑧ 소프트웨어 버전 : 버전번호는 CSCI, CSC의 기준선 또는 각각의 변경을 나타내는 식별자로서 영문자 V와 4자리 위치의 구독점이 있는 숫자로 기술한다.

예)버전 1.00이면 V1.00으로 표시한다.

⑨ 소프트웨어 수정번호 : CSCI, CSC의 결함을 발견하여 이를 수정하였을 때 부여하는 수정번호로서 수정이 완료되면 형상관리품목의 수준에 따라 최초의 기준선 CSCI, CSC를 대체하게 된다. 이 때 개정번호는 001, 002, 003, ... 과 같이 세 자리 숫자를 수정의 의미를 나타내는 영문자 REV 다음에 기술한다.

【별지 4】 분류체계 내장형 소프트웨어 총괄자료 목록

분류체계 내장형 소프트웨어 총괄자료 목록

분류코드명칭	무기체계명	운용군	획득유형	획득방법	무기체계 분류	시제업체	개발년도	배치년도	개발언어	적용 O/S	LOC	개발관리툴	미들웨어	DBMS	기간(월)	인력투입수(명)	SW개발비용(만원)	비용산정방법	산출물(종)	규격화(종)	CSCI 수
K2 SW	00	육군	연구개발	국과연개발	기동무기체계-전차-전투용	00	2000	2008	C/C++	VxWorks	660000	DOORS	없음	없음	10	00	00	M/M	12	5	5
①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱	⑲	⑳	㉑	㉒

※ 엑셀 파일로 작성

< 작성 방법 >

- ① 분류코드명칭 : “무기체계 약어 + SW”로 기술한다.
- ② 무기체계명 : 해당 무기체계명을 기술한다.
- ③ 운용군 : 체계를 사용하는 군을 기술한다. 예) 육군, 공군, 해군, 해병, 공통
- ④ 획득유형 : 체계를 획득하는 유형을 기술한다. 예) 연구개발, 국내구매, 해외구매 등
- ⑤ 획득방법 : 체계를 획득하는 방법을 기술한다. 예) 국방과학연구소주관개발, 업체투자개발, 업체주관개발
- ⑥ 무기체계 분류 : 국방전력발전업무훈령 별표2를 참조하여 대분류, 중분류, 소분류를 작성한다.
- ⑦ 시제업체 : 소프트웨어를 개발하는 주업체에 대해 기술한다.
- ⑧ 개발년도 : 소프트웨어 개발년도를 기술한다.
- ⑨ 배치년도 : 소프트웨어가 탑재되어 무기체계가 배치된 년도를 기술한다.

- ⑩ 개발언어 : 소프트웨어 개발언어에 대해 기술한다.
- ⑪ 적용O/S : 소프트웨어가 설치되어 운영되는 환경 운영체제(O/S)에 대해 기술한다.
- ⑫ LOC : 개발된 소프트웨어의 소스코드 라인수(Line Of Code)를 기술한다.
- ⑬ 개발관리툴 : 소프트웨어 개발시 사용된 대표적인 개발관리툴에 대해 기술한다. 예) 요구사항관리(DOORS), 형상관리(...), 품질관리 등
- ⑭ 미들웨어 : 소프트웨어 운영에 필요한 미들웨어를 기술한다.
- ⑮ DBMS : 소프트웨어에 사용된 DBMS를 기술한다.
- ⑯ 기간 : 소프트웨어 개발기간을 기술한다.(단위 : 월)
- ⑰ 인력투입수 : 소프트웨어 개발에 투입된 인원을 기술한다.(단위 : 명)
- ⑱ 소프트웨어 개발비용 : 소프트웨어 개발에 투입된 비용을 기술한다.(단위 : 만원)
- ⑲ 비용산정방법 : 소프트웨어 개발비용 산정방법을 기술한다. 예) FP, M/M, LOC, PRICE-S, COCOMO등
- ⑳ 산출물 : 소프트웨어 개발간 작성된 산출물의 종수를 기술한다.(단위 : 종)
- ㉑ 규격화 : 소프트웨어 개발간 작성된 산출물에 대하여 규격화를 실시한 종수를 기술한다.(단위 : 종)
- ㉒ CSCI수 : 소프트웨어 형상관리 단위인 CSCI수를 기술한다.(단위 : 수)

【별지 5】 분류체계 속성자료 목록

분류체계 속성자료 목록

분류코드명칭	분류체계코드	무기체계명	획득유형	무기체계분류	협력업체	개발언어	개발툴	적용 O/S	LOC	CASE 툴	미들웨어	DBMS	기간(월)	인투수(명)	SW개발비용(만원)	비용산정방법	SW주요기능	SW중요도	SW중요선정이유	사용설계	도면번호	규격번호	재고번호	디바이스	적용장부	기술분류코드	출판물일
K2 소프트웨어	...	00	연구개발	기동무기체계-전차-전투용	00	C++	VS6.0	VxWork	60000	ROSE	없음	없음	10	00	00	M/M	제어	상	재사용가능한 SW	적용	Q2350000	1122222	2350123412345	FDRAM	12345	1100	현수제어 CSC_SR S.HWP
①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱	⑲	⑳	㉑	㉒	㉓	㉔	㉕	㉖	㉗	㉘

※ 엑셀 파일로 작성

< 작성 방법 >

- ① 분류코드명칭 : “무기체계 약어 + SW”로 기술한다.
- ② 분류체계 코드 : CSCI/CSC별 1영역 ~ 4영역까지의 분류체계 코드를 기술한다.
예) A2350SCUK2-FPD75740/SCU/CON-F001A-CV1.00REV001

- ③ 무기체계명 : 해당 무기체계명을 기술한다.
- ④ 획득유형 : 체계를 획득하는 유형을 기술한다. 예) 연구개발, 국내구매, 해외구매 등
- ⑤ 무기체계 분류 : 국방전력발전업무훈령 별표2를 참조하여 대분류, 중분류, 소분류를 작성한다.
- ⑥ 협력업체 : 해당 CSCI/CSC별 소프트웨어 개발 협력업체를 기술한다.
- ⑦ 개발언어 : 해당 CSCI/CSC별 소프트웨어 개발언어를 기술한다.
- ⑧ 개발툴 : 소프트웨어 개발시 사용된 개발툴에 대해 기술한다. 예) Visual Studio 6.0, Code Composer 등
- ⑨ 적용O/S : 해당 CSCI/CSC별 소프트웨어가 설치되어 운영되는 환경 운영체제(O/S)에 대해 기술한다.
- ⑩ LOC : CSCI/CSC별 소프트웨어의 소스코드 라인수(Line Of Code)를 기술한다.
- ⑪ CASE툴 : CSCI/CSC별 소프트웨어 개발시 사용된 개발자동화툴에 대해 기술한다. 예) Rational ROSE, TOGETHER 등
- ⑫ 미들웨어 : CSCI/CSC별 소프트웨어 개발에 사용된 미들웨어를 기술한다.
- ⑬ DBMS : CSCI/CSC별 사용된 DBMS를 기술한다.
- ⑭ 기간 : CSCI/CSC별 소프트웨어 개발기간을 기술한다.(단위 : 월)
- ⑮ 인력투입수 : CSCI/CSC별 개발에 투입된 인원을 기술한다.(단위 : 명)
- ⑯ 소프트웨어 개발비용 : CSCI/CSC별 개발에 투입된 비용을 기술한다.(단위 : 만원)
- ⑰ 비용산정 방법 : CSCI/CSC별 개발비용 산정방법을 기술한다. 예) FP, M/M, LOC, PRICE-S, COCOMO 등
- ⑱ 소프트웨어 주요기능 : 해당 CSCI/CSC별 주요기능을 기술한다. 예) 제어, 연동, 처리, 고장진단 등
- ⑲ 소프트웨어 중요도 : 해당 CSCI/CSC별 중요도에 대해 자체 판단사항을 상, 중, 하로 기술한다.
- ⑳ 소프트웨어 중요도 선정이유 : 소프트웨어 중요도를 상, 중, 하로 선정한 이유를 기술한다.

예) 해당 무기체계의 주요핵심기능을 담당하는 소프트웨어, 무기체계 도메인별로 향후 재활용 비중이 높은 소프트웨어, 타 무기체계와 연동을 위한 소프트웨어, 향후 소프트웨어에 대한 업그레이드에 필요한 소프트웨어, 해당 무기체계 내에서만 사용되는 일반적인 소프트웨어인 경우 등 소프트웨어개발업체에서 의견 제시바람.

- ㉑ 재사용 설계 : 소프트웨어 개발간 재사용을 고려한 설계방법론을 사용하여 개발 여부를 기술한다. (모듈화 개발, CBD 적용, 기타 콤포넌트화 개발 등)
- ㉒ 도면번호 ~ ㉗ 기술분류코드 : “조달시스템”을 참고하여 작성한다.
- ㉘ 산출물 파일 : 해당 CSCI/CSC별 산출물 파일명을 기입하고 파일을 제출한다.
예) 현수제어 CSC SRS 문서일 경우 : 현수제어CSC_SRS.HWP, 현수제어CSC 소프트웨어일 경우 : 현수제어CSC_SW.ZIP
으로 기술한다.

형상변경 참조 양식

목 차

【별지 1】 형상변경요청서

【별지 2】 변경사항 세부내역

【별지 3】 담당자 검토의견서

【별지 4】 형상통제심의 의결서(단일안건)

【별지 5】 형상통제심의 의결서(다수안건)

[별지 1] 형상변경요청서

형상변경요청서			
분 야 명	①	요청번호	②
요 청 일	③	요청기관	④
요 청 자	⑤	형상통제 분류	⑥
우선순위	⑦		
제 목	⑧		
목 적			
⑨			
변경요청 사유			
⑩			
변경사항 (세부내역은 별지2 양식으로 별도 작성)			
변경 전		변경 후	
⑪			
영향성 분석			
분 석 자	⑫		
분석내용	1. 체계성능 2. 사업비용 3. 사업일정 4. 전력화지원 5. 기 타 6. 종합평가 ⑬		
변경승인			
승 인 자	⑭ (인)	승인일자	⑮
승인내용	변경 요청내용을 형상통제심의위원회 심의 요청		

※ 작 성 요 령

- ① 분 야 명 : 사업관리, 기반체계 등 대분류 분야명 기술
- ② 요청번호 : CR-YYYY-MMDD-XXX
(CR- 년 - 월일 - 일련번호)
- ③ 요 청 일 : YYYY년MM월DD일
- ④ 요청기관 : 소요군/방위사업청/개발업체/국과연/기품원 등
- ⑤ 요 청 자 : 형상변경 요청기관의 요청자
(소속, 계급/직책, 연락처, 성명)
- ⑥ 형상통제 분류 : I, II 급 기술변경, 규격완화, 면제
- ⑦ 우선순위 : 긴급, 보통
- ⑧ 제목 : 변경요청 내용을 쉽게 알 수 있도록 기재
- ⑨ 목적 : 변경요청을 하게 된 목적을 기재
- ⑩ 변경요청 사유
- ⑪ 변경사항 : 변경사항 세부내역[별지 2]에 상세하게 기술되므로 대표적인
변경사항만 간략하게 기술
- ⑫ 분 석 자 : 개발업체 해당분야 분석담당자(소속, 직책, 연락처, 성명)
- ⑬ 분석내용 : 항목별 영향평가결과를 구체적으로 기술
- ⑭ 승 인 자 : 요청기관 승인자(업체 사업관리자(PM) 등)
- ⑮ 승인일자 : YYYY년MM월DD일

[별지 2] 변경사항 세부내역

변경사항 세부내역				
요청번호	①			
대상 산출물	②			
③	변경 전		변경 후	
	장.절	본문	장.절	본문
	④	⑤	④	⑤
...	...		...	
③	변경 전		변경 후	
	장.절	본문	장.절	본문
	④	⑤	④	⑤

※ 작성요령

- ① 요청번호 : 형상변경요청서의 요청번호
- ② 대상 산출물 : 형상변경 요청으로 변경될 모든 산출물 명
- ③ 산출물 명
- ④ 해당 산출물에 대한 장, 절 ⑤ 해당 산출물에 대한 본문 내용 기술(필요시 별지 첨부)

[별지 3] 담당자 검토의견서

담당자 검토의견서(사업관리부서 작성)

1. 개 요
2. 요청번호 :
3. 제 목 :
4. 변경 요청내용

변경 전	변경 후

5. 담당자 검토결과

6. 영향성 판단

- 가. 체계 성능
- 나. 사업 비용
- 다. 사업 일정
- 라. 전력화지원
- 마. 타 형상항목 또는 타 체계에 대한 영향성 및 상호 운용성
- 바. 종합 의견

7. 결론/건의

예) 검토결과 내용으로 형상변경 승인을 건의함.

검 토 자 : 0 0 0 0 담당 소 령 홍 길 동 (인)

[별지 4 : 형상통제심의 의결서(단일 안전인 경우)]

00사업 형상통제심의회

☐ 형상통제심의위원회

☐ 형상통제실무회의

1. 일시/장소 :

2. 안 건

가) 분 야 명 :

나) 요청번호 :

다) 요청기관 :

라) 제 목 :

마) 형상통제 분류 : (I 급 또는 II 급)기술변경/규격완화/면제

바) 우선순위 : (긴급/보통)

사) 영향성판단

1) 체계 성능 :

2) 사업 비용 :

3) 사업 일정 :

4) 전력화지원 등 :

3. 심의결과 : (승인/조건부 승인/기각)

⇒ 필요시, 구체적인 심의결과 기술

4. 의결내용

참석자	소 속	직 급	성 명	의 건	서 명
위 원				(동의여부, 의견 기술)	
위 원					
위 원					
...					
위 원					

5. 결론 (위원장)

예) 상기 안전에 대한 변경을 승인함

주계약 업체 PM O O O (서명)

위 원 장 : 방위사업청 00사업팀장 대령 O O O (서명)

[별지 5 : 형상통제심의 의결서(다수 안전인 경우)]

00사업 형상통제심의회

☐ 형상통제심의위원회

☐ 형상통제실무회의

1. 일시/장소 :

2. 안전 및 심의결과 : 붙임 참조(별지 4-1의 "2", "3"항을 안전별 기술, 첨부)

순번	분야명	요청번호	제목	요청기관	심의결과

3. 의결내용

참석자	소 속	직 급	성 명	의 건	서 명
위 원				(동의여부, 의견 기술)	
위 원					
위 원					
...					
위 원					

4. 결론 (위원장)

예) 상기 안전에 대한 변경을 승인함

주계약 업체 PM O O O (서명)

위 원 장 : 방위사업청 00사업팀장 대령 O O O (서명)

【부록 12】 품질보증 참조 양식

품질보증 참조 양식

목 차

【별지】 품질보증계획서

【별지】 품질보증계획서

품질보증계획서

1. 개 요

- 1.1 목적
- 1.2 품질보증 대상
- 1.3 품질보증 적용범위

2. 품질관리 조직/역할/자원

- 2.1 조직구성
- 2.2 기관(부서)별 역할
- 2.3 필요자원(표준, 방법론, 도구 등)

3. 품질 요구사항

- 3.1 성능관련 품질 요구사항
- 3.2 개발관련 품질 요구사항
- 3.3 기타 품질 요구사항

4. 품질관리 계획 및 절차

- 4.1 단계별 품질관리 계획
- 4.2 품질관리 활동
 - 4.2.1 품질관리 활동계획
 - 4.2.2 검토회의(Review)
 - 4.2.3 검증(Verification)
 - 4.2.4 확인(Validation)
단위시험, SW통합시험, 체계통합시험, 시험평가 등 활동계획
- 4.3 품질보증
 - 4.3.1 산출물 보증
산출물 작성/검토 시기 등
 - 4.3.2 프로세스 보증
- 4.4 품질관리 활동 기록, 관리 및 배포 절차

5. 품질 공인 절차

- 5.1 참여 기관별 역할
- 5.2 공인 시기 및 주요 검토사항
(예) 단계별 검토회의에서 검토, 승인될 산출물 종류 및 절차 등 정의)
- 5.3 문제점 보고 및 시정조치

6. 협력업체 품질관리계획

- 6.1 품질보증 요구
- 6.2 품질검사 교육지원
- 6.3 품질 감사
- 6.4 검사
- 6.5 납품

7. 인도

- 7.1 포장
- 7.2 설치
- 7.3 인도

감리 참조 양식

목 차

【별지 1】 감리계획서

【별지 2】 감리보고서

【별지 3】 감리조치계획서

【별지 4】 감리결과 조치내역 확인보고서

【별지 5】 조달청 위탁 계약절차서

【별지 1】 감리계획서

00사업 00단계 감리계획서

1. 관련 근거

감리요청공문, 감리계약 공문 등 관련근거 기재

2. 감리 목적

감리 대상사업의 특성을 고려하여 감리의 목적을 기술

3. 감리 적용기준

감리에서 준용하고자 하는 기준, 표준, 지침 등을 명시

4. 감리 대상범위 및 단계

4.1 대상사업/단계 : 감리대상사업명/감리단계

4.2 감리대상 범위

4.2.1 세부 감리영역1

4.2.2 세부 감리영역2

• • •

5. 감리영역별 상세점검항목

감리대상 사업의 특성에 따라 현 감리단계에서 중점적으로 점검할 상세 점검항목을 감리영역별로 제시

6. 감리 일정

(착수회의, 감리 수행, 종료회의, 보고서 통보, 감리결과 조치내역 확인 등 일정 계획)

7. 감리원 편성

참여할 감리원(총괄감리원 및 감리원)의 성명과 감리원증 번호를 기재

8. 감리계획서 및 보고서 통보 기관

발주기관, 피감리인 등 통보대상 기관명 기재

9. 기타 행정사항

감리자료 제출 및 반납 절차, 감리일정 및 감리원의 변경 필요시 절차, 문서보안
관련사항 등을 기재

【별지 2】 감리보고서

00사업 00단계 감리보고서

I. 사업 개요

1. 사업명 : 감리대상 사업명
2. 사업 기간 : 년 월 일 ~ 년 월 일(개월)
3. 사업 목표
4. 사업 범위
5. 적용방법론(감리대상사업에서 적용하고 있는 개발 방법론 기재)
6. 감리대상 범위
7. 사업수행기관
 - 7.1 발주기관 : 발주기관명
 - 7.2 사업자 : 사업자명
8. 사업비 : 감리대상사업 총사업비

II. 종합의견

1. 각 분야별 현황

감리영역	평가	비 고
[감리영역 1] 예) 사업관리분야	* 사업관리분야 종합의견 기재	
[감리영역 2]		

2. 종합의견 : 사업 전반에 대한 종합의견

- 감리대상 정보화사업에 대한 전반적인 감리 의견으로 감리영역 전체를 대상으로 사업의 현황과 주요한 개선 필요사항에 대하여 요약 기술
- 감리대상사업의 목표와 현황, 사업수행에 성공적인 역할을 한 활동, 사업의 일정대비 목표달성도, 주요 문제점 및 개선사항 등을 중심으로 총괄감리원이 기술

3. 감리영역별 개선권고사항

개선권고사항	개선권고 내용	개선시점	중요도	해결주체
[감리영역 1]				
<u>[개선권고사항 제목 1]</u>		장기	상	사업관리부서
<u>[개선권고사항 제목 2]</u>		단기	중	연구개발 주관기관
...				
[감리영역 2]				
...				

Ⅲ. 개선권고사항

1. 감리영역 1

1.1 검토 현황

1.1.1 검토항목

상세 점검항목	결 과
<u>[상세점검항목 1]</u>	
<u>[상세점검항목 2]</u>	
...	
...	

1.1.2 검토의견 : 해당 감리영역에 대한 검토의견 기술

1.2 문제점 및 개선권고사항

1.2.1 문제점1 및 개선방향

1.2.2 문제점2 및 개선방향

• • •

2. 감리영역 2

상기와 동일

【별지 3】 감리조치계획서

00사업 00단계 감리조치계획서

1. 개 요

2. 추진 일정

2.1 현 장 감 리 기 간 : YYYY년 MM월 DD일~DD일(0주간)

2.2 조치계획 수립 : YYYY년 MM월 DD일~DD일

2.3 조치결과 확인요청 : YYYY년 MM월 DD일 제출 예정

2. 개선 및 권고사항 총괄표

감리 영역	개선권고사항(건)	세부 항목(건)	비고
예) 사업관리분야			

3. 개선 및 권고사항 조치계획

3.1 00분야

개선권고 내용	개선 시점	조치계획 내용	조치담 당	완료 예정일

3.2 00분야

• • •

【별지 4】 감리결과 조치내역 확인보고서

00사업 00단계 감리결과 조치내역 확인보고서

I. 개요

1. 관련근거

2. 조치내역 확인대상

2.1 대상 사업명 : 감리대상사업명

2.2 확인대상 범위 : [0단계] 감리결과 조치내역

3. 일정 및 참여자

3.1 조치내역 확인 기간 : YYYY년 MM월 DD일~DD일(0주간)

3.1.1 산출물 검토 : YYYY년 MM월 DD일~DD일

3.1.2 현장 확인 : YYYY년 MM월 DD일~DD일

3.2 참여자

3.2.1 총괄감리원 : 성명 (감리원증 발급번호)

3.2.2 수석감리원 : 성명 (감리원증 발급번호)

3.2.3 감리원 : 성명 (감리원증 발급번호)

3.2.4 감리원 : 성명 (감리원증 발급번호)

II. 확인결과

1. 전제조건 : 확인관련 전제조건 또는 제약사항

2. 종합의견 : 감리결과 조치내역에 대한 종합의견

3. 조치내역 확인결과

감리영역	개선권고내용	개선 시점	중요 도	조치 현황	검토의견	세부검토의견
예) 사업 관리		단기	상	조치완료	적정	
				조치중	미흡	
				반영불가	N/A	

※ ‘조치현황’ 범례

- 조치완료 : 해당 개선권고사항에 대해 조치를 완료하였다고 표시한 사항
- 조치 중 : 해당 개선권고사항에 대해 계획을 수립하고 조치 중이라고 표시한 사항
- 반영불가 : 해당 개선권고사항에 대해 피감리인과 발주기관의 협의 및 의사결정에 따라 조치하지 않기로 결정되었다고 표시한 사항

※ ‘검토의견’ 범례

- 적 정 : 해당 개선권고사항에 대해 적정하게 조치가 완료되었거나 적정한 계획 수립에 따라 양호하게 조치가 수행되고 있는 경우
- 미 흡 : 해당 개선권고사항에 대해 특별한 사유 없이 조치가 완료되지 않았거나 미흡하게 조치된 경우
- N/A : “반영불가” 사항 또는 “단순권고” 사항으로 조치내역 확인의 대상에서 제외된 경우

【별지 5】 조달청 위탁 계약절차

수 행 내 용	단 계	주 관
◇ 감리대상 여부, 기간, 예산, 범위 계약방법, 계약기관 등	감리계획 수립	사업팀
◇ 감리대상 사업내용, 업체에 제공할 내용, 업체제안서에 포함할 내용 등	제안요청서 및 계약특수조건 작성	사업팀
◇ 조달청 위탁구매	조달의뢰 및 위탁계약 의뢰	사업팀 일반장비계약팀
◇ 사전 규격공개 : 3 ~ 10일 정상공고 : 40일 이상 재공고, 긴급공고 : 10일 이상	입찰공고	조달청
◇ 감리대상 사업 및 제안요청 내용	사업설명회	사업팀
◇ 제안서 평가요소 및 기준에 따라 평가 ※ 종합평가(기술, 입찰가격)	제안서 접수 및 평가	기술(정량)평가: 사업팀 기술(정성), 가격평가: 조달청
◇ 제안서 평가(종합평가) 결과 고득점자	협상우선대상자 선정	조달청
◇ 기술, 조건 및 가격협상	협상	기술, 조건협상: 사업팀 가격협상: 조달청
◇ 협상결과에 따라 낙찰자 선정	낙찰자 선정	조달청
◇ 낙찰 업체와 계약체결	계약체결	조달청

공개SW 무기체계 적용 가이드라인

1. 목 적

본 가이드라인은 무기체계 연구개발 간 오픈소스 소프트웨어(이하 공개SW라 한다)에 대한 라이선스 관리, 소프트웨어 신뢰성 시험 · 보안성 시험 및 형상관리 등에 대한 적용 기준 및 업무수행 절차 등 세부사항을 제공함에 그 목적이 있다.

2. 관련 근거

가. 「방위사업관리규정」

나. 「무기체계 소프트웨어 개발 지원에 관한 규정」

다. 「무기체계 소프트웨어 개발 및 관리 매뉴얼」

3. 적용 범위

가. 적용 기관

1) 방위사업청

2) 국방과학연구소, 국방기술품질원, 방위산업기술지원센터 (이하 “기술지원기관”이라 한다)

3) 연구개발 주관기관(국방과학연구소, 업체)

4. 적용 기준

가. 라이선스

1) 무기체계 연구개발 시 소스코드 공개 의무가 있는 공개SW를 사용해서는 안 된다. 단, 라이선스 구매 및 저작권자와 협의 등을 통해 소스코드를 공개하지 않아도 되는 것을 증명한 경우에는 사용 가능하다.

※ 공개SW인 운영체제 사용 필요 시 운영체제만을 공개대상으로 하는 경우에 한하여 사업관리회의를 통하여 사용 여부를 결정한다.

2) 연구개발 주관기관은 공개SW 사용 시 특허권 등 권리관계의 포함 여부를 사전에 확인하여 지식재산권 분쟁이 발생되지 않도록 관리하여야 한다.

3) SW에 관한 지식재산권과 공개SW 라이선스 의무사항을 준수한다.

나. 소프트웨어 신뢰성(정적, 동적) 시험

1) 상세설계 단계에서 공개SW의 사용 필요성과 신뢰성 확보 여부를 제시하고 사업관리회의에서 신뢰성이 확보된 공개SW로 판단된 경우 신뢰성 시험 대상에서 제외할 수 있다.

※ 기술지원기관은 공개SW 신뢰성 확보 기준 명확화를 위해 기술지원 사례 축적 및 공개SW 통합관리(국방기술품질원) 등 지속적인 노력을 하여야 한다.

2) 신뢰성이 확보되지 않은 공개SW의 경우 [부록 7]의 “제3장 소프트웨어 동적 시험 절차 - 시험 기준 설정 방법”을 활용하여 아래와 같이 시험 수행 범위를 결정한다.

결함 영향도 (Severity)	결함 발생빈도 (Exposure)	결함 제어가능성 (Controllability)		
		C1	C2	C3
S1	E1	⊗	⊗	⊗
	E2	⊗	⊗	⊗
	E3	⊗	⊗	⊗
	E4	⊗	⊗	⊗
S2	E1	○	○	○
	E2	○	○	○
	E3	○	○	○
	E4	○	○	○
S3	E1	○	○	○
	E2	○	○	○
	E3	○	○	○
	E4	○	○	○
S4	E1	○	○	○
	E2	○	○	○
	E3	○	○	○
	E4	○	○	○

- 3) 시험 기준 설정은 공개SW가 포함된 독립적인 소프트웨어 구성요소 단위로 수행하며, 해당 소프트웨어 구성요소가 다른 소프트웨어 구성요소에 영향을 미칠 경우에는 연관된 구성요소 중 등급이 가장 높은 구성요소를 기준으로 시험 수행 범위를 결정한다.
- 4) ‘⊗’에 대해서는 상세설계 단계에서 사용하고자 하는 공개SW 적용 사례, 커뮤니티 성숙도를 제시한 경우 이를 종합적으로 판단하여 사업관리회의를 통해 신뢰성 시험 대상에서 제외할 수 있다.
- 5) ‘○’는 정적시험 및 동적시험을 수행하고, 동적시험 수행 기준은 [부록 7]의 “제3장 소프트웨어 동적시험 절차”를 따른다.
- 6) 수정을 한 경우에는 해당 함수에 대하여 신뢰성 시험을 수행하여야 한다.
- 7) 주체계 운영과 직접적인 관련이 없는 장비(예: 정비장비, 시험평가장비, 개발지원장비, CBT(Computer based training) 등)는 신뢰성 시험을 제외할 수 있다.

다. 소프트웨어 보안성 시험

- 1) 전장관리정보체계에 사용되는 공개SW에 대해서는 보안성 시험을 수행하는 것을 원칙으로 한다.
- 2) 주체계 운영과 직접적인 관련이 없는 장비(예: 정비장비, 시험평가장비, 개발지원장비, CBT(Computer based training) 등)는 보안성 시험을 제외할 수 있다.

라. 형상관리

- 1) 적용된 공개SW에 대한 보안취약점 패치 등을 통해 변경사항 발생 시에는 표준화 업무지침 형상통제업무 처리절차를 따른다.

5. 업무수행 절차

가. 사업추진 준비 단계

- 1) 사업관리부서장은 라이선스 위반 및 지식재산권 분쟁을 예방하기 위하여 제안 업체가 제안서 작성시 공개SW를 식별하고 적용 계획을 명시하도록 제안요청서를 작성 하여야 한다.
- 2) 기술지원기관은 제안요청서 검토 요청을 받은 경우, 공개SW를 적용 시 공개SW 적용 계획이 제안서에 포함될 수 있도록 제안요청서가 작성되었는지 확인한다.

나. 체계 요구사항분석 단계

- 1) 연구개발주관기관은 사용하고자 하는 공개SW를 식별하고, 신뢰성 확보 여부 (개발 시 관련 표준 준수, 신뢰성 시험 수행 여부 등)를 포함한 공개SW 사용 계획을 소프트웨어개발계획서(SDP)에 작성한다.
- 2) 사업관리부서장은 소프트웨어개발계획서(SDP)의 공개SW 사용 계획을 검토 하며, 필요 시 기술지원기관에 검토요청을 할 수 있다.
- 3) 기술지원기관은 소프트웨어개발계획서(SDP) 검토 요청을 받은 경우 공개 SW 사용 계획을 검토하고 그 결과를 사업관리부서장에게 제출한다.

다. 소프트웨어 상세설계 단계

- 1) 연구개발주관기관은 사용할 공개SW에 대한 라이선스 준수여부를 확인하여 [별지1] 공개SW 사용 및 라이선스 준수여부 확인 목록을 작성하고 소프트웨어설계기술서(SDD)에 포함한다.
- 2) 사업관리부서장은 소프트웨어설계기술서(SDD)의 공개SW에 대한 라이선스 준수여부를 검토하며, 필요 시 기술지원기관에 검토요청을 할 수 있다.
- 3) 기술지원기관은 소프트웨어설계기술서(SDD) 검토 요청을 받은 경우 공개SW 라이선스 준수여부를 검토하고 그 결과를 사업관리부서장에게 제출한다.
- 4) 연구개발주관기관은 사용한 공개SW에 대한 위험분석 결과 및 각 위험분석 등급별 신뢰성 시험계획을 소프트웨어통합시험계획서(STP)에 포함하여 작성한다.
- 5) 사업관리부서장은 소프트웨어통합시험계획서(STP)의 공개SW에 대한 신뢰성 시험 기준을 기술지원기관의 지원을 받아 확인한다.

라. 소프트웨어 구현 단계

- 1) 연구개발주관기관은 소프트웨어개발계획서(SDP)/소프트웨어통합시험계획서 (STP)에 명시한 프로그램 소스코드와 라이브러리에 대한 라이선스 준수여 부를 확인한다.
- 2) 연구개발주관기관은 공개SW에 대한 신뢰성 시험 및 보안성 시험을 지속적

으로 수행하여 결함을 사전에 제거한다.

마. 소프트웨어 통합 및 시험 단계

- 1) 연구개발주관기관은 공개SW에 대한 라이선스 준수여부를 검토하여 [별지1] 라이선스 준수여부 확인 결과를 소프트웨어통합시험결과서(STR)에 포함한다.
- 2) 연구개발주관기관은 소프트웨어통합시험계획서(STP) 및 소프트웨어통합시험절차서(STD)에 따라 공개SW에 대한 신뢰성 시험 및 보안성 시험을 실시한다.
- 3) 기술지원기관은 사업관리부서로부터 검토 요청을 받은 경우 연구개발주관기관이 제출한 공개SW 신뢰성 및 보안성 시험 결과와 라이선스 준수여부를 검토 및 확인하고 그 결과를 사업관리부서장에게 제출한다.
- 4) 사업관리부서장은 기술지원기관으로부터 받은 공개SW 검토 결과를 연구개발주관기관에 통보하고 수정 및 보완 조치 결과를 확인하며, 필요 시 기술지원기관에 검토요청을 할 수 있다.
- 5) 기술지원기관은 사업관리부서로부터 검토 요청을 받은 경우 공개SW 수정 및 보완 조치 결과를 검토하고 그 결과를 사업관리부서장에게 제출한다.

바. 규격화 단계

- 1) 연구개발주관기관은 개발된 소프트웨어(공개SW 포함)의 규격화 및 목록화 필요시 관련 자료를 「방위사업관리규정」, 「국방규격·표준서의 서식 및 작성에 관한 지침」, 「표준화 업무 지침」 등 관련 규정·지침에 따라 작성하여 사업관리부서 및 국방표준종합정보시스템에 제출한다.
- 2) 기술지원기관은 공개의무가 있는 소프트웨어 사용 여부 확인 등의 기술지원을 실시하고, 그 결과를 통합사업관리팀장에게 제출하여야 한다.

사. 양산 단계

- 1) 형상통제 제안기관(업체 포함)은 공개SW를 수정하는 경우, 무기체계 개발 단계의 소프트웨어 신뢰성 및 보안성 확보 활동 기준에 따라 수행하고, 그 결과를 형상통제 제안자료(형상통제제안서, 제안기관 자체검토서, 수정 후 국방규격 기술자료, 제안내용을 입증할 수 있는 관련자료 등)에 반영하여 제출한다.
- 2) 기품원은 형상통제 제안자료를 검토하고 그 결과를 형상관리책임기관에 제출한다

6. 참고자료

- 가. 오픈소스 소프트웨어 라이선스 가이드 (한국저작권위원회)
- 나. 공공 공개SW 거버넌스 가이드 (정보통신산업진흥원)
- 다. 공개SW 라이선스 가이드 (정보통신산업진흥원)

(별지1)

공개SW 사용 및 라이선스 준수여부 확인 목록

<< 공개SW 사용 목록>>

번 호	명칭	버전(Release 식별 가능 정보)	라이선스	공개 의무 여부	특허 침해 여부	라이브러리/관련 소스 파일	비고 (출처 사이트 정보 및 캡처 화면)
1	OO SW	v1.0 (2014.04.02. 배포)	MIT	없음	없음	OO.dll OO.h	별첨 1
2	...	...	...	...	...	...	...
3	...	...	...	...	...	...	...
4							
...							

【부록 15】 약어표

약 어 표

약 어	영 문	한 글
ASIC	Application Specific I.C.	주문형 반도체
AV	All View	공통관점
CBD	Component Based Development	컴포넌트기반개발
CDR	Critical Design Review	상세설계검토회의
CDRL	Contract Data Requirement List	계약자료요구목록
COM	Computer Operation Manual	컴퓨터운영매뉴얼
COTS	Commercial Off-The-Shelf	상용제품
CPM	Computer Programming Manual	컴퓨터프로그래밍매뉴얼
CPLD	Complex Programmable Logic Device	복합 프로그램 기능 논리소자
CSC	Computer Software Component	소프트웨어 구성품
CSCI	Computer Software Configuration Item	소프트웨어 형상항목
CSU	Computer Software Unit	단위 소프트웨어
DBDD	Data Base Design Description	데이터베이스설계기술서
DBMS	Data Base Management System	데이터베이스관리시스템
DT&E	Development Test and Evaluation	개발시험평가
EPLD	Erasable Programmable Logic Device	소형 프로그램 기능 논리소자
ERD	Entity Relation Diagram	개체관계도
FCA	Functional Configuration Audit	기능적형상확인
SIG	Software Installation Guide	소프트웨어설치절차서
FPGA	Field Programmable Gate Array	현장 프로그램 가능 게이트 어레이
GOTS	Government off-the-shelf	정부지정
HDL	Hardware Description Language	하드웨어 기술 언어
HWCI	Hardware Configuration Item	하드웨어 형상항목
ICD	Interface Control Document	인터페이스통제문서
IDD	Interface Design Description	인터페이스설계기술서
MND-AF	Ministry of Defense Architecture Framework	국방아키텍처프레임워크
ORD	Operational Requirement Document	운용요구서
OS	Operating System	운영체제

약 어	영 문	한 글
OT&E	Operational Test and Evaluation	운영시험평가
OV	Operational View	운용관점
PCA	Physical Configuration Audit	물리적형상확인
PDR	Preliminary Design Review	기본설계검토회의
PRR	Production Readiness Review	생산준비검토회의
SCOM	System Center Operator Manual	체계운영자지침서
SCS	Software Catalog Specification	소프트웨어목록명세서
SDD	Software Design Description	소프트웨어설계기술서
SDP	Software Development Plan	소프트웨어개발계획서
SFR	System Functional Review	체계 기능 검토회의
SIP	Software Installation Plan	소프트웨어설치계획서
SPS	Software Product Specification	소프트웨어산출물명세서
SRR	System Requirement Review	체계요구사항검토회의
SRS	Software Requirement Specification	소프트웨어요구사항명세서
SSDD	System/Subsystem Design Description	체계설계기술서
SSR	Software Specification Review	소프트웨어요구사항검토회의
SSRS	System/Subsystem Requirement Specification	체계요구사항명세서
STD	Software Test Description	소프트웨어통합시험절차서
STP	Software Test Plan	소프트웨어통합시험계획서
STR	Software Test Report	소프트웨어통합시험결과서
STrP	Software Transition Plan	소프트웨어전이계획서
SUM	Software User Manual	사용자지침서
SV	Systems View	체계관점
SVD	Software Version Description	소프트웨어버전기술서
TEMP	Test and Evaluation Master Plan	시험평가기본계획서
TRR	Test Readiness Review	시험준비검토회의
TV	Technical standards View	기술표준관점

무기체계 소프트웨어 개발 및 관리 매뉴얼

시행일 : 2020년 2월 13일

발행일 : 년 월 일

발행처 : 방위사업청 방위사업정책국 방위사업분석과